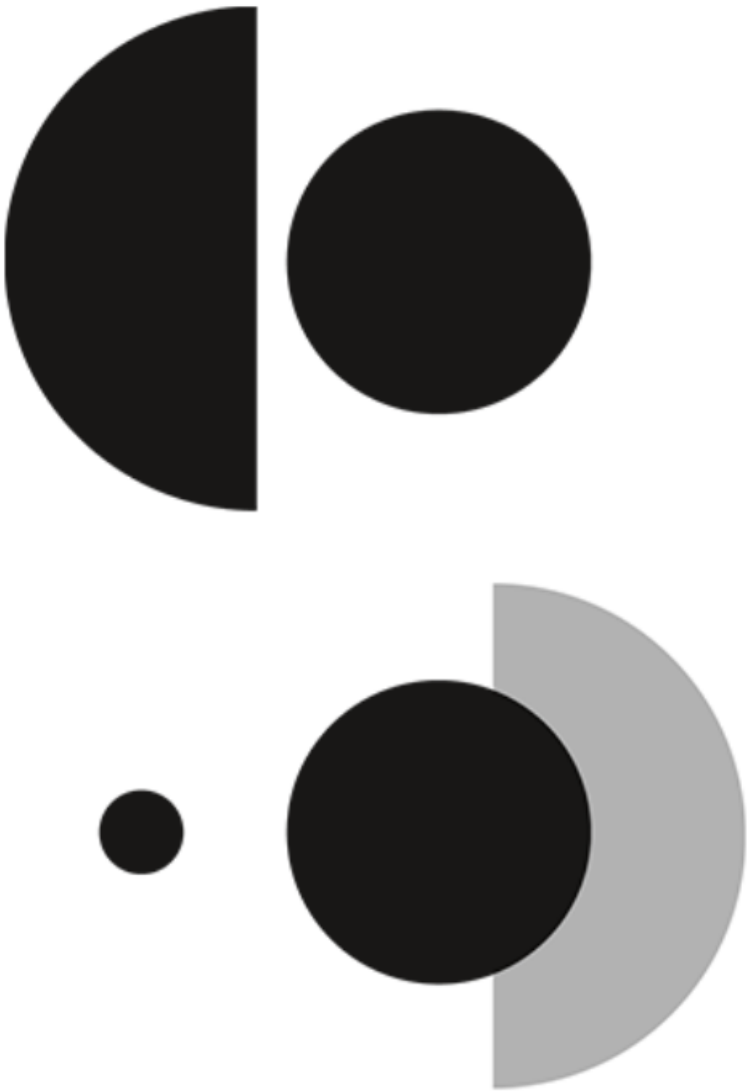


International Computer Music Conference 2026

May 10-16, 2026

Hamburg, Germany

The International Computer Music Association

Board Officers

President of ICMA: Kerry Hagan
Vice President for Membership: Mengjie Maggie Qi
Vice President for Conferences: Rodrigo Cadiz
Vice President for Asia/Oceania: PerMagnus Lindborg
Vice President for the Americas: Eric Honour
Vice President for Europe: Nikos Stavropoulos
Vice President for Preservation: Tae Hong Park
Music Coordinator: Ping Jin
Research Coordinator: Anthony Paul De Ritis
Publications Coordinator: Juan Parra Cancino
Social Media Officer: Patrick Hartono
Treasurer/Secretary: Chryssie Nanou

Board of Directors

At-Large Directors:
Henrik von Coler
Mengjie Maggie Qi
John Thompson
Jiayue Cecilia Wu
Americas Regional Directors:
Paulo C. Chagas
Stephen Leffue
Asia/Oceania Regional Directors:
Marc Battier
Mikako Mizuno
Europe Regional Directors:
Miriam Akkermann
Juan Parra Cancino

Non-Elected Position

ICMA Administrative Assistant Sandra Neal

Program Committee

Hasnizam Abdul Wahid	Universiti Malaysia Sarawak
Miriam Akkermann	TU Dortmund
Jeff Albert	Georgia Institute of Technology
Alejandro Albornoz	The University of Sheffield
Ariadna Alsina Tarres	Université Paris 8
Sebastia Vicenc Amengual Gari	Reality Labs Research
Alessandro Anatrini	Hochschule für Musik und Theater Hamburg
Torsten Anders	Independent
Moreno Andreatta	IRCAM-CNRS-UPMC
Pavlos Antoniadis	University of Ioannina
Micael Antunes Da Silva	University of Campinas
Alyssa Aska	University of Calgary
Nathan Asman	State University of New York (SUNY) at Oneonta
Kevin Austin	retired
Hanae Azuma	Tokyo University of the Arts
Erich Barganier	New York University
Constantin Basica	Constantin Basica
Marc Battier	Sorbonne University
Marlene Behrmann	ligeti center
Greg Beller	synekine
Jonathan Berger	CCRMA - Stanford University
Doug Bielmeier	Northeastern
Michael Blandino	LSU
Georges Bloch	Ircam
Per Bloland	Oberlin Conservatory
Mathieu Bonilla	PC member
James Borchers	Digipen Institute of Technology
Candida Borges	Transeuntis Mundi Institute
Paul Botelho	Bucknell University
Michael Boyd	Chatham University
Hector Bravo Benard	Universidad Nacional Autónoma de México
Benjamin Broening	University of Richmond
Courtney Brown	Center for Creative Computation, Southern Methodist University
Eric Browne	Munster Technological University
George Brunner	Brooklyn College Conservatory of Music
Barbara Bultmann	Ensemble Resonanz
Marco Buongiorno Nardelli	University of North Texas
J. A. Burgoyne	University of Amsterdam
Shane Byrne	Technological University of the Shannon: Midlands Midwest
Daniel Cabanzo	ICMC
Rodrigo Cadiz	Pontificia Universidad Catolica de Chile
Vincent Caers	LUCA school of Arts
David Cardona	Berklee College of Music
De Castellarnau Artero Carlos	PC member
Nicole Carroll	Queensland University of Technology

Guilherme Carvalho	UPV - Montpellier 3
Nicola Casetta	Conservatorio Rossini Pesaro
Jean-Michaël Celerier	LaBRI
Paulo Chagas	University of California Riverside
Arun Chandra	The Evergreen State College
Yuhsin Chang	Macau University of Science and Technology
Jean-Francois Charles	University of Iowa
Axel Chemla-Romeu-Santos	IRCAM
Meiyan Chen	PC member
Yu-Tung Cheng	Brandeis University
James Tsz-Him Cheung	Hochschule für Musik und Theater Hamburg
Daniel Chin	NYU
Ahyeon Choi	Seoul National University
Jaehoon Choi	Rensselaer Polytechnic Institute
Se-Lien Chuang	Atelier Avant Austria
Daniel Ciampolini	
Paul Clift	Hochschule der Künste Bern
Seth Cluett	Columbia University
Douglas Cohen	Conservatory of Music - Brooklyn College
Oscar Corpo	Hochschule für Musik und Theater Hamburg
Pierre Couprie	Centre d'Histoire Culturelle des sociétés contemporaine - Paris-Saclay University
Patti Cudd	University of Wisconsin-River Falls
Kevin Dahan	Université Gustave Eiffel
Shanshan Dai	CHINA INSTITUTE OF ARTS SCIENCE & TECHNOLOGY
Ricardo Dal Farra	Board ISEA International
Roger Dannenberg	Carnegie Mellon University
Rosiane de Freitas	IComp/UFAM
Anthony De Ritis	Northeastern University
Domenico De Simone	Conservatorio di Musica "U. Giordano" di Foggia
Jana De Troyer	Musikhochschule Lübeck
Wayne DeFehr	University of Alberta
Giovanni Dinello	None
James Dooley	The Open University
Paul Doornbusch	home
Krystoffer Dreps	–
Frédéric Dufeu	University of Huddersfield
Rodney Duplessis	WPI
David Durant	University of South Alabama
John Eckhardt	ICMC
Arne Eigenfeldt	Simon Fraser University
Aaron Einbond	City, University of London
Frank Ekeberg	none
Sivan Eldar	IRCAM
Kramer Elwell	University of California Santa Barbara
Alexis Emelianoff	independent artist
Tom Erbe	University of California San Diego

Cagri Erdem	Department of Informatics, University of Oslo
Kerem Ergener	Louisiana State University
Philippe Esling	IRCAM
Georg Essl	University of Wisconsin - Milwaukee
Carl Faia	Art Zoyd
Hongshuo Fan	Texas A&M University
Alireza Farhang	Ircam
Jian Feng	Wuhan Conservatory Of Music
José Miguel Fernandez	Ircam
Iván Ferrer-Orozco	InterAcciona
Ray Fields	University of Maryland
Rikhardur H. Fridriksson	Iceland University of the Arts
Xiao Fu	HfMT Hamburg
Takuto Fukuda	Concordia University
Joshua Jd Fuller	UNT
Marc Garcia Vitoria	Conservatori Superior de Música "Joaquín Rodrigo" de Valencia
Giorgos Gargalas	Manchester University
Travis Garrison	University of Central Missouri
Jason Geistweidt	Art on Wires / The World Opera
Giampiero Gemini	Hermes Intermedia
Marta Gentilucci	ircam
Garrison Gerard	University of North Carolina at Pembroke
Daniele Ghisi	IRCAM
Jean-Louis Giavitto	IRCAM - UMR STMS 9912 CNRS
John Gibson	Indiana University Jacobs School of Music
Nuria Giménez Comas	Composer
Michael Gogins	Irreducible Productions
Pedro González Fernández	Escola Superior de Música de Catalunya (ESMUC)
Matthew Goodheart	Rensselaer Polytechnic Institute
Beller Gregory	ligeti center, HfMT-Hamburg
Thomas Grill	University of Music and Performing Arts Vienna
Fernando Gualda	UFRGS
Anselmo Guerra	Universidade Federal de Goiás
Michael Gurevich	University of Michigan
Ian Guthrie	Kansas City Ballet
Ben Hackbarth	UCSD
Kerry Hagan	University of Illinois
Behzad Haki	Universitat Pompeu Fabra
Lu Han	Communication University of China
James Harley	University of Guelph
Blake Harrison-Lane	OSU
Todd Harrop	Hochschule für Musik und Theater Hamburg
Patrick Hartono	RMIT Vietnam
Test Hase	opel
Akiko Hatakeyama	
Ethan Hayden	Gannon University
Leonard-Nicola Hein	University of Music Lübeck

Benjamin Helmer	Hochschule für Musik und Theater Hamburg
Mara Helmuth	University of Cincinnati
Jonathan Higgins	Morley College London
Revital Hollander	Tel-Aviv University
Eric Honour	University of Central Missouri
Cat Hope	Monash University
Andrew Horner	The Hong Kong University of Science and Technology
Edmund Hunt	Royal Birmingham Conservatoire
Simon Hutchinson	University of New Haven
Jeremy Hyrkas	University of California San Diego
Sun Jung Im	Creative Mulorum
Ryan Ingebritsen	University of Illinois at Chicago/Urbana Champaign
Jeeyun Jang	Blue Garage
Christopher Jette	Center for Computer Research in Music and Acoustics (CCRMA), Stanford
Ping Jin	Music Coordinator of ICMA
Mathias Josefson	HfMT
Haesun Joung	Seoul National University
Julien Junod	tFSc!
Dragica Kahlina	none
Konstantinos Karathanasis	University of Oklahoma
Thorsten Kern	Hamburg University of Technology - Institute for Mechatronics in Mechanics
John C.S. Keston	University of St Thomas
Yonghyun Kim	Georgia Institute of Technology
David Kim-Boyle	The University of Sydney
Jenn Kirby	University of Liverpool
Hiromichi Kitazume	Hokkaido University of Education
Carmen Kleykens Vidal	Hochschule für Musik und Theater Hamburg
E. D. Koch	HfMT Hamburg
Yuriko Hase Kojima	Shobi University
Panayiotis Kokoras	University of North Texas
Tomás Koljatic	Pontificia Universidad Católica de Chile
Johannes Kretz	University of Music and Performing Arts Vienna
Matthias Krüger	Hochschule für Musik und Theater Hamburg
Vlatko Kucan	Hochshule für Musik und Theater Hamburg
Mikel Kuehn	Bowling Green State University
Kris Kuldkepp	Musikhochschule Lübeck
Mathieu Lacroix	University of Inland Norway
Leigh Landy	De Montfort University
Man Hei Law	Hong Kong University of Science and Technology
Frederic Le Bel	Ircam - Paris 8
Lukas Leck	independent
Steven Leffue	Pure Data dude
Sascha Lemke	HfMT
Eric Lemmon	Stony Brook University
Serge Lemouton	IRCAM
Pak Hei Leung	University of North Texas

Steven Lewis	University of California, Irvine
Kelian Li	George Mason University
Tao Li	University of Oregon
Xiaosha Li	Georgia Institute of Technology
Lin-Ni Liao	IReMus (Université Paris-Sorbonne - CNRS)
Tzu-Ning Liao	Helios Klinik
Yi-Cheng Lin	Computer Society of the Republic of China
Simon Linke	HAW Hamburg
Steffen Lohrey	Audio Engineer
Tade Lorenzen	HAW Hamburg
Eduardo Loria Lazcano	Hochschule für Musik und Theater Hamburg
Janina Luckow	ligeti zentrum
Pieter-Jan Maes	IPEM
Sanjay Majumder	Style tree
Raul Masu	ITI/LARSyS, FCT/NOVA
Tomoya Matsuura	Kyushu University
Benjamin Matuszewski	IRCAM
Andrew May	University of North Texas
Kieran McAuliffe	University of Cincinnati
Daniel McKemie	Independent
Robin Meier	
Christian Meznaric	ligeti zentrum
Romain Michon	INRIA
Scott Miller	St. Cloud State University
Lejun Min	CCRMA, Stanford University
Adam Mirza	Emory University
Daichi Misawa	0-9studio
Mikako Mizuno	Nagoya City University
Ronald Mo	University of Sunderland
John Moeller	Texas A&M University
Paolo Montella	Krich
Barry Moon	Arizona State University
Ted Moore	tedmooremusic
Takeyoshi Mori	Senzoku Gakuen College of Music
Jeffrey Morris	Texas A&M University
Tilemachos Moussas	PhD Candidate , Departement of Music Studies, University of Athens
Yunze Mu	University of Cincinnati
Jeremy Muller	Jeremy Muller
Chrysi Nanou	RPI
Luis Naon	Hes-SO Haute Ecole de Musique Geneva
Treya Nash	Louisiana State University
Fernanda Navarro	UCSC
Quirin Nebas	HAW Hamburg
Jon Nelson	University of North Texas
Peter Nelson	The University of Edinburgh
Max Neupert	The Center for Haptic Audio Interaction Research
Robert Newcomb	Partial Music

Charles Nichols	Virginia Tech
Pierre-Valery Njenji Tchetgen	Northeastern University
Julia Nordholz	Multimedia Composition HFMT
Dave O Mahony	ISSTA
Nicolas Obin	IRCAM
Ryan Olivier	Indiana University South Bend
Konstantina Orlandatou	University of Hamburg
Ken Paoli	College of DuPage
Charilaos Papaioannou	National Technical University of Athens
Orestis Papaioannou	Composer
Juan Parra Cancino	ORCiM-Orpheus Institute, Gent, Belgium
Dave Payling	Staffordshire University
Scott Petersen	Yale University
Sean Peuquet	Rocky Mountain College of Art + Design
Isabel Pires	CESEM - Faculdade de Ciências Sociais e Humanas, Universidade Nova de Lisboa
Sam Pluta	University of Chicago
Stephen Pope	University of California
Laurent Pottier	UJM-CIEREC
Bastien Pouillès	PC member
Miller Puckette	University of California San Diego
Ninad Puranik	McGill University
Marcin Paczkowski	Cornish College of the Arts at Seattle University
Maggie Qi	China Conservatory of Music, China
Raphael Radna	UC Santa Barbara
Izzi Ramkissoon	NYU NYCEMF
Todd Reed	Retired
David Reeder	Independent composer; media arts developer
Leah Reid	University of Virginia
Jaime Reis	ESML - Escola Superior de Música de Lisboa
Christian Renz	HfMT, Hamburg / Universidad del Norte, Barranquilla
Heike Ressel	
Michael Rhoades	The Perception Factory
Martin Ritter	GONAD
David Rizo	University of Alicante
Bradley Robin	DePaul University
Stephen Roddy	Trinity College Dublin
Christiana Rose	Independent
Sophie Rose	University of Melbourne
Robert Rowe	New York University
Joran Rudi	NTNU
Gabriel Saber	afiliation
Marcel Zaes Sagesser	University of Applied Sciences St. Polten
Jake Sandridge	Rice University
Mayank Sanganeria	Independent
Richard Savery	Georgia Institute of Technology
Carla Scaletti	Symbolic Sound Corporation
Marlon Schumacher	HfM Karlsruhe

Nadine Schwalb	ligeti zentrum
Diemo Schwarz	Ircam - CNRS STMS
Robert Seaback	Norwegian Academy of Music
Ambrose Seddon	Bournemouth University
Aigerim Seilova	HfMT Hamburg
Jacob Sello	HfMT
Jiwon Seo	composer
Serkan Sevilgen	Istanbul Technical University
Kyle Shaw	California State University, Bakersfield
Eric Sheffield	Louisiana State University
Seongah Shin	Keimyung University
Kang Shinae	Tokyo University of the Arts
Ryne Siesky	Ryne J.S. Music
Alexander Sigman	Keimyung University
Eric Simonson	Danville Area Community College
Alexandria Smith	Georgia Institute of Technology
Andrew Smith	SUNY Fredonia
Isaac Smith	Indiana University
Walker Smith	Stanford University
Rick Snow	Tulane University
Paolo Solcia	ICMC2026
Adam Stanovic	London College of Communication, The University of the Arts
Nikos Stavropoulos	Leeds School of Arts
D.Andrew Stewart	UNIVERSITY OF LETHBRIDGE
Li Su	Academia Sinica
Leon Sudahl	HAW Hamburg
Jacob Sudol	FIU
Zachary Sussman	Self
Marco Suárez-Cifuentes	Chaire art et Sciences
Akira Takaoka	J. F. Oberlin University
John Thompson	Georgia Southern
Seth Thorn	School of Arts, Media + Engineering, Arizona State University
Vincent Tiffon	Université de Lille-Nord de France / Ircam
Sever Tipei	University of Illinois at Urbana-Champaign
Todor Todoroff	UMONS/TCTS / ARTeM
Jutta Toelle	HfMT
Christopher Trapani	Columbia University
Pierre Alexandre Tremblay	Conservatorio della Svizzera italiana
Christian Tschirner	ligeti zentrum
Hans Tutschku	Harvard University
Ken Ueno	UC Berkeley
Kari Vakeva	(independent composer)
Akito van Troyer	Massachusetts Institute of Technology
Kyle Vanderburg	NoteForge
Juan Carlos Vasquez	Xi'an Jiaotong Liverpool University
Julien Vincenot	Harvard University
Henrik von Coler	Georgia Tech
Genoel Von-Lilienstern	

Jack Walker	The University of Edinburgh
Rodney Waschka	North Carolina State University
Johanna Wegerich	external
Bing Wei	The Central Conservatory of Music
Andreas Weixler	Anton Bruckner Privatuniversität
Joana Welteke	ligeti zentrum
Sunhuimei Xia	Wuhan Conservatory of Music
Keisuke Yagisawa	Tamagawa_university
Qiaoyu Yang	Georgia Institute of Technology
Wanjun Yang	Sichuan Conservatory of Music
Pingan Yao	Campion School
Victor Zappi	Northeastern University
Xinran Zhang	Central Conservatory of Music
Dong Zhou	Leuphana University
Qian Zhou	Shanghai Conservatory
Julie Zhu	University of Michigan
Anil Çamci	University of Michigan
Zeynep Özcan	University of Michigan

Organizing committee



Prof. Dr. Georg Hajdu

Conference Chair, ICMC HAMBURG 2026



Prof. Dr. Miriam Akkerman

Paper Chair, ICMC HAMBURG 2026



Dr. Konstantina Orlandatou

Paper Co-Chair, ICMC HAMBURG 2026



Dr. Xiao Fu

Music Chair, ICMC HAMBURG 2026



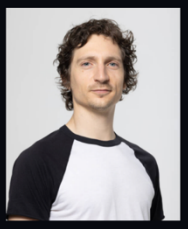
Dr. Greg Beller

Music Co-Chair, ICMC HAMBURG 2026



Prof. Dr. Jacob Sello

Installation Chair, ICMC HAMBURG 2026



Quirin Nebas

Installation Co-Chair, ICMC HAMBURG 2026



Dr. Simon Linke

Workshop and Demo Chair, ICMC HAMBURG 2026



Christine Preuschl

Organization Chair, ICMC HAMBURG 2026



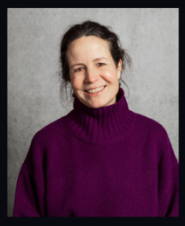
Benjamin Helmer

Organization Co-Chair, ICMC HAMBURG 2026



Prof. Dr.-Ing. Thorsten A. Kern

Finance Chair, ICMC HAMBURG 2026



Tanja Tangermann

Finance Co-Chair, ICMC HAMBURG 2026



Ali Elnwegy

Sponsoring Chair, ICMC HAMBURG 2026



Nadine Schwalb

Off-ICMC Chair, ICMC HAMBURG 2026



Joana Naomi Welteke

Off-ICMC Chair, ICMC HAMBURG 2026



Christian Tschirner

Off-ICMC Chair, ICMC HAMBURG 2026



Daria Radler

Communication, ICMC HAMBURG 2026

Klara Janina Luckow

Communication Design, ICMC HAMBURG 2026

Preface

Ladies and Gentlemen, dear participants!

Welcome to the 51st edition of the International Computer Music Conference, and–all good things come in threes–the third one on German soil. My teacher and friend, the composer and computer music pioneer Clarence Barlow, was the one who brought the first German edition of the ICMC to Cologne in 1988 – an edition in which I participated as a student and helping hand. Clarence would have turned 80 recently, and we commemorate his legacy with a performance of *Im Januar am Nil*. We are also delighted to welcome another pioneering figure of computer music, James (Andy) Moorer, who recently celebrated his 80th birthday and will present his work on two occasions. And we are deeply honored that John Chowning will join us again from California.

On behalf of György Ligeti, Chowning and Moorer were involved in the planning of the first German computer music center in the mid-1970s – a center that would bring the spirit of Stanford's Artificial Intelligence Laboratory to Germany.

Ligeti's idea was finally realized on occasion of his 100th birthday in 2023 in the form of the ligeti center whose name also stands for Laboratories for Innovation and General-audience Edification through the Translation of Ideas. The conference motto thus is, with slight modification: innovation, translation, participation, emphasizing the idea that art and research need to reach the minds and hearts of the public to stay relevant. We therefore revived the concept of the Off-ICMC offering a translation format for larger audiences. Our keynotes address questions of translation as exemplified in the work of Robert Henke, also co-founder of Ableton, the company behind Ableton Live. With Falk Hübner's keynote we are adding a new facet to the ICMC: namely, artistic research. Artistic research is a growing field combining art and research, reframing how we understand both human endeavors and their mutual interdependence. Music can be so much: means and interface of human expression, subject of investigation and mediation as well as data model for representing sound and musical structure; but with its profound impact on our mind and mood it can also be used to improve individual and societal wellbeing. The cognitive psychologist Psyche Loui will shed light on this in her keynote which, unfortunately, she won't be able to deliver in person. A special panel on music, technology and the mind will follow right after.

As you will see in the program, there will be an abundance of events, starting with the opening concert at the Elbphilharmonie on Sunday, May 10, 2026 and ending with a farewell at the Stellwerk on Saturday. We are particularly pleased to offer an excursion to Lübeck, once the capital of the Hanseatic League, and I'd like to thank my colleague Nicola Hein from the Lübeck Academy of Music to make this possible. This is also a great opportunity to express, as conference chair, my gratitude to a long list of people and institutions who have contributed to the realization of the ICMC 2026:

Foremost the chairs and co-chairs, the ligeti center's production team, the Elbphilharmonie, Ensemble Resonanz, our soloists, the Ensemble 404, the SPIIC ensemble, the technical teams, the leaderships of the universities involved, the board of the ICMA, the countless reviewers, the district of Harburg, the Hamburg Authority for Science, Research and Equality, the Hamburg Authority for Culture and Media and our sponsors.

And my thanks also go out to you, dear audience, for demonstrating, by your presence and your contributions to the conference, how important our topic is in in this new era of artificial intelligence and autonomous systems.

Enjoy the concert and the conference! We will offer more information about its logistics right after the concert and on Monday at 3:30PM before Robert Henke's keynote.

May 10, 2026
Berlin

Georg Hajdu (conference chair)

Table of Contents – **Part I – Papers**

Computer-Aided Composition: A Retrospective and Prospective Outlook 1
Andrea Agostini

MBHD: A Modular Audio Playback and Manipulation System for Loop-Based Performance 8
Neal Anderson and Sanjay Majumder

AI Framework for Dynamic Robotic Instrument Calibration 16
Colton Arnold, Zhaohan Cheng and Ajay Kapur

Studio Report: A Third-Order Periphonic Ambisonics System for Teaching and Research at FEUP and INESC TEC's SMC Lab 20
José Ricardo Barboza and Gilberto Bernardes

Data-driven algorithmic composition with large sample libraries: a modular system for the dynamic formation and control of spatialised sound groups 24
Nikos Baskozos and Thanos Polymeneas-Liontiris

Hyponoia: An Affective Computing System for Augmented Musical Performance – A Case Study 28
Penelope Bekiari and Anastasia Georgaki

Perpetual Dialogues: A Computational Analysis of Voice–Guitar Interaction in Carlos Paredes's Discography 35
Gilberto Bernardes, Nádia Moura and António Sá Pinto

ChoraleWind: An Expressive Wind-Quartet Dataset for End-to-End Rendering from the Neues Thüringer Choralbuch 43
Axel Berndt, Aida Amiryan-Stein, Manuel Peters, Meinard Müller and Stefan Balke

Automatic Hybrid Following in Real-Time Mixed Music A Case Study with Antescofo and ipt for Flute Playing Techniques 51
Nicolas Brochec and Jean-Louis Giavitto

20 Years Zirkonium and Klangdom at ZKM 59
Ludger Brümmer, Götz Dipper and Dan Wilcox

Composer-in-the-Loop Generation of Motivic Variants Using State-Space Models and Preference Learning 67
Rodrigo Cadiz

BIKES: A Mobile Networked Music Instrument in Interdisciplinary Research and Education 75
Laura Call Gomez, Gabriel Decker, Jayson Faupel, Aditya Rajesh Pawar, Jacob Westerstahl and Henrik von Coler

Scores That Run: Graphic Notation with Embedded Performance Semantics 81
Rob Canning

MetaConcert: A Shared VR Audio-Visual Experience Model Reducing User Isolation Through Synchronized 360 Video on HMDs and HOA Playback on a Multichannel Dome . 89
Mauro Cantonetti, Paolo Malpeli, Giuseppe Rizzo and Alessandro Anatrini

Sonic Urgency: Exploring Perceptual, Sociopolitical, and Participatory Dimensions of Spatial Listening 94
Teresa Carrasco

Beyond Execution: Unrealizability and the Ontology of Sound in Computer Music 102
Paulo C. Chagas

Classical Music Mashup System and Compatibility Heuristics 110
Yu Foon Darin Chau and Andrew Horner

Exploring Masked CE Losses to Enhance Word Offset Estimation in CTC-based Lyrics-to-Audio Alignment 114
Tian Cheng, Tomoyasu Nakano and Masataka Goto

Latent Music: Emergent Sonic Forms and Sonic Liminality in Text-to-Audio Systems 122
Guilherme Coelho

Amplifying Participation. The digital Barlow Archive (dBA) as an Approach to the Recording of a Digital Computer Music Legacy 132
Fabian Czolbe, Julian Rohrhuber and Bernd Härpfer

Pd++: A C++ Library of Pure Data's DSP Objects 136
Robert Esler

DSP and the Metalevel: Clamps - an integrated environment for algorithmic composition and interactive realtime performance 144
Orm Finnendahl

Real-Time, Low-Latency, High Resolution Audio Spectral Analysis: Phase Matters 152
Alexandre Francois

Magnetic Memory Rushnyk 160
Jocelyn Ho, Margaret Schedel and Sofy Yuditskaya

Vibrato Matching for Modulation Control and Blending in Sound Mixtures 166
Jeremy Hyrkas

Building Loopers 172
Jeff Kaiser and Gregory Taylor

Ponticello: An Interactive Conducting System for Mixed Music Performance 176
Nikolaus Knop

Factor Analysis of Similarity in the Same Orchestral Piece 182
Minami Kojima, Takayuki Itoh and Rafael Ramirez

Tree Rings: Ecological Memory and Linguistic Traces in an Immersive Dome Composition 190
Yu Chia Kuo

The History of Japanese Electroacoustic Music for Piano from the Perspective of Media Genealogy 195
Hyunmook Lim

Beyond Musique Concrète: Perceptual Morphing via Audio Latent Embeddings Manipulation 203
Xiangbin Lin, Du Huang, Qi Qian and Maosong Sun

Designing responsive rhythms utilizing the Impulse Pattern Formulation (IPF) 211
Simon Linke, Rolf Bader and Robert Mores

Extending Xenakis: From Architectural Geometry to Sonification of the Philips Pavilion .. 219
Changda Ma, Sunshiyu Wang, Canting Zhu and Alexandria Smith

Studio Report: Center for Computer Music 2025–2026 227
Hefang Ma, Jingyu Luo, Paul Francis, Mara Helmuth, Sangbong Nam and Wei-Huai Chen

De/Re:Generation: Exploring DIY Cassava-Starch Bioplastic Interfaces with EFT-Based Touch Sensing in an Interactive Sound Installation 231
Adriano C. Monteiro and Rafaela B. Pires

Studio Report: Laboratory of Advanced Music Production, Senzoku Gakuen College of Music 239
Takeyoshi Mori

Co-Composing with Plants: Early Experiments in Bio-Responsive Score Design 243
Solomiya Moroz, Nicolo Merendino and Massimo Sterlino

Acoustic Interactive Sand Tray Therapy System: An Embodied Interface for Multisensory Sound Interaction 251
Yunze Mu, Lorna Segall and Zhixin Xu

The Singing Skin: An Audience-Centered Biofeedback System for Musical Interaction Based on Galvanic Skin Response 255
Eun Ji Oh, Jun Woo Beck and Alexandria Smith

Emotion-Based Film Music Retrieval with Handcrafted and Deep Models 263
Lucas Ong, Ruby Crocker and George Fazekas

Vibrational Analysis of Traditional Persian Kamanche Sound Box: Experimental and Computational Investigation of Structural Modifications 271
Amir Abbas Orouji, Ayoub Banoushi and Gilberto Bernardes

BbMuse: A Blackboard-Driven Framework for Real-Time Interactive Music 275
Fabian Ostermann

Listening Across Spaces: Perceptual Evaluation of an Ambisonics-Based Sound Installation 283
Felipe Otondo and Leonardo Santos

Computational Analysis of Expressive Tempo in Irish Traditional Dance Music 287
Mário Pereira, António Sá Pinto, Treasa Harkin and Gilberto Bernardes

The MMMAudio Computer Music Environment 295
Sam Pluta and Ted Moore

DANCING CABIRIA: AN HYPER-ENVIRONMENT STUDY THROUGH CORPUS-BASED TECHNIQUES 303
Piero Poli

Music of Changing Lines: Toward a Culturally Situated Approach to the I-Ching 311
Ling Qi, Teng Ma and Alexandria Smith

Reimagining Athletic Gesture: Transforming Basketball Sound into Narrative Electroacoustic Music 317
Guanjun Qin, Yunxuan Jia and Neal Farwell

SARC Studio Report 321
Pedro Rebelo and Craig Jackson

Supervised Memory: How Machines Can Preserve What We Cannot Hold 325
Giovanni Roma and Alba Francesca Battista

CALM: Translating Somatic Experience into Compositional Structure as a Trauma-Informed Methodology 333
Sophie Rose

Snapping Matters: Context-Aware Onset Refinement for Automatic Music Transcription . 341
Abhirup Saha, Hans-Ulrich Berendes, Meinard Müller and Ben Maman

Spores: A Physarum-Inspired Instrument for Agent-Based Ecological Interaction 349
Kyle Smith and Alexandria Smith

oljud–nq(n): A Sonic Manifesto of Resistance to Generative AI in Music 354
Bob Sturm and Elin Kanhov

Linear A: A Composer's Integrated Workspace 362
Christopher Trapani

Recursive Radiance: Multimedia Interpretations of Traditional Chinese Aesthetics 365
Juan Carlos Vasquez and Zhonghao Chen

Gestalt: A Symbiotic Framework for Real-Time Collaboration between Performers and Mass Audiences 373
Sitong Wu and Jinshuo Feng

OrbitScore: A Domain-Specific Language for Polymetric Live Coding Based on Multilayered Temporal Structures 381
Hiroshi Yamato

CNN-BiLSTM Hybrid Model with Physical Constraints for Automatic Piano Fingering Generation 389
Tianze Zhang, Shingyui He and Lei Xuan

Hexagram-Based Semantic Composition: Discretizing Embedding Spaces into Symbolic Compositional States for Improvised Performance 397
Yuan Zhang and Xinran Zhang

Part II – Innovation Showcase

Papers Innovation Showcase 400

Computer-Aided Composition: A Retrospective and Prospective Outlook

Andrea Agostini
Orpheus Institute Ghent
Antwerp University and Royal Conservatoire
andrea.agostini@orpheusinstituut.be

ABSTRACT

Computer-aided composition was established as an autonomous discipline, distinct from the seemingly more general concept of computer composition, in the 1980s. Since then, it has prompted the development of dedicated software tools and specific compositional practices and attitudes. In spite of this, a definition of what computer-aided composition actually is and, subsequently, a retrospective outlook on its past evolution and a prospective one on its possible futures has seldom if ever been attempted. Also, while development and adoption of new tools has been uninterrupted through the decades, theoretical reflection was especially thriving until the late 1990s or early 2000s, and has lost vitality since.

In this article, we shall examine past literature in order to trace a historical overview of the term, implicitly outlining a tentative definition of it and following through the most significant developments of computer-aided composition and its associated toolsets; attempt a necessarily partial overview of how it is practically understood and adopted today; and sketch a personal and incomplete wishlist of what the term could come to mean in some desirable future.

1. INTRODUCTION: A HISTORY OF THE CONCEPT OF COMPUTER-AIDED COMPOSITION

The first mentions of the term *Computer-aided composition* (CAC for short) can, to my best knowledge, be traced back to some articles and internal reports written at Ircam starting from the mid-1980s [1]. Initially, it was strongly influenced by Fred Lerdahl and Ray Jackendoff's seminal text *A Generative Theory of Tonal Music* [2], but it quickly evolved to be associated with a quite well-defined set of tools, practices and aesthetic postures mostly related to post-serialism, spectralism and, to a limited extent, alea. Broadly speaking, this 'historicised' conception of computer-aided composition can be seen as an approach to compositional formalisation of contemporary classical vocal and instrumental music. According to Gérard Assayag [3], there are a few essential axes to this, which we shall review in the following.

Copyright: ©2026 Andrea Agostini. This is an open-access article distributed under the terms of the Creative Commons Attribution License 3.0 Unported, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

1.1 Notation

CAC, at least in its original conception, is aimed at producing music based upon traditional Western notation, with an almost exclusive focus on the fundamental parameters of pitch (mostly restricted to octaviant tempered tunings) and rhythm (organised according to the divisional principles that are central to Western music theory). CAC systems must thus account for the ability to produce readable musical notation and, in more advanced cases, to allow user interaction through the graphical manipulation of musical notation.

This primacy of pitch and rhythm has strong aesthetic implications, in the hierarchy of musical parameters that it establishes. It has often been said that the 20th Century has seen the emancipation of timbre (see for example [4]), as well as its rise to a prominent role (in some not sporadic cases, the most prominent one) in the musical work. Historically, CAC seems to have been rather oblivious of this paradigm change. [5] This has surely been motivated by the fact that pitch and rhythm are much easier to measure and express numerically than timbre—a consideration that was even truer at a time when research on timbre descriptors was still in its infancy. [1] On the other hand, the strong focus on those 'traditional' parameters was far from being neutral from a musical point of view. How much this encouraged composers with certain specific aesthetic preferences (and belonging to certain demographics) to gravitate towards CAC while pushing back others and, conversely, how much this influenced the compositional language of composers who chose to adopt CAC in their compositional practice has been amply discussed. [7, 8]

As a side note, said primacy of pitch and rhythm reflects the overarching idea that those parameters constitute an 'alphabet' on top of which a grammar, a syntax and a semiotic system can be erected. This idea is highly relevant in the discourse about common-practice music, especially from the Baroque, Classical and Romantic eras, and has informed the thinking of composers well beyond those times (see for example [9]). The fact that, especially in the discourse about computer music, notation is often referred to as *symbol* and opposed to *sound* or, sometimes, *signal* [2] is related to this view, which Fabien Lévy, expanding on Jacques Derrida's theories, has called *grammatological*. [11]

[1] Famously, Albert Bregman called timbre an *ill-defined wastebasket category*.[6]

[2] See for example the Computer Music Journal issue titled *Narrowing the Gap between Symbol and Sound*. [10]

1.2 Sound

The fact that materials for symbolic (in the above sense) composition can be obtained from sound analysis is the central tenet of the Spectralist movement [12] and has had a wide-ranging influence that still informs the practice of many a prominent composer. Spectralism was a highly influent compositional posture in France in the 1980s. It is thus not surprising that the idea of linking sound analysis and notational representation has been regarded as a crucial one in CAC, at least in the context of Ircam where the term took hold, since the aforementioned early articles and reports. [3]

This 'spectralist' view shifts the focus from the traditional parameters of the composition—the aforementioned pitch and rhythm—to timbre as the foundational aspect of the musical composition. Rather than operating through combinatorial and arithmetic processes in the pitch space, spectralism tends to favour operations in the frequency space, such as frequency shifts, generation of (potentially distorted) harmonic series, harmonic processes inspired by ring and frequency modulations, and so on. Such operations tend to be very tedious and error-prone if carried out with pencil and paper: the adoption of computers was especially welcome for composers who habitually employed them, and in fact Tristan Murail's [4] collection of software implementations of spectral and frequential processes, named *Esquisse*, is still widely used. As a final note, it can probably be debated whether in this case the computer should be regarded as a facilitator or an enabler of these specific practices.

Conversely, the idea of composing sound through its constitutive parameters (as opposed to composing *with* or *through* sound, as is the case in the classic spectralist view) has been one of the most important driving forces in the development of computer music. Computer music languages, such as those in the lineage of Max Mathew's MUSIC N, easily allowed one to algorithmically generate electronic scores in the form of text files, or to conceive electronic instruments capable of rich and complex low-level behaviours in response to simple, high-level inputs. Such approaches still constitute a significant facet of computer composition at large. On the other hand, computer-aided composition of music combining electronic sounds with traditional notation for instruments and voices ideally requires comprehensive tools that can address both sides of the problem. Although this had been on the roadmap since the very beginning, it only became reality with the sound manipulation capabilities of PWGL and OpenMusic, in the late 1990s.

Composing scores whose primary parameter is timbre rather than pitch is yet another problem. It potentially requires not only a new system of graphical signs dealing with descriptive and prescriptive aspects of timbre-based composition, but a set of novel concepts as well. The dimensionality of the performative parameter space can grow much higher, and the relation of parameters with time potentially requires the ability to represent continuous evolution, rather than the discretisation that is a fundamental feature of the grammatology of pitch. As of now, support for continuity in the above sense is lacking or haphazard not only in specialised software tools like OpenMusic, but in notation software as well. [5]

1.3 Language

Early CAC systems, such as preForm/Esquisse [3], consisted essentially of ready-to-use implementations of specific, codified musical processes. Subsequent research and reflection showed that musical formalisation can hardly be reduced to a set of parametrisable 'black boxes'. Rather, at least in some instances, formalisation is where the creative agency of the composer actually takes place, and it is therefore necessary that composers be enabled to express their own compositional processes in the easiest possible way, all while maintaining the highest flexibility.

At the very core of subsequent CAC systems lies the possibility for users to generate and process symbolic musical representation through algorithms expressed in a computer language. Whereas, in principle, any Turing-complete language can be used for this, the goal of CAC tools is to provide composers with computer systems already endowed with relevant building blocks, in term of both data representation and processing. Such building blocks may include data types and classes tailored for representing anything from basic parameters such as pitch and rhythm to complete scores; tools for performing fundamental, commonplace operations such as conversion from pitch to frequency space and viceversa; special programming building blocks that pertain to very particular areas of computer science but are specifically useful for implementing certain musical processes (for example, constraint programming, an advanced problem-solving technique that seems to be very apt to solving harmonic and rhythmic problems in a way that composers feel as 'natural' [13, 14]); and, of course, the already mentioned abilities to display musical notation, process data from sound analysis and potentially much more. Moreover, the urge was felt to save composers from the burden of learning a 'difficult' language, such as C, requiring to take care of minute details that do not immediately pertain to the musical problem they wish to implement. Ideally, the closer the representation could be to natural human language, the better.

The computer language that was chosen at the time was Lisp — a choice motivated by a number of factors, including the facts that Lisp had been for decades the language of choice for artificial intelligence and symbolic processing; and that the way it represents processes and data is widely regarded as elegant, economical and highly extendable. Even today, at least two major CAC systems, OpenMusic and Opusmodus, are based on Lisp, and the influence of Lisp is tangible in the *bach* system for Max and, even more so, in the *bach*-based collection of CAC processes *MOZ'Lib*. [15, 16]

As a matter of fact, the fact that CAC has come to be mostly associated with the idea of computerised musical

[3] The control of sound generation and processing through higher level representations had long been commonplace, dating back at least to the Music III program and its concept of *score*.

[4] Tristan Murail (b. 1943) is a prominent composer often associated with the first wave of the spectralist movement, alongside Gérard Grisey and Hugues Dufourt. It should be remarked that he rejects the label 'spectral' for his own music, preferring to call it 'frequential'.

[5] Many composers with a strong interest in timbre and gesture use vector graphics software such as Adobe Illustrator or Inkscape alongside notation tools such as Sibelius and Dorico for authoring their scores.

formalisation for symbolic scores essentially stems from this strong focus on computer language. It should be remarked that the idea that modernist composition is intrinsically formalised was a prominent one in the context of IRCAM in the 1980s, also because of the strong serialist legacy that was endorsed by Ircam's founder and director, Pierre Boulez.

1.4 Models

Perhaps, the most striking side to CAC, as described in the aforementioned papers, was the idea that, through CAC tools, composers would not just produce a composition, but rather set up a model capable of generating multiple, and potentially infinite, reifications [6] of a compositional idea. This would happen through the definition of parameters controlling aspects of the compositional outcome.

Such a view of the compositional process has strong implications that were not fully discussed in the original texts. For one thing, it assumes, at least to a certain extent, that composition happens in a top-down fashion, by which changing at least certain details of the musical work without renegotiating its overarching features is generally meaningful.

Also, it opens the view of the musical work as an instance of an array of alternative versions, among which the composer chooses one after careful evaluation.

2. EVOLUTIONS OF THE CONCEPT IN THE BROADER LANDSCAPE OF COMPUTER MUSIC

All those various considerations stemmed from research and experiments that had been carried out since the early 1980s and had led to the conception and development of tools such as preForm/Esquisse and eventually PatchWork, which laid the foundation for arguably the most popular CAC tools to come, OpenMusic and PWGL. These three programs share very similar architectures, based upon a graphical, node-based representation of the Lisp language and integrating musical notation editors.

2.1 A Revolution in Computer Music

The few years leading from preForm/Esquisse to OpenMusic witnessed a revolution in the practice of computer music, which went from a lab-coat activity, carried out in specialised research centres such as Ircam, to a craft that anyone with a personal computer could practice. For sure, the processing power of a late-1990s computer was orders of magnitude smaller than that of today's most frugal machines. Nonetheless, the kinds of operation that we tend to give for granted now, such as computerised musical engraving, non-destructive sound editing, real-time sound synthesis and processing, and so on, became available on personal computers precisely at that time. [7] [20] The ability to work in real time appeared to be especially attractive for music creators. Composition with live electronics became available to anyone with a personal computer, as well as advanced real-time MIDI processing. Laptop music, with a wildly varied scene of devoted musicians, ensembles and aficionados, became one of the staples of experimental music at the turn of the millennium. Realtime audio plug-ins became the go-to infrastructure for audio processing, even in non-realtime production settings, thanks to the intuitiveness of operation and the strong operational resemblance to hardware studio devices: not by chance, this was also the time when most professional audio productions started to be realised on computers—mostly personal ones, albeit often equipped with expensive professional audio interfaces—and, concurrently, personal computer-based home studios of sometimes near-professional quality became widely affordable and non-professional production of recorded music witnessed an explosive increase in both amount and technical quality.

2.2 Repercussions on CAC

The ripples of this paradigm shift took some time to propagate to the field of CAC. As mentioned above, OpenMusic and, to a certain extent, PWGL introduced the ability to work with audio materials. This enabled users to not only organise sound-based compositions through the affordances of computer languages, but also actively bridge sound- and symbol-based compositional paradigms, thanks to the possibility of using all the software facilities originally conceived for symbols to drive the sound-specific subsystem of OpenMusic.

Another significant addition to the OpenMusic environment was the *Maquette*, a graphical environment for organising formal structures by moving graphical blocks, each representing a compositional process, on a timeline. Significantly, the graphical properties of each block can be retrieved by the process it represents, thus allowing users to adopt a much more intuitive, high-level approach to musical composition. On the other hand, the Maquette is firmly rooted in the overarching principle of computer-aided composition as parametric, a fortiori in the case where the model parameters are linked to the graphical appearance of each block.

Furthermore, the reflection on the nature of real-time operations and the opportunities afforded by blending the realtime and non-realtime approaches were investigated through research projects such as FTM [21], Antescofo [22] and MuBu [23]. On the other hand, the classic opposition of score and orchestra [8] in computer music waxed and waned. It waxed, because typical digital audio workstations such as ProTools, Logic and Digital Performer are essentially based upon it, with their score being the timeline and their orchestra being the various sound and MIDI processors, which are just a modernised interpretation of the concept of unit generators. It waned, because alterna-

[6] The term *reification* is not used in the original papers, which do not address the fundamental question whether, how, and to what extent CAC changes the essence of the musical work and the act of musicking. In fact, with Lydia Goher's *The Imaginary Museum of Musical Works* [17] and Christopher Small's *Musicking* [18] yet to appear, the discussion on this subject seemed relegated to philosophical circles. [19]

[7] The advent of contemporary machine learning techniques, neural networks, large language models and so-called modern artificial intelligence can be seen as a game changer, and probably is. As the subject is controversial and potentially raises important questions about the meaning itself of music making, I am leaving it aside for now.

[8] *Score* and *orchestra* are the central concepts in Barry Vercoe's Csound programming language, as well as in its Max Mathews' predecessors MUSIC III, MUSIC IV and MUSIC V. In short, the orchestra contains the definition of a set of software modules for sound generation and processing, whereas the score represents a timeline of activations and parameters for them.

tive paradigms were proposed by tools such as SuperCollider, Max and PureData. And it somehow waxed again in Max, which, albeit programmatically rejecting it, integrated a *timeline* object until version 4, and then superseded it by the integration of Max itself within the Ableton Live environment, which in turn implements a very interesting and flexible mixture of realtime, non-realtime, linear and non-linear representation of time.

2.3 bach: automated composer's helper

In 2010, I embarked with my friend and colleague Daniele Ghisi in what, in retrospect, may have been a significant development in the field of computer-aided composition, that is, the *bach: automated composer's helper* extension for the Max environment. [9] [15] *bach* was conceived as a hub of different approaches to computer music that Daniele and I, as composers, felt were in need of tighter integration than was possible with the tools we had at hand at the time. Computer-aided composition has been at the forefront of the whole project since the beginning, by virtue of notation editors and data structures meant to represent symbolic musical information. On the other hand, the fact that *bach* is tightly embedded in the real-time environment of Max opened up a field of possibilities beyond what was afforded by previous CAC tools, and questioned the rifts between realtime and non-realtime operation, between symbol and sound, and between composition and performance, all while providing Max with a new, comprehensive approach to the venerable score-orchestra paradigm. [10]

3. WHAT HAPPENS NOW, AND A SEMI-PERSONAL WISHLIST

Sixteen years after, it appears clearly that *bach* is the starting point, not the destination. [24] Whereas academic discourse about algorithmic composition seems to have prospered over the last few years [25, 26], the same cannot be really said about CAC. If anything, it seems that, even today, many of the basic assumptions about CAC are still strongly indebted to the techno-cultural context of the late 1980s: by skimming through the articles of relevant conferences (such as TENOR, SMC or ICMC) or journals (such as the *Computer Music Journal*, the *Journal of New Musical Research* and the *Leonardo Music Journal*), it appears that the discourse still revolves essentially around the idea of computerised musical formalisation, although the original theoretical tenets outlined above are far less central than they were supposed to be in the 1980s; and, perhaps more importantly, and with few exceptions including, but not limited to, *bach*, [11] that research in computer-aided composition has been somewhat languishing after the early 2000s.

On the other hand, the landscape has profoundly changed, not only in terms of the technological affordances outlined above. The concepts themselves of computer music and computer composition have mutated, surely by virtue of the dizzying evolution of computer technology. Today's composers transcribe improvisations and sketches through the aid of digital audio workstations and audio analysis and segmentation tools [30, 31]; they assemble montages of instrumental sounds taken from organised databases [32, 33]; they leverage score-engraving tools not only for the actual typesetting but for sketching alternative hypotheses and versions, as well as working out the musical form in a non-linear fashion; they collaborate among them and with interpreters through the Internet by means of cloud storage, versioning systems and more or less music-oriented conferencing tools.

3.1 A Prospective Outlook

I think that there is much more yet to be explored; the concepts of computer-aided composition need to be expanded to incorporate such practices. Opportunities open up not only in the work methods, but in the outcomes of the compositional work as well. Tools associated with, and bringing affordances related to, a broader range of musical idioms may become a part of the workflow: this may include hardware controllers of all kinds, popular music-oriented software instruments and sound processing effects, and more. [12] Moreover, the potential outcomes of the compositional work are not only concert pieces: they may include albums of studio-produced recorded music, soundtracks for multimedia works and more, all things that, bar sporadic exceptions, involve computer technology. Predictably, this will also broaden the aesthetic applicability of the concept of computer-aided composition itself.

If, after more than two decades of conversations with friends and colleagues, I should dream of what computer-aided composition should mean today, I think some foundations would need to be reestablished.

In my book of dreams, of a CAC environment should be as a hub for multimodal input and output. Besides music notation, composers should be able to enter sound materials, graphical sketches, and descriptions in both natural language and computer code. Correspondingly, CAC tools should be able to generate scores with multimodal, parametric and algorithmic control of all the graphical aspects of notation; also, they should be able to accommodate standard, non-standard and graphical notation, as well as any combination of those, potentially through the definition of custom graphical semantics. They should allow generating, processing, editing and organising sound materials, by blending and intertwining textual (as in, say, Csound and SuperCollider) and graphical (as in a typical DAW) approaches. They should lend themselves to the analysis of sonic, symbolic and graphical materials, in order to both

[9] Among the various happy circumstances that happened to Daniele and myself over the last 15 years, a recent one has been the invitation to republish the first ICMC article about *bach* in a retrospective issue of Array, the journal of the International Computer Music Association.

[10] The fact that *bach* has brought the score-orchestra paradigm into Max was pointed out in an enlightening anonymous peer review of a paper I co-authored several years ago.

[11] Other relevant CAC-related projects are Georg Hajdu and Nick Didkovsky's *MaxScore* [27]; the already mentioned *MOZ'Lib* by Julien Vincenot, which is in turn partly based upon *bach*; and Carmine Emanuele Cella's *Orchidea* [28], specifically targeted at computer-aided orchestration. Also highly relevant is Juan Sebastián Vassallo's research about the affordances of intertwining machine learning and more traditional CAC practices. [29]

[12] It keeps striking me how a substantial parts of the advertisements that the YouTube platform picks for me are of software libraries of chords and melodic patterns for popular music producers—where 'producer', after all, is just how the composer is called in a different community. [34] This can be rightfully called an instance of computer-aided composition applied to vernacular musical idioms.

describe their properties in natural language, for the sake of musical analysis per se, and generalise them through the automatic generation of parametrisable computer code, so as to 'climb up' to a compositional model starting from an example, or a set thereof. Moreover, real-time operativeness should be as ample as possible, so as not to enforce a strict divide between composition and performance; and tools for cooperative editing should be accounted for.

Many aspects of such a system already exist in various pieces of software: highly configurable environments for sound-based composition such as Reaper and Ableton Live can be controlled respectively through Lua code and Max patches, in ways that are so advanced that they can be used to implement algorithmic compositional processes. As mentioned above, the bridging of realtime and non-realtime workflows has long been a preoccupation of computer music research, and has at least partly been addressed by various projects, including *bach*.

3.2 My Wishlist

On the other hand, some of the features that have been outlined above are almost non-existing. For one thing, no score-engraving tool of sufficient power exists to my knowledge. In some respects, Avid Sibelius seems to be the most advanced commercial product, allowing both local and remote scripting capabilities, through respectively the *ManuScript* programming language and the *Sibelius Connect* feature. Still, even if this may be technically possible, integrating such technologies in a multimodal workflow such as the one described above seems like a quixotic endeavour. Conversely, the fact that *LilyPond* accepts text files as its input clearly lends it to all kinds of algorithmic usage, but its lack of a WYSIWYG interface proper makes it cumbersome, to say the least, for less formalised approaches to music writing. [13] As a further consideration regarding music engraving environments, the fact that their behaviour at large rests upon assumptions that are not only graphical but music-theoretical as well [14] makes simple tasks easier, but hinders the representation of non-standard (with respect to common-practice music) musical idioms and objects. This includes, but is not limited to, the difficulty of embedding graphical notation and notation of extended instrumental techniques through music engraving tools.

Moreover, the usefulness to 'reason' graphically, and to import and process graphical sketches, is the intuition that drove the development of the Maquette in OpenMusic in the first place, and it has somehow informed other projects such as IanniX [35], in turn based upon Iannis Xenakis' UPIC system. Nonetheless, as of now, a composer willing to draw a graphic score and work computationally on it, potentially using it as an entry point for eventually producing a score in standard notation or a sound-based composition, would hardly know how to proceed, if not by creating a custom system probably requiring them to convert by hand the graphics into some compact numeric form. Such a task could probably be carried out efficiently by so-called artificial intelligence.

In fact, the integration of machine learning techniques is a complex and potentially controversial subject, which I am not going to tackle thoroughly in this text. On the one hand, machine learning and generative neural networks appear to have the potential to greatly simplify human-machine interaction, through the ability of expressing complex tasks in natural language. So-called 'vibe coding', [15] although problematic in critical applications, probably lends itself well to a setting such as musical composition, where there is generally room for a trial-and-error approach, the assessment of the quality of the result is strongly subjective and hard to formalise, and anyway potential errors do not have catastrophic consequences. A properly trained LLM could thus be seen as a highly skilled, although slightly unpredictable, assistant, just like a somewhat rebellious apprentice in a Renaissance workshop.

On the other hand, the discussion about the integration of artificial intelligence in the compositional activity mostly revolves around the supposed creative agency (or lack thereof) of such software tools. Positions among artists and scholars on the implications of this divisive subject are wildly varied: see, for example, the recent outcry of prominent rock musicians, including Paul McCartney, against the widespread inclusion of AI-generated content in musical streaming services, but also the positive outlook on the creative potential of AI in music as propounded by the *Posthuman Music* research project led by Paulo de Assis. Positions in both camps are motivated by considerations about ethics, psychology, art and music philosophy, economy, ecology, politics, as well as unspoken personal beliefs and dispositions. All this being said, it is undeniable that a comprehensive system for computer-aided composition in the current era should embed or, at least, interface to generative AI tools.

4. ACKNOWLEDGMENTS

This article is integral to my ongoing doctoral research at the Orpheus Institute of Ghent and the University and Royal Conservatoire of Antwerp. I wish to thank my advisors Jonathan Impett and Jouke Verlinden for their precious guidance and the all the insightful thoughts they have been sharing with me.

I also wish to thank Daniele Ghisi for all the work and the reflections on CAC that we have shared over the last two decades, and which have laid the ground for the considerations of this article; and Carmine Emanuele Cella for some engaging conversations that have strongly influenced many of the prospective ideas presented in this article.

[13] A visual editor for Lilypond, *Frescobaldi*, exists, and it vastly improves on LilyPond's ease of use. Still, the fact that the manipulation of musical notation has necessarily to be mediated by the writing of textual code seems to be perceived as an obstacle to the creative workflow by many composers I had the opportunity to speak to.

[14] Example of this include the facts that it is cumbersome to fill a measure in Sibelius or Dorico with more musical material than its time signature allows; that using arbitrary combinations of alterations on a single note is problematic, if not impossible; that barlines in a staff system are always vertically aligned, unless tricky graphical tweaks are put in place.

[15] *Vibe coding* is the vernacular name of a software development method in which developers describe desired functionality in natural language to a large language model generating the actual computer code. In principle, this allows humans to focus on high-level idea and iterative experimentation rather than the low-level details of computer language syntax.

5. REFERENCES

[1] F. Lerdahl and Y. Potard, "La composition assistée par ordinateur," Ircam, Tech. Rep., 1986.

[2] F. Lerdahl and R. Jackendoff, *A Generative Theory of Tonal Music*. The MIT Press, 1983.

[3] G. Assayag, *Les Cahiers de l'Ircam: La composition assistée par ordinateur*. Paris: Ircam — Centre Georges-Pompidou, 1993, ch. Cao: vers la partition potentielle.

[4] T. W. Adorno, *The Oxford Handbook of Timbre (1966)*. Oxford Handbooks, 2021, ch. The Function of Timbre in Music.

[5] C. Ames, "Automated Composition in Retrospect: 1956-1986," *Leonardo*, vol. 20, no. 2, pp. 169–185, 1987. [Online]. Available: http://www.jstor.org/stable/1578334

[6] A. Bregman, *Auditory Scene Analysis: The Perceptual Organization of Sound*. The MIT Press, 1990.

[7] G. Born, *Rationalizing Culture: IRCAM, Boulez, and the Institutionalization of the Musical Avant-Garde*. University of California Press, 1995.

[8] C. Agon, G. Assayag, and J. B. (editors), *The OM Composer's Book*. Editions Delatour France / Ircam-Centre Pompidou, 2006.

[9] B. Ferneyhough, *Collected Writings*. Routledge, 1995.

[10] *Computer Music Journal*, vol. 39, no. 2, 2015.

[11] F. Lévy, "Complexité grammatologique et complexité aperceptive en musique," PhD thesis, École des Hautes Études en Sciences Sociales, Paris, February 2004.

[12] F. Rose, "Introduction to the Pitch Organization of French Spectral Music," *Perspectives of New Music*, vol. 34, no. 2, pp. 6–39, 1996.

[13] M. Laurson, "PWConstraints," in *Atti del X Colloquio di Informatica Musicale*, Milano, 1993, pp. 332–335.

[14] A. Bonnet and A. Rueda, *Recherches et applications en informatique musicale*. Paris: Hermès, 1998, ch. Situation: un langage visuel basé sur les contraintes pour la composition musicale.

[15] A. Agostini and D. Ghisi, "A Max Library for Musical Notation and Computer-Aided Composition," *Computer Music Journal*, vol. 39, no. 2, pp. 11–27, 2015.

[16] J. Vincenot, "LISP in Max: Exploratory Computer-Aided Composition in Real-Time," in *Proceedings of the 2017 International Computer Music Conference*, Shanghai, 2017, pp. 87–92.

[17] L. Goher, *The Imaginary Museum of Musical Works: An Essay in the Philosophy of Music*. Oxford University Press, 1994.

[18] C. Small, *Musicking: The Meanings of Performing and Listening*. Wesleyan University Press, 1998.

[19] P. Simons, "Computer Composition and Works of Music: Variation on a Theme of Ingarden," *Journal of the British Society for Phenomenology*, vol. 19, no. 2, pp. 141–154, 1988.

[20] C. Dodge and T. Jerse, *Computer Music: Synthesis, Composition, and Performance*, 2nd ed. Macmillan Library Reference, 1997.

[21] N. Schnell, R. Borghesi, D. Schwarz, F. Bevilacqua, and R. Müller, "FTM — Complex Data Structures for Max," in *Proceedings of the 2005 International Computer Music Conference*, Barcelona, 2005.

[22] A. Cont, "ANTESCOFO: Anticipatory Synchronization and Control of Interactive Parameters in Computer Music," in *Proceedings of the 2008 International Computer Music Conference*, Belfast, 2008.

[23] N. Schnell, A. Röbel, D. Schwarz, G. Peeters, and R. Borghesi, "MuBu and Friends — Assembling Tools for Content Based Real-Time Interactive Audio Processing in Max/MSP," in *Proceedings of the 2009 International Computer Music Conference*, Montreal, 2009.

[24] D. Ghisi and A. Agostini, "A Compendium of bach's Shortcomings as it Turns Ten—Along with a Few Ideas on How to Overcome Them," in *Proceedings of the 17th Sound and Music Computing Conference*, Torino, 2020.

[25] G. Nierhaus, *Algorithmic Composition—Paradigms of Automated Music Generation*. Springer, 2009.

[26] R. Dean and A. M. (editors), *The Oxford Handbook of Algorithmic Music*. Oxford University Press, 2018.

[27] N. Didkovsky and G. Hajdu, "MaxScore—Music Notation in Max/MSP," in *Proceedings of the 2008 International Computer Music Conference*, Belfast, 2008.

[28] C. Cella, "Orchidea: A Comprehensive Framework for Target-Based Computer-Assisted Dynamic Orchestration," *Journal of New Music Research*, vol. 51, no. 1, pp. 40–68, 2022.

[29] J. S. Vassallo, "Outward Threads — Intuitive Computers / Rational Composers," PhD thesis, University of Bergen, Bergen, September 2025.

[30] D. Schwarz and B. Hackbarth, "Navigating Variation: Composing for Audio Mosaicing," in *Proceedings of the 2012 International Computer Music Conference*, Ljubljana, 2012.

[31] D. Rocchesso, G. Lemaitre, P. Susini, and P. B. S. Ternström, "Sketching Sound with Voice and Gesture," *Interactions*, vol. 22, no. 1, pp. 38–41, 2015.

[32] D. Ghisi, "Music Across Music: Towards a Corpus-Based, Interactive Computer-Aided Composition," PhD thesis, Université Pierre et Marie Curie, Université Paris-Sorbonne, Sorbonne Universités, Paris, December 2017.

[33] A. Einbond, C. Trapani, A. Agostini, D. Ghisi, and D. Schwarz, “Fine-tuned Control of Concatenative Synthesis with CataRT Using the bach Library for Max,” in *Proceedings of the 2012 International Computer Music Conference*, Athens, 2014.

[34] V. Moorefield, *The Producer as Composer: Shaping the Sounds of Popular Music*. The MIT Press, 2005.

[35] T. Coduys and G. Ferry, “IanniX Aesthetical/Symbolic Visualisations for Hypermedia Composition,” in *Actes des journées d’informatique musicale*, Paris, 2004.

MBHD: A Modular Audio Playback and Manipulation System for Loop-Based Performance

Neal Anderson
Style tree
neal@styletree.me

Sanjay Majumder
Style tree
sanjay@styletree.me

ABSTRACT

This paper outlines the development of MBHD (Modular Beat Handling Device), a real-time audio performance system using Cycling '74 Max that connects the reliability of DJing with the expressiveness of live electronic composition. While DAWs provide reliable synchronization, achieving a harmonically coherent alignment of loops from different library collections typically requires significant manual editing of metadata. MBHD addresses this challenge with a new, lightweight naming convention based on filename-encoded musical attributes (tempo, root note, and instrument role), providing automatic harmonic coherence. This system is organized around four independent layers of musical content that can be reused and rerouted (drums, bass, harmony, and melody); and utilizes real-time digital signal processing (DSP) to dynamically adjust the pitch and timing of each layer so as to align to a global key and tempo. In addition to the description of the system's architecture, we also describe the integration of the system with external environments (via Ableton Link) and the design of the user interface to allow for minimal latency during the performance process. Lastly, we report results of evaluations (technical benchmark, user study) of the MBHD, which demonstrate how transparent systems using filename-driven architectures can be used to facilitate the use of loops for improvisation.

1. INTRODUCTION

MBHD (Modular Beat Handling Device) is a modular loop playback device intended to bridge the gap between the structural reliability of DJing and the freedom of live electronic composition. MBHD was built using Cycling '74 Max. The system allows a performer to have a structured but open-ended approach to looping, manipulating, and arranging multiple tracks of musical groove, in an environment where all tracks are locked to a global tempo and key.

One problem in live loop-based performances has been creating harmonic coherence while not breaking the creative flow. The DAWs and DJ programs are very good at locking beats together; however, most programs do not provide much assistance for aligning key signatures across a wide variety of samples, unless the user wants to go through the process of manually editing each sample's metadata, or use a third-party utility program, which can be time-consuming and limiting.

MBHD removes this obstacle by introducing one new concept, a structured filename format. This format allows MBHD to read the necessary musical information from the filename of each loop. For example, MBHD parses tempo (BPM), root note, and instrument role from each filename, and then uses a real-time transposition algorithm to transpose each loop to fit the global selected key signature, thus removing the need to manually transpose each track. It should be noted that the current version of MBHD assumes 12-tone equal temperament to keep compatibility with standard Western electronic music styles.

Musical material is broken down into four distinct layers, drums, bass, harmony, and melody, and each layer can be controlled individually. The performer can switch, route, pitch shift, and mix these layers, allowing for hybrid approaches that combine the set-and-forget stability of DJ sets with the fine-grained control of live improvisation. Using the native DSP objects provided by Max for low latency processing, MBHD also supports synchronization over networks and ensembles through Ableton Link.

This paper outlines the architecture, interface, and implementation of MBHD. We situate the system within a lineage of live performance tools and demonstrate how modular, naming-driven systems can evolve traditional paradigms of musical interaction. Beyond technical implementation, this system acknowledges the 'aesthetic and cultural dimensions' of looping practice, aiming to support the performer's identity by reducing technical friction during the act of live remixing. To further reduce barriers to entry, a browser-based companion application has also been developed to demonstrate selected MBHD design principles in an accessible form.

2. RELATED WORK

The design of MBHD is situated between professional DAWs and experimental NIME (New Interfaces for Musical Expression) frameworks. Max [1] remains a primary environment for such custom performance systems. A notable precursor is mlr [2], which established a model for live sample-cutting on the Monome Grid. While mlr focuses on rhythmic slicing, MBHD prioritizes harmonic coherence to ensure the structural "groove" remains intact during key changes. Unlike systems like Debris [3], which utilize granular manipulation for non-linear exploration, MBHD provides a structured workflow necessary for beat-driven electronic music requiring consistent pulse

Copyright: ©2026 Neal Anderson and Sanjay Majumder. This is an open-access article distributed under the terms of the Creative Commons Attribution License 3.0 Unported, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

and tonality.

Commercial hardware, such as the Roland SP-404 [4] and Boss RC-505 [5], offers tactile reliability but often treats audio as static. In these devices, pitch shifting frequently requires menu-diving and introduces significant artifacts. Similarly, DAWs like Ableton Live [6] require manual metadata editing (e.g., warp markers and root note definition) for harmonic matching. MBHD automates this via filename parsing, aligning with "human-in-the-loop" (HITL) research seen in AdaptiveSound [7], where real-time user inputs—in our case, filenames—configure system behavior.

MBHD also incorporates findings from Bin and Keller [8] regarding the necessity of rapid feedback and spatiality. By avoiding nested menus in favor of one-to-one parameter mapping, MBHD achieves the "tactile immediacy" required for live sets. While multimodal loopers [9] and co-creative gestural systems [10] offer innovative control, they often suffer from mapping confusion. MBHD favors a transparent GUI to reduce cognitive load, emphasizing "interaction over syntax" compared to the high overhead of live coding environments [11, 12].

Within the Max ecosystem, the Karma looper [13] excels at capturing live input for evolving textures. MBHD, however, is optimized as a "performance engine" for pre-recorded stems, ensuring harmonic synchronization across layers. This focus on deterministic structural manipulation distinguishes MBHD from agent-based systems like the Reflexive Looper [14] or The Living Looper [15], which focus on loop interplay rather than user-led arrangement. Lastly, as Duarte and Sardo [16] note, reducing technical friction is essential to preserving performer identity—a core goal of the MBHD architecture.

3. SYSTEM ARCHITECTURE

The MBHD architecture is designed as a modular, parallel processing framework that decouples audio playback from specific DSP manipulation logic. Unlike monolithic playback systems, MBHD treats each musical layer (i.e., Drums, Bass, Harmony, and Melody) as a discrete, independent signal chain. This separation allows for granular control over pitch, spatialization, and volume without cross-channel interference, as illustrated in Figure 1.

3.1 Modular Signal Flow

The system has been segmented into four separate modules, each working independently to process and produce the final audio output. The individual modules are:

- Audio Ingestion: Loading stereo audio data into memory.
- Managing loop points, start/stop triggers, and retriggering mechanics.
- Spectral Manipulation: performing real-time frequency domain processing for pitch correction (transposition) independent of time stretching.
- Spatial Routing: Managing stereo placement and signal muting via a dynamic routing matrix.

Each of these modules provides a single 'voice' in the overall system that works independently of the other voices. Each module can perform the same function multiple times, but if it does, it will do so independently of the other instances of the same module. The four 'voices' are then combined and sent through a master mixer to provide a common gain level and limiters to prevent distortion of the audio.

3.2 Semantic Data Model

A further unique characteristic of the MBHD is the semantic nature of the file handling. Instead of relying on file formats and ID3 tags as provided by DAWs, the MBHD uses the file name itself as the database. When a Groove folder is loaded into the system, the system reads the file names to determine three key pieces of information:

- Tempo (BPM): Used to determine the amount of time to stretch or shrink the audio file.
- Root Key: Used to determine the number of semitones between the root key and the actual key of the file.
- Instrument Role: Used to determine which processing module the file should be directed to (for example, directing a file called "Bass" to the Bass processing module).

In this way, the file becomes a "smart" object that contains all the necessary information to configure the system appropriately when a new Groove folder is loaded.

3.3 Synchronization and Transport

Time is controlled using a global transport clock, which is the central reference point for all processing modules. The global transport clock is completely decoupled from the audio playback rate, so that the user can adjust the tempo and/or speed of the audio without altering the pitch of the currently playing loops. All communication between the MBHD and external equipment and software is done through the Ableton Link protocol, which enables the system to synchronize tempo and phase across devices with high temporal accuracy when connected to other Ableton Link-enabled devices.

3.4 Data Validation and Fallback Logic

Since the system relies on very specific formatting conventions for file names, a default behavior has been built-in to handle situations where a file does not follow the specified format (for example, missing BPM, root key, instrument role). If a file is loaded into the system that does not have the expected filename convention, the following default behaviors occur:

- Pitch Fallback: The system assumes that the file is in the key of C and performs no transposition.
- Tempo Fallback: The system attempts to automatically detect the tempo of the file using available metadata when present, or estimate it heuristically; otherwise it defaults to the current global tempo.

- Role Fallback: If the instrument role is unclear, the file is routed to a generic Auxiliary Bus. These default behaviors are implemented to prevent errors due to malformed filenames, but may lead to incorrect harmonics.

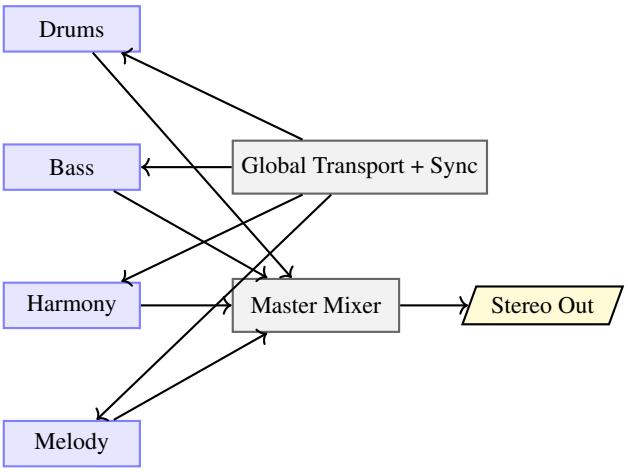


Figure 1. Compact system architecture of MBHD

4. INTERFACE DESIGN

The MBHD's interface is designed to implement the "Performative Visibility" philosophy; it makes the most latency critical parameters immediately available to the user while hiding the logic of how the system is configured (i.e., file paths) once it is running. The Graphical User Interface is built using the Max Presentation Mode and integrates the four independent signal chains into a single, color coded dashboard.

4.1 Hierarchy and Cognitive Load

To minimize the cognitive load associated with simultaneously controlling four different audio streams, the interface implements an extreme level of spatial consistency. As a result, there are four separate vertically oriented channel strips, one per Track (Drums, Bass, Harmony, Melody). Each of these channel strips will have the same set of controls arranged similarly across the horizontal axis. The overall layout is similar to a conventional DJ Mixer and utilizes existing muscle memory to help minimize the learning curve required to use the system.

At the top of each channel strip is displayed critical state information, specifically, the Parsed Root Key and the Current Transposition Interval, which the performer can see directly above each channel strip. This allows the performer to quickly determine if the system has successfully parsed the filename and if it has properly aligned the harmonic content of the four tracks.

4.2 The Dub-Style Mute Matrix

A major feature of the live performance is the Mute Matrix (Figure 2), a toggle matrix in the Master Section. The Mute Matrix is used for performing "dub-style" mixing. Instead of the typical Solo/Mute function of most digital audio workstations, the Mute Matrix enables the user to rapidly route any of the four Tracks to either the Left Channel, the Right Channel, or both channels simultaneously. The Mute Matrix also explicitly supports the concept of "dub mixing," which involves creating tension by dropping the bass out of the mix or throwing the snare drum into a particular delay line (if delayed externally).

4.3 Decoupled Pitch and Tempo Control

The interface has decoupled pitch controls from transport controls to emphasize the modular nature of the system.

4.3.1 Pitch UI

The Pitch controls are comprised of four separate numeric steppers, one for each Track, and are intended to support compositional moves (e.g., moving a Harmony layer up a perfect fifth to create counterpoint) independent of the Global Key.

4.3.2 Transport UI

The transport controls are located in the Header and visually separated from the Pitch controls. They are intended to be the source of truth for all Time-Stretching Operations.

4.4 Visual Feedback and Metering

Given the possibility of high gain levels when summing four pre-mastered loops, the interface places a high priority on visual feedback. There are High Resolution Peak Meters (`live.meter~`) located post-fader on every channel so that the user can accurately gauge the gain level. Also, the Sync Indicator provides a binary status (Locked/Drifting) of the Ableton Link connection, so the user can rely on the Timing Engine to provide accurate timing without having to constantly check the output through the speakers.

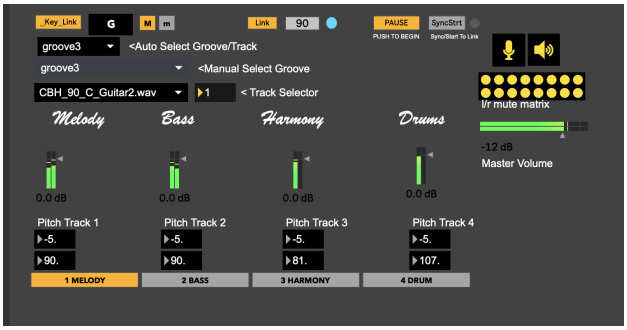


Figure 2. MBHD GUI for live performance. Provides full control over groove, pitch, routing, and sync.

5. IMPLEMENTATION DETAILS

The MBHD system is implemented in Cycling '74 Max 8, utilizing native MSP objects for signal processing and standard Max objects for control logic. The patch is optimized for standard consumer hardware (e.g., Apple Silicon M-series or Intel i7), prioritizing low-latency performance without requiring external DSP extensions.

5.1 Dynamic File Parsing and Buffer Management

File management is handled using the `folder` and `umenu` objects to index directories dynamically. Audio files are organized into named groove folders (e.g., `groove1`, `groove2`, etc.), each containing multi-track .wav/.aiff files (Figure 3). When a user selects a specific "Groove" folder, the system iterates through the file list, utilizing string parsing logic to deconstruct filenames based on a structured convention (e.g., `<BPM>_<Root>_<Instrument>.wav`).

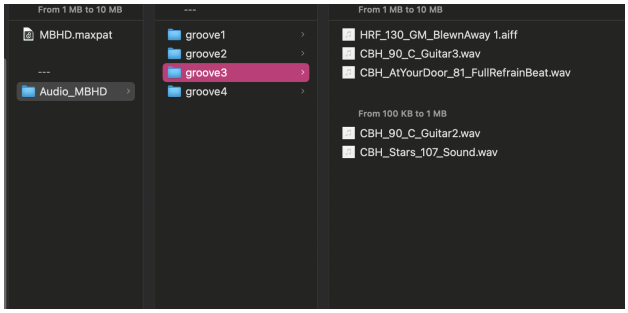


Figure 3. Folder hierarchy and file naming conventions used for groove-based audio content.

These parsed tokens are stored in a `dict` (dictionary) object, which maps each audio file to its corresponding metadata. This dictionary serves as the central lookup table for the playback engine, ensuring that when a file is loaded into a `buffer~` object, its associated pitch and tempo attributes are immediately available to the DSP signal chain.

5.2 DSP Playback Engine

Each of the four tracks (Drums, Bass, Harmony, Melody) is powered by a dedicated `groove~` object. We utilize `groove~` for its variable-speed playback capabilities, which are essential for the system's dual-layer time and pitch control strategy:

- **Time Stretching:** The playback rate of `groove~` is manipulated to match the global transport tempo using a ratio calculated as:

$$Rate = \frac{BPM_{global}}{BPM_{file}} \tag{1}$$

- **Sync:** Loop synchronization is maintained using the `transport` object, with `phasor~` or trigger-based resets ensuring phase alignment at the start of every bar.

5.3 Spectral Pitch Shifting

To achieve independent pitch control (transposing content without altering playback speed), the system employs frequency-domain processing. We utilize a dual-algorithm approach for pitch shifting. The system defaults to `gizmo~` object (based on phase-vocoder FFT logic) for low-latency performance during live monitoring. However, users can toggle to `retune~` (monophonic pitch correction) for higher-fidelity results on predominantly monophonic melodic content, at the cost of slightly increased CPU load and latency (approx. 100ms). This hybrid implementation balances instantaneous response with sonic quality.

The pitch shift amount is calculated using the standard equal-temperament ratio formula:

$$P_{ratio} = 2^{\frac{K_{target} - K_{source}}{12}} \tag{2}$$

Where K_{target} is the global key and K_{source} is the parsed root key of the file. This ratio is smoothed using `line~` to prevent audio artifacts during rapid key changes.

5.4 Harmonic Logic and Quantization

The "Harmonic Coherence" engine is implemented using integer math to derive semitone offsets. The system quantizes all pitch requests to the nearest integer semitone.

The current parsing logic is hard-coded for 12-tone equal temperament (12-TET). The pitch calculator interprets file tags (e.g., "C", "C#", "D") as integers 0 - 11. While robust for standard electronic music, this implementation requires modification to the `coll` (collection) data structures to support microtonal or non-Western tuning systems. It should be noted that this approach operates at the pitch-class level and does not explicitly model harmonic function or modal context; therefore, while it ensures pitch alignment, more complex tonal relationships are not inferred.

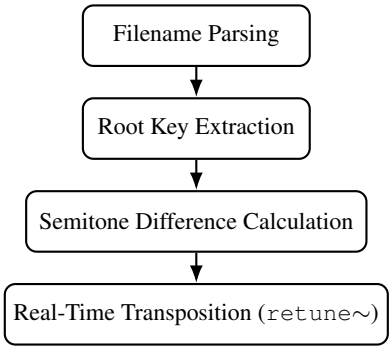


Figure 4. Harmonic coherence processing chain: the system parses filename metadata, computes pitch-class differences, and applies real-time transposition.

The harmonic coherence processing chain (Figure 4) operates linearly: the system first parses the filename metadata, extracts the root key, computes the pitch-class difference modulo 12, and finally applies the real-time transposition via the `retune~` object.

Filename	Detected Root	Global Root	Applied Shift (semitones)
90_C_Guitar.wav	C	E	+4
80_Gm_Bass.wav	G	F	-2
87_A_Synth.wav	A	D	+5
93_Dm_Pad.wav	D	D	0

Table 1. Example of filename-based parsing (tempo, key, and role) and automatic transposition.

Table 1 demonstrates MBHD's automatic transposition mechanism, maintaining harmonic consistency across loops of varying tonal centers.

5.5 Routing and Output

Signal routing is managed via a matrix of `gate~` and `matrixctrl` objects. This allows for low-artifact muting when combined with appropriate signal smoothing and stereo panning. The final output stage allows for soft clipping protection using `limi~` or simple multiplication logic (`*~`) to ensure the summed signal from the four tracks does not exceed 0 dBFS.

5.6 Web Implementation

To increase accessibility and provide a lightweight demonstration of the MBHD design principles, a browser-based companion application was developed (see Figure 5). Due to current constraints of web deployment and performance, this implementation does not replicate the full audio-loop playback architecture of the Max system. Instead, it focuses on a subset of concepts, namely structured musical recombination, instrument-role abstraction, and user-driven interaction, implemented using MIDI-based generation. While the web version differs technically from the Max implementation, it serves as a conceptual extension that illustrates the system's emphasis on modularity, accessibility, and rapid musical interaction.

6. USE CASES

The modular and performance-oriented design of MBHD supports a variety of use cases within live music and multimedia contexts. Below are several scenarios where the system has proven valuable:

6.1 Live Electronic Performance

MBHD functions as a versatile performance instrument for solo artists and DJs. It allows real-time swapping and remixing of musical layers during a set, enabling artists to transition fluidly between grooves or recontextualize tracks through pitch and spatial manipulation. The mute matrix and stereo routing capabilities allow performers to craft dynamic arrangements and spatial experiences using only a standard stereo output.

6.2 Collaborative Ensemble Settings

Through its Ableton Link integration, MBHD can synchronize with other devices or performers. In ensemble scenarios, each performer may control a different musical role (e.g., drums or harmony), allowing for distributed interaction. The system's tight synchronization and independent control layers make it suitable for both rehearsed and improvisational settings.

6.3 Educational and Pedagogical Tools

The system's visual structure and modular audio decomposition make it an effective teaching tool in music technology and production education. Students can explore concepts of groove construction, loop manipulation, pitch shifting, and digital signal routing using a tangible, auditory feedback loop.

6.4 Installation and Interactive Art

MBHD can be embedded within interactive sound installations, where user-triggered events (e.g., sensor data or motion input) manipulate groove parameters. The patch structure supports external control via MIDI [17], OSC [18], or hardware interfaces, making it adaptable to a wide range of interactive settings.

7. RESULTS AND EVALUATION

The MBHD system was evaluated through a series of structured user tests, technical benchmarks, and live performance trials. This section presents the qualitative and quantitative results obtained from these activities, focusing on system reliability, usability, creative affordances, and performance synchronization. All evaluations reported in this section were conducted using the Max-based implementation of MBHD; the browser-based version was not included in these tests due to its reduced feature scope.

7.1 Technical Performance

To assess the real-time audio processing capabilities of MBHD, the system was stress-tested with multiple groove sets consisting of four stereo audio tracks each, ranging from 1 to 16 bars in length. The testing was performed on a MacBook Pro M1 with 16 GB of RAM, using Max 8, a sample rate of 44,100 Hz, and a scheduler interval of 1 ms.

7.1.1 Latency

The system exhibits two forms of latency: I/O latency and algorithmic processing latency. The round-trip audio latency (including buffering and scheduling) was measured at approximately 5 ms under test conditions.

In addition, the use of frequency-domain pitch shifting (`gizmo~`) introduces an inherent processing delay due to FFT windowing (approximately 43 ms for a 1024-sample window). This delay is internal to the signal path and does not affect transport synchronization, as compensation is applied to maintain rhythmic alignment across tracks.

7.1.2 Audio Stability

A 60 minute automated stress test was run using random scene changes every 4-bar intervals. During the entire test period, there were no audio dropouts or buffer underrun events.

7.1.3 CPU Usage

Under a full-load condition (i.e., four separate stereo tracks each being played back independently with real-time pitch-shifting applied, and two GUI meters displaying active data), the average CPU usage of the system was kept at levels below 20%. There were occasional spikes to 30%; however, these only occurred when multiple large Groove files were loaded simultaneously.

7.2 User Study

A total of eight participants participated in the user study, including 3 professional electronic musicians, 3 advanced music technology students, and 2 audio engineers. Each participant performed a 10-minute improvised performance using a curated library of loops.



Figure 5. Browser-based companion application illustrating selected MBHD design principles, including structured musical interaction and visual performance controls.

7.2.1 Usability

The usability of the system was rated highly, with a mean SUS score of 86.3, corresponding to ”excellent” usability. Specifically, the visual ”Mute Matrix” (Figure 2) was identified as a key feature that allowed users to ”dub-style” mix their loops without having to navigate menus.

7.2.2 Creative Potential

Six out of eight participants reported that the separation of ”pitch” and ”speed” led to new performance techniques. As an example, one participant stated they could ”re-harmonize a percussive loop into a drone texture by pitching it down -12 semitones while keeping the rhythm locked”, something that would be difficult to do with a typical hardware looper. Another participant also complimented the file naming convention, noting that it removed the ”anxiety of mixing out-of-key samples”.

7.2.3 Workflow

All users were able to successfully navigate the folder structure and load custom grooves within ten minutes of training, thus verifying the effectiveness of the ”filename-as-database” approach.

7.3 Synchronization and Interoperability

Synchronization accuracy was assessed via Ableton Link integration with other software (Ableton Live [6]) and hardware (an iPad running Patterning). After thirty minutes, the system’s tempo drift was less than ± 1 ms. Although the synchronization accuracy is largely a result of the inherent stability of the Link protocol, our testing showed that the internal Max scheduler priority settings prevented any onset jitter - i.e., the timing of the audio playback of the `groove~` objects was always precise and synchronized with the downbeat, regardless of how much GUI activity was occurring concurrently.

8. DISCUSSION AND FUTURE WORK

The Modular Beat Handling Device (MBHD) demonstrates that a modular audio environment provides musicians with fine-grained control over their music in live performance settings; however, this has provided insight into opportunities for improvement, specifically with regard to content preparation and intelligent interaction.

8.1 Removing the ”Metadata Friction” via AI

Ingesting a new song requires manual renaming according to a rigid filename format (BPM_Root_Instrument). Therefore, one area of research focus moving forward is the direct integration of Music Information Retrieval (MIR) tools into the ingest pipeline. Through the use of Python scripts (*node.script* or *py* objects within Max), the system may be able to automatically analyze raw audio files to identify:

- Key and Mode: using chroma features to auto-label the root note.
- Tempo: performing transient detection to estimate BPM.
- Classification: using models for instrument classification to automatically route audio to either ”Drums” or ”Melody” tracks. This will enable performers to simply drop any loop library into the folder structure and have MBHD handle all of the semantic labeling and routing independently.

8.2 Generative "Co-Performer" Agents

We intend to develop MBHD further into a semi-autonomous performance partner beyond basic playback. The next generation of MBHD may utilize probabilistic logic (Markov chains) to suggest or queue up the next compatible groove based upon the current harmonic content and energy level of the performance. This will move MBHD from being a passive "playback engine," to an active co-performer in the creation of the live composition, providing continuity in the "flow" while the musician is free to focus on timbre and/or spatialization.

8.3 Advanced DSP and Tactile Control

The current GUI works well; however, the limitations of mouse-based interaction alone will require implementation of native OSC and MIDI 2.0 mapping to enable high resolution parameter control through hardware surfaces. We also intend to add additional functionality to the DSP module to include granular re-synthesis and spectral freezing enabling users to freeze loops in real-time and convert them into static textural pads.

9. CONCLUSION

The MBHD has been successful in demonstrating the viability of an entirely new approach to the way in which a "filename-driven" performance environment is achieved. The fact that the MBHD's audio content can be completely decoupled from its associated digital audio workstation (DAW) timeline while still enforcing harmonic rules using modular DSP makes possible a brand-new method of workflow for live electronic musicians. In addition, MBHD will continue to evolve through the integration of artificial intelligence (machine learning), enabling automated handling of metadata, thus transforming the MBHD from a structured loop-player to a fully fluid, intelligent, and responsive improvisational instrument.

10. AVAILABILITY

The Max patches for this project can be accessed at: `https://phewsh.com/mbhd/max/`. Additionally, a browser-based companion application is available at: `https://phewsh.com/mbhd/`.

11. DECLARATION OF GENERATIVE AI AND AI-ASSISTED TECHNOLOGIES

During the preparation of this work, the authors used Google Gemini to improve grammar, phrasing, and overall readability. After using this tool/service, the authors carefully reviewed and edited the text as needed and take full responsibility for the final content of the publication.

12. ETHICAL STATEMENT

This research adheres to ethical research and publication standards. All participants volunteered and provided their informed consent.

13. REFERENCES

[1] Cycling '74, *Max*, Accessed on May,2025, https://cycling74.com/products/max.

[2] B. Crabtree, "mlr: a max/msp application for the monome," 2011, accessed: 2025. [Online]. Available: https://monome.org/docs/grid/app/

[3] F. A. Robinson, "Debris: A playful interface for direct manipulation of audio waveforms," in *Proceedings of the New Interfaces for Musical Expression*, Shanghai, China, 2021, pp. 213–218.

[4] Roland Corporation, *Roland SP-404MKII Creative Sampler and Effector Owner's Manual*, Roland Corporation, 2021, accessed: 2025. [Online]. Available: https://www.roland.com/global/products/sp-404mk2/

[5] Boss Corporation, *RC-505 Loop Station Owner's Manual*, Roland Corporation, 2013, accessed: 2025. [Online]. Available: https://www.boss.info/global/products/rc-505/

[6] Ableton, *Ableton Link: Connect music making apps with Ableton Live*, Accessed on May, 2025, https://www.ableton.com/en/link/.

[7] H. Do, Q. Dang, J. Z. Huang, and D. Jain, "AdaptiveSound: An Interactive Feedback-Loop System to Improve Sound Recognition for Deaf and Hard of Hearing Users," in *Proceedings of the 25th International ACM SIGACCESS Conference on Computers and Accessibility*, New York, USA, 2023, pp. 1–12.

[8] T. Bin and D. Keller, "Real-time spatial interfaces in live electronic music: Performer strategies and design recommendations," *Journal of New Music Research*, vol. 47, no. 2, pp. 112–125, 2018.

[9] P. Kiliboz and E. Cumhur, "Multimodal Looper: A Live-Looping System for Gestural and Audio-visual Improvisation," in *Proceedings of the 9th International Conference on Movement and Computing*, Utrecht, Netherlands, 2024, pp. 1–8.

[10] I. Borovik and V. Viro, "Real-Time Co-Creation of Expressive Music Performances Using Speech and Gestures," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, Mexico City, Mexico, 2023, pp. 620–625.

[11] A. McLean and G. Wiggins, "Tidal - Pattern Language for Live Coding of Music," in *Proceedings of the 7th Sound and Music Computing Conference (SMC)*, Barcelona, Spain, 2010.

[12] C. Petrie, "Programming music with Sonic Pi promotes positive attitudes for beginners," *Computers Education*, vol. 179, p. 104409, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360131521002864

[13] R. Constanzo, "Karma: A Looping External for Max," 2015, accessed: 2025. [Online]. Available: https://www.rodrigoconstanzo.com/karma/

[14] M. Marchini, F. Pachet, and B. Carré, "Rethinking Reflexive Looper for structured pop music," in *Proceedings of the International Conference on New Interfaces for Musical Expression*. Copenhagen, Denmark: Aalborg University Copenhagen, 2017, pp. 139–144. [Online]. Available: http://www.nime.org/proceedings/2017/nime2017_paper0027.pdf

[15] V. Shepardson and T. Magnusson, "The Living Looper: Rethinking the Musical Loop as a Machine Action-Perception Loop," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, M. Ortiz and A. Marquez-Borbon, Eds., Mexico City, Mexico, 2023, pp. 224–231. [Online]. Available: http://nime.org/proceedings/2023/nime2023_32.pdf

[16] A. Duarte and S. Sardo, Eds., *Live Looping in Musical Performance: Lusophone Experiences in Dialogue*. London, UK: Routledge / Focal Press, 2024.

[17] MIDI Association, *Expanding, promoting, and protecting MIDI*, Accessed on May, 2025, https://midi.org/.

[18] M. Wright and A. Freed, *Open Sound Control*, Accessed on May, 2025, https://opensoundcontrol.stanford.edu/.

AI Framework for Dynamic Robotic Instrument Calibration

Colton Arnold
California Institute of the Arts
coltonarnold@students.calarts.edu

Zhaohan Cheng
California Institute of the Arts
zcheng@calarts.edu

Ajay Kapur
California Institute of the Arts
NYU Shanghai
akapur@calarts.edu

ABSTRACT

This paper presents a data-driven calibration framework for robotic musical instruments based on a hybrid ensemble model that combines K-nearest neighbors (KNN) and a multi-layer perceptron (MLP). KNN anchors predictions to recorded acoustic measurements, while the MLP enables nonlinear generalization and smooth interpolation across the instrument's playable range. A distance-dependent blending strategy integrates the two models, improving consistency across sparse and dense data.

The proposed approach produces stable and repeatable calibration estimates for both pitched and non-pitched instruments, outperforming standalone models across a range of sampling conditions. This work establishes a scalable foundation for automated calibration in robotic musical systems.

1. INTRODUCTION

The California Institute of the Arts (CalArts) maintains a collection of robotic instruments that require regular calibration to preserve consistent timbre, dynamic response, and overall performance quality. Over time, mechanical drift and component wear introduce nonlinear changes that alter each instrument's acoustic behavior, a well-documented issue in mechatronic systems. As these changes accumulate, the instruments no longer respond as originally intended by composers or performers.

To address this issue, we introduce a machine learning calibration framework that combines a K-nearest neighbors (KNN) with a multi-layer perceptron (MLP) to predict calibration adjustments across the instrument's playable range. By combining these complementary strengths, the system produces consistent predictions across different data densities. The remainder of this paper is organized as follows: Section 2 reviews related work on mechanical drift and data-driven calibration; Section 3 describes the proposed system and modeling framework; Section 4 presents the evaluation; and Section 5 concludes with discussion and future directions.

Copyright: ©2026 Colton Arnold et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License 3.0 Unported, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

2. BACKGROUND AND RELATED WORK

2.1 CalArts Machine Lab

The CalArts Machine Lab [1] hosts a collection of custom-designed robotic musical instruments [2] developed over the past fifteen years, including BreakBot, GanaPati, Lydia, MahaDeviBot [3, 4], MalletOTon [5], RattleTron, and Tammy. Most recently, an 8x8 grid of Modulets [5] has been added to the lab. As shown in Figure 1, these instruments vary significantly in construction, actuation, and acoustic behavior, requiring individualized calibration procedures to maintain consistent performance.



Figure 1. From bottom left clockwise: BreakBot, MalletOTon, Lydia, and GanaPati.

2.2 Mechanical Drift

Mechanical drift is a persistent challenge in robotic musical instruments. Over time, actuators loosen, mallets shift, and hardware components wear, leading to changes in timbre and dynamic output. As a result, an instrument's response can diverge from its calibrated state, introducing inconsistencies in both composition and performance. Traditional calibration methods rely on manual adjustment and subjective listening, which are time-consuming and difficult to reproduce [6, 7]. Because drift progresses slowly, these changes may remain unnoticed until performance becomes unreliable. This motivates the need for systematic, repeatable, data-driven calibration methods.

2.3 Data-Driven Calibration Methods

Sensor-driven and machine learning methods have been increasingly used to characterize and correct the acoustic behavior of robotic instruments. Spectral features such as MFCCs, combined with dimensionality reduction, have been applied to model loudness variation in robotic percussion, improving dynamic stability [8].

Subsequent work extended these ideas through closed-loop control, using audio feedback to automatically correct deviations in pitch, dynamics, and timing [6, 9]. These systems demonstrate that algorithmic calibration can maintain stable performance over extended periods while reducing reliance on manual adjustment.

More recent work focuses on learning richer, instrument-specific response models. Self-supervised exploration enables robots to infer geometric and acoustic properties directly from interaction data [10], while large-scale learning frameworks for robotic piano performance show that modern models can compensate for nonlinear mechanical variation through data-driven policy learning [11, 12].

Collectively, these approaches demonstrate that data-driven modeling can capture complex acoustic behavior, detect drift earlier than human perception, and improve calibration consistency over time. In this work, we introduce a hybrid KNN–MLP framework that integrates data-driven and nonlinear modeling to achieve stable and repeatable calibration.

3. SYSTEM OVERVIEW

3.1 System Architecture

The calibration framework uses an ensemble learning approach [13] that integrates a KNN model with an MLP. A single model often exhibits higher prediction variance, particularly when datasets are sparse. By combining the complementary strengths of both models, the system produces more stable and accurate calibration predictions. The workflow consists of three stages:

1. Constructing an instrument-specific dataset,
2. Training the MLP on the recorded samples, and
3. Generating and blending predictions from both the KNN and MLP.

The system is implemented using ChucK's [14] artificial intelligence library, ChAi [15], enabling direct integration with the Machine Lab code-base and real-time execution.

3.2 Dataset Construction

In the Machine Lab, each instrument requires its own dataset due to differences in actuator design, mechanical layout, and acoustic behavior. Training samples consist of $(\text{note}, \text{velocity})$ pairs with corresponding RMS values that capture the dynamic response of each strike. For each note, strikes are recorded across the full playable velocity range $(60-127)$, with a 1.5-second delay to avoid acoustic overlap. Each strike is stored as a JSON record containing the note, velocity, and RMS value. The overall data collection workflow is summarized in Figure 2.

3.2.1 Pitched Instruments

For pitched instruments such as the MalletOTon [5], notes are sampled across the playable range at configurable intervals, along with all playable velocities. Sampling at different intervals controls dataset density: wider spacing reduces collection time and avoiding redundancy from acoustically similar notes, such as those with similar overtone structure and microphone response, while denser sampling provides more detailed coverage of the instrument's response. The interval is chosen per instrument to balance collection efficiency and prediction accuracy.

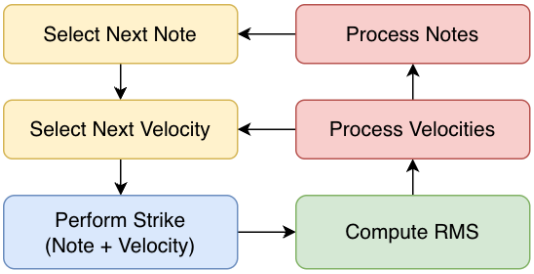


Figure 2. For each note and velocity, the system performs strikes, records RMS values, and iterates until all notes and velocities are sampled.

3.2.2 Non-pitched Instruments

For non-pitched instruments, such as GanaPati, each drum is treated as an independent acoustic source and requires a separate dataset. Individual drum differ in tuning, strike position, and timbral response, making a shared dataset across drums insufficient to capture their distinct characteristics. For each drum, a single mallet or beater is used to maintain consistency, and strikes are recorded using the same procedure as for pitched instruments.

3.3 K-Nearest Neighbors

The KNN [16] model provides predictions grounded in recorded data. By identifying nearest neighbors and interpolating between them, it preserves the instrument's measured acoustic behavior and avoids overfitting in well-sampled regions. However, its utility for full-range calibration is limited by its locality. When data is sparse, linear interpolation cannot capture the underlying nonlinear acoustic variation between notes or velocities. Without smoothing, KNN predictions reflect dataset noise, such as room acoustics and microphone response. This limits its ability to generalize beyond the observed samples.

3.4 Multilayer Perceptron

The MLP [17] complements the KNN model by learning nonlinear relationships beyond local interpolation. By modeling a continuous function, it captures smooth acoustic behavior across the instrument's playable range and reduces the impact of noise present in the dataset. However, these advantages come with limitations. When trained on sparse datasets, an MLP can extrapolate poorly, producing unrealistic RMS estimates near the boundaries of the training set. Its performance also depends on appropriate normalization and sufficient training data, with smaller datasets yielding less reliable behavior.

3.5 Hybrid Ensemble Model

The hybrid model combines the complementary strengths of KNN and MLP by blending their RMS predictions into a single output, as illustrated in Figure 3. The final calibration value is computed as:

$$\text{final\_volume} = \alpha \cdot v_{\text{mlp}} + (1-\alpha) \cdot v_{\text{knn}} \qquad (1)$$

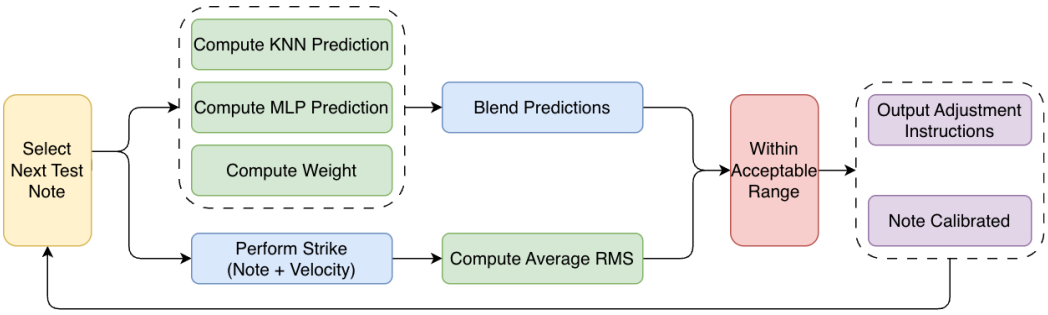


Figure 3. The system computes KNN and MLP predictions, measures the average RMS response, evaluates deviation from the predicted value, and determines whether mechanical adjustment is required.

where, v_{mlp} and v_{knn} denote the RMS predictions of the MLP and KNN models, respectively. The blending weight α is defined as a function of the distance between test sample and nearby training samples, allowing the model to balance local interpolation and nonlinear generalization. When a test sample coincides with a training sample, $\alpha = 0$, relying entirely on KNN. As the distance increases, α increases, progressively incorporating the MLP prediction. This weighting enables a smooth transition between data-driven interpolation and learned generalization.

4. EXPERIMENT

4.1 Experimental Setup

To evaluate the hybrid model, we use four dataset configurations across the playable range of MalletOTon. Specifically, 9, 13, 22, and 26 notes are sampled, corresponding to intervals of 5, 3, 2 semitones, and full coverage, respectively. These configurations simulate different levels of data availability, from sparse to dense sampling of the instrument. All experiments use 5-fold GroupKFold cross-validation, where folds are partitioned by note such that all velocity samples of a given note are assigned to the same fold. This ensures that no $(\text{note}, \text{velocity})$ samples from the same note appear in both training and validation sets, and that all predictions are made on unseen notes, requiring interpolation across notes.

4.2 Ensemble Weight Evaluation

We first examine the peak value of α, which controls the maximum contribution of the MLP. Peak values are varied from 0.15 to 0.85, with the optimal value found at $\alpha_{\max} = 0.60$, yielding the lowest average Mean Absolute Error (MAE) of 1.3853 across all dataset configurations.

Building on this result, we evaluate 61 weighting functions across five function families under the same cross-validation protocol. Table 1 presents the best-performing function from each family. The top-performing functions cluster within 0.006 MAE and consistently share the same peak value, indicating that performance is primarily governed by the maximum weighting rather than the specific functional form. Among the evaluated functions, the flat-top (trapezoidal) function achieves the lowest overall MAE by maintaining the peak across a wider portion of the interpolation range, resulting in a more consistent balance between MLP and KNN contributions.

Model	Parameters	Avg MAE
KNN	–	1.5251
MLP	–	1.4706
Hybrid (Flat)	$\alpha_{\max} = 0.60$	**1.3798**
Hybrid (Para)	$\alpha_{\max} = 0.60$	1.3853
Hybrid (Gauss)	$\alpha_{\max} = 0.60,\ \sigma = 0.4$	1.3880
Hybrid (Tri)	$\alpha_{\max} = 0.60$	1.4003
Hybrid (Sine)	$\alpha_{\max} = 1.00$	1.4664

Table 1. Average MAE across all dataset configurations.

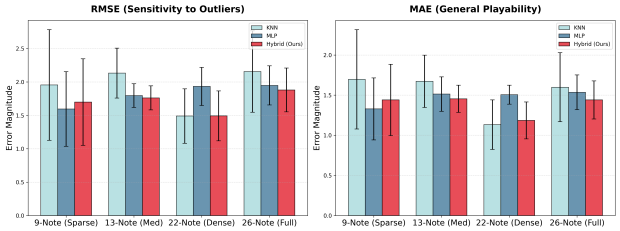


Figure 4. Performance of the hybrid model (flat-top weighting, $\alpha_{\max} = 0.60$) across dataset densities, compared with KNN and MLP using RMSE (left) and MAE (right).

4.3 Model Evaluation

Averaged across all dataset configurations, the hybrid model with flat-top weighting achieves the lowest MAE, outperforming both KNN and MLP (see Table 1). Individually, KNN performs best on densely sampled data (22-note), while MLP performs best under sparse sampling (9-note). However, neither model generalizes consistently across configurations. The hybrid model maintains stable performance without prior knowledge of dataset density. By preserving a KNN contribution in the weighting, it anchors predictions to recorded acoustic measurements while leveraging the MLP for smooth generalization and noise reduction. A tradeoff is observed in variance: sinusoidal weighting yields lower fold-to-fold variance in some cases, while capped functions achieve 5–6% lower MAE. In this context, lower error is prioritized for calibration accuracy.

5. CONCLUSIONS AND FUTURE WORK

This paper introduces a data-driven calibration framework for robotic instruments using a hybrid of KNN and MLP. By combining data-grounded interpolation with nonlinear approximation, the system produces accurate and repeatable predictions for pitched and non-pitched instruments.

Evaluation across multiple dataset densities shows that the hybrid model consistently outperforms standalone approaches, achieving the lowest average MAE. The results further indicate that calibration performance is primarily governed by the maximum blending weight rather than the functional form, providing a simple and robust design principle for balancing interpolation and generalization.

Beyond performance, the system provides a modular calibration pipeline. Its dataset design supports diverse instruments, while adaptive weighting enables stable predictions without requiring prior knowledge of data density. These properties make the framework suitable for long-term deployment where mechanical drift is unavoidable.

Future work will extend the model in several directions. Incorporating additional acoustic features may improve calibration fidelity beyond RMS. Integrating closed-loop feedback could enable continuous adjustment during performance, rather than relying on offline calibration. Finally, exploring more expressive models may improve behavior in sparse or rapidly changing conditions.

Acknowledgments

We thank Michael Darling, Ethan Brewer, Eric Browning, Samantha Cheng, Valentin Correa, Reese Downes, Amogh Dwivedi, Nathaniel Gladstone-Lyon, Ilai Gilbert, Daniel Gonzalez, Jake Morgan, Alexandra Pasquale, Will Rinkoff, Sol Rosenthal, and Ruiqi Wang for their contributions to the Modulets and the CalArts Machine Lab. We also thank Ge Wang, Yikai Li, and the Stanford CCRMA team for developing ChAI, which supported this research.

6. REFERENCES

[1] N. Villicaña-Shaw, S. Salazar, and A. Kapur, "The Machine Lab: A Modern Classroom to Teach Mechatronic Music," in *Proceedings of the 2017 International Computer Music Conference*, Shanghai, China, 2017, pp. 482–487.

[2] A. Kapur, M. Darling, D. Diakopoulos, J. W. Murphy, J. Hochenbaum, O. Vallis, and C. Bahn, "The machine orchestra: An ensemble of human laptop performers and robotic musical instruments," *Computer Music Journal*, vol. 35, no. 4, pp. 49–63, 2011.

[3] A. Kapur, E. Trimpin, A. Singer, G. Suleman, and G. Tzanetakis, "A comparison of solenoid-based strategies for robotic drumming," in *Proceedings of the 2007 International Computer Music Conference*, Copenhagen, Denmark, 2007.

[4] A. Kapur, E. Singer, M. S. Benning, G. Tzanetakis, and Trimpin, "Integrating hyperinstruments, musical robots & machine musicianship for North Indian classical music," in *Proceedings of the 7th International Conference on New Interfaces for Musical Expression*, New York, USA, 2007, p. 238–241.

[5] A. Kapur, J. Murphy, M. Darling, E. Heep, B. Lott, and N. Morris, "Malletoton and the modulets: Modular and extensible musical robots," in *Proceedings of the 16th International Conference on New Interfaces for Musical Expression*, Brisbane, Australia, 2016, pp. 69–72.

[6] J. Murphy, M. Mathews, A. Kapur, and D. Carnegie, "Robot: Tune Yourself! Automatic Tuning for Musical Robotics," in *Proceedings of the 14th International Conference on New Interfaces for Musical Expression*, London, UK, 2014, pp. 565–568.

[7] J. Murphy, D. Carnegie, and A. Kapur, "Using Expressive Musical Robots: Working with an Ensemble of New Mechatronic Instruments," in *Proceedings of International Symposium for Electronic Arts*, 2016.

[8] S. R. Ness, S. Trail, P. F. Driessen, W. A. Schloss, and G. Tzanetakis, "Music Information Robotics: Coping Strategies for Musically Challenged Robots." in *Proc. of the 12th Int. Society for Music Information Retrieval Conf.*, Miami, USA, 2011, pp. 567–572.

[9] J. Long, A. Kapur, and D. A. Carnegie, "The closed-loop robotic glockenspiel: Improving musical robots using embedded musical information retrieval," in *Proceedings of the 16th International Conference on New Interfaces for Musical Expression*, Brisbane, Australia, 2016, pp. 2–7.

[10] M. Görner, N. Hendrich, and J. Zhang, "Pluck and Play: Self-supervised Exploration of Chordophones for Robotic Playing," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 18 286–18 293.

[11] Y. Huang, R. Li, and Z. Tu, "PANDORA: Diffusion Policy Learning for Dexterous Robotic Piano Playing," *arXiv preprint arXiv:2503.14545*, 2025.

[12] L. Chen, Y. Zhao, J. Schneider, Q. Gao, S. Guist, C. Qian, J. Kannala, B. Schölkopf, J. Pajarinen, and D. Büchler, "Dexterous Robotic Piano Playing at Scale," *arXiv preprint arXiv:2511.02504*, 2025.

[13] T. G. Dietterich, "Ensemble methods in machine learning," in *International workshop on multiple classifier systems*, 2000, pp. 1–15.

[14] G. Wang, P. R. Cook, and S. Salazar, "Chuck: A strongly timed computer music language," *Computer Music Journal*, vol. 39, no. 4, pp. 10–29, 2015.

[15] Y. Li and G. Wang, "Chai: Interactive ai tools in chuck," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, Utrecht, Netherlands, 2024, pp. 553–559.

[16] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE transactions on information theory*, vol. 13, no. 1, pp. 21–27, 1967.

[17] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of control, signals and systems*, vol. 2, no. 4, pp. 303–314, 1989.

Studio Report: A Third-Order Periphonic Ambisonics System for Teaching and Research at FEUP and INESC TEC's SMC Lab

José Ricardo Barboza
Faculty of Engineering - University of Porto
jrbarboza@fe.up.pt

Gilberto Bernardes
INESC TEC, Faculty of Engineering - University of Porto
gba@fe.up.pt

ABSTRACT

We present the design, implementation, and educational deployment of a third-order periphonic Ambisonics loudspeaker system at the FEUP and INESC TEC's Sound and Music Computing Lab. The installation comprises twenty lightweight coaxial loudspeakers mounted at the vertices of a dodecahedral layout in four elevation rings, yielding symmetric sampling of the sound field around a central sweet spot. The system decodes 16 higher-order Ambisonics (HOA) channels (N=3) to 20 outputs and was specified to be laptop-friendly, cross-platform, and cost-effective. We justify the choice of HOA over Dolby Atmos and Wave Field Synthesis and detail the geometric derivation of loudspeaker directions, practical mounting solutions, and a calibration workflow that combines precise mechanical alignment with decoder-level angle and gain compensation. Over four years of continuous use, the array has supported courses, theses, and studio projects in immersive audio, with consistent reports of convincing externalization and localization despite modest driver fidelity. We share azimuth/elevation coordinates and integration notes for open-source Ambisonics tools, enabling reproducibility and rapid onboarding for new users. The system offers a flexible foundation for research and teaching and a clear upgrade path toward higher orders and hybrid reproduction formats.

1. INTRODUCTION

The Sound and Music Computing Lab at FEUP and INESC TEC provides the physical and management infrastructure to support pedagogical, research, and development activities in the area of sound and music computing. It hosts integrated researchers at the institution, as well as graduate students pursuing both their PhD and master's degrees, particularly supporting the Doctoral Program in Digital Media and the Master's in Multimedia, both established in 2009. A branch in Interactive Music and Sound Design has been active within these programs.

Via an intensive hands-on curriculum, students aim to attain skills in developing advanced work in the area of interactive music systems, including the design, creation, and operation of these systems in a variety of application contexts, including (but not limited to) games, and interactive installations, as well as the ability to develop advanced work in the field of sound design for cinema and other audiovisual media. Special attention is given in specific classes, such as Sound Design Digital Media, where 3D audio is central to the theory and practice, particularly in fostering innovative content and guideline practices. Previous research on related areas, namely in notating and analyzing [1], spatial annotation cues [2], the impact of spatial audio on attention [3], and music for immersive media [4, 5] have been pursued, to cite a few.

Prior to 2021, all spatial audio activities within the laboratory were executed using *ad hoc* loudspeaker configurations, which were assembled uniquely for each project. In response to the increasing frequency and scope of such activities, a decision was made to implement a permanent multichannel audio system. The design objectives for this installation were twofold. First, to accommodate the technical requirements of established research and creative practices as comprehensively as possible. Second, to provide a flexible foundation capable of supporting anticipated future projects.

The remainder of this report presents the design principles underpinning the selected system configuration (Section 2) and details its physical implementation (Section 3). Finally, Section 4 synthesizes the principal findings of the multichannel system and discusses their implications for laboratory activities.

2. DESIGN

The initial phase of the system design process involved selecting a foundational audio technology. After evaluating the available options, which included Wave Field Synthesis (WFS) and Dolby Atmos, a decision was made to implement a higher-order periphonic Ambisonics framework [6].

This decision was primarily motivated by the following technical characteristics of the Ambisonics approach:

- Full-sphere representation, ideal for VR/AR and psychoacoustic studies;
- Flexible decoding to any audio format;
- Open source technology;
- Cost-effective.

Despite its status as a prevailing industry standard, Dolby Atmos was deemed unsuitable. This determination was based on two principal factors: its significant cost and its

Copyright: ©2026 José Ricardo Barboza et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License 3.0 Unported, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Loudspeaker	Azimuth	Elevation
01	0º	52,62º
02	72º	52,62º
03	144	52,62º
04	-144º	52,62º
05	-72º	52,62º
06	0º	10,81º
07	72º	10,81º
08	144	10,81º
09	-144º	10,81º
10	-72º	10,81º
11	0º	-10,81º
12	72º	-10,81º
13	144º	-10,81º
14	-144º	-10,81º
15	-72º	-10,81º
16	0º	-52,62º
17	72º	-52,62º
18	144º	-52,62º
19	-144º	-52,62º
20	-72º	-52,62º

Table 1. Loudspeakers positioning.

inherent hemispherical reproduction paradigm, which was incompatible with the laboratory's requirements for full-spherical sound projection. Conversely, the Wave Field Synthesis (WFS) system was excluded primarily due to its prohibitive computational demands. A core design prerequisite was that the system must allow users to interface directly with their own portable computers, a requirement that the WFS architecture could not feasibly support.

The second phase of the design process involved defining the operational objectives and physical constraints for the spatial audio reproduction system. The principal objective was to achieve the highest practicable Ambisonic order. A symmetrical loudspeaker array was established as a fundamental design requirement to provide uniform spatial sampling of the sound field, resulting in consistent and predictable spatial resolution (localization accuracy, sound quality) in all directions.

The system configuration was bound by two primary constraints: the dimensions of the installation room and the available project budget.

With the system's objectives and constraints established, the subsequent phase of the design process was to determine its geometric configuration. It was predetermined that the loudspeaker arrangement would conform to a Platonic solid. Following an evaluation aligned with the budgetary limitations, a dodecahedral format was selected. In this configuration, each polyhedron vertex serves as a mounting point for a loudspeaker driver, and the loudspeaker is positioned to face the center of the solid.

As a reference for the speaker placement, we used the arrangement employed in the IKO system [7]. The azimuth angles used are identical, and the elevation angles were offset by 90 degrees to use the center of the solid (sweet spot) as a reference. The angles are listed in Table 1 and the Figure 1 shows the dodecahedron where the speakers are positioned at the vertices facing the center of the symmetrical solid.

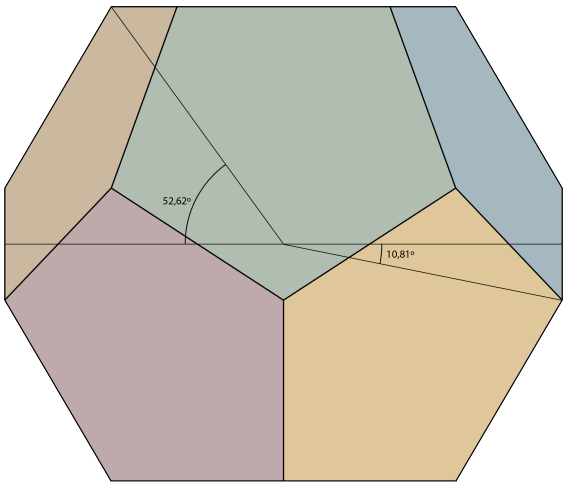


Figure 1. The speaker system arrangement in a dodecahedron shape, with the speakers positioned at the vertices, creating four rings at different elevations.

3. IMPLEMENTATION

Following the finalization of the system design, attention turned to specifying and procuring the necessary equipment. Component selection was guided by technical requirements and budgetary constraints.

Loudspeakers (Fluid Audio FX50): The primary selection criterion mandated the use of coaxial loudspeakers. This design was prioritized for its acoustic advantages, specifically the alignment of driver acoustic centers, which minimizes phase and temporal discrepancies to enhance the accuracy of spatial source localization [8]. Secondary requirements included a wide frequency response and a lightweight physical profile to facilitate mounting.

Audio Interface (MOTU 24 AO): The audio interface was required to offer a minimum of 20 discrete output channels. Furthermore, it necessitated USB connectivity and cross-platform compatibility with both Windows and OSX operating systems to ensure broad user accessibility.

Mounting Hardware: Mounting solutions were specified according to their position within the array. The mounts for the central (Millenium Floor Monitor Stand DM1) and ceiling (Millenium Multi Speaker Wallmount MSW 5) rings required integrated angling mechanisms to achieve precise loudspeaker orientation. In contrast, the mounts for the lower ring were designated to be constructed from wood, providing a stable and cost-effective solution.

Cabling: All connection cables were specified to meet the exact length requirements determined by the physical layout of the installed system.

The definitive dimensions of the dodecahedral loudspeaker array were finalized only after establishing the requisite vertical clearances. These clearances were calculated to accommodate the spatial offsets introduced by the mounting hardware, specifically the specialized ceiling anchors and the custom floor stands.

A tiered mounting strategy was employed according to the speaker ring. The loudspeakers on the upper ring were affixed to the ceiling using purpose-designed anchors that allow post-installation adjustment of the angle. For the intermediate rings, loudspeakers were installed on simple, adjustable stands offering control over both height and an-

gulation. Finally, the loudspeakers of the lower ring were placed on custom wooden stands with a fixed height and inclination.

Precise speaker positioning and angulation were achieved using a laser rangefinder and a digital level. Following the determination of the optimal position for each loudspeaker, the placement of every stand supporting the intermediate and lower rings was permanently marked on the floor. Cable management was implemented with a differential approach: power and signal cables for the fixed upper ring were secured, whereas cables for the lower and intermediate rings were left unfastened. This design facilitates the temporary relocation of these speakers for necessary activities such as maintenance or room reconfiguration.

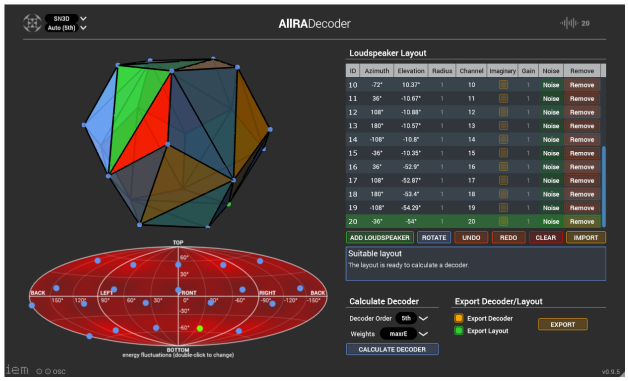


Figure 2. Ambisonic IEM AllRA Decoder plugin and the speakers system's azimuth and elevation final measured angles from the center (sweet spot).

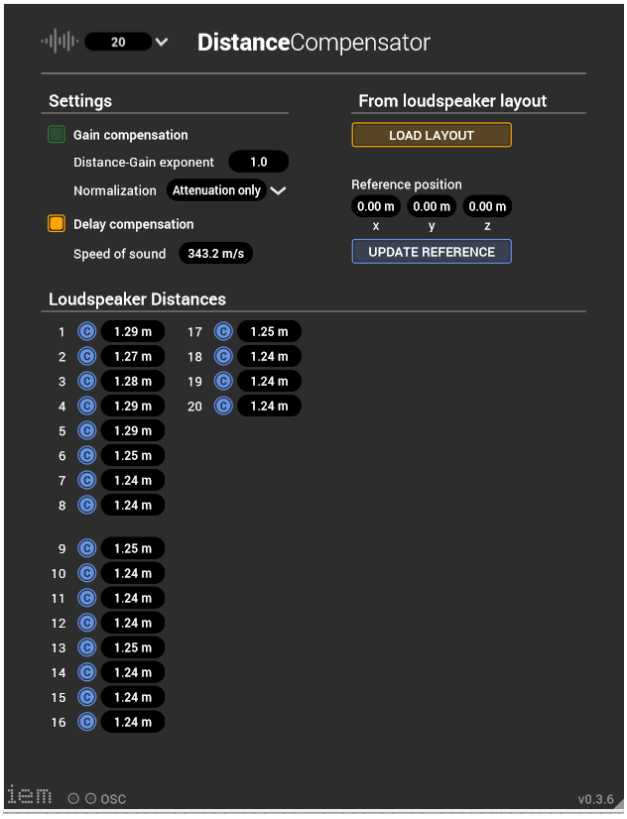


Figure 3. Ambisonic IEM Distance Compensator plugin and the speakers system's final measured distances from the center (sweet spot).

Residual positional errors that could not be resolved through physical adjustments to the loudspeakers or their mounts were compensated for in the digital configuration of the plugin IEM AllRA Decoder (Figure 2) and the plugin IEM Distance Compensator (Figure 3).

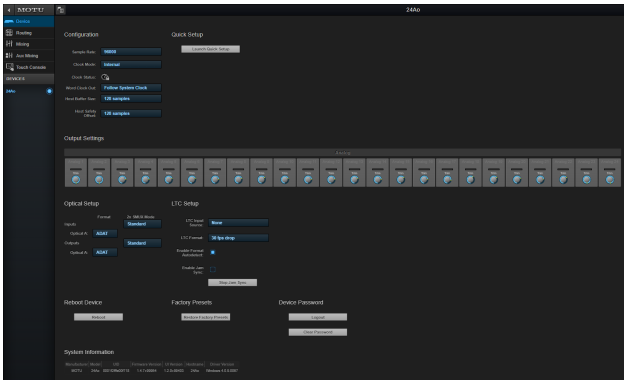


Figure 4. MOTU Pro Audio Control software and the speakers system's final gain reduction to normalize the system SPL.

To finalize the calibration process, the sound pressure level (SPL) of each loudspeaker was measured using an AZ8922 SPL meter. Measurements were conducted at the designated sweet spot, with the meter configured to C-weighting and slow time response. Each loudspeaker reproduced a -20 dB FS pink noise signal, with the target level set at 72 dB SPL [9]. Gain adjustments were subsequently applied within the audio interface's software (MOTU Pro Audio Control - Figure 4) to normalize output levels, thereby ensuring a uniform acoustic response across the entire loudspeaker array.

4. CONCLUSIONS

Over the past four years, the installed multichannel audio system has become a core facility, consistently supporting pedagogical, research, and development activities. It serves as an essential tool for the sound design curriculum, where students often experience spatial audio for the first time. This exposure frequently seeds ideas for master's theses and, in several cases, has led to the continuation of this research line in doctoral studies.

From a technical perspective, the system's sonic fidelity is constrained by the inherent limitations of the selected loudspeakers. However, its subjective performance in spatialization and source localization is consistently reported as highly effective and immersive, a quality that outweighs its fidelity constraints for its primary applications.

A significant design strength is the system's inherent flexibility, facilitated by the 24-channel audio interface with discrete outputs and ADAT inputs. This architecture enabled the integrated installation of a complementary high-fidelity stereo loudspeaker pair, a subwoofer, and a dedicated headphone output. This configuration enables direct, in-situ comparative listening of identical content across multiple formats, including spatial audio, spatial audio with subwoofer, stereo, and stereo with subwoofer. Furthermore, it enables a rapid workflow for recording and evaluation; material can be captured from the sweet spot using the headphone output as a monitor and immediately listened to through the full spatial array.

In summary, the system successfully balances pedagogical utility, research capability, and operational flexibility within its defined budgetary and spatial constraints, establishing itself as a versatile and impactful resource for spatial audio work.

5. REFERENCES

[1] A. Lopes, J. R. Barboza, and G. Bernardes, "Instrument position in Immersive Audio: An empirical review of award-winning practices," *2023 Immersive and 3D Audio: from Architecture to Automotive (I3DA)*, pp. 1–8, 2023.

[2] J. R. M. X. Costa, "Anotações Interativas de Áudio Espacial em Vídeos 360°," Master's thesis, Universidade do Porto (Portugal), 2019.

[3] I. Salselas, R. Penha, and G. Bernardes, "Sound design inducing attention in the context of audiovisual immersive environments," *Personal and Ubiquitous Computing*, vol. 25, no. 4, pp. 737–748, 2021.

[4] J. R. Barboza, G. Bernardes, and E. Magalhães, "A Tripartite Framework for Immersive Music Production: Concepts and Methodologies," in *2025 Immersive and 3D Audio: from Architecture to Automotive (I3DA)*, 2025, pp. 1–7.

[5] J. R. Barboza, E. Magalhães, and G. Bernardes, "Towards Best Practices in Spatial Audio Post Production: A Case Study of Brazilian Popular Music," in *2021 Immersive and 3D Audio: from Architecture to Automotive (I3DA)*, 2021, pp. 1–7.

[6] Z. Zhu, Y. Zhang, W. Guo, C. Pan, and Z. Zhao, "Asaudio: A survey of advanced spatial audio research," in *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, 2025, pp. 417–442.

[7] F. Zotter, M. Zaunschirm, M. Frank, and M. Kronlachner, "A Beamformer to Play with Wall Reflections: The Icosahedral Loudspeaker," *Computer Music Journal*, vol. 41, pp. 50–68, 09 2017.

[8] A. Mäkivirta, J. Väisänen, I. Martikainen, T. Lund, and S. Naghian, "The Acoustic Design of Minimum Diffraction Coaxial Loudspeakers with Integrated Waveguides," *Journal of The Audio Engineering Society*, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:114745160

[9] EBU, "Listening Conditions for the Assessment of Sound Programme Material: Supplement 1 – Multichannel Sound," European Broadcasting Union (EBU), Geneva, Switzerland, EBU Technical Document Tech 3276-E, May 2004.

Data-driven algorithmic composition with large sample libraries: a modular system for the dynamic formation and control of spatialised sound groups

Nikos Baskozos
NKUA
nbaskozos@gmail.com

Thanos Polymeneas-Liontiris
NKUA
athpoli@gmail.com

ABSTRACT

This paper presents a Max/MSP abstraction library for data-driven algorithmic composition. It utilises large sample collections for the formation and temporal control of multiple subselections of the corpus. Each subselection is hosted in a separate container object and may be formed with its own querying rules, using various querying modes. These subselections are typically predefined and can be dynamically recalled and modified. Each subselection container is connected to its own pitch, timing, effects and playback modules. In addition to sequential playback of samples, which is more suitable for melodic and rhythmic explorations, the presented system offers the possibility of vertical playback. The vertical playback module provides temporal control of individual voices and is more suitable for harmonies and spectral techniques. Granular synthesis is possible with both playback modules. The system has been used for the composition of a few fixed-media works. The piece 'Remnants', described in this paper, partially explores the system's available features.

1. INTRODUCTION

This paper introduces a prototype system for data-driven algorithmic composition in the Max/MSP environment [1]. It uses Music Information Retrieval (MIR) along with various querying modes to navigate large sample libraries. It takes the form of an abstraction library, offering multiple modules for each stage of processing.

The analysis and querying modules are based on the Fluid Corpus Manipulation toolkit (FluCoMa) [2] and use code adapted from SP-Tools [3] and Data Knot [4]. Dedicated modules are available for filtering, amplitude envelopes, pitch and timing control. Two playback modules are available, one for sequential triggering of samples and another for vertical control of individual voices over time. Pattrstorage has been used for storing and recalling presets at the macrostructural level. Various configurations have been explored with Spat [5] for per-sample spatial positioning.

Much related research [6, 7] employs large sample libraries, and this system does likewise. Here, a 'large library' is taken to mean any number of samples that exist on the user's storage drive and may fit into RAM. This may amount to tens of thousands of samples, stored without folder organisation or information encoded in their filenames. The system's goal is to provide tools to sculpt sound groups with sufficient control.

The system favours operations applied at the group level rather than on individual samples and, in this sense, is suitable for composition with sound masses, although it is not confined to this approach. Characteristic techniques supported by the system include data-driven granular synthesis and spectral techniques with recorded sound, such as additive instrumental synthesis [8].

Copyright: ©2026 Nikos Baskozos et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License 3.0 Unported, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

2. RELATED WORK

Data-driven, corpus-based musical applications have been used for the past few decades. Most of these applications are designed for particular workflows. In Max/MSP, the MuBu [9] toolbox, and more recently the FluCoMa toolbox, have opened the way to user-defined workflows by providing access to MIR and machine learning technologies within creative coding environments. The presented system builds on this development towards user-defined workflows. Real-time workflows related to concatenative synthesis and corpus-based sampling tend to operate with similarity and interactivity, utilising a similarity-based visualisation of the corpus or audio matching with a live input [10, 6, 3]. This system operates primarily with predefined and dynamically modified queries, without relying on interactive browsing or matching. This approach supports the temporal organisation of multiple subcorpus areas and the integration of data-driven methods with Max/MSP's wider functionality for algorithmic composition.

3. THE SYSTEM

3.1 External libraries

This system relies heavily on external libraries. The analysis of the sample library and the principal querying functions are based on the FluCoMa toolkit [2] and are implemented through code adapted from SP-Tools [3] and its successor Data Knot [4]. The ibuffer~ collection of objects from AHarker Externals [11] provides efficient corpus loading into RAM and playback. While Max/MSP's native polybuffer~ stores audio in 32-bit floating-point format, ibuffermulti~ allows files to be loaded at their original bit depth, reducing memory footprint. The iana~ object by Todor Todoroff, which is included in the Max Sound Box library [12], is used for real-time analysis of the partials of a sound source. The Spat library [5] is used in various configurations for spatial audio rendering. Spat features

a modular architecture that can be adapted to diverse use cases and reproduction configurations, which suits a music system intended to be versatile.

3.2 System description

3.2.1 Corpus analysis and querying

The first step in the system's workflow is the offline analysis, which may require several hours to complete when the library contains tens of thousands of samples. Since segmentation is not part of the process, it is preferable to include only one-shot samples in order to reduce RAM usage. A JSON file is produced from the analysis, which is used to load the samples into RAM and the data into the system. The descriptors available come from the SP-Tools sp.corpusanalysis~ patch. These are loudness, centroid, flatness, pitch, pitch confidence and Mel Frequency Cepstral Coefficients (MFCCs). They are computed for two time frames: the first 4,410 audio samples and the entire duration. Duration and time centroid are also included.

The sample library is loaded into RAM once, and multiple, possibly overlapping subcorpus groups can be configured using different querying modes. Four querying modes are available, each implemented as a separate module: descriptor range queries, K-Nearest Neighbors (KNN) queries with a K-Dimensional tree, cluster queries using the K-Means clustering algorithm, and queries with a list of indices. Descriptor queries can be formed with number boxes or typed in a text editor, with each representation updating the other, allowing them to be used together to form a single query. An 'and' boolean operator defines an intersection of descriptor conditions, and an 'or' boolean operator defines a union of descriptor conditions. The 'or' operator can define two disjoint ranges for a single descriptor, from the minimum to a given value and from a higher value up to the maximum. The descriptor query module is built around the filtering functions included in the sp.corpusmatch patch.

The descriptor query module was implemented first. Intuitively querying a large corpus with descriptor ranges proved challenging, as it was difficult to isolate desired sample groups with sufficient control. This issue was subsequently addressed by implementing KNN queries, based on the fluid.kdtree~ object. KNN queries are performed by typing a sample's name or its unique index number. The KNN function then returns a user-specified number of the most similar samples. A scatterplot, using a slightly modified version of dk.plotter from Data Knot, is optionally employed to browse samples offline and predefine queries. The KNN search function can be built from eight different data sets available in the JSON analysis file: descriptor data and MFCC data computed on the first 4,410 audio samples of each sound, roughly describing the attack, and over the whole duration of each sound. Each of these data sets is available in a normalised and a non-normalised form. When a descriptor data set is used, individual descriptors may be weighted differently for each query, using a function adapted from the dk.corpusmatch patch.

Cluster querying uses the same target-sample selection method and data sets as KNN querying. The difference is that, instead of defining the number of neighbours to return, the user specifies the number of clusters into which the corpus will be partitioned, and the query returns the samples comprising the cluster to which the target sample belongs. Clustering is implemented using fragments of the abstraction patch dk.clustercreate, which is based on the fluid.kmeans~ object.

A feature included in the descriptor query module enables smooth morphing from one preset query to another. Preset descriptor queries may be defined directly with descriptors, or formed via a KNN query for ease of use when a homogeneous selection is desired and then translated into a descriptor range query. The minimum and maximum values for each active descriptor are saved to a coll file as a preset. The user sets an interpolation duration, and when the function is triggered, ramps with the line object are generated from the minimum and maximum values of each active descriptor in the starting query to those of the target query (see Figure 1). At a definable rate, typically once per second, the ramp values are bundled and sent to fluid.datasetquery~ to perform a query. The main difficulty is that the query often returns zero results at the intersection of multiple ramps, causing playback to repeat the last sample. To address this, a hybrid query was implemented. When the descriptor query returns fewer than a user-specified number of samples, KNN is enabled. A sample is selected at random from the sample group returned by the most recent descriptor query to initiate a KNN search. This process is repeated until the descriptor query returns results above the threshold.

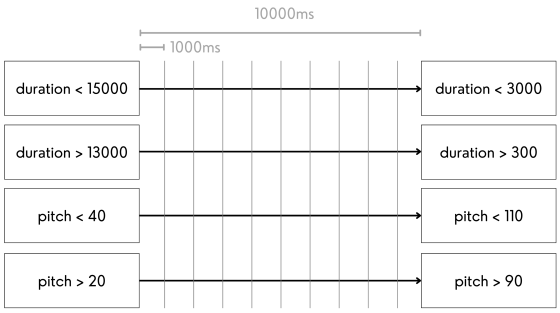


Figure 1. Interpolation between preset descriptor queries.

The system includes two modules used as sample containers. The corpus module contains the audio files, while the subcorpus module is used for subselections of the corpus and stores only sample indices. Each subcorpus module is connected to a parent group, which may be either the corpus module or another subcorpus module. In the latter case, a stacked query is formed, operating as a boolean intersection (see Figure 2). A querying module connects to a subcorpus module, and each query always filters the samples of the parent group one level above. An arbitrary number of subcorpus groups may be created, with arbitrary branching. Such an arrangement is useful when the user first wishes to filter out an area of the large corpus and then form multiple subgroups from that set. At present, this feature has limitations, as only descriptor queries and KNN queries can be stacked. Queries can be executed dynamically, but only at the final branching level. The querying functionalities, and especially the combination of different modes of querying, have considerable scope for additions and improvements.

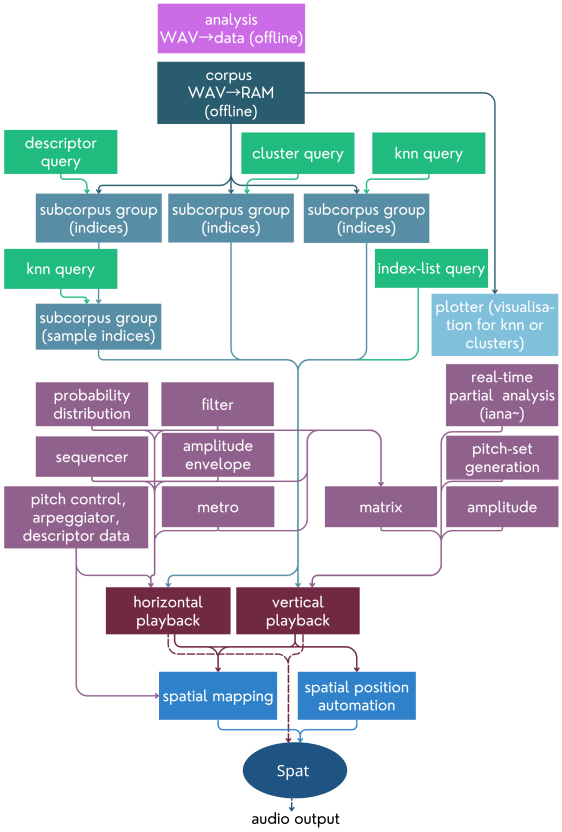


Figure 2. Data and signal flow diagram. Boxes indicate modules. Continuous lines indicate data. Dotted lines indicate audio.

3.2.2 Playback, pitch, timing and effects modules

Two playback modules are available, to which the subcorpus modules and various processing and timing modules are connected (see Figure 2). One operates sequentially, that is horizontally. When a playback command is triggered, a sample from the connected group is selected at random and sent to a poly~ instance. The sample's frequency information from the analysis is also sent to the poly~ instance for pitch control. This playback module is better suited to melodic, rhythmic and granular explorations. The second playback module operates vertically, and the user can control the behaviour of the voices over time. When a voice is switched on with a matrix, three playback options are available: i) it can play back a randomly selected sample from the connected sample group and stop, ii) loop the sample within a definable time range, or iii) select another sample when it is finished. The vertical playback module is more suitable for harmonies, granular synthesis and spectral techniques.

Samples can be played back in three different ordering modes: i) with an urn object, in which case all samples in the group are played once before the list is reset, ii) with a random object for unconstrained random selection, or iii) with a counter object so that they are played back in corpus order. An available function for the horizontal playback module limits the number of samples in the group that will be considered for playback, for greater control of rhythmic and melodic patterns. Additionally, there is a function to repeat playback of the selected sample at any given moment and an option to switch the repeat function on and off at random.

Among the remaining modules in the system, some are shared by the two playback modules, while the pitch control modules differ. The shared modules are a filter, an amplitude envelope, and various modules for time control, including sequencers and probability distribution modules. In the vertical playback case, these modules are first connected to a routing matrix that determines which voices will receive commands from each module separately (see Figure 2). The pitch handling module in the horizontal playback case focuses on scales and arpeggiators. The corresponding module for vertical playback focuses on generating sets of pitches (harmonic series, pitch sets with an arbitrary number of divisions per octave, random frequencies, and manually provided lists of frequencies) and controlling them over time. It includes a feature for stretching or compressing these frequencies and a feature for producing a 'sound splitting' effect (microtonal spacing around a single frequency) [13]. Furthermore, for the vertical playback module, an additional module is included. This module implements real-time partial analysis of an audio input using the iana~ object. When a partial is detected, the module sends the frequency information to a poly~ voice and initiates playback of a sample.

A range can be set in both playback modules within which the sample's playback speed may change to reach the target pitch. If this cannot be achieved within that range, then either a process is activated to select a different sample from the group until one is found, or the sample is omitted from playback. The system's last available feature is to transpose the sample to the nearest note within the target pitch class. The pitch control modules are also responsible for sending descriptor data to other modules, such as those handling spatialisation attributes, so that mappings between descriptors and spatial parameters can be set.

3.2.3 In practice

The system has been tested in various configurations and used in a few compositions focused on the horizontal playback module, as the vertical playback module is a more recent addition. Examples of compositional setups include a single group used for time-varying arpeggios and query interpolation. Another setup uses multiple groups, each corresponding to a note and collectively forming chords. A third setup uses multiple groups for different vertical phases of a single perceptual audio unit implemented with probability distributions. Most often, a corpus of about 20,000 one-shot samples is used, consisting of commercial libraries, personal recordings, and random sounds stored in unsorted folders.

4. A CASE STUDY: REMNANTS

This fixed-media piece explores rhythmic patterns and descriptor query interpolation. It uses four groups of samples and maintains a similar palette of sounds throughout. The horizontal playback module is used for all groups. One transport is used at 130 bpm, and each group has its own metro object. Some level of rhythmic variation is introduced by omitting metro bangs according to an itable object. At times the variation is applied per group, and at other times it is common to all groups. The functions of re-

peating the playback of a sample and limiting the number of samples in the group that will be considered for playback are employed extensively. Limiting a group to fewer than ten samples creates rhythmic patterns that are varied but remain consistent and easy to follow. This playback mode resembles CataRT's beat mode [10] when used with polyphony. Interpolation is used within a single group at a time, which acts like a 'browsing solo' integrated into the rhythm. When the solo is finished, the voice returns smoothly to its normal role. The four groups share a total of 150 Spat sources for a binaural rendering. Spatial positioning focuses on azimuth changes. This includes circular movement through incremental per-source changes in azimuth, randomisation, and mapping to the spectral centroid descriptor. In the latter case, less bright sounds are placed towards the rear and brighter sounds towards the front, while left and right positions are random. The macrostructure was created with a combination of presets and improvisation, leading to long recordings that were edited afterwards.

5. CONCLUSIONS AND FUTURE DIRECTIONS

The system presented in this paper is based on the integration of available data-driven and corpus-based applications with Max/MSP's wider functionality for algorithmic composition. Its main focus is on the use of flexible querying strategies to define and control subcorpus selections over time. This approach provides dynamic access to large sample libraries and enables versatile and aleatoric modes of orchestration. Characteristic design features include rule-based querying and time-varying control of vertically structured voices, supporting granular and spectral techniques with recorded sound.

To a large extent, dealing with an 'as found' sample library and forming desired sound groups proved feasible with the available querying modes. Some level of curation was needed to reduce RAM usage and to remove a few sample collections that did not blend well with the others and were difficult to filter out. A few additional features could further improve the querying efficiency. One could add folder path queries. Another would be to develop a single querying abstraction object that incorporates all querying modes, without having to stack one subcorpus group after another to form a multi-conditional query.

The biggest issues so far have been i) CPU limitations, caused mainly by spatialisation and by triggering a large number of samples per second, and ii) achieving sufficient control over transformations per-group at the macrostructural level. Different approaches will be explored in future work to address both issues. One of these is a Max for Live implementation of the system that is currently under development. In this implementation, each track corresponds to a subcorpus sample group, which enables multithreading and helps address the first issue. Per-group transformations can be controlled using the hierarchical timeline and automation envelopes of the digital audio workstation.

The main development goal at this point is to move on from this prototype towards an application that can be beta tested by other users.

Acknowledgments

Thanks to Areti Andreopoulou, Anastasia Georgaki, Diemo Schwarz, Thibaut Carpentier, Rodrigo Constanzo, Alexander Harker for their support.

6. REFERENCES

[1] D. Zicarelli, "An extensible real-time signal processing environment for Max," in *Proceedings of the 1998 International Computer Music Conference*, 1998.

[2] P. A. Tremblay, G. Roma, and O. Green, "Enabling Programmatic Data Mining as Musicking: The Fluid Corpus Manipulation Toolkit." *Computer Music Journal*, vol. 45, no. 2, p. 9–23, 2022.

[3] R. Constanzo, "SP-Tools," software, version 0.9, 2023, [Online]. Available: https://github.com/rconstanzo/SP-tools.

[4] ——, "Data Knot," software, version 1.0, 2025, [Online]. Available: https://github.com/rconstanzo/data-knot.

[5] T. Carpentier, "A new implementation of Spat in Max." in *15th Sound and Music Computing Conference*, 2018, pp. 184–191.

[6] L. Garber, T. Ciccola, and J. C. Amusategui, "AudioStellar, an open source corpus based musical instrument for latent sound structure discovery and sonic experimentation," in *Proceedings of the 2020 International Computer Music Conference*, 2020.

[7] H. Tutschku, "Composing with Large Sound Collections," https://tutschku.com/videos-type/composing-with-large-sound-collections/, 2020, accessed 17 December 2025.

[8] R. Hasegawa, "Gérard Grisey and the 'nature' of harmony." *Music Analysis*, vol. 28, no. 2-3, pp. 349–371, 2009.

[9] N. Schnell, A. Röbel, D. Schwarz, G. Peeters, and R. Borghesi, "MuBu and Friends – Assembling Tools for Content Based Real Time Interactive Audio Processing in Max/MSP," in *Proceedings of the 2009 International Computer Music Conference*, 2009.

[10] D. Schwarz, G. Beller, B. Verbrugghe, and S. Britton, "Real-time corpus-based concatenative synthesis with catart." in *Proceedings of the 9th International Conference on Digital Audio Effects*, 2006, pp. 279–282.

[11] A. Harker, "AHarker Externals," software, GitHub repository, version latest-2025-05-10, 2025, [Online]. Available: https://github.com/AlexHarker/AHarker_Externals.

[12] IRCAM, "Max Sound Box," software, Max/MSP library, version 2022-09, 2022, [Online]. Available: https://forum.ircam.fr/projects/detail/max-sound-box/.

[13] R. Hasegawa, "Clashing Harmonic Systems in Haas's Blumenstück and in vain." *Music Theory Spectrum*, vol. 37, no. 2, pp. 349–371, 2015.

Hyponoia:
An Affective Computing System for Augmented Musical Performance — A Case Study

Penelope Bekiari

National and Kapodistrian University of Athens
Department of Digital and Cinema

pbekiari@dcarts.uoa.gr

Anastasia Georgaki

National and Kapodistrian University of Athens
Department of Music Studies

georgaki@music.uoa.gr

ABSTRACT

This paper investigates how EEG-driven biofeedback systems influence performability and listening strategies in contemporary electroacoustic performance. We introduce Hyponoia (Hyper-Observational Neuro-Oscillation Interactive Agency), a real-time interactive system that translates performers' neurophysiological activity into state-based compositional behaviours. Unlike conventional EEG-based musical interfaces that map signals to discrete parameters, Hyponoia operates at the level of musical processes, structuring sonic form through inferred neuro-affective states. The system integrates EEG and heart-rate data within a closed biophysical feedback loop, in which performers' internal cognitive and affective states dynamically interact with the evolving sonic environment. A comparative case study was conducted with expert and non-expert musicians performing in open-form electroacoustic contexts. We hypothesise that performers with sound-based expertise exhibit distinct patterns of neural engagement and interaction with the system. Results indicate that expert performers demonstrate richer theta activity, more coherent alpha modulation, and greater neural variability, associated with enhanced internal auditory imagery and anticipatory listening. In contrast, non-expert performers exhibit more constrained neural responses and reduced sensitivity to spectromorphological change. These findings suggest that performability in biofeedback-driven systems depends less on instrumental technique than on listening literacy and embodied sonic awareness. Rather than acting as an autonomous agent, the system functions as a responsive mediator that amplifies differences in perceptual and cognitive engagement, contributing to an emerging performance aesthetic grounded in physiological feedback and real-time interaction.

Copyright: © 2026 Penelope Bekiari et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License 3.0 Unported, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

1. AFFECTIVE TECHNOLOGIES IN BIOMUSIC : STATE OF THE ART

Since the early twentieth century, the performing body has increasingly been conceptualized as a sonic-generative entity: a corporeal instrument whose expressive and physiological capacities may structure, modulate, and even originate sound. This reflects a broader shift in artistic practice that foregrounds the body's materiality, affective presence, and internal rhythms as central components of performance. The question, however, is not only how the body acquired this acoustic agency, but why modern and contemporary culture developed such a strong impulse to render internal physiological processes audible. Part of the answer lies in the rapid evolution of electronic and digital technologies, which opened new pathways into the biophysical interior of performance. Systems capable of measuring heart-rate variability, galvanic skin response (GSR), electromyography (EMG), and electroencephalography (EEG) gradually migrated from laboratory environments into artistic contexts [1], [2]. Although these technologies were initially designed for clinical, neuroscientific, and human–computer interaction research, artists adopted them to explore autonomic, prereflective, and affective aspects of bodily experience. Their artistic appeal lies in continuity: physiological processes persist even in stillness, generating an involuntary stream of data through which performance may be extended, transformed, or reimagined.

This technological turn coincided with a major epistemic shift in performance theory. Contemporary scholarship increasingly describes performance as relational, embodied, and dynamically co-constituted rather than as a stable representation of a work [3], [4]. In music performance, this implies that interpretation is inherently performative: the body does not simply execute a score but actively shapes temporal, affective, structural, and sound-parameter dynamics [5]. Within this framework, the concept of performability designates not technical feasibility but the extent to which a work emerges through the performer's embodied, affective, and cognitive engagement.

Biofeedback technologies therefore introduce not only new tools but new performative and compositional

modes. Physiological signals enable the materialisation of otherwise invisible internal states, transforming them into sonic, spatial, or interactive elements. This reconfigures the performer–instrument relationship into a neurophysiologically mediated field of interaction, in which expressive musical intention, affective perception, and bodily regulation are continuously interwoven. Recent computational and neurophysiological research demonstrates that musical performance is shaped by dynamic couplings between acoustic features, emotional communication, and neural activity, situating physiological processes as integral to expressive agency rather than as secondary correlates [6]. Research in affective computing further indicates that biosignals reveal fluctuations in arousal, attention, and cognitive load, allowing these dynamics to operate as compositional resources [7], [8]. At the same time, neuroaesthetic studies show that music performance is deeply entangled with neural mechanisms of prediction, embodiment, and emotional resonance [9], [10]. Integrating EEG, ECG, or GSR into performance thus establishes a feedback loop between internal cognitive–affective states and the external sonic environment, foregrounding the performer's physiological and sensory agency as a core component of musical form.

The integration of physiological and affective technologies into music has therefore generated a significant reconceptualisation of the body as both expressive and generative medium. Advances in wearable sensing, affective computing, and physiological signal processing have enabled artistic systems that treat somatic activity not as hidden background behaviour but as a structural agent shaping musical form, temporal development, and performative experience [2], [7]. Within this landscape, physiological data cease to function merely as indicators of emotion and instead become active compositional material. Examples include Lisa Park's *Eunoia* and Angie C's *Tonto*, installation-performances in which electrodermal activity, respiration, and cardiac rhythm structure sonic form through affective modelling, modulating timbral density, spectral morphology, and spatial diffusion. The body does not accompany the performance; it constitutes it — resonating with neuroaesthetic accounts of prediction, embodiment, and sensorimotor coupling [9], [10].

A collective articulation of physiological performativity appears in *Heart Chamber Orchestra* [11], where twelve musicians wear ECG sensors that stream cardiac data into a central generative system. The ensemble's shared physiological behaviour generates real-time musical scores that are immediately performed, creating a recursive loop of biological feedback and musical response. A related trajectory grounded in neural activity extends the lineage inaugurated by Alvin Lucier's *Music for Solo Performer* [12]. Contemporary EEG-based systems — particularly those developed by Eduardo Reck Miranda — translate alpha, theta, and beta activity into real-time sonic transformations and generative score processes [13]. In these environments, cognition becomes a compositional parameter that interacts with algorithmic behaviour and performer intention.

Collectively, these practices illustrate a decisive movement from representational expression toward embodied co-production. Musical form becomes emergent rather than fixed, contingent upon physiological variability and sensory engagement. This resonates strongly with Umberto Eco's notion of the *open work* [14]: in physiological performance, openness is not only aesthetic but biological, as neural and autonomic variability render every performance inherently unrepeatable

Within this context, the need emerged to develop a system capable of interacting dynamically with neurophysiological data in performance, not simply as a control layer but as an active contributor to musical structure and timbral identity. The intention was to explore how such interaction might reshape performability itself, foregrounding the affective, sensory, and self-regulatory dimensions of musicianship through a continuously responsive biofeedback environment..

2. HYPONOIA —HYPER-OBSERVATIONAL NEURO-OSCILLATION INTERACTIVE AGENCY: AFFECTIVE TECHNOLOGIES IN PERFORMANCE

Hyponoia derives from the Greek term meaning *inner thought or underlying meaning*, evoking the invisible cognitive–affective strata that shape perception and expression. The term frames performance as a dialogue between internal physiological states and external sonic form. Within this perspective, affective technologies do not simply monitor the body but participate in the emergence of musical structure, allowing inner neural and autonomic processes to become active agents in performance.

2.1. Architecture of the System - Hyponoia

To investigate performability through real-time neurophysiological interaction, we designed Hyponoia (Hyper-Observational Neuro-Oscillation Interactive Agency), a closed-loop biofeedback system that maps EEG-derived states to compositional behaviours. EEG data were acquired using the Muse 2 headset from four electrodes (TP9, TP10, AF7, AF8), sampled at 256 Hz and streamed via Lab Streaming Layer (LSL) into Python. Heart rate (HR) was recorded in parallel and normalised using performer-specific minimum–maximum ranges to account for individual differences in physiological arousal. EEG signals were band-pass filtered (1–45 Hz) and notch filtered at 50 Hz to remove power-line interference. Power spectral density (PSD) was estimated using Welch's method (Welch, 1967) over sliding windows of 2 seconds with 50% overlap. Absolute bandpower values were computed for four canonical frequency bands: theta (4–7 Hz), alpha (8–12 Hz), beta (13–30 Hz), and gamma (30–45 Hz).

Mean bandpower values were calculated across all electrodes at each time step to reflect distributed cortical engagement. In parallel, broadband neural activity was estimated using root mean square (RMS) amplitude computed over the same sliding window. To ensure performer-specific sensitivity, bandpower values were z-normalised using a rolling baseline buffer of approxi-

mately 60 seconds. This adaptive normalisation accounts for baseline drift and inter-individual variability in EEG activity. Bandpower values exceeding $z > 1$ were classified as high-activation events. This threshold was selected empirically following preliminary testing across thresholds (0.5–1.5), where $z > 1$ provided a balance between sensitivity and stability in real-time performance conditions. To reduce noise from movement artefacts—particularly relevant for wind instrument performance—segments with excessive RMS amplitude were attenuated, and rapid transient spikes were smoothed using temporal averaging.

To prevent rapid and unstable switching between states, a temporal smoothing mechanism was applied. A state transition is only confirmed if the condition persists for at least two consecutive analysis windows (~4 seconds). Additionally, abrupt transitions between unrelated states are constrained via intermediate neutral states, reducing perceptual discontinuities in the resulting sound. The inferred state, along with continuous z-score values and HR-derived arousal zones, is transmitted via OSC to Max/MSP. Importantly, the system does not operate as an autonomous agent; rather, it functions as a responsive mediation layer, translating performer physiology into structured compositional behaviours. To support reproducibility, the implementation of the system is available at: `GitHub link.`

2.2. Mapping

State No.	State Name	Process	Sonic Behaviour
0	Neutral	Baseline, no dominant rhythms	No FX
1	Alpha-Theta	Resonance with sustained note	Pitch-recognised synth accompaniment
2	Beta-Gamma	Live granulation	Live granulation
3	Beta-Alpha	Reverse delay	Reverse delay
4	Alpha-Gamma	Loop of the previous phrase	Loop of the previous phrase
5	Theta dominant	High time-stretched downward reverberation	High time-stretched downward reverberation

Figure 1. Mapping triggering procedure in Max/MSP

Once a mental state is inferred by the classifier, it is returned to the performance environment, activating a corresponding compositional behaviour within the electronic layer. The mapping strategy is designed at the level of process and temporal behaviour, rather than surface parameter control, allowing the performer's internal regulation to be externalised as evolving sound. The neutral state maintains sparse textures, low dynamic activity, and minimal reverberant residue, functioning as a point of sonic equilibrium. The alpha–theta state, associated with relaxed attention and inward focus, activates diffuse granular clouds, slow spectral stretching, and extended reverberation, producing a sense of temporal suspension and perceptual inwardness. Through this mapping strategy, temporal persistence, resonance, and reverberant decay become primary carriers of form, allowing internal states to be translated into externalised corporeal behaviour in sound. Musical form thus emerges through a continuous oscillation between internal regulation, reverberant inscription, and mnemonic return, situating performability as an embodied negotiation between sound, memory, and bodily presence in real time (see Figure 2).



Figure 2. Performance of one the Pieces - Live set. The performer wears the headset and the heartbeat sensor.

3. A COMPARATIVE STUDY

This study investigates how performers with different forms of musical expertise engage with an EEG-driven interactive system in open-form electroacoustic performance. The primary aim is to examine how sound-based listening practices influence neural activity, embodied interaction, and performability within a biofeedback environment. We hypothesise that performers with experience in electroacoustic and sound-based practices will exhibit distinct patterns of neural engagement—particularly in theta and alpha activity—reflecting enhanced internal auditory imagery, attentional regulation, and sensitivity to spectromorphological change. The study adopts an exploratory case-study design. Due to the small sample size, the results are not intended for statistical generalisation but rather to provide insight into the interaction between performer expertise and biofeedback-driven systems.

3.1. Participants and Expertise Criteria

The study involved four performers across two instrumental conditions: cello and trumpet. For each instrument, one performer was classified as an expert and one as a non-expert. Expertise was defined not in terms of technical virtuosity alone, but in relation to sound-based musical practice.

Expert performers met the following criteria:

- at least two years of experience in electroacoustic music, sound-based composition, or contemporary performance practice
- familiarity with spectromorphological listening and timbral analysis
- experience with open-form structures and non-notational approaches to music-making
- prior engagement with interactive or technology-mediated performance environments

Non-expert performers were trained in classical or jazz instrumental traditions but had limited exposure to electroacoustic or sound-based practices. Their performance approach was primarily notation-based and oriented toward pitch, rhythm, and predefined structure. All performances were conducted in a controlled rehearsal environment using the same electroacoustic soundscape and system configuration. Each performer improvised over a pre-composed soundscape while wearing the EEG headset and heart-rate sensor. No audience was present during data collection, minimising external variability. Performers were asked to improvise within an open-form electroacoustic environment guided by a driver score. This score provided structural cues—such as density, gesture, and spectral evolution—without imposing fixed pitch or rhythmic constraints. The driver score functioned as a shared experimental framework, ensuring comparability across performances while preserving interpretative freedom. Rather than executing predefined material, performers engaged with the soundscape through listening, gesture, and timbral interaction. This design allowed the study to focus on performability as an emergent property of embodied interaction, rather than as the execution of a fixed musical structure.

Several limitations should be acknowledged. First, the study is based on a small sample size (n = 4), which limits generalisability. The findings should therefore be interpreted as indicative rather than conclusive. Second, while EEG preprocessing steps were applied, real-time performance conditions—particularly for wind instruments—introduce potential artefacts that may influence bandpower estimation. Third, the threshold for high activation ($z > 1$) and the rule-based classification model are based on empirically derived parameters. Although preliminary testing informed these choices, further validation and sensitivity analysis are required. Finally, the present analysis focuses on aggregate neural patterns rather than event-based correlation with specific musical gestures. Future work could examine how neural activation aligns with particular moments in the performance or score, providing a more fine-grained understanding of performer–system interaction.

3.2. Cello Case Study: Neural Activity Patterns

In the cello performance, the expert performer exhibited consistently higher neural activation across all measures relative to the non-expert. Mean theta activity was substantially elevated, with multiple extended episodes surpassing the high-activity threshold ($z > 1$), indicating prolonged periods of internal auditory engagement and predictive sound processing. Alpha modulation also presented more stable and coherent patterns in the expert, reflecting efficient attentional regulation during transitions between gestures and textural states. RMS values across raw EEG channels (TP9, TP10, AF7, AF8) were notably higher and more variable, indicating greater oscillatory richness and dynamic neural responsiveness. By contrast, the non-expert performer demonstrated lower-amplitude bandpower profiles, limited theta activation and fewer discrete high-activity events. RMS traces were comparatively narrow and less dynamic, suggesting more constrained neural responses to the sound-based task. A small number of simultaneous theta microstates were observed across both performers; however, these were shorter and predominantly driven by peaks in the expert's signal.

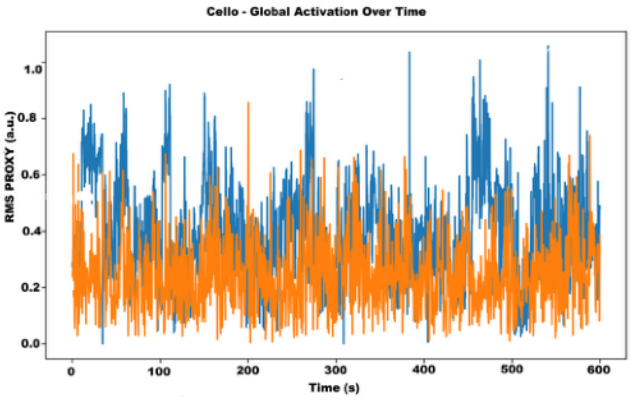


Figure 3. Comparative Neural Activation in Expert and Non-Expert Cello Performance

3.3. Trumpet Case Study: Neural Activity Patterns

The trumpet condition further reinforced the distinction between expert and non-expert neural profiles. The expert trumpet performer exhibited markedly elevated mean theta activity—approximately an order of magnitude greater than that of the non-expert—and more pronounced alpha regulation across the performance duration. Theta bursts were longer, more structured and more frequent, indicating deeper involvement in internal auditory imagery and sound-based attentional strategies. RMS analysis revealed strong broadband activation, with raw-channel amplitudes two to three times higher than those of the non-expert performer. In contrast, the non-expert trumpet performer displayed restricted neural variability, low oscillatory intensity and very few occurrences of theta or alpha values exceeding the high-activity threshold. Temporal inspection showed that the expert maintained heightened neural dynamism throughout the performance, whereas the non-expert remained close to baseline for sustained intervals. As with the cello data, a limited number of simultaneous microstates occurred across both performers, but these were brief and largely dominated by activity peaks from the expert's neural signal.

3.4. Analysing Results

Across both instruments, the study reveals that sound-based expertise shapes neural engagement in ways that

diverge from traditional instrumental training. The expert performers were not simply more skilled; rather, they demonstrated a deeper capacity to understand the logic of sound—its gestures, textures, trajectories, and spectral identities. Their background in electroacoustic practice and contemporary listening aesthetics afforded them fluency in Smalley's spectromorphological principles, including gesture, energy–motion, spectral evolution, and referential sound imagery [16]. This listening literacy was directly reflected in the EEG data. Expert performers exhibited richer and more structured theta oscillations, which have been associated with internal auditory imagery, affective engagement, and predictive shaping in musical cognition [17]. Their alpha modulation was more coherent and efficient, aligning with research showing that musical expertise enhances higher-order brain functioning and attentional control [18]. Within the broader theoretical framework of music cognition proposed by Peretz and Zatorre [19], these neural patterns indicate deeper integration of auditory, motor, and associative networks—a mode of embodied sonic thinking grounded not in notation but in the perceptual organisation of sound. In practice, expert performers appeared able to "speak the language of sound." Their interactions with the Max/MSP system showed anticipatory sensitivity to gesture, spectral density, and dynamic motion, enabling them to navigate the environment as a responsive sonic ecosystem.

Research on alpha/theta neurofeedback provides strong theoretical support for the patterns observed in the present study. The work of Gruzelier proposes that the enhancement of theta relative to alpha activity is associated with increased long-distance functional connectivity between frontal and posterior cortical regions, facilitating creative performance and psychological integration in expressive domains such as music and dance [20]. This state has been linked to heightened internal imagery, relaxed attentional control, and deeper embodied engagement with creative tasks. Complementary studies further indicate that alpha/theta modulation supports sustained affective attention, introspective awareness, and integrative modes of cognition that are crucial for creative performance [21]. Within the context of the present biofeedback-driven performances, these findings suggest that elevated theta and coherent alpha activity do not merely reflect physiological activation but correspond to an embodied mode of listening and performative cognition, through which performers regulate attention, imagination, and sonic anticipation. This theoretical framework reinforces the interpretation that sound-based expert performers may access and stabilise such internal states more efficiently, allowing bodily regulation, listening perception, and performability to converge within the interactive sound environment. Their neural signatures—elevated theta, stable alpha, and higher raw signal richness—suggest that they engaged the system intuitively, treating sound as a living and malleable medium rather than a symbolic representation. Non-experts, by contrast, displayed neural patterns indicative of externally oriented, effortful engagement: flatter oscillations, weaker theta activity, and reduced spectral sensitivity. Their limited exposure to sound-based practices led to difficulty perceiving micro-gestures and spectral cues, resulting in attempts to map traditional pitch-based strategies onto an open-form electroacoustic framework. This contrast highlights that expertise in contemporary sound-based music is not reducible to technical virtuosity; it requires a cognitive reframing of music as sound-movement rather than pitch-execution—an expertise fundamentally situated in listening, embodiment, and perceptual imagination. Crucially, the findings of this case study open an important dialogue: interactive systems that rely on biodata appear to function more fluidly and meaningfully with performers who possess an open, exploratory relationship to sound. These performers tend to interpret the system's feedback not as a constraint but as part of an expanded performative ecology, responding adaptively to subtle changes in their own neural and physiological signals. This observation is significant for the performing arts, as it suggests that contemporary sound-based musicianship is increasingly aligned with modes of performability that foreground listening, perception, feedback sensitivity, and corporeal awareness.

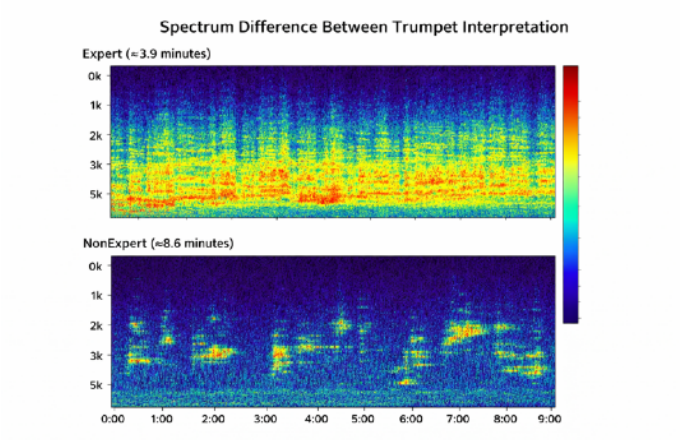


Figure 4. Spectrum difference between trumpet interpretations. The combined spectrogram provides a direct visual comparison of the spectral energy distribution across the two recordings. In the upper panel, the Expert performance (≈3.9 minutes) shows consistently higher average acoustic energy, with dense and continuous activity in the low- to mid-frequency bands, producing a fuller and more saturated spectral texture. In contrast, the lower panel, representing the Non-Expert performance (≈8.6 minutes), distributes its energy over a longer duration and displays a more discontinuous pattern, characterised by localised peaks and generally lower intensity within the same frequency regions. The comparison highlights the distinction between a stable, energetically sustained sound field and a more sparse, fluctuating spectral profile.

4. DISCUSSION

The findings of this study suggest that differences observed between expert and non-expert performers cannot be attributed solely to technical proficiency, but rather to distinct modes of listening, perception, and interaction with the biofeedback system. While expert performers demonstrated higher neural variability, increased theta activity, and more stable alpha modulation, these patterns should not be interpreted simply as markers of superior performance. Instead, they reflect a greater capacity to

engage with sound as a dynamic, spectromorphological process [16]. Increased theta activity has been associated with internal auditory imagery and predictive listening processes [17], while stable alpha modulation has been linked to attentional regulation and higher-order cognitive control in musical contexts [18].

Crucially, the results indicate that the Hyponoia system does not function as a neutral interface but as a responsive mediator that amplifies underlying differences in listening strategy. Performers with experience in electroacoustic and sound-based practices appear better able to interpret and respond to the system's feedback, integrating it into their performative decision-making. In this sense, the system privileges forms of listening that are sensitive to texture, gesture, and spectral evolution, rather than to pitch-based or notation-driven structures. This aligns with broader frameworks in music cognition that emphasise the integration of auditory, motor, and associative processes in musical expertise [19]. This reframing shifts the interpretation of the results: rather than concluding that expert performers are more "engaged," it suggests that the interaction between performer and system is contingent upon a form of perceptual literacy. The system effectively exposes how different musical backgrounds shape the way performers inhabit and navigate an open-form sonic environment. Research on alpha–theta neurofeedback further supports this interpretation, suggesting that increased theta relative to alpha activity is associated with enhanced creative performance, internal imagery, and integrative cognitive states [20], [21].

Within the context of contemporary performance aesthetics, Hyponoia aligns with a shift toward performative structures that prioritise contingency, instability, and real-time negotiation over predetermined formal hierarchies. Rather than executing a fixed score, performers engage in a continuous feedback loop in which internal physiological states and external sonic processes co-evolve. In this framework, performability is redefined as the capacity to regulate attention, perception, and embodied response within a dynamically changing environment. Importantly, the system does not operate as an autonomous agent. While it introduces structured transformations based on physiological input, these processes remain dependent on performer activity and interpretation. Agency therefore resides with the performer, while the system functions as a responsive and constraining mediation layer that shapes, rather than determines, the unfolding musical form.

A related perspective is articulated by Spitzer, who argues that contemporary music–technology design increasingly seeks to integrate itself into the body's internal processes rather than extend external instrumental control [24]. From this perspective, musical expressivity emerges through micro-gestures, proprioception, and embodied awareness, blurring the boundary between performer and technological system.This perspective suggests that biofeedback systems do not merely extend existing performance practices but actively contribute to the emergence of new modes of structuring and performative capability. Performability here is no longer defined by the execution of a score, but by the ability to navigate an evolving sonic ecology shaped by feedback, embodiment, and perception. This resonates with contemporary aesthetic tendencies found across electroacoustic and post-digital practices, where instability, fragmentation, and affective intensity foreground process over permanence [22], [23]. A key limitation of the present analysis is that it focuses on aggregated neural measures rather than event-based correlations between musical actions and physiological responses. Future work could examine how specific moments in the performance—such as changes in texture, density, or gesture—relate to fluctuations in neural activity, enabling a more fine-grained understanding of performer–system interaction.

5. CONCLUSION

In conclusion, the study demonstrates that neural and physiological responsiveness varies significantly according to the performer's experience with sound-based and electroacoustic practices. Expert performers showed greater neural adaptability, smoother transitions between cognitive and attentional states, and stronger integration of auditory and embodied processes during interaction with the system. Non-experts, by contrast, displayed more constrained neural patterns and limited engagement with the open-form, sound-centre framework of the piece. These findings suggest that contemporary performance environments involving biodata and interactive systems benefit from performers who approach sound as a dynamic, perceptual and gestural medium rather than solely through traditional pitch-based or technique-centred models. As such, the results point toward evolving definitions of performability in contemporary and electroacoustic music—definitions that emphasise listening acuity, perceptual awareness, and the capacity to respond creatively to dynamic sonic and physiological information. This case study therefore highlights both the artistic and research potential of biofeedback-enhanced performance, especially as such technologies become increasingly accessible to musicians and integrated into new forms of open-form creative practice.

Acknowledgments

The authors would like to thank Dr. Manousos Klados for valuable insights and support regarding the neurophysiological aspects of this research.We also express our sincere gratitude to Associate Professor Alexandros Tzallas (Department of Informatics and Telecommunications, University of Ioannina) for his expertise in biomedical engineering and guidance on EEG-related methodologies. In addition, we thank Assistant Professor Orestis Karamanlis (Department of Digital Arts and Cinema, National and Kapodistrian University of Athens) for his contribution to the development of compositional ideas and for his support in shaping the artistic direction of the work.

6. REFERENCES

[1] A. Tanaka, "Sensor-based musical instruments and interactive music," in *The Oxford Handbook of Computer Music*, R. T. Dean, Ed. Oxford, UK: Oxford University Press, 2009, pp. 233–257.

[2] I. Daly, D. Williams, F. Hwang, A. Kirke, E. R. Miranda, and S. J. Nasuto, “Electroencephalography reflects the activity of subcortical brain regions during approach–withdrawal behaviour while listening to music,” *Scientific Reports*, vol. 9, art. no. 9415, 2019.

[3] H.-T. Lehmann, *Postdramatic Theatre*. London, UK: Routledge, 2006.

[4] E. Fischer-Lichte, *The Transformative Power of Performance: A New Aesthetics*. London, UK: Routledge, 2008.

[5] M. Leman, *The Expressive Moment: How Interaction (with Music) Shapes Human Empowerment*. Cambridge, MA, USA: MIT Press, 2016.

[6] Lyberatos, V., Kantarelis, S., Zioga, I., Anagnostopoulou, C., Stamou, G., & Georgaki, A. (2025). *Music interpretation and emotion perception: A computational and neurophysiological investigation*. arXiv.

[7] S. H. Fairclough, “Fundamentals of physiological computing,” *Interacting with Computers*, vol. 21, no. 1–2, pp. 133–145, 2009.

[8] R. A. Calvo and S. K. D’Mello, “Affect detection: An interdisciplinary review of models, methods, and applications,” *IEEE Transactions on Affective Computing*, vol. 1, no. 1, pp. 18–37, 2010.

[9] E. Brattico and M. Pearce, “The neuroaesthetics of music,” *Psychology of Aesthetics, Creativity, and the Arts*, vol. 7, no. 1, pp. 48–61, 2013.

[10] S. Koelsch, “Brain correlates of music-evoked emotions,” *Nature Reviews Neuroscience*, vol. 15, no. 3, pp. 170–180, 2014.

[11] E. Berger and PURE, “Heart Chamber Orchestra,” *Leonardo Music Journal*, vol. 20, pp. 15–18, 2010.

[12] A. Lucier, *Music for Solo Performer*, 1965.

[13] E. R. Miranda and J. Castet, Eds., *Guide to Brain–Computer Music Interfacing*. London, UK: Springer, 2014.

[14] U. Eco, *The Open Work*. Cambridge, MA, USA: Harvard University Press, 1989.

[15] P. D. Welch, “The use of fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms,” *IEEE Transactions on Audio and Electroacoustics*, vol. 15, no. 2, pp. 70–73, 1967.

[16] D. Smalley, “Spectromorphology: Explaining sound-shapes,” *Organised Sound*, vol. 2, no. 2, pp. 107–126, 1997.

[17] J. Bhattacharya and H. Petsche, “Phase synchrony analysis of EEG during music perception,” *NeuroReport*, vol. 16, no. 7, pp. 737–740, 2005.

[18] M. S. Oechslin, M. Meyer, and L. Jäncke, “Neural plasticity in musicians,” *Frontiers in Psychology*, vol. 4, art. no. 1, pp. 1–10, 2013.

[19] I. Peretz and R. J. Zatorre, Eds., *The Cognitive Neuroscience of Music*. Oxford, UK: Oxford University Press, 2003.

[20] J. H. Gruzelier, “EEG-neurofeedback for optimising performance. I: A review of cognitive and affective outcome in healthy participants,” *Neuroscience and Biobehavioral Reviews*, vol. 44, pp. 124–141, 2014.

[21] J. H. Gruzelier, “EEG-neurofeedback and biofeedback: A review of affective and cognitive effects,” *Applied Psychophysiology and Biofeedback*, vol. 34, no. 2, pp. 73–89, 2009.

[22] M. Fisher, *Ghosts of My Life: Writings on Depression, Hauntology and Lost Futures*. Winchester, UK: Zer0 Books, 2014.

[23] A. Harper, *Infinite Music: Imagining the Next Millennium of Human Music-Making*. Winchester, UK: Zer0 Books, 2019.

[24]M. Spitzer, *The Musical Human: A History of Life on Earth*. London: Bloomsbury, 2021.

Perpetual Dialogues: A Computational Analysis of Voice–Guitar Interaction in Carlos Paredes's Discography

Gilberto Bernardes
INESC TEC, Faculty of Engineering, University of Porto
gba@fe.up.pt

Nádia Moura
CEIS20, Faculty of Arts and Humanities, University of Coimbra
nmoura@fl.uc.pt

António Sá Pinto
INESC TEC, Faculty of Engineering, University of Porto
asapinto@fe.up.pt

ABSTRACT

Computational musicology enables systematic analysis of performative and structural traits in recorded music, yet existing approaches remain largely tailored to notated, score-based repertoires. This study advances a methodology for analyzing voice–guitar interaction in Carlos Paredes's vocal collaborations—an oral-tradition context where compositional and performative layers co-emerge. Using source-separated stems, physics-informed harmonic modeling, and beat-level audio descriptors, we examine melodic, harmonic, and rhythmic relationships across eight recordings with four singers. Our commonality–diversity framework, combining multi-scale correlation analysis with residual-based detection of structural deviations, reveals that expressive coordination is predominantly piece-specific rather than corpus-wide. Diversity events systematically align with formal boundaries and textural shifts, demonstrating that the proposed approach can identify musically salient reorganizations with minimal human annotation. The framework further offers a generalizable computational strategy for repertoires without notated blueprints, extending Music Performance Analysis into oral-tradition and improvisation-inflected practices.

1. INTRODUCTION

In "blueprint"-driven performances, those relying on documented structures such as scores, interpretative decisions primarily involve nuanced changes in timing and dynamics while leaving compositional content intact. Music Performance Analysis (MPA) [1] has traditionally focused on such contexts, comparing multiple performances of a shared blueprint to study expressive deviation and interpretative choice. This approach has naturally favored Western art music, where notated scores and extensive performance corpora enable systematic comparative analysis [1, 2].

Conversely, MPA faces significant challenges when addressing repertoires rooted in oral transmission, popular music, or improvisation, where no explicit blueprint exists, and compositional and performative dimensions are inseparable [3, 4]. In such practices, musical ideas emerge, evolve, and are shaped through performance itself, with recordings often capturing relatively stable realizations of these processes rather than interpretations of a pre-existing score [5].

Within this context, the work of Carlos Paredes provides a particularly revealing case. Widely recognized as a solo composer–performer, Paredes transformed the Portuguese guitar from a primarily accompaniment role into an autonomous musical voice, redefining the expressive potential of Portuguese musical traditions in the 20th century. This aesthetic position emerged from a formative trajectory grounded in collective practice: born into a lineage of Coimbra guitarists, Paredes learned the instrument through oral transmission from his father, Artur Paredes, and began his career as a second guitarist in ensemble contexts, where he already demonstrated a marked tendency to reshape and elaborate accompaniments beyond conventional models [6, 7]. The resulting tension between accompaniment and artistic autonomy underpins both his solo output and his vocal collaborations.

Our work establishes the first steps toward the study of Carlos Paredes's recorded vocal collaborations, some framed in the Portuguese *Canção de Coimbra* tradition, spanning his entire career. Although these collaborations represent only a small fraction of his largely solo discography, they provide a consistent and well-defined setting in which voice and Portuguese guitar form two stable interacting layers. We adopt audio content analysis to address both *performative* and *structural* dimensions, which, despite their somewhat artificial separation, help frame the agentic interplay through which compositional and interpretative processes co-shape these works.

By leveraging audio source separation, we isolate these two layers to enable independent analysis of their melodic, harmonic, and rhythmic characteristics. This approach allows us to explore how these layers interact across different works, revealing patterns of dialogue, contrast, and complementarity that define Paredes's collaborative style. Our analytical framework centers on statistical tools that capture commonalities, through correlations and linear relationships, and diversity, through residual analysis and outlier detection.

Given this perspective, our research addresses the following questions: (i) What performative and structural characteristics distinguish individual pieces, specific collaborations, and the corpus as a whole? (ii) What patterns of

[1] While MPA extends beyond the analysis of musical audio and encompasses multiple dimensions such as visuals and staging, this study focuses on audio analysis.

Copyright: ©2026 Gilberto Bernardes et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License 3.0 Unported, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

voice–guitar interaction, through commonality and diversity analysis, can be identified across Paredes's collaborative recordings, and how do these patterns vary between recordings and collaborators?

This work contributes to MPA in three key ways. First, it introduces an integrated framework that jointly models structural and performative dimensions in repertoires where no notated blueprint exists. Second, the approach combines source separation, physics-informed harmonic descriptors, and multi-scale statistical analysis to reveal interaction patterns that are otherwise obscured in polyphonic recordings. Third, by framing commonality and diversity through both correlations and residual-based deviations, the method identifies not only stable expressive relationships but also structurally meaningful departures that align with perceptually salient musical events.

The remainder of this paper is structured as follows. Section 2 reviews relevant work in MPA, source separation, and structural analysis. Section 3 introduces the corpus of Paredes's vocal collaborations. Section 4 presents the computational pipeline, including source separation, descriptor extraction, and statistical methods. Section 5 reports the assessment of our descriptors. Section 6 details the main analytical findings on melodic, harmonic, and rhythmic characteristics and their interactions. Section 7 provides a musicological interpretation of the results considering the broader context of Paredes's artistic practice. Section 8 concludes with implications for the computational study of oral-tradition–based repertoires and outlines directions for future work.

2. RELATED WORK

MPA has traditionally focused on score-based traditions where interpretative decisions involve nuanced changes in timing and dynamics while preserving notated compositional content [8, 1]. Computational models of expressive performance enable systematic evaluation of performance hypotheses through quantitative analysis [8]. However, these approaches are fundamentally biased toward repertoires with explicit notational blueprints, making them less applicable to musical traditions rooted in oral transmission or improvisational practices [3, 4].

In traditions where composition and performance are inseparable, such as jazz, popular music, and other orally transmitted practices, the continuum from strictly notated to fully improvised music reflects distinct music traditions and social dynamics [5, 9]. In oral musical cultures, improvisation involves the internal process of recalling previously heard music rather than manipulating preexisting notated material [4]. This distinction is particularly relevant for analyzing Carlos Paredes's work within the *Canção de Coimbra* tradition, where compositional ideas emerge and crystallize through performance and recording rather than through predetermined scores [7, 6].

Computational musicology has made substantial progress in automated music analysis through audio content descriptors and machine learning approaches [10]. Music source separation, the task of decomposing recordings into constituent instrumental or vocal components, has seen tremendous advancement with deep learning methods [11]. These techniques divide into model-based approaches that explicitly represent musical structure through repetition or harmonic patterns [12], and data-driven deep learning methods that achieve state-of-the-art performance given sufficient training data [13, 14]. Recent work has applied deep learning frameworks to specific vocal ensemble challenges, including the analysis of choral singing and multi-voice separation [15].

Following source separation, structural analysis techniques identify musical organization through repetition detection and pattern discovery [16, 17]. The automatic extraction of repeating patterns enables music summarization, indexing, and structural segmentation [18, 19]. These methods typically rely on self-similarity matrices computed from acoustic descriptors to detect recurring segments at multiple temporal scales. However, most computational approaches focus on identifying structural commonality through repetition, with less emphasis on explicitly modeling diversity or performative dimensions beyond timing variations [20].

The integration of structural and performative analysis remains underexplored, particularly for repertoires lacking explicit scores. While some studies examine rhythm-pitch interactions in source-separated tracks [21], performance aspects such as dynamics, articulation, and expressive phrasing receive limited attention in computational frameworks. This gap likely reflects both the predominance of structure-focused research questions and the scarcity of datasets with detailed performance annotations [8, 22].

Our study addresses these limitations by applying integrated structural and performative analysis to Carlos Paredes's vocal collaborations, a corpus representing the intersection of composed and improvised elements within the Portuguese guitar tradition. By leveraging audio source separation to isolate vocal and guitar layers, we enable independent melodic, harmonic, and rhythmic characterization while examining their interaction patterns through statistical analysis of correlations and residuals. This approach extends MPA methodology to oral-tradition-based popular music, where the artificial distinction between structural and performative dimensions remains essential for computational tractability while acknowledging their fundamental inseparability in practice.

3. PAREDES'S COLLABORATIONS CORPUS

While Carlos Paredes's discography is predominantly centered on solo guitar works, it includes six collaborations with singers between 1958 and 1983 (a total of 26 tracks, as documented in the complete discography published in [23]). Precisely because of their limited number, these recordings constitute a well-defined and analytically relevant corpus for examining guitar–voice interaction in the *Canção de Coimbra* tradition. The present study intentionally adopts a narrow subset of the corpus, focusing on eight pieces (two songs per collaboration) to establish the methodological framework. The following recordings were selected:

- Augusto Camacho, *Fado de Coimbra* (1958): 'Adeus a Coimbra' and 'Quando os Sinos Dobram';
- Luiz Goes, *Coimbra de Ontem e de Hoje* (1967): 'Balada do Mar' and 'Canção da Infância';

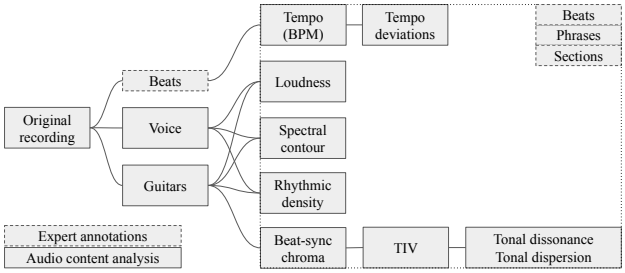


Figure 1. Computational pipeline for the extraction of musical descriptors. Dashed boxes indicate expert-driven manual annotations used to compute specific descriptors and to impose hierarchical segmentations.

- Cecília de Melo, *Meu País* (1970): 'Não Choro por me Deixares' and 'Render dos Heróis';
- Adriano Correia de Oliveira, *Que Nunca Mais* (1975): 'Recado a Helena' and 'Tejo que Levas as Águas'.

This selection features three singers from the *Canção de Coimbra* tradition and one lyrical singer (Melo), whose stylistic background contrasts with the popular and fado-related traditions represented in the rest of the corpus. This focused set serves to demonstrate the pipeline that will be used to analyze the entire collection of collaborations.

The corpus was annotated using Sonic Visualiser [24]. Annotations were initially carried out independently by two musicologists to address subjective bias. A subsequent review stage involving a third musicologist was used to consolidate discrepancies and compile a shared annotation set, which served as the basis for the analyses presented in this study. Four annotation layers were defined: (1) beat and bar-level timing; (2) formal structure, capturing large-scale sectional organization; (3) tonal phrase segmentation, describing melodic contour units; and (4) harmonic progression, including chordal changes and harmonic functions.

4. COMPUTATIONAL METHODS

Figure 1 presents the computational pipeline employed in this work. Our input data comprise the original track (uncompressed audio at 44.1 kHz sample rate) and expert temporal segmentation annotations for beats, phrases, and sections. Prior to audio content analysis, we apply the Ht-Demucs model [25] to separate the vocal and guitar tracks into two distinct audio streams. The guitar track is further decomposed into harmonic and percussive components using the harmonic-percussive source separation (HPSS) algorithm by Fitzgerald [26]. We refer to these separated audio streams as 'guitar harmonic' and 'guitar percussive' components, respectively. Our implementation is available online at: `https://github.com/bernardes7/Paredes`. Unless otherwise specified, the framework relies on version 0.11 of the `librosa` library [27] for audio processing and `SciPy` [28] library for statistical and time series processing.

4.1 Audio Content Analysis

For each dimension under analysis, we extract a time series at beat-level resolution. This resolution was chosen as a suitable compromise between human interpretability and computational efficiency, given that the genre under study exhibits limited nuanced expression below the beat level.

Dynamic descriptors capture the temporal evolution of loudness across the voice and guitar tracks separately, computed as the root mean square energy per frame and converted to decibels (dB) relative to the maximum amplitude to ensure comparability across sources and enhance perceptual representation. These frame-level descriptors are then aggregated at beat positions. Finally, a loudness mask for voice and guitars is computed to identify regions below -40 dB, denoting inactive segments, which are then used to filter out all remaining descriptors.

Rhythmic descriptors include the tempo curve, overall tempo, tempo deviations, and rhythmic density (computed per voice and guitar parts). The tempo curve is obtained by converting the expert inter-beat intervals annotations into beats per minute (BPM). From this curve, we estimate the piece's tempo as the median value within the first section where voice and guitars interact (typically, the second section following an introduction by the guitars). This section is assumed to represent the baseline tempo established by the performers during their initial interaction. Rhythmic density refers to the number of note onsets per beat. For the guitar part, it is computed from a flux-driven novelty detection applied to the guitar percussive component. For the voice part, it is obtained through note transcription based on the pYIN algorithm, followed by post-processing using Hidden Markov Models (HMM), as implemented in the corresponding Vamp plugin [29].

Melodic and harmonic contour descriptors are derived from the voice and harmonic guitar tracks, respectively. We compute these contours as the spectral centroid of the Constant-Q Transform (CQT) representation of each track, providing a systematic inference under the assumption of relatively stable timbre. While this approach mitigates inference errors potentially introduced by adopting advanced pitch-tracking algorithms, only the relative curve shape can be interpreted: the centroid integrates over all spectral energy and thus tracks register rather than fundamental frequency. Inactive beats in both melodic and harmonic curves are excluded by applying the respective dynamic masks.

Remaining **harmonic descriptors**, computed from the guitar's harmonic track, include dissonance and tonal dispersion. The baseline representation is a frame chroma, derived from the guitar's harmonic track, obtained through a pipeline incorporating CQT analysis, denoising, whitening, and Non-Negative Least Squares (NNLS) decomposition using an instrument-specific dictionary tailored to the specificities of the Portuguese and fado guitars found in literature [30, 31, 32]. This process requires parameterizing the NNLS dictionary to reflect the complex physical and acoustic properties inherent to these instruments.

The dictionary incorporates key physical acoustic behaviors. Harmonic amplitudes a_k, where k denotes the partial index, decrease exponentially with harmonic order, controlled by a decay rate factor s [31]. Odd harmonics are selectively scaled by a factor $\gamma > 0$, yielding the modified amplitude a'_k, to model the timbre characteristic of these guitars [31]. Inharmonicity is introduced to account for string stiffness, causing partial frequencies f_k to deviate

slightly from integer multiples of the fundamental f_0 by a small coefficient β [31]. Furthermore, body resonances are modeled as $R = 3$ Gaussian gains $g(f)$, each with amplitude G_ℓ, centered at specific frequencies f_ℓ (at 120, 470, and 1000 Hz, with σ_ℓ widths of 30, 80, and 150 Hz, respectively) to reflect the substantial soundboard and air cavity amplification [31, 32]. S_{res} represents the resonance gain scaling factor [2]. The complete formulation for the final partial amplitude A_k is:

$$
\begin{aligned}
a_k &= s^{(k-1)}, \\
f_k &= f_0 \cdot k \cdot \sqrt{1 + \beta k^2}, \\
a'_k &= \begin{cases} a_k \cdot \gamma, & k \text{ odd}, \\ a_k, & k \text{ even}, \end{cases} \\
g(f) &= S_{res} \cdot \sum_{\ell=1}^{R} G_\ell \cdot \exp\Big(- \frac{(f - f_\ell)^2}{2\sigma_\ell^2} \Big), \\
A_k &= a'_k \cdot g(f_k).
\end{aligned}
\tag{1}
$$

After median beat aggregation, Tonal Interval Vectors (TIV) are computed as the discrete Fourier transform (DFT) of beat-synchronized chromas, whose coefficients are weighted by empirical consonance ratings [33]. This transformation enables higher-level harmonic analysis grounded in perceptual principles and aligns with music theory frameworks using the DFT [34]. From these TIVs, we derive tonal dissonance and tonal dispersion using the `tiv.lib` library [35, 36]. To define a harmonic center, from which the tonal dispersion Euclidean distances are computed, we adopt a TIV vector resulting from the centroid of all beat-synchronous TIV of the section where guitars and voice first interact.

4.2 Statistical Analysis

We quantify commonalities and diversity in performance by analyzing pairwise linear dependencies among the extracted musical descriptors. Because descriptor time series are typically non-stationary and contain multi-scale fluctuations, we first apply empirical mode decomposition (EMD) and focus on slower-varying components by reconstructing each descriptor from IMFs 3 and higher, $x^{(\geq 3)}(t) = \sum_{j \geq 3} \mathrm{IMF}_j(t)$. This reduces the influence of rapid local fluctuations and stabilizes dependence estimates at the macro-structural level.

For each song, we compute the Pearson correlation matrix across descriptors using the reconstructed signals $x^{(\geq 3)}(t)$. To obtain collaboration- and corpus-level summaries, correlations are aggregated across songs using Fisher's z-transform: $z_n = \operatorname{atanh}(r_n), \bar{z} = \frac{1}{N} \sum_n z_n$, and $\bar{r} = \tanh(\bar{z})$, preserving sign and avoiding the bias induced by direct averaging of r. Descriptor pairs are retained as candidate interaction patterns when the aggregated correlation exceeds a practical effect-size threshold (e.g., $|\bar{r}| > 0.6$); in supplementary analyses we verify robustness using bootstrap confidence intervals [3].

To identify moments of diversity relative to an established dependency, we model the within-song relationship between each retained descriptor pair by robust linear regression, $y(t) = \alpha + bx(t) + \varepsilon(t)$, and examine the time-resolved residuals $\varepsilon(t)$. Residuals are standardized using a robust scale estimate (median absolute deviation), and extreme deviations are flagged when $|z(t)| > 2.5$. Positive residual outliers indicate transient exaggeration beyond the typical linear coupling (values above the fitted relation), whereas negative residual outliers indicate transient attenuation or reversal (values below the fitted relation).

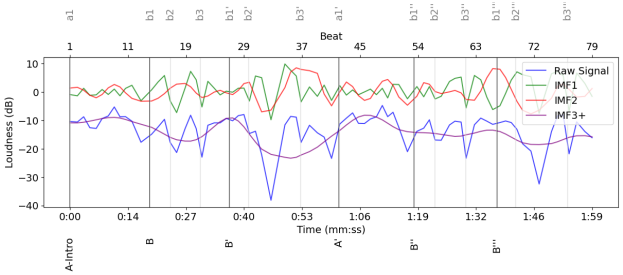


Figure 2. Empirical Mode Decomposition of the guitar loudness descriptor in 'Balada do Mar' by Paredes and Goes, showing the raw signal alongside intrinsic mode functions IMF 1, IMF 2, and IMF 3 and higher.

5. DESCRIPTORS ASSESSMENT

Given that all signal-processing descriptors derive from separated stems, we conducted a qualitative assessment of the Ht-Demucs outputs to ensure that artifacts did not systematically bias analyses. Separated tracks were reviewed by two expert listeners to identify separation failures. Although occasional artifacts were observed, they were predominantly confined to high-energy consonants or rapid ornamental figures, providing sufficient grounds for interpreting descriptor-level correlations as reflections of musical interaction.

The descriptors adopted in this work are well-documented and evaluated in related literature. However, a significant refinement was introduced in the harmonic descriptors. The quality of the beat-synchronous chroma representation was enhanced by incorporating a specialized dictionary specifically adapted to the Portuguese and fado guitars. To validate this adaptation and quantify the efficacy of the physical modeling, a preliminary evaluation was conducted via an ablation study.

This study measured the contribution of various physics-informed components to the NNLS beat-synchronous

[2] The lowest center, 120 Hz, approximates the highly efficient air cavity mode (Helmholtz resonance) experimentally identified around 125 Hz in Coimbra models [32]. The intermediate center, 470 Hz, is crucial for encompassing the major structural resonance group observed in the 250-500 Hz range, specifically covering higher longitudinal dipole modes (e.g., at 406 and 458 Hz) that contribute significantly to the radiated sound spectrum. The highest frequency center, 1 kHz, models the substantial high-frequency energy contributions, utilizing a broad gain function to capture the brilliance and clarity required for the distinct metallic timbre of the instruments.

[3] We evaluated whether the cutoff $|r| \geq 0.6$ was appropriate by estimating 95% bootstrap confidence intervals (moving-block bootstrap, $B = 1000$) for every pairwise correlation across all pieces. No feature pair in the corpus exhibited a confidence interval entirely above 0.6, whereas a substantial majority had confidence intervals entirely below this threshold. The remaining near-threshold cases displayed wide confidence intervals crossing 0.6 and highly variable bootstrap probabilities of exceeding the threshold, indicating that these correlations are not statistically stable. Thus, 0.6 acts as a conservative but empirically justified boundary: correlations below it are reliably weak, while correlations above it would require evidence stronger than what is present in the data.

chroma estimation accuracy, benchmarked against binary chroma representations derived from expert annotations [4]. The ablation analysis systematically investigated the contribution of six harmonic modeling components: frequency whitening (WHI) and median filtering (MED); and dictionary factors including harmonic decay (DEC), s, odd-harmonic boost (ODD), γ, body resonances (RES), and inharmonicity (INH) (β). Following individual component assessment, Bayesian optimization with Gaussian process surrogates fine-tuned three progressive configurations using cosine similarity: (A) WHI+DEC; (B) WHI+DEC+ODD+RES; and (C) the full model (WHI+DEC+ODD+RES+INH+MED).

The analysis showed that WHI and DEC were the most impactful individual components and their combination (A) achieved mean cosine similarity of .739. Progressive optimization revealed that the highest score of .741 was attained by (B), slightly outperforming (C), the full model scoring .738. This suggests the addition of INH and MED slightly diminished performance. Optimal parameters for (B) included zero spectral detuning, maximum decay strength ($s = .5$), an odd harmonic scaling factor of $\gamma = .95$, and maximized resonance gain scale of 1.4. These results underscore the critical importance of modeling strong harmonic decay and utilizing prominent body resonance frequencies (120.0 Hz, 470.0 Hz, 1000.0 Hz) to capture the instrument's distinct characteristics.

6. RESULTS

While the complete set of results is available online in the complementary materials to this paper [5], we focus here on the interpretation of correlations in 'Balada do Mar', a collaboration between Paredes and Goes, shown in Figure 3, as an illustrative example of our analytical workflow.

Two descriptor relationships of particular interest are the correlation between tonal dissonance and guitar rhythmic density ($r = .81$) and between guitar loudness and voice rhythmic density ($r = -.71$). Figure 4 shows these two time series overlaid to facilitate inspection of their correlation in musical time. As indicated by the negative correlation, voice melodic onsets exhibit contrapuntal motion (mirrored behavior) with guitar loudness. An exception to this correlation is observed at the beginning of section B, where both parameters exhibit parallel motion from high to low density and loudness, as shown in the positive outlier indicated in the residual analysis (see Figure 5). Posthoc perceptual inspection confirmed these findings, with the dense guitar ostinato audibly filling the space during sustained or silent vocal passages, while the outlier at section B corresponds to a rare moment of simultaneous articulation in both layers.

Figure 6 presents the aggregated matrices per collaboration using Fisher's z-transform. Examining the matrices at the level of individual collaborations reveals a set of interactional signatures that are both internally consistent and context-dependent. In the collaboration with Camacho,

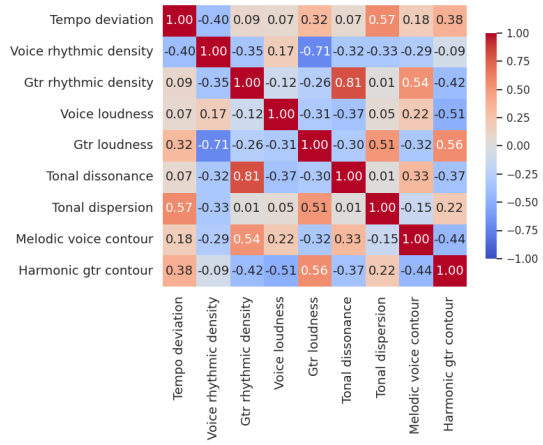


Figure 3. Correlation matrix for the song 'Balada do Mar' (Paredes and Goes), aggregated using Fisher's z-transformation of Pearson correlations.

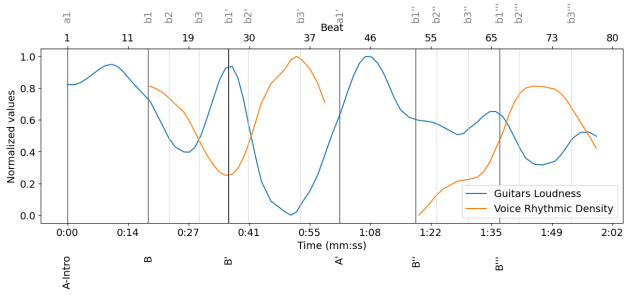


Figure 4. Time series of the Empirical Mode Decomposition (IMF 3+) for guitar loudness and voice rhythmic density in 'Balada do Mar' by Paredes and Goes. Masked segments correspond to periods of vocal inactivity below -40 dB. Vertical black lines denote section boundaries, while vertical grey lines denote phrase boundaries. Section labels are displayed at the bottom of the plot, and phrase labels are displayed at the top.

the strongest correlations involve the guitar layer (guitar rhythmic density-guitar loudness, $r = .75$; guitar rhythmic density-harmonic guitar contour, $r = .5$), alongside the moderate correlation between voice rhythmic density and voice loudness ($r = .5$). The Goes collaboration shows correlations involving cross-layer and harmonic descriptors: between voice rhythmic density and guitar loudness ($r = -.6$), tonal dissonance and harmonic guitar contour ($r = -.65$), and guitar loudness and harmonic guitar contour ($r = .5$). In collaboration with Correia de Oliveira, expressive correlations are restricted to the vocal layer (voice rhythmic density-voice loudness, $r = .59$; voice melodic contour-voice loudness, $r = .6$). The Melo collaboration shows a negative correlation between guitar rhythmic density and tempo deviation ($r = -.51$) and a positive correlation between guitar rhythmic density and voice rhythmic density ($r = .56$). Taken together, these results show that the strongest linear dependencies differ across collaborations both in magnitude and in the descriptor domain.

When aggregating all songs across collaborations into a single global correlation matrix (Figure 7), no dominant interaction patterns emerge (correlations remain within the interval $[-.43, .51]$). The absence of strong, consistent correlations at this level suggests that there is no single, corpus-wide marker governing Paredes's vocal collaborations. Instead, interaction strategies appear to be adapted

[4] A perfect alignment with the binary chroma is not expected due to the presence of additional spectral components inherent to instrumental performance, including overtones, transients, and noise-related partials.

[5] The correlation matrix per piece is made available at `https://github.com/bernardes7/Paredes`.

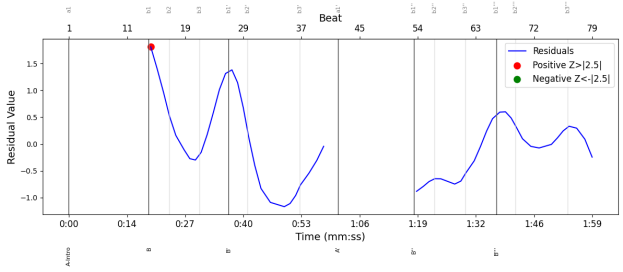


Figure 5. Residual analysis of the guitar loudness and voice rhythmic density interaction in 'Balada do Mar' by Paredes and Goes. The highlighted point identifies a positive outlier with a residual z-score exceeding $+2.5$, indicating a breakdown of the typical contrapuntal relationship at the beginning of section B. Vertical black lines denote section boundaries, while vertical grey lines denote phrase boundaries. Section labels are displayed at the bottom of the plot, and phrase labels are displayed at the top.

to the specific combination of singer, repertoire, and piece-level musical structure.

The strongest correlation that remains perceptible at the global level concerns the relationship between vocal melodic contour and vocal loudness ($r = .51$), where higher pitch regions tend to coincide with increased intensity. Although this pattern reflects a general tendency for vocal performance rather than a guitar–voice dependency, its persistence across collaborations provides a reference point against which more variable guitar–voice interactions can be interpreted.

7. PRELIMINARY MUSICOLOGICAL INSIGHTS

The degree of variation observed across the corpus is consistent with expectations for this repertoire. It is unsurprising that only a single trait, correlation between vocal melodic contour and vocal loudness, remains perceptible at a global level. Although both Carlos Paredes and the collaborating singers display recognizable timbral and expressive characteristics, each recorded song functions as a self-contained musical miniature, shaped by its own poetic, melodic, and formal specificities, with expressive traits articulated idiosyncratically at the song level, and sometimes even diverging within collaborations involving the same singer.

This is illustrated in the recordings with Luiz Goes. Both 'Balada do Mar' and 'Canção da Infância' employ ternary subdivision (12/8 and 6/8, respectively). In 'Balada do Mar', Paredes constructs a chromatic pitch-descending ostinato in sixteenth notes, whereas in 'Canção da Infância' he accompanies Goes with a more conventional arpeggiated pattern in eighth notes, closely aligned with the harmonic framework. While Goes's vocal interpretation remains broadly consistent across the two recordings, Paredes adopts contrasting accompanimental approaches: the former foregrounds a continuously active, textural, and unsettled guitar gesture, while the latter reflects a more traditional and harmonically stable accompanimental role. In the analyzed descriptors, this divergence is reflected in the relationship between vocal melodic contour and guitar rhythmic density, which is positive in 'Balada do Mar' and negative in 'Canção da Infância', indicating contrasting configurations of melodic motion and guitar activity within the same collaboration.

At the level of individual collaborations, additional patterns emerge. In both the Camacho and Correia de Oliveira collaborations, the strongest correlations are concentrated within individual layers rather than across them, suggesting that expressive organization is primarily articulated internally within each part. These results suggest that while vocal approaches remain relatively stable within collaborations, Paredes's guitar language is not applied uniformly but selectively reconfigured in response to local structural and performative contexts. By contrast, the Goes collaboration exhibits a negative correlation between voice rhythmic density and guitar loudness, pointing to a more dynamic redistribution of expressive weight between layers, while the Melo recordings show alignment between guitar and voice rhythmic density, suggesting coordination centered on shared melodic–rhythmic intentions. Overall, while singers tend to maintain a stable expressive profile within a collaboration, Paredes's guitar adapts flexibly to the demands of individual pieces, consistent with his broader artistic identity as a versatile musician whose work engaged with theater, dance, and cinema, as well as with performers from diverse musical backgrounds [7].

Consequently, no single interactional signature can be attributed to the *Canção de Coimbra* singers as a group, a result that is musically coherent given the stylistic diversity and limited size of the analyzed repertoire. This contrasts with score-based singer–accompanist traditions, where interactional hierarchies are largely stabilized by notation, and aligns instead with accounts of orally transmitted and improvisation-informed practices, in which musical structure and coordination emerge through performance and remain sensitive to piece-specific demands [3, 9, 4].

In the diversity analysis, only sparse residual outliers were detected (approximately two to three per piece). These outliers correspond to moments of substantial reorganization within individual songs, including the introduction of new harmonic changes or musical material (e.g., transitional instrumental sections, repeated ostinatos, or overlapping layers that occur only once), the most pronounced accelerandos and ritardandos relative to comparable passages, entrances of the voice, and fade-outs. Outliers were most frequently observed at the beginnings and endings of songs; when they occurred mid-piece, they consistently aligned with manually annotated formal section boundaries. This behavior is particularly evident in 'Render dos Heróis' (Melo collaboration), a piece characterized by a succession of short, contrasting sections reflecting its original conception as theater music. Here, outliers capture several structurally salient events, including harmonic shifts manifested through changes in the relationship between tonal dissonance and vocal rhythmic density, the introduction of a novel instrumental passage between vocal sections, the only segment in which voice and guitar articulate the same melodic–rhythmic profile in unison (with additional harmonic elaboration in the guitar), and the final ritardando. These findings indicate that residual-based diversity measures are sensitive to moments of significant structural and performative transformation rather than to local expressive fluctuation.

Beyond this corpus, the proposed framework is in principle transferable to other repertoires where composition and

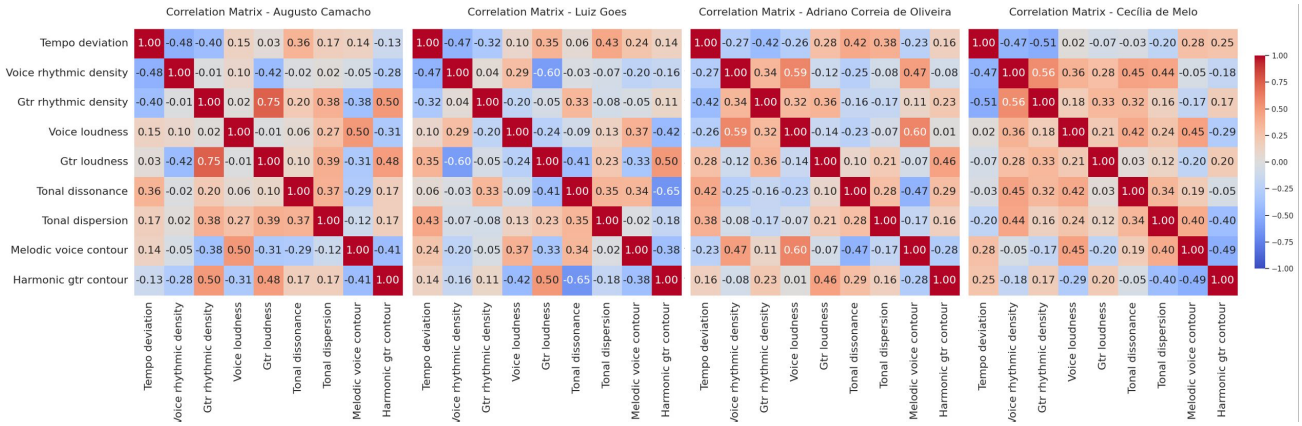


Figure 6. Correlation matrices for each Paredes vocal collaboration, aggregated using Fisher's z-transformation of Pearson correlations.

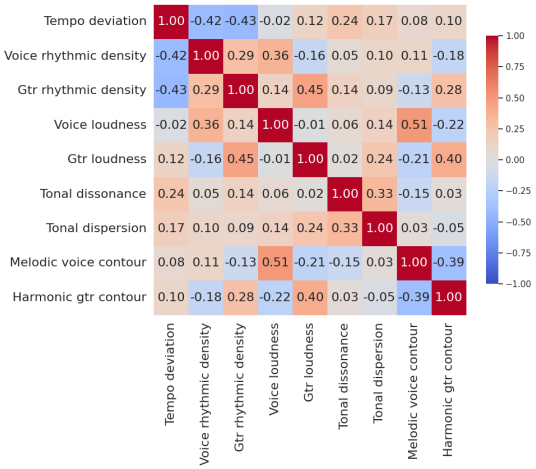


Figure 7. Correlation matrix for all Paredes vocal collaboration adopted in our study, aggregated using Fisher's z-transformation of Pearson correlations.

performance are inseparable. The core requirements are modest: multi-layer recordings amenable to source separation, and a temporal segmentation that need not be score-derived, beat tracking or automatic structural segmentation would suffice in many cases. This positions the approach as applicable to traditions such as jazz, fado, or flamenco, where analogous voice–instrument dialogues unfold without notated blueprints, and where the commonality–diversity dichotomy maps naturally onto the contrast between idiomatic conventions and piece-specific expressive choices. That said, two practical boundaries currently condition its applicability: source separation quality sets a ceiling on descriptor reliability in texturally dense or ornamentally rich repertoires, and the dependence on expert temporal annotations, while essential for interpretive grounding, limits scalability to corpora where such annotation is feasible.

8. CONCLUSIONS AND FUTURE WORK

This paper introduced a computational framework combining source separation, beat-level descriptors, and residual-based diversity detection to analyze voice–guitar interaction in eight of Carlos Paredes's vocal collaborations. Expressive organization proved largely piece-specific, with diversity events aligning consistently with formal boundaries. Future work should expand the descriptor set with timbral features suited to the Portuguese guitar, explore automatic substitutes for expert annotation, and extend the corpus to trace broader stylistic trajectories across Paredes's career.

9. REFERENCES

[1] C. Palmer, "Music Performance," *Annual Review of Psychology*, vol. 48, no. 1, pp. 115–138, 1997.

[2] N. Moura, M. Vidal, A. M. Aguilera, J. P. Vilas-Boas, S. Serra, and M. Leman, "Knee flexion of saxophone players anticipates tonal context of music," *npj Science of Learning*, vol. 8, no. 1, 2023.

[3] K. E. Prouty, "Orality, Literacy, and Mediating Musical Experience: Rethinking Oral Tradition in the Learning of Jazz Improvisation," *Popular Music and Society*, vol. 29, no. 3, pp. 317–334, 2006.

[4] M. Tani, *Traditional Iranian Music: Orality, Physicality and Improvisation*. Trans Pacific Press, 2024.

[5] S. Rosenblum, C. Schaffrath, P.-J. Maes, and M. Leman, "The Varying Social Dynamics in Orally Transmitted and Notated vs. Improvised Musical Performance," *Front. in Psychology*, vol. 14, p. 1106092, 2023.

[6] A. R. Correia, *Do Choupal até à Lapa: Para uma Etnografia do Constructo da Canção de Coimbra*. Lisboa: Edições Sílabo, 2016.

[7] O. Fonseca, *Carlos Paredes: A Guitarra de um Povo*. Portugal: Tradisom, 2025.

[8] C. E. Cancino-Chacón, M. Grachten, W. Goebl, and G. Widmer, "Computational Models of Expressive Music Performance: A Comprehensive and Critical Review," *Front. in Digital Humanities*, vol. 5, p. 25, 2018.

[9] A.-M. Christodoulou and O. Lartillot, "A multimodal dataset of Greek folk music," in *Proceedings of the 12th Intl. Conf. on Digital Libraries for Musicology*, 2025, pp. 19–27.

[10] D. Meredith, *Computational Music Analysis*. Cham, Switzerland: Springer, 2016.

[11] Y. Chen, S. Zhang, X. Wang, T. Zhao, and W. Xiang, "An In-depth Review on Music Source Separation," *arXiv:2401.11378*, 2024.

[12] K. Schulze-Forster, G. Richard, C. S. Doire, and R. Badeau, "Unsupervised Music Source Separation Using Differentiable Parametric Source Models," *IEEE/ACM Transactions on Audio, Speech, and Language Proces.*, vol. 31, pp. 1276–1289, 2022.

[13] A. Défossez, "Hybrid Spectrogram and Waveform Source Separation," in *Proc. of the ISMIR 2021 Workshop on Music Source Separation*, 2021.

[14] F.-R. Stöter, S. Uhlich, A. Liutkus, and Y. Mitsufuji, "Open-Unmix-A Reference Implementation for Music Source Separation," *J. of Open Source Software*, vol. 4, no. 41, p. 1667, 2019.

[15] P. Chandna, H. Cuesta, D. Petermann, and E. Gómez, "A Deep-Learning Based Framework for Source Separation, Analysis, and Synthesis of Choral Ensembles," *Front. in Signal Proces.*, vol. 2, p. 808594, 2022.

[16] R. B. Dannenberg and N. Hu, "Pattern Discovery Techniques for Music Audio," in *Proc. of the ISMIR*, 2002, pp. 63–70.

[17] J. Paulus and A. Klapuri, "Music Structure Analysis Using the Patterns," *IEEE Transactions on Audio, Speech, and Language Proces.*, vol. 18, no. 3, pp. 451–462, 2010.

[18] L. Lu, L. Wenyin, and H.-J. Zhang, "Repeating Pattern Discovery and Structure Analysis from Acoustic Music Data," in *Proc. of the 6th ACM SIGMM Intl. Workshop on Multimedia Information Retrieval*, 2004, pp. 275–282.

[19] M. Müller, P. Grosche, and N. Jiang, *Handbook of Signal Proces. in Acoustics*. Springer, 2015, ch. Music Structure Analysis from Acoustic Signals, pp. 305–331.

[20] A. Flexer, E. Pampalk, and G. Widmer, "A Case for Predominance of Novelty-Based Segmentation for Music Structure Analysis," *Music Information Retrieval Exchange (MIREX)*, 2012.

[21] C. Dittmar and D. Gartner, "Audio-Based Drum Transcription using Dynamical Models of Percussive Sounds," in *Proc. of the Sound and Music Computing Conf.*, 2012.

[22] A.-M. Christodoulou, O. Lartillot, and A. R. Jensenius, "Multimodal music datasets? Challenges and future goals in music processing," *Intl. J. of Multimedia Information Retrieval*, vol. 13, 2024.

[23] C. Paredes, "O Mundo Segundo Carlos Paredes: Integral 1958–1993," 8-CD compilation, Portugal, 2002, catalogue number: EMI 5 80304 2.

[24] C. Cannam, C. Landone, and M. Sandler, "Sonic Visualiser: An Open Source Application for Viewing, Analysing, and Annotating Music Audio Files," in *Proc. of the ACM Multimedia 2010 Intl. Conf.*, 2010, pp. 1467–1468.

[25] S. Rouard, F. Massa, and A. Défossez, "Hybrid transformers for music source separation," in *IEEE ICASSP*. IEEE, 2023, pp. 1–5.

[26] D. Fitzgerald, "Harmonic/percussive separation using median filtering," in *Proc. of the Digital Audio Effects Conf.*, vol. 13, 2010, pp. 1–4.

[27] B. McFee *et al.*, "librosa: Audio and music signal analysis in python." *SciPy*, vol. 2015, pp. 18–24, 2015.

[28] P. Virtanen, R. Gommers, T. E. Oliphant *et al.*, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020.

[29] M. Mauch and S. Dixon, "pYIN: A fundamental frequency estimator using probabilistic threshold distributions," in *IEEE ICASSP*, 2014, pp. 659–663.

[30] O. Inácio, F. Santiago, and P. Cabral, "The Portuguese Guitar Acoustics: Part 1–Vibroacoustic Measurements," in *Proc. of the IV Iberoamerican Acoustics Congress*, 2004.

[31] V. Debut, J. Antunes, M. Marques, and M. Carvalho, "Physics-based modeling techniques of a twelve-string Portuguese guitar: A non-linear time-domain computational approach for the multiple-strings/bridge/soundboard coupled dynamics," *Applied Acoustics*, vol. 108, pp. 3–18, 2016.

[32] P. Serrão, A. Silva, V. Infante, and A. Ribeiro, "Experimental-numerical correlation of the dynamic behavior of the Portuguese guitar," *Applied Acoustics*, vol. 116, pp. 136–148, 2017.

[33] G. Bernardes, D. Cocharro, M. Caetano, C. Guedes, and M. E. Davies, "A multi-level tonal interval space for modelling pitch relatedness and musical consonance," *J. of New Music Research*, vol. 45, no. 4, pp. 281–294, 2016.

[34] E. Amiot, *Music Through Fourier Space: Discrete Fourier Transform in Music Theory*, ser. Computational Music Science. Cham, Switzerland: Springer, 2016.

[35] A. Ramires, G. Bernardes, M. E. Davies, and X. Serra, "TIV. LIB: An open-source library for the tonal description of musical audio," in *Proc. of the Digital Audio Effects Conf.*, 2020, pp. 304–309.

[36] G. Bernardes, M. E. Davies, and C. Guedes, "Automatic musical key estimation with adaptive mode bias," in *IEEE ICASSP*, 2017, pp. 316–320.

ChoraleWind: An Expressive Wind-Quartet Dataset for End-to-End Rendering from the Neues Thüringer Choralbuch

Axel Berndt[1] **Aida Amiryan-Stein**[1] **Manuel Peters**[2] **Meinard Müller**[2] **Stefan Balke**[2]
[1]Paderborn University [2] International Audio Laboratories Erlangen
axel.berndt@uni-paderborn.de

ABSTRACT

We introduce ChoraleWind, a dataset along with a framework for a reproducible end-to-end rendering from the Neues Thüringer Choralbuch (NTCB). The dataset comprises 311 four-part chorales and covers the full pipeline from symbolic score encoding to performance-level rendition and synthesized audio. ChoraleWind includes a rule-based performance model that generates expressive timing, dynamics, and articulation, including metric and structural accents as well as phrase-end gestures from high-quality MEI encoding of the NTCB chorales, combined with a wind-instrument synthesis based on physical modeling that produces isolated stems and ensemble mixes. The dataset provides aligned symbolic representations, performance annotations, and multitrack audio, enabling systematic training and evaluation of score-to-audio wind-quartet rendering methods under fully controlled conditions. Rather than aiming at state-of-the-art purely data-driven synthesis, ChoraleWind is designed as a transparent and reproducible testbed for studying expressive performance generation, timbre modeling, and evaluation of wind-quartet rendering systems.

1. INTRODUCTION

Recent years have seen growing interest in computational methods for modeling expressive performance and realistic timbre for wind instruments. In contrast to piano music, however, research on wind music is hindered by the scarcity of openly available, systematically annotated datasets. Existing corpora are often limited in size, heterogeneous in instrumentation, or recorded with substantial cross-talk, which makes them difficult to use as training material for data-driven models and challenging to evaluate in a controlled and reproducible manner.

Previous work such as ChoraleBricks [1] has demonstrated the value of curated wind-ensemble recordings aligned with symbolic representations. Producing such datasets, however, is labor-intensive, costly, and difficult to scale, as it requires professional performers, studio environments, and detailed alignment procedures. Synthetic audio, while not matching the acoustic richness of real performances, offers an attractive complementary approach: it enables controlled experiments, supports self-supervised and contrastive learning setups, and allows large-scale training without recording constraints. At the same time, synthetic datasets can serve as transparent and reproducible baselines for studying expressive performance modeling and score-to-audio rendering.

Copyright: ©2026 Axel Berndt et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License 3.0 Unported, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

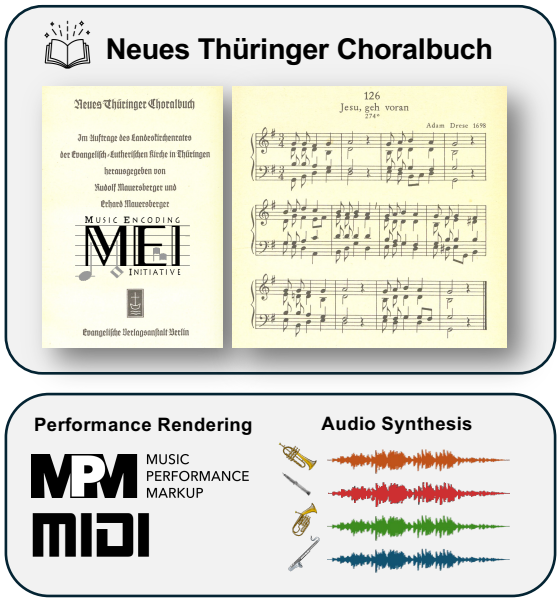


Figure 1. Illustration of the ChoraleWind dataset. Based on high-quality MEI encodings of the Neues Thüringer Choralbuch, we generate expressive renditions using a rule-based performance model. These renditions are then synthesized as multitrack audio for a variety of wind instruments using a synthesizer based on physical modeling.

In this paper, we introduce *ChoraleWind*, an expressive wind-quartet dataset and reproducible end-to-end rendering pipeline based on the "Neues Thüringer Choralbuch" (NTCB). ChoraleWind comprises 311 four-part chorales (**S**oprano, **A**lto, **T**enor, **B**ass) derived from high-quality MEI encodings of the NTCB chorales compiled by Rudolf and Erhard Mauersberger [2, 3]. As illustrated in Figure 1, the dataset covers the full processing chain from symbolic score encoding to expressive performance modeling and synthesized audio.

Expressive timing, dynamics, and articulation are generated using a rule-based performance model that incorporates domain knowledge about chorale performance, including metric and structural accentuation, phrase-end gestures, and controlled humanization. The resulting performance descriptions are rendered into expressive MIDI and synthesized part-wise into multitrack audio using a commercial physical-modeling wind-instrument synthesizer, producing isolated stems for every part as well as

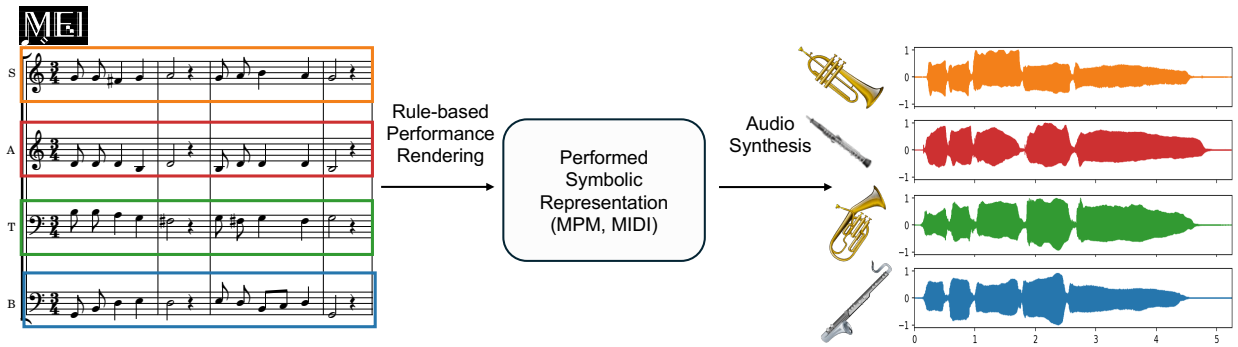


Figure 2. Overview of the rendering pipeline. Starting from the MEI encodings of the SATB chorales, MIDI data is extracted and passed to the performance model. The resulting performance MIDI is then synthesized into multitrack audio recordings.

a large variety of ensemble mixes for 16 different wind instruments. Alongside the audio, ChoraleWind provides aligned symbolic representations and performance-level annotations, enabling systematic training and evaluation of score-to-audio wind-quartet rendering methods under fully controlled and reproducible conditions.

Furthermore, ChoraleWind serves as a superset of ChoraleBricks [1], which contains real recordings of ten selected chorales from the same source with thirteen different instruments and human performers. This connection enables direct and systematic comparisons between synthetic and real performances. Rather than aiming at state-of-the-art purely data-driven synthesis, ChoraleWind is designed as a transparent baseline and controlled testbed for studying expressive performance generation, timbre modeling, and reproducible evaluation of wind-quartet rendering systems.

2. RELATED WORK

Only a limited number of datasets including wind music have been released to date. The most prominent examples are Bach10 and URMP. The Bach10 dataset [4] comprises recordings of ten Bach chorales performed on violin, clarinet, saxophone, and bassoon. The URMP dataset [5] contains 44 pieces spanning a variety of instruments and provides approximately 1.2 hours of isolated multitrack audio. A more recent resource is ChoraleBricks [1], which includes ten chorales with isolated recordings of thirteen instruments, enabling different instrument combinations and ensemble permutations. For datasets involving full orchestra recordings, notable examples include the Beethoven Anechoic dataset [6], the Aalto Anechoic recordings [7], Operation Beethoven [1], and the Spheres dataset [8]. Further symbolic encodings for chorales can be found in the "Chorale Corpus" [9].

An emerging trend, also observed in recent datasets such as EnsembleSet [10], SynthSOD [11], and CocoChorales [12], is the use of synthesizers to generate large amounts of data. While this approach enables the creation of substantial datasets at relatively low cost, the resulting variability in timbre, articulation, and modulation still falls short of what can be captured in real recordings. Nevertheless, Wu et al. [12] demonstrated that synthetically generated four-part ensemble data can improve performance in downstream tasks such as music transcription and music source separation, particularly for instruments with limited data availability.

With respect to performance modeling, two main approaches can be distinguished: rule-based systems and data-driven systems. Rule-based systems represent the more "classical" approach. In these systems, core performance parameters such as tempo and dynamics are modeled using explicit rules derived from the underlying score, for example by identifying phrase boundaries and shaping tempo and velocity curves accordingly. An overview of such systems up to 2013 is provided in [13]. In addition to the rule-based systems, data-driven approaches have emerged as an alternative to rule-based systems. For an overview, we refer the reader to [14, 15]. While these approaches give up much of the explicit control offered by rule-based models, they open up new opportunities for analyzing existing performances and generating novel renditions from symbolic scores. However, given the focus on the chorale scenario and the limited availability of suitable training data from real performances, we adopt a rule-based approach similar to the system proposed in [16], aiming at explicit and reproducible performance renderings.

3. SYMBOLIC REPRESENTATION

Figure 2 provides an overview of the ChoraleWind rendering pipeline, which converts symbolically-encoded four-part chorales into expressive wind-quartet performances using a structured rule set tailored to wind-instrument idioms. The pipeline comprises symbolic preprocessing, rule-based expressive modeling, and audio synthesis. In the following, we start by introducing the Music Encoding Initiative (MEI) format and the applied preprocessing steps.

3.1 MEI Encoding

The musical scores for our dataset have been sourced from the critical digital music edition "Neues Thüringer Choralbuch digital". [2] The "Neues Thüringer Choralbuch" (English: "New Thuringian Chorale Book") [3] is a collection

[1] https://openmusic.academy/revs/JwXCJwcEyAXsJDNsU7DtN3, last access: Dec. 2025.

[2] https://github.com/axelberndt/Neues-Thueringer-Choralbuch-digital , last access: Dec. 2025.

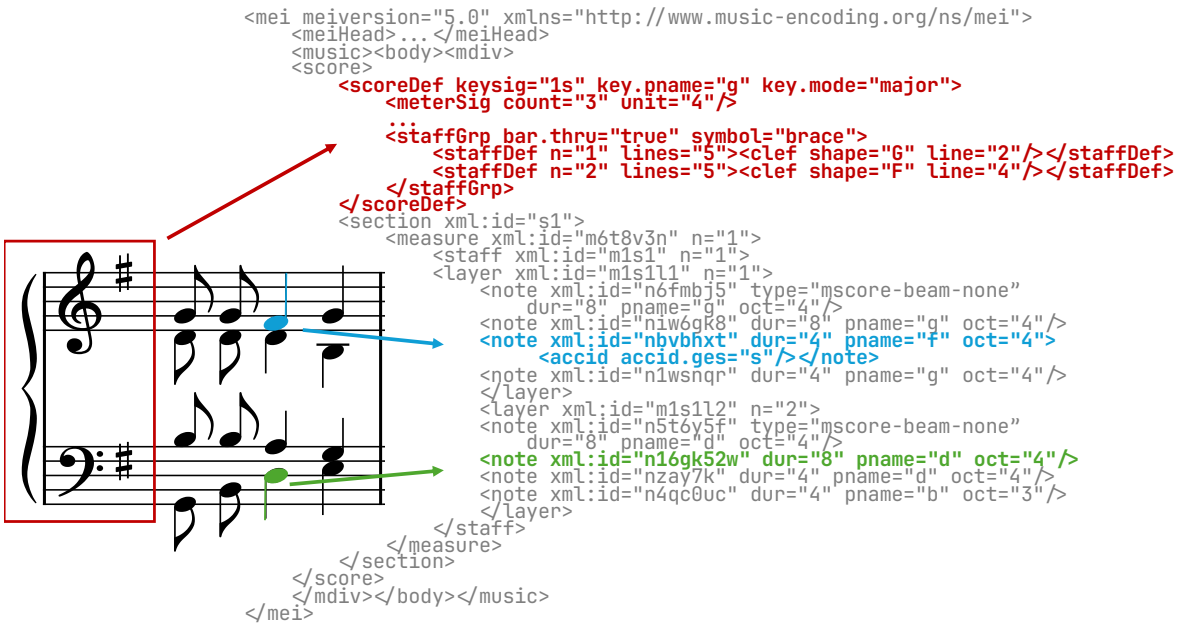


Figure 3. First measure of the chorale "Jesu, geh voran" by Adam Drese and its corresponding MEI encoding. The figure illustrates how score-level information (key, meter, staves) and note-level attributes are represented in MEI.

of more than 300 four-part harmonizations that accompanied the two Protestant chorale books "Deutsches Evangelisches Gesangbuch" and "Evangelisches Kirchengesangbuch". Although these chorale books were succeeded by the current "Evangelisches Gesangbuch" in 1996 and are therefore considered outdated, the chorales of the "Neues Thüringer Choralbuch" remain relevant and in active use due to frequent reprints, such as [17, 18, 19, 20]. Apart from a few exceptions, the vast majority of the harmonizations follow a traditional four-part homophonic cantional style, in which alto, tenor, and bass share the rhythm of the melody, while the "cantus firmus" is consistently placed in the soprano part.

Figure 3 depicts a rendering of the first measure of the chorale "Jesu, geh voran", along with its MEI encoding. Following the original print layout, the four musical parts are encoded across two staves. The upper staff contains tlo he soprano and alto parts as separate layers under a treble clef, while the lower staff similarly encodes tenor and bass under a bass clef. Our MEI encodings follow the international standard (version 5.0 of the "mei-all" schema definition) for digital music editions [2].

Several properties of the MEI encodings are relevant for the subsequent preprocessing steps and are therefore discussed here. The `<meiHead>` section contains detailed metadata, including the (German) song title, book number, composer of the melody, composer of the harmonization, encoder, and license information. The `<music>` section defines one `<mdiv>` (musical division, for example used to separate movements in a symphony) per chorale. Each `<measure>`, `<staff>`, `<layer>`, `<note>`, `<rest>`, and performance instruction carries an `@xml:id` attribute and is therefore addressable.

Tonality and time signature are defined globally at the `<scoreDef>` level (Figure 3, red section), rather than individually per `<staffDef>` as in some other music editions. While tonality remains constant throughout a chorale, time signatures may change—sometimes on a per-measure basis and often without a printed indication. These changes are made explicit in the encoding by inserting a `<scoreDef>` element with an invisible `<meterSig>` between the corresponding measures. Pickup measures are marked with `@metcon="false"` which temporally deactivates metrical conformance. Some chorales, however, do not specify an underlying time signature at all.

Repetitions are encoded using standard repetition bar lines (𝄆 and 𝄇). If the initial section of a chorale is repeated, the opening repetition mark 𝄆 is not printed and therefore not encoded. More complex repetition schemes, such as sections repeated multiple times, are represented using the MEI `<expansion>` mechanism.

3.2 Symbolic Preprocessing

Some preprocessing steps are applied to prepare the symbolic music data for performance generation. For this purpose, we utilize and extend the functionality of the *meico* MEI converter framework [21].

Part separation: Since each musical part is treated individually later on, the two staves are separated into individual parts. Four new staff definitions `<staffDef>` replace the two existing ones in the MEI encoding. Accordingly, each `<measure>` receives new `<staff>` elements, to which the individual `<layer>` elements are reassigned. To remain consistent with the print and common practice, the staff order follows the part order, i. e., (top to bottom) soprano, alto, tenor, bass. As a result, MEI-to-MIDI conversion generates one MIDI track per part in this order.

Addition of IDs: To ensure that all relevant musical elements are addressable, the MEI encoding is checked and missing `@xml:id` attributes are added.

Caesura to breath: The original print, and thus the MEI encoding, use the caesura symbol (') to indicate phrase boundaries. A caesura denotes a "break, pause, or interrup-

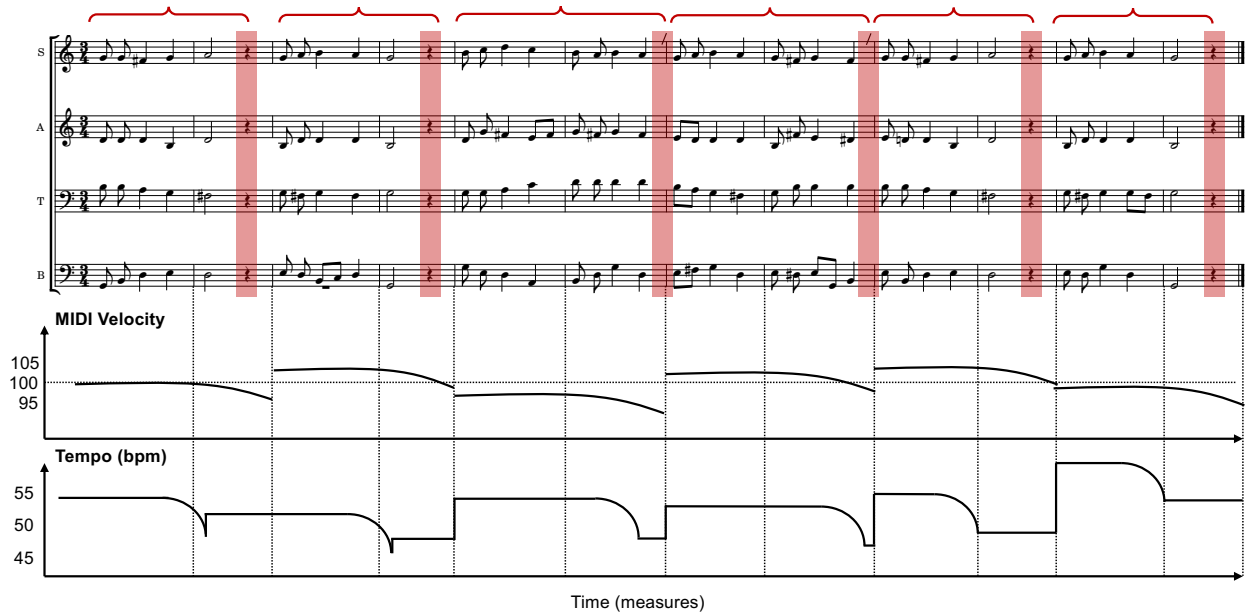


Figure 4. Phrasing mechanism of the performance model illustrated based on the chorale "Jesu, geh voran" by Adam Drese and its harmonization by Rudolf Mauersberger. The top panel shows the four parts (SATB) in individual staff notation, where red arcs indicate the detected phrases and red boxes highlight general rests and caesuras across all parts. The lower panels displays the resulting tempo and velocity curves generated by the performance model.

tion in the normal tempo of a composition"[3] and is commonly interpreted as a breath mark. While tempo modulation at phrase endings is handled by performance modeling, caesuras do not encode the articulatory modifications needed to create a short breathing break. Therefore, all `<caesura>` elements are renamed to `<breath>`.

Breath insertions: Caesura or breath marks are typically omitted where rests in the melody provide a natural breathing opportunity, which is unproblematic. However, they are also missing immediately before repetition end marks (:‖). As these positions always correspond to phrase endings in the dataset but would otherwise go undetected, `<breath>` elements are inserted at the end of the corresponding measures.

Score expansion: To ensure that repetitions are rendered in performance, the score is expanded into a through-composed form. Repeated sections are copied and inserted into the musical sequence, with all copied elements receiving unique IDs and all ID references updated accordingly.

Fix fractured measures: Repetition marks are encoded as bar lines (e. g., `<measure n="5" right="rptend">`) but are often placed within a measure. This results in fractured measures: a shortened measure before the repetition mark and a remaining segment after it or at the start of the repeated section. Performance modeling would interpret these as time signature changes and apply inappropriate metrical accentuation patterns, although the logical time signature remains unchanged. To prevent this, we suppress all subsequent time signature changes that sum to their preceding time signature.

In the ChoraleWind dataset, we provide several files that represent both the score and its rendered performance. These include the original MEI file, which contains two systems (soprano and alto in the first system, tenor and bass in the second), as well as a preprocessed MEI version in which each part is placed in a separate system. In addition, we provide an MPM file encoding the performance parameters, an MSM file that serves as an intermediate representation for exporting MEI to MIDI, and, finally, a performed MIDI file.

[3] See MEI Guidelines, `https://music-encoding.org/guidelines/v5/elements/caesura.html`, last access: Dec. 2025.

4. PERFORMANCE RENDERING

Expressive performance parameters such as timing, dynamics, and articulation, are generated using curated rules that emulate typical wind-instrument ensemble practice. Performance decisions are based on musical properties and modeled in the Music Performance Markup (MPM) format [22]. The MPM API, part of the *meico* framework, renders these into expressive MIDI sequences. This process also produces augmented MEI encodings, adding each note's MIDI velocity (`@vel`) and onset/offset times in milliseconds (`@tstamp.real`, `@tstamp2.real`). Figure 4 shows the sheet music and the resulting velocity and tempo curves for the chorale "Jesu, geh voran". In the following, we describe the underlying rules in detail.

Articulations: During conversion, all articulation instructions from the MEI score are placed into part-specific MPM `<articulationMap>`s and then merged into a global map. Some articulations, particularly breath instructions (former caesuras; cf. Section 3.2), apply to all parts. This ensures all parts breathe together, not just the soprano. The MPM API provides default definitions for standard articulations, which are slightly adjusted, e. g., the breath articulation shortens note duration by 200 ms to allow a break. Text-based articulations are supported but not used here due to the absence of lyrics in most sources.

Metrical accentuation: We implemented standard ac-

#	Instrument	Abbr.	Parts	Dur. (hh:mm:ss)
1	Flute	fl	S	03:10:31
2	Oboe	ob	S	03:10:31
3	Soprano Saxophone	ss	S	03:10:31
4	Alto Saxophone	as	S, A	06:21:02
5	Clarinet	cl	S, A	06:21:02
6	Flugelhorn	fh	S, A	06:21:02
7	Trumpet	tp	S, A	06:21:02
8	English Horn	eh	A	03:10:31
9	Tenor Saxophone	ts	A, T	06:21:02
10	Baritone	bar	T, B	06:21:02
11	Bass Clarinet	bcl	T, B	06:21:02
12	Baritone Saxophone	bs	T, B	06:21:02
13	Bassoon	bsn	T, B	06:21:02
14	French Horn	fho	T, B	06:21:02
15	Trombone	tb	T, B	06:21:02
16	Tuba	tba	B	03:10:31
		$\sum$	27	85:44:14

Table 1. Overview of the synthesized instruments in ChoraleWind, including abbreviations, the SATB parts each instrument covers, and the overall duration of isolated audio tracks.

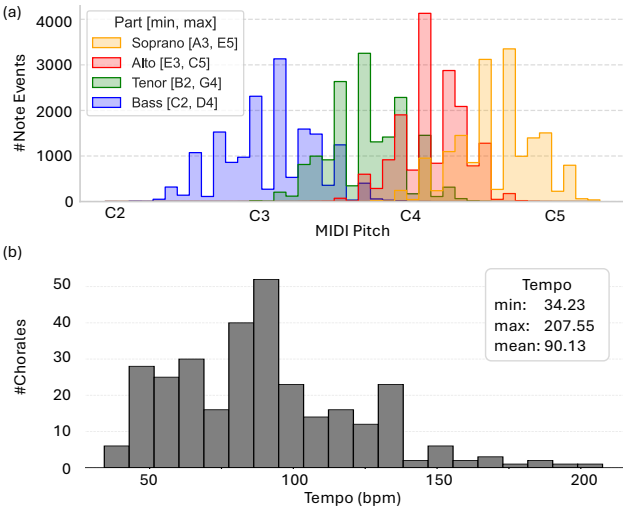


Figure 5. Histograms of (a) the musical pitches for the four parts (SATB) and (b) the overall tempi of the chorales.

centuation patterns for all relevant time signatures, a pattern handling pieces which start with a shortened measure, and a fallback pattern for other cases. Following historical notation conventions, the strength of the accent depends on two features. First, if no time signature is defined, no accentuation is applied. Otherwise, the bar line method is consulted: absent bar lines or "takt" lines [4] deactivate accentuation. Mensural lines between staves indicate a latent time signature, so accents are scaled to 50% (± 8 MIDI velocity units [5]) of the full accentuation (± 16 MIDI velocity units) used with regular bar lines. Depending on the synthesizer, this scaling factor may be adjusted, as each interprets velocity differently [23].

Basic tempo and dynamics: The baseline tempo of each chorale is derived from the average phrase length, assuming that a phrase is sung and played on a single breath. We employ a reference breath cycle duration of approximately 4.3 seconds. However, while this works well for the vast majority of chorales in our dataset, there are several outlier cases with rather short or long phrases or missing indications for phrase boundaries (rendering the whole piece in one extremely long phrase, technically speaking). Furthermore, we add a subtle random offset of ± 3 bpm to the basic tempo of each phrase to emulate natural human variation. This is modeled as a correlated variation with a Brownian distribution to avoid erratic behavior. For dynamics we assume a basic loudness of 100 MIDI velocity units. Like with basic tempo, basic loudness is also varied with each new phrase, this time in a range of ± 5 velocity units via an (uncorrelated) uniform distribution.

Phrasing: Breath marks and general rests define phrase boundaries, guiding tempo and dynamics shaping. However, general rests can be unreliable: some chorales include them only in the melody while other parts continue playing. Rests shorter than one beat are also ignored. As shown in Figure 4, towards phrase endings, dynamics decrease to 70% of the basic MIDI velocity and tempo slows down to 95% of the base tempo (ritardando); by the end of the verse, tempo reaches 78%.

Randomization: Each note's onset (via MPM's Compensating Triangle Distribution), velocity (Gaussian), and duration (uncorrelated Triangle) are individually randomized per musical part. This emulates the natural imprecision of human performances, commonly called "humanization", and prevents mechanically exact renderings. Offset magnitudes can be adjusted to reflect the ensemble's assumed professionalism.

Velocity range normalization: If calculated velocities fall outside the MIDI range, we apply a nonlinear compression that preserves the relative dynamic structure while ensuring valid values.

The output of the performance rendering model is an MPM file, which describes the performance in a structured format. Velocity and tempo contours are represented as time-continuous Bézier functions. To control subsequent synthesizers, these MPM files are exported as MIDI files, providing a widely supported interface while preserving the expressive timing and dynamics.

5. AUDIO SYNTHESIS

For each chorale, we generate an expressive MIDI file containing all parts as separate MIDI tracks and synthesize audio using a commercially available physical-modeling wind instrument engine, for a first overview on the topic we refer to e. g., [24]. [6] The engine is provided as a VST plugin, which we load into the digital audio workstation Reaper. [7] In the future, other instrument-specific synthesis approaches such as [25] could be considered. [8] Within Reaper, we prepare a project in which each instrument is assigned to a separate track, as listed in Table 1. For every chorale, we render 27 isolated tracks with 16 unique

[4] "short bar line through a subset of staff lines", according to the MEI Guidelines, `https://music-encoding.org/guidelines/v5/data-types/data.BARMETHOD.html` , last access: Dec. 2025.

[5] MIDI velocity is a unitless, integer-valued parameter in [0:127] that encodes the note intensity.

[6] `https://audiomodeling.com/swam-engine/`, last access: Dec. 2025.

[7] `https://www.reaper.fm/`, last access: Dec. 2025.

[8] As a freely available alternative, SoundFonts could be employed; however, our preliminary experiments have yielded mixed results.

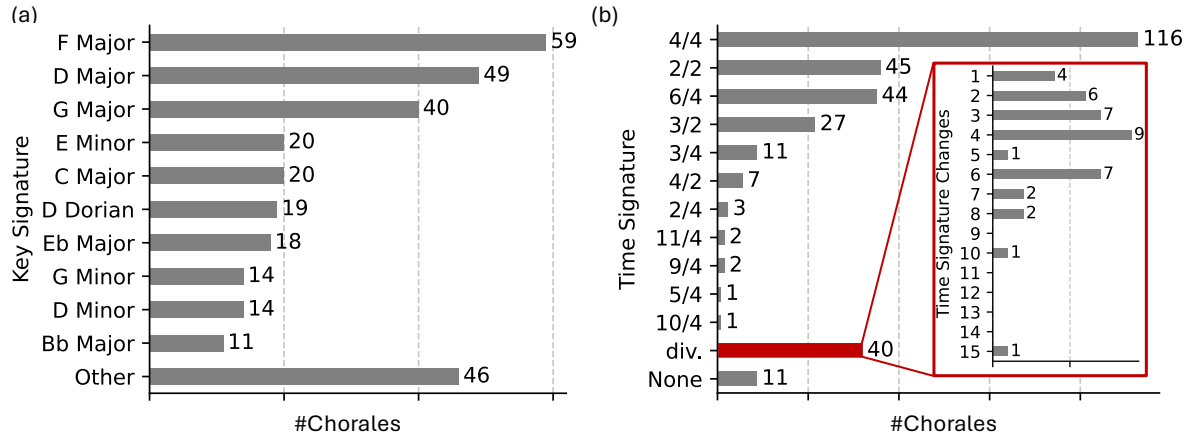


Figure 6. Distribution of key and time signatures in the dataset. (a) Distributions of top 10 key signatures. The category "Other" aggregates all remaining, less frequent keys. (b) Distributions of time signatures. The category "div." indicates that the chorale was notated with time signature changes, as depicted in the red box. The category "None" indicates that no time signature was present in the MEI file.

instruments. Note that some instruments can cover multiple parts; e. g., the trumpet is used for both the soprano and alto part. The mapping from instruments to parts is determined by matching playable pitch ranges and by the authors' practical experience. All instruments are tuned to a tuning frequency of 442 Hz.

The entire synthesis process is automated using Reaper's internal scripting language, ReaScript, which is accessed via its Lua API. Using Lua requires no additional setup, and the script can be executed directly from within Reaper. Compared to sample-based synthesizers, physical-modeling synthesizers require less memory and offer greater flexibility in adapting the generated sound. In our initial renderings, all synthesizer parameters are kept at their default settings without any reverb. Executing the full synthesis pipeline took approximately 90 minutes on a notebook equipped with an AMD Ryzen AI 5 Pro processor and 32 GB of RAM. The Reaper project file, as well as the Lua script, are available on the accompanying website. [9]

The isolated parts were intentionally exported without artificial reverberation. Although reverberation and stereo panning are important for perceptually realistic ensemble sound, ChoraleWind is designed as a machine learning dataset. This requires precise alignment between audio and annotations, which reverberation would compromise through temporal smearing that obscures note onsets and weakens the correspondence to symbolic labels. By providing dry signals, we prioritize annotation fidelity and controllability, while allowing users to add reverberation and spatialization as needed.

As listed in Table 1, rendering a single instrument for one part yields approximately 3 hours of audio. Rendering all parts and instruments results in approximately 86 hours of isolated audio tracks. Considering the possible four-part ensemble combinations (7 Soprano, 6 Alto, 7 Tenor, and 7 Bass) for 311 chorales, this theoretically amounts to approximately 640,000 distinct ensemble configurations.

In addition to providing the complete dataset for download via the accompanying website, we offer multitrack recordings of ten chorales that can be explored using the `trackswitch.js` player [26]. These chorales are identical to those used in the ChoraleBricks dataset and therefore enable direct comparisons between real performances and the corresponding synthetic renderings. Moreover, folder names and file naming conventions are consistent with ChoraleBricks, allowing the use of the same existing Python toolbox for sampling ensembles from the dataset. Further information is available on the website.

[9] `https://audiolabs-erlangen.de/resources/MIR/2026-ChoraleWind`, last access: Mar. 2026.

6. STATISTICS AND ANNOTATIONS

In the following, we summarize essential statistics for the ChoraleWind dataset regarding pitch, tempo, key, and time signatures. Figure 5a shows the distribution of pitches for the four SATB parts in MIDI notation. The lowest pitch observed is C2 in the bass part, while the highest pitch is E5 in the soprano part. The histogram indicates that each part occupies a characteristic pitch range; however, the ranges overlap in the boundary regions.

Figure 5b illustrates the distribution of tempi across the entire chorale corpus, directly extracted from the metadata of the MPM's tempo maps. The minimum tempo is 34.23 bpm in "O Traurigkeit, o Herzeleid." This chorale is characterized by many short phrases which in combination with the breath cycle duration of 4.3 seconds leads to a very slow tempo. In contrast, the maximum tempo of 207.55 bpm occurs in "Herr, du hast alles, Himmel und Erden", which contains extended passages without breaks or caesuras. Overall, the average tempo across the corpus is 90.13 bpm.

Figure 6a shows the distribution of the ten most frequent key signatures in the corpus. Of these, six correspond to major modes and four to minor modes (including D Dorian). Moreover, 255 of the 311 chorales (approximately 82%) are in keys with at most three sharps or flats, making them particularly suitable for performance by wind instruments. The category "Other" aggregates less frequent modes, such as E Phrygian or F Mixolydian, and illustrates the level of detail captured in the corpus annotations.

Figure 6b shows the distribution of time signatures in the corpus. A total of 116 chorales (approximately 37%) are notated in 4/4, followed by 45 chorales (approximately 14%) in 2/2. In addition, 40 chorales (approximately 13%)

exhibit changing time signatures. This is mainly due to the fact that some chorales originally do not specify a time signature; representing these pieces in modern notation required the introduction of changing time signatures. However, in 11 chorales no time signature is given.

The combination of MEI, MPM/MSM, and MIDI files provides rich annotation and performance-level information for the audio renderings. The MEI files include detailed score metadata (e.g., title, composer, key) as well as full note-level information, enabling analyses of melody, harmony, and part interactions. The MPM files encode expressive parameters such as tempo and articulation, allowing extraction of continuous velocity and tempo trajectories for synthesis or modeling. The MIDI files offer a convenient representation for machine learning tasks, providing precise onset and pitch information for each note event.

7. CONCLUSIONS AND POTENTIAL APPLICATIONS

We introduced *ChoraleWind*, a large-scale expressive wind-quartet dataset derived from the *Neues Thüringer Choralbuch* (NTCB), along with a fully reproducible end-to-end rendering pipeline. It provides aligned scores, performance annotations, and multitrack audio for 311 four-part chorales, covering the full chain from MEI to expressive MIDI and physical-modeling synthesis. The dataset enables controlled evaluation of score-to-audio rendering through precise ground-truth annotations of pitch, timing, dynamics, and articulation, which are hard to obtain from real recordings. By combining a rule-based performance model with physical-modeling synthesis, ChoraleWind captures key aspects of expressive wind-quartet performance while remaining fully reproducible and extensible.

Future extensions of the performance model include incorporating sung text to inform expressive timing and emotional coloring, as well as modeling long-range phrasing across multiple phrases. Currently, velocity and tempo contours are sampled at note onsets and held constant over each note; more advanced representations of continuous contours could enable more expressive and musically coherent synthesis.

Although synthetic, the dataset supports a broad range of applications, such as audio synthesis, audio–score alignment, and automatic music transcription (AMT). For audio synthesis, ChoraleWind can serve as a baseline approach for score-to-audio models, expressive MIDI generation, timbre modeling, and ensemble rendering, e. g., using DDSP-based [27] or diffusion-based [28] approaches to synthesize wind instrument sounds directly from score. For audio–score alignment [29, 30], ChoraleWind allows evaluation at the instrument level. This granularity enables detailed analyses of alignment behavior, such as identifying which instruments are prioritized (e. g., trumpets in the soprano or tuba in the bass), yielding insights that support the refinement of alignment methods. In AMT, early work focused primarily on piano recordings [31]. While recent Transformer-based models [32] extend transcription to other instruments, performance on diverse datasets like URMP [5] remains limited. With its varied instrumentation and modular design, ChoraleWind provides a valuable resource for evaluating and improving AMT models across a broader range of instruments.

Its compatibility with *ChoraleBricks* [1] enables direct comparisons between synthetic and real performances, establishing a transparent reference point for evaluating methods and advancing research across the aforementioned applications.

Acknowledgments

The music encoding is supported by KreativInstitut.OWL: a consortium consisting of Ostwestfalen-Lippe (OWL) University of Applied Sciences and Arts, Detmold University of Music, and Paderborn University, funded by the Ministry of Economic Affairs, Industry, Climate Action and Energy of the State of North Rhine-Westphalia, Germany. The International Audio Laboratories Erlangen are a joint institution of the Friedrich-Alexander-Universität Erlangen Nürnberg (FAU) and Fraunhofer Institute for Integrated Circuits IIS. MP, MM, and SB are supported by the Deutsche Forschungsgemeinschaft (DFG; German Research Foundation) under grant numbers 500643750 (MU 2686/15-1) and 555525568 (MU 2686/18-1).

8. REFERENCES

[1] S. Balke, A. Berndt, and M. Müller, "ChoraleBricks: A Modular Multitrack Dataset for Wind Music Research," *Transaction of the International Society for Music Information Retrieval (TISMIR)*, vol. 8, no. 1, pp. 39–54, 2025.

[2] A. Hankinson, P. Roland, and I. Fujinaga, "The Music Encoding Initiative as a Document-Encoding Framework," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Miami, Florida, USA, October 2011, pp. 293–298.

[3] R. Mauersberger and E. Mauersberger, *Neues Thüringer Choralbuch*, 6th ed., H. Peter, Ed. Berlin, Germany: Evangelische Verlagsanstalt Berlin, 1990.

[4] Z. Duan and B. Pardo, "Soundprism: An Online System for Score-Informed Source Separation of Music Audio," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 6, pp. 1205–1215, 2011.

[5] B. Li, X. Liu, K. Dinesh, Z. Duan, and G. Sharma, "Creating a Multitrack Classical Music Performance Dataset for Multimodal Music Analysis: Challenges, Insights, and Applications," *IEEE Transactions on Multimedia*, vol. 21, no. 2, pp. 522–535, 2019.

[6] C. Böhm, D. Ackermann, and S. Weinzierl, "A multi-channel anechoic orchestra recording of Beethoven's Symphony no. 8 op. 93," *Journal of the Audio Engineering Society*, vol. 68, no. 12, pp. 977–984, 2021.

[7] M. Schedl, D. Hauger, M. Tkalčič, M. Melenhorst, and C. C. S. Liem, "A Dataset of Multimedia Material about Classical Music: PHENICX-SMM," in *Proceedings of International Workshop on Content-Based Multimedia Indexing (CBMI)*, 2016, pp. 1–4.

[8] J. Garcia-Martinez, D. Diaz-Guerra, J. Anderson, R. Falcon-Perez, P. Cabañas-Molero, T. Virtanen, J. J. Carabias-Orti, and P. Vera-Candeas, "The Spheres Dataset: Multitrack Orchestral Recordings for Music Source Separation and Information Retrieval," *arXiv preprint arXiv:2511.21247*, 2025.

[9] K. Gerhardt and M. Kirsch, "Choralbücher from Northern Germany c. 1800 - a Dataset for Studies in Hymnology, Music Culture and Figured Bass," *Transactions of the International Society for Music Information Retrieval*, vol. 7, no. 1, pp. 1–10, 2024.

[10] S. Sarkar, E. Benetos, and M. Sandler, "EnsembleSet: A New High Quality Dataset for Chamber Ensemble Separation," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Bengaluru, India, 2022, pp. 625–632.

[11] J. Garcia-Martinez, D. Diaz-Guerra, A. Politis, T. Virtanen, J. J. Carabias-Orti, and P. Vera-Candeas, "SynthSOD: Developing an Heterogeneous Dataset for Orchestra Music Source Separation," *IEEE Open Journal of Signal Processing*, vol. 6, pp. 129–137, 2025.

[12] Y. Wu, J. Gardner, E. Manilow, I. Simon, C. Hawthorne, and J. Engel, "The Chamber Ensemble Generator: Limitless High-Quality MIR Data via Generative Modeling," 2022. [Online]. Available: http://arxiv.org/abs/2209.14458

[13] A. Kirke and E. R. Miranda, *An Overview of Computer Systems for Expressive Music Performance*. London: Springer London, 2013, pp. 1–47. [Online]. Available: https://doi.org/10.1007/978-1-4471-4123-5_1

[14] C. E. Cancino-Chacón, M. Grachten, W. Goebl, and G. Widmer, "Computational Models of Expressive Music Performance: A Comprehensive and Critical Review," *Frontiers in Digital Humanities*, vol. Volume 5 - 2018, 2018.

[15] A. Kirke and E. R. Miranda, *Performance Creativity in Computer Systems for Expressive Performance of Music*. Cham: Springer International Publishing, 2021, pp. 521–584. [Online]. Available: https://doi.org/10.1007/978-3-030-72116-9_19

[16] A. Friberg, R. Bresin, and J. Sundberg, "Overview of the KTH rule system for musical performance," *Advances in Cognitive Psychology*, vol. 2, no. 2, p. 145, 2006.

[17] S. Bauer, K. Klek, and B. Reich, Eds., *Orgelsätze zum Evangelischen Gesangbuch*. Munich, Germany: Verband Evangelische Kirchenmusik in Württemberg, Strube Verlag, 1996, Ausgabe für die Evangelische Landeskirche in Württemberg.

[18] M. Heinig, H. Rau, and D. Schuberth, Eds., *Orgelbuch zum Evangelischen Gesangbuch*. Kassel, Germany: Evangelische Kirche Deutschland, Bärenreiter-Verlag, 1993.

[19] Sächsische Posaunenmission, Ed., *Wachet auf – Ein Posaunenbuch*, 9th ed. Radebeul, Germany: Sächsische Posaunenmission e.V., 2002.

[20] F. Plewka, Ed., *Der Tag ist nicht mehr fern: Bläsermusik zum Weihnachtsfestkreis*. Munich, Germany: Posaunenwerk der Evangelischen Kirche in Mitteldeutschland, Strube Verlag, 2023.

[21] A. Berndt, S. Waloschek, and A. Hadjakos, "Meico: A Converter Framework for Bridging the Gap between Digital Music Editions and its Applications," in *Proceedings of the Audio Mostly Conference*. Wrexham, North Wales, UK: ACM, 2018.

[22] A. Berndt, "The Music Performance Markup Format and Ecosystem," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Bengaluru, India, 2021, pp. 50–57.

[23] R. B. Dannenberg, "The Interpretation of MIDI Velocity," in *Proc. of the Int. Computer Music Conf. (ICMC)*. Tulane University, New Orleans, USA: International Computer Music Association, Nov. 2006, pp. 193–196.

[24] D. H. Keefe, "Physical Modeling of Wind Instruments," *Computer Music Journal*, vol. 16, no. 4, pp. 57–73, 1992.

[25] I. Derenyi and R. B. Dannenberg, "Synthesizing Trumpet Performances," in *Proceedings of the International Computer Music Conference (ICMC)*, Ann Arbor, Michigan, USA, 1998.

[26] N. Werner, S. Balke, F.-R. Stöter, M. Müller, and B. Edler, "trackswitch.js: A Versatile Web-Based Audio Player for Presenting Scientific Results," in *Proceedings of the Web Audio Conference (WAC)*, London, UK, 2017.

[27] J. Engel, L. Hantrakul, C. Gu, and A. Roberts, "DDSP: Differentiable Digital Signal Processing," in *Proceedings of the International Conference on Learning Representations (ICLR)*, Virtual, 2020. [Online]. Available: https://openreview.net/forum?id=B1x1ma4tDr

[28] B. Maman, J. Zeitler, M. Müller, and A. H. Bermano, "Multi-Aspect Conditioning for Diffusion-Based Music Synthesis: Enhancing Realism and Acoustic Control," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 33, pp. 68–81, 2025.

[29] R. B. Dannenberg and C. Raphael, "Music Score Alignment and Computer Accompaniment," *Communications of the ACM, Special Issue: Music Information Retrieval*, vol. 49, no. 8, pp. 38–43, 2006.

[30] M. Müller, A. Arzt, S. Balke, M. Dorfer, and G. Widmer, "Cross-Modal Music Retrieval and Applications: An Overview of Key Methodologies," *IEEE Signal Processing Magazine*, vol. 36, no. 1, pp. 52–62, 2019.

[31] E. Benetos, S. Dixon, Z. Duan, and S. Ewert, "Automatic Music Transcription: An Overview," *IEEE Signal Processing Magazine*, vol. 36, no. 1, pp. 20–30, 2019.

[32] B. Maman and A. H. Bermano, "Unaligned Supervision for Automatic Music Transcription in The Wild," in *Proceedings of the International Conference on Machine Learning (ICML)*, Baltimore, Maryland, USA, 2022, pp. 14 918–14 934.

Automatic Hybrid Following in Real-Time Mixed Music: A Case Study with Antescofo and ipt˜ for Flute Playing Techniques

Nicolas Brochec
Tokyo University of the Arts
STMS, CNRS, IRCAM, Sorbonne Université
nicolas.brochec@ircam.fr

Jean-Louis Giavitto
STMS, CNRS, IRCAM, Sorbonne Université
jean-louis.giavitto@ircam.fr

ABSTRACT

This paper investigates how real-time recognition of instrumental playing techniques can extend automatic score following beyond the limits of pitch-based alignment. While systems such as Antescofo provide robust and largely plug-and-play score following, their listening model is primarily designed for stable, pitched events aligned with a fixed symbolic score. This makes them difficult to adapt to extended techniques, unpitched sounds, and musical forms involving partial improvisation or open notation.

To address these limitations, we explore a hybrid approach that combines multiple listening machines with complementary capabilities and allows dynamic switching between them during performance according to the musical context. Specifically, we integrate Antescofo with `ipt˜`*, a real-time playing technique recognition system based on lightweight machine learning models. We focus on the integration of real-time instrumental playing technique recognition as a means to enrich the listening process and support technique-aware navigation of the score.*

We evaluate this approach on the case of extended flute techniques, assessing both the feasibility of technique-aware following and the trade-off between system generality and performance. Results suggest that learning-based listening modules provide a practical compromise: they improve robustness for specific techniques while preserving much of the plug-and-play character supporting multiple works and performers. The results highlight a promising balance between generality, specificity, and performative robustness.

1. INTRODUCTION

1.1 Mixed Music

During the 1950s, experiments with electronic devices led to the emergence of electronic and electroacoustic music and to their blending. Unlike electronic music, which uses only electronic devices, or electroacoustic music, composed of pre-recorded sounds, mixed music (from French *musique mixte*) combines one or more instruments with an electroacoustic apparatus in a single musical work. The electroacoustic apparatus can be represented by one or more performers manipulating the tape or directly transforming acoustic sounds with electronic devices. Almost all the 20th-century leading mixed music composers were primarily contemporary composers, with complex musical languages and extensive use of instrumental gestures and techniques.

One of the main challenges in performing mixed music is synchronizing the instrumental and electronic parts. Unlike orchestra and ensemble music, mixed music combines electronic and instrumental parts performed through different media. The electronic part is executed by a machine operating on absolute time, in seconds, while the instrumental part is performed by a musician playing in relative time, the *tempo*. These two parts follow different time paradigms and cannot be synchronized as an orchestra would, where only the relative time paradigm is used. As for instrumental music, synchronization between the instrumental and electronic parts is necessary, as the quality of its performance relies on it [1].

Antescofo (Anticipatory Score Follower) is a system that enables real-time alignment of the electronic part with the live instrumental performance [2]. To our knowledge, Antescofo is one of the most robust following systems. However, it may have difficulties to follow complex musical gestures, such as extended playing techniques which are often used in contemporary mixed music. A second drawback is that the Antescofo's listening mechanism relies on alignment paradigm that is relevant for written music, but limits the system's use to entirely written forms, leaving out more open forms.

1.2 Instrumental Playing Techniques

Instrumental Playing Techniques (IPTs) comprise a diverse family of performance techniques related to "musical gestures" [3] and encompass the actions the performer applies to the instrument to produce different timbres. These actions can be types of articulations (e.g., *staccato*), ornaments (e.g., *trill*), preparation such as with mutes (e.g. straight mute), and sound-producing methods (e.g. *harmonic*) depending on physical features of the instrument. They have been practiced throughout the history of music worldwide, shaping local music identities [4].

The performance of playing techniques in Western written music is closely tied to their notation. As instrumental music gained independence from the 15th to the 17th centuries, treatises such as Praetorius's Syntagma Musicum (1618) [5] described instruments with minimal indications of playing technique. The 19th century introduced virtuosic performance, prompting the need to standardize the

Copyright: ©2026 Nicolas Brochec et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License 3.0 Unported, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

notation of playing techniques. In the 20th century, explorations of timbre through *Klangfarbenmelodie* and later avant-garde movements led composers such as Lachenmann, Ferneyhough, and Grisey to expand instrumental expression through extended playing techniques, establishing sound and timbre as core compositional elements [6].

1.3 Alignment versus Recognition

In score following, the alignment paradigm refers to the continuous matching of a live audio stream against a predefined symbolic representation of the musical work (or, in the audio-to-audio alignment, to another audio stream). Antescofo is a representative and successful instance of this paradigm: its listening machine estimates pitch and tempo to align performance events with a priori written symbolic score. While this approach has proven robust for a wide range of notated repertoires, it implicitly presupposes the existence of a fixed, determinate form against which the performance can be aligned. As a consequence, alignment-based following becomes problematic in musical situations where the form is partially open, non-deterministic, or only loosely specified. This includes scores that offer branching choices to the performer, as well as sections involving guided or free improvisation, where the notion of a unique reference trajectory through the score no longer applies. In such contexts, recognition-oriented approaches—focusing on identifying salient musical states, gestures, or playing techniques rather than precise score positions—appear more appropriate.

Conversely, approaches based solely on recognition do not fully address the requirements of mixed-music performance. While recognition systems can identify musical features such as playing techniques, timbral states, or gesture classes without relying on a fixed score position, they generally lack an explicit temporal and structural model of the musical form. As a result, they provide limited support for anticipation, synchronization, and the precise scheduling of electronic processes over extended time spans. In the context of mixed music, where the electronic part often unfolds according to carefully designed temporal relations, the absence of a reliable notion of progression through the piece can lead to fragile or musically unsatisfactory interactions. Recognition-based systems are therefore better suited to describing local musical states than to managing long-range temporal dependencies.

1.4 Hybrid Following: Extending alignment with specific recognition

This tension between alignment and recognition suggests that neither alignment nor recognition alone is sufficient, and that a productive approach lies in their combination: alignment provides a global temporal and formal framework, while recognition supplies context-sensitive information that can guide, constrain, or disambiguate the alignment process, especially in presence of IPT. In the other hand, recent studies in the field of real-time instrumental playing technique recognition have led to the development of reliable systems, such as `ipt~` Max external object [7], which are applied in the context of interactive co-creative systems [8].

These considerations motivate the present work and frame our exploration of hybrid listening strategies that can extend the applicability of score following beyond pitched sound and strictly pre-composed, fully specified forms. For the needs of our study, we focused on flute playing techniques as recent studies [8, 9] provide sufficient elements to implement a robust system and a comprehensive evaluation. Note, however, that some extended techniques for the flute still preserve sufficient spectral information for the limitations of current pitch-based listening machines to remain moderate. Our aim is not to address the performance of the two approaches to flute IPT, but rather to assess the feasibility of a hybrid listening approach within the framework of Antescofo, the robustness (*i.e.*, determinism and stability) of the system, and to examine how such an approach impacts the specification and writing of the electronic parts in mixed-music works.

1.5 Organization of the paper

In the first section, we present various flute-playing techniques, introduce Antescofo's score-following system, and examine existing methodologies for real-time recognition of playing techniques. In the second section, we detail our proposed system that merges Antescofo and `ipt~`. In the third section, we design an evaluation to observe how our proposed method performs in a real-world, similar setting for mixed music performance. Finally, we discuss the results of this evaluation and draw our conclusions.

2. BACKGROUND

2.1 The Diversity of Flute Playing Techniques

Each type of musical instrument has a wide range of instrumental playing techniques. In the case of the flute, various playing techniques can be produced by different blowing, tonguing, and ornamentation techniques, which can be performed separately or simultaneously. The sound of the flute is produced by the friction of the blown air into the mouthpiece [10]. Differences in blowing pressure, the degree of mouthpiece coverage, and blowing direction can produce distinct playing techniques [11]. The flute register extends from medium (C3 $\approx$ 261Hz) to treble pitch (C6 $\approx$ 2093Hz) with some exceptions [11]. Thanks to a unique key, professional flutes can produce B2 ($\approx$ 247 Hz) [12]. Professional flutists can reach pitches up to G6 ($\approx$ 3136 Hz) [11].

We provide a brief overview of flute playing techniques to illustrate the range of sounds the performer can produce. The *ordinario* is the ordinary sound of the flute; it is stable in frequencies. The *aeolian* sound involves blowing into the mouthpiece to make a noisy sound similar to the sound of air, often with a pitch component. The *flatterzunge* involves blowing while rolling the tongue, creating a sound of successive attacks. The *key click*, also known as *key percussion*, involves using the finger to strike flute keys and producing a percussive sound. The *multiphonics* consist of using a specific fingering to produce chord-like sounds [12], which is rarely stable over time and involves a lot of fluctuation in frequencies. The *pizzicato*, also known as *slap*, consists of making a powerful attack with the tongue. The *play and sing* technique involves singing while blow-

ing into the flute. The *staccato* is a short sound made with a sharp attack. The *tongue ram* technique involves forcefully propelling the tongue into the mouthpiece of the flute, producing a percussive, pitched sound. The *trill* involves alternating two adjacent notes rapidly, typically a main note, and a minor or major second above or below this main note. The *whistle tone* technique involves blowing a very soft, focused stream of air across the mouthpiece to produce high-pitched, whistle-like tones. Some of the playing techniques above are contemporary, as they extend the flute's sonic palette by using unusual gestures on the instrument. These techniques are the *key click*, *multiphonics*, *play and sing*, *pizzicato*, *tongue ram*, and *whistle tone*, and most of them have emerged in 20th-century contemporary music.

In the music of the 20th and 21st centuries, the use of playing techniques is often combined; some composers, such as Lachenmann and Ferneyhough, specialize in the extensive use of intricate playing techniques [13]. Combining playing techniques can be achieved through simultaneous actions on the instrument (blowing, tonguing, and ornamentation). For example, blowing and producing an *aeolian* sound with a *flatterzunge* tongue and trilling a key simultaneously are ergonomically possible.

2.2 Automatic Score Following with Antescofo

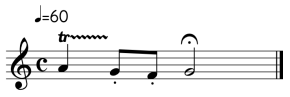
Score following is the process of tracking live players as they play through a composer's written score. The resulting information can be used, for example, to provide automatic computer accompaniment. The process involves entering a musical score representation into the computer, playing it as live MIDI or audio input, and comparing the computer's stored score to the live input. The early score-following systems of Dannenberg [14] and Vercoe [15] were based on MIDI.

Antecofo couples two different systems, a listening machine and a domain-specific programming language [16]. Composers use it to create an augmented score that describes musical scenarios in which electronic processes are scheduled to interact with a live performance.

An augmented score is a script composed of three main elements: *events*, which are recognized by the listening machine to describe the performance; *actions*, which are triggered when corresponding events are detected, when logical conditions are met, or simply as time elapses; and *synchronization strategies*, which link the performer's musical flow to the electronic processes.

Events can be grouped into three main categories. Abstract events (EVENT) do not correspond to a musical sound and are typically triggered manually. They were originally designed to enable an operator to manually trigger an event, causing the listening machine to wait for this notification and resume execution after a predefined delay. Atomic events (NOTE) represent elementary musical units, such as a single pitch. Finally, compound events act as containers for other events and describe more complex musical structures, such as simultaneous pitches (CHORD), rapid alternations (TRILL), or ordered sequences of heterogeneous events (MULTI).

In addition to basic events and actions, Antescofo also includes other functionalities, such as event attributes notated with a `@` sign. Event attributes enable the specification of additional information about the live performance. For example, the event attribute `@staccato` indicates that the event has a short temporal morphology, helping the listening machine to detect the corresponding event. The same applies to attribute `@fermata`, which indicates to the listening machine that the related event may be out of tempo.




```
BPM 60
TRILL (A4 B4) 1
NOTE G4 1/2 @staccato
NOTE F4 1/2 @staccato
NOTE G4 2 @fermata
```

Figure 1. Example of a conventional music score alongside its score for Antescofo, which includes *staccato* technique and *fermata* sign.

By combining the different attributes, Antescofo can cover a fair range of standard and extended flute playing techniques. Table 1 shows playing techniques and how to note them in the augmented score.

Playing Techniques	Antescofo Notation
ordinario	`NOTE p d`
staccato	`NOTE p d @staccato`
trill	`TRILL (p1 p2) d`
flatterzunge	`TRILL (p1 p1) d`
multiphonics	`CHORD (p1 p2 ...) d`
glissando	`MULTI (p1) -> (p2) d`

Table 1. Antescofo score notation corresponding to playing techniques, with `p` the pitch and `d` the duration.

Nevertheless, contemporary flute performance is far from being limited to the six techniques described in Table 1 as described in Section 2.1. Even though they can be notated in the augmented score, this does not guarantee that the listening machine will recognize them accurately during the performance. Some of them are out of the range of Table 1, such as unpitched playing techniques, *i.e*, *aeolian*, *key-click*, and *pizzicato*, which makes Antescofo difficult to use for these playing techniques. We think that using an external listening machine specialized in playing techniques in parallel of Antescofo would benefit the automatic following of specific IPT used in contemporary mixed music. In the following section, we examine the existing real-time recognition systems for playing techniques.

2.3 Real-Time Playing Techniques Recognition

Playing technique recognition is part of the Musical Information Retrieval (MIR) domain. Studies on the subject encompass a wide range of instruments. For Western instruments, studies mainly focus on acoustic guitar [17, 18], electric guitar and bass guitar playing techniques [19, 20], piano playing techniques [21], violin [22], cello [23], and flute [8]. Studies also include ones on Eastern instruments, as traditional music includes a wide range of playing techniques. Recent studies focus on the Chinese bamboo flute [24], and the bonang barung, an Indonesian Javanese gamelan instrument [25].

Yet, only a few studies focus on real-time recognition. These studies are about the cello [23], the guitar [18], the electric guitar [7], and the flute [8]. From these four studies, we can highlight several key points necessary for robust real-time playing recognition: sufficient data diversity

and quantity, a playing technique taxonomy when applicable, a meaningful audio signal representation, a lightweight classification algorithm, and a fast inference routine via dedicated buffers.

3. PROPOSED SYSTEM

In this section, we detail the proposed system that enables the automatic following of flute playing techniques. The first subsection describes the design of our recognition system, drawing on existing methodologies. The second subsection details how the IPT recognition system is linked to Antescofo.

3.1 Real-time Flute IPT Recognition System Design

The methodology for real-time flute IPT recognition is divided into the following steps, each with its own dedicated subsection: data selection, taxonomy, dataset setup and preparation, system architecture, training, and results.

3.1.1 Data Selection

A limited number of sound banks that include contemporary flute playing techniques exist. We identified four sound banks to ensure diversity and quantity: conTimbre [26], FullSOL [27], xSample [1] , and Geidai Flute Database (GFD) [28]. We selected these sound banks because, to our knowledge, they are the most exhaustive sound banks of flute IPTs, which include variable performance of the same IPT [26], recordings of different acoustic conditions, and with various performers, named as "P1" and "P2" in GFD [28].

3.1.2 From Data To Taxonomy

Unlike the cello and the guitar [23, 18], the performance of the flute IPTs is not only based on limb actions, but is also based on the quality of the blown air, e.g., *ordinario* and *aeolian*. These characteristics make gesture-based or hierarchical taxonomies harder to apply to the flute. Considering these constraints, we adopted a non-hierarchical taxonomy based on the 11 IPTs available in GFD [28]. Our task is then an 11-class classification problem using *aeolian*, *flatterzunge*, *key-click*, *multiphonics*, *ordinario*, *pizzicato*, *play-and-sing*, *staccato*, *tongue-ram*, *trill*, *whistletone* playing technique.

3.1.3 Dataset Setup and Preparation

We designed our dataset by combining the most populated datasets we have, "P1" from GFD, conTimbre, and xSample, by carefully moving the audio files in the corresponding playing technique category in GFD. The rest have been discarded. "P2" from GFD serves as the validation set, and FullSOL as the test set. To prepare the audio files for each subset, we first downsampled audio from 96 kHz and 48 kHz, depending on the audio file's origin, to 44.1 kHz, as flute harmonics can exceed 10 kHz [11]. Silence is trimmed with a threshold of -60 dBFS, as it is irrelevant. Using both short and long playing techniques, audio files are segmented into 1/3-second contiguous sequences and padded with zeros if shorter. To increase the duration of our training dataset, we applied domain-specific data augmentations, such as in [18]. Playback speed was randomly varied by ± 0.1 to emulate variability in IPT execution speed. Recordings were detuned by ± 100 Hz around A440 to reflect natural tuning fluctuations. Gaussian noise was added to replicate the noise introduced by signal amplification. Thanks to data augmentation, the training dataset length increased, totaling 16 hours.

[1] Available on the web: `https://www.xsample.de/xsample_flute.htm`.

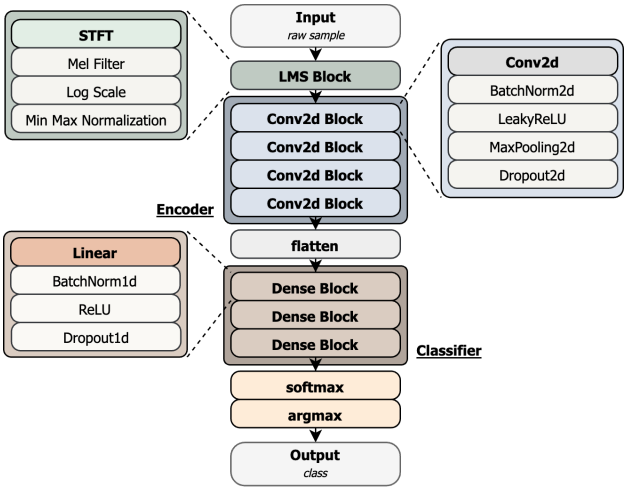


Figure 2. System architecture using a Logarithmic Mel Spectrogram, an encoder composed of four convolutional layers, and a classifier comprising three fully connected (dense) layers.

3.1.4 System Architecture

For the system architecture, we adopted the architecture from [29] and removed redundancy in the deep layers to make the model lighter. Our architecture consists of a Logarithmic Mel Spectrogram (LMS) block that first transforms raw audio into a grayscale spectrum. Long-duration IPTs benefit from long-range features, whereas short IPTs require high-resolution features [30]. As our dataset includes long and short IPTs, a compromise must be found between long-range and high-resolution features. Inspired by the study [31], in which concatenating spectrograms from different FFT window sizes to improve onset detection is suggested, we adopted a similar strategy to enhance short IPT classification (*e.g.*, *key-click*, *pizzicato*, and *staccato*). Additionally, a previous study showed that high-resolution features improve classification performance [23]. Based on these insights, we defined an LMS configuration that stacks three spectrums with FFT sizes of 512, 1024, and 2048, utilizing a hop size of 128 samples and 384 mel bands, which is three times that used for the cello [23]. The minimum frequency is set to 120 Hz based on the lowest reachable pitch (through *tongue-ram*).

The LMS block is followed by an encoder composed of convolutional layers and a classifier comprising three fully connected (dense) layers. Batch normalization and dropout layers are introduced to stabilize training and prevent overfitting. A flattened layer between the encoder and the classifier ensures dimension compatibility. The channels from the convolutional layers are 30, 60, 120, and 120, and the hidden sizes of the dense layers are 60 and 30. Kernel sizes for convolutional layers were set to `4x8`, `3x6`, `2x4`, and `2x4` to increase the receptive field along the time axis.

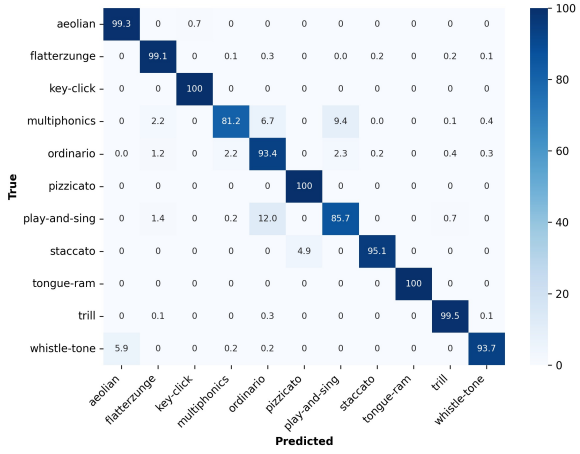


Figure 3. Normalized composite confusion matrix across the five tests showing true over predicted classes (%).

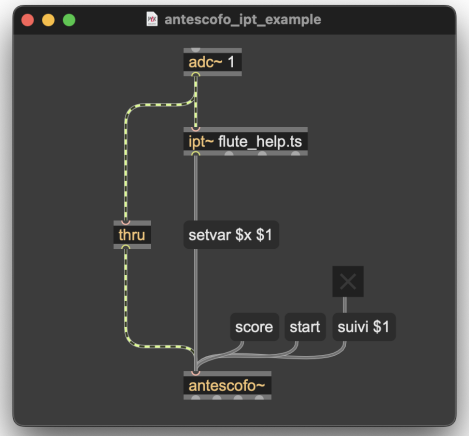


Figure 4. Max patch implementing `antescofo~` alongside `ipt~` object. IPT class index is sent to `antescofo~` using `setvar` message.

3.1.5 Training

We trained our configurations with a batch size of 128, cross-entropy loss, and the ADAM optimizer with a weight decay of 1×10^{-5}. The architecture is trained for up to 100 epochs. Training is stopped when the validation loss stops improving after 10 epochs. Five independent training runs are performed to ensure robust measurements.

3.1.6 Results

Following prior studies, we used macro F1 as our primary metric, as it provides a more accurate assessment of performance on unbalanced datasets [32]. Results show a strong macro F1 score of 93.4%±0.1, indicating robust generalization across separate training and test sources. The normalized composite confusion matrix across the five tests, as shown in Figure 3, indicates that models are accurate for almost all classes, except for *multiphonics* and *play-and-sing* (81.2% and 85.7%).

3.2 Crossbreeding Antescofo with `ipt~`

Our approach builds on existing mechanisms in Antescofo to integrate IPT recognition, as processed by `ipt~` into the score-following process without altering its core architecture. In particular, we take advantage of abstract events (EVENT), which were originally introduced to allow an operator to manually signal the occurrence of a musical event. In the standard workflow, the listening machine suspends progression until such a notification is received, after which the system resumes following.

In our hybrid approach, these abstract events are repurposed to represent musical events associated with instrumental playing techniques. Rather than relying on a human operator to notify their occurrence, the detection of these events is delegated to `ipt~`. The interaction between `ipt~` and Antescofo is achieved through explicit message passing, allowing the two systems to remain loosely coupled (they can be upgraded independently) while exchanging minimal but sufficient information.

More specifically, `ipt~` continuously outputs the index of the currently recognized playing technique. This information is transmitted to Antescofo using the `setvar` message, which updates a global internal variable. Inside the Antescofo score, this variable is monitored using the construction `whenever` which evaluates logical conditions on variable updates [33]. When the expected playing technique is recognized, the corresponding condition becomes true and triggers an action. In our case, this action simply advances the system to the next event, effectively replacing the manual notification that would otherwise be performed by an operator.

To prevent unwanted interference with the standard score-following mechanism, particular care is taken to control the behavior of the listening machine during these passages. While IPT-based detection is active, Antescofo's following is temporarily disabled to avoid erroneous jumps to subsequent events that could be mistakenly detected by the pitch-based listening module. At the exit of the IPT passages, the score following is reactivated. This ensures a clear separation between the alignment-driven and the ecognition-driven phases while preserving global coherence.

This design allows extended playing techniques to be handled in the same event-based framework as other musical events. As a result, the electronic part specification remains uniform, and all the advantages of Antescofo's programming model are preserved, including the organization of electronic processes around events and tempo-based synchronization. Importantly, Antescofo's tempo inference is based solely on inter-onset intervals, which allows tempo estimation to remain active even during passages where alignment is suspended and event progression is driven by IPT recognition.

In this first study, communication remains unidirectional: `ipt~` informs Antescofo, but Antescofo does not influence the behavior of `ipt~`. However, the same messaging infrastructure would allow actions defined in the augmented score to dynamically adapt recognition parameters according to the musical context. Such bidirectional interaction opens promising perspectives for future work, in which listening strategies could be actively shaped by the structure and semantics of the score itself.

For the validation presented in this work, we deliberately restricted the interaction to the mere occurrence of events detected by `ipt~`, without exploiting additional information provided by the recognition process. However, the

proposed scheme readily supports such extensions. making possible to design error-recovery strategies, for instance to handle missed events, delayed detections, or uncertain classifications during performance.

4. EVALUATION

The proposed hybrid approach proved straightforward to implement and to integrate within existing Antescofo-based workflows. Its evaluation was conducted at two complementary levels. First, we designed ad hoc scores specifically written to stress and probe the behavior of the listening machines, allowing controlled comparisons between different following strategies. Second, the approach was deployed in the production of full-scale musical works that were rehearsed and performed in concert conditions.

These real-world pieces enabled validation of the transparency of the hybrid approach from the perspective of electronic writing, including the synchronization between the human performer and the electronic voices. This transparency largely stems from the fact that extended playing techniques can be treated within the same "event-with-duration" paradigm as other musical events. As a result, the specification of the electronic part remains unchanged: electronic processes are still organized, triggered, and synchronized through events and tempo, exactly as in standard Antescofo practice. This confirms that the hybrid approach does not introduce a conceptual rupture in the compositional workflow.

In the remainder of this section, we focus on a quantitative comparison between Antescofo used alone and the proposed hybrid system. The evaluation targets listening performance along two main criteria: *robustness*, understood here as determinism and consistency of event detection across performances, and *precision*, defined as the temporal accuracy of detected events with respect to their reference onset times. This comparison is meaningful because most flute playing techniques preserve enough spectral information for Antescofo's listening machine to achieve alignment. In addition to pitch, Antescofo exploits temporal cues such as notated durations and real-time tempo inference, allowing progression when pitch information is unreliable. However, in this case, the alignment-based approach is expected to show greater variability across performances, which motivates comparison with the more deterministic behavior targeted by technique-aware listening.

4.1 Scores Setups

The first score of our evaluation, top of Figure 5, aims to test two situations: the alternation of timbres on the same pitch, and the chaining of *aeolian* and *ordinario* across different pitches. The second score, bottom of Figure 5, includes all playing techniques available in our recognition model at the exception of *ordinario*, *staccato*, and *trill* whose recognition can be safely left to Antescofo. It aims to test a situation in which playing techniques are chained without repetition across different pitches.

As previously explained in Section 3.2, a `whenever` statement monitors the global variable that reflects the index of the detected IPT class. These indices are attached to playing techniques starting with *aeolian* (0) and ending with *whistle-tone* (10) in alphabetical order. Each event

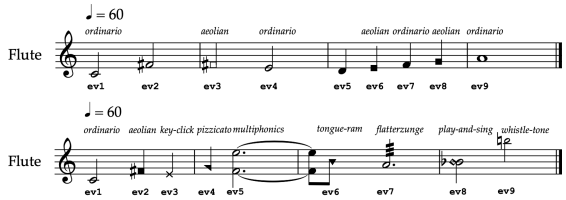


Figure 5. Top: Score 1 including *aeolian* and *ordinario* techniques. Bottom: Score 2 including *aeolian*, *key-click*, *pizzicato*, *multiphonics*, *tongue-ram*, *flatterzunge*, *play-and-sing* and *whistle-tone* techniques.

is labeled, and each score totals nine events. In the case where Antescofo is used alone, the corresponding augmented scores utilize as many notations as possible from Table 1. The parameters of `ipt~` are set to facilitate fast, rather than stable, recognition.

4.2 Metrics

Following a previous methodology [34], we define a reference onset time for each score event by mapping event timings onto a reference grid, thereby defining the ground truth between the score and its performance. We define the **error** $e_i = t_i^e - t_i^r$, the time lapse between the estimated alignment time t_i^d, the time at which the system reports the event, and the reference time t_i^r, the reference onset time according to the reference grid, for each event i.

Misaligned notes are notes that are correctly detected, but are too far from the reference onset time to be considered correct. We define a **misalign rate** such defined in [34], which is the percentage of correctly aligned notes p_e with absolute error $|e_i|$ greater than a threshold θ_e. We consider this alignment rate as the second metric for our experiments and use 300, 250, and 200 ms as our thresholds.

4.3 Results

Line charts Figures 6 and 7 represent the mean latency for each event across the six performance of scores in Figure 5.

Line chart Figure 6 shows that Antescofo performs well alone on events 3, 6 and 8, that are all *aeolian* techniques, see Figure 5. Line chart Figure 7 shows that Antescofo has difficulties in following techniques on events 5 and 9, which are respectively *multiphonics* and *whistle-tone* playing techniques, see Figure 5.

This line chart also shows that standard deviation for each event represented by the error bars is smaller for Antescofo and `ipt~` than Antescofo alone on events 2, 5, 6, 7, 8, and 9. Despite being slower than Antescofo on all event except on the fifth and ninth, this result shows that the use of `ipt~` led to a more robust following across different performances.

Table 2 shows that only when the threshold θ_e is set to 300 ms, the setup with Antescofo and `ipt~` performs better than Antescofo alone.

5. DISCUSSION

In the first experiment (Score 1), which alternates *aeolian* and *ordinario* techniques, Antescofo alone performs better than the hybrid approach. *Aeolian* sounds combine air noise with a pitched component, which appears sufficient for Antescofo's pitch- and time-based listening to operate

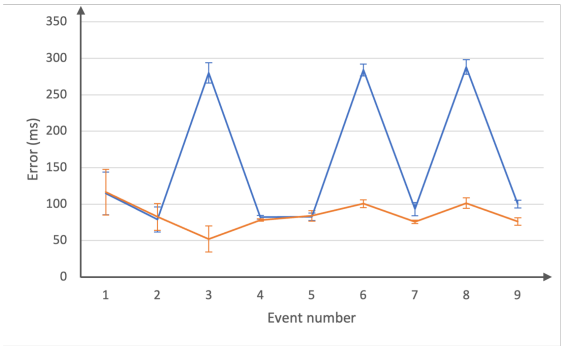


Figure 6. Mean error for each event across the six performances for the score 1, with error bars representing the standard deviation (ms). Antescofo performance (orange), Antescofo with `ipt~` performance (blue).

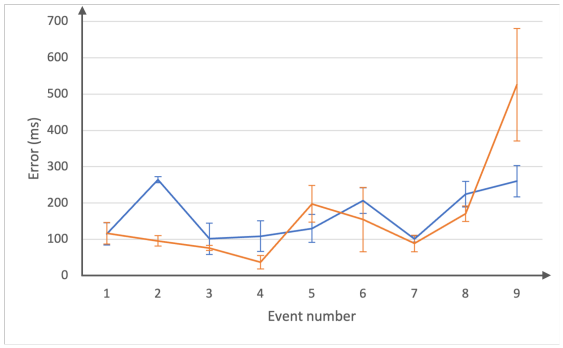


Figure 7. Mean error for each event across the six performances for the score 2, with error bars representing the standard deviation (ms). Antescofo performance (orange), Antescofo with `ipt~` performance (blue).

	Setup	Threshold θ_e		
		300 ms	250 ms	200 ms
Score 1	Antescofo	0.00	0.00	0.00
	Ant. and `ipt~`	7.41	31.48	33.33
Score 2	Antescofo	11.11	16.67	22.22
	Ant. and `ipt~`	**3.70**	20.37	40.74

Table 2. Misalignment rates for each setup across 6 performances (%).

reliably. In this case, the additional latency introduced by playing technique recognition leads to slower alignment without improving robustness. Misalignment rates confirm that Antescofo alone is fully adequate for such situations.

The second experiment (Score 2), which involves a broader range of extended techniques, reveals a different behavior. When used alone, Antescofo exhibits difficulties with techniques such as *multiphonics* and *whistle tones*, which are characterized by unstable or very low-energy spectral content. These difficulties result in greater variability across performances, as reflected by higher standard deviations in alignment time. In such cases, the progression of the listening machine relies primarily on temporal information to approximate the occurrence of musical events, which in turn leads to less reliable tempo inference. These issues are expected to worsen as the duration of sequences dominated by instrumental playing techniques increases. In contrast, the hybrid approach produces more consistent alignment for these events, even though it remains slower overall. This increased robustness comes at the cost of additional latency introduced by the recognition process.

These two experiments also reveal some limitations of the recognition model and `ipt~`, especially in the time it takes to recognize specific techniques. The line charts show that for the *aeolian* techniques, it takes more than 250 ms to be recognized. *Aeolian* techniques are events 3, 6, 8 on score 1, and event 2 on score 2. Line chart on Figure 7, shows that *tongue-ram*, *play-and-sing*, and *whistle-tone* techniques are recognized in more than 200 ms. Further improvements are required to enable faster recognition on the playing techniques recognition model side, which would make it more competitive against Antescofo's listening machine alone.

The error bars, representing the standard deviation in alignment time across the six performances, show that, for Antescofo, alignment time varies widely. When using `ipt~`, the error bars for events 2, 5, 6, 7, 8, and 9 show that playing technique recognition led to minor variations of alignment time, even though it scores a slower one than Antescofo alone. The misalignment rates in Table 2 show that for the performance on score 2 with a threshold of 300 ms, the use of `ipt~` yielded a lower misalignment rate (3.70%). However, when the threshold is shorter, misalignment rates show that Antescofo alone performed better, 16.67% against 20.37% for a threshold of 250 ms, and 22.22% against 40.74% for a threshold of 200 ms. The results suggest that, if robustness across performance is preferred to alignment time, or if the score is composed of many *multiphonics* and *whistle-tone* techniques, then our proposed hybrid approach is recommended. In any case other than these, Antescofo is recommended alone, and the usage of `ipt~` in the context of following flute playing techniques is case-specific.

Misalignment rates further highlight this trade-off. When using a relatively tolerant threshold (300 ms), the hybrid system outperforms Antescofo alone on the more complex score. For stricter thresholds, however, Antescofo alone yields better results. These observations suggest that the hybrid approach is particularly relevant when robustness and consistency across performances are prioritized over fast alignment, especially in scores dominated by unstable or weakly pitched techniques.

6. CONCLUSIONS

In this work, we investigated how integrating real-time playing-technique recognition can enhance automatic score following in mixed music contexts. By combining Antescofo with `ipt~`, we introduced a technique-aware, hybrid, score-listening workflow that handles a wide range of contemporary flute techniques. Our recognition model, trained on diverse sound banks, achieved strong generalization and provided reliable IPT information to drive conditional navigation in Antescofo scores.

Evaluation with professional flutists showed that the hybrid system improves robustness for unstable techniques, even at the cost of additional alignment time. These results suggest that technique-aware following is beneficial when stability across performances that involve many playing techniques is more important than fast alignment. The experiments also expose limitations of the current recognition model, notably the time required to reliably identify specific techniques, such as *aeolian*, *tongue-ram*, *play-and-sing*, and *whistle tones*. Reducing recognition latency would be a key factor in improving the competitiveness of the hybrid approach. The evaluation remains limited

to short excerpts and does not address failure cases such as missed detections or performer errors. Assessing these situations and testing longer, structurally richer pieces, as well as ones with open notation or partial improvisation, and exploring adaptation with other instruments constitutes an important direction for future work.

In addition to these observations, the proposed approach was successfully validated in a real-world mixed music setting. It was implemented in an original mixed music composition of approximately eight minutes, in which 20% of the events relied on flute playing technique recognition to drive the electronic part. This work was submitted by the first author to the music track of this conference.

Acknowledgments

This research was supported by the ERC REACH Project (GA #883313) and a MEXT scholarship from the Japanese Government awarded to Nicolas Brochec.

7. REFERENCES

[1] A. Cont and M. Rhéaume, "Synchronisme musical et musiques mixtes: du temps écrit au temps produit," *Circuit*, vol. 22, no. 1, pp. 9–24, 2012.

[2] A. Cont, "ANTESCOFO: Anticipatory Synchronization and Control of Interactive Parameters in Computer Music." in *International Computer Music Conference (ICMC)*, 2008, pp. 33–40.

[3] C. Cadoz and M. M. Wanderley, "Gesture-music," *Trends in gestural control of music*, 2000.

[4] P. V. Bohlman, *The Cambridge history of world music*. Cambridge University Press, 2013.

[5] M. Praetorius *et al.*, *Syntagma Musicum II: De Organographia, Parts III–V with Index*. Zea Books, 2014.

[6] M. Solomos, *From music to sound: The emergence of sound in 20th- and 21st-century music*. Routledge, 2019, pp. 28–30.

[7] M. Fiorini, N. Brochec, J. Borg, and R. Pasini, "Introducing EG-IPT and ipt˜: a novel electric guitar dataset and a new Max/MSP object for real-time classification of instrumental playing techniques," in *New Interfaces for Musical Expression Conference (NIME)*, Canberra, Australia, Jun. 2025.

[8] N. Brochec, M. Fiorini, M. Malt, and G. Assayag, "Interactive Music Co-Creation with an Instrumental Technique-Aware System: A Case Study with Flute and Somax2," in *International Computer Music Conference (ICMC 2025)*, Boston (MA), United States, Jun. 2025.

[9] N. Brochec, T. Tanaka, and W. Howie, "Microphone-based Data Augmentation for Automatic Recognition of Instrumental Playing Techniques," in *International Computer Music Conference (ICMC)*, 2024.

[10] N. H. Fletcher and T. D. Rossing, *The physics of musical instruments*. Springer Science & Business Media, 2012, pp. 503–537.

[11] J. Meyer, *Acoustics and the performance of music: Manual for acousticians, audio engineers, musicians, architects and musical instrument makers*. Springer Science & Business Media, 2009, pp. 64–70.

[12] C. Levine and C. Mitropoulos-Bott, *The Techniques of Flute Playing I/Die Spieltechnik der Flöte I*. Bärenreiter-Verlag, 2019.

[13] R. Feller, "Resistant Strains of Postmodernism: The Music of Helmut Lachenmann and Brian Ferneyhough," in *Postmodern music/postmodern thought*. Routledge, 2013, pp. 249–262.

[14] R. B. Dannenberg, "An on-line algorithm for real-time accompaniment," in *International Computer Music Conference (ICMC)*, vol. 84, 1984, pp. 193–198.

[15] B. Vercoe, "The synthetic performer in the context of live performance," in *Proceedings of International Computer Music Conference*, 1984, pp. 199–200.

[16] J.-L. Giavitto, J.-M. Echeveste, A. Cont, and P. Cuvillier, "Time, timelines and temporal scopes in the antescofo dsl v1. 0," in *International Computer Music Conference (ICMC)*, 2017.

[17] L. Su, L.-F. Yu, and Y.-H. Yang, "Sparse Cepstral, Phase Codes for Guitar Playing Technique Classification." in *International Society for Music Information Retrieval Conference (ISMIR)*, 2014, pp. 9–14.

[18] A. Martelloni, A. P. McPherson, and M. Barthet, "Real-time Percussive Technique Recognition and Embedding Learning for the Acoustic Guitar," *arXiv preprint arXiv:2307.07426*, 2023.

[19] Y.-P. Chen, L. Su, Y.-H. Yang *et al.*, "Electric Guitar Playing Technique Detection in Real-World Recording Based on F0 Sequence Pattern Recognition." in *ISMIR*, 2015, pp. 708–714.

[20] J. Abeßer and G. Schuller, "Instrument-centered music transcription of solo bass guitar recordings," *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, vol. 25, no. 9, pp. 1741–1750, 2017.

[21] B. Liang, G. Fazekas, A. McPherson, and M. Sandler, "Piano pedaller: a measurement system for classification and visualisation of piano pedalling techniques," 2017.

[22] L. Su, H.-M. Lin, and Y.-H. Yang, "Sparse modeling of magnitude and phase-derived spectra for playing technique classification," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 12, pp. 2122–2132, 2014.

[23] J.-F. Ducher and P. Esling, "Folded CQT RCNN for real-time recognition of instrument playing techniques," in *International Society for Music Information Retrieval*, 2019.

[24] C. Wang, E. Benetos, and E. Chew, "CBFdataset: A Dataset of Chinese Bamboo Flute Performances," Dec. 2021. [Online]. Available: https://doi.org/10.5281/zenodo.5744336

[25] V. L. Hardjanto and W. Wahyono, "Recognising Bonang Barung Gamelan Instrument Playing Technique Using Convolutional Neural Networks," *Applied Ethnomusicology*, vol. 1, no. 1, pp. 71–86, 2025.

[26] T. A. Hummel, "Algorithmic orchestration with contimbre," in *Journées d'Informatique Musicale*, 2014.

[27] C. E. Cella, D. Ghisi, V. Lostanlen, F. Lévy, J. Fineberg, and Y. Maresz, "OrchideaSOL: a dataset of extended instrumental techniques for computer-aided orchestration," *arXiv preprint arXiv:2007.00763*, 2020.

[28] N. Brochec and W. Howie. (2025, Jan.) GFDatabase: A Database of Flute Playing Techniques. Accessed: 2025-12-16. [Online]. Available: https://doi.org/10.5281/zenodo.14712391

[29] M. Fiorini and N. Brochec, "Guiding Co-Creative Musical Agents through Real-Time Flute Instrumental Playing Technique Recognition," in *Sound and Music Computing Conference (SMC)*, Porto, Portugal, Jul. 2024.

[30] D. Li, M. Che, W. Meng, Y. Wu, Y. Yu, F. Xia, and W. Li, "Frame-level multi-label playing technique detection using multi-scale network and self-attention mechanism," in *Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[31] J. Schlüter and S. Böck, "Improved musical onset detection with convolutional neural networks," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 6979–6983.

[32] M. Grandini, E. Bagli, and G. Visani, "Metrics for multi-class classification: an overview," *arXiv preprint arXiv:2008.05756*, 2020.

[33] J.-L. Giavitto. (v1.1, 2055) The Antescofo Documentation. Accessed: 2025-12-15. [Online]. Available: https://antescofo-doc.ircam.fr/

[34] A. Cont, D. Schwarz, N. Schnell, and C. Raphael, "Evaluation of real-time audio-to-score alignment," in *International Symposium on Music Information Retrieval (ISMIR)*, 2007.

20 Years Zirkonium and Klangdom at ZKM

Ludger Brümmer
HfM Trossingen
info@ludgerbruemmer.com

Götz Dipper
ZKM | Hertzlab
goetz.dipper@zkm.de

Dan Wilcox
ZKM | Hertzlab
dan.wilcox@zkm.de

ABSTRACT

2026 marks the 20th anniversary of the first release of the Zirkonium spatialized-music composition environment, created over three versions from 2006-2026. Initiated for use with the Klangdom, ZKM's 47.4-channel dome sound system, Zirkonium is focused on accessible use of spatialization algorithms for composers and live performers. This paper describes the historical development stages of the three versions of the Zirkonium software, provides an overview of practical working methods with Zirkonium, and explains the current technical status of development in the third generation of the project, as well as future plans.

1. INTRODUCTION

The desire to use spatial position as a compositional parameter gained significant importance in the history of music with the advent of electroacoustic music in the 20th century. A spectacular milestone in spatial music was the German pavilion's Kugelauditorium[1] at the 1970 World Expo in Osaka, in which the audience was surrounded in all directions by loudspeakers, 50 in total, mounted on the walls of a spherical concert hall. However, one major technical problem could not be satisfactorily resolved at the time: how to control these 50 speakers in a flexible and convenient way, so that composers could freely assign spatial positions to a large number of sounds and easily create movements between positions.

Analog audio technology was used in the Kugelauditorium in 1970 and 30 years later, at the start of the 2000s, digital technology had advanced to the point where computers had become sufficiently powerful and affordable and audio interfaces could handle the required number of audio channels. During this period, the ZKM Institute for Music and Acoustics launched the Klangdom project in which nearly 50 speakers are arranged in a dome-like formation around the audience. A central component of the project is the Zirkonium software which enables flexible and convenient control of the Klangdom, thereby opening up new expressive possibilities. Zirkonium continues to be developed to this day. This paper first describes the development of the hardware and software in roughly chronological order, then outlines their flexibility in practical use, and finally offers a look ahead to the future.

Copyright: © 2025 Ludger Brümmer, et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License 3.0 Unported, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

[1] http://www.medienkunstnetz.de/works/stockhausen-im-kugelauditorium/

2. PROJECT START

In 2003, following a change in management at the former Institute for Music and Acoustics (now Hertzlab) at ZKM Karlsruhe and initiated by funding from the Baden-Württemberg Future Initiative, the idea arose to create a universal multi-channel playback environment. This system was to focus on the precise positioning and dynamic movement of sounds in space, rather than prioritizing simulations of acoustics, such as SPAT[2], or an interpretation of sound, such as through acousmonium.

The first spatial sound works were realized with the Zirkonium software in the ZKM Klangdom ("sound dome") as early as 2005-06, see Figure 1. Ludger Brümmer's composition "Glasharfe" [1] served to test and verify the functionality of the production chain, but also formulated compositional possibilities for spatial sound environments as the first work performed in the environment.

3. DEVELOPMENT GOALS OF THE KLANGDOM AND ZIRKONIUM



Figure 1. The Klangdom (sound dome) in the ZKM Cube with DMX-controllable LED feedback system. Circa 2015

In 2003, there were only a few permanently-installed spatial sound systems. IRCAM operated a larger loudspeaker system in the Espace de Projection. The majority of the other systems were smaller, such as the IEM Cube in Graz. In addition, there were a number of acousmoniums at GRM Paris, Bourges, Birmingham, and Brussels.

[2] https://forum.ircam.fr/projects/detail/spat/

The new spatial sound environment at ZKM should be universal, portable, scalable, and easy to operate via GUI for first time use, enabling composers to perform their works not only in the ZKM Cube studio which housed the Klangdom, but also in a wide variety of locations without the support of a technician. Creatives should be able to present their spatial sound choreographies with good quality in a variety of architectures with any number of speakers and they should be able to simulate the composition on their own computer, e.g., using headphones. The digital system should be transportable and compatible with different hardware and the acoustic system had to be able to be set up and operated without the expertise of specialists. Consequently, the spatial sound simulation process had to be robust enough to reproduce the compositional results even under difficult conditions. In addition, the system should be interoperable and controllable via the OSC (Open Sound Control) protocol.

No such system existed at that time, and the task of combining all of these requirements was complex. Following a comprehensive evaluation of all available spatialization methods, the decision was made to use VBAP (Vector-Based Amplitude Panning) [2] initially with HOA (Higher Order Ambisonics) added as an additional option later on.

After defining the individual development steps, the project team began working with Dr. Chandrasekhar Ramakrishnan (Zirkonium Software), Dipl.-Ing. Bernhard Sturm (implementation of the loudspeakers in the cube architecture), Dr. Joachim Goßmann (concept and implementation), and Ludger Brümmer (management and development of the artistic practical tasks).

4. HARDWARE

The choice of loudspeakers was 47 Meyersound UPJ-1P with rotatable horns and four 700 HP subwoofers. The loudspeakers are hung from mobile trolleys that can be adjusted horizontally. These trolleys can be dynamically positioned on three of the six loudspeaker rings, with the third ring being height-adjustable, see Figure 1. [3]

The audio channels are transmitted between the computer, mixing console, and DA converter in the ZKM Cube via fiber-optic MADI and can be controlled individually, currently using a LAWO console. For performances outside the ZKM, the Zirkonium software, a medium-performance computer, an appropriately sized DA multichannel converter, amplifiers, and loudspeakers are required.

5. ZIRKONIUM 1

5.1 3D Realtime Feedback System

A key focus was the development of a visual feedback system whose interface had to be capable of displaying 50 or more speakers and 50 or more sounds and their movement simultaneously and in real time. We opted for a 3D model in the form of a hemisphere, on which the speakers were displayed in green numbers in their triangulated grouping, see Figures 2 and 3 for a design concept. The gray (later green) speaker icons, could indicate amplitudes by changing color and size. A number next to the speaker icon could optionally represent the

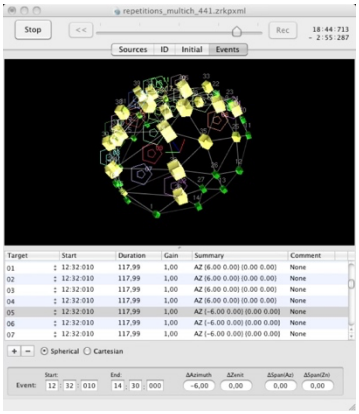


Figure 2. Zirkonium 1 interface. 2007

speaker's hardware output channel. The single channels were represented as sound IDs by pentagonal outline objects with definable colors and numbers. These were also able to display the respective amplitude of the channel by changing their size. The 3D display could be rotated during operation to show not only the horizontal distribution but also the height gradation of the respective sounds.

In addition to the digital display of sound movement, a DMX-controlled feedback system was developed for the room. A high-intensity LED module was installed on each speaker and software was programmed to control these modules with information from Zirkonium, so that the movement of the sounds could also be displayed in real space, see Figure 1. [4]

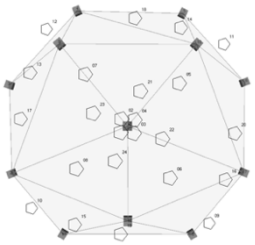


Figure 3. Zirkonium 1 Dome view concept. ID pentagons, speakers, and VBAP speaker triplets shown. 2005

5.2 Environment Setup

Special setup windows allow the configuration of speakers in terms of their height, distance, and angle. Another window assigns speakers to hardware outputs and IDs to sound file channels or audio interface inputs. A test mode sound helps to identify discrepancies between the digital and real representations of the Klangdom using a mouse and white noise. The speaker patch and routing can be saved to a file and exchanged between computers.

5.3 Editing Sound Movement

Zirkonium 1 supports the input of mono or various interleaved sound files. Each sound file channel is assigned a unique ID, which enables individual or grouped move-

ment in the sound dome. The mathematical description of the movement in Zirkonium 1 is done using spherical or cartesian parameters, which are absolutely reproducible and can represent complex movements extremely efficiently with just a few numbers. For example, azimuth value 2 (Pi) rotates a sound 360 degrees modulo horizontally and zenith value 1 (Pi) rotates it 180 degrees palindrome modulo vertically. Temporal start and end values, the spreading of the sound in azimuth and zenith span (0 to 2 or 0.5), and the adjustment of the volume per time segment or "event" are also taken into account. As an alternative to spherical parameterization with delta values, cartesian parameterization with values between X/Y -1 and 1 offers a way to define linear movements. Thus, each segment is described only by the parameters start time, end time, delta azimuth, delta zenith, span azimuth, span zenith, and gain. A composition can be reproducibly choreographed with 30 or 40 segments, for example, i.e., 280 values. See Figure 2 for the Zirkonium 1 interface with both 3D Dome view and event list.

When outputting Zirkonium to the DA converter, all channels can be converted to any sampling rate in real time. Additionally, an alternative HRTF simulation of sound output is provided for headphone use.

5.4 Additional Production Software Tools

To facilitate flexibility with other environments, the Zirkonium 1 software was joined by additional software tools: the Zirk2 AudioUnit plugin for controlling Zirkonium from a DAW; the DMX controller Zirkonium Licht, in which data from Zirkonium could be used to generate visual information; and ZirkQT, a Quicktime player that synchronizes video files with Zirkonium playback.

6. ZIRKONIUM 2

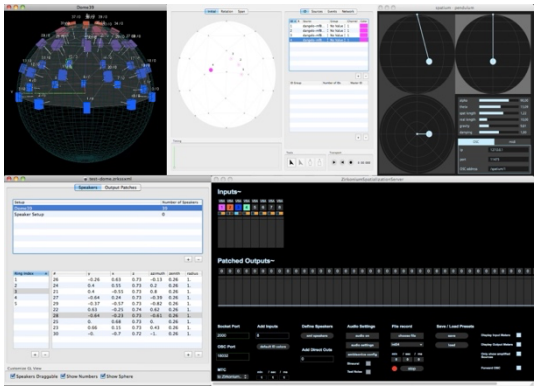


Figure 4. Zirkonium 2 interface. Top (left to right): Dome view, recording and graphical trajectory editor, Spatium[3] pendulum movement generator; Bottom: speaker setup editor, and spatialization server. 2012

Due to its 32-bit architecture, Zirkonium 1 is compatible with Mac OS X 10.3 up to and including macOS 10.14 and runs stably on these operating systems. However, starting in 2009, a second version of Zirkonium was developed by David Wagner with a completely new approach. [5] The previous native macOS application was replaced by a server programmed with Max that could be controlled by various clients, see Figure 4. This innovation was intended to promote interoperability with other software solutions and the easier development of custom clients by users. One important innovation was a client for graphical control of sounds via mouse whereby the movements of individual channels could be edited graphically in various modes. In addition, the existing plugin in Zirkonium 1 was replaced with a new strategy to facilitate synchronization with any DAW using MIDI Time Code (MTC) and OSC. The architecture of Zirkonium 2 enabled spatialization parallel to editing in any DAW, whereas in this usually only took place after the composition process was complete and a render imported into Zirkonium 1. From the outset, Zirkonium 1 and 2 enabled remote control of movements or real-time inputs via OSC and live input audio from internal streaming software such as Soundflower or a mode in which Zirkonium 1 is defined as an output device.

Since it has always been a priority for the development team to integrate Zirkonium as part of a production chain with other software solutions, the openness of the new modular architecture quickly proved to be advantageous. An earlier version of the SpatGris project, developed by the Research Group in Spatial Immersion University Montreal, took the functionality of the Zirkonium 1 AudioUnit plugin and refined it, first for Zirkonium 1 and then for the Zirkonium 2 server. [6]

7. ZIRKONIUM 3

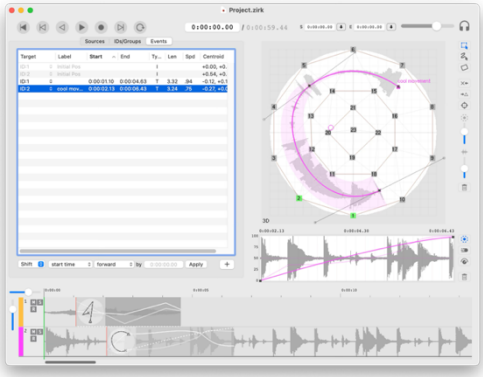


Figure 5. Zirkonium 3.7 interface. 2021

The trajectory path-drawing metaphor introduced as separate client for Zirkonium 2 improved accessibility, however the modular nature of requiring multiple applications to work with a piece was often a stumbling block to new users. Beginning in 2015, Chikashi Miyama took over development on the project and consolidated the Zirkonium 2 trajectory editor with a new spatialized sound server using libpd [7] into a single application. The event path metaphor was further extended by introducing DAW-style GUI views for the movement and event sequencing: selected-event "automation curves" in the *Motion view*, and "ID tracks" with event rectangles in the *Event view*. Event movements and control curves were implemented as multi-node Bezier curves and, as a nod to Zirkonium 1,

[3] https://ruipenha.pt/spatium

parameter-based tools allow for generating rendered circular or spiral-based paths. Additionally, control curves allow for non-linear movement along trajectory paths for better matching of dynamic events to positions. This renewed focus on providing an easy visual spatialization workflow allows for quick usage by new users, guest artists, and production engineers: add sound channels, map channels to IDs, draw event movement, play. Quick gestural movement can even be recorded via mouse or incoming OSC messaging. See Figure 5 for the Zirkonium 3 interface.

The SpeakerSetup tool for creating Zirkonium speaker setup XML files was also rewritten with an intuitive 3D view showing listener and speaker positions and can now accept speaker delay compensation in ms. [8] Additionally, the ZirkPad application for iPad was developed for live, multi-touch control of ID and Group positions (see Figure 8) and the SpatDIF exchange format [9] was integrated for universal import/export. [10]

7.1 Zirkonium 3.4

With the transition from development by Chikashi Miyama to Dan Wilcox in the summer of 2017, the state of the software project was re-examined. Zirkonium 1 "Classic" was a stable tool and in continual use, although whose longevity was in doubt. At the time, Zirkonium 3 represented a new experiment with it's exciting and intuitive interface and visual path paradigm, yet suffered from performance and stability issues due to a rapid development process. Since both versions 1 and 3 were in use concurrently, which project should be the focus of limited resources moving forward? The source code of both were studied and it was decided to improve version 3 as the version 1 code would have required a much more extensive rewrite due to changes in the Obj-C language and Apple frameworks since 2006. The refactoring of Zirkonium 3 commenced in early 2018 and finished with the first "stable" release of Zirkonium 3.4 for the *next_generation* festival at ZKM in June of 2019.

The application has been continually updated between 2019 and 2022 based on user feedback. Event gain curves, a master volume slider, an automatic timestamped-save option, and system "dark mode" support have been added. Newer versions now better facilitate more modern workflows with external software such as syncing playback over MIDI Clock, inter-app audio routing using JACK [11], expanded live control via OSC, including customizable ID message formatting for other spatialization servers such as IRCAM Panoramix [12]. Due to stalled SpatDIF development in the meantime, support for the format was removed. As of Zirkonium 3.7, the software is built for both Intel and Apple Silicon (arm64) architectures, an important step for current longevity.

At the time of writing, with the upcoming release of Zirkonium 3.8, the project comes full circle: Zirkonium 1 "Classic" parameter-based movements are now natively supported, including the import of original project files, thus allowing continued playback of older pieces on newer systems. Tested internally over the course of 4 years, projects were compared in both Zirkonium 1 and 3 to ensure the accuracy of the translation.

8. PRACTICAL USAGE

The following section describes various scenarios in which Zirkonium is used. Different practical requirements are presented, along with the corresponding features that enable Zirkonium to meet these requirements.

8.1 Concert Practice at ZKM

We maintain an intensive concert practice at ZKM with more than 1000 concerts since its founding in 1989. The focus is on electroacoustic music, with a wide range of different genres, such as fixed-media pieces, combinations with acoustic instruments, live electronics, live coding, and installation concerts.

Our concert hall, the "Kubus" (Cube), was inaugurated in 1997. At that time, it was equipped with an 8-channel setup consisting of d&b 1220 loudspeakers; since 2006, it has been equipped with our Klangdom (sound dome).

8.1.1 Guest Artists

We do not see ourselves primarily as a concert promoter; rather, we organize concerts to present to the public new works by our guest artists which are created during their residencies at the ZKM. During their time with us, composers have the opportunity to specifically design for and extensivly test their works on the Klangdom. They are welcome to use Zirkonium, but they can also use their own spatialization software at any time. Since 2006, around 70 works in the fixed-media composition genre have been created through residencies, in addition to numerous other works with live performances.

8.1.2 Other Compositions

In addition to concerts featuring works newly created at ZKM, there are also concerts to which composers and performers are invited to play pieces from their repertoire, typically not native to the Klangdom. Here we face the challenge of adapting pieces over comparatively short rehearsal times so that they make the best possible use of the Klangdom's capabilities. Zirkonium is often used for this purpose.

8.1.3 Zirkonium Challenges

Zirkonium faces the dual practical challenge of integrating itself into a wide variety of technical setups on the one hand, and supporting different design approaches to spatialization on the other. Both challenges will be explained in more detail below.

8.2 Dramaturgical vs. Statistical Spatialization

In our experience, there are two fundamentally different creative approaches to sound spatialization, which we would describe as the "dramaturgical" and "statistical" approaches.

8.2.1 Dramaturgical Spatialization

Here, the composer uses the spatial position and movement of individual voices, sounds, or tracks for dramaturgical purposes. For example, specific spatial directions can be used (front/back, top/bottom, etc) or movements with different directions and/or speeds. Typically, only individual voices are treated in this way, namely those that are in the foreground in terms of sound. A graphical tool is ideal for this type of spatialization, as it allows the trajectory of a movement to be drawn manually with just a few mouse clicks. Such a tool was introduced by Zirkonium 2 in 2014.

8.2.2 Statistical Spatialization

When it comes to the number of voices that can be spatially arranged in a dramaturgical way, our hearing sets limits. When listening, we can probably perceive and follow no more than about three voices simultaneously and independently of each other. This perhaps corresponds to what is possible in musical polyphony with independent voices, where, with a higher number of voices, individuals are perceptually combined into groups. If such a higher number of voices is to be spatialized, a statistical approach is appropriate. Here, the composer is not interested in specifying a precise spatial direction or concrete movement for a particular sound. Rather, they want to distribute the sounds as diffusely as possible within the sound dome — or, if applicable, within a limited area.

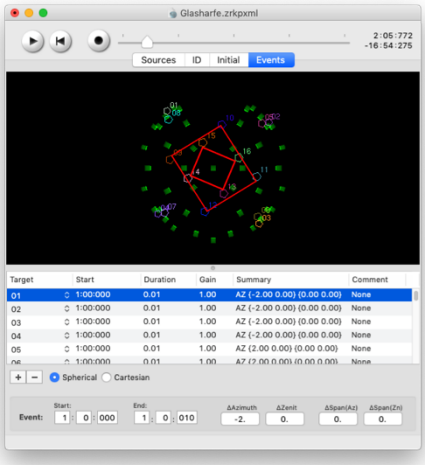


Figure 6. *Glasharfe* (Ludger Brümmer) [1] in Zirkonium 1. Two quad sources marked by red rectangles. 2007

An algorithmic tool is ideally suited for statistical spatialization and such a tool was introduced by Zirkonium 1 in 2006. With little effort, it can be used to generate, for example, movements of many overlapping spirals that distribute the sounds quasi-randomly on the surface of the sound dome. In addition, algorithmic movements can be generated with other software (e.g., Max or SuperCollider) and sent to Zirkonium via OSC messages.

The fact that, above a certain number, individual voices merge into a group in our perception is useful in practice: it only makes sense to manually spatialize a small number of simultaneous voices. Beyond that, we can resort to more efficient algorithmic methods.

Zirkonium 3 is particularly well-suited for combining dramaturgical and statistical spatialization because it includes both the graphical tool from Zirkonium 2 and the algorithmic tools from Zirkonium 1.

8.2.3 Extreme Case: Granular Synthesis

Granular synthesis is an extreme example of the advantageous application of statistical spatialization. Here, it may be desirable to assign each individual grain its own location in the sound dome. The following strategy has proven effective in keeping the effort required for spatialization manageable: using software employed for granular synthesis, the grains are played back on a small number of tracks — e.g., quad or an 8-channel circle —where panning between tracks can also be used. In Zirkonium, these tracks are then mapped as virtual loudspeakers, which can also be moved within the sound dome. If panning has also been used, care must be taken to ensure that the respective virtual loudspeakers always remain spatially consistent with one another, and that no single speaker moves independently. See Figure 6, which shows two quad sources marked with red rectangles. Here each quad can be moved independently, for example they can be rotated or moved up and down. However, the virtual speakers inside a quad group must always maintain their fixed positions within the quad.

8.3 Native Zirkonium Compositions

In terms of performance practice, two main scenarios of spatialization can be distinguished: live spatialization and spatialization prepared in advance using a digital audio workstation (DAW). Of course, hybrid forms are also possible, in which parts of the spatialization are prepared in advance and other parts are spatialized live.

8.3.1 DAW Spatialization

DAW spatialization is the usual strategy when a work is composed using a common DAW such as Reaper or ProTools. The result is typically a fixed-media composition or its fixed-media part.

If spatialization is to be used as an independent musical parameter in such a work, it is important that the composer is able to design the spatialization from the very beginning of the composition process and design of individual sounds. This was only possible to a limited extent with Zirkonium 1 via an AudioUnit plugin for live streaming and DAW automation. Many other software projects still only offer a similar option, e.g. the IEM Plugins[4], spatGRIS [6], Spatial Sound Wave[5], and Spatial Audio Designer[6]. In our opinion, the major disadvantage of this method is the movement of a sound source, i.e. trajectory, can only be edited in a very unintuitive manner. This is

[4] https://plugins.iem.at
[5] https://www.idmt.fraunhofer.de/content/dam/idmt/documents/IL/spatialsound_wave_en.pdf
[6] https://www.newaudiotechnology.com/products/spatial-audio-designer-producer-plug-in

true even if the trajectory can be "drawn" in one stroke using a mouse movement on a 2D graphic with value and time as their axes. This entered or drawn trajectory is typically broken down into a series of points and stored as DAW automation, in the form of either cartesian X/Y values or polar azimuth/elevation values, often in separate "tracks".

If the composer wants to adjust a movement that has already been entered, they must edit the relatively abstract X and Y (or azimuth and elevation) values independent of one another without combined axis visual feedback, which is not intuitive, but rather laborious and prone to error. Another disadvantage of this method is that the playback of the composition is tied to the specific DAW software, which makes long-term availability and archiving of the work difficult.

Zirkonium offers a solution to this problem that, to our knowledge, is not included in any comparable software in this form to date. Zirkonium 2 (and Zirkonium 3) can be synchronized with a DAW via MIDI Time Code (or MIDI Clock), with the play/pause/stop functions controlled by the DAW. The DAW plays stems to Zirkonium via audio routing software, e.g., Soundflower (historical), JACK [11], or Blackhole. Each source channel from the DAW can be freely placed and moved within the sound dome by Zirkonium as an ID and the movement is drawn as a trajectory in a 2D representation using a convenient graphical tool. The trajectory can be changed at any time, with various editing options available, e.g., add/move/remove nodes; move/resize/rotate multiple nodes; shape lines with Bezier curves. The trajectories are saved by Zirkonium to the project file. Finally, Zirkonium plays the audio output to the audio interface which is then sent to the sound dome.

At the end of the composition process, once the sound design has been completed within the DAW and the trajectories have been finalized, all stems can be exported from the DAW as individual, rendered sound files and imported into Zirkonium. Zirkonium can then be used as a standalone application to play back the work; the DAW is no longer required.

8.3.2 Live Spatialization

Live spatialization is meant here in the sense that, unlike DAW spatialization described above, there are no graphical trajectories edited within Zirkonium itself. Typically, movement data is generated by external software and sent to Zirkonium via OSC messages.

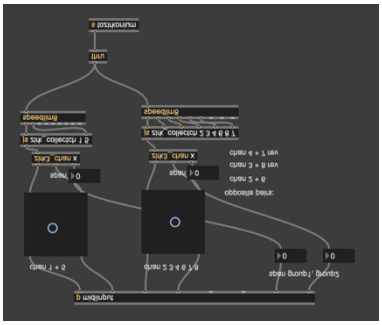


Figure 7. Zirkonium live control example Max patch. 2024

An example of manual live control would be a MIDI fader box connected to software such as Max. One fader could control the horizontal position of a sound source, for example, and a second fader could control the vertical position. The fader movement is then read by Max, formatted as corresponding OSC messages, and sent to Zirkonium. See Figure 7 for an example.

An initutive and convenient option is offered by the iOS app, ZirkPad (2016) [10], which was developed by Chikashi Miyama as a supplement to Zirkonium 3. Here, Zirkonium can be remotely controlled via OSC from an iPad with multi-touch handling. See Figure 8.

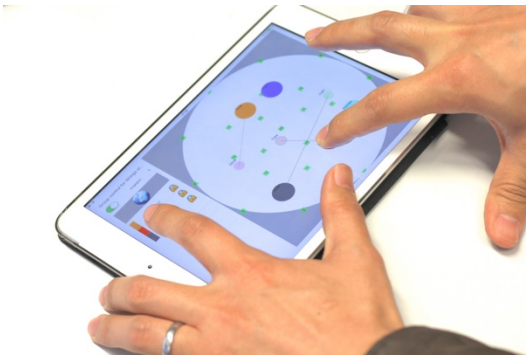


Figure 8. Zirkonium live control ZirkPad iOS app. 2014

Similarly, a spectacular example from the early years of Zirkonium 1 was the multi-touch device "Lemur", which allowed several sound sources to be moved simultaneously and independently of each other[7] — at least as far as the performer's finger dexterity allowed.

Manual virtuosity can, of course, be supported by algorithms, and Zirkonium can also be controlled entirely algorithmically via OSC. There are endless possibilities for connecting external sources. Guest artists at the ZKM are welcome to create custom solutions or to draw upon ready-made examples. One such example is a collection of Max and Max4Live patches by sound engineer Ben Miller (2024)[8], which guest artists can use directly or as a starting point for further development.

A recent external software project interesting for use with Zirkonium is Grapes 3D[9] (2024), which provides a graphical interface, clip-based movement metaphor, and toolbox to generate a wealth of algorithmically-controlled trajectories that can be sent over OSC.

8.4 Sustainability and Compatibility

Sustainability and compatibility are important aspects of Zirkonium development. This includes composers being able to perform their pieces independently and without reliance on ZKM, Klangdom, or even Zirkonium, as mentioned above. This also means that pieces can be stored and performed again in an uncomplicated manner, even in the distant future.

[7] https://en.wikipedia.org/wiki/Lemur_(input_device)
[8] https://zkm.de/de/zirkonium
[9] https://grapes-3d.com

8.4.1 Recording of Trajectories

A particularly important feature of Zirkonium versions 2 and 3 is the ability to record trajectories that are generated externally and sent to Zirkonium via OSC. This applies to all scenarios described above in the section "Live Spatialization", whose movement data can be recorded and stored together with the corresponding audio data as object-oriented audio.

This approach is especially suited for projects which work with DAW automation and which otherwise could not be stored in a future-proof manner as they would remain dependent on the specific DAW software used. Furthermore, it is always interesting and valuable to make the infinite possibilities of manually or algorithmically controlled trajectories available for DAW spatialization through recording. Live spatialization can thus become a useful tool for DAW composition.

Another conceivable application would be the recording of spatialization data from live performances for documentation purposes. This is possible in Zirkonium 3, for example, using an externally triggered "live bounce."

8.4.2 Non-native Klangdom Pieces

As previously mentioned, concert practice at the ZKM means that works originally composed for other loudspeaker systems are repeatedly performed in the Klangdom.

In the case of DAW compositions, the ideal strategy is not to use the final mix, but to go back one or more steps, if possible, and use tracks or stems of the DAW session. These tracks or stems can be imported into Zirkonium and freely positioned and moved in the Klangdom.

The most difficult pieces to adapt to the Klangdom are usually those designed for a stereo system. In this case, you typically do not want to lose the work that went into designing the stereo panorama, which sometimes limits your options for positioning and moving the stems in the Klangdom.

Pieces that were originally composed for a circular speaker system, e.g., for four speakers in a quad setup or for an 8-channel circle, work more effectively. Here, the four- or eight-channel stems can be placed as virtual speaker circles in the sound dome, e.g., at different heights, thus taking advantage of the additional dimension of the sound dome. The virtual speaker circles can also be moved.

Even if we only have the final mix, a composition can benefit from additional movement in the Klangdom, such as that gently introduced by a live performer. As an example, see the video recording of a performance of Daniel Blinkhorn's 2014 composition *frostbYte -- wildflower*, for which the final mix is available in 5.0-channel format. The five channels were arranged in a circle within Zirkonium, and the performer rotated the circle and moved it up and down in real time using a fader box[10].

8.4.3 Archiving

At ZKM Hertzlab, we are pursuing several different strategies in parallel to ensure that Klangdom works remain future-proof.

First, we are attempting to establish backward compatibility so that newer versions of Zirkonium can also play older Zirkonium works. This is of course the ideal solution, as long as Zirkonium remains available.

Second, we are working on creating a human-readable XML format in which the movement data of a work can be exported and imported. This means that the spatialization of a composition can be reconstructed at any time, albeit possibly with some effort, even in the distant future, when Zirkonium may no longer exist.

Third, we are creating channel-based versions of the Klangdom works, which we are storing permanently in our archive. We proceed as follows: first, decide on a specific dome speaker setup, e.g., a 24-channel setup. Next, the speaker setup is loaded into Zirkonium together with the sound dome work. Now, with just a few mouse clicks, the Klangdom composition can be bounced as a channel-based version for this specific speaker configuration. This channel-based version could now be performed without effort on any other sound dome in such a way that the 24 speakers of the original setup are placed as virtual sources in the target setup. Zirkonium is not required for this; any other software that can pan sounds on the surface of a sound dome can be used instead. The natural disadvantage of this method is that the spatial resolution of works is limited by the specified number of speakers, 24 in this example case.

9. FUTURE

As with its past, the future of Zirkonium is bright. The intuitive nature of the project's parameter and visual metaphors allows for an accessible approach to spatialization that has allowed for the creation of numerous new works at ZKM over the years. Quite a few avenues of new exploration remain, the first of which is expanding the community-aspects of the project. Zirkonium has always been “freeware” and we intend to fully open source the software in 2026, hopefully in time for the *next_generation* festival in the summer at ZKM. It is hoped that users and institutions will be able work with us to shape how the project moves forward.

More concretely, goals for a future roadmap could include a universal data format, consumer spatial audio support, and a Zirkonium 4 modernization. Currently, Zirkonium projects are stored in an Apple CoreData-specific XML schema. In the spirit of the SpatDIF library, internal work in this area has already implemented a more universal, human-readable XML description for both import and export to allow for better support for archiving as well as project file creation outside of the macOS application itself. Next, consumer audio formats such as Dolby Atmos [13] allow for source placement, movement, and listener orientation, such as dynamic head tracking on Apple devices. The import of Atmos spatial source files and render/bounce to these formats would expand the potential usage and audience for Klangdom pieces. Last, the Zirkonium3 codebase has been updated to some degree since 2015 however relies on a few deprecated APIs

[10] https://zkm.de/de/media/videos/daniel-blinkhornfrostbyte-wildflower2014

such as OpenGL and a transition from Obj-C to Swift and SceneKit in a "Zirkonium 4" could bring certain advantages such as a smaller, easier to maintain project.

Most excitingly, cross-platform support already exists in the form of the Pure Data spatialization server engine[11]. In 2021, a minimal C wrapper called *zirkon* was created using libpd as a sort of headless, "mini Zirkonium" in order to allow for continual looped playback of a Zirkonium piece within an 8-channel 3D live-controlled interactive media installation called "Agents" for the *BioMedia* exhibition at ZKM [14].

Although created for a specific use case, this wrapper has been designed around the beginnings of a "libzirkonium" which encapsulates the project file format and conceptual relationships (sources, IDs, groups, events). This library could be extended to include the sequencing and path interpolation egnines, then used as the core for a Zirkonium 4 modernization, validating accuracy with A/B testing. Once ready, a libzirkonium can be used as the core of new editing and playback applications as well as within other environments and platforms: for instance, as a commandline player ala *zirkon*, a Max or Unity plugin, or an interactive 3D visualization.

9. CONCLUSIONS

20 years of practical experimentation at ZKM have resulted in numerous sound compositions and works realized both for the Klangdom as well as external systems. Zirkonium the project and the software are both a tool and ongoing experiment for intuitive approaches for practical spatial composition, informed by both internal and external users. Even in the face of the proliferation of newer spatial plugins and systems, Zirkonium's parameter and visual path-based event sequencing allows artists and composers a still unique perspective. As with the move to open-source the project, expanding the possibilities for creation and usage of Zirkonium projects to new platforms and outlets leads to exciting possibilities.

Zirkonium is available for download at `https://zkm.de/zirkonium`

Acknowledgments

Many thanks to sound engineer Ben Miller for his invaluable input and many fruitful discussions for this paper, to Jens Barth for his valuable enhancements of the Zirkonium 1 user interface, and to numerous others, both internal and external, over the years. The contribution of Ludger Brümmers students at the HfM Trossingen as testers throughout development cycles is especially noted.

REFERENCES

[1] Ludger Brümmer. Glasharfe, multichannel, 19'04'' [Online] Available: `https://open.spotify.com/track/54yotJ8qyJCSo9iiffGdcb?si=7d4faf97ef6747dd`

[2] V. Pulkki, & M. Karjalainen, "Directional quality of 3-D amplitude panned virtual sources." Georgia Institute of Technology. 2001

[3] C. Ramakrishnan, J. Goßmann, and L. Brümmer, "The ZKM klangdom," in *Proceedings of the 2006 Conference on New Interfaces for Musical Expression*, 2006, pp. 140-143.

[4] C. Ramakrishnan. "Zirkonium: Non-invasive software for sound spatialisation." *Organised Sound* 14.3, 2009, pp. 268-276.

[5] D. Wagner, L. Brümmer, G. Dipper, and J.A. Otto. "Introducing the Zirkonium MK2 system for spatial composition," in *Proceedings of the 40th International Computer Music Conference (ICMC)*, 2014.

[6] R. Normandeau, O. Bélanger, C. Lengelé, & D. Ledoux. "SPATGRIS/SERVERGRIS, Creative Tools for 2D and 3D Sound Spatialization," in *Proceedings of the 44nd International Computer Music Conference (ICMC)*, 2018, pp 291-297.

[7] P. Brinkmann, P. Kirn, R. Lawler, C. McCormick, M. Roth, and H.C. Steiner. "Embedding pure data with libpd," in *Proceedings of the Pure Data Convention,* 2011, vol. 291.

[8] C. Miyama, G. Dipper, and L. Brümmer. "Zirkonium MK III - A toolkit for spatial composition," in *Journal of the Japanese Society for Sonic Arts* 7.3, 2015, pp. 54-59.

[9] N. Peters, T. Lossius, and J.C. Schacher. "SpatDIF: Principles, specification, and examples," in *Proceedings of the 9th Sound and Music Computing Conference*, 2012.

[10] C. Miyama, G. Dipper, G., L. Brümmer, "Zirkonium 3.1 - a toolkit for spatial composition and performance," in *Proceedings of the 42nd International Computer Music Conference (ICMC)*, 2016.

[11] S. Letz, D. Fober, Y. Orlarey, and P. Davis. "Jack Audio Server: MacOSX port and multi-processor version," in *Journées d'Informatique Musicale-Sound and Music Computing*, 2004.

[12] T. Carpentier. "Panoramix: 3D mixing and post-production workstation," in *Proceedings of the 42nd International Computer Music Conference (ICMC)*, 2016.

[13] Dolby Laboratories Inc. Dolby Atmos—cinema technical guidelines, [Online]. Available: `https://www.dolby.com/us/en/technologies/cinema/dolby-atmos.html`

[14] ZKM Karlsruhe. Ludger Brümmer, Yasha Jain | Agents 2021 [Online]. Available: `https://www.youtube.com/watch?v=LOY7K_Fr9KY`

[11] https://github.com/zkmkarlsruhe/ZirkoniumSpatializationServer

Composer-in-the-Loop Generation of Motivic Variants Using State-Space Models and Preference Learning

Rodrigo F. Cádiz
Music Institute and Department of Electrical Engineering
Pontificia Universidad Católica de Chile
rcadiz@uc.cl

ABSTRACT

Most current approaches to symbolic music generation rely on large-scale deep learning models trained on massive corpora and operate exclusively on pitch and duration, disregarding the articulations and dynamics that are fundamental to musical expression. We present a composer-in-the-loop system that addresses both limitations. A pre-composed motive, complete with pitch, rhythm, articulation, and dynamics, is modeled as a reference trajectory in a musically interpretable state space, and variants are generated by sampling structured stochastic deviations inspired by Kalman filtering. A neural network modulates variance and structural edit probabilities based on composer preference, learning from the composer's own selections rather than from external data. Implemented as a simple browser-based application, the system supports real-time audition and persistent model reuse. The approach represents a first step toward a compositional workflow in which larger musical structures are built by concatenating and varying short, fully expressive motivic ideas.

1. INTRODUCTION

Motivic variation is one of the most fundamental compositional techniques in Western art music, underpinning practices from Baroque invention to twentieth-century developing variation. Variations must preserve perceptual identity while introducing controlled transformation across pitch, rhythm, articulation, and dynamics, a balance that remains difficult to formalize algorithmically.

Two pervasive shortcomings in current computational approaches motivate this work. First, the dominant paradigm in symbolic music generation relies on deep neural networks trained on massive corpora [1, 2, 3]. Such models are data-hungry by design: they require thousands of scores to learn style-specific patterns, making them impractical for professional composers who seek personalized, responsive tools rather than imitations of an aggregated style. Second, the overwhelming majority of these systems operate on MIDI-level representations, sequences of pitches and durations, while ignoring the articulations and dynamics that are fundamental to what written music actually *is*. A melody notated with staccato, crescendo, and phrasing is a qualitatively different musical object from a flat list of note-on/note-off events. Any generative system that aspires to serve composers must treat these expressive dimensions as first-class parameters, not as optional metadata to be added after the fact.

Earlier algorithmic approaches based on rule-based transformations, stochastic grammars, and Markovian processes offer interpretability but struggle with expressive nuance [4, 5]. Probabilistic state-space models, including Kalman filtering and related Bayesian techniques, have been applied successfully to music transcription and temporal modeling [6, 7], yet they have rarely been explored as generative tools for composition. Interactive and preference-driven systems using Bayesian optimization or evolutionary computation acknowledge the subjective nature of musical aesthetics [8, 9, 10], but they are typically built on opaque latent spaces without explicit music-theoretical structure.

This paper proposes a framework that addresses both the data and representation problems while unifying three research traditions: probabilistic state-space modeling, neural preference learning, and interactive motif variation. We introduce a browser-based system in which melodic motives, complete with articulation and dynamics, are represented as intervallic state trajectories and evolved through a Kalman-filter-inspired stochastic process. A lightweight neural network learns from the composer's own selections rather than from external training data, adapting variance parameters and structural edit probabilities to shape the space of variation. The result is a system that treats composition as a problem of control under uncertainty, requiring zero pre-existing data and producing musically rich output from the very first interaction.

The primary contribution is a novel integration of Kalman filtering principles with neural preference learning for symbolic motivic variation encompassing pitch, rhythm, articulation, and dynamics on equal footing. Neural networks modulate the generative process through user feedback rather than replacing it, yielding a compositional tool that is interpretable, interactive, and grounded in control-theoretic principles. More broadly, this work represents a first step toward a compositional workflow in which larger musical structures are assembled by concatenating and varying short, fully expressive motivic ideas.

2. STATE-SPACE PROBLEM FORMULATION

This work formulates motivic variation as a problem of controlled stochastic evolution in a musical state space.

Copyright: ©2026 Rodrigo F. Cádiz. This is an open-access article distributed under the terms of the Creative Commons Attribution License 3.0 Unported, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Rather than treating melody as a static symbolic sequence, we model it as a trajectory of musically meaningful parameters evolving over discrete events. This perspective is inspired by state-space modeling and Kalman filtering, which provide a principled framework for describing dynamical systems subject to uncertainty.

2.1 State-Space Models and Kalman Filtering

A state-space model describes a system in terms of an internal state vector that evolves over time according to a dynamical process. In its general form, a linear stochastic state-space model is defined as

$$\mathbf{x}_k = \mathbf{A}_k \mathbf{x}_{k-1} + \mathbf{w}_k, \tag{1}$$

$$\mathbf{y}_k = \mathbf{C}_k \mathbf{x}_k + \mathbf{v}_k, \tag{2}$$

where $\mathbf{x}_k$ is the latent state at time step k, $\mathbf{y}_k$ is an observation, and $\mathbf{w}_k$ and $\mathbf{v}_k$ are process and observation noise, typically modeled as Gaussian.

The Kalman filter provides an optimal recursive solution for estimating the state of such systems under Gaussian assumptions. In music research, Kalman filtering has been widely used for estimation tasks such as tempo tracking, expressive timing analysis, and transcription, where musical structure is inferred from noisy observations.

In this work, we adopt the state-space formalism not for estimation, but for generation. The composed melody acts as a reference trajectory, and new melodic variants are generated by sampling controlled stochastic deviations around this reference. The Kalman framework is particularly suitable for this purpose because it explicitly separates structure (the reference trajectory) from uncertainty (the permissible deviation), mirroring a fundamental aspect of compositional variation.

2.2 Motivic Variation as Trajectory Deviation

Motivic variation in composition rarely consists of arbitrary transformation. Instead, it involves systematic deviation from an identifiable idea: a motive is stretched, ornamented, displaced, or dynamically reshaped while remaining perceptually coherent. This suggests modeling variation as a continuous deformation of an underlying trajectory rather than as a symbolic rewrite.

By framing a melody as a trajectory in a state space, we can interpret variation as movement within a neighborhood of that trajectory. The size and shape of this neighborhood determine how conservative or exploratory a variation may be. This conception aligns naturally with control-theoretic notions of reference tracking and stochastic disturbance.

2.3 Musical State Definition

Each melodic event is represented by a state vector

$$\mathbf{x}_k = \begin{bmatrix} \Delta p_k \\ \log d_k \\ g_k \\ v_k \end{bmatrix}, \tag{3}$$

where Δp_k is the pitch interval between events $k-1$ and k (in semitones), d_k is the event duration, g_k is a continuous articulation or gate parameter and v_k is a continuous loudness parameter.

Pitch is represented in terms of intervals rather than absolute values in order to avoid constraining generated variants to the original melodic contour. Interval-based representations preserve relative motion while allowing global transformations such as transposition and contour inversion.

Durations are modeled in logarithmic space to ensure positivity and to support multiplicative rhythmic variation. Articulation and dynamics are treated as continuous parameters that can later be mapped to discrete notational symbols.

2.4 Reference Trajectory and Reconstruction

From the original motive, we extract a reference state trajectory

$$\{\mathbf{x}_k^*\}_{k=0}^{N-1}.$$

Absolute pitch is reconstructed by accumulation:

$$p_0 = p_0^* + t, \qquad p_k = p_{k-1} + \Delta p_k, \tag{4}$$

where t is a global transposition parameter sampled once per variant. This formulation allows the same intervallic structure to be realized in different registral contexts.

2.5 Stochastic State Evolution

Melodic variants are generated by sampling stochastic deviations around the reference trajectory:

$$\mathbf{x}_k = \mathbf{x}_k^* + \boldsymbol{\epsilon}_k, \qquad \boldsymbol{\epsilon}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k). \tag{5}$$

Here, $\mathbf{Q}_k$ is a covariance matrix that controls the extent of variation permitted at each event. Small covariance values yield conservative variants closely aligned with the original motive, while larger values produce more exploratory deviations.

This formulation parallels the process noise term in traditional Kalman filtering, but its role is generative rather than inferential. Instead of compensating for uncertainty in observations, the noise explicitly defines the creative latitude of the system.

2.6 Why a State-Space Formulation?

The state-space formulation provides local per-event control, smooth stochastic trajectories that preserve gestural coherence, direct interpretability (each dimension is a musically meaningful parameter), and a clean separation between structure (the reference trajectory) and creativity (the controlled uncertainty). These properties make it particularly well suited to composer-in-the-loop systems, where musical intent and exploratory variation must coexist.

3. ADAPTIVE VARIATION AND PREFERENCE LEARNING

While the state-space formulation defines how melodic variants are generated as stochastic deviations from a reference trajectory, additional mechanisms are required to shape this process adaptively, allow structural reconfiguration, and incorporate composer feedback. This section describes how uncertainty is modulated, how variable-length variants are produced, how expressive notation is derived, and how preference learning closes the interactive loop.

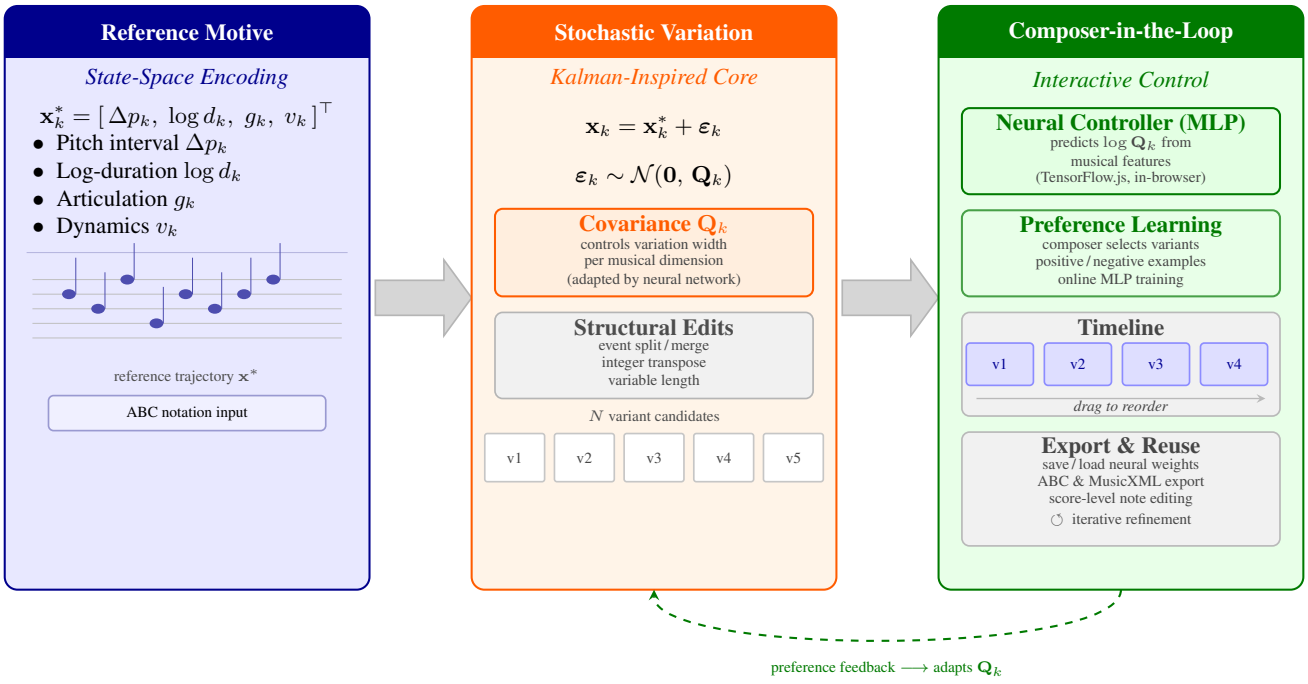


Figure 1. Conceptual overview of the proposed composer-in-the-loop system. A reference motive is encoded as an interval-based state-space trajectory $\mathbf{x}_k^*$, where each state vector $\mathbf{x}_k$ represents the musical parameters of event k, including pitch interval, duration, articulation, and dynamics. Motivic variants are generated through Kalman-inspired stochastic deviation, $\mathbf{x}_k = \mathbf{x}_k^* + \boldsymbol{\varepsilon}_k$, where $\boldsymbol{\varepsilon}_k \sim \mathcal{N}(0, \mathbf{Q}_k)$ is a zero-mean Gaussian perturbation defining the permissible space of variation. A lightweight neural controller adapts the covariance $\mathbf{Q}_k$ based on composer preference, closing an interactive feedback loop that supports guided exploration, selective retention of variants via timeline arrangement, and iterative model reuse.

3.1 Neural Modulation of Uncertainty

The extent of permissible deviation in each state dimension is governed by the covariance matrix $\mathbf{Q}_k$. Rather than fixing this matrix a priori, we parameterize it using a neural controller that adapts to musical context and composer preference.

Specifically, the covariance is defined as

$$\mathbf{Q}_k = \mathrm{diag}\left(\exp(\boldsymbol{\ell}_k)\right), \tag{6}$$

where $\boldsymbol{\ell}_k$ is a vector of log-variances predicted by a neural network

$$\boldsymbol{\ell}_k = f_\theta(\boldsymbol{\phi}_k). \tag{7}$$

The input feature vector $\boldsymbol{\phi}_k$ is derived from the original motive and includes musically interpretable descriptors such as interval magnitude, contour sign, relative phrase position, and local rhythmic density. The network is implemented as a small multilayer perceptron and does not generate musical content directly. Instead, it modulates the degree of stochastic freedom available to the state-space sampler at each event.

This design choice preserves interpretability and ensures that learning affects *how much* the system may vary the motive rather than *what* musical material is produced.

3.2 Global Transformations

In addition to local stochastic variation, the system supports global transformations that preserve internal structure while expanding the search space. A global transposition parameter

$$t \sim \mathcal{N}(0, \sigma_t^2) \tag{8}$$

is sampled once per variant and applied uniformly to the reconstructed pitch trajectory. The variance σ_t^2 is treated as a trainable parameter, allowing the system to learn preferred registral displacement ranges from user interaction.

Optional contour inversion may be applied by negating interval states over a contiguous segment, enabling the exploration of structurally related but directionally inverted variants.

3.3 Variable-Length Edit Layer

While the state-space trajectory generates event-aligned variants, motivic variation in compositional practice often involves changes in rhythmic density through ornamentation or simplification. To support this, a discrete edit layer is applied after continuous trajectory sampling.

For each event index k, the system samples binary edit decisions:

$$z_k^{\text{split}}, z_k^{\text{merge}} \sim \mathrm{Bernoulli}(\pi_k), \tag{9}$$

where the probabilities π_k are produced by additional neural outputs.

A *split* operation replaces a single event with two shorter events whose combined duration equals the original. Pitch realization for the inserted event follows musically motivated strategies such as repetition, passing tones, or neighbor tones. A *merge* operation replaces two adjacent events with a single event whose duration is their sum.

These operators preserve total phrase duration while allowing rhythmic elaboration or reduction. By integrating edit probabilities into the same neural controller that modulates continuous variation, the system learns not only how

far to deviate from the reference trajectory, but also where structural modification is musically appropriate.

3.4 Derivation of Expressive Spans

The state-space representation treats articulation and dynamics as continuous parameters. To produce conventional symbolic notation, these trajectories are post-processed into higher-level expressive constructs.

Crescendo and decrescendo hairpins are derived from the sign and magnitude of the loudness derivative over contiguous event spans. To avoid conflicts with beamed note groups, hairpin closure tokens are placed as trailing markers on the preceding note rather than as leading tokens on the following note. Slurs are generated from sustained articulation values exceeding a threshold, and accent markings are introduced sparsely based on local prominence in the dynamic profile.

This approach allows expressive markings to emerge from continuous control data rather than being explicitly encoded in the reference motive, enabling the appearance of articulations and dynamic gestures not present in the original material.

3.5 Preference-Based Learning

The system is designed for interactive use, with the composer evaluating and selecting preferred variants. Learning is formulated as a preference-ranking problem rather than supervised regression.

Given two generated variants i and j, the probability that variant i is preferred is modeled as

$$P(i \succ j) = \sigma\left(-\mathcal{L}_i + \mathcal{L}_j\right), \tag{10}$$

where $\mathcal{L}_i$ is the negative log-likelihood of variant i under the current model and $\sigma(\cdot)$ denotes the logistic function.

The loss function minimized during interaction is

$$\mathcal{J} = -\log P(i \succ j), \tag{11}$$

which adjusts neural parameters so that preferred variants become more probable under the stochastic model.

Importantly, learning affects variance parameters and edit probabilities rather than symbolic content. This ensures that the system adapts its notion of acceptable variation while preserving the composer-defined reference structure.

Together, these mechanisms transform the state-space formulation into an adaptive, composer-in-the-loop system. Continuous stochastic variation, discrete structural edits, expressive span derivation, and preference-based learning operate in concert to support the exploratory transformation of melodic motives. The result is a framework in which compositional intent, probabilistic control, and interactive learning are tightly integrated.

4. IMPLEMENTATION

The proposed system is implemented under the title *State Space Composer*, a self-contained, browser-based application designed for interactive compositional use. All computation, including stochastic sampling, neural learning, symbolic rendering, and audio playback, occurs locally without reliance on a server. The application requires no installation or backend infrastructure, and is available for full usage at `https://rodrigocadiz.github.io/software/state-space-composer`. This design emphasizes immediacy, transparency, and iterative exploration, allowing the composer to remain continuously engaged with the generative process.

4.1 User Workflow

The typical use of the system proceeds through an iterative cycle of specification, generation, evaluation, and adaptation. The composer begins by entering an initial melodic motive in ABC notation. This motive serves as the reference trajectory for the state-space model and defines the structural identity around which variations are generated. Upon request, the system produces a set of melodic variants by sampling stochastic deviations from the reference trajectory, applying optional transformations and variable-length edits, and post-processing the result into conventional symbolic notation.

Generated variants are displayed visually as musical scores and can be auditioned immediately using an integrated Web-Audio synthesizer. A per-variant quantization toggle allows rhythm grids to be adjusted after generation without resampling. Individual notes may be edited directly by clicking on the rendered score to open a pitch and duration editing panel. The composer evaluates the variants and may select one or more preferred outcomes. These selections serve two purposes: they may be added to a timeline for sequencing and export, and they provide preference feedback that updates the neural controller.

This process can be repeated iteratively. As interaction proceeds, the system adapts its notion of acceptable variation, gradually aligning its output with the composer's aesthetic preferences while maintaining the structural constraints imposed by the reference motive.

4.2 Preference-Based Adaptation and Training

Learning in the system is driven entirely by preference selection rather than by explicit target outputs. When the composer selects a preferred variant among several alternatives, the system interprets this choice as a ranking signal. The neural controller is updated so that the selected variant becomes more probable under the stochastic model relative to the rejected alternatives.

Importantly, training does not alter the symbolic structure of the reference motive. Instead, it affects the parameters governing stochastic deviation and edit likelihoods: variance in pitch interval deviation, variance in rhythmic deformation, variance in articulation and dynamics, probabilities of split and merge operations, and the range of global transposition. This design ensures that learning shapes the *space of variation* rather than memorizing specific musical outcomes. As a result, the system remains responsive and exploratory even after extended interaction.

4.3 Timeline and Variant Management

The system provides a timeline mechanism for organizing and sequencing generated variants. Selected variants can be added to a horizontal timeline displayed at the bottom of the interface. Timeline slots may be reordered by drag-and-drop, removed individually, or cleared in bulk. The

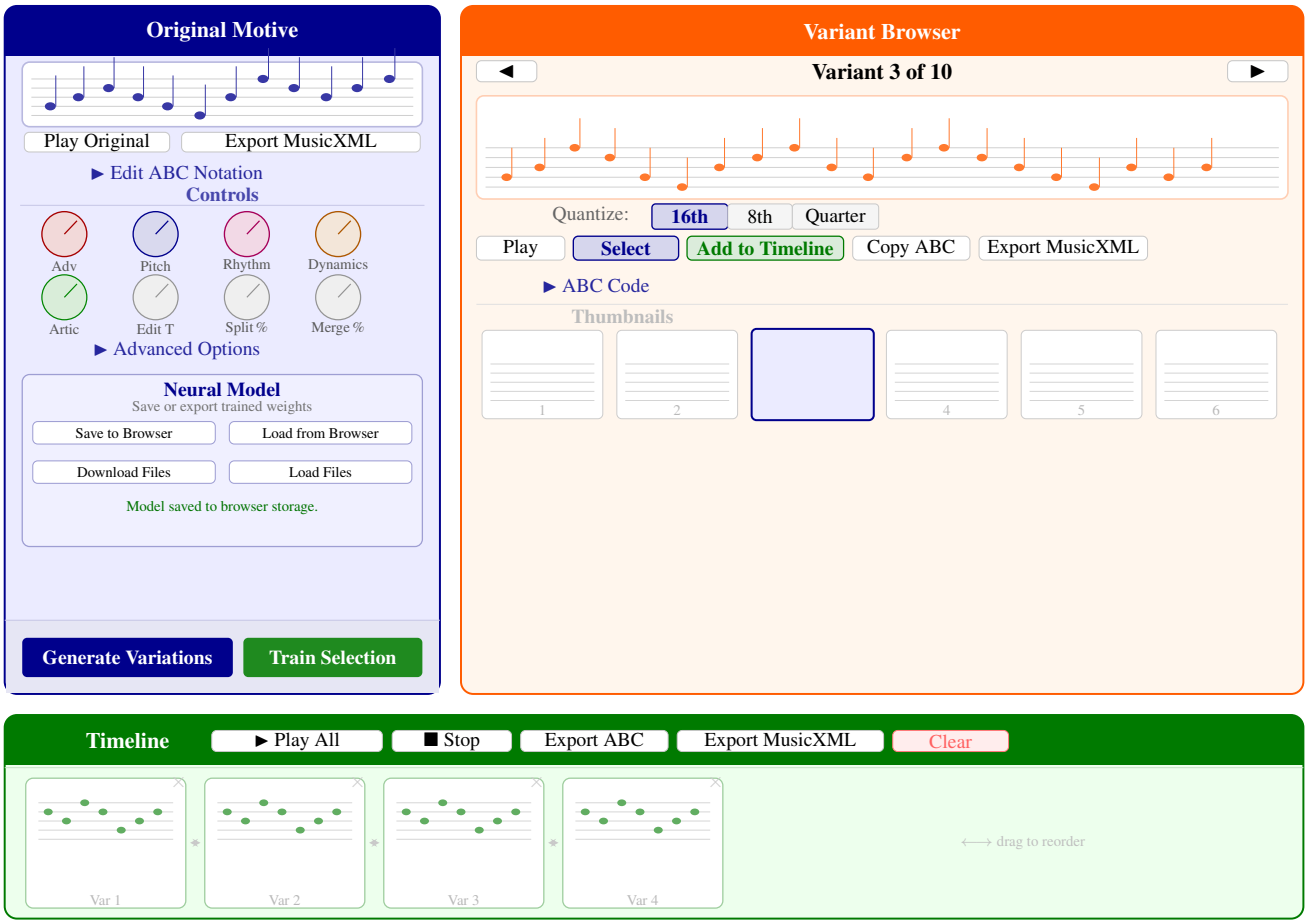


Figure 2. Schematic diagram of the *State Space Composer* interface. The left panel (blue) handles reference motive input and controls: a rendered score with note-click editing, eight rotary knobs for variation parameters, a collapsible Advanced Options section, and a permanently visible Neural Model panel with save-to-browser and file export/import actions. The right panel (orange) is the variant browser: a carousel displaying one generated variant at a time with quantization toggle, action buttons (*Select*, *Add to Timeline*, etc.), and a thumbnail strip for navigation. The bottom bar (green) is the horizontal timeline: generated variants can be dragged into ordered slots, played back sequentially, and exported as ABC or MusicXML. Live application available at `https://rodrigocadiz.github.io/software/state-space-composer`.

full sequence can be played back from left to right using sequential audio rendering, and the timeline can be exported as a concatenated ABC file or as MusicXML for import into notation software.

The timeline replaces the notion of an unordered save list with a temporally ordered compositional sketch, allowing the composer to evaluate the flow and continuity of a developing piece. Variants include their complete ABC notation and are rendered as thumbnail scores in each timeline slot.

4.4 Score-Level Note Editing

To support fine-grained correction and adjustment after generation, the system includes a lightweight note editor. Clicking any rendered note in either the reference motive or a generated variant opens a floating property panel. The panel displays the current pitch (as a note name and octave) and allows the composer to shift it up or down by semitone increments using dedicated buttons. Duration can be changed via a dropdown menu covering standard rhythmic values from thirty-second notes to whole notes. Changes are applied to the underlying ABC string immediately, and the score is re-rendered to reflect the edit. This interaction follows the Tier A graphical editing model described in the design rationale: click-to-select with a property panel, without requiring a full drag-and-drop notation editor.

4.5 Model Persistence and Reuse

To support long-term use and iterative refinement, the system allows the neural controller to be saved and restored. A dedicated *Neural Model* panel, visible at all times in the left column, provides four actions: saving weights to the browser's local storage (persistent across page reloads within the same browser), loading them back, downloading the weights as a pair of TensorFlow.js-compatible files (`.json` metadata and `.bin` weight tensors), and loading previously downloaded files. This design makes the two storage tiers explicit: local storage for convenience within a single device, and file export for backup, cross-device transfer, or sharing with collaborators.

This functionality allows composers to develop personalized models over time, experiment with multiple aesthetic profiles, and share trained controllers across projects or devices.

4.6 Symbolic Rendering and Audio Feedback

Each generated variant is rendered into ABC notation, which serves both as a symbolic output and as an intermediate

representation for audio playback. Continuous state variables are mapped to discrete musical symbols, including pitch, duration, dynamics, articulations, slurs, and hairpins.

Audio feedback is provided via a Web Audio API synthesizer that renders parsed note events directly, without external soundfont dependencies. Each note is produced by a multi-oscillator piano voice consisting of five layered partials: a fundamental sine, an octave harmonic, a fifth harmonic, a two-octave harmonic, and a triangle for body warmth, with a percussive amplitude envelope (short attack, exponential decay to sustain, release) and a brief band-pass filtered noise burst simulating hammer transient. Higher harmonics are progressively detuned to model string inharmonicity. Velocity sensitivity maps to both amplitude and harmonic brightness. This approach produces a piano-like timbre that provides meaningful feedback for pitch, rhythm, contour, and dynamics assessment.

4.7 Exposed Parameters

The application exposes a set of parameters that allow the composer to control the behavior of the generative process. These parameters are designed to be musically interpretable and to map directly to aspects of the underlying model.

Lower temperature values yield conservative variations closely aligned with the reference motive, while higher values encourage exploratory transformation. The separation of pitch, rhythm, articulation, and dynamics into distinct controls allows composers to focus variation on specific musical dimensions.

4.8 Design Considerations

Several design decisions reflect compositional priorities. First, all learning and generation of variants occur locally in the browser, ensuring immediacy. Second, neural networks are deliberately lightweight and interpretable, modulating uncertainty rather than generating musical symbols directly. Finally, the use of ABC notation [11] provides a compact, human-readable representation that bridges algorithmic processing and conventional musical notation. To lower the barrier for users unfamiliar with ABC syntax, the application includes an embedded quick-reference panel accessible from the interface header.

Together, these choices position the system as a compositional tool rather than an autonomous generative agent, supporting iterative exploration while preserving authorial intent.

5. DISCUSSION

5.1 Control Theory as a Compositional Paradigm

By modeling melodic variation as stochastic deviation from a reference trajectory, the proposed system introduces a control-theoretic perspective into symbolic composition. In this paradigm, the composed motive defines a desired trajectory, while uncertainty governs the range of permissible variation. This separation closely mirrors compositional practice, where a musical idea is retained as an underlying structure even as it undergoes transformation.

Unlike rule-based or grammar-driven systems, which often encode transformation explicitly, the state-space formulation allows variation to emerge implicitly from controlled noise. This enables smooth, continuous deformation across multiple musical dimensions simultaneously, including pitch, rhythm, articulation, and dynamics. Such continuity is typically absent from sequence-based neural generators.

The use of Kalman-inspired models is particularly significant in this context. While Kalman filtering has traditionally been applied to estimation problems in music research, its reinterpretation as a generative mechanism highlights its potential as a creative model. Process noise, rather than compensating for uncertainty in observation, becomes an explicit compositional parameter that defines expressive latitude. This reframing suggests broader opportunities for control-theoretic methods in creative domains beyond their conventional analytical applications.

5.2 Interpretability, Expressiveness, and Musical Meaning

A central motivation of this work is the rejection of the reductive view of symbolic music as sequences of pitches and durations. In the proposed system, each dimension of the state vector corresponds to a musically meaningful parameter, namely pitch interval, rhythm, articulation, and dynamics, and each user-adjustable control has a clear conceptual interpretation. Articulations and dynamics are not post-hoc annotations but first-class components of the generative process, varied simultaneously and on equal footing with pitch and rhythm. This stands in contrast both to MIDI-centric generative models, which typically ignore expressive parameters entirely, and to latent-space based systems, where musical meaning is distributed across high-dimensional representations that are difficult to interrogate or manipulate directly.

The interval-based pitch representation further reinforces this interpretability. By focusing on relative motion rather than absolute pitch, the system supports transformations such as transposition and contour inversion without compromising motivic identity. This design choice aligns with music-theoretical notions of intervallic structure and allows the system to generate variants that are perceptually related despite substantial surface-level change.

Similarly, the variable-length edit layer provides a computational analogue to ornamentation and rhythmic reduction. Rather than imposing a fixed event structure, the system allows rhythmic density to fluctuate in response to learned preferences, resulting in variants that differ not only expressively but structurally. This flexibility distinguishes the approach from many symbolic systems that assume fixed-length representations.

5.3 Role of Neural Networks

The neural component of the system plays a deliberately constrained role. Rather than generating musical symbols, the neural network modulates variance parameters and edit probabilities. Learning thus affects the shape of the variation space rather than the content of individual realizations.

This design choice addresses several concerns raised in the literature regarding deep generative models for mu-

Table 1. User-adjustable parameters and recommended ranges.

Parameter	Description	Typical Range
Adventure (Chaos) [1]	Global scaling of process noise (quadratic above 1.0)	0.0 – 10.0
Pitch Temperature	Scales variance of pitch interval deviation	0.5 – 10.0
Rhythm Temperature	Scales variance of duration deformation	0.5 – 8.0
Dynamics Temperature	Scales variance of loudness trajectory	0.5 – 6.0
Articulation Temperature	Scales variance of gate/legato behavior	0.5 – 4.0
Edit Temperature	Sharpens or softens split/merge decisions	0.5 – 1.5
Split Probability	Global tendency to ornament by splitting events	0.1 – 0.7
Merge Probability	Global tendency to simplify by merging events	0.05 – 0.5
Contour Inversion Probability	Probability of inverting interval signs	0.0 – 0.5
Transposition Variance	Variance of global pitch transposition	0 – 12 semitones
Dynamic Range Expansion	Enables extended dynamics (ppp–fff)	on/off
Number of Variants	Number of alternatives generated per cycle	2 – 50

sic. First, it avoids the opacity associated with large end-to-end networks. Second, it eliminates the need for pre-existing training data entirely: the system produces musically meaningful output from the very first interaction, and the neural controller adapts exclusively from the composer's own selections. This is a fundamental departure from data-hungry deep generative models that require thousands of scores before they can be used. Third, it reduces the risk of stylistic overfitting or memorization, as the reference motive remains the primary source of musical structure.

From a compositional standpoint, this approach aligns with the idea of cultivating a personal variation practice rather than training a model to reproduce a style. The neural controller gradually adapts to the composer's selections, shaping how adventurous or conservative the system behaves across different musical dimensions.

5.4 Composer-in-the-Loop Interaction

The interactive design of the system situates the composer as an integral component of the generative process. Instead of providing a single output, the system presents a set of alternatives, encouraging comparative evaluation and aesthetic judgment. This mode of interaction resonates with established compositional workflows, where ideas are refined through selection, rejection, and iteration.

The timeline mechanism further supports this process by allowing variants to be accumulated, reordered by drag-and-drop, and auditioned as a continuous sequence. In this sense, the system functions less as a generator of finished material and more as a sketching environment that externalizes and extends the composer's exploratory thinking.

Importantly, preference-based learning operates incrementally and transparently. The composer is not required to define explicit objective functions or target outputs; instead, aesthetic criteria remain implicit and embodied in selection behavior. This approach acknowledges the subjective and context-dependent nature of musical judgment while still enabling adaptive computational behavior.

5.5 Toward Motive-Driven Composition

The current system generates variants of a single motive in isolation, but it also suggests a broader workflow in which a composer builds a library of short, fully expressive motives and creates larger forms by concatenating, juxtaposing, and layering their variations over time. In this workflow, each motive has its own trained preference model, and composition becomes a process of assembling and moving between variation spaces. This positions the present work as a first step toward a motive-driven, bottom-up approach to computer-assisted composition.

Although the current system uses conventional pitch and rhythmic notation, the state-space formulation is not limited to tonal or Western music. The state vector could include spectral, timbral, or spatial parameters, and the reference trajectory could represent microtonal scales, non-retrogradable rhythms, or extended techniques. Future work could apply this probabilistic framework to these dimensions, extending it to experimental and non-Western compositional contexts.

5.6 Limitations

Despite its flexibility, the system has several limitations. The current implementation focuses on monophonic melodic material and does not model harmonic context explicitly. While the state-space framework is extensible, additional design work would be required to accommodate polyphony, voice-leading constraints, or harmonic progression.

Moreover, the expressive mapping from continuous state variables to discrete notation is necessarily heuristic. While sufficient for exploratory composition, more sophisticated models of performance practice could improve the realism and nuance of generated markings. Finally, the evaluation presented here is qualitative and compositional in nature; systematic user studies would be valuable for assessing long-term creative impact.

6. CONCLUSION

This work introduces a compositional framework that combines control-theoretic modeling, symbolic representation, neural preference learning, and interactive design to support motivic variation. It treats a composed melody as a reference trajectory in a musically interpretable state space, jointly describing pitch, rhythm, articulation, and dynamics, to generate structured yet flexible variants that preserve identity while enabling exploration. Unlike data-driven generative models, it needs no training corpus and learns solely from the composer's aesthetic judgments.

The use of Kalman-inspired state-space models provides a principled foundation for this interaction. By reinterpreting process noise as creative latitude, the system establishes a direct connection between probabilistic control theory and musical variation, in a context where continuity, coherence, and local control are essential. Neural networks play a deliberately constrained role: modulating the space of variation rather than generating musical content, preserving transparency and authorial control.

From a practical standpoint, the browser-based implementation demonstrates that such models can be deployed as accessible tools with no server infrastructure. Real-time auditioning, model persistence, symbolic export, and timeline-based arrangement support sustained creative engagement rather than isolated experimentation.

Looking ahead, the system is an initial step toward a much broader compositional paradigm in which composers work with short, fully expressive motives and build larger structures by concatenating and varying them over time. Natural next steps include extending the state-space representation to harmonic, timbral, or spatial dimensions, supporting voicing and polyphony, and conducting empirical studies with composers. Ultimately, this work advances a view of algorithmic composition grounded in control, interaction, and interpretability, one that augments rather than replaces the composer's agency.

Acknowledgments

This research was funded by ANID Fondecyt 1230926 and ANID Núcleo Milenio Animupa NCS2025_12. Anthropic's Claude 3.5 Sonnet and Google's Gemini 1.5 Pro were used to assist with computer coding, language editing, grammar improvement, and writing style refinement. The authors reviewed and edited the final content and take full responsibility for the article's content.

7. REFERENCES

[1] C. Walder, "Modelling Symbolic Music: Beyond the Piano Roll," in *Proceedings of The 8th Asian Conference on Machine Learning, ACML 2016, Hamilton, New Zealand, November 16-18, 2016*, ser. JMLR Workshop and Conference Proceedings, R. J. Durrant and K. Kim, Eds. JMLR.org, 2016, pp. 174–189. [Online]. Available: http://proceedings.mlr.press/v63/walder88.html

[2] H. Dong, W. Hsiao, L. Yang, and Y. Yang, "MuseGAN: Multi-track Sequential Generative Adversarial Networks for Symbolic Music Generation and Accompaniment," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, S. A. McIlraith and K. Q. Weinberger, Eds. AAAI Press, 2018, pp. 34–41. [Online]. Available: https://doi.org/10.1609/aaai.v32i1.11312

[3] G. Hadjeres and F. Nielsen, "Anticipation-RNN: enforcing unary constraints in sequence generation, with application to interactive music generation," *Neural Comput. Appl.*, vol. 32, no. 4, pp. 995–1005, 2020. [Online]. Available: https://doi.org/10.1007/s00521-018-3868-4

[4] J. D. Fernández and F. J. Vico, "AI Methods in Algorithmic Composition: A Comprehensive Survey," *J. Artif. Intell. Res.*, vol. 48, pp. 513–582, 2013. [Online]. Available: https://doi.org/10.1613/jair.3908

[5] C. Liu and C. Ting, "Computational Intelligence in Music Composition: A Survey," *IEEE Trans. Emerg. Top. Comput. Intell.*, vol. 1, no. 1, pp. 2–15, 2017. [Online]. Available: https://doi.org/10.1109/TETCI.2016.2642200

[6] A. T. Cemgil, H. J. Kappen, and D. Barber, "A generative model for music transcription," *IEEE Trans. Speech Audio Process.*, vol. 14, no. 2, pp. 679–694, 2006. [Online]. Available: https://doi.org/10.1109/TSA.2005.852985

[7] C. H. Koneputugodage, R. Healy, S. Lamont, I. Mallett, M. Brown, M. Walters, U. Attanayake, L. Zhang, R. T. Dean, A. Hunter, C. Gretton, and C. Walder, "Computer Assisted Composition in Continuous Time," *CoRR*, vol. abs/1909.05030, 2019. [Online]. Available: http://arxiv.org/abs/1909.05030

[8] Y. Zhou, Y. Koyama, M. Goto, and T. Igarashi, "Generative Melody Composition with Human-in-the-Loop Bayesian Optimization," in *Proceedings of the 2020 Joint Conference on AI Music Creativity (CSMC-MuMe)*, 2020. [Online]. Available: https://arxiv.org/abs/2010.03190

[9] N. Masuda and H. Iba, "Musical Composition by Interactive Evolutionary Computation and Latent Space Modeling," in *IEEE International Conference on Systems, Man, and Cybernetics, SMC 2018, Miyazaki, Japan, October 7-10, 2018*. IEEE, 2018, pp. 2792–2797. [Online]. Available: https://doi.org/10.1109/SMC.2018.00476

[10] X. Ma, Y. Wang, and Y. Wang, "Content based User Preference Modeling in Music Generation," in *MM '22: The 30th ACM International Conference on Multimedia, Lisboa, Portugal, October 10 - 14, 2022*, J. Magalhães, A. D. Bimbo, S. Satoh, N. Sebe, X. Alameda-Pineda, Q. Jin, V. Oria, and L. Toni, Eds. ACM, 2022, pp. 2473–2482. [Online]. Available: https://doi.org/10.1145/3503161.3548169

[11] C. Walshaw, "ABC Notation Standard, version 2.1," https://abcnotation.com/wiki/abc:standard:v2.1, 2011, accessed 2025.

BIKES: A Moving Networked Music Instrument in Interdisciplinary Research and Education

Laura Call Gomez
School of Music, Georgia Tech
lgomez47@gatech.edu

Gabriel Decker
School of Music, Georgia Tech
gdecker3@gatech.edu

Jayson Faupel
School of Music, Georgia Tech
jfaupel3@gatech.edu

Aditya Rajesh Pawar
School of Music, Georgia Tech
apawar43@gatech.edu

Jacob Westerstahl
School of Music, Georgia Tech
jwesterstahl3@gatech.edu

Henrik von Coler
School of Music, Georgia Tech
hvc@gatech.edu

ABSTRACT

This paper describes how a moving, networked instrument for music and sound art is used as a platform for interdisciplinary research and creative practice in higher education. The long-term project, BIKES, provides students with the opportunity to engage with real-world challenges by combining music technology, experimental composition, and industrial design. Project activities include interactive installations and sound rides, iterative development of hardware and software, as well as the design and fabrication of a new prototype for exhibition contexts. After its first year, BIKES demonstrates how the multifaceted nature of a modular instrument can facilitate collaborative work and increase the visibility of student-led research and development.

1. INTRODUCTION

1.1 The Interdisciplinary Nature of BIKES

Projects in the field of computer music and sound art are inherently interdisciplinary. Instruments must be designed and built, often involving digital fabrication, circuit design, and hardware prototyping. Software must be developed using a range of programming languages and paradigms. The processes of composition and performance require not only creative thinking, but also critical contextualization and methods drawn from the humanities. In an educational context, this interdisciplinarity presents both significant challenges and substantial opportunities.

With BIKES, we employ a networked musical instrument as a platform for interdisciplinary research, artistic practice, and pedagogical experimentation, exploring mobile, spatially situated performances and installations, as shown in Figure 1. In the context of this project, up to eight undergraduate students are engaged, co-led by graduate students and faculty. With most participating students majoring in music technology, the BIKES project positions this field as a hub discipline that coordinates methods from interaction design, industrial design, fabrication, embedded systems, and sound studies. In doing so, the project foregrounds learning through practice, encouraging students to engage with unfamiliar disciplinary perspectives and to reflect critically on their own design and artistic processes.

Copyright: ©2026 Laura Call Gomez et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License 3.0 Unported, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.



Figure 1: Visitors interacting with BIKES at an installation Copyright Kevin Beasley

Through project-based exploration, interdisciplinary design work supports a collaborative learning environment in which students can integrate technical and artistic approaches in the development of systems such as BIKES. By integrating disciplines with differing pedagogical frameworks, the project combines linear knowledge structures common in STEM fields with dialogue-driven learning approaches often found in arts and non-STEM education [1]. This interdisciplinary structure aligns with findings that students benefit from learning environments that blend independent technical work with interactive, hands-on collaboration [1].

This paper highlights the interdisciplinarity of BIKES by outlining its development across multiple project phases and illustrating the diversity of artistic and technical practices involved.

1.2 Background and Context

Building on the networked music system *SPRAWL* [2], BIKES is conceived as a moving, networked platform for artistic expression and interaction, aiming to increase accessibility by bringing sound art to new locations and enabling new forms of composition and performance. Mobile sound practices in public spaces have been explored in a range of artistic projects that treat physical movement as a constitutive element of the work. Earlier work, such as Max Neuhaus' *Listen*, established walking and mobility as integral to listening, positioning movement through public space as a central artistic concern [3]. More recently, Kaffe Matthews' *Sonic Bikes* employs bicycle-based mobility in urban environments as a compositional and experiential strategy, foregrounding movement through the city as part of the musical structure [4].

Although BIKES is a flexible platform for various forms of sound art and music, it has primarily been used to create urban soundscapes, realized either as stationary installations or through movement in public space. In developing a soundscape-oriented interactive system, the sound design of BIKES draws on R. Murray Schafer's descriptive framework, which distinguishes between *Keynote* sounds, *Signals*, and *Soundmarks* as functional components of an acoustic environment [5]. Within this framework, Keynote sounds form a continuous sonic ground, often shaped by environmental or geographic conditions, while Signals demand focused attention as foreground elements intended to be heard. Soundmarks denote distinctive sounds associated with the acoustic identity of a particular place or community [5].

Beyond Schafer's taxonomy, later soundscape practices have emphasized situated listening, participation, and critical engagement with urban environments, particularly in response to social, political, and ecological contexts [6, 7]. Taken together, these perspectives informed the design of a subsequent iteration of BIKES, which employs layered and interactive sonic elements to construct responsive soundscape environments that evolve in relation to place and movement.

The progression of BIKES can be categorized into three distinct phases. Phase 1, introduced in Section 2, focused on the design and development of a first working prototype and delivered a proof of concept for the complete instrument through its initial activations in spring 2025. Section 3 reports on Phase 1.1, which introduced several improvements to the hardware and software of each bicycle, followed by professional activations and increased outreach in summer and fall 2025. The ongoing Phase 2 of BIKES is dedicated to the implementation of a new prototype based on a custom-designed loudspeaker enclosure. Section 4 introduces first results from this phase, alongside documentation of an exhibition of the project.



Figure 2: BIKES in the laboratory during development.

2. PHASE 1

2.1 Hardware and Software

The BIKES system is based on an electric cargo bike manufactured by Edison Bicycles, an Atlanta-based electric bike company [8]. The model features a 750 W motor that supports both pedal assist and throttle control. A rear cargo section provides sufficient structural support to house the audio hardware while maintaining mobility. The payload capacity of 181 kg allows the integration of professional audio equipment without compromising range or road speed. Four of these bikes are equipped with music technology to form the complete instrument used in Phase 1.

Each bike runs on a Raspberry Pi 5 computer [9], mounted in a custom 3D-printed enclosure fitted to the handlebar. This enclosure also holds a 10-inch touchscreen that serves as the main user interface. One battery-powered loudspeaker per bike, along with an external audio interface and additional peripherals, is mounted on the rear cargo section. Further details of the hardware setup are described in a previous publication [8].

The networked music system *SPRAWL* informed parts of the BIKES design [2]. While *SPRAWL* relied on wired Ethernet connections to implement a fully connected audio routing system for ensemble interaction, BIKES uses a WiFi network for the exchange of control data rather than audio signals. In its first iteration, BIKES operated with a centralized network topology, in which three bicycle nodes were connected to a single central router carried by one bicycle. While this centralized topology was sufficient for an initial proof of concept, the intended direction was to transition to a decentralized mesh architecture in order to reduce single points of failure and network dropouts.

The networking capabilities constitute a core aspect of the system, both in terms of compositional possibilities and technological demands. Communication between the nodes is implemented using OSC (Open Sound Control) [10]. Additionally, *p2psc* [1] is used to simplify communication between the bicycles . This Python-based tool allows each node to register with a unique identifier and to subscribe to specific OSC paths, enabling flexible message exchange between devices [8].

[1] https://github.com/l42i/p2psc

SuperCollider was used for audio signal processing and the design of graphical user interfaces (GUIs) in the first phase. Each Raspberry Pi runs an independent SuperCollider server and language instance, with graphical user interfaces implemented in SuperCollider to provide touch-based control over audio and network processes [8].

Finally, *Ansible* is used to control and configure the instrument. This professional IT tool enables remote configuration and deployment of individual compositions across the system. Configuration is handled via SSH commands organized in playbooks, which specify ordered sequences of script operations executed on a set of hosts. An Ansible script is used to remotely start the SuperCollider processes and other software on each Raspberry Pi, eliminating the need to manually launch programs on individual nodes. This approach allows BIKES to be deployed on site without per-node configuration.

2.2 Activations and Proof of Concept

To demonstrate the system as a proof of concept and to evaluate its strengths and limitations, BIKES was deployed in a range of public scenarios in spring 2025. These deployments included tests of the user interface and battery life, as well as interactive installations in which the system was stationary and fully networked rides that presented the system in motion [8].

A first stress test took place at an Undergraduate Research Symposium at Georgia Tech, where a single bike was displayed and visitors interacted with the touch interface while triggering harmonic sounds over a looped background. This was followed by an interactive quadraphonic installation during Launchpad, the annual showcase at Georgia Tech's College of Design, where the installation was exposed to thousands of visitors and demonstrated the system's reliability, portability, and synchronization in a high-throughput public setting. Finally, the full system was evaluated in motion during the Inman Park Parade, where four fully equipped bikes performed along a two-mile route, demonstrating both a synchronized clock mode and an independent node paradigm. This activation provided the first opportunity to test and explore the potential of real-time sound control and spatial audio in a dynamic, moving performance context.

3. PHASE 1.1

3.1 Hardware and Software

Phase 1.1 represents the current working configuration of BIKES. In terms of the hardware design, only minor changes were made to the first working version in order to increase the robustness of the system for extended rides. These measures included additional padding for loudspeakers and peripherals, bungee cords to secure the lid of the wooden enclosure, and minor modifications to the display enclosure.

The most significant change in this phase concerns the network. Using one UniFi AC Mesh router on each bike, BIKES was upgraded to a mesh network, representing a substantial improvement over the first iteration. This model was chosen for its extended outdoor range of 600 ft, reducing connection dropouts and latency in comparison to the previous setup. One router is assigned as the parent node connected to a network gateway, while all routers are interconnected to form a mesh. Each Raspberry Pi is connected to its associated mesh router via Ethernet, resulting in the structure shown in Figure 3.

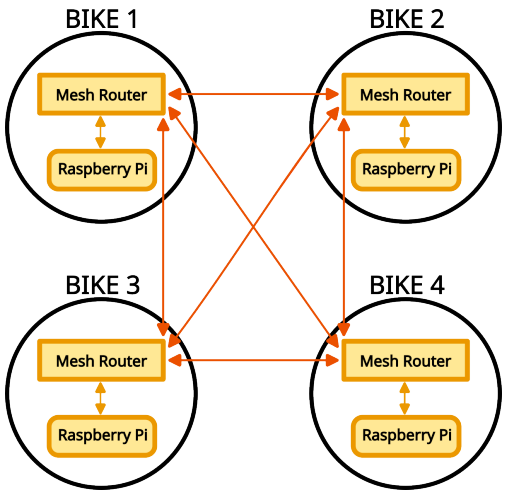


Figure 3: Updated network concept for Phase 1.1.

SuperCollider is retained as the main audio backend. Its ability to run fully scripted and headless makes it well suited for this application. The OSC communication approach is also retained, relying on direct communication between the SuperCollider instances and p2psc. To introduce new possibilities for interactive design, Python's Pygame library is used to create graphical user interfaces for installations in Phase 1.1. This portable, SDL-based framework remains lightweight and accessible in an educational context, while providing access to graphics, audio, and other multimedia hardware across operating systems [11].

3.2 Activations and First Soundscape Compositions

After the proof of concept in the first phase and with improved hardware and software, BIKES was deployed in several activations in fall 2025. The project was presented as a stationary interactive installation at SITE, a multimedia arts festival at The Goat Farm in Atlanta[2]. This installation marked the first presentation of a distributed soundscape installation specifically designed for BIKES. The four bicycles were arranged in a rectangle in an alleyway to take advantage of the acoustics, with each bike's speakers facing inward, as shown in Figure 4.

Following Schafer's framework, several distinct soundscapes were designed. These ranged from a serene ocean setting to a noisy industrial space and a vibrant rainforest environment. Keynote sounds were played simultaneously across the four bicycles and continuously for the duration of the performance, acting as a central foundation for the soundscape. Signals and Soundmarks could be triggered by the audience through the graphical user interfaces. Each bike's display showed a simple GUI that allowed visitors to trigger sound events played back on all loudspeakers,

[2] https://www.thegoatfarm.info/site2025



Figure 4: Setup of BIKES at SITE for an interactive installation.

following quasi-random patterns. The screen displayed an image representing the environment, divided into a grid of buttons. Depending on the button pressed, a single instance was played by the same bicycle, repeated instances were passed between bicycles in a predetermined pattern, or repeated instances were passed around in a random pattern. This resulted in a dense and immersive sonic environment with direct means for audience interaction.

Building on the successful presentation of an interactive installation, Phase 1.1 of BIKES was deployed as a moving instrument during a public group bike ride held in collaboration with The Fulton County Public Art Program [3], MOBB [4], and Red Bike and Green [5]. The aim of this site-specific experience was to highlight strengths and weaknesses of Atlanta's urban environment, both in terms of acoustics and safety for bicycles and pedestrians. This capacity for critical engagement with the surrounding environment constitutes a key aspect of the instrument. More than 30 riders accompanied the instrument on a two-hour route through the inner city, temporarily occupying full traffic lanes in the manner of a Critical Mass ride [12].

Two concepts were explored in this activation. The first involved a playlist of songs by established and popular local artists, curated for the ride by the students. When in motion, all four bikes played the same stereo file per piece, synchronized via SuperCollider. Through the connection to Atlanta as a hub for music production and performance, this approach created a direct connection to the incidental audience. The compositions were easily recognized and elicited immediate positive responses from bystanders and participants along the route.

The second concept was intended for impromptu installations during the ride. The group stopped at previously selected locations to demonstrate the soundscapes introduced earlier in this subsection. Some of these locations were chosen as illustrative examples of favorable or challenging urban acoustic environments, including freeway overpasses, tunnels, intersections, and paths through parks. Other locations were added as stops due to their status as iconic landmarks in the city. The soundscapes were designed to either contrast or complement the surrounding acoustics, encouraging critical listening among the audience.

[3] https://www.fultonarts.org/publicart
[4] https://www.mobbcycling.com/
[5] https://www.redbikegreenatl.com/

4. PHASE 2

4.1 Prototype Design and Fabrication

Following the proof of concept and initial public activations of the improved system, Phase 2 of BIKES shifted its focus toward the design and fabrication of a custom hardware solution for housing the loudspeakers and onboard technology. This transition corresponded with a shift in project activities, moving from software development and artistic exploration toward industrial design and digital fabrication.

To address hardware limitations present in the working instrument, a new, rugged loudspeaker enclosure was designed and built. With the aim of improving stability and long-term durability, a wooden frame was constructed around the rear wheel to lower the center of mass and improve rideability. Based on these design principles, a rapid prototyping framework was selected to develop an enclosure that integrates not only the system's loudspeakers but also the electronic hardware required for music generation and network interaction in future design iterations.

Crucially, prior feedback from field tests conducted in Phase 1 informed these design decisions. For example, operators of the initial prototype reported safety concerns related to a high center of gravity, which resulted in unpredictable steering at low speeds. This feedback led to the decision to position the enclosure around the rear wheel, ensuring a center of gravity as low as possible to maximize rider maneuverability while avoiding ground contact on uneven terrain. Figure 5 shows a 3D model of the basic design, which features two 15 inch woofers mounted on either side and one 12 inch woofer at the rear.

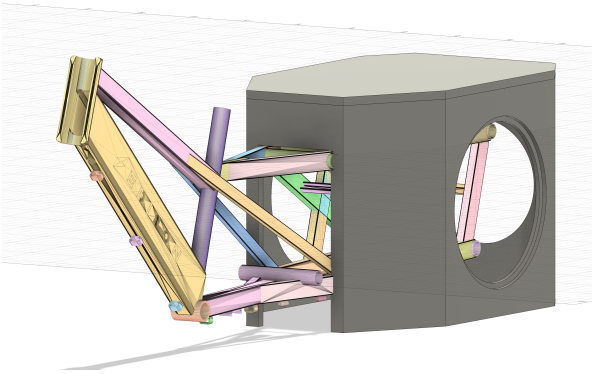


Figure 5: CAD Model of Phase II speaker cabinet.

Given the timing of the second design phase and the plan to present the system as a stationary object in a display context, a *looks-like* model was fabricated to communicate the artistic and material qualities of BIKES without the ability to produce sound. In this context, a looks-like model refers to an appearance-focused mock-up intended to convey form, scale, and material presence, whereas a *works-like* prototype prioritizes demonstrating functional behavior [13]. These distinctions from industrial design and HCI research informed the development of Phase II, with the initial iteration of BIKES serving as the works-like prototype [2], while the second phase presents an explanatory, non-functional artifact focused on visual and spatial communication.

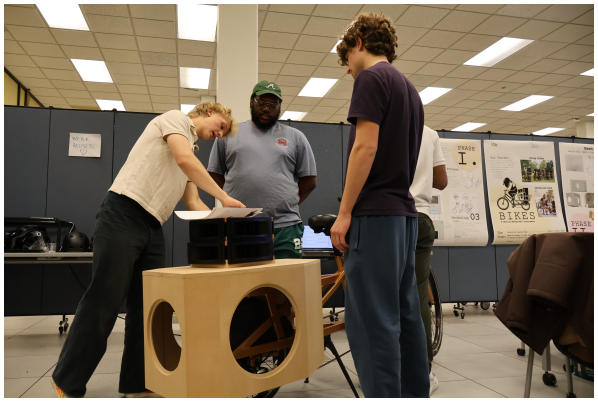


Figure 6: Phase 2 prototype manufacturing in the lab.



Figure 7: Phase 2 speaker cabinet mounted on the cargo bike.

The design process followed a series of iterative steps consistent with research-through-design methodologies, in which design activity itself functioned as a mode of inquiry, supporting both creative exploration and team communication throughout development. First, the strengths and limitations of the original model were systematically analyzed to identify design constraints and opportunities. Based on this analysis, initial concepts were developed through hand sketches and reflective discussion, establishing a provisional understanding of the instrument's overall form. Subsequently, scale-accurate drawings were produced to examine proportions and identify inconsistencies, enabling iterative refinement prior to fabrication. These refined drawings were then translated into Computer-Aided Design (CAD) models using software such as Autodesk Fusion, which served as the basis for fabrication on an industrial Computer Numerical Control (CNC) machine, allowing the enclosure to be assembled by hand. Figure 6 shows the addition of the upper structure supporting eight 10.5 inch horn tweeters to the basic enclosure in the laboratory. The tweeters are arranged in a square configuration to facilitate omnidirectional sound projection.

The complete enclosure is coated with a weather-resistant loudspeaker finish, matching the black visual language of the cargo bicycle. Figure 7 shows the assembled prototype, highlighting the lowered center of gravity. Although this iteration constitutes a non-functional prototype, the design is intended to accommodate the necessary hardware to be developed into a fully operational system.

4.2 Exhibition Concept and Design

The Phase 2 prototype introduced in Section 4.1 was specifically developed for display in a public exhibition context at Atlanta Hartsfield–Jackson International Airport, curated by Georgia Tech Arts[6]. Figure 8 shows an early concept sketch for the display, illustrating the project's trajectory by presenting the appearance model alongside contextual materials, including posters and a documentary video showing the operational instrument in use.

In addition to the physical prototype, the display incorporated a set of supplementary materials intended to contextualize the system for a non-specialist public. A series of three posters presented key aspects of Phase 1 and Phase 2, emphasizing visual media and concise text blocks to communicate the project's development and design logic. To further convey the concept and scope of BIKES, a professionally produced video was developed in collaboration with Kevin Beasley. The video documents the project's motivation and use through real-time footage of the instrument in operation, complemented by short interviews with students and team members involved in its development. Animated visualizations in the video illustrate the operation of the mesh network within BIKES, interwoven with documentation from multiple public deployments throughout 2025. These materials were presented as part of a year-long display alongside the Phase 2 prototype at Hartsfield–Jackson Atlanta International Airport, as shown in Figure 9.

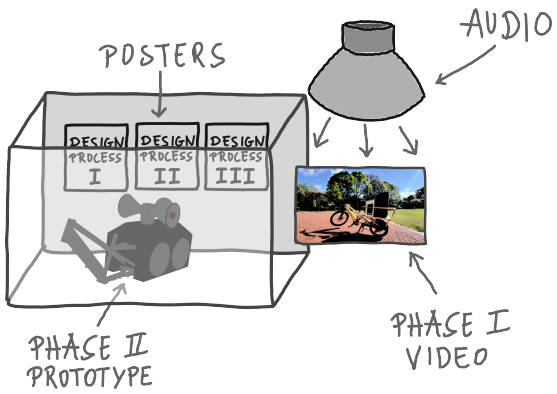


Figure 8: Original concept for the airport exhibition.

5. CONCLUSION

Across its first two phases, BIKES developed from a networked musical experiment into a moving spatial audio system integrating aspects of music technology, interaction design, and industrial fabrication. Phase 1 established the technical and musical foundations of the system, including a mesh-based network architecture, distributed audio rendering, and mechanisms for audience interaction, while Phase 2 addressed questions of physical integration, durability, and visual legibility in public contexts. The

[6] https://design.gatech.edu/feature/design-airport-exhibit-2025



Figure 9: Exhibition with prototype and posters.

project was realized through an interdisciplinary development process in which compositional practice, software engineering, industrial design, and digital fabrication informed successive design iterations. Students from different disciplinary backgrounds contributed to the project at multiple stages, engaging in hands-on collaboration across technical, artistic, and organizational domains. As a result, BIKES functions both as a mobile ensemble for spatial sound works and as a platform for iterative design and experimentation in public space.

Future work will integrate the functional capabilities demonstrated in Phase 1 with the design principles developed in Phase 2 to realize a fully operational version of the system. In its second year, BIKES will be developed within a structured Vertically Integrated Projects (VIP) program, providing an organizational framework for continued student-led, interdisciplinary research and development.

Acknowledgments

The project would not have been possible without the support from our partners, including Ryan Hersh from Edison Bicycles, Marius Vieth for the woodwork, Birney Roberts for curating the airport exhibition, as well as The Goat Farm and The Fulton County Public Arts Program. Special thanks to our team members Shirin Sathe, Donald Buck, and Orlando Kenny.

6. REFERENCES

[1] R. Owston, D. N. York, T. Malhotra, and J. Sitthiworachart, "Blended Learning in STEM and Non-STEM Courses: How Do Student Performance and Perceptions Compare?." *Online Learning*, vol. 24, no. 3, pp. 203–221, 2020.

[2] von Coler, Henrik and TonnäTonntt, Nils and Kather, Vincent and Chafe, Chris, "Sprawl: A network system for enhanced interaction in musical ensembles," in *Proceedings of the 18th Linux Audio Conference*, 2020, pp. 33–37.

[3] M. Neuhaus, "Listen," Sound art project, 1966, series of guided listening works foregrounding movement through public space. [Online]. Available: https://www.max-neuhaus.info/listen

[4] K. Matthews, "Sonic Bikes," Art project, 2013, mobile sound art project using bicycles in urban environments. [Online]. Available: https://www.kaffematthews.net/projects/sonic-bikes

[5] R. M. Schafer, *The soundscape: Our sonic environment and the tuning of the world*. Simon and Schuster, 1993.

[6] P. Cusack, *Favourite Sounds: Memory, Imagination and the Search for a Sonic Common Ground*. Soundscape Journal, 2012, soundscape and participatory listening projects in urban environments.

[7] A. McCartney, "Soundwalking: Creating Moving Environmental Sound Narratives," *Leonardo Music Journal*, vol. 20, pp. 19–24, 2010.

[8] Kenny, Orlando and Gomez, Laura and Sathe, Shirin and von Coler, Henrik, "BIKES: A Moving Network Instrument for Music and Sound Art," in *IEEE International Symposium on the Internet of Sounds*, L'Aquila, Italy, 2025.

[9] H. D. Ghael, L. Solanki, and G. Sahu, "A review paper on raspberry pi and its applications. International Journal of Advances in Engineering and Management," in *IJAEM*, 2020.

[10] M. Wright, A. Freed *et al.*, "Open soundcontrol: A new protocol for communicating with sound synthesizers," in *ICMC*, 1997.

[11] P. N. Shinde, Y. J. Chavan, S. G. Chilka, G. N. Patil, and M. R. Kharde, "Pygame: Develop games using python," *International Journal for Research in Applied Science and Engineering Technology*, 2021.

[12] C. Carlsson, Ed., *Critical Mass: Bicycling's Defiant Celebration*. Oakland, CA: AK Press, 2002.

[13] D. Mathias, C. Snider, B. Hicks, and C. Ranscombe, "Accelerating product prototyping through hybrid methods: Coupling 3D printing and LEGO," *Design Studies*, vol. 62, pp. 68–99, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0142694X19300183

Scores That Run: Graphic Notation with Embedded Performance Semantics

Rob Canning
Independent Researcher
rc@kiben.net

ABSTRACT

This paper presents an approach to digital graphic notation in which performance semantics are embedded directly into the visual surface of the score. Working in standard SVG and authored entirely in Inkscape, the score is composed as a graphic–semantic document: visual elements carry lightweight cue structures encoded in their identifiers, and these cues are executed in real time by a browser-based runtime. The score therefore functions simultaneously as image, temporal structure, and performative interface, without reliance on symbolic engraving or external playback systems.

The framework supports hybrid formal topologies, including continuous scrolling trajectories, page-based local environments, and patterned navigation between sectional states. Animated motion fields provide shared gestural resources for ensemble coordination and may optionally drive live electronic processes, enabling a unified grammar of acoustic and electronic gesture. All cue semantics—structural, temporal, gestural, textual, and media-based—are authored within the same executable layer as the notation, so behaviour and interpretation arise from a single surface.

Because the system is based entirely on open web standards, it enables a direct draw-and-perform workflow accessible to composers and performers without specialised technical infrastructure.

1. INTRODUCTION

Graphic notation continues to play a significant role in contemporary composition, offering visual structures that invite interpretation, gesture, spatial metaphor, and temporal fluidity. However, the transition from the drawn score to a performable one often requires additional mediation: rehearsal explanation, auxiliary timing systems, proprietary playback environments, or manual synchronisation cues. In many cases, the notation and its performance logic remain separated, with behaviour residing outside the score in software patches, conductor instruction, or group negotiation.

Copyright: © 2026 Rob Canning. This is an open-access article distributed under the terms of the Creative Commons Attribution License 3.0 Unported*, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.*

This paper presents an approach in which the score itself encodes its own performance semantics, allowing the visual domain and the temporal domain to coexist within a single representational layer. The system described here employs SVG as the underlying score format, authored in the WYSIWYG environment of Inkscape, with performance behaviour expressed through lightweight cue microsyntax embedded into standard element identifiers. When loaded in the browser, these cues are parsed and enacted in real time, making the score executable: it animates, signals entrances, triggers durations, and guides the unfolding of form without requiring external software environments or scripting languages.

A browser-based runtime interprets these embedded semantics to support timeline-oriented, page-based, and sectional score geometries, enabling the composer to shape temporal structure according to the demands of the material rather than a predetermined display model. When used in ensemble contexts, device synchronisation is handled through a coordinating process that maintains a shared temporal state while leaving visual rendering and interpretation local to each performer's browser. The result is a direct authoring-to-performance workflow in which visual composition, temporal behaviour, and performance execution remain tightly integrated.

This work extends the author's earlier investigations into web-based score distribution and real-time networked notation. The NodeScore system [1] demonstrated how web standards—HTML5, SVG, and WebSockets—could provide infrastructure for networked score coordination, while subsequent work on parallax scrolling interfaces [2] explored continuous timeline navigation for composed improvisation. Whereas the earlier Oscilla paper [3] introduced the system as a synchronized performance environment, the present paper addresses the score as a graphic–semantic notation system, authored directly in SVG and enacted in performance through embedded cue semantics and animated gesture fields.

2. RELATED WORK

The question of what constitutes a musical score in digital environments has been explored across systems that treat notation variously as visual composition, reactive inter-

face, control structure, or networked performance space. The approach developed here aligns with work that treats the score as a graphic surface endowed with temporal and behavioural semantics, while differing in that these semantics are authored visually and embedded directly into the document itself.

2.1 Graphic Composition: Drawing as Score

A foundational precedent appears in Xenakis's UPIC system [4], where drawn lines, shapes, and trajectories define musical structure and are rendered into sound via synthesis. UPIC proposed that graphic form could itself be compositional material rather than a representation of something composed elsewhere.

Oscilla shares this premise—that musical thought may proceed visually—but diverges in its treatment of performance. In UPIC, drawings are translated into an independent sonic artefact. In Oscilla, performance semantics remain attached to the drawing: the score continues to operate as a score, read, interpreted, and enacted by performers, even when it also animates or influences electronics.

2.2 Dynamic and Message-Driven Notation

INScore [5] introduces a model for scores modifiable at runtime. Visual elements can be repositioned, transformed, and updated through OSC messaging, making the score a live visual environment responsive to external control.

Oscilla differs less in capability than in authorship model. INScore relies on script-based construction and continuous message control; Oscilla embeds performance semantics into the SVG document itself. Authoring remains entirely visual: behaviour is declared locally in the drawing rather than orchestrated from an external control layer.

Systems such as MaxScore [6] and the Bach library [7] situate notation within Max/MSP's dataflow environment, enabling close coupling between visual material and algorithmic or DSP processes. These frameworks have shaped much practice in digital notation, but they position notation as an outcome of patch logic rather than as the primary compositional site. Oscilla takes the opposite stance: the visual page remains the principal site of composition.

2.3 Trajectory-Based Control: IanniX

Building on conceptual ground opened by UPIC, IanniX [8] extends drawn form into a real-time control environment. Trajectories, triggers, and cursors are evaluated at runtime, generating OSC and MIDI messages that drive external sound engines.

Like Oscilla, IanniX treats the graphic layer as active and performative. The divergence lies in orientation: IanniX positions the score primarily as a controller for electronic systems, whereas Oscilla treats the SVG document as a performable score in its own right. Animated motion fields, navigation cues, and temporal structures are intended to be read by musicians at the same time as they may produce control data. Notation and execution coincide rather than being divided into controller and instrument.

2.4 Distributed Performance Environments

Hajdu's Quintet.net [9], launched in 2000, established an early model for networked notation under conductor control, connecting performers across local and wide-area networks through a shared symbolic score environment. It demonstrated that distributed coordination could be managed through notation interfaces rather than audio synchronisation alone.

More recently, the Decibel ScorePlayer [10] developed by Hope, Vickery, and Wyatt provides network-synchronised scrolling of graphic notation across iPads. The system supports animated scores, embedded audio, OSC communication with external processes, and a "canvas layer" for real-time drawing.

Oscilla shares Decibel's commitment to graphic notation and distributed synchronisation, but differs in authoring model and platform. In Oscilla, cue semantics are embedded directly into SVG identifiers and authored in a general-purpose vector editor (Inkscape); no dedicated authoring tool or platform-specific app is required. The score is a standard web document interpreted by any modern browser, and behaviour arises from the same surface as the notation rather than from a separate runtime layer.

2.5 Programmatic Notation: neoscore

neoscore [11], developed by Yoon under the DigiScore project, takes a code-first approach to graphic notation. As a Python library, it treats scores as programmatically constructed shapes and text, supporting generative score-making, animation, and live-coding.

Where neoscore requires programming, Oscilla requires only drawing. Both reject the assumptions of traditional engraving software, but they occupy opposite ends of the authoring spectrum: neoscore suits composers comfort-

able with code; Oscilla suits those who think through visual arrangement and spatial relationship.

2.6 Positioning the Present Work

Across these systems, multiple models emerge for how notation might participate in performance: as graphical composition, as dynamic interface, as controller surface, as networked document.

Vear's taxonomy of the digital score [12] identifies defining features including the "animated score," "technological conductor," and "system as score." Oscilla spans several of these categories. It is an animated score in which motion fields and temporal cues unfold visually; it embeds conductor-like temporal semantics that coordinate ensemble behaviour without external control; and the browser runtime that interprets the SVG is inseparable from the score itself—the system *is* the score.

The contribution of Oscilla is to treat the score as a single executable surface in which visual composition, temporal structure, performer interpretation, and optional electronic control are co-authored inside the same drawing. Cues, navigation, and animation are not layered on afterwards or driven from external scripts; they are embedded into the graphic objects themselves and enacted at performance time.

3. SYSTEM OVERVIEW

A preliminary account of OscillaScore as a synchronized scrolling display environment appears in Canning [3], where the emphasis is on network coordination and distributed performance. The present work addresses the score itself as a composed graphic–semantic document, focusing on authoring workflow, cue structure, hybrid formal topology, and gesture semantics.

3.1 Authoring in Inkscape

Scores are created directly in Inkscape, using line, shape, spatial organisation, typographic emphasis, and contour as primary compositional material. Conventional notation can be included when useful, but the system does not assume symbolic engraving as the default. Visual form and spatial relationships are treated as musically meaningful structures, not illustrations of another underlying representation.

Performance semantics are introduced by assigning identifiers to SVG elements through Inkscape's XML editor (Ctrl+Shift+X), inserting the cue microsyntax directly into each element's id attribute.[1] The composer remains inside a visual workflow: notation remains graphic, while semantics are introduced through small, local adjustments rather than scripting or external configuration. The score remains a readable, editable drawing that directly encodes its own performance behaviour.

3.2 Cue Semantics Embedded in SVG Identifiers

Performance actions and structural transitions are defined through a compact microsyntax embedded in SVG id attributes. For example:

```
nav(mode(scroll@F), speed:0.8)
pause(10, count:true)
```

These identifiers are parsed at performance time to form an internal representation of temporal flow, navigation, and interpretive events. The semantics remain co-located with the visual elements they govern: the score's behaviour is part of its surface, not an auxiliary control patch.

3.3 Runtime Interpretation

The performance runtime is built on standard web technologies: JavaScript, the SVG DOM, and requestAnimationFrame timing. When a score is loaded, the runtime reads the SVG structure, extracts cues, and constructs either a continuously traversed timeline or a network of navigable score regions, depending on layout mode.

In ensemble settings, navigation and temporal state are synchronised across devices, allowing the score to be performed collaboratively without a dedicated playback master or conductor. The SVG file itself remains unchanged; the runtime enacts what the document declares.

3.4 Score Topology, Animated Gesture, and Media

Oscilla supports a hybrid approach to form in which scroll-based continuity and page-based locality coexist. Scroll mode emphasises large-scale directional movement, allowing the score to unfold horizontally through time. Page mode defines spatially discrete environments in which performers may remain, explore, or develop material. Transitions between these states are authored directly in the score, making the articulation of form visible, repeatable, and performable.

Animated transformations within the score—motion along paths, rotation, scaling sequences, orbital trajectories—are authored visually and evaluated at runtime.

1 A dedicated Inkscape extension for authoring cue semantics is in beta testing.

These motion fields may be read interpretively by performers, or they may expose continuous control data to live electronics. For example:

```
o2p(path:p03, dur:60, osc:1, oscaddr:"/
o2p/pvoc/freq")
```

When osc:1, the motion produces streams of OSC data suitable for environments such as SuperCollider. Acoustic and electronic layers can therefore share a single, visually authored grammar of gesture.

Audio and video may also be embedded directly within the score and treated as composable performance material rather than external accompaniment:

```
video(file:"reflection-III.mp4")
audio(file:"breath.wav", loop:2)
```

Media may be activated automatically at structural points or triggered by performers. In both cases, it participates in the same runtime semantics as navigation and gesture, entering the performance environment from within the document rather than from an external playback system.

These mechanisms position the score as a place where behaviour is composed. The following section considers the notational consequences of treating cues as part of the score's surface.

4. CUE SEMANTICS AS TEMPORAL AND PERFORMATIVE NOTATION

In Oscilla, cue semantics are treated as notation rather than metadata or auxiliary control logic. A cue contributes simultaneously to how a passage is read and to how it unfolds in real time. The same visual marks that shape interpretation also activate behaviour in the runtime, so that temporal structure, orientation, and execution coexist on the surface of the score.

4.1 Cues as Encoded Temporal Structure

Certain cues articulate form, duration, or transition as explicit temporal notation. A cue such as:

```
metro(bpm:90, beats:3,hold:10)
```

brings up a visual metronome for ten seconds—a performer-facing temporal aid authored directly into the score. Such cues occupy a role analogous to rehearsal letters, fermatas, or tempo markings in traditional notation, but differ in that the temporal structure is not merely indicated—it is executed.

Temporal structure is therefore authored directly in the drawing: the score does not merely suggest time; it performs it.

4.2 Cues as Orientation and Perceptual Condition

Not all cues impose structural change; some shape attention, environment, and perceptual stance without prescribing specific sonic outcomes.

Text and media cues work this way. A text cue does not simply place static words on the page—it references external material and reveals it as a temporally evolving object: lines may appear sequentially, characters may surface gradually, fragments may emerge in shifting order. Similarly, embedded audio or video is treated as score-native material arising from the same semantic layer as notation. A video may function as a moving field of attention; an audio cue may establish resonance, operate as a slowly evolving texture, or be delivered privately to an earpiece to enable close imitation.

Temporal orientation can also be composed without specifying musical content. A countdown pause holds the score in place while sound continues. Alternatively, timing may be shaped by altering the traversal rate rather than stopping it; a cue such as

```
speed(value:1.4, dur:6, ease:inOutQuad)
```

functions as an embedded accelerando, changing how quickly material arrives and how coordination unfolds, without dictating the material itself. These cues treat orientation—textual, medial, temporal—as compositional material in its own right.

4.3 Interpretation & Synchronisation in Performance

During performance, cues are evaluated relative to the current scrolling or page-based state of the score. When a cue is reached, its semantics are executed locally in each performer's browser, while shared temporal position remains synchronised across devices.

Structural coherence is maintained without eliminating interpretive agency. The score functions simultaneously as a graphic notation shaping space and gesture, a temporal specification articulating when form unfolds, an interpretive environment orienting perception, and an executable surface enacted directly in performance.

No distinction is made between the score that is read and the system that is performed: they are the same object.

5. SYSTEM IN PRACTICE

Oscilla supports both fixed-form composition and situations where material unfolds responsively in performance. Rather than prescribing a particular notational style, the framework enables structural, interpretive, and gestural layers to intersect within a single score.

5.1 Structure and Attention

Oscilla shapes performer attention through two complementary modes: navigable structure and animated gesture. Both are authored visually and enacted at runtime, directing where the ensemble is within the work and what they attend to while there.

Scrolling sections articulate directional time, while page environments operate as discrete nodes: places the ensemble may enter, inhabit for a period, and return to under different conditions. A page is not simply turned; it is activated: `page(form-G3)`. From there, the score may remain in that state or move on. A navigation cue acts as a hinge, returning the ensemble to the scrolling trajectory or redirecting to another page:

`nav(mode(scroll@F), speed:0.8)`.

Page mode functions less as a linear succession of frames and more as a network of connected locations within the score.

Pattern structures can specify traversal through these locations:

`Pxrand([C1:30,C2:120,C3:45],inf)`

Here, the work moves among several nodes according to probabilistic choice, and each node carries a proposed duration. Form arises through navigation rather than playback: the same environments may appear in different orders and under different temporal conditions, while remaining fully contained within the notational layer. Both deterministic and indeterminate approaches are possible, and complexity can be increased through nesting. This pattern syntax draws on SuperCollider's pattern library [13], adapted for score traversal rather than event generation.

Pauses, timers, and speed changes articulate how long the ensemble remains in these states, but it is the nodal organisation—sometimes fixed, sometimes open—that determines where they are within the piece. The degree of indeterminacy is a compositional decision rather than a property of the system.

Hypertextual traversal may also be explicitly chosen rather than generated. A set of pages can function as a navigable menu, allowing a conductor or performer to move intentionally between environments in performance. The composer authors the available states and their connections, but not the exact route through them. Form emerges through situated decision-making within a network the score itself defines.

Within these structural states, animated transformations —motion along paths, rotation, scaling, and orbital trajectories—act as shared gestural attention fields. A gesture may remain purely interpretive, or it may generate continuous OSC control for live electronics:

```
rotate(dir:-1, dur:6, osc:1, oscaddr:"/
spat/4chPan/src3", uid:ar03)
```

When osc:0, motion is read. When osc:1, motion is also transmitted, allowing acoustic shaping and electronic processing to share a unified grammar of gesture authored in the score.

Across these modes—structural and gestural—the score composes conditions for listening and attention, not merely directives for sound production.

5.2 Distributed Authoring: Propagation and Reuse

Beyond individual cues, Oscilla supports large-scale authoring techniques that remain entirely within the visual workflow.

The propagate(...) pre-parser expands a cue or animation template across all children of a group, assigning each element its own evaluated parameters. This enables variation across many related objects—for instance, clusters of OSC-triggering shapes whose pitch, envelope, or density are derived from their geometry. A collection of freely drawn objects may be assigned a single propagated cue, after which each behaves as if it had been individually composed:

```
propagate(
  osc(
    addr:"/synth/pontalist",
    pitch:deg(${1}, ${2}),
    env:size,
    uid:rndoct1),

  irand(0,11),
  rnd([1,3,4]))
```

Here, irand generates new random values continuously, while rnd randomises only at initial parse—a choice between static variation and ongoing mutation.

Complementing propagation, reusable blocks allow structural modules to be defined once and redeployed elsewhere in the score. Groups marked for reuse can be inserted via use(...), preserving internal cues and animations while automatically avoiding identifier collisions. This supports repeatable interface elements, menu structures, and recurring formal modules without leaving the graphic domain.

Together, propagation and reuse introduce modularity and scalability into the notation: visual structures may be repeated, distributed, and differentiated, while remaining authored and legible as part of a single SVG document.

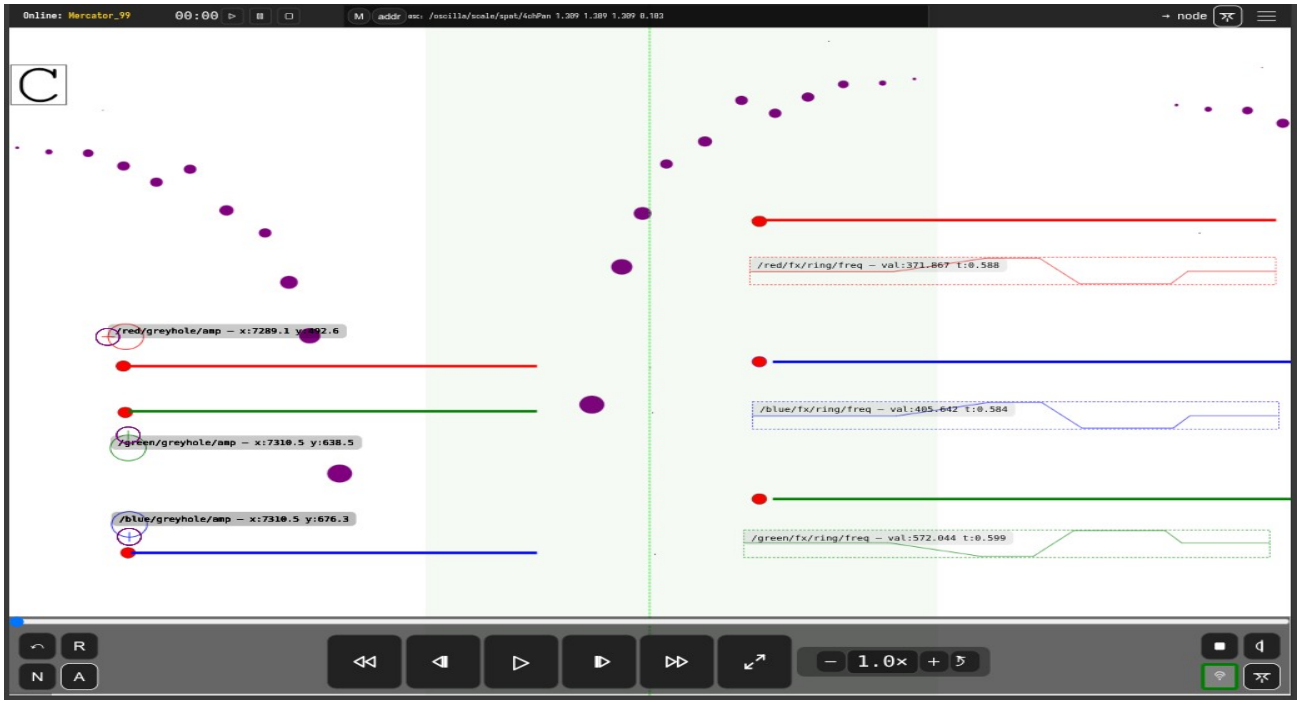


Figure 1. *Section C from Ways Through, Not To.: This excerpt shows the scrolling environment with play-head-triggered nodes (Y=pitch, size=amplitude), indeterminate instrumental lines, o2p gestures for spatial control, and breakpoint curves driving ring modulation.*

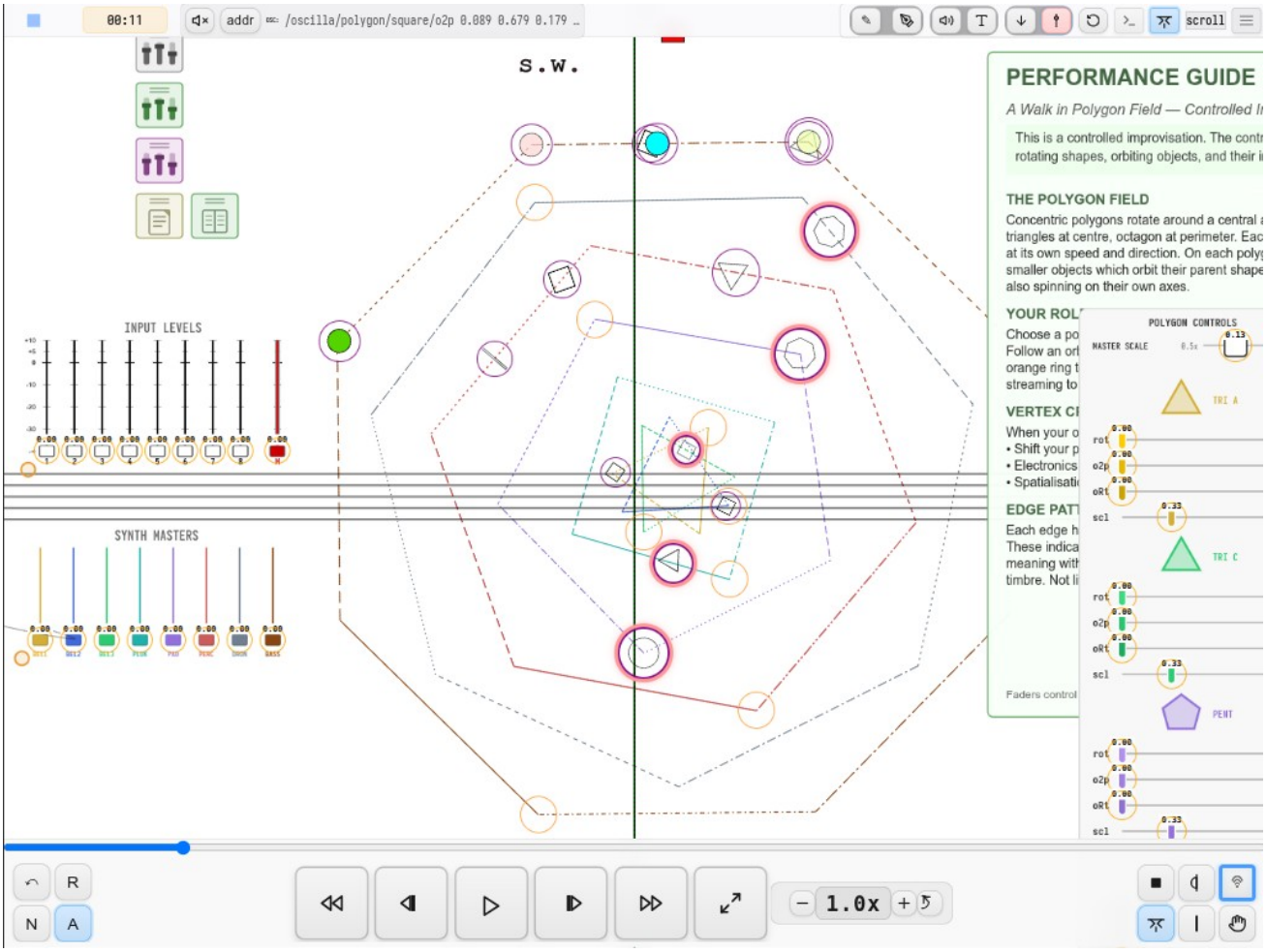


Figure 2 Polygonal paths and an outer spatialisation ring structure performer and electronic behaviour within a constrained yet open environment.

5.3 From Framework to Repertoire

These structures enable scores in which form is composed yet remains flexible, interpretation is guided without being prescribed, and animated gesture functions simultaneously as notation and behaviour. The following case study from Ways Through, Not To. demonstrates how these principles operate in a concrete musical context.

6. CASE STUDY: "A WALK IN POLYGON FIELD"

The preceding sections addressed the system in general terms. The following example presents a section from the developing work *Ways Through, Not To.*, showing how structural, animated, and electronic elements operate when embedded in a real score. This section—titled *A Walk in Polygon Field*—occupies a page-based environment where geometry, motion, and control data become formative agents of form.

6.1 Polygon Fields and Distributed Behaviour

A Walk in Polygon Field takes place within a page-based environment built from concentric polygonal forms. At the centre, a pair of triangles is enclosed by a square, which is in turn enclosed by a pentagon, and so outward to an octagon at the perimeter. Each polygon rotates around the central axis at its own speed and direction, with default states adjustable via a control panel authored directly within the score.

On each polygon sits a pair of smaller "mini polygons," which both orbit their parent shape and rotate on their own axes. These nested motions—orbit and spin—are independently controllable, producing a layered field of visual polyrhythm.

Control faders for all animations are constructed within the score itself, using o2p objects moving along straight paths. These faders communicate via Oscilla's internal signal bus—a subscription system analogous to Pure Data's send and receive objects—allowing any visual parameter to be adjusted in performance. Alongside the animation controls, the score includes mixer faders for instrument input and output levels, effects sends, and synthesiser levels. These mixer controls communicate directly with the SuperCollider patch via OSC, which handles all live electronics and spatialisation.

Any animated element may also expose its motion as OSC data: double-clicking its handle enables continuous streaming of rotation or orbit position to the electronics performer, who may map these streams to synthesis parameters, spectral processing, or spatialisation. In addition, discrete OSC triggers are sent each time an orbiting mini polygon crosses a vertex of its parent shape, providing a rhythmic pulse of attack events synchronised to the visual structure.

Performers may interact with the score directly, pausing and resuming their follower objects via touch control on their tablets. All animations and score state are synchronised over the network; each performer's selection is indicated by a distinct colour assigned at score load time, making individual agency visible within the shared field.

Each side of each polygon carries a distinct stroke pattern —solid, dashed, or dotted—indicating different performance modes: attack type, articulation, or timbral approach, to be negotiated among performers. Performers follow an orbiting mini polygon along these paths, transitioning from one mode to another as their object crosses a vertex. These transitions coincide with the vertex triggers sent to electronics, binding acoustic gesture and electronic response.

Spatialisation is similarly coupled to the score. Each performer's microphone output is mapped to the 8.1 speaker ring according to the rotational position of their parent polygon; vertex crossings trigger shifts in ring modulation frequency applied to that instrument's signal. The result is a tightly intertwined sonic space in which instrumental sound, spatial position, and electronic transformation move together.

Additional structural layers are available to performers who choose to engage with them. The polygon group sits atop a 2x2 quadrat; as an object moves from one quadrant to another, this provides a structural marker that may indicate a mapping change. Collisions or proximity with another performer's object may suggest interaction—whether dynamic, mimetic, or otherwise. There are as many interpretive layers as performers can hold in mind; the score offers depth without mandating it.

The live electronics provide a harmonic and textural field —a space the musicians inhabit as they listen and interpret. Performers are not obliged to follow the score unquestioningly; they should listen, find space, respond, pause, using the animating notation as a guide rather than a dictate.

A performance note explaining how to interpret the score is embedded within the score itself.

7. CONCLUSION

The approach presented here treats the score as a graphic–semantic surface in which visual form, temporal

structure, and performance behaviour are composed together. By embedding cue semantics directly into standard SVG notation authored in Inkscape, the score functions simultaneously as image, structural framework, and executable performance environment. Temporal articulation, sectional navigation, interpretive orientation, media integration, and gestural motion fields are composed at the level of the drawing itself, without requiring symbolic engraving, custom playback interfaces, or auxiliary software patches.

Because authoring occurs in a general-purpose vector editor, the workflow remains accessible to composers and performers without programming expertise or specialised technical infrastructure. The score may be created, revised, and expressed entirely in visual terms. This lowers the barrier to working with animated and temporally active notation, supporting practices where compositional thinking proceeds through drawing, spatial reasoning, and embodied interpretive knowledge rather than formal symbolic encoding.

The resulting performance practice emphasises coordinated attention, shared orientation, and collectively navigated form. Future work will focus on refining authoring ergonomics, expanding documentation and educational resources, and developing further repertoire that explores the musical implications of executable notation.

More broadly, this approach suggests that digital graphic scoring need not replicate symbolic idioms nor treat the score as a passive display, but may instead operate as a composed environment for performance—one in which notation and behaviour coincide on the same surface.

The Oscilla system, documentation, and example scores are available at [14].

8. REFERENCES

[1] R. Canning, "Real-time web technologies in the networked performance environment," in *Proc. Int. Computer Music Conf. (ICMC)*, Ann Arbor, MI, 2012.

[2] R. Canning, "Interactive parallax scrolling score interface for composed networked improvisation," in *Proc. Int. Conf. New Interfaces for Musical Expression (NIME)*, 2014, pp. 144–146.

[3] R. Canning, "OscillaScore: A Modular Platform for Graphic Notation in Networked Music Performance," in *Proc. Int. Conf. Technologies for Music Notation and Representation (TENOR)*, Beijing, 2025.

[4] I. Xenakis, *Formalized Music: Thought and Mathematics in Composition*, Pendragon Press, 1992.

[5] D. Fober, Y. Orlarey, and S. Letz, "INScore – An Environment for the Design of Live Music Scores," in *Proc. Linux Audio Conf. (LAC)*, Stanford, 2012, pp. 47–54.

[6] N. Didkovsky and G. Hajdu, "MaxScore: Music Notation in Max/MSP," in *Proc. Int. Computer Music Conf. (ICMC)*, Belfast, 2008, pp. 483–486.

[7] A. Agostini and D. Ghisi, "Bach: An Environment for Computer-Aided Composition in Max," in *Proc. Int. Computer Music Conf. (ICMC),* Ljubljana, 2012, pp. 373–378.

[8] T. Coduys and G. Ferry, "IanniX: Aesthetical/Symbolic Visualisations for Hypermedia Composition," in *Proc. Sound and Music Computing Conf. (SMC)*, 2004.

[9] G. Hajdu, "Quintet.net: An Environment for Composing and Performing Music on the Internet," *Leonardo*, vol. 38, no. 1, pp. 23–30, 2005.

[10] C. Hope, A. Wyatt, and L. Vickery, "The Decibel ScorePlayer: New Developments and Improved Functionality," in *Proc. Int. Computer Music Conf. (ICMC)*, Denton, 2015.

[11] A. Yoon, "neoscore: A Python Library for Notating Music in a Graphics-First Paradigm," 2022. Available: https://neoscore.org/

[12] C. Vear, *The Digital Score: Musicianship, Creativity and Innovation*, Routledge, 2019.

[13] J. McCartney, "SuperCollider: A New Real-Time Sound Synthesis Language," in *Proc. Int. Computer Music Conf. (ICMC)*, 1996.

[14] R. Canning, "Oscilla" 2025. Available: https://oscilla.cc

MetaConcert: A Shared VR Audio-Visual Experience Model Reducing User Isolation Through Synchronized 360° Video on HMDs and HOA Playback on a Multichannel Dome

Mauro Cantonetti
Conservatorio di Musica "G.Nicolini"
Piacenza, Italy
mauro.cantonetti@conservatorio.piacenza.it

Paolo Malpeli
Conservatorio di Musica "G.Nicolini"
Piacenza, Italy
paolo.malpeli@conservatorio.piacenza.it

Giuseppe Rizzo
Conservatorio di Musica "G.Nicolini"
Piacenza, Italy
giuseppe.rizzo@conservatorio.piacenza.it

Alessandro Anatrini
Conservatorio di Musica "G.Nicolini"
Piacenza, Italy
Hochschule für Musik und Theater
Hamburg, Germany
alessandro.anatrini@conservatorio.piacenza.it

ABSTRACT

We introduce MetaConcert, a system that integrates a VR head-mounted display with multichannel loudspeaker-dome audio. It employs a dedicated workflow for 360° video capture, Ambisonic audio recording, and dome-oriented rendering. A key component is a synchronization solution using OSC communication between the WebXR video player and SuperCollider for audio rendering.

The system renders third-order Ambisonics, decoded for a multichannel in-room speaker array. Synchronizing 360° video playback in WebXR with multichannel audio in SuperCollider via OSC messages enables a fully immersive, headphone-free experience, making it ideal for shared listening environments.

Framed within the concepts of presence and plausibility [1], we discuss how dome-based listening reduces the isolation typical of HMD use and fosters scenarios of enhanced social presence.

1. INTRODUCTION

The rise of head-mounted displays (HMDs) for consumer use is becoming increasingly prevalent today, introducing users to experiences that were unimaginable just a few years ago, such as Virtual Reality, Augmented Reality, Mixed Reality, and immersive audio/visual content. Alongside this, audiovisual production has evolved, keeping pace with these technologies, and as a result, audio has also received dedicated treatment, such as the use of Ambisonics, Dolby Atmos, and binaurally rendered audio experienced through headphones or built-in speakers on HMDs.

According to Bosman [2], the presence of audio has a consistently positive impact on perceived realism and overall enjoyment of the experience: virtual environments accompanied by coherent sound are perceived as more credible and more pleasant [3, 4]. The relationship between audio and presence—understood as the subjective sense of "being inside" the simulated reality—is more complex. While some studies [3, 5] report a significant increase in presence attributable to sound, others do not observe equally clear effects. This suggests that audio plays a fundamental role in the sense of physical presence and that, together with increased plausibility of the sound field, it contributes to a higher degree of sensory immersion [1]. At the same time, the level of immersion appears to be directly related to factors such as rendering quality, audio-visual congruence, and the type of task performed by the user.

It is also worth noting the relative scarcity of research investigating the role of sound in the social dimensions of VR, such as communication and interaction among multiple participants sharing the same virtual environment [2].

Despite the high degree of technological sophistication characterizing HMD-mediated sensory experiences, these systems introduce non-negligible conditions. In particular, isolation from the external environment caused by the use of headphones for binaural listening [6] limits opportunities for interaction and connection with individuals physically present in the surrounding real space. Although many contemporary HMDs partially address this issue by incorporating small built-in loudspeakers, such solutions do not provide a fully immersive auditory experience. At the same time, the use of Ambisonic loudspeaker arrays entails the acceptance of intrinsic characteristics and constraints of the system itself, including cross-talk, room acoustics, sweet-spot limitations, and localization variability. When these parameters are properly considered and adequately managed, the combination of HMDs and Ambisonic arrays can offer advantages in controlled contexts, albeit subject to constraints related to listening position and room acoustics.

With MetaConcert, the objective has been to provide users with an experience explicitly designed to avoid dynamics of individual isolation from the surrounding environment.

Copyright: ©2026 Mauro Cantonetti et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License 3.0 Unported, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

2. COMPONENTS AND COMMUNICATION PROTOCOLS

The technological stack selected for the development of the system and for achieving the intended objectives consists of technologies that have reached a high level of maturity and offer the advantage of not relying on proprietary or commercial solutions. The choice of components therefore focused on open-source projects and open standards developed by academic institutions or international consortia.

The system is based on the following core technologies:

- **WebXR**: used for the visual content delivery and user interaction layer;
- **SuperCollider**: employed for audio rendering and auditory content playback;
- **OSC** (Open Sound Control): adopted as the communication protocol enabling synchronization and data exchange between the system components.

3. IMPLEMENTATION

The audio-visual content targeted by the system is divided into two separate streams—audio and video—through a post-production pipeline described below.

For the video component, the HMD's web browser is used to load an HTML page delivering content via WebXR, specifically a 360° video. Video rendering relies on an XREquirectLayer (a media layer for spherical video representation in VR applications that uses an equirectangular projection to map a sphere onto a two-dimensional rectangle).

The audio component is instead managed directly by SuperCollider, which plays back a pre-rendered multichannel audio file obtained by decoding the Ambisonic content into 16 discrete channels, each corresponding to a loudspeaker of the dome.

The loudspeakers of the dome are full-range and arranged as follows (Table 1):

Speaker ID	Azimuth	Elevation
1	0°	0°
2	45°	0°
3	90°	0°
4	135°	0°
5	180°	0°
6	225°	0°
7	270°	0°
8	315°	0°
9	0°	45°
10	90°	45°
11	180°	45°
12	270°	45°
13	0°	−45°
14	90°	−45°
15	180°	−45°
16	270°	−45°

Table 1: Loudspeaker layout of the Ambisonics dome

The HMD web browser and the SuperCollider server communicate via the OSC protocol, enabling bidirectional exchange of messages related to content execution (e.g., play, pause, rewind) and ensuring synchronization between audio and video streams. Since, for security reasons, web browsers cannot communicate directly using the OSC protocol, the osc.jslibrary was employed. This solution enables the conversion of OSC messages from WebSocket to OSC packets transmitted over a UDP transport layer through a Node.js proxy server (see Figure 1).

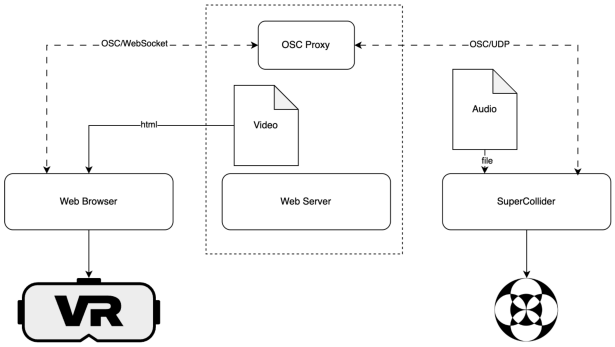


Figure 1: System Architecture

3.1 Web Server

A central role is played by a web server installed on a PC connected to a local area network (LAN) via Ethernet. This server is responsible for providing the following resources and services:

- the HTML page delivering WebXR content;
- the 360° video asset;
- a proxy service for converting the transport layer of OSC messages from WebSocket to UDP and vice versa.

The communication flows between the web server and the other components of the system are illustrated in the diagram shown in Figure 2, which specifies the corresponding communication protocols, the ports in use, and the direction of information exchange.

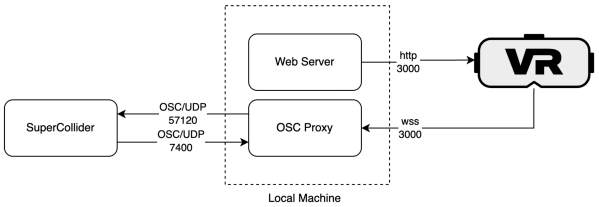


Figure 2: Communications

3.2 SuperCollider

The SuperCollider software, which in this implementation runs on the same PC hosting the web server, serves a dual purpose: it performs multichannel audio playback synchronized with the video content and provides a basic interface for controlling video playback.

At system startup, the SuperCollider software preloads the entire audio content into a dedicated buffer and enters a listening state, awaiting a start trigger signal sent by the HMD. This strategy minimizes audio playback start latency, making it perceptually negligible and temporally predictable, as SuperCollider execution upon receiving the trigger is effectively immediate.

In the implemented system, the HMD also continuously transmits the current temporal position of the video content to SuperCollider, enabling potential correction operations for accumulated audio–video drift. No instrumental measurements of drift were conducted; however, audio–video synchronization was perceptually verified for content durations of up to approximately 10 minutes. Further investigations for longer durations were not performed, as such scenarios fell outside the scope of the present work.

Experimental tests additionally revealed that latencies associated with the delivery of OSC commands cannot be predicted with high precision, as they are strongly dependent on network congestion. Empirically, these latencies were observed to be on the order of a few milliseconds (approximately 5–10 ms).

3.3 OSC Address Pattern

The OSC messages exchanged between the HMD and SuperCollider are bidirectional and can be classified according to their direction of transmission (see Figure 3).

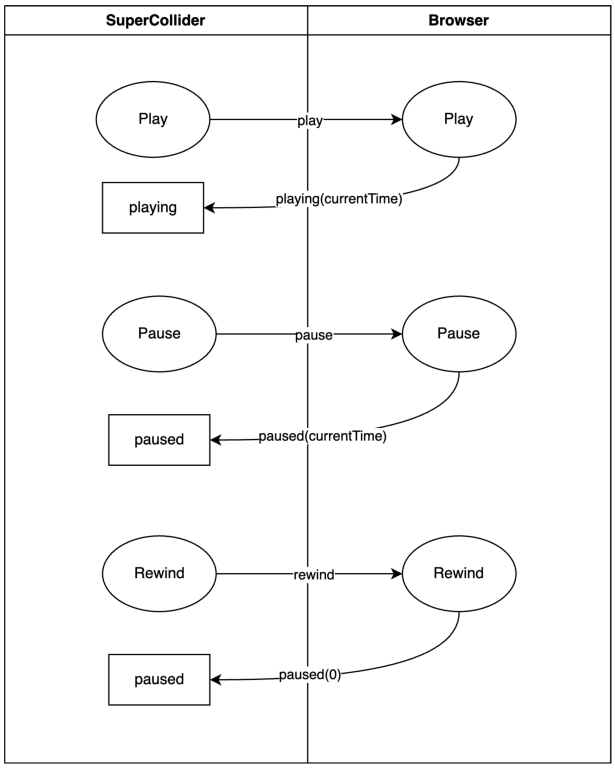


Figure 3: OSC messages sequences

4. USE CASE

In order to test the system, an original audiovisual production was created for a musical piece specifically composed for this purpose and involving eight musicians. The piece, entitled *"Quiet"*, was composed by Daniele Delogu.

Comando	Descrizione
/play	instructs the browser to start video playback
/pause	instructs the browser to stop video playback
/rewind	instructs the browser to reposition the video to its beginning

Table 2: from SuperCollider to WebXR Browser

Comando	Descrizione
/playing(time)	communicates the current video playback time to SuperCollider
/paused(time)	communicates the paused state of the video to SuperCollider
/pose(yaw, pitch, roll)*	communicates the user's head orientation to SuperCollider

* The user's head orientation (pose), expressed as yaw, pitch, and roll, is currently not utilized by the system but could be employed to enable more interactive XR experiences.

Table 3: from WebXR Browser to SuperCollider

The work was recorded using a 360° video camera, an Ambisonic microphone, and additional spot microphones placed in close proximity to the performers. Spherical video was captured using an Insta360 One RS 1-Inch Leica camera. The recorded files, originally in the proprietary `.insv` format, were subsequently converted into equirectangular `.mov` format using the proprietary software Insta360 Studio. These files were then edited to trim the desired start and end points, and additional graphics and visual effects were applied using DaVinci Resolve.

Ambisonic and spot microphone recordings were processed, synchronized, and mixed within the DAW Reaper. The Ambisonic microphone track was upmixed from first to third order using the Compass Upmixer plugin by Sparta Audio (Figure 4), employing ACN channel ordering and SN3D normalization. Mono spot microphone tracks were instead directly encoded into third-order Ambisonics with SN3D normalization using the MultiEncoder plugin by IEM (Figure 5). The sources were positioned on the azimuth plane in accordance with the spatial placement of the musicians as shown in the video.

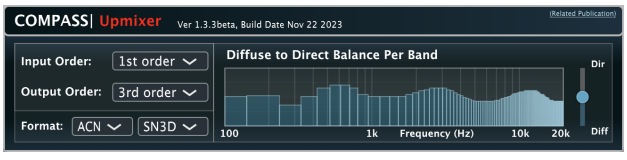


Figure 4: Sparta Audio Compass Upmixer Plugin

For decoding on the master track—configured with 16 discrete output channels—the IEM AIIRA Decoder plugin was used, set to third-order Ambisonic decoding with maxrE weighting, SN3D normalization, and a loudspeaker layout corresponding to the configuration described previously (Figure 6). As a result, audio playback does not

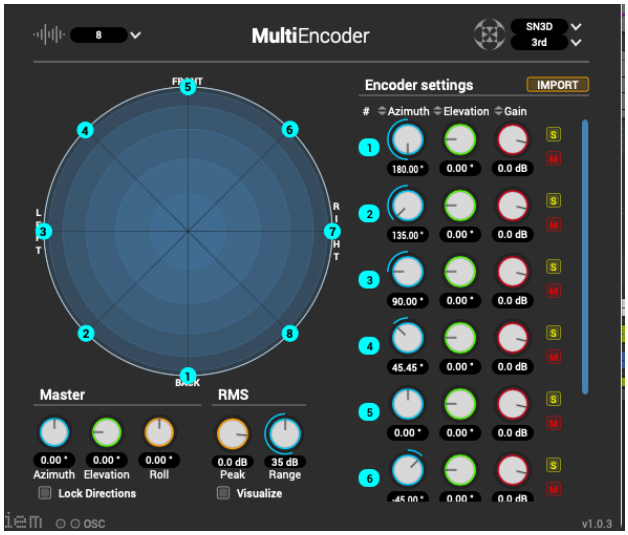


Figure 5: IEM MultiEncoder Plugin

require any additional real-time decoding within the MetaConcert system during the user experience.

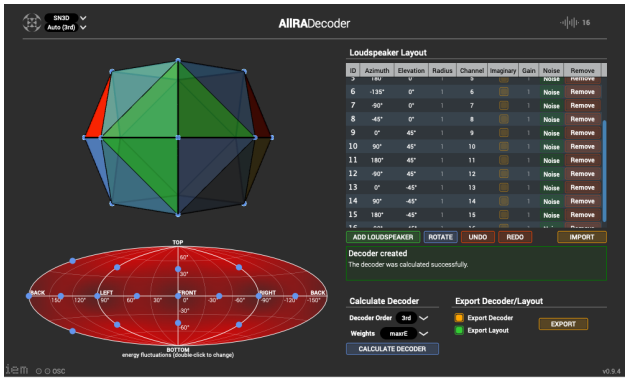


Figure 6: IEM AIIRA Ambisonics Decoder

The video produced in DaVinci Resolve was subsequently imported into Reaper in order to achieve the desired audio–video synchronization.

The final output files consisted of:

- `video.mkv`: H.264, 6K resolution, 30 fps;
- `audio.wav`: 16 discrete channels, 44.1 kHz, 24-bit.

Once the files were loaded into the system, the user was positioned at the center of the Ambisonic loudspeaker dome (Figure 7) in a seated posture while wearing the HMD. This positioning was required because the sound field reconstruction assumes a central listening point; any displacement from this position introduces phase, amplitude, and directional errors that compromise the spatial accuracy of the sound scene [7]. The seated posture was additionally chosen because alterations in visual perception and balance induced by VR experiences may impair postural stability, increasing the risk of loss of balance, falls, or collisions with the surrounding physical environment [8].

In MetaConcert, audio rendering remains anchored to the physical geometry of the loudspeaker dome (world-fixed).

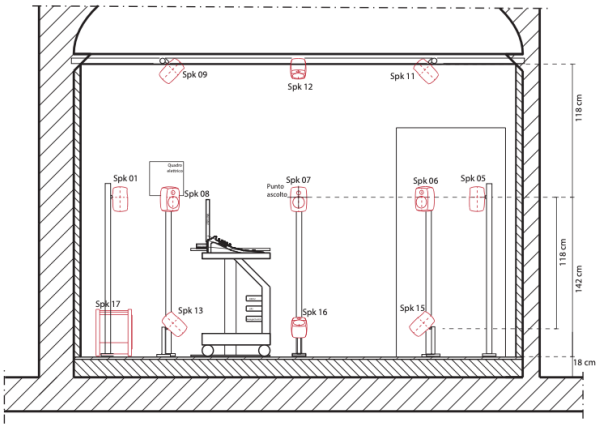


Figure 7: Ambisonics Dome

To ensure audio–video coherence, the "front" of the video is aligned with the "front" of the dome by means of the HMD's reset view function. For this reason, it is not necessary to rotate the audio scene according to head movements. If the user performs a reset view operation during playback, a new calibration is required.

Once these conditions are met, the system operator can initiate playback from the host computer.

5. CONCLUSION

Drawing inspiration from the work of Bosman *et al.* [2], this study outlines directions for further development of the MetaConcert system that place the social contribution of audio at the center of HMD-based experiences. In particular, future work envisions a multi-user scenario in which the same immersive content can be experienced simultaneously by multiple users within the same loudspeaker dome.

The implementation of such a scenario requires a more advanced audio–video synchronization system capable of ensuring both temporal and perceptual coherence across multiple HMDs operating concurrently. From this perspective, future research will focus on strengthening the synchronization architecture as a key element for the success of shared immersive experiences.

In parallel, in order to improve system accessibility and enhance the operator's workflow, it would be desirable to develop a unified software solution executable directly on a standard computer. Such a solution would enable the initiation and management of the experience without requiring complex installations or configuration procedures. Further improvements in accessibility could also be achieved by supporting listening configurations other than the loudspeaker dome.

In this sense, future developments of the project can be articulated along two main and coherent directions: on the one hand, the extension toward multi-user scenarios supported by robust synchronization mechanisms; on the other hand, the standardization and portability of the software infrastructure, while maintaining a strong focus on the immersive dome context and the shared experience it enables.

6. REFERENCES

[1] H. Lee, "A conceptual model of immersive experience in extended reality," *Computers in Human Behavior Reports*, p. 100663, 2025.

[2] I. d. V. Bosman, O. O. Buruk, K. Jørgensen, and J. Hamari, "The effect of audio on the experience in virtual reality: a scoping review," *Behaviour & Information Technology*, vol. 43, no. 1, pp. 165–199, 2024.

[3] A. C. Kern and W. Ellermeier, "Audio in vr: Effects of a soundscape and movement-triggered step sounds on presence," *Frontiers in Robotics and AI*, vol. 7, p. 20, 2020.

[4] S. Serafin, M. Geronazzo, C. Erkut, N. C. Nilsson, and R. Nordahl, "Sonic interactions in virtual reality: State of the art, current challenges, and future directions," *IEEE computer graphics and applications*, vol. 38, no. 2, pp. 31–43, 2018.

[5] R. Nordahl and N. C. Nilsson, "The sound of being there: presence and interactive audio in immersive virtual reality," 2014.

[6] C. Oh, F. Herrera, and J. Bailenson, "The effects of immersion and real-world distractions on virtual social interactions," *Cyberpsychology, Behavior, and Social Networking*, vol. 22, no. 6, pp. 365–372, 2019.

[7] J.-D. Polack, "Ambisonics encoding of other audio formats for multiple listening conditions."

[8] R. S. Kennedy, N. E. Lane, K. S. Berbaum, and M. G. Lilienthal, "Simulator sickness questionnaire: An enhanced method for quantifying simulator sickness," *The international journal of aviation psychology*, vol. 3, no. 3, pp. 203–220, 1993.

Sonic Urgency: Exploring Perceptual, Sociopolitical, and Participatory Dimensions of Spatial Listening

Teresa Carrasco Garcia

Bern Academy of the Arts. Bern University
of Applied Sciences
teresa.carrasco@hkb.bfh.ch

ABSTRACT

This paper explores spatial listening as a multidimensional practice linking perception, phenomenology, and sociopolitical discourse. It outlines psychoacoustic foundations of sound localization and traces key listening theories—from reduced listening and acoustic ecology to spectromorphology and spatial dramaturgy—framing listening as an active, interpretive process. It then examines phenomenological, participative, and political aspects, proposing spatial listening as an embodied, situated, and relational practice, and calls for expanded listening models suited to contemporary sonic environments.

1. INTRODUCTION

Across different disciplines, including electroacoustic music, sound art, and psychoacoustics, artists and researchers have developed a variety of frameworks for understanding how we listen and, more specifically, how listening constructs space. These approaches offer not just tools for analysis or graphical representation, but also strategies for creation, orientation, and transformation within sonic environments. In this sense, listening can be regarded as a method of engaging with space and society as a dynamic, relational medium. The following text will present a comprehensive contextualization of theoretical frameworks, artists, and research findings, with the aim of identifying main references, authors, and topics that are linked to the perception, creation, and practice of musical space. A select number of the author's own pieces will be considered and analyzed in brief in order to illustrate the interconnection between the theoretical framework and her own artistic practice.

Copyright: © 2026 Teresa Carrasco García. This is an open-access article distributed under the terms of the Creative Commons Attribution License 3.0 Unported, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

2. PERCEPTUAL DIMENSIONS

2.1. Spatial Hearing and Localization: Dickreiter, Blauert

To understand soundspace as a compositional and experiential field requires an inquiry into both the qualitative dimensions of sound in space and the conditions of listening that bring them into being. At the foundation of spatial sonic experience lies the architecture of the listening space itself. As Dickreiter (1997) notes, several key acoustic parameters determine the qualitative nature of sound in an environment, including: Room shape and geometry; Volume and scale; Surface materials and their absorption coefficients; Reflection patterns and reverberation times. These physical elements contribute directly to what Dickreiter calls *Raumeindruck* (spatial impression), *Hörsamkeit* (audibility), and *Durchsichtigkeit* (transparency)—three interrelated perceptual categories that describe the clarity, spatial character, and intelligibility of complex auditory environments. These parameters frame the conditions under which sound behaves and is perceived. [1]

Jens Blauert's foundational research into spatial hearing (1997) provides a detailed scientific framework for understanding how human auditory perception constructs spatiality from sound. His investigations explain how we are able to locate, identify, and navigate auditory events in three-dimensional environments, drawing on physiological mechanisms such as interaural time differences (ITD), interaural level differences (ILD), spectral filtering by the pinnae (outer ear), as well as early reflections and reverberant cues provided by architectural surfaces. [2]

Moreover, Blauert's concepts provide a bridge to the fields of immersive media, virtual environments, and interactive sound installations, where spatial hearing becomes the primary interface through which the artwork is accessed and understood. Composers and sound artists working in these fields might understand the auditory system itself as an instrument, one that can be played through precisely constructed spatial gestures, exploiting known perceptual thresholds and ambiguities. These mechanisms are deeply embedded in evolutionary functions—supporting orientation, environmental awareness, and human interaction—but their implications extend far beyond biological utility, providing a conceptual and

technical framework for soundspace composition, enabling artists to choreograph listening across space and time.

2.2. Historical perspectives on auditory perception: Schaeffer, Thoresen, Roy, McAdams, Smalley, Nyström et al.

Pierre Schaeffer, in his *Traité des objets musicaux* (*Treatise on Musical Objects*, 1966), articulated a foundational framework for understanding auditory perception, distinguishing four modes of hearing: *ouïr* (to hear), *entendre* (to perceive or recognize), *écouter* (to actively attend to), and *comprendre* (to understand or interpret). This progression represents a deepening engagement with sound, moving from passive reception to active cognitive interpretation. These distinctions are not merely linguistic, but also underpin Schaeffer's broader theory of listening as both a perceptual and compositional act. [3]

Central to Schaeffer's theory is acousmatic listening, a term inspired by the ancient Pythagorean practice where students listened to a teacher speaking from behind a curtain. In Schaeffer's formulation, acousmatic listening involves hearing a sound without seeing its source, shifting focus from causality and function to its intrinsic, sonic qualities—texture, shape, duration, and dynamic evolution. From this perspective emerges the *objet sonore* or *sonorous object*: a sound perceived as an autonomous perceptual event rather than a signal representing an external source.

Building on Schaeffer, other authors have developed analytical systems for music perception and electroacoustic composition. Lasse Thoresen developed his spectromorphological analysis of sound objects from Schaeffer's typomorphology, using diagrammatic representations created with a specially designed font, "Sonova." This system serves both to analyze and graphically represent electroacoustic music, and as a conceptual and compositional tool for sketching new works. [4]

Similarly, Stephan Roy, in *Analysis of Electroacoustic Music: Models and Proposals*, provides analytical models that focus on listening, perception, and sound morphology, offering methodological tools to describe, represent, and interpret sonic structures while linking analysis to compositional practice. [5] Research by McAdams, Noble, and Bonin further investigates perception of electroacoustic music, exploring temporal ranges and timescales. [6]

Denis Smalley extended Schaeffer's ideas by developing spectromorphology, a framework analyzing how the spectral (frequency-based) and morphological (shape-based) characteristics of sound evolve over time. [7]

While Schaeffer emphasized the perceptual isolation of sound objects from their sources, Smalley focused on how these objects behave within the listener's experiential field—how they emerge, expand, decay, and suggest spatial presence. He introduced the concept of spatial articulation, describing how sound is perceived to occupy space—not just through physical placement, but through impressions of movement, distance, density, diffusion, and trajectory. [8]

In this way, space becomes intrinsic to the sound itself, rather than an external addition. Smalley distinguished between source-bonded listening (sound linked to an identifiable physical source) and spectromorphological listening (sound freed from representational anchoring). Source-bonded listening relies on environmental cues, whereas spectromorphological listening treats space as a sculptural, perceptual terrain shaped by internal sonic dynamics. Through this lens, sound does not merely exist in space, but it actively generates spatial relationships through spectral behavior and morphological flow. Smalley's framework integrates gesture (directional, intentional sonic movement) and texture (non-directional, diffuse sonic fields), enabling a spatially and temporally dynamic approach to composition and analysis.

Erik Nyström extended Smalley's theoretical foundation into the artistic and compositional realm with the concept of spatial texture, which reconceptualizes sound as a volumetric, distributed material rather than as discrete points. [9] Spatial texture emphasizes sonic densities, layered surfaces, dynamic voids, and fluid flows, constructing immersive and manipulable sonic topologies. This perspective encourages composers to foreground the movement, transformation, and tension of sound fields over time rather than merely placing sounds in fixed positions. Nyström's approach opens up possibilities for architectural and imagined soundspaces, where sound is treated analogously to visual texture or sculptural volume. Composers can create immersive atmospheres, tactile listening experiences, and perceptual depth through trajectories, gradients, and morphologies of sound. These concepts align seamlessly with contemporary immersive audio technologies and multichannel diffusion systems, which allow for the dynamic distribution of sound in three-dimensional environments, enhancing spatial perception and listener engagement.

2.3. Phenomenological perspectives on auditory perception: *Verzerrte Abweichungen*

When composing my piece *Verzerrte Abweichungen* (2025), my intention was to create an acousmatic work that interrogates the auditory residues of obsolete technologies while exploring their potential as acousmatic objects, spectromorphological entities, and spatial textures. The piece draws on recordings of 1980s machines—printers, fax machines, typewriters, analog cameras, and early arcade games—which I treated not just as nostalgic symbols of a bygone era, but as raw sonic matter to be transformed and recontextualized within a multichannel electroacoustic framework. [10]

In approaching these materials, I was guided by Pierre Schaeffer's notion of the *objet sonore* as articulated in his *Traité des objets musicaux*. The sonic residues of these machines reveal their potential most fully when detached from their causal origins and apprehended through what Schaeffer calls *écoute réduite*. For me, the clatter of the dot matrix printer or the whirring of a camera motor became sound objects, malleable and open to reshaping. Their timbral and morphological properties—the attack, sustain, spectral density, and decay—provided the vocabulary from which the work could be constructed.

Spatialization was also a strong compositional element. Working with eight channels enabled me to experiment with what Smalley refers to as *spatial articulation*: the dynamic movement of sound through space as an expressive parameter in its own right. Rather than statically positioning the materials, I sought to generate shifting topographies in which sounds traverse, collide, and envelop the listener. Here I found resonance with Erik Nyström's concept of *spatial texture*. I conceived the diffusion not simply as placement but as a fabric of sonic motion—a texture that could immerse the audience in a constellation of vibrating machines, simultaneously familiar and estranged.

While I deliberately avoided nostalgic treatment, I recognize that these sounds inevitably awaken auditory memories in listeners, evoking an era when such technologies were woven into daily life. The mnemonic force of the sound object, which Schaeffer had already acknowledged, became a crucial part of the work's affective dimension. The inclusion of 8-bit arcade-game sounds was especially significant: these playful bleeps and pulses, juxtaposed with mechanical timbres, underscore the temporal layering of different technological cultures. Together, they chart a spectromorphological spectrum of technological temporality—analog obsolescence interwoven with digital emergence.

The premiere of *Verzerrte Abweichungen* at the Kontakte Festival in Berlin, curated by Miriam Ackermann and Nicola Hein, provided an ideal context for presenting this exploration. The festival's commitment to intermedia and experimental practices situates the piece within a broader discourse of media archaeology, immersive art, and electroacoustic innovation. For me, this context reinforced the idea that working with outdated sonic materials is not a backward-looking gesture but an inquiry into how sound can carry cultural meaning beyond its technological function.

In composing *Verzerrte Abweichungen*, I sought to synthesize Schaeffer's phenomenology of the sound object, Smalley's spectromorphological and spatial frameworks, and Nyström's conception of spatial texture. The work became a way of asking how obsolete technologies, when reframed acousmatically, can reveal themselves as both physical phenomena and carriers of memory. It is precisely in their distortion and deviation from their original function that they become musically potent: no longer machines of the past, but sonic presences in the here and now.

2.4. Ecological Space Listening: Schafer, Krause, Schabert, Schäfer, Carrasco.

The ecological spatial field becomes explicit in the work of R. Murray Schafer, who coined the term *soundscape* to describe the totality of acoustic environments as they are perceived and understood by individuals or communities. [11]

In contrast to the abstract spatialities of Schaeffer and Smalley— where space is formed by internal perceptual and morphological dynamics—Schafer reintroduces the listener into a concrete ecological context, where space is constructed through listening but also socially and environmentally situated. His approach provides an essential bridge between compositional and environmental theories of sound, grounding spatial listening in lived, material reality. For Schafer, sound is not simply perceived; it is inhabited. Listening becomes a critical means of navigating, understanding, and evaluating the spatial qualities and health of the world around us.

Key to Schafer's theory is the distinction between *hi-fi* and *lo-fi* soundscapes. In a hi-fi environment —such as a quiet rural village or open field—sonic details are discernible, distances are perceptible, and the spatial field is clearly articulated. Each sound contributes to an overall acoustic clarity that enables orientation and deep listening. Conversely, in a lo-fi environment, typically urban and industrialized, background noise saturates the acoustic field, masking spatial cues and compressing sonic perspective. This results in a blurring of spatial boundaries, where the ear must work harder to parse meaningful sounds from an undifferentiated sonic haze.

Schafer's concern here is not simply aesthetic but ecological: the shift from hi-fi to lo-fi reflects a broader degradation of the acoustic environment and, by extension, our sensory engagement with space. To counteract this loss of spatial acuity, Schafer introduces the practice of *ear cleaning*—a kind of auditory reeducation aimed at restoring the listener's capacity to attend to subtle sonic details. Ear cleaning involves developing a heightened sensitivity to the acoustic textures of everyday life, training the listener to hear spatial distinctions, locate sounds accurately, and reawaken an embodied awareness of sonic space. In this sense, ear cleaning is not only a perceptual technique but also an ethical stance—an invitation to listen more attentively and responsibly to one's environment.

A further concept introduced by Schafer is the concept of *soundmark*, modeled after the visual term "landmark," to designate sounds that are unique, locally significant, and culturally meaningful within a particular acoustic environment. *Soundmarks* might include the call of a specific bird species, the bells of a local church, or the hum of a traditional marketplace. These sounds are deeply embedded in the identity of a place; they define spatial experience not just through localization or proximity, but through memory and social resonance. Their erosion or disappearance—often a consequence of urbanization and noise pollution—marks not just the loss of sonic diversity but the erasure of spatial and cultural identity.

Crucially, Schafer's work reframes spatial listening as a relational and critical act. Listening is a way of intervening in the world—a method of diagnosing and perhaps reshaping the acoustic and spatial conditions in which we live. Schafer's acoustic space is a dynamic field shaped by human behavior, architecture, technology, and power. The listener, through practices such as ear cleaning and soundscape composition, is positioned as an active agent capable of perceiving and contesting this field.

Moving forward, we may encounter Bernie Krause's concept of biophony, defined as the collective acoustic signature produced by all living organisms within a specific habitat. In this context, ecological listening emerges

as a method for assessing the health, diversity, and balance of an ecosystem. Within a thriving habitat, each species contributes its own voice within a time-frequency niche, thereby producing what Krause describes as a kind of natural orchestration. This acoustic layering is not random; it reflects long-evolved strategies of coexistence and communication. [12]

However, as ecosystems collapse due to climate change, deforestation, urbanization, and extractivism, this sonic balance begins to deteriorate. The biophony becomes fragmented, masked by anthropogenic noise or silenced altogether. What once was a dense and polyphonic soundscape transforms into a distorted, thinning acoustic field—a sonic expression of ecosystemic rupture.

In this context, ecological space listening is understood as a political and ethical act. It challenges the human-centered tendency to filter out non-human sounds and instead invites us to attune to what is fading or missing. The absence of voices — the silencing of frogs, birds, and insects — becomes as significant as their presence. Ecological listening thus operates as both sensory knowledge and a form of mourning, attuning us to the losses that define the contemporary era. Furthermore, this approach offers an alternative to the dominance of visual perception in environmental discourse.

Dorothee Schabert (*1952) engages with her piece *Hörlandschaften* (auditory landscapes) to explore how sonic environments operate as cultural and sensory spaces, where listening becomes an active mode of spatial navigation and interpretation. The concept of *Hörlandschaften* serves as a particularly evocative metaphor for linking sonic architectures with the topology of soundscapes understood as audible landscapes. This metaphor facilitates a nuanced understanding of how sound organizes and shapes spatial experience as a constructed auditory environment. [13]

Similarly, Sabine Schäfer's work, notably *TopoPhonien* and her site-specific installation *Bats 'n' Insects*, exemplifies this approach by foregrounding the interplay between ecological sounds and spatial form. Schäfer's practice underscores how soundscapes function as dynamic topologies, revealing hidden layers of acoustic activity that articulate the environment in ways both perceptually and conceptually rich. [14]

Together, these perspectives highlight the productive intersections between sound, space, and landscape, positioning auditory phenomena as fundamental components of architectural and ecological discourse. My piece *Un giardino sonoro* (2025) imagined a space where technology and nature could meet in a continuous dialogue. The installation part of it lives entirely from the sun: its photovoltaic panels provide all the energy needed to sustain the system, so it is autonomous, self-sufficient, and silent in its physical presence. All that it gives back is sound. The work listens constantly. Microphones capture the surrounding soundscape—traffic, voices, wind, the calls of birds—and through real-time analysis of the soundscape, new sonic textures are resynthesized, redefining the sonic space into something new. I was especially interested in two extremes: the harshness of noise pollution and the delicacy of birdsong. By filtering and reworking both these elements, I let the city's disturbances and Nature's fragility coexist and transform into a continuously evolving sound texture in dialog with the instrumental solos and duos. [15]

For me, the composition is not simply a reproduction of reality, but a transformation of it: the environment becomes its own composer, shaping the music through its changes. What emerges is a garden of sound— triggered by the players and the soundscape, sometimes dense and restless, sometimes open and fragile—always alive, always reacting. Standing within it, you are invited to listen differently: to hear what surrounds you, to recognize how it shifts, and to discover beauty even in noise.

When I was conceiving the installation *Un giardino sonoro*, I was deeply influenced by the listening practices and theoretical frameworks developed by R. Murray Schafer and later expanded by Bernie Krause, Sabine Schäfer and Ursula Damm. Schafer's notion of the *soundscape* as a key to understanding the acoustic identity of a place shaped my approach. I wanted to reveal it as a living presence. Schafer spoke about *lo-fi* and *hi-fi* soundscapes, about the masking effects of urban noise, and about the need to cultivate what he called *clairaudience*—a clear and sensitive listening. *Un giardino sonoro* makes possible that kind of listening in many ways: it makes audible the interplay of elements often ignored, turning them into an aesthetic and perceptual experience.

3. SOCIOPOLITICAL DIMENSIONS

3.1. Listening as a social and political possibility, and the listening imagination: Vogelin, Truax

Salomé Voegelin's book *Listening to Noise and Silence: Towards a Philosophy of Sound Art*, brings a vital phenomenological and political perspective to the discussion of spatial listening. [16]

Voegelin argues that sound is not simply an object to be perceived or a field to be mapped, but an ephemeral and co-constituted event—an event that arises through listening and, crucially, through the listener. In this view, space becomes the temporal, relational, and subjective act of listening itself. Voegelin challenges the visual and cartesian assumptions that often underlie Western models of space—those that prioritize distance, objectivity, and stasis. She posits that these visual logics tend to reduce space to something that can be surveyed, measured, and owned, reinforcing hierarchies of control and detachment. In contrast, listening enacts a radically different mode of spatial engagement: immersive, participatory, and unstable. When we listen, we enter into proximity with sound. We do not observe from a distance; we are implicated. This proximity reconfigures spatial experience as something intimate and situated, where boundaries are porous and meanings are emergent.

In this sense, listening is both epistemological and ontological—it is a way of knowing the world, but also of *being* in the world. Sound unfolds in time and only in relation to a listener; it cannot be paused, objectified, or entirely possessed. For Voegelin, this fleeting quality is

not a limitation but a generative possibility. It means that listening can produce new spatial imaginaries—spaces that are provisional, relational, and open-ended. These are not the monumental spaces of architectural permanence, but the lived, shifting topographies of attention and encounter.

Crucially, Voegelin frames listening as a *political* act. Because it resists the visual urge to fix and dominate, listening becomes a site of resistance—a way to challenge normative structures of spatial order and to make room for otherness, ambiguity, and change. The body that listens is not detached or neutral; it is marked by history, culture, and position. Spatial listening, in Voegelin's view, is an embodied negotiation with space—a way of engaging with the world that foregrounds vulnerability, presence, and responsiveness.

This approach to listening expands the idea of spatial composition beyond the transformation of sound in physical environments or immersive technologies. It shifts attention to the act of listening itself as a compositional gesture—an ongoing, performative process through which space is shaped by those who listen. Here, sound does not merely fill or represent space; it calls space into being. And that space is inherently unstable, always contingent upon who is listening, how they are listening, and under what conditions.

Voegelin's phenomenology aligns productively with Pauline Oliveros's notion of *deep listening*, in which the cultivation of attentional depth opens new dimensions of sonic and spatial awareness. Both thinkers emphasize listening as a practice of radical receptivity—one that dissolves the boundary between listener and environment, subject and object. Together, they invite us to reconsider spatial listening not as a passive sensory mode but as a situated, intentional act with aesthetic, philosophical, and political implications. In this reorientation, space is not pre-existing but emergent, created through the act of listening and shaped by its temporal, affective, and social dimensions.

Building on the phenomenological and political framing of Salomé Voegelin, the notion of spatial listening as an emergent, relational act finds further development in the work of Barry Truax, particularly as explored in *The Listening Imagination* (Oxford University Press). Truax extends Schafer's ecological perspective by emphasizing not only how we listen to environments, but how environments listen back—how sonic feedback loops, systems of interaction, and dynamic relationships shape both perception and spatial understanding. For Truax, listening is not a one-way reception but a dialogic process, where spatial configurations are continuously mediated by technologies, bodies, and social conditions. [17]

Truax's concept of *acoustic communication* suggests that sound operates as a connective tissue— one that binds individual perception with collective spatial organization. Within this framework, spatial listening becomes both personal and systemic: it situates the individual within a larger network of sounds, meanings, and interactions. The act of listening thus participates in a larger process of spatial formation, one that includes both human agency and technological mediation. In soundscape composition, for instance, Truax shows how recorded and composed environments don't merely depict space—they reimagine it. The studio becomes a place of spatial invention, and the listener becomes an active co-creator of space through imagination and memory.

In this view, spatial listening is an act of orientation, a response to sonic stimuli; it is a creative, interpretive act that bridges perception and invention, material sound and imagined space. This convergence of ecological, phenomenological, and compositional perspectives opens the door to a more expansive notion of space: one that is not only heard but continually made through attention, resonance, and the ethics of listening itself.

3.2. Listening Space as a Collective Practice: Oliveros

Listening to space also reveals itself as a practice: a mode of attention, awareness, and relational engagement that unfolds over time and through the body. Few have articulated this dimension more profoundly than Pauline Oliveros, whose concept of *Deep Listening* reorients listening from a perceptual act to a lifelong, embodied discipline. Developed over decades of teaching, performance, and reflection, *Deep Listening* invites an expanded awareness of sound, space, and self—one that blurs the boundary between art and life, composer and listener, inner and outer worlds. [18-22]

Oliveros's Deep Listening emerges from her commitment to improvisation, sonic meditation, and environmental engagement. She distinguishes between *hearing* (the passive registration of sound) and *listening* (the active process of focusing attention). In her writings, particularly *Quantum Listening: From Practice to Theory* and later essays such as "Improvising with Spaces" (2007), Oliveros describes listening as inherently multidimensional—capable of encompassing not only physical and acoustical aspects of space but also emotional, social, and energetic dimensions. To listen deeply is to open oneself fully to the field of sound: its subtleties, resonances, distances, textures, and transformations. Space, in this mode, is not a neutral stage but a dynamic participant in the sonic experience.

Oliveros's 1975 piece *Collective Environmental Composition* exemplifies this ethos. Designed as a participatory listening and sounding event, it invites groups of performers and listeners to engage with an environment—urban, rural, or otherwise—through both sounding and listening. This piece cultivates attentiveness to the sonic potential of a place, asking participants to respond to its acoustic affordances, temporalities, and resonances. In doing so, it breaks divisions between composer, performer, and audience, enacting a form of communal spatial improvisation where space itself becomes the score. Central to this practice are Oliveros's Deep Listening exercises—guided protocols that sharpen awareness of sound, silence, breath, and body. These exercises, often deceptively simple, ask participants to attend to the act of

listening as a multilayered and self-reflective process. For example, one exercise instructs listeners to "listen to the sound of your listening," turning the focus inward toward the feedback loop between perception and awareness. Others involve walking meditations, sonic interactions with architectural spaces, or the vocalization of sounds found in the surrounding environment. Each exercise positions listening as a practice of presence, connection, and transformation.

Importantly, Deep Listening is also a spatial practice. Oliveros emphasizes the idea of listening *to space* and *through space*. In her text "Improvising with Spaces," she encourages performers to treat architectural and environmental acoustics as collaborators—factors that shape phrasing, resonance, and timing. Rather than controlling space, the Deep Listener adapts to it, explores it, and learns from it. The room, the hall, the landscape, or the corridor becomes an improvisational partner, offering reverberation, delay, diffusion, and reflection as materials for sonic dialogue. Listening space becomes both a compositional or aesthetic concept and a practice of care as well—toward the environment, others, and oneself. The Deep Listening practitioner cultivates what Oliveros calls a "heightened state of receptivity," which opens up the possibility of healing, empathy, and community.

In framing listening space as a practice (2007), Oliveros offers a model of spatial engagement that is improvisational, participatory, and deeply attuned. It is a practice that resists the objectification of space and instead embraces its fluid, relational, and experiential qualities. As such, Deep Listening stands as both a methodology and a philosophy: an invitation to compose, inhabit, and reimagine space through sound—and to do so with humility, generosity, and presence. Oliveros's concept of *multidimensional listening* encompasses physical, emotional, and social dimensions of space, offering a profoundly expanded understanding of how space is perceived and performed through sound.

3.3. Situated listening. Questioning spatial norms, hierarchies and exclusions. Biserna

Elena Biserna's curatorial and theoretical research extends Oliveros's spatial listening practices into the realm of movement and urban experience. In her project *Walking from Scores*, Biserna explores how listening unfolds through walking—not merely as a way to hear sound in place, but as a generative, embodied mode of composing space. The scores featured in this project often involve simple instructions that guide listeners through cities, parks, or buildings, transforming mundane movement into aesthetic and critical acts. [23] Biserna's approach foregrounds the performative and political dimensions of situated listening. Drawing from feminist and decolonial perspectives, her work treats walking not only as a method for exploring soundscapes but also for questioning spatial norms, hierarchies, and exclusions. Like Oliveros, she reimagines the score not as a fixed script, but as a proposition—an invitation to notice, to respond, and to co-create. Where Oliveros listens with the space, Biserna listens with the public space; both position listening as a spatial practice that is contingent, affective, and always in motion.

What emerges from this dialogue between Oliveros and Biserna is a model of listening that is inherently spatial and relational. It is improvisational in nature, unfolding through the lived contingencies of site, weather, traffic, architecture, and embodiment. It challenges the visual dominance of spatial thinking by offering an aural, experiential counterpoint—one in which meaning arises from embeddedness and interaction. Ultimately, listening space as a practice involves cultivating a heightened sensitivity to how we inhabit and shape space through our sonic attention. It asks how we move through space, how we listen with others, and how we might attune ourselves to the political and ecological dynamics of our environments. Whether through Oliveros's Deep Listening meditations or Biserna's ambulatory scores, this practice foregrounds the ethical, aesthetic, and spatial potential of listening as a mode of engagement with the world.

4. PARTICIPATIVE DIMENSIONS

4.1. Sonic Urgency: Creating Sonic Spaces for Transformation through Participation

In an era of intensifying ecological crises, I have found it increasingly vital to turn to sound not merely as a medium of aesthetic articulation, but as a powerful vehicle for ecological awareness and political intervention. Rooted in the intertwined disciplines of sound ecology—which examines the relationships between living organisms and their sonic environments—and bioacoustics—which studies the production and reception of sound among biological entities—my compositional practice seeks to explore how sound registers ecological transformations long before they become visible. It is within this context that I developed *KlimaAlarm* (2020), a participatory, multimedia composition that merges bio-acoustic methods with sonic activism, intended to provoke both critical reflection and political consciousness in relation to climate change. [24]

KlimaAlarm was commissioned for the Bern Music Festival, which took place from September 2 to 6, 2020, across multiple venues in the city. The festival was thematically structured around three key climate-related challenges: biodiversity, inequality, and transformation. I was invited by curators Thomas Meyer and Andri Probst to contribute a work for the section focused on transformation, specifically addressing potential responses to the climate crisis. I proposed a collaborative, public-space project—a democratically developed, site-specific performance involving artistic and scientific communities.

One of my primary aims in developing this work was to foster transdisciplinary collaboration. Thanks to the support of the festival, I had the opportunity to work closely with Prof. Dr. Peter Messerli, Director of the Centre for

Development and Environment (CDE) and Professor of Sustainable Development at the University of Bern. Our collaboration allowed the project to unfold at the intersection of artistic expression and scientific inquiry.

I conceived *KlimaAlarm* as a literal and metaphorical alarm—a sonic manifestation of the planetary emergency wrought by anthropogenic climate change. Equally crucial was the project's participatory dimension. To initiate the process, I issued an Open Call for Video Statements, inviting individuals to reflect on the concept of transformation in the context of climate justice. The invitation was made public through the Bern Music Festival's platform. I asked participants—scientists, musicians, artists, and theorists—to respond to the following questions: What can we do to bring about the transition to a more sustainable and just world? What measures should we consider as a society? How can I, as an individual, take independent action? What is your perspective on the current climate situation?

To date, I have collected approximately 35 video contributions, which have become both the conceptual and material foundation of the composition. I am deeply grateful to the friends and colleagues across the arts, academia, and research communities who have supported this endeavor. These video statements are integrated into the work as both visual projections and audio sources. Some are presented unchanged, while others are manipulated in real time by participating sound artists using custom-built digital tools.

To facilitate this, I developed two software applications in Max/MSP—one for real-time audio processing and another, built using Max/MSP/Jitter, for video manipulation. These tools allow performers to engage with the material interactively, shaping sound and image through effects, layering, and spatialization. An especially distinctive feature of the piece involves translating the speech from the video statements into Morse code in real time. Generated Morse code is then used as a percussive score. Two performers play historic telegraphs as instruments, whose rhythms—derived from real speech—are amplified and extended through live electronics, generating new sonic textures in real time.

Throughout the compositional process, I created *KlimaAlarm* as a framework for collective interpretation. Performers are encouraged to rehearse and perform the work collaboratively, shaping its audiovisual elements in dialog. The intent is to create a space of encounter—one in which the content of the statements is communicated clearly, but also one that allows for musical experimentation and affective resonance. Artistic decisions, including musical, conceptual, and technical aspects, are made collectively and adapted to the specific spatial and logistical context of each performance. The work remains open to future iterations, with the possibility of integrating new statements, languages, and voices over time.

Beyond its sonic and performative elements, *KlimaAlarm* addresses critical questions of sustainability, agency, and ecological ethics. It prompts reflection on both collective responsibility and individual action. For me, it is essential that this work should contribute to understanding sound as a political instrument. Drawing on post-humanist and ecofeminist perspectives, I aim to reposition sound not merely as a sensory experience, but as an intervention—a tool of disruption, a medium of resistance, and a catalyst for consciousness. The final motif of the alarm is central to this intervention. It is a universally recognizable signal, designed to awaken, disrupt, and mobilize. In my project notes, I describe it as follows:

*"*KlimaAlarm *is a sonic warning—uncomfortable, raw, and designed to evoke an affective response in the listener. Through its structure, it challenges the comfort zones of passive listening and aims to awaken a sense of urgency and agency."*

This interplay between discomfort and heightened awareness resonates with the work of theorists such as Brandon LaBelle, who articulates sound as a form of sonic agency—an active force capable of disrupting dominant narratives and opening up new spaces for political and imaginative intervention. [25]

Within this conceptual framework, *KlimaAlarm* functions as a politico-acoustic dispositif: a sonic structure charged with both epistemic potency and affective intensity. It urges audiences not simply to hear, but to listen critically—to confront the ecological urgencies of our time and to recognize the imperative for action before the threshold of irreversibility is crossed. Beyond its affective impact, the work also transmits knowledge, shaped by the diverse statements of its participants. In doing so, it fosters awareness, provokes critical reflection, and has the potential to reshape listeners' understanding of ecological and political realities.

Moreover, the incorporation of algorithmic composition techniques, in conjunction with real-time transformation and environmental sounds, positions the work within the ambit of a techno-ecological discourse. In this context, technology must be recognized as having dual implications; it serves as a mechanism for ecological observation whilst concurrently contributing to environmental degradation. In *KlimaAlarm*, the paradox is explored by adopting an approach that involves the utilization of the very tools employed as instruments of resistance, communication and reflection to address the crisis, highlighting the voices of the individuals. The question remains: what are we waiting for?

5. REFERENCES

[1] M. Dickreiter, *Handbuch der Tonstudiotechnik*, 6th ed. Wiesbaden: Breitkopf & Härtel, 1997.

[2] J. Blauert, *Spatial Hearing: The Psychophysics of Human Sound Localization*, rev. ed. Cambridge, MA: MIT Press, 1997.

[3] P. Schaeffer, *Traité des objets musicaux: Essai interdisciplinaire*. Paris: Éditions du Seuil, 1966.

[4] L. Thoresen, A. Hedman, *Spectromorphological analysis of sound objects: An adaptation of Pierre Schaeffer's typomorphology. Organised Sound*, vol. 12, no. 2, pp. 129–141. 2007.

[5] S. Roy, *L'analyse des musiques électroacoustiques : Modèles et propositions*. Paris, France: L'Harmattan. 2003.

[6] J. Noble, T. Bonin & S. McAdams, *Experiences of Time and Timelessness in Electroacoustic Music*. *Organised Sound*, vol. 25, no. 2, pp. 232–247, 2020.

[7] D. Smalley, "Spectromorphology: Explaining sound-shapes," *Organised Sound*, vol. 2, no. 2, pp. 107–126, 1997.

[8] D. Smalley, "Space-form and the acousmatic image," *Organised Sound*, vol. 12, no. 1, pp. 35–58, 2007.

[9] E. Nyström, "Topology of spatial texture," *Organised Sound*, vol. 27, no. 3, pp. 314–325, 2022.

[10] T. Carrasco „V*erzerrte Abweichungen" Acousmatic music composition for 8 Channels.* 2025.

[11] R. M. Schafer, *The Soundscape: Our Sonic Environment and the Tuning of the World*. Rochester, VT: Destiny Books, 1994.

[12] Krause, Bernie. *The Great Animal Orchestra: Finding the Origins of Music in the World's Wild Places*. New York: Little, Brown and Company, 2012.

[13] D. Schabert, *Hörlandschaften*. Available online: https://dorotheeschabert.de/?page_id=41 (accessed Dec. 19, 2025).

[14] S. Schäfer, *Works*. Available online: https://sabineschaefer.de/?lang=de (accessed Dec. 19, 2025).

[15] T. Carrasco, *Un giardino sonoro.* Composition and Installation. 2025.

[16] S. Voegelin, *Listening to Noise and Silence: Towards a Philosophy of Sound Art*. New York, NY: Continuum, 2010.

[17] B. Truax, "Imagining acoustic space," in *The Oxford Handbook of Sound and Imagination*, vol. 1, M. Grimshaw-Aagaard, M. Walther-Hansen, and M. Knakkergaard, eds. New York, NY: Oxford University Press, 2019, pp. 659–663.

[18] P. Oliveros, *Deep Listening: A Composer's Sound Practice*. New York, NY: iUniverse, 2005.

[19] P. Oliveros, "Quantum listening: From practice to theory (to practice practice)," *Leonardo Music Journal*, vol. 6, pp. 83–87, 1996.

[20] P. Oliveros, "Improvising with spaces," in *Sounding the Margins: Collected Writings 1992–2009*. Kingston, NY: Deep Listening Publications, 2010, pp. 49–55.

[21] P. Oliveros, *Sonic Meditations*. Baltimore, MD: Smith Publications, 1971.

[22] P. Oliveros, *Collective Environmental Composition*, score, Deep Listening Institute Archives, 1975.

[23] E. Biserna, *Walking from Scores*, ongoing curatorial project. Available online: https://walkingfromscores.org (accessed 2025).

[24] T. Carrasco, *KlimaAlarm! und die dringende Wende zu einer nachhaltigen Entwicklung.* Participative music composition. Available online: https://www.teresacarrasco.com/about-1 (accessed 2026)

[25] B. Labelle, Sonic Agency: Sound and Emergent Forms of Resistance. London: Goldsmiths Press/MIT Press, 2018.

Beyond Execution: Unrealizability and the Ontology of Sound in Computer Music

Paulo C. Chagas
University of California, Riverside
paulo.chagas@ucr.edu

ABSTRACT

This paper proposes an ontological reorientation of computer music grounded in the concept of unrealizability. Drawing on Giorgio Agamben's notion of potentiality without act, it argues that dominant paradigms of electroacoustic and computer music have historically privileged realization, execution, and operability as the primary conditions of sonic being. From early studio practices at the GRM and WDR to the consolidation of computer music as an executable, code-based discipline, sound has largely been understood as something that exists in order to be realized.

Against this background, the paper proposes to examine a series of practices that destabilize the primacy of execution. Practices such as granular synthesis, live electronic and interactive systems, and machine-learning-based processes foreground forms of sonic potentiality that cannot be fully individuated, predicted, or exhausted by realization, thereby suggesting unrealizability not as a limitation but as a constitutive dimension of contemporary computer music.

By framing sound as a field of suspended potential rather than a command to be executed, the paper advances an alternative ontology in which listening becomes a mode of use rather than consumption. This perspective invites a reconsideration of compositional agency, technological apparatuses, and the political implications of sound practices beyond execution, emphasizing openness, contingency, and inoperativity as critical resources for computer music today.

1. FROM REALIZATION TO CAPTURE: THE CYBERNETIC INHERITANCE

The emergence of electroacoustic and computer music in the mid-twentieth century coincided with a broader epistemic shift toward cybernetics, systems theory, and information science. Within this framework, sound increasingly came to be understood not as an irreducible phenomenon, but as a set of parameters susceptible to control, prediction, and execution [1, 2, 3, 4]. This shift profoundly shaped both the aesthetics and the ontology of early electronic and computer music.

From the earliest experiments in tape music and studio-based synthesis, compositional practice was oriented toward the realization of latent structures. Musical ideas were conceived as abstract configurations—of pitch, duration, timbre, density, or spatial position—that awaited technical actualization. Technology was not merely a medium but a guarantor of realization, enabling composers to bring into audible form sonic possibilities that exceeded instrumental performance [5, 6]. In this sense, electroacoustic music was often framed as a project of emancipation: liberating sound from the constraints of notation, instrumental technique, and human physiology [7].

However, this narrative of emancipation also carried an implicit ontological assumption: that the value of sound lies in its realizability. *Potentiality* was implicitly subordinated to act; sonic processes were meaningful insofar as they could be made to happen, rendered audible, fixed on tape, or executed by a machine. This assumption mirrors what Giorgio Agamben identifies as a foundational tendency of Western metaphysics: the privileging of actuality over potentiality, and of production over suspension [8, 9].

The rise of computer music intensified this orientation. In early algorithmic composition and digital sound synthesis, musical ideas were formalized as executable procedures. Sound was no longer only shaped by machines; it was defined by code, grounded in discrete-time signal processing, numerical representation, and algorithmic control structures that formalize sonic behavior across temporal and spectral scales. [10, 6]. Within this paradigm, to exist musically increasingly meant to be computable, schedulable, and executable. The ontology of sound thus increasingly aligned with what could be specified unambiguously and realized deterministically.

A decisive consolidation of this ontology of execution occurs with the emergence of computer-based sound synthesis in the United States, particularly at Bell Laboratories and Stanford University. Max Mathews's development of the MUSIC programming languages in the late 1950s introduced a formal paradigm in which sound is defined explicitly as a sequence of computable operations. In MUSIC I–V, the composer no longer manipulates sound directly, but specifies algorithms that must be executed in order for sound to exist. The score becomes a program, and the realization of sound is inseparable from its computation. In this framework, sound does not precede execution—it appears only insofar as it is brought into act through computation, effectively collapsing potentiality into execution [11, 10, 6].

A similar logic underlies John Chowning's development of frequency modulation (FM) synthesis at Stanford in the 1960s. While FM synthesis expanded the sonic palette dramatically, its significance lies not only in its timbral richness but in its computational formulation. Complex spectra emerge from mathematically defined relationships between

Copyright: ©2026 Paulo C. Chagas. This is an open-access article distributed under the terms of the Creative Commons Attribution License 3.0 Unported, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

oscillators, making sound generation a function of parameterized control and algorithmic precision. Here, sonic complexity is not discovered or materially constructed, but generated through the execution of formal procedures [12, 6].

This paradigm reached a broader technological and cultural consolidation with the commercial implementation of FM synthesis in the Yamaha DX7 synthesizer (1983). The DX7 translated a highly abstract and computationally intensive model of sound generation into an accessible and widely disseminated instrument, embedding the logic of algorithmic execution within musical practice at a global scale. Sound became directly programmable through parameterized control structures, reinforcing the idea that sonic identity emerges from the precise execution of encoded relationships rather than from material interaction. In this sense, the DX7 represents a crucial moment in which the ontology of executable sound moves from research laboratories into mainstream musical production [6, 5].

These developments mark a shift from the studio-based manipulation of sound toward its full integration into a computational paradigm in which execution becomes the condition of possibility for sonic existence. Sound is no longer merely realized through technology; it is constituted as something that exists only insofar as it is executed. In this sense, the work of Mathews and Chowning—and its subsequent dissemination through instruments such as the Yamaha DX7—does not simply extend earlier electroacoustic practices, but renders explicit an ontological assumption already implicit within them: that sound exists insofar as it can be computed, specified, and brought into act through technical systems.

Within this paradigm, *unrealizability* acquired a negative connotation. What could not be rendered audible—due to computational limits, resolution constraints, real-time requirements, or system errors—was treated as noise, failure, or inefficiency. Optimization, precision, and control became dominant values, reinforced by advances in processing power and real-time systems. The compositional act was often framed as the design of increasingly refined mappings between abstract musical intent and technical execution. Here, "execution" refers not only to performance, but to the computable realization of sound as a deterministic or formally specified process.

Beyond its technical and ontological implications, the paradigm of execution also participates in the production of value and legitimacy within computer music. As Pierre Bourdieu's notion of *cultural capital* suggests, technical mastery, control, and precision function as markers of distinction within a field of practice [13]. Systems that demonstrate higher degrees of computational sophistication, real-time responsiveness, or algorithmic complexity often acquire elevated status, reinforcing a hierarchy in which executability becomes a primary condition for the attribution of value.

This regime is further sustained by what Agamben describes as the capture of potentiality by apparatuses [9]. Technical systems do not merely enable sound production, but define the conditions under which sound becomes recognizable, legitimate, and meaningful. To foreground *unrealizability* is therefore not only to propose an alternative ontology of sound, but also to unsettle the criteria by which computer music is valued. Practices that resist full realization—through opacity, indeterminacy, or inoperativity—challenge not only the limits of execution but the very conditions under which value is assigned.

Agamben's critique of modern political ontology offers a productive lens through which to reassess this history. He argues that when potentiality is valued only insofar as it culminates in realization, it becomes susceptible to capture—subsumed by apparatuses that demand productivity, efficiency, and operability [8, 9, 14]. Applied to computer music, this suggests that the cybernetic inheritance does not merely concern technology or style, but a deeper assumption about what sound is: something that exists in order to be executed [3, 4].

This inheritance remains visible in contemporary discourse around computer music, particularly in emphases on real-time performance, responsiveness, optimization, and system transparency. While these developments have undoubtedly expanded expressive possibilities, they also risk reinforcing an ontology in which sound is fully exhaustible by its realization.

Yet, as the history of electroacoustic music itself demonstrates, this ontology of execution has never been fully stable nor uncontested. Even within highly technical and formally controlled contexts, composers have persistently introduced elements of indeterminacy, opacity, excess, and drift—gestures that resist full realization and challenge the primacy of execution [15, 16]. These practices do not merely extend the paradigm of execution; they expose its limits, revealing a dimension of sound that cannot be entirely captured by computation, control, or technical realization.

The following sections build on this tension by articulating unrealizability not as a technical limitation, but as a critical and productive category for computer music. Drawing on Agamben's notion of potentiality without act, we argue that specific electroacoustic practices—particularly those involving granular sound, live systems, and machine learning—foreground modes of sonic potentiality that exceed execution. In doing so, they point toward an ontology of sound irreducible to realization, opening new political and aesthetic possibilities for listening and composition [17].

2. POTENTIALITY WITHOUT ACT: AGAMBEN AND THE ONTOLOGY OF SOUND

To develop this ontological reorientation, this paper turns to Giorgio Agamben's concept of *potentiality without act*, developed most explicitly in *The Unrealizable: Toward a Politics of Ontology* [17]. Agamben's intervention is not primarily aesthetic or historical, but ontological: it concerns what it means for *potentiality* to persist without being exhausted in realization. His argument challenges the deeply rooted assumption—central to Western metaphysics, political thought, and technological rationality—that potentiality exists only in order to be fulfilled in actuality [8].

Agamben revisits Aristotle's notion of *dynamis* (potentiality), foregrounding a dimension often overlooked: the capacity not to pass into act. Potentiality is thus not merely oriented toward realization, but includes the possibility of its own suspension. It persists precisely where it is not exhausted by actuality, maintaining a relation to what it

can be without being compelled to become it [8, 17]. This insight has far-reaching implications for practices—such as computer music—that are structurally oriented toward execution, performance, and output.

Translated into the domain of sound, Agamben's argument invites a fundamental reorientation. In much electroacoustic and computer music discourse, sound is implicitly understood as something that must be made audible: synthesized, rendered, spatialized, executed in real time, or fixed as an artifact. Sonic potential—whether conceptual, algorithmic, or perceptual—is thus valued only insofar as it culminates in realization. What remains unrealized is typically framed as latency, error, inefficiency, or unrealized potential [4].

Agamben's notion of the *unrealizable* allows us to rethink this framework at a fundamental level. Unrealizability is not impossibility; it is a positive ontological condition in which something exists without needing to be actualized. Applied to sound, this implies that sonic potential does not require realization in order to be musically or conceptually operative. Sound may thus persist as tension, threshold, latency, or statistical field—without being fully individuated or exhausted by perception, and without being reducible to its audible manifestation [17].

This perspective resonates with the electroacoustic paradigm understood as an apparatus-driven field of subjectivation, in which sound is inseparable from the technical, cultural, and cognitive structures that shape it [4]. However, whereas earlier paradigmatic analyses emphasized the expansion of sonic consciousness through apparatuses, Agamben's contribution lies in identifying a critical limit: apparatuses capture potentiality by forcing it into operability [9, 14]. When sound is fully subordinated to execution—whether through code, signal flow, or real-time responsiveness—it risks losing the openness that constitutes its expressive and political force.

Agamben names the counter-movement to this capture *inoperativity*. Inoperativity does not mean inactivity or negation, but the suspension of a function from its predetermined end. An apparatus rendered inoperative is not destroyed; it is freed from necessity [9]. Transposed to sound practice, inoperativity designates situations in which sonic systems are allowed to hover between possibility and realization, where sound is used without being fully exhausted by execution.

This notion of *use* is central. Agamben distinguishes use from production or possession: to use something is to enter into a relation that neither exhausts nor appropriates it [18]. In sonic terms, this implies a relationship to sound that is not governed by completion, optimization, or finality. Listening, in this sense, becomes a primary site of ontological significance—not as passive reception, but as a practice that allows sound to remain partially unrealized, indeterminate, or excessive in relation to its technical instantiation [4].

From this perspective, the history of electroacoustic and computer music can be reread as oscillating between two ontological poles: on the one hand, an ontology of execution, in which sound exists to be realized; on the other, an emergent ontology of unrealizability, in which sound retains a remainder that resists full actualization. The latter does not reject technology or computation, but calls into question the assumption that execution constitutes the ultimate horizon of sonic being [17].

The following sections examine how this unrealizable dimension becomes audible—or, more precisely, how it operates at the threshold of audibility—in specific electroacoustic practices. Granular synthesis, live electronic systems, and machine-learning-based processes each articulate forms of sonic potentiality that exceed deterministic realization and resist full capture by execution. These practices do not mark a retreat from computer music, but a reconfiguration of its ontological ground: from sound as something to be executed to sound as a field of potentiality that remains open, unexhausted, and inoperative.

3. DIVERGENT ORIGINS, SHARED HORIZON: GRM, WDR, AND THE ONTOLOGY OF REALIZATION

The early history of electroacoustic music is marked by a well-known aesthetic and methodological divergence between the Paris-based *musique concrète* of the Groupe de Recherches Musicales (GRM) [1] and the *elektronische Musik* developed at the WDR studio in Cologne. This divergence should not be minimized; on the contrary, it reflects two fundamentally different approaches to sound, listening, and composition, each grounded in a distinct epistemology [19, 20, 21].

At the GRM, sound was approached phenomenologically as a sound object derived from recorded reality. Composition proceeded from acts of listening rather than from abstract formal construction. Through practices such as reduced listening, sound was detached from its causal source and treated as an autonomous perceptual event. The technical apparatus—most notably the tape recorder—served to isolate, repeat, and transform sonic fragments, allowing sound to emerge as a perceptual unit accessible through attentive listening [19, 7]. In this context, sound was discovered rather than invented, and composition unfolded as an exploration of auditory experience.

By contrast, the WDR studio in Cologne pursued an explicitly constructivist orientation. Sound was not derived from the acoustic environment but synthesized from elementary electronic processes. Signal generators, filters, and serial procedures were employed to construct sound from abstract parameters, extending compositional thinking beyond pitch and rhythm into the domain of timbre and microstructure. Here, electronic sound was conceived as a continuation of instrumental composition by other means, grounded in formal coherence and structural control rather than in perceptual immediacy [20, 21].

These two approaches articulate genuinely different musical poetics: one centered on listening and perceptual emergence, the other on synthesis and compositional construction. As such, they represent distinct aesthetic responses to the crisis of tonal organization and to the expanding role of technical apparatuses in musical practice. Any ontological account of electroacoustic music must begin by acknowledging this difference.

However, viewed through Agamben's ontology of potentiality, a deeper convergence becomes visible—one that

[1] The Groupe de Recherches Musicales (GRM) was formally established in 1958 as a continuation and institutional consolidation of Pierre Schaeffer's earlier Groupe de Recherche de Musique Concrète, integrating musique concrète into a broader research framework within French radio (ORTF).]

operates beneath aesthetic divergence. Despite their contrasting methods, both GRM and WDR practices ultimately presuppose a common horizon: sound attains significance only through its realization. Whether discovered through listening or constructed through synthesis, sound is understood as something that must be rendered audible, fixed, and stabilized through technical means. The sound object and the electronic signal alike culminate in a realized artifact—on tape, in broadcast, or in performance [8, 4].

From this perspective, the divergence between phenomenological and constructivist approaches masks a shared ontological assumption: potentiality is valued only insofar as it passes into act. Sonic possibilities—whether latent in recorded material or embedded in abstract procedures—are meaningful to the extent that they are realized as audible events. What remains unrealized is treated as preparatory, residual, or extraneous to the musical work.

This shared horizon becomes even more explicit with the emergence of computer music. In early computer-based composition, sound is no longer merely shaped or captured, but defined as executable. Algorithms, scores, and signal flows function as commands whose fulfillment lies in execution [10, 6]. In this context, realization is no longer simply a goal but an ontological requirement: sound exists only insofar as it can be computed and rendered. The convergence of GRM and WDR approaches thus finds its technical consolidation in the ontology of executability that characterizes computer music.

This diagnosis aligns with a broader understanding of electroacoustic music as a paradigmatic field shaped by technical apparatuses, rather than as a mere collection of stylistic tendencies. As argued elsewhere, paradigms operate as ontological operators that determine what can appear as sound, how it is produced, and how it is perceived [4]. From this perspective, the historical opposition between sound object and sound synthesis unfolds within a single electroacoustic paradigm whose apparatuses privilege realization, fixation, and operability.

Agamben's notion of unrealizability allows us to reassess this paradigm without negating its historical necessity or aesthetic achievements. His critique does not target specific techniques or institutions, but addresses the underlying assumption that realization exhausts potentiality. Applied to electroacoustic music, this suggests that the critical question is no longer whether sound is discovered or constructed, but whether its being is fully exhausted by realization [17].

The importance of this question becomes evident in practices that strain the limits of realizability itself. Techniques such as granular synthesis, statistical sound masses, live electronic systems, and machine-learning-based processes introduce sonic potentials that cannot be fully individuated, predicted, or stabilized. In these contexts, sound no longer appears as a discrete object or fully executable process, but as a field of potentialities, many of which remain unrealized or only partially manifest.

The following section examines granular sound as a paradigmatic case in which unrealizability becomes musically operative. Granular techniques foreground a sonic remainder that resists full realization, opening a space in which sound exists not as a command to be executed, but as suspended potential—suggesting an alternative ontology of sound beyond execution.

4. GRANULAR SOUND AND THE UNREALIZABLE REMAINDER

Granular sound synthesis occupies a distinctive position in the history of computer music because it fundamentally destabilizes the relationship between sound, realization, and perception. While often framed as a technique for constructing complex timbres from microscopic units, it simultaneously articulates an ontology in which sound emerges from processes that resist full realization and exceed perceptual individuation [22, 16].

At the technical level, granular synthesis operates through the aggregation of large numbers of short sonic fragments—grains—typically lasting from a few milliseconds to a few tens of milliseconds. Crucially, the individual grain does not function as a perceptual unit: its duration falls below the threshold at which sound is ordinarily perceived as an event. What becomes audible is not the grain itself, but the statistical behavior of grain populations—densities, envelopes, spectral distributions, and spatial dispersions [6, 22]. This corresponds to a shift from event-based representation to the control of statistical distributions in time-frequency space.

This asymmetry between the microscopic and the perceptual marks a decisive departure from an ontology of realization. Although grains are technically generated and executed by the computer, their individual realization is musically irrelevant. The grain persists as a potential element whose singular actualization is inconsequential. What becomes operative is the emergent sonic field arising from the aggregation of innumerable unrealized—or more precisely, unindividuated—events.

From an Agambenian perspective, this structure foregrounds a crucial distinction between execution and realization. While grains are executed computationally, they are not realized as musical events; their being remains suspended between calculation and perception. They constitute a sonic remainder that conditions audible form without itself attaining perceptual presence. Granular sound thus exemplifies a mode of potentiality that does not culminate in act, but persists as operative precisely in its unrealizability [17].

This stands in contrast to earlier electroacoustic paradigms, in which sound objects or synthesized signals were conceived as discrete, fixed, and perceptually identifiable entities. In granular synthesis, the compositional focus shifts from the control of such entities to the shaping of fields, textures, and sound masses. The composer no longer specifies events, but conditions: ranges of duration, probabilistic distributions, density curves, and spatial behaviors. Sound is no longer commanded to occur in a determinate way, but emerges from a system of constrained indeterminacy [16, 23].

The affinity between granular sound and statistical or stochastic thinking is well documented [16, 22]. For example, Xenakis's stochastic synthesis already articulates sound as the macroscopic effect of probabilistic micro-events, while later granular frameworks formalize this through density-based control of grain populations rather than discrete sonic objects. What remains less frequently articulated, however, is its ontological implication: sound exists as a population of potentials rather than a sequence of acts. The individual grain aligns closely with Agamben's notion

of unrealizability—not as impossibility, but as a mode of being that exceeds and resists exhaustion by actualization. The grain is essential, yet it remains unclaimed by perception, unpossessed by the listener, and unexhausted by execution.

Listening to granular sound reinforces this ontological shift. The listener cannot attend to individual grains, nor reconstruct the process from perception alone. What emerges is a continuous flux—a dynamic field whose internal operations remain partially opaque. Listening becomes an engagement with emergence rather than events, with tendencies rather than actions. In this sense, granular sound invites a mode of listening aligned with Agamben's concept of *use*: sound is not consumed as a sequence of objects, but inhabited as a temporal and spatial condition [18, 4].

Granular techniques also destabilize the notion of compositional control. While the system may be precisely specified, its audible outcome remains fundamentally unpredictable. Small variations in parameters, random seeds, or input material can produce disproportionate perceptual effects. The composer relinquishes detailed command over realization and instead configures a space of sonic possibility. This shift does not diminish agency, but transforms it—from command to configuration, from execution to suspension.

In this sense, granular synthesis can be understood as a paradigmatic instance of *inoperative sound*. The apparatus remains active, yet its operations are no longer oriented toward the production of discrete, exhaustible events. Instead, it sustains a field in which sound continuously arises and dissolves without closure. The system is rendered inoperative with respect to the traditional demand that sound be fully realized, individuated, and stabilized, suspending its function without negating its operation [9].

This perspective resonates with a broader understanding of electroacoustic music as a paradigm shaped by apparatuses that both expand and constrain sonic experience [4]. Granular synthesis does not reject the apparatus; rather, it exposes its limits by allowing sound to exceed its own operational logic. What emerges is a sonic practice in which unrealizability is no longer a technical deficiency, but a constitutive resource of musical thought and experience.

Granular sound thus marks a decisive step beyond the ontology of execution. It demonstrates that computer music can operate productively in a domain where sound is neither fully commanded nor fully realized—where potentiality persists as an audible, yet never fully graspable, remainder. This remainder anticipates forms of practice in which unrealizability becomes even more pronounced, particularly in live electronic systems and machine-learning-based processes, where real-time interaction and algorithmic opacity further suspend the primacy of execution.

The next section extends this argument by turning to live electronic and interactive systems, where unrealizability emerges not only at the level of micro-temporality, but through temporal contingency, system feedback, and relational indeterminacy.

5. LIVE ELECTRONICS, CONTINGENCY, AND INOPERATIVE SYSTEMS

Live electronic and interactive systems introduce a further displacement of the ontology of execution in computer music. Unlike fixed media or precomputed sound processes, they unfold in real time through the interaction of performers, technical apparatuses, and acoustic environments. In this context, sound is no longer fully specified in advance, nor exhaustively realizable through execution alone. Instead, it emerges from contingent relations, feedback processes, and temporal indeterminacy [5, 24].

At a basic level, live electronic systems retain the computational infrastructure of execution: algorithms are run, signals are processed, and parameters are updated in real time. However, the presence of performers, acoustic space, and temporal immediacy fundamentally transforms the status of realization. Sound is no longer the direct outcome of a command, but emerges from a situated interaction whose course cannot be fully predicted or controlled. Execution thus becomes a necessary but insufficient condition for realization.

This shift challenges the ontology inherited from early computer music, in which sound was defined primarily as executable code. In live systems, execution is continuously exposed to interruption, delay, error, and adaptation. Latency, feedback, and instability—often treated as technical problems to be minimized—become constitutive elements of the musical experience. Sound no longer appears as a fixed event, but as a process of becoming shaped by the unfolding present [24].

From an Agambenian perspective, live electronic systems introduce a space of *inoperativity* within operation itself. The system remains active, yet its functioning is no longer subordinated to the realization of predetermined outcomes. Its operations are suspended from their instrumental end and opened to contingency. The apparatus does not cease to function; rather, it functions otherwise [9].

This inoperative dimension becomes especially evident in interactive systems that respond to human input in nondeterministic or loosely coupled ways. Sensors, controllers, and machine listening modules translate gestures, sounds, or movements into data streams, yet these translations are never neutral or complete. They introduce thresholds, ambiguities, and delays that preclude any direct mapping between action and sound. As a result, the system resists transparency and undermines the logic of command [25].

In such contexts, sound is neither fully authored nor fully autonomous, but arises in the interstice between human intention and machinic response. This relational ontology undermines the notion of realization as fulfillment. What matters musically is not that a sound occurs as planned, but that it remains open to deviation—susceptible to influence without being reducible to it. The system sustains a field of sonic potentiality that is continually negotiated rather than executed.

Live electronic practices thus foreground the temporal dimension of unrealizability. Unlike granular synthesis, where unrealizability operates at the level of microstructure, here it emerges through temporal exposure: the impossibility of fully anticipating what will happen next. Each moment carries unactualized possibilities that may never be

realized, yet nonetheless shape the musical present. Unrealized sound becomes operative as an active force, influencing perception and expectation without ever attaining perceptual presence.

Listening in this context shifts from the recognition of sonic objects toward an engagement with unfolding processes. The listener encounters sound as a fragile and transient event, inseparable from the conditions of its emergence. This mode of listening aligns with Agamben's concept of *use*, insofar as sound is neither possessed nor exhausted by perception. It is experienced as a shared temporal condition rather than a consumable artifact [18, 4].

Importantly, live electronic systems do not negate compositional agency; they transform it. The composer no longer specifies sounds to be realized, but designs conditions of interaction: response curves, thresholds, feedback paths, and temporal constraints. Composition becomes the configuration of a space in which sound may appear, rather than the determination of what must appear. This shift echoes the movement from constituent to destituent power identified by Agamben: rather than founding new structures of control, the composer renders existing structures inoperative by suspending their necessity to act in a prescribed way [9].

This transformation also affects the status of performance. The performer is no longer an executor of predetermined material, but a participant in a dynamic system whose behavior cannot be fully mastered. Performance becomes an act of navigation rather than execution—a continuous negotiation with a system that retains its own agency. The resulting sound is irreducible to either human intention or machine operation.

Live electronics thus occupy a critical position in the trajectory beyond execution. They demonstrate that unrealizability can arise not only from microscopic sound structures, but from the relational and temporal complexity of interactive systems. Sound persists as a potential never fully exhausted by realization, even as it unfolds in real time.

The final section extends this argument to machine-learning-based systems, where unrealizability assumes a new dimension. In AI-driven computer music, the opacity of learning processes and the redistribution of agency further destabilize the ontology of execution, opening a space in which sound emerges from non-conscious cognitive assemblages rather than from explicit command.

6. MACHINE LEARNING, OPACITY, AND NON-CONSCIOUS SOUND

Machine-learning-based systems introduce a decisive transformation in the ontology of computer music, as exemplified in contemporary interactive systems (e.g., at IRCAM), one in which unrealizability is no longer primarily a matter of microstructure (as in granular synthesis) or temporal contingency (as in live electronics), but of epistemic opacity and the redistribution of agency. In contrast to rule-based or explicitly programmed systems, machine-learning models operate through statistical inference and pattern extraction processes that remain only partially accessible to human understanding, even when their technical mechanisms are known [26, 27].

From the perspective of execution, machine learning appears to extend the logic of computation to unprecedented levels of automation. Models are trained, optimized, and deployed to generate or transform sound with remarkable efficiency. Yet this intensification of executability conceals a fundamental ontological shift: the relation between intention, process, and outcome is no longer transparent or linear. The sound produced by a trained model cannot be fully traced to explicit compositional decisions, nor exhaustively predicted in advance [28, 25].

This opacity has important consequences for the status of realization. In machine-learning-based composition, sound no longer emerges as the fulfillment of a command, but as the outcome of probabilistic processes distributed across large datasets and parameter spaces. The composer does not specify sounds to be realized, but curates training materials, architectures, and constraints whose interaction yields outcomes that remain partially indeterminate. Realization is partially displaced from explicit execution to statistically mediated emergence. [27].

Agamben's concept of *unrealizability* provides a critical framework for understanding this displacement. The unrealizable is not what cannot happen, but what cannot be fully brought into act without remainder. In machine-learning systems, this remainder assumes the form of non-conscious cognitive operations that exceed both compositional intention and perceptual grasp. Sound emerges from processes that neither the composer nor the listener can fully appropriate or exhaust [17, 26].

This condition marks a departure from earlier paradigms of computer music, in which sound was defined as executable code and compositional agency resided in the explicit specification of processes. In machine-learning-based systems, agency is distributed across human and non-human actors—composers, datasets, algorithms, computational infrastructures, and acoustic environments. Sound becomes the surface manifestation of a complex assemblage whose internal operations remain only partially accessible [4].

Such assemblages resonate with a broader understanding of electroacoustic music as shaped by apparatuses that mediate between living beings and technical systems [9, 4]. Machine learning, however, intensifies this mediation to the point where the apparatus no longer merely extends or captures human intention, but actively reconfigures the space of potentiality itself. The system does not simply realize pre-existing possibilities; it generates new statistical relations that were not explicitly conceived in advance [28].

Listening to sound produced by machine-learning systems reflects this ontological shift. The listener encounters sonic forms that resist clear causal attribution. Familiar gestures may appear without recognizable sources, and continuity may emerge without discernible structure. Listening becomes an engagement with effects whose causes remain opaque, reinforcing a mode of perception attuned to emergence rather than control. Sound is no longer consumed as an object, but experienced as a manifestation of processes that remain partially withdrawn [26].

From an Agambenian perspective, this withdrawal is not a defect, but a condition of freedom. By suspending the direct link between potentiality and act, machine-learning systems render the apparatus inoperative with respect to total control. The system continues to function, yet no longer guarantees the realization of intention. This *inoperativity* opens a space in which sound persists as a potential that is neither fully

commanded nor fully realized [9, 17].

Importantly, this does not imply a relinquishment of responsibility or critical agency; on the contrary, it demands a reconfiguration of compositional practice. The role of the composer shifts from authoring sounds to orchestrating conditions of emergence, from specifying outcomes to negotiating with systems whose behavior cannot be fully predicted. Composition becomes a practice of ethical and aesthetic positioning within a field of non-conscious operations [26, 28].

Machine-learning-based computer music thus crystallizes the trajectory traced throughout this paper. From early electroacoustic practices oriented toward realization, through granular sound and live interaction, to AI-driven systems characterized by opacity and emergence, we observe a gradual displacement of execution as the dominant ontological horizon. What comes into view is an alternative understanding of sound—one in which unrealizability is no longer a technical limitation, but a constitutive dimension of musical being.

This shift invites a reconsideration of computer music not as a discipline defined by what can be executed, but as a practice concerned with how sound can remain open, unexhausted, and shared. The conclusion that follows draws together these strands and outlines the implications of an ontology of unrealizability for contemporary computer music research and practice.

7. CONCLUSION: BEYOND EXECUTION

This paper argues for an ontological reorientation of computer music grounded not in new techniques or stylistic prescriptions, but in an ontological shift: from sound understood as something to be executed toward sound understood as a field of unrealizable potentiality. Drawing on Giorgio Agamben's notion of potentiality without act, it has argued that the history and practice of electroacoustic and computer music cannot be fully understood through the paradigm of realization alone [17].

By revisiting the divergent origins of electroacoustic music at the GRM and WDR studios, the paper has shown that despite profound aesthetic and methodological differences, both traditions converge on a shared ontological horizon in which sound attains significance through realization and fixation. This horizon becomes technically and conceptually consolidated in early computer music, particularly with the emergence of algorithmic sound synthesis and programming environments such as the MUSIC languages, in which sound is defined as executable code and realization becomes an ontological requirement rather than a compositional choice [3, 4].

Against this background, the paper has traced a series of practices that destabilize the primacy of execution—granular synthesis, live electronic systems, and machine-learning-based processes—each foregrounding forms of sonic potentiality that exceed realization. Taken together, these practices show that unrealizability is not a limitation to be overcome, but a constitutive condition of contemporary computer music. Sound persists as a potential that is never fully exhausted by its realization, remaining open to emergence, variation, and reinterpretation.

In this context, listening acquires a renewed ontological and political significance. It is no longer the consumption of fully realized sonic objects, but a mode of use that allows sound to remain unpossessed and unexhausted. To listen is to inhabit a sonic field without demanding its completion or closure. In Agamben's terms, such listening is political not because it conveys explicit messages or positions, but because it suspends the imperative that sound—and by extension, technical systems—must culminate in operability and control [18].

From this perspective, the history of electroacoustic and computer music can be reread as a persistent tension between two ontological orientations rather than as a linear progression of techniques as summarized in Table 1. This table does not describe opposing camps, but coexisting tendencies shaping contemporary practice. This paper does not reject execution, computation, or technological rigor, but argues that computer music becomes critically and creatively vital when execution is no longer the final horizon of sound.

This perspective opens several directions for future research: (1) a renewed historiography attentive to unrealized potentialities in electroacoustic practice; (2) a theoretical articulation of the relation between unrealizability and cultural value, in dialogue with Bourdieu; and (3) an exploration of how emerging technologies—such as machine learning and immersive systems—can sustain forms of sonic potential that remain unexhausted.

Beyond execution, computer music opens a space in which sound remains open, contingent, and shared—an inoperative yet generative field where potentiality is preserved rather than exhausted. Listening is thereby repositioned as a critical practice that sustains indeterminacy rather than resolving sound to realization.

Table 1. Ontological tensions in computer music between execution and unrealizability.

Ontology of Execution	**Ontology of Unrealizability**	**Musical / Technical Examples**	**Listening / Experiential Examples**
Sound as command	Sound as potential	Deterministic algorithmic score; rule-based synthesis	Expectation of correct realization
Executability	Suspension	Offline rendering; fixed DSP processes	Anticipation, latency
Optimization	Drift	Parameter smoothing; feedback instability	Emergent textures
Control	Contingency	Live electronics with feedback loops	Fragile sonic presence
Event	Field	Note-based synthesis vs. granular sound masses	Immersive continuity
Realization	Use	Predefined output vs. exploratory interaction	Listening as inhabitation
Transparency	Opacity	Rule-based algorithms vs. machine-learning models	Causal indeterminacy

References

[1] Thomas S. Kuhn. *The Structure of Scientific Revolutions*. 2nd. Chicago: University of Chicago Press, 1970.

[2] Norbert Wiener. *The Human Use of Human Beings: Cybernetics and Society*. 2nd. Garden City, NY: Doubleday, 1954.

[3] Paulo C. Chagas. *Unsayable Music: Six Reflections on Musical Semiotics, Electroacoustic and Digital Music*. Leuven: Leuven University Press, 2014.

[4] Paulo C. Chagas. "Sound, Truth, and Paradigm". In: *Sounds from Within: Phenomenology and Practice*. Ed. by Paulo C. Chagas and J. C. Wu. Cham: Springer Nature, 2021, pp. 1–28.

[5] Joel Chadabe. *Electric Sound: The Past and Promise of Electronic Music*. Upper Saddle River, NJ: Prentice Hall, 1997.

[6] Curtis Roads. *The Computer Music Tutorial*. Cambridge, MA: MIT Press, 1999.

[7] Pierre Schaeffer. *In Search of a Concrete Music*. Trans. by Christine North and John Dack. Berkeley: University of California Press, 2012.

[8] Giorgio Agamben. *Potentialities: Collected Essays in Philosophy*. Trans. by Daniel Heller-Roazen. Stanford, CA: Stanford University Press, 1999.

[9] Giorgio Agamben. *What Is an Apparatus? And Other Essays*. Trans. by David Kishik and Stefan Pedatella. Stanford, CA: Stanford University Press, 2009.

[10] Charles Dodge. *Computer Music: Synthesis, Composition, and Performance*. 2nd. New York: Schirmer Books, 1997.

[11] Max V. Mathews. "The Digital Computer as a Musical Instrument". In: *Science* 142.3592 (1963), pp. 553–557. DOI: 10.1126/science.142.3592.553.

[12] John M. Chowning. "The Synthesis of Complex Audio Spectra by Means of Frequency Modulation". In: *Journal of the Audio Engineering Society* 21.7 (1973), pp. 526–534.

[13] Pierre Bourdieu. *Distinction: A Social Critique of the Judgement of Taste*. Trans. by Richard Nice. Cambridge, MA: Harvard University Press, 1984.

[14] Michel Foucault. *Power/Knowledge: Selected Interviews and Other Writings, 1972–1977*. Ed. by Colin Gordon. New York: Pantheon, 1980.

[15] John Cage. *Silence: Lectures and Writings*. Middletown, CT: Wesleyan University Press, 1961.

[16] Iannis Xenakis. *Formalized Music: Thought and Mathematics in Composition*. Stuyvesant, NY: Pendragon Press, 1992.

[17] Giorgio Agamben. *The Unrealizable: Toward a Politics of Ontology*. Trans. by Alberto Toscano. London: Seagull Books, 2025.

[18] Giorgio Agamben. *The Use of Bodies*. Trans. by Adam Kotsko. Stanford, CA: Stanford University Press, 2016.

[19] Pierre Schaeffer. *Traité des objets musicaux*. Paris: Seuil, 1966.

[20] Herbert Eimert. "What Is Electronic Music?" In: *Die Reihe* 1 (1958), pp. 1–10.

[21] Karlheinz Stockhausen. "Electronic and Instrumental Music". In: *Die Reihe* 5 (1963), pp. 50–59.

[22] Curtis Roads. *Microsound*. Cambridge, MA: MIT Press, 2001.

[23] Henri Pousseur. *Fragments théoriques I: Sur la musique expérimentale*. Bruxelles: Éditions de l'Institut de Sociologie de l'Université Libre de Bruxelles, 1970.

[24] Denis Smalley. "Spectromorphology: Explaining Sound Shapes". In: *Organised Sound* 2.2 (1997), pp. 107–126.

[25] Gérard Assayag and Shlomo Dubnov. "Using Factor Oracles for Machine Improvisation". In: *Soft Computing* 8.9 (2004), pp. 604–610.

[26] N. Katherine Hayles. *Unthought: The Power of the Cognitive Nonconscious*. Chicago: University of Chicago Press, 2017.

[27] Gérard Assayag, Laurent Bonnasse-Gahot, and Joakim Borg. "Cocreative Interaction: Somax2 and the REACH Project". In: *Computer Music Journal* 46.4 (2022), pp. 7–25. DOI: 10.1162/comj_a_00662.

[28] Margaret A. Boden. *The Creative Mind: Myths and Mechanisms*. 2nd. London: Routledge, 2004.

Classical Music Mashup System and Compatibility Heuristics

Yu Foon Darin Chau
Hong Kong University of Science and Technology
yfdchau@connect.ust.hk

Andrew Horner
Hong Kong University of Science and Technology
horner@cse.ust.hk

ABSTRACT

We investigate symbolic classical music mashups and introduce a retrieval-based pipeline for generating them. Unlike audio-domain mashups, symbolic mashups offer perfect voice isolation and allow for post-generation reinterpretation of tempo, dynamics, and instrumentation. While prior work in audio mashups emphasises harmony, rhythm, and balance, symbolic mashups in classical repertoires remain underexplored and lack clear compatibility heuristics. To this end, we conduct controlled listening tests on classical music excerpts to isolate factors shaping perceived compatibility. Results indicate effective mashups should respect the recognizability of motivic materials, underlying cadential logic, and be presented polyphonically. We designed a symbolic mashup pipeline for classical piano music around these findings that maximises pairwise piece compatibility. We discuss implications and limitations for algorithmic composition, pedagogical tools, and future extensions to broader styles, longer forms, and richer evaluative methodologies.

1. INTRODUCTION

In recent years, music mashups have emerged as a promising direction for algorithmic music composition and machine creativity. Contemporary systems typically analyse and index large corpora, estimate musical compatibility between segments, and separate, align, and recombine musical parts into new artefacts. While a significant body of work has been done in the audio domain, focusing on automatic pop/rock mashup pipelines [1, 2], classical music mashups and automatic mashup systems in the symbolic domain have remained relatively underexplored.

Audio mashups rely on imperfect source separation and are prone to artifacts when time-stretching or pitch-shifting. In contrast, operating in the symbolic domain offers unique affordances that set it apart from audio mashups. Symbolic data separates the composition from the performance, allowing for flawless post-mashup interpretation of tempo, velocity, articulation, and instrumentation. The inherent isolation of voices in symbolic formats allows for accurate motif extraction and application of strict, rule-based compositional constraints such as species counterpoint. Symbolic mashups also yield readable sheet music, transforming the output from a static audio file into an interactive pedagogical tool that can be analyzed and performed by human musicians. Works like Woolard's Classical Music Mashup videos [3] demonstrate that effective symbolic mashups have great value both in entertainment and education. As such, developing frameworks, heuristics, and mashup algorithms for symbolic music analysis could be valuable for advancing algorithmic composition techniques, designing targeted feedback software, and writing controlled etudes for compositional ear training.

Classical mashup research faces at least two obstacles. First, it is not clear what factors contribute to the compatibility of two pieces of music. While the literature has demonstrated that matching harmony and rhythm largely suffices as a compatibility heuristic in the audio domain in short time spans [1, 4], they have noted that it is more nuanced, with factors such as timbre and song recognizability playing a not-insignificant role. Second, auditory music seems to attract more research effort and thus enjoy better algorithms and modelling techniques than its symbolic counterpart. This also seems to be partly due to data availability in their respective modalities. Variations in symbolic semantics, such as MIDI, MusicXML, Humdrum/Kern, and the lack of standardised annotations for pedagogical tools like phrase boundaries, cadences, tonalities, or voice-leading rules, further complicate the process. As such, MIR tools for symbolic music can be limited in scope and can fail to capture nuances.

In this research project, we investigate the notion of compatibility pertaining to symbolic music. We design listening tests that isolate a few factors that might contribute to the compatibility, and propose a mashup system that creates basic symbolic mashups. Our contributions are as follows:

1. Our tests found that in Western classical music, cadence logic and melodic recognizability play a larger role in modelling compatibility.

2. Our tests confirm that combining pieces as contrapuntal interplays, compared to collages or simple melody swaps, is generally perceived as more effective mashups.

3. We propose a retrieval-based mashup system that can effectively search and generate classical music mashups in the symbolic domain.

2. RELATED WORKS

Music mashups have been extensively studied in the audio domain. One extensively cited approach is AutoMashupper's mashability score [1], which defines compatibility as harmonic similarity and spectral balance after rhythmic

Copyright: ©2026 Yu Foon Darin Chau et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License 3.0 Unported, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

alignment. This formulation relied on robust MIR components: downbeat tracking, chord detection, and source separation models. Subsequent works refined this by using approaches such as deep neural networks to model compatibility and by explaining design choices through listening tests [4, 2].

Similar MIR tools have been developed previously for symbolic music. For tonality, classic cognitive and probabilistic models (e.g., Krumhansl–Schmuckler [5]; Temperley's Bayesian key/meter [6]) provide principled features and baselines, for melodic identity and recognizability, sequence-alignment and edit-distance approaches remained standard. Graph-based approaches have been attempted to detect cadential structures [7]. Robust MIR tools[8, 9, 10] also offer symbolic music manipulation and analysis capabilities. In terms of data availability, recent years have also seen a surge in large-scale symbolic music datasets for various tasks, such as AriaMIDI [11] and MetaMIDI [12] for score analysis and generative pretraining, MAESTRO/ASAP [13, 14] for audio-score transcription, and SymphonyNet [15] for symphonic music generation.

3. CLASSICAL MUSIC MASHUP PIPELINE

This section outlines a basic retrieval-based symbolic music mashup pipeline centred on a large symbolic music corpus and a modified notion of musical compatibility. The pipeline creates symbolic music mashups by extracting motivic elements, matching tempo and cadences, then find pairs of melodies that, when presented simultaneously, admit minimal voice-leading violations.

We start with the subset of midi-classical-music on Hugging Face [16], which contains 3.2k pieces of baroque, classical, and early-romantic-era music. This choice is because the files of these pieces are more cleanly notated, and the pieces tend to have a clearer sense of tonality, making analysis easier. Then, each piece is standardised to XML via MuseScore to dynamically quantise note values. Instrument information is discarded in this process to eliminate the effects of timbre and instrumentation on the final mashups. Tempo markings are also smoothened, with variations within a 10% threshold assumed to be rubato or other inconsistencies and discarded. This cleaning step gives uniformly notated MIDI files along with tempo and the score-level rhythmic relationships between notes.

The compatibility between two melodies is measured by how well the underlying cadential logic matches, followed by counting the number of voice-leading violations in the combined melodies. The individual voices are extracted using a graph-based approach due to Foscarin et al. [17], and the motivic elements are detected using the COSIATEC [18] algorithm and subsequently TF-IDF, operating on the assumption that motifs are the thematic elements that appear frequently in a piece but rarely in other pieces in the corpus. For the cadential logic, we use another graph-based method [7] to determine the absolute harmony at every point in time, and the key is inferred by tracking V-I progressions, where the perfect cadence usually anchors the key of a passage.

When the user provides an input piece, it is cleaned similarly, and the cadence is extracted as above. Filtering for melodies with a similar tempo and cadential progression, a list of candidate melodies is extracted. For each pair of

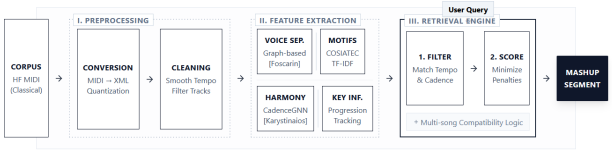


Figure 1. Pipeline Architecture. The system processes the Hugging Face MIDI corpus through three stages: (I) Standardization, (II) Parallel extraction of melodic and harmonic features using graph neural networks and heuristic algorithms, and (III) A retrieval engine that scores candidates against a user query based on voice leading rules.

melodies, we calculate a penalty score of how improperly the two melodies treat voice leading according to species counterpoint rules. This finally yields a target segment from a candidate that we can perform a mashup on. One can also choose to find candidate mashup pairs among the corpus, or create three-or-more song mashups by noting that both the voice-leading rule and cadence logic compatibility heuristic of the sum can be treated as the sum of that of pairwise melodies. Figure 1 summarizes our pipeline.

4. EXPERIMENTS AND RESULTS

This section reports two listening studies that identifies the musical factors that affect perceived quality the most in symbolic classical mashups, and whether our proposed retrieval pipeline can generate convincing musical mashups respectively. Mashup compatibility perception is a time-local phenomenon, and thus, we evaluate on short excerpts of 8 bars.

In test 1, we identified and isolated four factors that contribute to mashup compatibility: harmony alignment, cadential logic, motif recognizability, and texture. For each of them, we picked two pairs of candidate mashup songs, and for each pair, we composed two sets of mashups with one of the following:

- Harmony: altering the harmony to match both songs' underlying chords up to inversions,
- Cadential logic: matches the tonality and tonal function of the chord progression,
- Motif recognizability: assumes lightly altered materials are more desirable, and
- Texture: one set will be a contrapuntal presentation (where both mashup melodies are present simultaneously) and the other set a homophonic presentation (where we combine the melody of song A and the accompaniment/countermelody of song B)

Each excerpt is presented in the listening test as MIDI playback on the piano with constant default velocity, dynamics, articulation, and technique, and ornamentations are kept as written on the score. The listening test is given to a group of 30 students with varying musical training. Figure 2 presents the average preference score. Importantly, on qualitative assessment, we noticed that when the harmony does not match, but the underlying cadential logic does (e.g. secondary dominants vs. augmented sixths), it is generally still perceived as compatible. However, when the harmony matches, but the underlying cadential logic

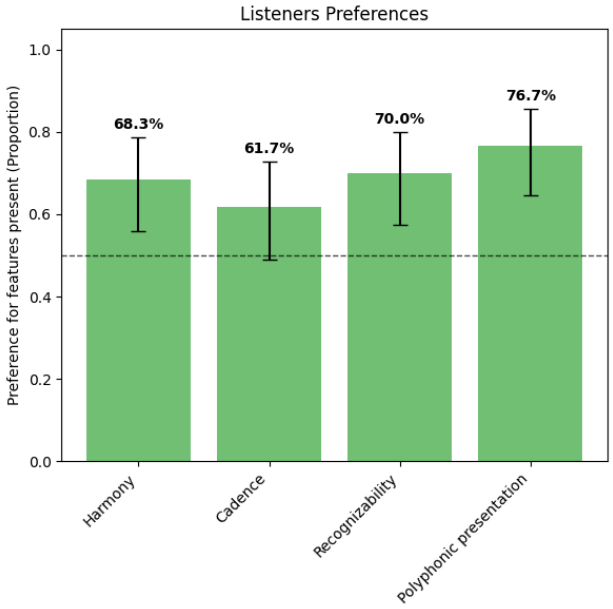


Figure 2. Results of the listening study evaluating listener preference for specific musical features. The bar chart displays the proportion of listeners who preferred excerpts where the specific feature (Harmony, Cadence, Recognizability, or Polyphonic presentation) was present versus absent. Error bars indicate 95% Wilson's CI, and the dashed line represents equal preference. Percentages above the bars denote the mean preference score for each factor.

does not (this might happen when the phrase boundary does not align), it is perceived as incompatible. The results align with common perception as in Woolard's mashups, and justify our design choices to prioritise cadential logic and tonality, use lightly-modified thematic materials, and present our mashups polyphonically.

In test 2, we used a handpicked subset of midi-classical-music containing 100 popular pieces and found all candidate cross-pairings using our system. The best candidates are then picked out and played back on the same controlled MIDI setting as above, and presented to our subjects. This test is done qualitatively, with 5 senior undergraduate music students providing expert opinions on each of the mashups. The experts generally have noted a few points:

- The results are effective mashups in the sense that the source materials are recognisable on first listen.
- The thematic materials can be difficult to follow due to our controlled MIDI setting; this could be alleviated by techniques or voicing in real-time performances, and picking pieces with contrasting rhythmic identity and articulation could help make the materials more recognisable.
- A constant contrapuntal presentation could be demanding on the listeners. When generalising to full-length compositions, it might be advisable to control the intensity of the piece by alternating between different presentation textures and the number of simultaneous voices.

5. LIMITATIONS

First, the tests used only short segments with uniform piano MIDI with constant dynamics and articulation. While this setting allows us to control for certain factors like timbre and voicing, it undermines expressive cues and broader musical context in shaping musical recognizability. For instance, Woolard[3] sometimes makes minor adjustments to the melody to make mashups more effective, but exactly when to break the rules is difficult to quantify. The modelling of larger-scale form, mood and emotions, or long-range tension-release - factors central to classical listening - are also not considered in this study.

Second, our corpus is restricted with narrow stylistic diversity and limited exposure to chromatic, modal, or post-tonal idioms whose cadential logics are less well-captured by traditional assumptions. This provides a good starting point for our study, and we speculate that this method can generalise to a mainstream pop/rock corpus, but nevertheless poses a bias: for instance, our method might not generalise or would require a modified heuristic to late/post-romantic music, other genres like jazz, or music of other ethnic origins.

Finally, our evaluation sample is quite modest, with a small student cohort and a qualitative expert study. This limits statistical power and might leave us with an incomplete view of listeners' judgements and open questions about the effects on the judgement of factors like audience expertise and training effects.

6. CONCLUSION

We introduced a retrieval-based pipeline for symbolic classical mashups and examined various factors that contribute to a successful mashup in this regard. Our listening tests suggest that cadential logic and melodic recognizability are primary determinants of perceived fit, and a contrapuntal presentation is more convincing than melody swaps, contrary to pop-style mashup systems. These findings align with pedagogical practices in tonal and counterpoint theory and are supported to some extent by both historical and contemporary music compositions. We designed our system around these findings to discover potential pairings through cadential alignment and a rule-based penalty for voice-leading violations. Qualitative expert feedback indicates that the resulting mashups are recognizably sourced and musically coherent under controlled conditions, while also highlighting limitations like texture and articulation, and generalisability both to other genres and to long-form mashup compositions.

Future work can extend both analysis and evaluation along several dimensions. The corpus can be scaled beyond common-practice styles, which could accommodate music from more diverse genres and enrich our heuristics and models. The system can be extended to long-form or multi-instrument compositions, incorporating factors such as phrasing, emotions, and performance techniques, as well as macro-level texture, form, and tension management. Multi-song compatibility heuristics can also be explored, investigating whether the current two-song factors generalize. More comprehensive user studies with larger, stratified listeners and mixed methods, like MOS ratings or think-aloud protocols, will further validate the proposed

heuristics. These steps will lead us towards a better understanding of what constitutes an effective mashup and lead us to designing better mashup systems in the symbolic domain.

7. REFERENCES

[1] M. E. P. Davies, P. Hamel, K. Yoshii, and M. Goto, "AutoMashUpper: An Automatic Multi-Song Mashup System," in *Proc. Int. Society for Music Information Retrieval Conf. (ISMIR)*, Curitiba, Brazil, 2013, pp. 575–580. [Online]. Available: https://doi.org/10.5281/zenodo.1416050

[2] M. Delabaere, L. Miqueu, M. Moreno, G. Bigois, H. Duong, E. Fernandez, F. Manent, M. Salgado-Herrera, B. Pasdeloup, N. Farrugia, and A. Marmoret, "AutoMashup: Automatic Music Mashups Creation," *arXiv preprint arXiv:2508.06516*, 2025. [Online]. Available: https://arxiv.org/abs/2508.06516

[3] R. Pentreath, "A genius has created a classical music mashup of 70 pieces by famous composers," *Classic FM*, Jun. 2021, updated: 2021-06-11. Accessed: 2026-01-04. [Online]. Available: https://www.classicfm.com/music-news/videos/classical-music-mashup-70-pieces-composers/

[4] X. Wu and A. B. Horner, "An automated mashup system with drum-bass stem swapping," *Journal of the Acoustical Society of America*, 2023.

[5] C. L. Krumhansl, "A key-finding algorithm based on tonal hierarchies," in *Cognitive Foundations of Musical Pitch*. New York: Oxford University Press, 1990, ch. 4, algorithm written in collaboration with Mark Schmuckler; first edition 1990.

[6] D. Temperley, *Music and Probability*. MIT Press, 2010.

[7] E. Karystinaios and G. Widmer, "Cadence Detection in Symbolic Classical Music using Graph Neural Networks," *arXiv preprint arXiv:2208.14819*, 2022. [Online]. Available: https://arxiv.org/abs/2208.14819

[8] M. S. Cuthbert and C. Ariza, "music21: A Toolkit for Computer-Aided Musicology and Symbolic Music Data," in *Proc. Int. Society for Music Information Retrieval Conf. (ISMIR)*, Utrecht, Netherlands, 2010, pp. 637–642. [Online]. Available: http://ismir2010.ismir.net/proceedings/ismir2010-108.pdf

[9] C. McKay, J. Cumming, and I. Fujinaga, "jSymbolic 2.2: Extracting Features from Symbolic Music for Use in Musicological and MIR Research," in *Proc. Int. Society for Music Information Retrieval Conf. (ISMIR)*, 2018, pp. 348–354.

[10] C. E. Cancino-Chacón, S. D. Peter, E. Karystinaios, F. Foscarin, M. Grachten, and G. Widmer, "Partitura: A Python Package for Symbolic Music Processing," in *Proc. Music Encoding Conference (MEC)*, Halifax, Canada, 2022.

[11] L. Bradshaw and S. Colton, "Aria-MIDI: A Dataset of Piano MIDI Files for Symbolic Music Modeling," *arXiv preprint arXiv:2504.15071*, 2025. [Online]. Available: https://arxiv.org/abs/2504.15071

[12] J. Ens and P. Pasquier, "Building the MetaMIDI Dataset: Linking Symbolic and Audio Musical Data," in *Proc. 22nd Int. Society for Music Information Retrieval Conf. (ISMIR)*, 2021. [Online]. Available: https://archives.ismir.net/ismir2021/paper/000022.pdf

[13] C. Hawthorne, A. Stasyuk, A. Roberts, I. Simon, C.-Z. A. Huang, S. Dieleman, E. Elsen, J. Engel, and D. Eck, "Enabling Factorized Piano Music Modeling and Generation with the MAESTRO Dataset," in *Int. Conf. on Learning Representations (ICLR)*, 2019. [Online]. Available: https://openreview.net/forum?id=r1lYRjC9F7

[14] S. D. Peter, C. E. Cancino-Chacón, F. Foscarin, A. P. McLeod, F. Henkel, E. Karystinaios, and G. Widmer, "Automatic Note-Level Score-to-Performance Alignments in the ASAP Dataset," *Transactions of the International Society for Music Information Retrieval*, 2023.

[15] J. Liu, Y. Dong, Z. Cheng, X. Zhang, X. Li, F. Yu, and M. Sun, "Symphony Generation with Permutation Invariant Language Model," in *Proc. Int. Society for Music Information Retrieval Conf. (ISMIR)*, 2022. [Online]. Available: https://arxiv.org/abs/2205.05448

[16] drengskapur, "midi-classical-music," Hugging Face, 2023, accessed: 2026-01-04. [Online]. Available: https://huggingface.co/datasets/drengskapur/midi-classical-music

[17] F. Foscarin, E. Karystinaios, E. Nakamura, and G. Widmer, "Cluster and Separate: a GNN Approach to Voice and Staff Prediction for Score Engraving," 2024. [Online]. Available: https://arxiv.org/abs/2407.21030

[18] D. Meredith, "COSIATEC and SIATECCompress: Pattern Discovery by Geometric Compression," in *Music Information Retrieval Evaluation eXchange (MIREX)*, Curitiba, Brazil, 2013.

Exploring Masked CE Losses to Enhance Word Offset Estimation in CTC-based Lyrics-to-Audio Alignment

Tian Cheng, Tomayasu Nakano, and Masataka Goto
National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, Japan
tian.cheng@aist.go.jp t.nakano@aist.go.jp m.goto@aist.go.jp

ABSTRACT

Lyrics-to-audio alignment is an important task for real-world applications such as karaoke systems. Despite alignment performance improved with the release of large datasets and the utility of advanced deep learning models, accurate word offset estimation remains challenging. To address this problem, we extend our previously proposed masked cross-entropy (CE) loss by proposing new masks to enforce model predictions at masked frames with frame-wise phoneme labels derived from word-level annotations. We train a Convolutional Recurrent Neural Network (CRNN) by using both the masked CE loss and the Connectionist Temporal Classification (CTC) loss. By comparing the results obtained by using different masks in the masked CE loss, we find that word offset estimation performance is improved by using masks which cover all silent frames. In addition, we find that masks on word onset frames are essential for improving word onset estimation performance. We achieve comparable word onset estimation results and provide benchmark word offset estimation results for future research.

1. INTRODUCTION

In this paper, we address the lyrics-to-audio alignment task, which aims to synchronize lyrics with musical audio. There are two inputs for a lyrics-to-audio alignment system: the musical audio (a mix of singing and musical accompaniment) and the corresponding lyrics. The output is time-synchronized lyrics, with the start time (onset) and end time (offset) of each lyric unit [1]. Depending on the application scenario, the time-synchronized lyrics can be categorized into different granularity levels: such as line, word, and phoneme levels, as illustrated in Figure 1. This task is crucial for real-world applications that require accurate, large-scale time-synchronized lyrics, such as karaoke systems, lyrics-based music retrieval [2], and lyrics animation for music videos [3].

Lyrics-to-audio alignment is a challenging task due to the unique characteristics of singing voice and the interference introduced by musical accompaniment. Singing is generally more expressive than speech [2, 4, 5], with lyrics sung with varying dynamics, pitches, and durations. This variability in pronunciation makes it difficult to build an effective acoustic model. Furthermore, the musical accompaniment is highly correlated with the singing voice, which reduces the effective signal-to-noise ratio for singing voice processing. To address these challenges, some methods [6, 7, 8, 9, 10] separated singing vocals from the mixed audio by using advanced source separation methods [11] before alignment. Alternatively, other methods exploited temporal correlations between lyrics and musical elements, such as melody, chords and accompaniment. They improved lyrics-to-audio alignment performance by learning it jointly with other related tasks, such as chord estimation [12], singing pitch detection [9], singing voice separation [13], and singing voice synthesis [14].

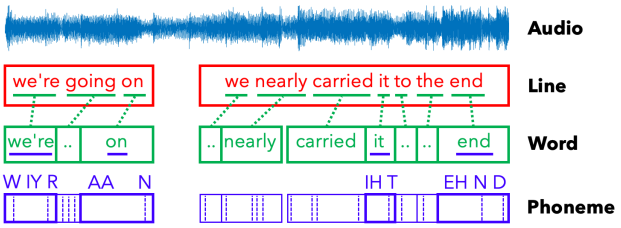
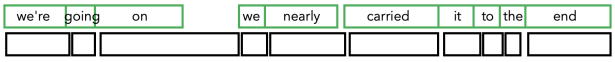


Figure 1: An illustration of the lyrics-to-audio alignment task with the input audio in waveform and the output time-synchronized lyrics in three levels: line, word, and phoneme levels.

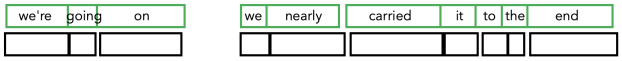
With the release of the DALI dataset (a large Dataset of synchronised Audio, LyrIcs and notes) [15, 16], lyrics-to-audio alignment performance has been improved substantially [17, 9, 18, 19]. However, most available datasets provide only weak (line-level or word-level) annotations, lacking frame-wise (phoneme-level) annotations necessary for supervised training. To train models using such weak annotations, researchers have developed different approaches. Gupta et al. [20] employed a conventional automatic speech recognition method, using predicted frame-wise phoneme labels as ground truth for training. Stoller et al. [21] proposed an end-to-end model based on the Wave-U-Net architecture and trained it using the Connectionist Temporal Classification (CTC) loss [22]. Other methods leveraged cross-modal models by mapping audio and text embeddings into a shared latent space via contrastive learning [23, 19]. Kang et al. [18] acheived lyrics-to-audio alignment by reformulating the alignment task as a word onset estimation task and trained the model with the cross-entropy (CE) loss.

Among these methods, the CTC loss has attracted increasing attention in recent work [7, 24, 9, 10], because it simplifies model training by enabling learning from weak,

Copyright: ©2026 Tian Cheng et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License 3.0 Unported, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.



(a) Estimated results by using the method in [25] (with mask_p)



(b) Estimated results by using the proposed method (with onsets obtained by using mask_onset and offsets obtained by using mask_onset_silence)

Figure 2: Estimated word onsets and offsets by using the method in [25] and the proposed method, with the ground truth indicated by the above green boxes. See Section 3.2 for details of the masks.

line-level annotations without explicit alignment. However, the CTC loss is suitable for transcription tasks but not sensitive to the precise timing of individual elements in a sequence, which may limit alignment accuracy. To address this limitation, Teytaut et al. [24] added a spectral reconstruction loss to the CTC loss to improve the temporal coherence in the model's predictions. Huang et al. [9] proposed a joint learning approach that integrates pitch estimation to enhance alignment accuracy.

In our previous work [25], we also applied a CTC-based model and proposed a masked CE loss to mitigate the timing imprecision inherent in the CTC loss. In the masked CE loss, frame-wise phoneme labels were derived from word-level annotations at key frames (word onset and offset frames, as well as silent frames), with details shown in Section 3.1. Then, we enforced the model phoneme predictions at the key frames by using the masked CE loss with a mask on these frames. The experimental results showed that incorporating the masked CE loss improved word onset estimation. However, we observed incorrect offsets in the results: a word often ends just before the next word with the silent frames undetected, as shown in the Figure 2(a).

Therefore, in this paper, we work on improving word offset estimation performance in the lyrics-to-audio alignment system. We extend our previously proposed masked CE loss [25] by proposing new masks to explore the effects of different masks on alignment. We achieve better results on both word onset and offset estimation by using the new masks. By carefully examining the results of using different masks in the masked CE loss, we find that using masks covering all silent frames enhances word offset estimation; and masks on word onsets are essential for improving on word onset estimation. We combine the best onset and offset estimations as the output of the proposed method, as shown in Figure 2(b). The contributions of this paper are summarized as follows:

- We propose four new masks for the masked CE loss, improving word onset and offset estimation performance.
- We compare the effects of using different masks to find the essential frames in the masks for performance improvements on word onset and offset estimation, respectively.
- We conduct evaluation on word offsets. Since there are no existing results on word offset estimation, we present our results as a benchmark for future research.

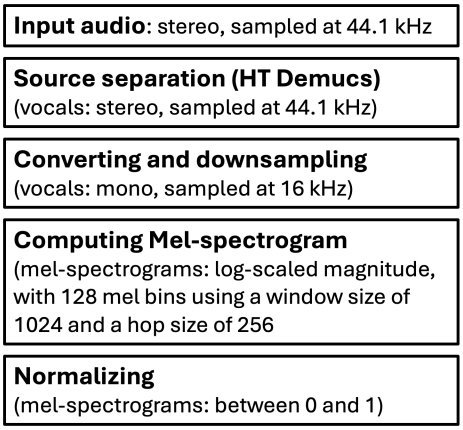


Figure 3: Audio pre-processing

The rest of this paper is organized as follows. Section 2 describes the baseline method and Section 3 illustrates the proposed extension in the masked CE loss. The experimental setup and results are presented in Section 4. We conclude the paper in Section 5.

2. THE BASELINE METHOD

In a lyrics-to-audio alignment system, lyrics are aligned with audio to produce time-synchronized lyrics. To do that, we first transform the musical audio into a spectrogram and convert the lyrics into a ground-truth phoneme sequence. Then, we build and train a model to predict phoneme probabilities for each frame based on the spectrogram. Finally, we align the model predictions with the ground-truth phoneme sequence to produce time-synchronized lyrics.

This section describes the baseline lyrics-to-audio alignment method, which is based on our previously proposed method in [25]. Readers are referred to [25] for further details.

2.1 Audio and Lyrics Pre-processing

For audio pre-processing, we first apply an advanced source separation method, HT Demucs [11], to extract the singing vocals. Then, we convert the extracted audio to a mono track and downsample it. We compute the log-scaled magnitude mel-spectrogram and normalize it to a range between 0 and 1. See Figure 3 for details.

For lyrics pre-processing, we convert the word sequence in the lyrics into a phoneme sequence. Word phonetization follows the conventions of the CMU Pronouncing Dictionary [1], which uses a set of 39 phonemes. In addition to these phoneme tokens, we add a token `0` to represent the blank symbol ϵ (staying in the same state) used in the CTC loss. We also add a token `40` for silence or space, resulting in 41 distinct tokens. Figure 4 shows an example of lyrics along with the corresponding phonemes and token representations.

[1] `http://www.speech.cs.cmu.edu/cgi-bin/cmudict`

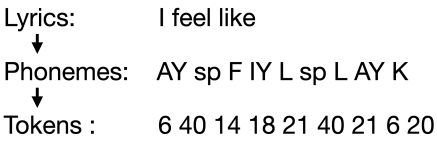


Figure 4: An example of converting lyrics to phonemes for lyrics pre-processing

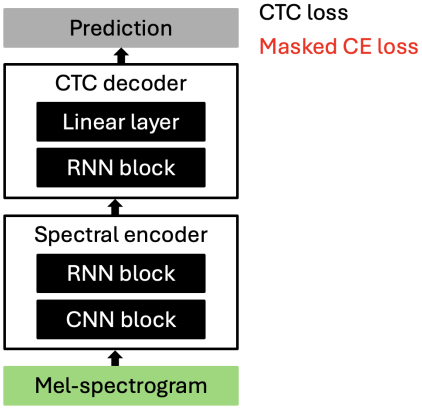


Figure 5: Model architecture. The baseline model is trained with the CTC loss. We propose and compare different masked CE losses to improve alignment performance.

2.2 Model Architecture

We adopt a simple Convolutional Recurrent Neural Network (CRNN) based on the models used in [24, 25]. The network architecture is shown in Figure 5: the mel-spectrogram is fed into a spectral encoder and a CTC decoder to produce frame-wise phoneme probabilities as model predictions.

The spectral encoder consists of a CNN block and an RNN block. The CNN block contains two 2D convolutional layers with 16 and 32 filters, respectively. Both layers use 3×3 kernels and `ReLU` activation. A sequence of operations are applied after each convolutional layer, including batch normalization, a pooling operation that halves the feature dimension, and a 25% dropout. The RNN block contains two bi-directional LSTM layers with 512 units in each layer.

In the CTC decoder, there are an RNN block (identical to the one used in the spectral encoder) and a linear layer with an output dimension of 41 (corresponding to the number of tokens). We apply a `softmax` function on the model output to obtain frame-wise phoneme probabilities.

This network is a simplified version of the model used in [25], where a spectral decoder was stacked after the CTC decoder to reconstruct the spectrogram for computing a reconstruction loss. In the paper, we discard the spectral decoder to focus on the masked CE loss.

2.3 The CTC Loss

We predict the frame-wise phoneme probabilities $\mathbf{D} = \mathbb{P}(\hat{\mathbf{y}}|\mathbf{X})$ from the input mel-spectrogram $\mathbf{X} \in \mathbb{R}^{N_F \times T}$, where N_F is the number of frequency bins, T is the number of time frames, and $\hat{\mathbf{y}} = \{\hat{y}_t\}, t \in [0, \cdots, T-1]$ represents the predicted phoneme sequence. In the phoneme probabilities $\mathbf{D} \in \mathbb{R}^{N_T \times T}$ (where N_T is the number of tokens), its element $D_{i,t} = \mathbb{P}(\hat{y}_t = i|\mathbf{X})$ denotes the probability of predicting token i at time t. Since the frame-wise annotations are unavailable, it is not possible to train the model by minimizing the loss between the predicted and ground-truth phonemes at each frame. Instead, we minimize the loss between the predicted and ground truth phoneme sequences by using Connectionist Temporal Classification (CTC) loss [22].

Set the phoneme sequence of the ground-truth lyrics $\mathbf{y} = \{y(m)\}$, $m \in [0, \ldots, M-1]$, where M is the length of the ground-truth phoneme sequence, $y(m) \in \mathcal{A}$, and $\mathcal{A}$ denotes the set of all possible phonemes. The predicted phoneme sequence $\hat{\mathbf{y}}$ is an extended sequence of the ground-truth phoneme sequence $\mathbf{y}$. There is no time synchrony between these two phoneme sequences; but there is order synchrony between them (the ground-truth and predicted phoneme sequences happen in the same order). Usually, each phoneme in $\mathbf{y}$ spans one or several frames in $\hat{\mathbf{y}}$ with $M \leq T$. To maximize the probability of the predicted phoneme sequence $\hat{\mathbf{y}}$, we train the learnable model parameters Θ under the CTC constraints of $\hat{\mathbf{y}} \in \mathcal{A} \cap \{\epsilon\}$, and $\mathcal{B}(\hat{\mathbf{y}}) = \mathbf{y}$,

$$\mathbb{P}(\mathbf{y}|\mathbf{X};\Theta) = \sum_{\hat{\mathbf{y}},\mathcal{B}(\hat{\mathbf{y}})=\mathbf{y}} \prod_{t=0}^{T-1} \mathbb{P}(\hat{y}_t|\mathbf{X};\Theta), \tag{1}$$

where $\mathcal{B}$ is a mapping function that collapses repeated labels and removes blank symbols ϵ. For example, $\mathcal{B}(\epsilon aaa\epsilon\epsilon bb\epsilon\epsilon) = ab$. The CTC loss is defined as the negative log-likelihood of the above probability:

$$\mathcal{L}_{\text{CTC}}(\Theta) = -\log \mathbb{P}(\mathbf{y}|\mathbf{X};\Theta). \tag{2}$$

2.4 Post-processing

Before forced alignment, we remove the silent segments at the beginning and end of each song to improve performance and robustness. First, we sum the magnitude mel-spectrogram along the frequency axis and normalize it to a maximum value of 1. Then, we apply a threshold of 0.05 to identify frames above the threshold as non-silent frames. We take the first non-silent frame as the start point and the last non-silent frame as the end point. We extend both points by one second (63 frames) to ensure that all singing parts are included. We only use the audio segment between the start and end points for alignment.

We align the model predictions with the ground-truth phoneme sequence using the CTC segmentation, a forced alignment method adapted for CTC-based model predictions [26]. The CTC segmentation is based on dynamic programming, in which the probability of staying in the same state is represented by the probability of the blank symbol ϵ as the way used in the CTC loss. We use the PyTorch implementation [2] of the CTC segmentation for forced alignment.

For phoneme-level alignment, we detect phoneme boundaries from the predicted phoneme sequence obtained by forced alignment. The segmented phonemes are merged into words (or lines) for word-level (or line-level) alignment. In this paper, we produce word-level alignment results for evaluation.

[2] `https://pytorch.org/audio/stable/tutorials/forced_alignment_tutorial.html`

Word	Phonemes	Phoneme labels	Onset	Offset
I	AY	6	0.087 seconds	0.184 seconds
feel	F IY L	14 18 21	0.281 seconds	0.377 seconds
like	L AY K	21 6 20	0.474 seconds	0.571 seconds

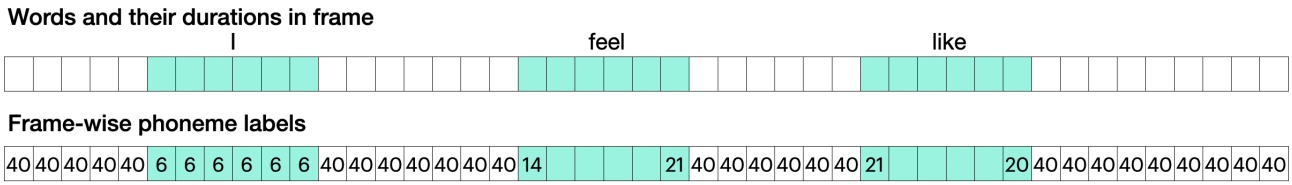


Figure 6: Word-level annotations and the derived phoneme-level labels at word onsets, word offsets, and silent frames.

3. EXPLORING MASKED CE LOSSES

Our work mainly focuses on using the masked CE loss along with the CTC loss to improve lyrics-to-audio alignment performance. Computing the masked CE loss includes two steps: (1) deriving frame-wise phoneme labels for key frames from word-level annotations and (2) constructing a mask on the key frames [25]. In this paper, the derived frame-wise phoneme labels in the first step remain the same as in our previous work [25]. Our extension lies in the second step: we propose four new masks and examine their impact on lyrics-to-audio alignment to identify effective masks for improving alignment performance.

3.1 The Derived Phoneme Labels for Key Frames

From weak annotations at the word level, we can derive frame-wise phoneme labels for specific frames, such as word onset and offset frames. For example, a musical clip with the lyrics "I feel like" would have word-level annotations that specify the onset and offset times of each word, as shown in Figure 6. From such annotations, we can derive phoneme labels for three types of frames:

1. For words containing two or more phonemes (e.g., "feel" and "like"), the first phoneme is assigned to the onset frame, and the last phoneme is assigned to the offset frame.

2. For words consisting of a single phoneme (e.g., "I"), all frames within the word duration are labeled with that phoneme.

3. For frames outside any word segments, a silence/space label (token `40`) is assigned.

3.2 Proposed Masks

These labeled frames (frames with derived labeled) play an important role in determining word onsets and offsets, which are essential for word-level alignment. To enforce the model prediction at these frames, we need to construct a mask to compute the masked CE loss on the masked frames.

In our previous work, we created a binary mask (mask_p) with ones at the labeled frames [25]. However, due to the introduction of the blank symbol ϵ in the CTC loss, non-onset frames can be represented by either phonemes or ϵ, according to the CTC mapping function ($\mathcal{B}(\epsilon aaa\epsilon\epsilon bb\epsilon\epsilon) = ab$). In contrast, onset frames are always labeled by phonemes. To relax the constraint on non-onset frames, we set mask_p to zeros for ϵ (token 0) at these non-onset frames.

- mask_p: ones at all labeled frames, but zeros at non-onset frames for ϵ (token 0), as shown in Figure 7(b).

In this paper, we propose four new masks to explore the essential frames in the masks for performance improvements. However, we include ϵ for all masked frames in the new masks for two reasons: (1) the blank symbol ϵ is important in the CTC loss and post-processing; and (2) our previous results showed that the above mask (mask_p) does not help with word offset estimation. The proposed masks are listed as follows:

- mask_all: ones at all labeled frames, as shown in Figure 7(c). The masked CE loss is computed over all labeled frames to ensure that the model predicts the ground-truth phonemes at those frames.

- mask_onset: ones at word onset frames, as shown in Figure 7(d). The masked CE loss is computed over word onset frames to ensure that the model predicts the ground truth phonemes at onset frames.

- mask_onset_silence: ones at word onset frames and all silent frames, as shown in Figure 7(e). The masked CE loss is computed over word onset frames and all silent frames to improve silent frame detection for better word offset estimation.

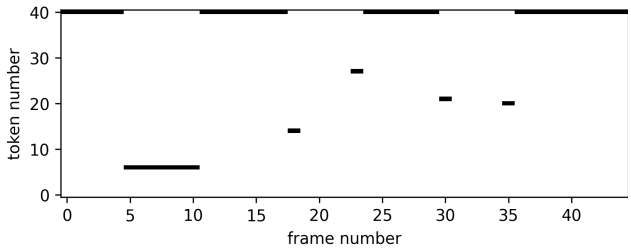
- mask_onset_slstart: ones at word onset frames and frames where silences begin, as shown in Figure 7(f). The masked CE loss is computed over word start frames and silence start frames, treating successive silent frames as a single segment.

3.3 The Masked CE loss

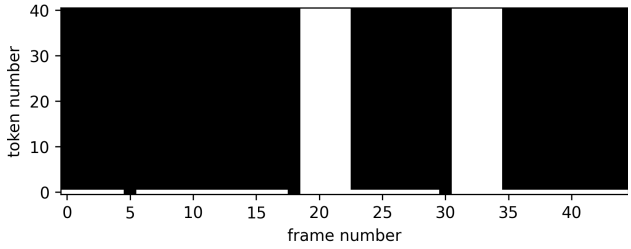
The masked CE loss is computed over the masked frames as follows:

$$\mathcal{L}_{\text{maskedCE}} = -\frac{1}{N_{\text{mask}}} \sum_{i} \sum_{t} B_{i,t} H(L_{i,t}, D_{i,t}), \qquad (3)$$

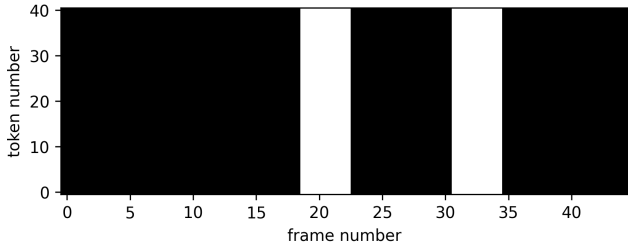
where $\mathbf{B} \in \mathbb{R}^{41 \times T}$ represents the mask and $\mathbf{L} \in \mathbb{R}^{41 \times T}$ represents the one-hot encoding of the derived phoneme



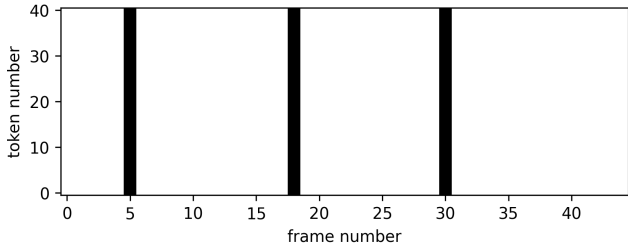
(a) The derived phoneme labels (one-hot encoding)



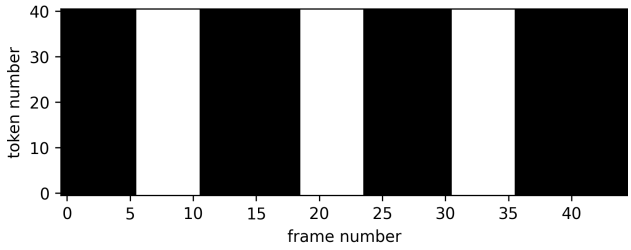
(b) Mask_p



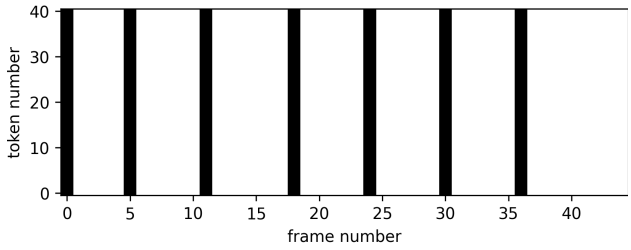
(c) Mask_all



(d) Mask_onset



(e) Mask_onset_silence



(f) Mask_onset_slstart

Figure 7: Derived frame-wise phoneme labels (one-hot encoding) and different masks, with black for ones and white for zeros. See Section 3.2 for mask details.

labels (see Figure 7 for examples). $\mathbf{D} \in \mathbb{R}^{41\times T}$ denotes the model predictions. $H(a,b) = a\log(b)$, and N_{mask} is the total number of masked frames.

We can enforce the model predictions at the masked frames by applying the masked CE loss to assist training. In the experiments, the CRNN model is trained to minimize the following combined loss $\mathcal{L}$:

$$\mathcal{L} = \mathcal{L}_{\text{CTC}} + \lambda\mathcal{L}_{\text{maskedCE}}, \tag{4}$$

where λ is the weight of the masked CE loss. We set $\lambda = 1$ for all the masks, as this value performed best in the preliminary experiment.

4. EXPERIMENTS AND RESULTS

4.1 Datasets

For training and validation, we use the DALI dataset [15], which contains over 5,000 songs along with annotations at four hierarchical levels: note, word, line, and paragraph. In our experiments, we use an English subset of the DALI dataset with 3,352 audio-available songs. This subset is randomly split into training and validation sets with an 80/20 ratio, resulting in 2,681 training songs and 671 validation songs.

To evaluate our method, we use the publicly available Jamendo dataset [23], which provides both line-level and word-level annotations. We present word-level results on 20 English songs in the Jamendo dataset and compare our results with other state-of-the-art (SOTA) results obtained on the same dataset.

During training and validation, the vocal audio is divided into 10-second segments with a 5-second hop size. For each segment, only the words that fall entirely within the segment are used. In contrast, during testing, the full vocal audio of each song is used without segmentation.

4.2 Training

We train the model for up to 20 epochs by using a batch size of 32 and the RMSprop optimizer with a learning rate of 10^{-4}. To reduce variability caused by random initialization and dropout, each model is trained ten times with different random seeds. We report the results of all ten trained models, as well as their average.

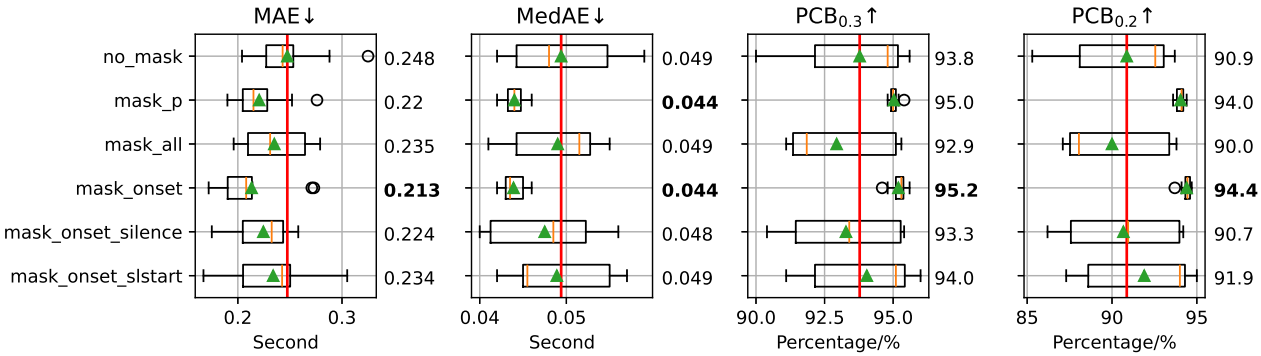
4.3 Evaluation Metrics

We conduct word-level evaluation by comparing the estimated word boundaries with the ground-truth word boundaries. The results are reported with different metrics, including: Mean Absolute Error (MAE) [27], Median Absolute Error (MedAE), and Percentage of Correct Boundaries (PCB) [3] with two tolerance windows [12].

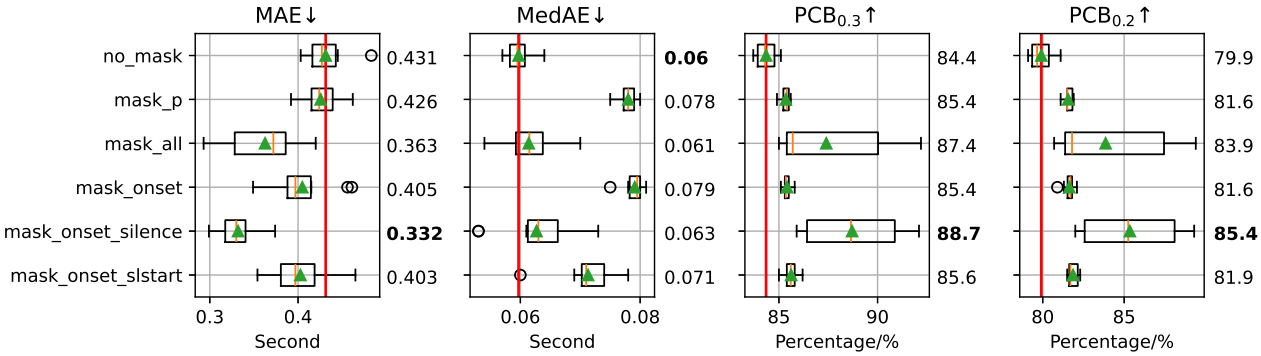
$$\text{MAE} = \frac{1}{W}\sum_{w=1}^{W}|t^w_{pred} - t^w_{ref}|, \tag{5}$$

$$\text{MedAE} = \underset{1\le w\le W}{\text{median}}(|t^w_{pred} - t^w_{ref}|), \tag{6}$$

[3] We use Percentage of Correct Boundaries (PCB) here in contrast to commonly-used Percentage of Correct Onsets (PCO) because we evaluate on two word boundaries (onsets and offsets).



(a) Word onset estimation results.



(b) Word offset estimation results.

Figure 8: Word-level lyrics-to-audio alignment results on the test Jamendo dataset. For each mask setting, the results of ten trained models are shown via a box-plot, with an orange line at the median and circles at fliers. The average value is indicated by the green triangle and also shown on the right. The baseline results (no_mask) are marked by red vertical lines for comparison. Bold numbers indicate the best results.

$$\text{PCB}_\tau = \frac{1}{W}\sum_{w=1}^{W} 1_{|t^w_{pred}-t^w_{ref}|<\tau} \times 100\%, \qquad (7)$$

where W denotes the total number of words in a song, and w denotes the word index. t^w_{pred} and t^w_{ref} represent the predicted and reference boundaries of the wth word, respectively. We compute PCB with two tolerance window sizes: $\text{PCB}_{0.3}$ and $\text{PCB}_{0.2}$, representing the percentage of correctly predicted word boundaries within 0.3 seconds and 0.2 seconds, respectively. We evaluate both word onsets and word offsets with results averaged over songs.

There is another commonly used evaluation metric: Percentage of Correct Segments (PCS), referring to the percentage of correct duration in the total duration [28]. This metric should consider both word onsets and offsets when determining the correct duration. However, in the practical, PCS is computed using only word onsets, with a word offset indicated by the onset of the next word [21, 18, 29]. Since this onset-only version of PCS does not help evaluate word offset estimation, we omit this metric in this paper.

4.4 Lyrics-to-Audio Alignment Results

Figure 8 illustrates the lyrics-to-audio alignment results of our proposed method, evaluated on word onsets and offsets, respectively. The results demonstrated that the masks had different effects on the estimation of word onsets and offsets. From Figure 8(a), we found that mask_onset performed best for word onset estimation with an MAE of 0.213 seconds. Mask_p followed closely behind with an MAE of 0.22 seconds.[4] By comparing the masks, we found that, for word onset estimation, masks on word onsets were more important than masks on other frames.

On the other hand, mask_onset_silence yields the best word offset estimation with an MAE of 0.332 seconds, as shown in Figure 8(b). Mask_all achieved the second best word offset estimation results with an MAE of 0.363 seconds. These results suggested that masks covering all silent frames were more effective for word offset estimation. By comparing word offset estimation results obtained by using mask_onset and mask_onset_slstart, we found that adding silence start frames in the mask (mask_onset_slstart) improved offset estimation results, while the MAE remained around 0.4 seconds. These suggested masks covering only silence start frames were insufficient to improve word offset estimation, whereas applying a strong constraint by masking all silent frames proved necessary.

Since we want to improve word offset estimation without compromising onset estimation performance, we combine the best onset results obtained by using mask_onset and the best offset results obtained by using mask_onset_silence as the results of the proposed method, as shown in Figure 2(b). The combined results are used for comparison in Table 1.

[4] For a fair comparison, we use the results obtained with the masked CE loss and without the reconstruction loss in [25] as the results of Mask_p.

	Onset				Offset				
Method	MAE ↓	MedAE ↓	$PCB_{0.3}$ ↑	$PCB_{0.2}$ ↑	MAE ↓	MedAE ↓	$PCB_{0.3}$ ↑	$PCB_{0.2}$ ↑	Training Dataset
GC [17]	0.22	0.05	94%						DALI
HBE [9]	0.23		94%						DALI
KGLW [19]	0.20		94%						DALI
HX-D [18]	0.42	**0.043**	89%	87%					DALI
HX-IH [18]	0.16	**0.043**	93%	91%					In-house 67k
DSE [23]	**0.15**		92%						In-house 88k
Mask_p [25]	0.220	0.044	95.0%	94.0%	0.426	0.078	85.4%	81.6%	DALI
Baseline (no_mask)	0.248	0.049	93.8%	90.9%	0.431	**0.060**	84.4%	79.9%	DALI
Proposed	0.213	0.044	**95.2**%	**94.4**%	**0.332**	0.063	**88.7**%	**85.4**%	DALI

Table 1: Comparison with SOTA methods. For the proposed method, the onset results are obtained by using mask_onset and the offset results are obtained by using mask_onset_silence. Bold numbers indicate the best results across all methods, while underlined numbers indicate the best results among methods trained using the DALI dataset.

4.5 A Comparison to SOTA Methods

Although our main goal is to improve word offset estimation in lyrics-to-audio alignment, we did not find any direct word offset estimation results from previous research. We present our word offset estimation results as a benchmark for future research and can only conduct a comparison on word onset estimation as follows.

Table 1 shows word onset estimation results of our proposed method (with mask_onset) and other SOTA lyrics-to-audio alignment methods. In comparison to other models trained on the DALI dataset, our model produced comparable results with an MAE of 0.213 seconds. While better MAEs were reported by models trained on large-scale in-house datasets: MAEs of 0.15 seconds and 0.16 seconds by DSE [23] and HX-IH [18], respectively. With respect to $PCB_{0.3}$, and $PCB_{0.2}$ metrics, our method outperformed all other methods, including those trained on in-house datasets.

5. CONCLUSIONS

In this paper, we extended our previous work of applying the masked CE loss to improve CTC-based lyrics-to-audio alignment. The extension aimed to enhance word offset estimation by exploring different masks in the masked CE loss. By comparing the results of using different masks, we identified the essential frames in the masks for improving word offset estimation and onset estimation: masks covering all silent frames were beneficial for word offset estimation; while masks covering word onset frames were essential for improving onset estimation. Our word onset estimation results were comparable to the SOTA results obtained by models trained on the same DALI dataset. Our word offset estimation results set a benchmark for future research since no existing results were found. In the future, we plan to explore phoneme-level lyrics-to-audio alignment and investigate the effectiveness of the masked CE loss on phoneme-level alignment.

Acknowledgments

This work was supported in part by JST CREST Grant Number JPMJCR20D4 and JST CRONOS Grant Number JPMJCS25K1, Japan. We used ABCI 3.0 provided by AIST and AIST Solutions with support from "ABCI 3.0 Development Acceleration Use."

6. REFERENCES

[1] "MIREX2024: Lyrics-to-Audio Alignment," https://music-ir.org/mirex/wiki/2024:Lyrics-to-Audio_Alignment, accessed: 2025-12-10.

[2] H. Fujihara and M. Goto, "Lyrics-to-Audio Alignment and its Application," in *Multimodal Music Processing*, ser. Dagstuhl Follow-Ups. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2012, vol. 3, pp. 23–36. [Online]. Available: https://drops.dagstuhl.de/entities/document/10.4230/DFU.Vol3.11041.23

[3] J. Kato, T. Nakano, and M. Goto, "TextAlive: Integrated Design Environment for Kinetic Typography," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (ACM CHI 2015)*, 2015, pp. 3403–3412.

[4] A. M. Kruspe, "Bootstrapping a system for phoneme recognition and keyword spotting in unaccompanied singing," in *Proceedings of the 17th International Society for Music Information Retrieval Conference (ISMIR 2016)*, 2016, pp. 358–364.

[5] E. J. Humphrey, S. Reddy, P. Seetharaman, A. Kumar, R. M. Bittner, A. Demetriou, S. Gulati, A. Jansson, T. Jehan, B. Lehner, A. Krupse, and L. Yang, "An Introduction to Signal Processing for Singing-Voice Analysis: High Notes in the Effort to Automate the Understanding of Vocals in Music," *IEEE Signal Processing Magazine*, vol. 36, no. 1, pp. 82–94, 2019.

[6] B. Sharma, C. Gupta, H. Li, and Y. Wang, "Automatic Lyrics-to-audio Alignment on Polyphonic Music Using Singing-adapted Acoustic Models," in *Proceedings of the 2019 IEEE International Conference on Acoustics, Speech, and Signal Processing (IEEE ICASSP 2019)*, 2019, pp. 396–400.

[7] A. Vaglio, R. Hennequin, M. Moussallam, G. Richard, and F. d'Alché Buc, "Multilingual lyrics-to-audio alignment," in *Proceedings of the 21th International Society for Music Information Retrieval Conference (ISMIR 2020)*, 2020, pp. 512–519.

[8] E. Demirel, S. Ahlbäck, and S. Dixon, "Low Resource Audio-to-Lyrics Alignment from Polyphonic Music

Recordings," in *Proceedings of the 2021 IEEE International Conference on Acoustics, Speech, and Signal Processing (IEEE ICASSP 2021)*, 2021, pp. 586–590.

[9] J. Huang, E. Benetos, and S. Ewert, "Improving lyrics alignment through joint pitch detection," in *Proceedings of the 2022 IEEE International Conference on Acoustics, Speech, and Signal Processing (IEEE ICASSP 2022)*, 2022, pp. 451–455.

[10] J.-Y. Wang, C.-I. Leong, Y.-C. Lin, L. Su, and J.-S. R. Jang, "Adapting Pretrained Speech Model for Mandarin Lyrics Transcription and Alignment," in *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU 2023)*, 2023.

[11] S. Rouard, F. Massa, and A. Défossez, "Hybrid Transformers for Music Source Separation," in *Proceedings of the 2023 IEEE International Conference on Acoustics, Speech, and Signal Processing (IEEE ICASSP 2023)*, 2023.

[12] M. Mauch, H. Fujihara, and M. Goto, "Integrating additional chord information into HMM-based lyrics-to-audio alignment," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 200–210, 2012.

[13] K. Schulze-Forster, C. S. J. Doire, G. Richard, and R. Badeau, "Phoneme Level Lyrics Alignment and Text-Informed Singing Voice Separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 2382–2395, 2021.

[14] Y. Ren, X. Tan, T. Qin, J. Luan, Z. Zhao, and T.-Y. Liu, "DeepSinger: Singing Voice Synthesis with Data Mined From the Web," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD 2020)*, 2020, pp. 1979–1989.

[15] G. Meseguer-Brocal, A. Cohen-Hadria, and G. Peeters, "DALI: a large Dataset of synchronized Audio, LyrIcs and notes, automatically created using teacher-student machine learning paradigm," in *Proceedings of the 19th International Society for Music Information Retrieval Conference (ISMIR 2018)*, 2018, pp. 431–437.

[16] ——, "Creating DALI, a large dataset of synchronized audio, lyrics, and notes," *Transactions of the International Society for Music Information Retrieval (TISMIR)*, vol. 3, no. 1, pp. 55–67, 2020.

[17] C. Gupta, E. Yılmaz, and H. Li, "Automatic Lyrics Alignment and Transcription in Polyphonic Music: Does Background Music Help?" in *Proceedings of the 2020 IEEE International Conference on Acoustics, Speech, and Signal Processing (IEEE ICASSP 2020)*, 2020, pp. 496–500.

[18] M. Kang, S. Park, and K. Choi, "HCLAS-X: Hierarchical and Cascaded Lyrics Alignment System Using Multimodal Cross-Correlation," arXiv 2307.04377, 2023. [Online]. Available: https://arxiv.org/abs/2307.04377

[19] T. Kick, F. Grötschla, L. A. Lanzendörfer, and R. Wattenhofer, "Contrastive Lyrics Alignment with a Timestamp-Informed Loss," in *Proceedings of the NeurIPS Workshop on AI-Driven Speech, Music, and Sound Generation*, 2024.

[20] C. Gupta, E. Yılmaz, and H. Li, "Acoustic modeling for automatic lyrics-to-audio alignment," in *Proceedings of the 20th Annual Conference of the International Speech Communication Association (Interspeech 2019)*, 2019, pp. 2040–2044.

[21] D. Stoller, S. Durand, and S. Ewert, "End-to-end lyrics alignment for polyphonic music using an audio-to-character recognition model," in *Proceedings of the 2019 IEEE International Conference on Acoustics, Speech, and Signal Processing (IEEE ICASSP 2019)*, 2019, pp. 181–185.

[22] A. Graves, S. Fernández, F. J. Gomez, and J. A. Schmidhuber, "Connectionist Temporal Classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd International Conference on Machine Learning (ICML 2006)*, 2006, pp. 369–376.

[23] S. Durand, D. Stoller, and S. Ewert, "Contrastive learning-based audio to lyrics alignment for multiple languages," in *Proceedings of the 2023 IEEE International Conference on Acoustics, Speech, and Signal Processing (IEEE ICASSP 2023)*, 2023.

[24] Y. Teytaut and A. Roebel, "Phoneme-to-Audio Alignment with Recurrent Neural Networks for Speaking and Singing Voice," in *Proceedings of the 22th Annual Conference of the International Speech Communication Association (Interspeech 2021)*, 2021, pp. 61–65.

[25] T. Cheng, T. Nakano, and M. Goto, "Improving lyrics-to-audio alignment using frame-wise phoneme labels with masked cross entropy loss," in *Proceedings of the 28th International Conference on Digital Audio Effects (DAFx25)*, 2025.

[26] L. Kürzinger, D. Winkelbauer, L. Li, T. Watzel, and G. Rigoll, "CTC-Segmentation of Large Corpora for German End-to-End Speech Recognition," in *Speech and Computer*, A. Karpov and R. Potapova, Eds., 2020, pp. 267–278.

[27] A. Mesaros and T. Virtanen, "Automatic alignment of music audio and lyrics," in *Proceedings of the 11th International Conference on Digital Audio Effects (DAFx08)*, Espoo, Finland, 2008.

[28] H. Fujihara, M. Goto, J. Ogata, and H. Okuno, "LyricSynchronizer: Automatic Synchronization System Between Musical Audio Signals and Lyrics," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 6, pp. 1252 – 1261, 2011.

[29] "MIREX2024: Lyrics-to-Audio Alignment Results," https://music-ir.org/mirex/wiki/2024:Lyrics-to-Audio_Alignment_Results, accessed: 2025-12-10.

Latent Music: Emergent Sonic Forms and Sonic Liminality in Text-to-Audio Systems

Guilherme Coelho
Audio Communication, TU Berlin
guilherme.coelho@thenewcentre.org

ABSTRACT

Text-to-audio (TTA) systems such as Udio have introduced a mode of music-making in which linguistic prompts activate high-dimensional latent manifolds to generate contingent, non-repeatable sonic artifacts. This generative architecture — operating through intersemiotic translation between linguistic signs and latent space — produces distinctive aesthetic conditions that have yet to be adequately theorized. This paper introduces latent music as an aesthetic category for sound objects produced through processes of interpolation, recombination, reterritorialization, and associative drift within these systems. Such objects occupy interstitial spaces between recognizable musical codes, timbral identities, and genre formations, producing sonic forms marked by liminality, semiotic instability, and ontological suspension. Drawing on Deleuzian aesthetics, Simondon's theory of individuation, Jakobson's concept of intersemiotic translation, and a corpus of prompt-generated artifacts, the paper develops a conceptual framework for analyzing these emergent forms. It proposes a critical vocabulary that includes relational sign units, prompt assemblages, latent catalysts, spectremes, and semiotic cartography. Through selected case studies, the paper demonstrates that latent music constitutes a sonic condition native to this technology, whose artifacts invite listening practices oriented toward emergence, semiotic drift, and the productive tensions of the not-yet-formed.

1. INTRODUCTION

Technological mediation has long shaped musical aesthetics by reconfiguring the materials, procedures, and perceptual conditions through which music is made and understood. From the tape operations of musique concrète to the glitch-oriented practices of the digital avant-garde, new tools have repeatedly altered the organization of sound, the boundaries of the musical object, and the listening habits through which sonic form is apprehended (Schaeffer, 1966; Sterne, 2003; Cascone, 2000).

Copyright: © 2025 First author et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License 3.0 Unported, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Throughout history, tools of production have introduced distinct ontologies of sound, epistemologies of listening, and new terrains of aesthetic possibility. Within this lineage, the recent proliferation of AI-based generative systems marks a significant paradigmatic shift. Text-to-audio (TTA) systems such as Udio, operate through large-scale generative models trained on extensive musical corpora, whose materials are dissolved into distributed relations within high-dimensional latent space. In this setting, prompting functions as a mode of activation: linguistic input enters a field of learned associations, shaped by the encoded sediment of existing musical works, and translated into sonic form.

Public and scholarly discussions of these systems have largely concentrated on technical improvements, replication, appropriation, authorship, and training data. This emphasis is warranted, and the ethical and legal implications of training on unconsented musical corpora remain to be resolved. Less attention, however, has been given to what is distinctive about latent synthesis as a generative modality and the specific sonic formations that emerge when generation does not stabilise into established emulations. This paper attends to that under-theorised dimension and to forms characterised by floating signifiers, semiotic drift, and the reterritorialisation of encoded musical material into emergent configurations. Such formations are typically prompted by divergent prompt assemblages: combinations of words, genre descriptors, and associative cues whose relational tensions activate the unstable regions of the latent manifold.

After many hundreds of generations across varying prompts, continuations, input audios, remixes, and similarity settings, a subset of the outputs repeatedly drew my attention. They shared a distinctive aesthetic profile: a sensation of *almostness*, of voices and instruments flickering between identities, and afterimages of genre that never become fully secure. The analysis developed here grows from repeated listening to this corpus and from an early attempt to theorise such objects.

I refer to these artifacts as *latent sound objects*[1] and to the broader aesthetic condition they instantiate as latent music. Latent music names an emergent sonic condition proper to generative TTA systems, in which musical materials arise through processes of interpolation, drift, recombination, and reterritorialization within latent space. This term

[1] **An ontological note.** By *latent sound object* I designate a mode of existence proper to model-mediated audio: an eventful condensation in a latent manifold, activated by a prompt assemblage and never closed by actualization. Such objects retain their constitutive latency; their identities are relational (to prompts, seeds, and similarity settings), historical (to continuation chains and remix lineages), and revisable (subject to reprompting, extension, and recombination). This paper focuses on a subset exhibiting a recognisable behaviour profile, here gathered under the rubric latent music.

identifies a specific formation within AI-generated music: sonic objects whose identities remain liminal and whose sonic character emerges from the recombination and cross-materialization of signifiers, timbres, and stylistic residues drawn from the manifold's substrate. Their relation to established genre categories, instrumental logics, and compositional templates is marked by a distinctive haecceity, or this-ness, that inflects these categories without merely reproducing them. The argument developed here is both phenomenological and practical: it asks how such artefacts sound, what kind of listening they invite, and how they may be approached through particular prompting strategies. These systems deconstruct, recombine, and reassemble musical forms in compelling ways.[2]

The discussion focuses primarily on outputs generated with Udio.[3] Among the public TTA systems currently available, Udio yields outputs that are less homogenized and less tightly oriented toward stabilized mainstream musical forms than those associated with systems such as Suno. Its generations are often more permeable to ambiguity, hybridization, and drift, offering especially productive access to the latent sonic artifacts considered in this paper.

The framework proposed here enters into dialogue with earlier attempts to theorize technologically mediated sonic novelty. Just as Kim Cascone (2000) theorized the aesthetics of failure, John Oswald (1985) articulated plunderphonics, and Jon Hassell (1980) developed Fourth World music as concepts rooted in the aesthetic possibilities of specific technological interventions, this paper introduces latent music as an emergent aesthetic form. The present argument also draws on Michel Chion's writings on listening (1994), Christoph Cox's sonic materialism (2011), Kodwo Eshun's sonic fiction (1998), and Thomas DeLio's work on nonlinear form (2002), alongside Deleuzian concepts such as deterritorialization, reterritorialization, and becoming (Deleuze & Guattari, 1980), and Goodman's writings on Ways of Worldmaking (1978).[4]

2. LATENCY AS CONCEPT AND CONDITION

The term *latent* derives from the Latin *latens*, meaning "hidden," "concealed," or "lying in wait" (Oxford English Dictionary n.d.). Across disciplinary contexts — from psychoanalysis and information theory to biology and philosophy — latency names a state of suspended potential: something present but unmanifest, structurally embedded but not yet actualized and awaiting conditions for emergence. This double register of structural presence and formal absence makes latency a productive concept for theorizing the sonic artifacts produced by TTA systems, whose outputs materialize from vast encoded fields of possibility that, prior to the act of prompting and generation, exist only as material substrate — unarticulated, unimagined, and awaiting the specific assemblage of signs (words) that will bring them into being.

In philosophy, the concept of latency finds a productive resonance in Deleuze's (1968) theory of the virtual. Although Deleuze does not employ the term *latent* directly, his account of virtuality maps closely onto this logic. For Deleuze, the virtual constitutes an operative dimension of existence: a field of differential relations, pre-individual multiplicities, and intensive potentials from which actual entities emerge. The virtual possesses its own productive consistency as a generative matrix, structured and operative prior to any particular actualisation. Actualization is a creative process through which the virtual differentiates itself into particular expressions, and crucially, the virtual is never consumed by this process, unlike the real. It persists alongside the actual as a reservoir of ongoing potential, an inexhaustible field from which each actualization draws without depleting. This persistence of the virtual alongside every actualized form provides a philosophical foundation for understanding TTA systems, where each generated sound object constitutes one actualization among an effectively endless field of possible outputs — a field that remains available for further traversal after every generation.[5] Simondon's (2005) concept of individuation complements this framework by emphasizing the metastable conditions from which forms are crystallized. For Simondon, individuals emerge from systems rich in potential energy, and the process of individuation is never complete; a pre-individual reservoir accompanies every individuated form, sustaining the possibility of further transformations.

Applied to TTA generation, this means that every sound object carries the trace of its generative conditions — the broader field of sonic potentiality from which it was momentarily drawn and to which it implicitly refers. The linguistic prompt functions here as the catalyst of individuation: a specific constellation of words, genre

[2] **Audio Examples.** These initial examples demonstrate the characteristic liminality and interreferential drift of latent music: https://drive.google.com/drive/folders/1eEur_uVxNfuBnw57pcYgD8K-hoVaaPqL?usp=sharing

[3] Udio is available at https://www.udio.com/home. The outputs discussed in this paper were generated using Udio's v1 and v1.5 models, released in 2024. Future versions of the platform may differ significantly in their generative behaviour: Udio has announced a forthcoming licensed iteration developed in partnership with Universal Music Group, which may introduce constraints on output characteristics, download functionality, and the breadth of stylistic territories accessible through prompting.

[4] Another useful musical point of orientation lies in practices that cultivate unstable genre adjacency, synthetic ambiguity, and hybridized sonic identity. Labels such as Orange Milk Records provide one precedent for listening to forms that hover between recognizability and estrangement, even though the generative conditions examined here are technologically distinct.

[5] In practice, this inexhaustibility is directly observable. Text-to-audio systems are stochastic: identical prompts yield different outputs with each generation. The more semantically varied or internally contradictory the signifiers within a prompt, the wider the region of the manifold that is activated, producing correspondingly more varied outputs. Furthermore, any generated sound object can be re-entered into the system and remixed with the same prompt or with entirely different prompt assemblages, initiating new trajectories from the existing material.

descriptors, and associative cues that precipitate one crystallization from the metastable field. Each resulting sound object is singular — a contingent individuation whose precise form could not have been predicted from its prompt alone and which, once generated, stands as an unrepeatable artifact of a particular traversal.[6]

In machine learning and artificial intelligence, the term *latent space* refers to the high-dimensional representational field learned by a model during training (Bengio, Courville & Vincent, 2013). Audio, text, or other forms of data are encoded into compressed vector representations that capture the structural relationships among features organized according to proximity and statistical co-occurrence within the training corpus. These relationships are further structured by taxonomic systems of tags and metatags (genre labels, mood descriptors, instrumentation categories, and artist associations) that organize the co-location of semantic concepts and sonic features within the representational field. Latent space constitutes a continuous manifold of learned associations in which similar qualities cluster together and stylistic, timbral, and generic boundaries form overlapping regions of varying density. The materials of the training corpus are dissolved into probability distributions and vector relationships that encode structural proximities across the full extent of the representation field. Generation consists of traversing this manifold: a path through the space, conditioned by input parameters, that resolves into a singular output.

These two senses of latency, philosophical and computational, converge in the generative process of TTA systems. The latent space of a generative model operates as a virtual field in a Deleuzian sense: a topology of differential relations that conditions every actualized output. Each generation constitutes an individuation in Simondon's terms: a momentary crystallization from a metastable system whose potential persists beyond the event. The sound objects produced are latent in a double sense, originating from a computational latent space and embodying latency as an ontological condition: emergent actualizations from a field of perpetual potentiality.They are spawned through prompt assemblages whose relational interplay activates regions of the manifold and precipitates sonic forms shaped by the combinatorial logic of the signs that summon them. This double latency grounds the concept of latent music developed below.

3. GENERATIVE SYNTHESIS, INTERSEMIOTIC TRANSLATION AND THE SPAWNING OF SOUND

3.1 The Generative Pipeline

TTA systems such as Udio generate sound through a multistage pipeline in which linguistic input is transformed into audio output. Although commercial architectures remain proprietary, current TTA research generally combines a text encoder, a generative model conditioned on text embeddings, and an audio decoder (Huang et al., 2023; Liu et al., 2023). [7] Text-audio contrastive models such as CLAP align language and audio representations in shared embedding spaces (Wu et al., 2022). Stochastic sampling ensures that identical prompts can yield different outputs, making each sound object a singular traversal of the latent manifold. Each generated sound object is a singular actualization — a unique traversal of the latent manifold that cannot be precisely repeated, even under identical input conditions.

3.2 Intersemiotic Translation and the Relational Sign Unit

This generative architecture hinges on what Jakobson (1959) termed *intersemiotic translation*: the transformation of signs from one semiotic system into another, here from linguistic signs into sonic materiality. When a user enters a prompt, each word or phrase is converted into a vector embedding positioned within the model's semantic topology, where it exerts a conditioning force on the trajectory of the generation. What distinguishes TTA from other forms of digital sound production is that this translation is fundamentally associative and relational. The prompt does not specify an output in the manner of a score or instruction set but activates regions of the latent manifold through the differential relations among its constituent signs. Each output is therefore an emergent spawn from a material and prompted substrate — a generated artifact rather than a sampled reproduction of particular source materials.

I propose that each word within a prompt functions as a relational sign unit: a semiotic element whose generative force depends on its relational position within the prompt assemblage — the specific configuration of genre descriptors, timbral adjectives, artist references, mood indicators, and para-musical language that together constitute the input. A prompt assemblage operates as a constellation of sign units whose interactions are combinatorial: "avant-garde neoclassical plunderphonics" does not produce an average of genres; it initiates a transversal emergence through conflicting stylistic attractors, activating the tensions and proximities between them. The generative behavior of these sign units is rhizomatic in the sense of Deleuze and Guattari (1980): any sign unit can connect to any other, and the relations among

[6] This condition of singularity is more pronounced in large-scale TTA systems such as Udio and Suno than in models trained on smaller datasets, where the reduced size and homogeneity of the training corpus tend to produce overfitting and constrain the range of variation across generations.

[7] The general pipeline typically involves some combination of transformer-based sequence modelling, audio codec tokenisation (e.g. EnCodec), and diffusion-based or adversarial upsampling stages. Udio has not publicly disclosed its architecture; Suno's earlier open-source model Bark employed a multi-stage autoregressive transformer pipeline using EnCodec tokenisation. The analysis in this paper does not depend on the specifics of any single architecture but on the structural features common to current TTA systems: text-conditioned traversal of learned latent representations, stochastic generation, and intersemiotic translation from linguistic to sonic signs.

them propagate in emergent ways. [8] Within the latent manifold, any point of the material substrate can connect to any other, drawing together timbral qualities, production signatures, and stylistic traces from regions that share no conventional musical relationship.

3.3 Excavated Archives and Rhizomatic Emergence

It is essential to acknowledge the material substrate from which these sound objects emerge. TTA models are trained on vast corpora, which are large-scale collections of musical works, often sourced without the consent of their creators (Drott, 2024). The latent space is, in this sense, a compressed archive of human musical production: an encoded sediment of styles, performances, production techniques, and cultural histories. This situates TTA systems within a lineage that includes plunderphonics (Oswald, 1985) and sample-based music more broadly — practices grounded in the appropriation and recombination of existing sonic materials. However, the mechanisms fundamentally differ. Whereas plunderphonics operates through direct quotation and citational juxtaposition, generative synthesis works through statistical abstraction and emergent recombination. The training corpus is dissolved into probability distributions and vector relationships, and its materials are dispersed across the latent manifold in a rhizomatic fashion (Deleuze & Guattari, 1980).

Influences propagate through multiplicities of connection. The resulting sound objects are composed of what I term *spectremes*[9]: spectral traces of prior musical material that structure every generated output while remaining irrecoverable as themselves — latent presences that condition the sonic object without being individually identifiable as the source. Unlike plunderphonics, where source materials remain audibly present as quotations, TTA generation dissolves its sources into the latent manifold. The resulting sound objects carry a deep spectral indebtedness to the works of the training corpus. While no singular source can be fully recovered from the output in the general case, there are nonetheless instances in which targeted metatag-based prompting produces outputs whose artistic likeness and compositional proximity to specific source materials are reproducible and systematically identifiable (Coelho, 2025). [10] What persists across these cases are spectremes — distributed resonances, emergent colorations, stylistic residues — that shape the generated object from within while remaining individually untraceable.

4. DEFINING LATENT MUSIC: SONIC FEATURES AND AESTHETIC CONDITIONS

A curated selection of latent music sound objects was compiled to support the following discussion. [11] All examples are direct, unmodified outputs from Udio, presented without post-processing or additional production, preserving their character as artifacts of latent space traversal. Readers are encouraged to listen to these examples while engaging with this section to apprehend the sonic characteristics being described.

4.1 Toward a Working Definition

Latent music names a category of sound objects generated through AI-mediated processes of intersemiotic translation that exhibit liminal formation, semiotic drift, and emergent morphogenesis as constitutive features. These are threshold objects — sonic forms that dwell in the interstitial sign-spaces between genres, timbres, instruments, and established musical codes, resisting the categorical stability that typically organizes musical perception. Latency itself functions as a defining aesthetic condition: these sound objects exist in states of perpetual becoming, their sonic identities emerging and dissolving across their duration, approaching recognizable forms without stably representing them.

Four interrelated characteristics define this sonic condition. First, *interstitial positioning*: latent music materializes from the spaces between recognizable territories where established musical categories thin out and overlap. A sound object may present timbral qualities suggestive of bowed strings, yet the texture shades into granular synthesis or vocal resonance; a pattern gestures toward a genre before dissolving into another. These artifacts emerge from semiotic interstices — zones of the latent manifold where encoded associations from distinct musical domains converge, producing sonic forms that inhabit several adjacent territories simultaneously without settling decisively into any one of them.

Second, *spectral liminality*: latent music carries traces of instruments, stylistic codes, production signatures, and

[8] Derrida's (1976) notion of the *trace* offers a semiotic complement to this argument. For Derrida, the trace of absent signs remains operative in any act of signification — meaning arises through deferral and difference, through networks of relation rather than self-present identity. Applied to the prompts and sonic objects under discussion, this suggests that their meaning is always relational, always shadowed by the sounds, genres, and cultural codes that condition their intelligibility yet remain present as differential residue.

[9] *Spectreme*: a neologism combining *spectre* (after Derrida's hauntology) and *spectrum* (acoustic analysis), formed on the model of *phoneme* and *morpheme*. Spectremes designate the traces of prior musical material that compose every AI-generated output — latent presences that structure the generated object without being individually identifiable or recoverable as source.

[10] Coelho, G. (2025). The Artist Is Present: Traces of Artists Residing and Spawning in Text-to-Audio AI. *Proceedings of the AI Music Creativity Conference (AIMC 2025)*, Brussels. Available at: https://zenodo.org/records/16946623. The paper demonstrates that strategically constructed metatag-based prompt assemblages based on public taxonomies of artists profiles can systematically microlocate artist-conditioned regions of the latent manifold, yielding reproducible stylistic proximities to specific artists.

[11] Audio repository available at: https://drive.google.com/drive/folders/1eEur_uVxNfuBnw57pcYgD8K-hoVaaPqL?usp=sharing

sonic identities that surface at the threshold of recognition. A timbre may suggest a bowed string without resolving into one, and a textural quality may evoke a specific production technique that, on closer listening, proves unplaceable. These traces inhabit a liminal zone between presence and absence, being audible enough to inflect the character of the sound object, but too diffuse to be isolated as discrete sources. They are a joint consequence of the training corpus's distributed presence within the latent manifold and the synthesis process itself, which generates sound by traversing and interpolating between encoded regions rather than reproducing any one of them — spawning emergent sonic forms whose timbral and stylistic identities are recombined in unfamiliar configurations. [12]

Third, *ontological suspension*: the forms feel immanent yet unfulfilled, gesturing toward stable identities that defer their arrival. There is a persistent sense of *almostness* and representational suspension — of sonic material on the verge of becoming something recognizable, something nameable, that remains instead in a condition of productive irresolution. This suspension constitutes a defining quality that distinguishes latent music from conventional composition and genre-faithful AI generation.

Fourth, *semiotic drift*: The direction and character of a latent sound object remain fundamentally unpredictable across its duration. The listener encounters a sonic trajectory whose unfolding resists anticipation, where genre traces, timbral identities, and formal gestures appear, migrate, and dissolve according to a logic internal to the generative process and opaque to expectation. This condition of anticipatory uncertainty, of listening without a stable horizon of expectation, is intrinsic to the latent character of these objects of sound.

A prompt such as “deconstructed chamber strings plunderphonics glitch” may yield fragmented string gestures interweaving with indeterminate processed textures: the timbral memory of chamber music present as inflection, gestures that point toward a musical grammar without committing to its syntax. Listeners encounter identities that hover at the edge of recognition. The music itself sounds latent.

Latent music exhibits distinctive material and formal characteristics: *timbral liminality* (sounds at thresholds of instrumental recognition—a guitar-like timbre shading into bowed artifact, a vocal presence hovering between human and synthetic); *genre slippage* (sliding between stylistic territories, transitions feeling like drifts rather than modulations); *gradient morphogenesis* (processual evolution through continuous transformation); *referential opacity* (haunted by influence rather than anchored by citation); *sonic spectrality* (fragments briefly surfacing before dissolving); *aesthetic indeterminacy* (suspending rather than fulfilling expectation); and *cross-pollination* (materials intermingling without fusing into stable synthesis, an aesthetic of friction and adjacency).

These characteristics produce a listening experience governed by *interreferential drifts*, where sonic outputs appear proximate to known signifiers without reproducing them. Recognition is triggered but not fully consummated. Musical signs operate adjacently; genre tropes become floating signifiers, and meaning accumulates through suggestion and association.

4.2 Spectral Semiotics and Modes of Reterritorialisation

A defining trait of latent music is its unresolved semiotic regime: a sonic surface populated by spectres. Sound objects produced through prompt assemblages operate through *floating signifiers*, drifting across aesthetic domains and accumulating partial meanings without categorical anchorage. Traces of genre, idiom, and production artifacts flicker across the material without settling into a stable reference. A track may begin with a groove suggestive of IDM, only for that groove to distort, stretch, and fragment until the rhythmic identity collapses toward something else. What emerges is an aesthetic of *almostness*: the sonic object feels close to something recognizable yet drifts elusively. Following Derrida's (1967) *différance*, meaning here is perpetually deferred: the latent sound object accumulates partial significations without achieving stable categorical anchorage.

The dynamics of deterritorialization and reterritorialization (Deleuze & Guattari, 1980) provide a productive framework for the unresolved semiotic regime of latent music. In latent music, familiar sonic codes — genre signatures, timbral conventions, and compositional idioms — are invoked and simultaneously disassembled and dislodged from the territories in which they conventionally operate. These dislodged materials are then drawn into new configurations through an ongoing and unresolved process of reterritorialization: sonic materials from distinct domains are perpetually displaced, partially re-grounded, and brought into emergent relations, producing objects whose identities are defined by their movement across territories. Reterritorialization in this context refers to a continuous process of cross-materialization: materials from the substrate of one encoded territory — its timbral signatures, its rhythmic logics, its production conventions — permeate and saturate materials from another, each inflecting the other through what can be described as *sonic osmosis*.

The latent manifold operates as a semi-permeable membrane through which encoded musical materials migrate, merge, and partially reconstitute themselves in forms that carry the residue of their origins while belonging to none of them exclusively. A prompt such as "deconstructed choir plunderphonics glitch" precipitates a reterritorialized assemblage: choir gestures saturated with the aesthetics of sampling, deconstructive fragmentation inflected by choral resonance — each territory bleeding into and emerging from the material substrate of the others. Music functions here as a medium in the chemical as well as

[12] This condition is audible in this file: https://drive.google.com/file/d/1x9Jy65MFM_IratlHYtZ7bx0z_ydrOSE4/view?usp=sharing

communicative sense: a substrate through which materials pass, transform, and recombine.

4.3 Gradient Identities and Asignifying Ruptures

These sound objects can be understood through Baudrillard's (1994) theory of simulacra: copies without originals, entities in which the distinction between source and reproduction collapses.. Latent music does not straightforwardly reproduce or represent existing musical works, it is emergently spawned from the statistical sediment of the training corpus, referencing sonic materials whose origins have been dissolved into distributed relations. The outputs tend toward what Baudrillard terms the fourth order of simulation — a domain in which signs circulate among other signs, and where questions of authenticity or fidelity to a singular source become increasingly difficult to pose, even as traces of that source material remain operative within the generated object.

Within this simulacral condition, sonic identity itself becomes more gradual. What I call gradient identities names a mode of sonic existence in which musical properties — timbre, genre, instrumentation, production style — exist along continuous spectra, reflecting what DeLanda (2002) describes as intensive morphogenesis. The identity of a latent sound object occupies regions rather than a specific point, drifting across its duration, between adjacent sonic territories. A texture may begin as something resembling a piano and migrate, without a clear boundary, into a timbre closer to a prepared instrument or a granular vocal fragment. These are *flickering identities* — sonic presences that oscillate between recognizable poles, momentarily invoking one territory before shading into another, their character defined by the movement itself.

This gradient condition is further marked, and at times interrupted, by what Deleuze and Guattari (1980) term *asignifying ruptures*: breaks in the semiotic chain where signs cease to function within their conventional systems of meaning. In the context of latent music, asignifying ruptures name those moments where the sonic material escapes recognizable musical codification entirely or where the material abruptly shifts from one semiotic regime to another. These ruptures are structural consequences of the generative process. When a prompt assemblage combines sufficiently divergent or contradictory sign units, the model's traversal of the latent manifold may pass through regions where encoded associations thin out, overlap incoherently or cancel each other's categorical pull. The result is sound dislodged from stable semiotic frameworks — sound that demands attention on its own material terms. Asignifying ruptures make the limits of musical signification itself audible, and these are the moments where latent music is most fully itself — where the emergent, substrate-dependent, and combinatorially singular character of TTA generation becomes most audible.

5. PROMPTING STRATEGIES AND THE NAVIGATION OF LATENT SPACE

5.1 Latent Catalysts and Semantic Triggers

Latent music is fundamentally shaped by the construction of prompts. In TTA systems, prompts function as semiotic scaffolds — provisional frameworks of sonic reference that configure the generative trajectory. The most compelling latent music tends to originate from prompts that are semantically ambiguous or compositionally eclectic, assembling divergent cues without resolving them into a coherent syntax.

Two categories of prompt elements deserve particular attention. The first is what I term *latent catalysts*: individual words or combinations that, when placed within any prompt assemblage, tend to trigger rhizomatic exploration across the latent manifold, regardless of the other sign units present. Terms such as "plunderphonics," "deconstructed," or "experimental" function as catalysts because their semantic profiles are inherently triggering and unstable — they do not point toward a fixed sonic territory but toward operations of fragmentation, recombination, and displacement. Their presence in a prompt assemblage destabilizes the gravitational pull of more genre-specific terms, opening the generation toward reterritorialized zones. A term such as 'plunderphonics,' for instance, will reshape the generative trajectory of any assemblage it enters: 'classical plunderphonics,' 'plunderphonic nu-metal,' and 'plunderphonic ambient' will each yield fundamentally different directions to their referential signifiers, yet all share the destabilizing, deconstructive orientation that the catalyst introduces.

Similarly, the second category is *semantic triggers*: combinations of words that destabilize genre or emotional anchors through internal contradictions or paradoxes. Pairings such as "nihilistic joy" or "metal ambient" create semantic fields that the model cannot resolve into a single affective or stylistic register, producing outputs characterized by tonal ambiguity and affective oscillation.

5.2 Latent Vocabularies and Semiotic Cartography

A crucial dimension of prompt construction lies in the vocabulary available to the semiotic navigator. Public music taxonomies — most notably RateYourMusic (RYM) and its associated database Sonemic — constitute extensive community-built repositories of genre descriptors, micro-genre labels, and stylistic qualifiers that map the breadth of documented musical production. These taxonomies provide what functions, in the context of TTA systems, as a *latent vocabulary*: a reservoir of culturally encoded terms whose presence in the model's training metadata makes them operative within the latent manifold as conditioning forces on generation. As I have discussed at greater length elsewhere (Coelho, 2025), these taxonomic vocabularies enable the targeting of both macro and micro-locations within the manifold, with broad genre

terms establishing a gravitational field while micro-level descriptors introduce the tensions and lateral associations from which interstitial formations emerge. Prompt assemblages that draw on dozens of tags across multiple artist profiles and genre references tend to yield more varied and interstitially positioned outputs than brief, general-purpose prompts. The public taxonomies of RYM and similar repositories can serve as a shared, evolving lexicon through which the encoded musical territories of the latent manifold become navigable and targeted.

6. CASE STUDIES

6.1 Plunderphonic Sound Object: Iterative Extension from a Single Prompt[13]

This sound object originated from the prompt: *"john oswald plunderphonics experimental, deconstructed, classical music, stockhausen, no vocals."* An initial 32-second generation was repeatedly extended in 32-second increments using the same prompt, without modification to the assemblage at any stage. Additional variations are available in the audio folder.

The prompt assembles several latent catalysts and genre descriptors into a single constellation. "Plunderphonics" and "deconstructed" function as catalysts in the sense defined in Section 5: their semantic instability destabilises the more genre-anchored terms ("classical music," "stockhausen"), preventing the generation from settling into faithful stylistic reproduction. What is audible across the object's duration is a sustained process of reterritorialization and cross-materialization. Piano textures migrate into accordion-like timbres, electric guitar traces surface briefly before dissolving into woodwind sonorities, and passages evocative of MIDI-sequenced music appear and fade into granular textures. These instrumental identities do not succeed one another in discrete sections; they co-exist, co-form, and interpenetrate, each carrying the spectral residue of the others. The result is a field of continuous gradient transition between recognizable poles, in the manner described in Section 4.3.

The object exemplifies several defining features of latent music. Its interstitial positioning is audible in instrumental identities occupying zones between established timbral categories. Its spectral liminality surfaces in the trace-like quality of its sonic references: classical music gestures present as coloration rather than stable generic commitment. Its semiotic drift is evident in the unpredictability of each 32-second extension's trajectory, even as the overall object maintains an emergent coherence across its accumulating length.

The iterative extension method is significant. Because the same prompt is applied at each stage, the object demonstrates how a single assemblage can rhizomatically yield continuously varied material. Each extension traverses a different region of the latent manifold while remaining conditioned by the same configuration of sign units. This generative inexhaustibility is a direct consequence of the stochastic and combinatorial character of latent space traversal discussed in Sections 2 and 3.

6.2 Ontological Suspension and the Almost-Formed[14]

Where Case A demonstrates latent music's capacity for sustained reterritorialization across iterative extensions, this second case study illustrates a different dimension of the latent condition: *ontological suspension*, the quality of almost representing something without fully becoming it. Two sound objects are described, each from a single generation; additional variations are available in the audio folder.

The first object (6.2.1) was generated from the prompt: *"tape music, electroacoustic, ambient, glitch, toypop, new age, surreal, playful, avant-garde, atmospheric, mellow, peaceful, sparse, bittersweet, soft, ethereal, melancholic, sampling, minimalistic, quirky, mysterious, cryptic, warm, soothing, hypnotic, psychedelic, sentimental, futuristic, instrumental, nostalgia, childhood, piercing."* The density of this assemblage, with over twenty-five sign units spanning genre descriptors, affective qualifiers, and para-musical references, deters the model from resolving toward any single genre territory. What emerges is an object in which blips, woodwinds, and strings glide in pitch between synthesis representations and organic timbres, a sonic surface where instrumental identity is perpetually suspended between the electronic and the acoustic, the synthesized and the performed.

The second object (6.2.2) was generated from the prompt: *"glitch, plunderphonics, electroacoustic, experimental, deconstructed, indeterminacy, classical music."* This compressed assemblage is dense with latent catalysts — *plunderphonics*, *deconstructed*, *indeterminacy* — each destabilising the assemblage and preventing the generation from producing a stable classical-music output. The resulting sound object occupies a liminal ontological space in which synthesis behaviors and strings merge, blend, and dissolve into one another. Classical gestures are audible at times, a bowed quality, a sense of harmonic movement, but they remain on the verge of consolidating without arriving.

Across both objects, identities are held by a pervasive sense of latent instability. Rather than drifting in the manner of Case A's cross-materialising transitions, they inhabit a sustained condition of liminality — an ontological estrangement in which the sound object's identity remains suspended between representations.

7. LISTENING ORIENTAITONS: TOWARD A LATENT LISTENING AND SEMIOTIC MUSICKING

Latent music invites a mode of listening that is fundamentally associative, porous and speculative. Where traditional musical forms guide listeners through teleological structures of tension,

[13] Link to audio files: https://drive.google.com/drive/folders/1uncbX919H1kDoq3vuvsW1bFYqUMbxZ6P?usp=sharing

[14] Link to audio files. https://drive.google.com/drive/folders/1jtoBMaX0aLia4oyEHbCO86V644Tv-XLz?usp=sharing

development, and closure, latent sonics suspend these logics. The listener drifts across an incomplete, suggestive, and permeable terrain, navigating aesthetic adjacency: openness to what is nearly recognizable, to what almost coheres, to what might be and what might have been. Additionally, because no latent sound object is rigidly repeatable, each listening event feels *encountered* rather than received.

This speculative, associative mode of reception constitutes what can be called semiotic musicking: an extension of Christopher Small's (1998) concept of musicking into the domain of semiotic engagement. In Small's formulation, all participants in a musical event, including listeners, are musicking. Latent music intensifies this participatory condition: because the sound object's semiotic identity is unstable and its referents are floating, the listener is drawn into an active process of semiotic construction — assembling meaning, projecting genre associations, tracing timbral lineages, and navigating the spectral ecology of the latent manifold through the act of listening itself. Listening to latent music is musicking at the level of signs.

CONCLUSION

This paper has developed a framework for understanding latent music as a distinct aesthetic category within the broader field of AI-mediated sound production. The work presented here has developed concepts for considering these emergent forms — their distinctive sonic character, the generative and intersemiotic conditions from which they arise, and the modes of listening and compositional practice they invite.

The paper has proposed several original concepts — relational sign units, prompt assemblages, latent catalysts, semantic triggers, spectremes, latent genres, and semiotic cartography — as a critical vocabulary for engaging with this emerging domain. The case studies and prompting strategies presented here exemplify the particular potentialities and ontological uniqueness of TTA systems and generative AI more broadly.

ETHICS STATEMENT

This research was conducted using the publicly available Udio text-to-audio system. All sound examples are direct, unmodified outputs. I acknowledge the legal and ethical disputes surrounding generative music systems, including claims concerning unconsented musical corpora. This analysis aims to advance theoretical understanding while acknowledging the broader ethical implications of generative technologies in creative fields. This paper was prepared with the assistance of large language models used for refining prose and strengthening articulation. The conceptual framework, theoretical arguments, analytical vocabulary, and critical claims were developed through the author's reading, listening, and writing practice.

REFERENCES

Baudrillard, J. (1994). *Simulacra and Simulation*. University of Michigan Press.

Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8), 1798–1828.

Cascone, K. (2000). The aesthetics of failure: “Post-digital” tendencies in contemporary computer music. *Computer Music Journal*, 24(4), 12–18.

Chion, M. (1994). *Audio-Vision: Sound on Screen*. Columbia University Press.

Coelho, G. (2025a). The Artist Is Present: Traces of Artists Residing and Spawning in Text-to-Audio AI. In *Proceedings of the AI Music Creativity Conference (AIMC 2025)*, Brussels. https://doi.org/10.5281/zenodo.16946623

Cox, C. (2011). Beyond representation and signification: Toward a sonic materialism. *Journal of Visual Culture*, 10(2), 145–161.

DeLanda, M. (2002). *Intensive Science and Virtual Philosophy*. Continuum.

Deleuze, G. (1968). *Difference and Repetition*. Translated by P. Patton. Columbia University Press.

Deleuze, G., & Guattari, F. (1980). *A Thousand Plateaus: Capitalism and Schizophrenia*. Translated by B. Massumi. University of Minnesota Press.

DeLio, T. (2002). *Circumscribing the Open Universe*. University Press of America.

Derrida, J. (1967). *Of Grammatology*. Translated by G. C. Spivak. Johns Hopkins University Press.

Drott, E. (2024). Copyright, compensation, and commons in the music AI industry. *Creative Industries Journal*, 17(1), 28–46.

Eshun, K. (1998). *More Brilliant than the Sun: Adventures in Sonic Fiction*. Quartet Books.

Hassell, J., & Eno, B. (1980). *Fourth World Vol. 1: Possible Musics* [Album]. EG Records.

Huang, Q., et al. (2023). Noise2Music: Text-conditioned music generation with diffusion models. *arXiv preprint arXiv:2302.03917*.

Jakobson, R. (1959). On linguistic aspects of translation. In R. A. Brower (Ed.), *On Translation* (pp. 232–239). Harvard University Press.

Liu, H., et al. (2023). AudioLDM: Text-to-audio generation with latent diffusion models. *Proceedings of ICML 2023*.

Oswald, J. (1985). *Plunderphonics*. Mystery Laboratory.

Oxford English Dictionary. (n.d.). Latent. In *OED Online*. Oxford University Press.

Schaeffer, P. (1966). *Traité des objets musicaux*. Éditions du Seuil.

Simondon, G. (2005). *L’individuation à la lumière des notions de forme et d’information*. Éditions Jérôme Millon.

Small, C. (1998). *Musicking: The Meanings of Performing and Listening*. Wesleyan University Press.

Sterne, J. (2003). *The Audible Past: Cultural Origins of Sound Reproduction*. Duke University Press.

Wu, Y., et al. (2022). Large-scale contrastive language–audio pretraining with feature fusion and keyword-to-caption augmentation. *Proceedings of ICASSP 2023*. IEEE.

Amplifying Participation. The digital Barlow Archive (dBA) as an Approach to the Recording of a Digital Computer Music Legacy

Fabian Czolbe
ligeti zentrum | Hochschule für Musik und Theater Hamburg
fabian.czolbe@hfmt-hamburg.de

Julian Rohrhuber
Institute for Music And Media (IMM)
Robert Schumann Hochschule Düsseldorf
julian.rohrhuber@rsh-duesseldorf.de

Bernd Härpfer
Independent Researcher
haerpfer.net

ABSTRACT

The archiving of computer music presents specific challenges that arise from the process-oriented, software-based, and technologically contingent nature of digital compositional practices. Digital artifacts such as source code, algorithmically generated data, and custom compositional tools encode not only musical outcomes but also procedural knowledge that is often implicit and difficult to formalize. This paper does not propose a comprehensive framework for an archive of Barlow's digital legacy. Instead, its objective is to elucidate the process of establishing such an archive and its structural framework, to underscore salient inquiries, and to facilitate a forum for discourse. The Digital Barlow Archive (dBA) can be regarded as a case study in addressing these challenges through a translational and participatory approach to archiving computer music.

The born-digital legacy of Clarence Barlow (1945–2023) is open-ended and comprises heterogeneous materials. For doing justice to this openness and diversity, the dBA adheres to existing archival standards while extending them to account for computer-music-specific objects and workflows. Thereby, an object/event framework is employed to translate non-linear and iterative compositional processes into structured metadata representations that remain interoperable with institutional and international archival infrastructures. At the same time, the framework acknowledges the limits of formalization and preserves interpretative openness.

Extending the old idea of computing as an amplification of the intellect, this paper argues that such archival methods do not only passively conserve material, but need to translate and amplify the possibility of participation: they actively shape access, interpretation, and creative reuse of digital musical materials. Archiving should be conducted as an epistemic practice that mediates between technological history, compositional knowledge, and the contemporary computer music community.

1. INTRODUCTION

The Digital Barlow Archive (dBA) is conceived as an organization that provides distributed, research-oriented infrastructure for the preservation, and scholarly use of the digital legacy of Clarence Barlow (1945–2023). Beyond itscoordinating function for research activities, it extends the scope of long-term digital preservationto include the systematic structuring, annotation, and contextualization of heterogeneous digital artifacts. These materials document compositional workflows, algorithmic strategies, and technological developments at the intersection of computer music, algorithmic composition, and media art (Boutard, 2013). Within this framework, the archival description of such materials necessarily involves acts of translation, understood in a dual sense. Operationally, translation signifies the concrete procedures by which compositional processes, technological dependencies, and implicit forms of knowledge are mapped onto structured, interoperable representations within archival systems.[1] Concurrently, translation functions as a conceptual and philosophical framework, insofar as these processes entail not merely transfer but transformation: forms of practice-bound, often tacit knowledge are reconfigured within the constraints and affordances of formalized descriptive schemas. Archival representation, therefore, is not merely a neutral encoding; rather, it is a mediating operation that reshapes the epistemic status of the digital object itself. In this context, the oeuvre of Barlow assumes particular relevance. While maintaining a close connection to contemporary discourses in creative coding and computational musicology, he developed custom compositional software environments and formal models. In this respect, his practice can be regarded as a prototypical example of later paradigms in the domain of computer-assisted and algorithmic music. Consequently, it offers a particularly illuminating perspective on the translational processes that underpin archival formalization.

In his cybernetic concept of the amplification of intelligence, W. Ross Ashby describes how structured systems extend human capacity to act within complex environ-

[1] The theoretical foundation of our project is broad, drawing from the concepts of scholars such as Jacques Derrida, Michel Foucault, Bruno Latour, and Wolfgang Ernst. Our objective is to identify approaches that facilitate knowledge access across diverse disciplines and medial dimensions. The development of formal, declarative models, such as metadata frameworks and ontologies, is a central tenet of this endeavor, with the objective of ensuring the effective management and preservation of digital objects. In this context, Ippolito and Rinehart's variable media framework offers practical descriptive tools from a media art perspective that support the documentation, recreation, and preservation of media art, while foregrounding the interpretive decisions inherent in any act of archival translation. (Ippolito & Rinehart, 2014)

Copyright: © 2026 Fabian Czolbe et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License 3.0 Unsorted*, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.*

ments by regulating variety rather than reducing it (Ashby, 1956). When transferred to the context of digital archives, this concept can be rephrased as an amplification of participation: the capacity of users to meaningfully engage with heterogeneous, process-oriented materials. From this perspective, archival taxonomies and metadata models do not merely organize information, but actively mediate access, interpretation, and reuse. The dBA approaches the archiving of computer music accordingly, understanding archival structure as a means of amplifying participation while preserving the complexity inherent to software-based compositional practices.In the context of the dBA, participation refers to the collaboration of composers, music theorists, musicologists, archivists, and media theorists regarding taxonomies, categorization, and references within the digital estate. However, we are not only focused on cataloging archival objects; we are also exploring potential artistic and scholarly perspectives on these objects and structures. Additionally, participation can reveal opportunities to develop ideas or explore new approaches based on the dBA.

The analogue components of Barlow's estate are currently being incorporated into the music archive of the Academy of Arts in Berlin. In contrast, the dBA focuses on the extensive body of born-digital material and is designed to be interoperable with both this institutional archive and other digital platforms within the international computer music community (Boutard, 2016). In conjunction with related resources—such as performance-oriented materials, documentation of realizations, and software-based work descriptions—the dBA enables research across different institutions. In this sense, the archive functions as an interface that translates between institutional archival standards and situated compositional practices, thereby shaping modes of participation for researchers, composers, and artists (Akkermann, 2021).

The necessity of such a distributed digital archive arises directly from the characteristics of Barlow's working methods. He developed idiosyncratic digital composition tools, frequently in the form of custom-written code and algorithmic systems, theorized their use, and continuously integrated teaching, research, and composition. Consequently, we have to do with a wide range of digital objects: source code for algorithmic compositions and compositional tools, algorithmically generated datasets which function as precursors to scores or as intermediate representations within compositional processes, digitally produced texts and graphics, as well as audio and video files documenting renderings, simulations, and performances.

These objects resist straightforward classification: they often function simultaneously as tools, documentation, and aesthetic artifacts. Hence, the dBA needs to address two interrelated challenges: the development of a suitable taxonomy for archival description that builds upon existing metadata standards while remaining sensitive to the specificities of computer music artifacts, and the contextualization of these objects within technological, compositional, and computer-music-historical frameworks. Essentially, both challenges are considered as translational problems. The manner in which these challenges are addressed has a significant impact on access, interpretation, and, ultimately, future engagement in an open-ended mode of compositional thinking.

2. THE ARCHIVAL OBJECTS

The material of the dBA currently comprises several hard drives containing multiple terabytes of data. The archived items include source code written in various programming languages, algorithmically generated datasets that serve as precursors to scores or as intermediate stages in compositional workflows, digital components of writings and teaching materials such as texts, diagrams, and graphics, as well as audio and video files documenting simulations, sound synthesis results, rehearsals, and performances. It assembles the unfinished with the finished, seriousness with jokes. From the perspective of computer music research, these materials are best understood not as fixed works, but as representations of compositional processes, toolchains, and software environments. As Laura Zattra clearly demonstrates, the structures emerging here allow conclusions to be drawn about working methods, but also about certain interpretations, via folder logic, containers, and metadata. The dBA should therefore be understood as the born-digital equivalent of this "work process" reconstruction: objects (files, tools, code fragments) and events (versioning, executions, performances) make Barlow's digital workshop and thinking comprehensible. (Zattra, 2021) Their archival treatment therefore requires a translation of procedural and performative practices into describable and retrievable archival objects.

For initial cataloguing, the dBA adopts the taxonomy developed by the archive project *arkumu.nrw*. Each digital object is first associated with an institution, an author, and a specific event or several of each of these. Events may refer to a musical composition, a software development phase, a lecture, or a broadcast, and are defined using identifiers such as Wikidata, with optional links to GND or LIDO event definitions. This semantic enrichment is based on established international standards for the authenticity, accuracy, and reliability of digital documents (Lee 2006), and it supports interoperability with archival, musicological, and digital humanities infrastructures. Simultaneously, it is a framework within which computer-music-related practices may become legible.

In the case of computer music materials, the accompanying metadata is extended to include information on hardware configurations, operating systems, programming environments, and software dependencies. Object-level description then follows using parameters such as media type, preservation strategy, genesis, licensing status, MIME type, file modification dates, object type, system requirements, and technical metadata extracted with tools including JHOVE, DROID, MediaInfo, metadata-extractor, and ExifTool.[2] The selection and granularity of these parameters directly influence which form of engagement

[2] JHOVE is a file format identification, validation and characterization tool. DROID is a software tool developed by The National Archives to perform automated batch identification of file formats and uses internal signatures to identify and report the specific file format and version of digital files. These signatures are stored in an XML signature file, generated from information recorded in the PRONOM technical registry.

—analytical, historical, or creative—are enabled, and thus shape conditions of participation within the archive.
The talk will present selected examples from Barlow's digital legacy, demonstrate their taxonomic classification, and discuss the extent to which the resulting metadata supports research and creative reuse. In this taxonomy, particular attention is given to gaps in event- and object-level description that arise specifically from computer music practices and that may require iterative or participatory refinement.

3. THE OBJECTS / EVENTS FRAMEWORK

The adequacy of taxonomic description leads to the second cornerstone of the project: the objects / events framework. As with many artistic estates, Barlow's digital legacy is characterized by heterogeneity and a certain inconsistency. Similar conditions can be observed in historical composer estates, where materials often document incomplete or open-ended creative processes. In the context of computer music, this condition is intensified by the iterative, exploratory, and software-driven nature of composition.
Individual events—such as the development of an algorithm, the testing of a synthesis method, or the preparation of a specific composition—produce multiple digital objects that may diverge, branch, or re-converge at later stages. These objects form non-linear, asynchronous timelines of genesis. Accordingly, the dBA requires the systematic development of an object-/event-oriented metadata structure which, following Waters, traces a participatory, process-sensitive documentation of compositional processes. (Waters, 2006) Here, the object/event framework functions as a translational model that renders such processes readable within archival systems without imposing a linear or teleological narrative. In Barlow's practice, digital tools and code fragments are frequently developed in response to specific situations and later reintegrated into different compositional contexts, further challenging definitive classification. (Barlow, 2011)
These challenges are directly reflected in the current state of the dBA's development. We are currently developing a taxonomy specific to the dBA while prototypically cataloging individual thematically related data sets within specific file folders. It has become apparent, however, that the framework provided by arkumu.nrw can serve only as a starting point. On the one hand, there should be sufficient options for assigning files—that is, objects—to the individual event categories. On the other hand, free-text comments will be necessary to share expert knowledge. This raises the question of how, over time, research findings developed with the help of the archive can be fed back into the cataloging system to increase the degree of interoperability among the elements in the archive. The following example illustrates this challenge; a more detailed discussion will be provided in the accompanying presentation. In connection with a specific composition or score, there are a number of files in various formats, such as proprietary Sibelius files, MIDI files, as well as DOC and PDF files. All files belong to the same event, but it is not always immediately clear what dependencies exist between them and how they should therefore be linked within the system. A MIDI file, for example, may have been generated from the final Sibelius score, but it may also represent a preliminary stage, a section, or a voice within the composition, or it may have been generated using a self-written tool, as Barlow often did. The Sibelius files may similarly contain only certain parts of the score, necessitating comparison with, for instance, the PDF of the latest version. The DOC files, in turn, represents a glossary, a legend, a commentary, or program notes. For an initial inventory, the material may therefore be only provisionally categorized, so that in later phases researchers can incorporate their knowledge into the structure of the archive—whether consisting of expertise developed independently of the archive or emerging only through engagement with it.
While the ability of making such interdependencies legible represents a key epistemic virtue of a digital archive, the example also raises the question of how far relationships between objects, events, and technologies should be formalized at the point of archival ingestion. Not all relations can be exhaustively specified without limiting future interpretation. Participation therefore emerges not only through access to archival objects, but through the possibility of contextualization, reinterpretation, and ongoing engagement with partially specified archival structures. The dBA approach is therefore not primarily aimed at preserving bitstreams, but rather at preserving intelligible relationships between code, tools, data, and compositional processes. (Dahan, 2020) We should keep in mind that, from a cultural perspective, each artifact is of a transitional character – it has a past open to interpretation, and may become part of a future artistic and intellectual continuation. Rather than sealing up the investigation, preservation should clearly expose its own open-endedness. It should amplify participation by making preliminary order explicit.

4. CONCLUSION

While contemporaries may more directly participate in each other's oeuvres, absence requires methods of translation many of which remain to be developed. This paper has presented the initial groundwork toward the digital Barlow Archive (dBA) as a case study in archiving computer music as a translational and participatory practice. By examining taxonomic strategies and an object/event framework tailored to software-based compositional processes, it highlights how archival decisions shape access, interpretation, and reuse of digital musical materials. Drawing on the concept of amplification of intelligence, we aim to employ digital means to amplify the participatory character of research and musical thinking.
In the medium term, the objective is to transfer Barlow's entire digital legacy into a fully developed archival structure, making it accessible for both research and artistic

practice. Given that these materials embody compositional thought and process, the dBA is conceived not only as a repository but as a generative resource — one that may catalyze new compositional methodologies and the further development of existing ideas. Rather than proposing a fixed model, this contribution invites discussion about adaptive and community-oriented approaches to archiving computer music as an evolving cultural practice. The structural development of the dBA, including its evolving taxonomies and cataloging practices, remains ongoing; the accompanying presentation will discuss this work in progress in greater detail, offering a first step toward community-oriented archival standards for computer music.

Acknowledgments

We gratefully acknowledge the executor and heir of Clarence Barlow's estate for the access to the material, as well as the support and cooperation.

5. REFERENCES

[1] J. Ippolito and R. Rinehart: *Re-Collection : Art, New Media, and Social Memory* (= Leonardo), Cambridge, Mass.: MIT Press 2014.

[2] G. Boutard, "A Digital Archives Framework for the Preservation of Artistic Works," *International Journal of Digital Curatio*n, vol. 8, no. 1, 2013.

[3] W. R. Ashby, *An Introduction to Cybernetics*, Chapman & Hall, London, 1956.

[4] G. Boutard, "Computer Music as Born-Digital Heritage," *Proc. Int. Computer Music Conf. (ICMC)*, 2016.

[5] M. Akkermann, "Systematising (at) the present for the future: Reflections on archiving electroacoustic music," in *Archival Notes: Sources and Research from the Institute of Music*, G. Borio, Ed., vol. 6, pp. 77–85, 2021.

[6] L. Zattra, "The Electroacoustic Music Archives at the Fondazione Giorgio Cini: A Review of the Camillo Togni, Fausto Romitelli and Giacomo Manzoni Collections," *Archival Notes: Sources and Research from the Institute of Music*, vol. 6, pp. 87–99, 2021.

[7] B. Lee, "Issues Surrounding the Preservation of Digital Music Documents," *Archivaria*, vol. 50, pp. 193–204, 2006.

[8] S. Waters, "Making the Archive and Archiving the Making: Insights and Outcomes from a Major Research Project," *Organised Sound: An International Journal of Music Technology*, vol. 11, no. 2, 2006.

[9] C. Barlow, "Algorithmic Composition, illustrated by my own work", *The Journal of the Korean Electro-Acoustic Music Society*, pp.7-15, 2011.

[10] K. Dahan, "Musings on Computer Music Perennity," *Array: The Journal of the ICMA*, pp. 31–35, Jan. 2020.

Pd++: A C++ Library of Pure Data's DSP Objects

Robert Esler
Lead Developer, urbanSTEW and Sound Design
Faculty, Glendale Community College, AZ
robert.esler@gccaz.edu

ABSTRACT

Pd++ is a real-time C++ audio synthesis library that implements Pure Data's DSP (digital signal processing) objects as C++ classes, making it usable with object-oriented programming languages like C++, Java, or C#. The library has been designed to follow similar logic and naming conventions of Pure Data. It includes bindings for Java which allows the library to work with the Processing development environment and C# providing a native code interface to the Unity game engine. Pd++ has also been extensively tested on all major operating systems including iOS and Android, single board CPUs like the Raspberry Pi, as well as C++ based Application Programming Interfaces (APIs) such as Unreal Engine, Wwise, JUCE and FMOD. In this article the author presents how the library works in design, practice and philosophy, its perceived workflow as a design and educational tool, as well as future developments for Pd++.

1. INTRODUCTION

Pd++ is an open-source C++ library based on the audio signal processing objects from Pure Data. It was originally developed from 2012 through 2014 initially to create real-time, duplex audio on the very first Raspberry Pi (model B) chips. Pd++ is purely code, no patching, UI, or patch compilation. In fact, the patching concept only remains in design. We started by experimenting with a few classes: an oscillator, noise generator and a few filters but the project evolved to upwards of forty objects from Pure Data. It has been tested and implemented in numerous projects such as plugins, games, interactive artworks and mobile applications. Currently, Pd++ is maintained by the US non-profit urbanSTEW.

Over the past several years, we created bindings for Java and C# which allows the library options to extend several other creative coding environments such as *Processing* [1], *Open Frameworks* [2], *Cinder* [3], *Max* [4] and *Pure Data* [5] as well as game engine platforms *Unreal Engine* [6], *Unity* [7], *Wwise* [8] or *FMOD* [9]. Essentially, any framework or environment that uses a C-based language, or interface, can work with Pd++. This capability has also provided an opportunity for education, especially in the areas of audio programming for games, plugins and interactive media. In addition to its versatility, one of our core missions with this project is to create a common vocabulary within audio programming, reigniting interest in audio programming through traditional text-based coding.

Copyright: © 2026 Robert Esler. This is an open-access article distributed under the terms of ***the Creative Commons Attribution License 3.0 Unported****, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.*

2. PD++ DESIGN PRINCIPLES

2.1 Design Concept

The design of the Pd++ library was meant to emulate the same logical flow and naming conventions as Pure Data whenever possible. The majority of the signal processing objects in Pure Data, known as tilde (~) objects, are wrapped into C++ classes, with a few exceptions. For example, it was not necessary to wrap classes around the signal level arithmetic or logical operators (e.g., `[+~]`, `[>~]`, etc.), nor other common logical behaviors such as control objects like `[moses]`, `[sel]`, `[route]` or `[spigot]`, as these are more easily implemented through common decision-based statements like *if/else* or *switch*.

However, we chose to include a few non-tilde (e.g., control) objects such as common unit converters like `[mtof]` (MIDI to frequency) or `[rmstodb]` (RMS to dB) as well as static system variables like the sample rate, block size or the FFT window size eliminating objects like `[samplerate~]` and `[block~]`. These are included as member functions and static variables of a super class called `PdMaster`. All Pd++ objects inherit this class and can access these variables and functions.

Time-based objects like `[metro]`, `[line]` and `[timer]` use the audio clock instead of Pure Data's time scheduler (`t_clock`). This was in part to keep all tasks on the same thread, but also because there are numerous ways to schedule time-based processes in object-oriented languages. Different platforms may provide access to a scheduler through a variety of interfaces. For example, many graphics engines have built-in render loops that may use a variable frame rate. Therefore, Pd++'s time-based classes, `Line`, `Metro` and `Timer` use a sample counter to calculate time based on the project sample rate.

2.2 From Pure Data to Pd++

The Pure Data software is programmed in C, and though its source code is not object-oriented, it does utilize a design pattern similar to object-oriented languages. Each Pure Data object has an abstract class constructor (`class_new()`), a de-constructor (`class_free()`) as well as a data space (C-struct) where all data types, pointers, or data structures are defined for each object. Though there is no explicit polymorphism, Pure Data's code is an "object-like" C API.

This was helpful when designing Pd++. All Pure Data's signal objects are wrapped into C++ classes (e.g. `[noise~]` equals the `Noise` class). Each class accesses

the input and output buffers directly. In the Pure Data source, every DSP object contains a `class_perform()` function where all audio data is written to the audio buffer. Here is an example from the `[noise~]` object's source code:

```
static t_int *noise_perform(t_int *w) {
  t_sample *out = (t_sample *)(w[1]);
  int *vp = (int *)(w[2]);
  int n = (int)(w[3]);
  int val = *vp;
  while (n--) { //← line 6, blocking loop
    *out++ = ((t_sample)((val & 0x7fffffff) -
    0x40000000)) * (t_sample)(1.0 / 0x40000000);
    val = val * 435898247 + 382842987;
  }
  *vp = val;
  return (w+4);
}
```

Figure 1. Pure Data's `noise_perform()`. [10]

```
double Noise::perform() {
  int *vp = &x_val;
  int val = *vp;
  double output = ((float)((val & 0x7fffffff) -
        0x40000000))*(float)(1.0/ 0x40000000);
  val = val * 435898247 + 382842987;
  *vp = val;
  return output;
}
```

Figure 2. Pd++ `Noise::perform()` in C++. [11]

For the sake of legacy and readability, we chose the same name (`perform()`) for our audio process function. As one can ascertain in **Figure 1** and **Figure 2**, the DSP code between Pure Data and Pd++ is quite similar, with the notable exception of the blocking loop (e.g., line 6: `while(n--)` in **Figure 1**). This was removed from Pd++ as all blocking is handled by the audio I/O client, in which case Pd++ uses *PortAudio*. [12]

The Pd++ library emulates Pure Data's graphical conventions by allocating all signal level inlets, or the main inlet, as arguments to each class's `perform()` function. For example, the following is a Pure Data patch and the same process using Pd++:

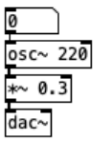


Figure 3. Pure Data patch, plays an oscillator at 220Hz.

```
Oscillator osc;
double freq = 220;
double perform() {
  return osc.perform(freq) * .3;
}
```

Figure 4. Play an oscillator at 220Hz in Pd++. This code is simplified for display purposes.

Outlets are represented as the return value of a class's `perform()` function. For example, filtering white noise in Pure Data:

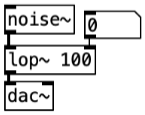


Figure 5. Low-pass filtered white noise in Pure Data.

```
Noise noise;
LowPass lop;
double cutoff = 100;
double perform()
{
  lop.setCutoff(cutoff);
  return lop.perform( noise.perform() );
}
```

Figure 6. Low-pass filtered white noise in Pd++.

Any inlets, other than the main inlet, that are control variables, such as the cutoff frequency for `[lop~]`, use getter/setter functions. This was in order to safeguard any data passing from a separate thread, like a GUI, or MIDI stream. It also allows data to be updated outside the audio thread depending on the developer's desired implementation. Similar to other software development environments, threading is generally the responsibility of the developer using Pd++. We will go deeper into how the issues of threading, audio I\O and MIDI are handled in section 3 of this article.

2.3 Audio File Management

In most cases, all classes in Pd++ adapt the same code as the Pure Data source code but without specific Pure Data types (`t_int`, `t_float`), data structures (`t_class*`) and the DSP chain (`dsp_add()`). The only Pd++ classes that did not borrow the original source code were those that involve reading and writing sound files. The `[soundfiler]` source code in Pure Data is heavily reliant on pointers to a patcher's graphical objects, such as tables or arrays, to read or write audio data. Because of this, adapting `[soundfiler]` into a Pd++ class was not a reasonable option. Instead, we chose to use parts of *The Synthesis ToolKit (STK)* [12], which is open-source and written in C++. From *STK* we adapted the file read/write classes (`FileWrite`, `FileRead`, etc.) as the backend to our `SoundFiler`, `ReadSoundFile` and `WriteSoundFile` classes. Ultimately, this approach proved much simpler to implement and yielded the same result.

2.4 MIDI Handling and Control Messaging

Parsing MIDI input, envelope scheduling or control messages are also relatively straight forward in Pd++. Here we have an example of a simple Pure Data monophonic synthesizer:

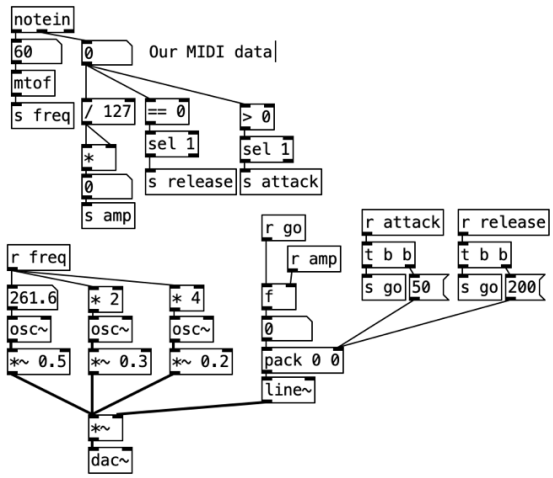


Figure 7. A Pure Data monophonic, 3-oscillator, synthesizer.

The Pd++ equivalent is below, the `runAlgorithm()` function is ultimately called by the audio client, with the variables, in this example, `outputL` and `outputR` being the actual values to be written to the audio buffer, similar to `[dac~]`:

```
void runAlgorithm(double in1, double in2) {
    notes = midi.noteIn();
    double freq = this.mtof(notes.note);
    amp = (double)notes.velocity/127.0f;
    amp *= amp;

    if(notes.velocity == 0)
        setBang(false);
    else
        setBang(true);

    outputL = outputR = (osc1.perform(freq)* .5 +
                         osc2.perform(freq * 2)*.3 +
                         osc3.perform(freq * 4)*.2)
                          * env;
    if(getBang())
        env = line.perform(amp, attack);
     else
        env = line.perform(amp, release);
  }
```

Figure 8. Pd++ version of the same monophonic synthesizer. MIDI is implemented in a separate class. Example from Processing (see [1] and [15]).

There is no MIDI client actually built into Pd++, as this is usually handled by the operating system or host API. In the example above the MIDI stream is wrapped into a separate class.

In designing the Pd++ library we wanted to ensure that anything made in Pure Data sounded the same if it were programmed in Pd++. This meant negotiating how to extract the code from Pure Data's source and implement it in an object-oriented language while maintaining a connection to the original patcher ideology. Though most concepts like inlets, outlets, abstractions, etc. are represented in some fashion within the limits of structured programming, there are still aspects of the Pd++ design that diverge from the Pure Data software.

3. PD++ IN PRACTICE

There are two notable Pure Data object families missing from Pd++. These are the audio I/O objects `[dac~]` and `[adc~]` and the MIDI family of objects, `[notein]`, `[ctlin]`, etc. Every operating system, or platform's API, handles accessing the audio or MIDI hardware differently, which makes audio programming at times difficult. There are several third-party options to handle some of these issues but few that can handle all the various architectures and operating systems.

This is a huge barrier to overcome as a developer because it depends entirely on how someone may want to use Pd++. However, since Pd++ is C++, one can interface with any C-based code. This includes almost all industry level audio APIs like *JUCE*, *AU/VST*, *Unreal Engine* or *Unity*. In those instances, each API has their own methods and processes for accessing the input and output audio buffers and MIDI stream. However, if you want to use Pd++ to develop an application for a specific operating system then it would be necessary to either create your own audio and MIDI clients or use a third-party's source. This can complicate things such as open-source licensing and automated build tools.

3.1 Pd++ for Processing (Java)

To simply these issues, in 2021, we created Java bindings using the Java Native Interface (JNI) [14] so that Pd++ could be used with the *Processing* development environment, the library is called *Pd4P3* or (Pd++ for Processing 3). [15] For audio input and output, *Pd4P3* provides the choice of two native audio I/O clients, one that uses *PortAudio* through its experimental *jportaudio* package [16] and one that uses the Java Sound package. [17] The *PortAudio* interface allows for more flexibility with multichannel audio, whereas Java Sound only supports two channel, duplex audio. *Pd4P3* also provides examples of how to interface with the MIDI protocol through the *Java Sound* package.

Processing is a simple integrated Java development environment for artists that embeds packages for graphics, networking, sound and animation. Like Pure Data, it has a huge community of support for learning, education and development. Additionally, *Processing's* syntax is simplified for beginners. For example, to print in Java:

```
System.out.println("Hello World");
```

In Processing:

```
println("Hello World");
```

Processing also provides different development modes such as for Android devices. As of 2022, *Pd4P3* additionally provides support for Android audio and MIDI through several provided examples. We will go into more detail on this in section 5 of this article.

3.2 Pd++ for Unity (C#)

In 2023, we created C# bindings for Pd++ that can be used with the Unity game engine, this library is known as *Pd4Unity*. [18] The C# library utilizes the *P/Invoke* interface, similar to the JNI for Java. [19] In *Unity's* scripting API, a developer can inherit the method `OnAudioFilterRead()` through its parent class `MonoBehavior` which provides access to the DSP chain on a separate thread for all parent and child game objects. Though simple in implementation, this has proven to be limited in its capabilities because `MonoBehavior` also internally schedules garbage collection which can cause occasional audio dropouts or other undefined behavior.

As of *Unity* version 6000.3.x, there is a new interface called the *Scriptable Audio Pipeline*, which allows for developers to write audio routines to an optimized signal chain free from garbage collection. [20] The tests so far show a marked improvement of the performance of Pd++ and overall quality. The only drawback is the C# interface for the new pipeline is much more complex than the original `OnAudioFilterRead()` method.

Pd++ has also been tested in other audio industry specific software environments such as Unreal Engine's *MetaSounds*, the *JUCE* Framework for audio plugin development, and the *Wwise* and *FMOD* audio middleware platforms. All of these packages utilize C++ for their API backends. In these tests we used the same C++ class for each platform and wrapped our `perform()` function within the APIs audio process.

4. THE PD++ PHILOSOPHY

The overarching goal of the Pd++ project is to provide a cross-platform, common vocabulary for sound and music in a traditional coding environment. We want to remain independent from the Pure Data software, but continue to inspire the same exploration and creativity. With that, education is also a part of our mission. Visual programming languages (VPLs) like Pure Data and Max have been part of educational programs for decades, thus it seemed logical to continue this legacy but adapted towards principles of structured programming. We aim to inspire a revival of audio programming through object-oriented languages in traditional educational settings and creative workflows.

In response to this mission, Pd++ conforms to a certain philosophy:

1. *A Common Vocabulary:* Pd++'s vocabulary, examples and design principles should remain consistent with the Pure Data software ecosystem.
2. *Simplicity:* Pd++ should be simple, flexible and extensible.
3. *Open Access:* Pd++ should always be open source.
4. *Creativity First:* Pd++ should inspire creativity by providing only the essential audio building blocks.

We want to support creativity, and encourage potential users the value of programming in a text-based language. All implementations of Pd++, including *Pd4P3* and *Pd4Unity* have plentiful examples with detailed documentation. Many of these examples are novel to the distribution while others are adapted from Miller Puckette's original Pure Data examples, or emulations of patches from Andy Farnell's book "Designing Sound". [21]

5. THE PD++ WORKFLOW

5.1 Background

When Pd++ was originally conceived in 2012, libraries such as *libpd* [22] or compilers like *hvcc*[1] [23] were not yet common resources for developers and composers. The initial goal was to have a resource that could enhance our design and development workflow, such that anything that we may be prototyping in Pure Data could be easily translated into C++ code later. We needed a framework for audio synthesis and music composition that could work on a variety of different devices and modalities. This include such areas as: single board CPUs, Android devices, large datasets, C++ APIs for plugins and game engines. At the time, this was not possible any other way than through C/C++ and Java.

By 2014 we had an alpha version of the fully developed library and for several years we used it internally for various commissions and projects. Though entirely experimental at first, we realized that we had a workflow tool that could utilize the strengths of visual programming and text-based programming without sacrificing large-scale code refactoring.

5.2 Utilizing VPLs and Text-Based Programming

This type of workflow translates both to the development and learning of audio programming. Several recent studies have shown those learning a text-based language side-by-side with a VPL have better rates of comprehension. [24,25] Text-based programming's strengths are also well documented, especially when scaling, debugging, using recursion and logic programming. [26,27] However, VPLs like Pure Data have clear benefits to the learning and the creative process too, such as:

1. Rapid iteration
2. Fewer syntax errors
3. Easier access to hardware (MIDI, audio, networking, etc.)
4. Interpreted at runtime, no compiling needed.
5. Built-in GUI objects

With this in mind, we can rapidly prototype various audio processes or experimental music systems early in the design process, and quickly translate these into native code, thus generating a minimum viable product for our target host environment. This also allows one to leverage other low-level aspects such as direct access to API level

[1] *hvcc* is also known as the Heavy Compiler Collection.

features, memory management, optimization, scalability and clearer arithmetic and logical flow. Programming concepts can also be broken into smaller parts and built upon as users increase their coding facility.

5.2.1 Rapid Prototyping on Hardware

In order to understand this workflow, it is best to provide an example of how it has been used. When we developed the *Processing* Android examples, we decided to create a keyboard instrument for a tablet and/or mobile device. The instrument needed a custom touch-based UI, a piano keyboard that could scale to any size screen, control of various audio effects and the ability for the application to be used as a MIDI input to another device. Additionally, we wanted to be able extend this instrument in the future, possibly incorporating other features (e.g. motion sensors, camera, microphone, environmental sensors, etc.) or audio processes at a later time.

For the sake of this article, we will not discuss in detail how the graphics were created in *Processing*. Though simply put, it consists of drawing various colored rectangles. Each having a specific function when touched within their defined boundary. See the UI example below:

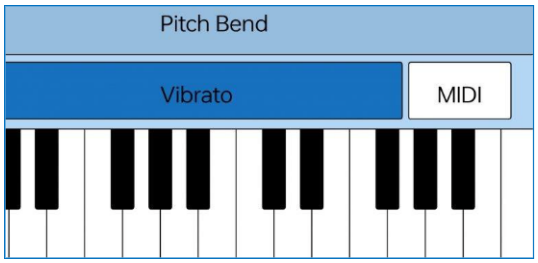


Figure 9. The *Pd4P3* Android Piano Keyboard example in Processing.

To begin however, we did not need the UI, instead we focused on prototyping our synthesizer, testing various sounds and effects. For that we used our VPL, Pure Data:

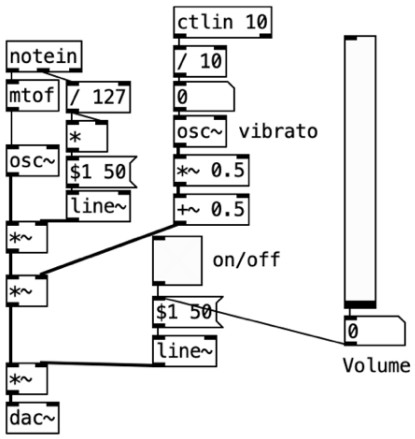
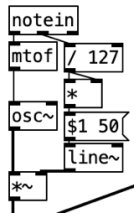


Figure 10. A simple monophonic MIDI synthesizer.

For clarity, we will not include every detail of this particular example. Instead, we will focus on how we typically work between Pure Data and Pd++ with more simplified patches and code examples.

Once we settled on a sound design concept, we began to translate each part into code, in this case using the *Pd4P3* library. We started with the basic sound generator:



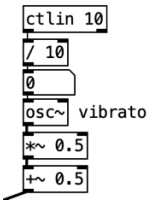

```
Oscillator osc = new Oscillator();
Line line = new Line();

double perform(double note, int vel) {
    double env = vel/127;
    return osc.perform(osc.mtof(note)) *
            line.perform(env*env, 50);
}
```

Figure 11. Our sound generator, patch and code (Java).

Note that in the particular example in Figure 11, the oscillator `[osc~]` could be any sound generator, or even an abstraction or sub-routine, with corresponding input and output types.

Next, we created our vibrato. For that we only need a few more lines of code:

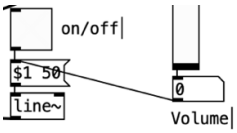



```
Oscillator osc = new Oscillator();
Oscillator oscVib = new Oscillator();
Line line = new Line();

double perform(double note, int vel, double
ctlVal) {
    double env = vel/127;
    double vib = (oscVib.perform(ctlVal/10)
                  * .5) + .5;
    return osc.perform(osc.mtof(note)) *
          line.perform(env*env, 50) * vib;
}
```

Figure 12. Adding the vibrato, patch and code (Java).

Then, we add the on/off and volume controls. The volume and on/off values are wrapped into getter/setter methods (`getOn()`, `getVol()`):




```
Oscillator osc = new Oscillator();
Oscillator oscVib = new Oscillator();
Line line = new Line();
Line lineVol = new Line();
int on = 0;
double vol = 0;

double perform(double note, int vel, double ctlVal) {
    double env = vel/127;
    double vib = (oscVib.perform(ctlVal/10) * .5) + .5;
    return osc.perform(osc.mtof(note)) *
           line.perform(env*env, 50) * vib *
           getOn() * lineVol.perfom(getVol(), 50);
}
```

Figure 13. Our complete synthesizer. Note, not all of the code is represented here; this method would need to be wrapped inside of a class. (See [28] for source)

We still have to free our memory from the dynamic library[2], since we are using *Processing*, so we add the method:

```
void free() {
    Oscillator.free(osc);
    Oscillator.free(oscVib);
    Line.free(line);
    Line.free(lineVol);
}
```

Figure 14. Free our memory

Our MIDI handler is contained in a separate class. The MIDI client comes from the *JavaSound* package, which can be used as-is within *Processing* and generally behaves similar to most other MIDI libraries.

5.2.2 Extensibility

With the scaffolding of a basic synthesizer our code now becomes easily extensible. For example, we can implement this synthesizer to be polyphonic with as many voices as we need without refactoring our synthesizer code. We can create an array of synthesizers, an array of notes and an array of velocities:

```
for (int i = 0; i < maxVoices; i++)
 output += synths[i].perform(notes[i], vels[i]*m, getVib());
```

Figure 15. Handling polyphony, the variable '`m`' mixes the voices depending on `maxVoices`.

In the complete example, there are six classes implemented (see [28] for source).

1. MIDI Handler (contains our MIDI client and `[notein]` and `[ctlin]` object emulation.)
2. Synthesizer (similar to an abstraction in Pure Data)
3. Main Audio Processor (like a patch, handles all audio I/O and control. This is called a sketch in *Processing*.)
4. Polyphonic Voice Handling (an emulation of `[poly]` polyphonic voice allocator)
5. Piano Keyboard and UI (our custom graphics)
6. Key Handler (schedules note/velocity pairs based on the key pressed)

More features can be added later, like audio effects. One way to add an effects chain:

```
for(int i = 0; i < maxVoices; i++)
    out += synths[i].perform(notes[i], vels[i]*m, getVib());

out += effect1.perform(out) + effect2.perform(out);
```

Figure 16. An example of an effects chain.

Each effect is tested first in Pure Data, then implemented as a separated class using the same workflow as previously presented. Other controls could be added via the device's sensors using the same method:

```
for(int i = 0; i < maxVoices; i++)
    out += synths[i].perform(notes[i], vels[i]*m, getVib());

out += effect1.perform(out) * getAccelX() +
       effect2.perform(out) * getAccelY();
```

Figure 17. Adding sensor data, `getAccelX()` and `getAccelY()` come from the device's accelerometers.

This example shows one way such a workflow can allow for a rapid paced design or a learning environment while implementing and debugging a prototype on a hardware device. In our development process we leveraged the strengths of the VPL in the exploratory stages and text-based code in the application development stages to create a viable product for our target hardware that uses that same vocabulary and logic across each paradigm.

6. FUTURE ENHANCEMENTS

As mentioned previously, extensibility is an essential element of the Pd++ philosophy. With the growing popularity of the Python scripting language, we want to explore options for a Pd++ interface to Python. There were some initial tests in the last year using Python's *ctypes* [29] library, which can access C-style functions from a dynamic library. However, these tests were mostly successful at getting data in and out of the library but not in creating any audio output. More time would be needed to fully explore how to best use Pd++ with an interpreted language like Python. There will be some difficult choices on how to implement the audio I/O client for Python or whether there are reasonable options for real-time audio based on the current design of the Pd++ library.

Another area of exploration has been with *WebAssembly* [30], as a method of running Pd++ in the modern browser. Several years ago, we tried some very basic tests with the Pd++ oscillator. Though they were successful (we could hear audio coming from the browser), but there was a large latency (~100 ms) and working with the *emscripten* [31] compiler was quite difficult. However, having a web interface to practice writing or prototyping audio processes in Pd++ could be a useful addition to the workflow and a valuable extension for a variety of applications.

Finally, a more experimental idea we are exploring is a scriptable audio plugin using Pd++. Similar to *ReaScript* in REAPER [32], Blue Cat's *Plug'n Script* [33], *Protoplug* [34], or the scriptable MIDI plugins in Logic Pro X [35], this would be a text-based interface where one could work with the various inputs (audio or MIDI) and create custom filters, synthesizers or other audio processing inline to their plugin DSP chain.

Though there are numerous hurdles that arise with such an endeavor. Should we use an interpreted language? If so which one? How do you stay true to the Pure Data signal flow paradigm? How would transpiling work? Regardless of these possible limitations, it would be a practical

[2] The JNI is in C and does not handle garbage collection, this has to be done manually for each object.

approach to both testing and teaching audio programming basics and signal processing techniques.

7. CONCLUSIONS

Pd++ is not as fancy or as sophisticated as many larger projects related to the Pure Data landscape. However, its simplicity is its greatest asset. With the core library being written in C++ it provides numerous options to interface with a variety of creative coding tools and platforms. Pd++ performs and compiles on all major operating systems and by limiting its functionality and features to purely being code-based makes it simple and efficient.

Pd++ is a paradigm shift and meant to inspire a revival of text-based coding techniques amongst students, artists and composers. It supports this vision through its technical implementation as well as its design, theory and practice. We aim to encourage innovation, enhance audio programming workflows and open new opportunities for creativity and design.

Acknowledgements

The Pd++ project is supported and maintained by urbanSTEW an arts and technology, 501(c)3, non-profit organization based in the United States.

REFERENCES

[1] B. Fry and C. Reas, "Processing: A flexible software sketchbook and language for learning how to code," Available: https://processing.org/. [Accessed: Dec. 5, 2025].

[2] OpenFrameworks Community, "openFrameworks: An open-source C++ toolkit for creative coding," Available: https://openframeworks.cc/. [Accessed: Dec. 5, 2025].

[3] A. C. R. Cinder Project, "Cinder: A community-developed, free and open-source library for professional-quality creative coding in C++," Available: https://libcinder.org/. [Accessed: Dec. 5, 2025].

[4] Cycling '74, "Max: Visual programming language for music and multimedia," Available: https://cycling74.com/products/max. [Accessed: Dec. 5, 2025].

[5] M. Puckette, "Pure Data: Open-source visual programming language for multimedia," Available: https://puredata.info/. [Accessed: Dec. 5, 2025].

[6] Epic Games, "Unreal Engine: Blueprint Visual Scripting and game development platform," Available: https://www.unrealengine.com/. [Accessed: Dec. 5, 2025].

[7] Unity Technologies, "Unity: Game development platform," Available: https://unity.com/. [Accessed: Dec. 5, 2025].

[8] Audiokinetic, "Wwise: Cross-platform sound engine for interactive media," Available: https://www.audiokinetic.com/products/wwise/. [Accessed: Dec. 5, 2025].

[9] Firelight Technologies, "FMOD: Sound effects engine and audio middleware," Available: https://fmod.com/. [Accessed: Dec. 5, 2025].

[10] M. Puckette, "Pure Data [Source code]," GitHub. Available: https://github.com/pure-data/pure-data. [Accessed: Dec. 6, 2025].

[11] R. Esler, "Pd++ [Source code]," Bitbucket. Available: https://bitbucket.org/resler/pd. [Accessed: Dec. 6, 2025].

[12] R. Bencina and P. Burk, "PortAudio – Portable Audio I/O Library," Version 19. Available: http://www.portaudio.com/. [Accessed: Dec. 6, 2025].

[13] P. R. Cook and G. Scavone, "Synthesis Tool Kit (STK) [Software library]," Available: https://ccrma.stanford.edu/software/stk/. [Accessed: Dec. 6, 2025].

[14] Oracle, "Java Native Interface Specification Contents," Java SE 8 Documentation. Available: https://docs.oracle.com/javase/8/docs/technotes/guides/jni/spec/jniTOC.html. [Accessed: Dec. 6, 2025].

[15] R. Esler, "Pd4P3: A real-time audio synthesis library for Processing," GitHub repository. Available: https://github.com/robertesler/Pd4P3. [Accessed: Dec. 6, 2025].

[16] P. Burk, "JPortAudio [Software library]," Available: https://github.com/philburk/portaudio-java. [Accessed: Dec. 9, 2025].

[17] Oracle, "javax.sound (Java SE 25 & JDK 25) – Package Summary," Oracle Documentation, Dec. 2025. Available: https://docs.oracle.com/en/java/javase/25/docs/api/java.desktop/javax/sound/package-summary.html. [Accessed: Dec. 6, 2025].

[18] R. Esler, "Pd4Unity: Pd++ for Unity – C# library interfacing Pure Data signal processing with Unity's API," GitHub repository. Available: https://github.com/robertesler/Pd4Unity. [Accessed: Dec. 6, 2025].

[19] Microsoft, "Native library loading – .NET," Microsoft Learn Documentation. Available: https://learn.microsoft.com/en-us/dotnet/standard/native-interop/native-library-loading. [Accessed: Dec. 6, 2025].

[20] Unity Technologies. (2025). *Scriptable Audio Pipeline (Unity 6.3.x)* [Software Documentation].

https://docs.unity3d.com/6000.3/Documentation/Manual/audio-scriptable-processors.html [Accessed: Dec. 9, 2025].

[21] A. Farnell, Designing Sound. Cambridge, MA: MIT Press, 2010.

[22] libpd Project, “libpd: Pure Data embeddable audio synthesis library,” Available: https://github.com/libpd/libpd. [Accessed: Dec. 5, 2025].

[23] Wasted Audio, “Heavy Compiler Collection, a python-based dataflow audio programming language compiler …” Available: https://github.com/Wasted-Audio/hvcc. [Accessed: Dec. 9, 2025].

[24] D. Weintrop, “Minding the gap between blocks-based and text-based programming: Evaluating introductory programming tools,” in Proc. ACM SIGCSE Student Research Competition, 2015, pp. 29–30, doi: 10.1145/2676723.2677305.

[25] T. Tóth and G. Lovászová, “Visual vs. textual programming: A case study on mobile application programming by teenagers,” AD ALTA: J. Interdiscip. Res., vol. 11, no. 1, pp. 337–347, 2021, doi: 10.33543/1101337347.

[26] M. Noone and A. Mooney, “Visual and textual programming languages: A systematic review of the literature,” Journal of Computers in Education, vol. 5, no. 2, pp. 161, 2018. doi: 10.1007/s40692-018-0101-5.

[27] R. S. Ribeiro, L. O. Brandão, T. V. M. Faria, and A. A. F. Brandão, “Programming web-course analysis: how to introduce computer programming?,” in Proc. Frontiers in Education Conf. (FIE), Madrid, Spain, Oct. 2014, pp. 1, doi: 10.1109/FIE.2014.7044085.

[28] R. Esler, “Pd4P3: Android PianoKeyboard example,” GitHub, https://github.com/robertesler/Pd4P3/tree/main/examples/Android/PianoKeyboard. [Accessed: Jan. 1, 2026].

[29] Python Software Foundation, “ctypes: A foreign function library for Python,” Available: https://docs.python.org/3/library/ctypes.html. [Accessed: Dec. 5, 2025].

[30] World Wide Web Consortium (W3C), “WebAssembly: A binary instruction format for a stack-based virtual machine,” Available: https://webassembly.org/. [Accessed: Dec. 5, 2025].

[31] Mozilla, “Emscripten: An LLVM-to-WebAssembly compiler,” Available: https://emscripten.org/. [Accessed: Dec. 5, 2025].

[32] Cockos Incorporated, “ReaScript: Scripting API for REAPER DAW,” Available: https://www.reaper.fm/sdk/reascript/reascript.php. [Accessed: Dec. 5, 2025].

[33] Blue Cat Audio, “Plug’n Script: Audio and MIDI scripting plugin,” Available: https://www.bluecataudio.com/Products/Product_PlugNScript/. [Accessed: Dec. 5, 2025].

[34] J. Wilkes, “Protoplug: Lua scripted audio plugin environment,” Available: https://github.com/pac-dev/protoplug. [Accessed: Dec. 5, 2025].

[35] Apple Inc., “Logic Pro: Scriptable MIDI plugins,” Available: https://support.apple.com/guide/logicpro/use-the-script-editor-lgcecc16550d/11.2/mac/14.4. [Accessed: Dec. 9, 2025].

DSP and the Metalevel
Clamps – an integrated environment for algorithmic composition and interactive realtime performance

Orm Finnendahl
Hochschule für Musik und Darstellende Kunst Frankfurt
orm.finnendahl@selma.hfmdk-frankfurt.de

ABSTRACT

Integrating low level DSP operations and highlevel concepts for organizing musical material has been a longstanding repeated topic in the discussion of computer music. Although many capable DSP systems with advanced features concerning the organization, visualisation and transformation of musical material on a higher level are in widespread use today, they either suffer from an ongoing separation between the higher level and the DSP level or the lack of a satisfying infrastructure for the integration of both worlds.

Clamps [1] *[1] is a Common Lisp Package built on top of Incudine[2] for the DSP part, CLOG[3] for the GUI and other music related CL packages to create a unified platform, intended to combine realtime performance, algorithmic composition and notation in a single application language and memory space. It has been successfully used for a wide a range of applications from traditional compositional work and the production of Electroacoustic Music to Interactive Live-Performances.*

1. INTRODUCTION

The systems currently in widespread use for realtime audio processing, like *SuperCollider*, *Pure Data*, *Max/MSP*, *Csound*, *ChucK* or *FAUST* offer sophisticated DSP engines and mechanisms for the high level organization of musical material, add-ons for notation or an impressive number of approaches to access or even integrate different programming languages, systems or DSLs [2] for compositional work. Even though it is obvious that the basic programming paradigm of the user interfaces/DSLs of the systems either facilitate a hierarchical juxtaposition of different temporal levels with many interconnections, or a separation of the DSP part from the score, it can be argued that the origins of their DSLs focused on the organisation of the definition of DSP code, extending those techniques to different temporal levels, rather than the development of sophisticated tools for the definition and handling of musical objects and algorithmic processes on the metalevel[4]. In the case of Max/MSP or Pure Data, high level language constructs or external systems for notation, score display, etc. are normally added as plugins, using a language different from the language/programming paradigm of the core system. Although *SuperCollider*'s Sclang offers expressive high level language constructs, this comes at the price of a clear separation between the language part and the DSP engine.

In addition, the work with DSP processing in those systems is characterized by at least one of various unsurmountable barriers: [3]

- The use of precompiled code (Ugens/objects/opcodes) for low level processing tasks written in C or C++, a language different from the language/paradigm of the system itself. This requires a separate development process using editors/compilers different from the used system and require a restart of the entire system on every change, making the development and testing of new components a tedious task. [4]
- The distinction between control-rate and audio-rate processing. This is especially obvious in Pure Data and Max/MSP, as they use different low-level objects for these rates, requiring extra effort to bridge both worlds.
- The separation of a DSP server and a language client in *SuperCollider* requires efforts for their timing synchronization and the avoidance of bottlenecks concerning the transfer of data structures between client and server using TCP/IP.

In contrast, most Common Lisp based music systems, like *OpenMusic*, *Common Music*, *CLM*, *OpusModus* or *Slippery Chicken* focused on the metalevel and were initially developed for non-realtime work as a means to facilitate algorithmic composition, in many cases for instrumental work, fixed media pieces or combinations of both. [5] Many of these systems have their origins at a time, when realtime computer work on non mainframe sytems was just emerging and often used non-realtime audio systems like *Csound* for the DSP rendering work, or MIDI softsynths for immediate acoustical feedback.

[1] Clamps stands for **C**ommon **L**isp **A**ided **M**usic **P**roduction **S**ystem
[2] The acronym *DSL* is short for *Domain Specific Language*

Copyright: ©2026 Orm Finnendahl. This is an open-access article distributed under the terms of the Creative Commons Attribution License 3.0 Unported, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

[3] *FAUST* doesn't share these limitations, but currently still appears to have a strong focus on DSP work with limited infrastructure or support for high level work/notation, etc.
[4] It should be noted that the designers of those systems suppose that changes in those modules are generally not done by the users and that the development of new modules is conceptually clearly separated from *using* the system.
[5] *CLM* focuses on DSP, albeit in a non-realtime fashion.

Several additions to *OpenMusic* aimed at adding a tighter connection to acoustical exploration by adding soundfile playback modules and external processing tools[5], but the DSP processing was never done within OpenMusic and the same barriers between the high level organisation and the DSP engine exist, although in this case from the opposite perpective.

With the advent of *Incudine*, a performative environment for DSP programming entirely written in Common Lisp, the situation has changed considerably in the past decade and finally enables an approach free of any barriers between DSP processing and the language.

Clamps is built on top of Incudine and other music and GUI related applications, written in Common Lisp to create a unified platform, intended to combine algorithmic composition, notation and interactive realtime performance in a single application and language. A main advantage of using Common Lisp is the integrated immediate accessibility of all of the source code of *Clamps*, its external components and even the Common Lisp language itself down to the lowest implementation level while working with it.

All DSP routines, the scheduler and the control engine run as functions or macros within the same Lisp image and memory space resulting in a seemless experience of transparency on every level of the working process. Changes to every detail can easily be done on-the-fly, even while realtime DSP routines are running, resulting in an immediate acoustical feedback of the changes, providing the unique experience typical for most Lisp based systems.

2. SPECIAL FEATURES OF CLAMPS

2.1 SVG as central worksheet format

One of the rather unusual features of *Clamps* is the central use of SVG as a format for the representation, display and manipulation of musical data.

The SVG format has several advantages: It is open source, easy to parse, extendable and can be rendered natively by most web browsers (in the latter case it can also be styled with CSS and even manipulated with Javascript). Inkscape is available as an open source, mature and professional editor for SVG files, allowing for a sophisticated use of Layers, Groups or control over the visibility of objects, their selection, transformation, etc.

In *Clamps*, the integration of SVG is realized by extending Common Music with an SVG input/output class. Import/export methods have been implemented for MIDI, sfz and poolevt event classes (for clarification see Sections 2.4 and 2.5). In addition, the system has been specifically designed to enable the definition of custom event classes for projects which use specialized synths in electroacoustic contexts on the fly with a few lines of code (see Section 3).

The elements of *Clamps* are represented by line objects (paths) in the SVG file. When imported, their coordinates, stroke-color and opacity is supplied to *Clamps* and the import method of the event type can decide the specific mapping of the data to the event slots. For the defined event types, the x and y coordinates of the SVG elements are used to specify keynumber/transposition, time and duration of the events. The opacity is used for mapping the amplitude (Fig. 2).

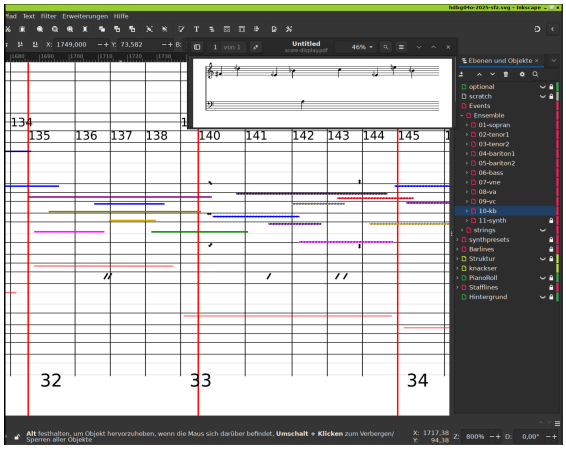


Figure 1. Example of the SVG display in Inkscape. The selected events can be played back and displayed as a score as shown in the upper part using custom Inkscape extensions.

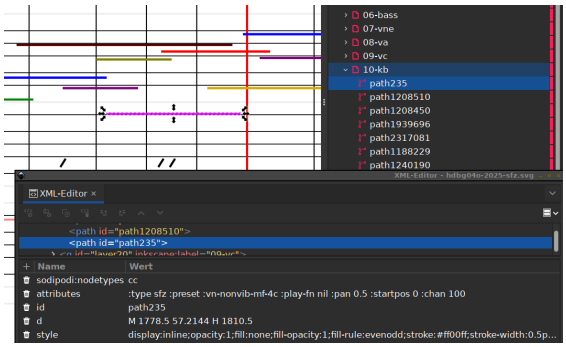


Figure 2. Detail of the same section as Fig. 1, showing the contents of the attributes slot of one selected Sfz event in Inkscape's XML editor.

Only events in the "Events" Layer of the SVG file are imported into *Clamps*. This enables other graphical elements, like timeline, bar, barnumber, staff system, pianoroll, score or annotation backgrounds to be added to the SVG and selectively displayed in other Layers without affecting the playback. The dispatch of the event type on import is realized using the :type field in an "attributes" tag in every SVG element. The attributes field can also be used to encode other custom parameters for the event type in question.

For seemless integration with *Clamps*, Inkscape has additionally been extended by plugins which play back selected elements, or show them in a notation view using the Inkscape Menu or custom keystrokes (Fig. 1).

In this sense, Inkscape can be considered as some sort of DAW for working in the framework of *Clamps*. [6]

2.2 Browser based GUI

A well known major drawback of Common Lisp is the lack of a stable connection to current GUI frameworks. OpenMusic avoids this problem by using LispWorks, which implements its own GUI toolkit at the price of being proprietary.

[6] In connection with the svg-player of *Clamps* the concept also has similarities to a Maquette in OpenMusic, although it is different in the sense that by using a widely accepted open source graphics format, it can take advantage of the editing capabilities of full fledged external graphics programs, like Inkscape.

The CLOG framework is developed since late 2020 by David Botton to solve this problem using web technology, displaying the GUI in a web browser. This has several advantages:

- The GUI and the application are separated, enabling the control of the application with external devices, like Mobile Phones and Tablets without the need to install additional software.
- Synchronising the display of multiple devices is fairly straightforward, enabling shared access to the same machine by multiple performers or displaying information to them, opening up low level opportunities for shared performances.
- The availability of the rendering engine and the program logic of the web browser can reduce the use of resources on the dsp audio engine.

Clamps uses CLOG for its GUI. For that purpose, the most common GUI widgets of audio applications, like sliders, multisliders, buttons, number boxes, radiobuttons, VUmeters and SVG displays have been added and integrated into *Clamps*. The synchronization between GUI elements is managed using ref-objects, compute and watch functions with a declarative syntax, similar to JavaScript frameworks like React or Vue.

Custom synchronized GUIs across devices can easily be developed (see Fig. 3. Infrastructure and GUIS for NanoKontrol2, FaderFox and Asparion MIDI/OSC and motion controllers of Mobile Phones are available and can be used as blueprints for other types of MIDI Controllers.

VU Meters of the Audio in/outputs and a SVG and ATS Player are built into *Clamps* and accessible on startup.

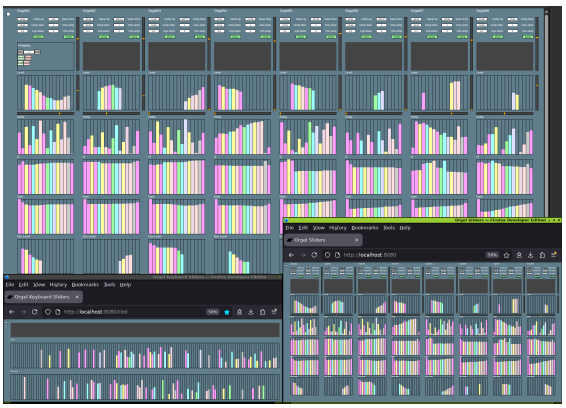


Figure 3. Custom GUI for the control of 728 parameters of a custom electroacoustic instrument based on acoustical feedback. The bottom left window shows another ordering of sliders. Note that all browsers share the synchronized parameter state at all times, enabling collective performances with multiple performers.

2.3 Interactive realtime FFT exploration

A unique feature of *Clamps* is the interactive realtime FFT explorer. Similar to SPEAR[6], any soundfile can be analyzed and its peaks organized into individual tracks using ATS, developed at CCRMA by Juan Pampin in the late 1990s[7]. In contrast to Spear, not only the sinusoidal components, but also the residual signal is analyzed using the residual energy in the frequency ranges of the Bark scale. This information can be used in the resynthesis using colored noise.

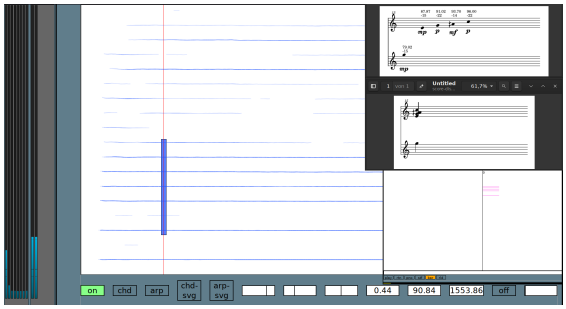


Figure 4. Example of the interactive realtime FFT explorer of *Clamps*: The position of the vertical blue rectangle in the FFT display is controlled by the mouse and specifies the region for playback, with its opacity defining the amplitude threshold for played or exported events. On the left are VU meters of the system. On the right are different notation displays and the svg player showing the exported events of the selected area, accessible with the buttons below the fft display.

Custom synths built for *Clamps* enable realtime controlled resynthesis of the analysis data with independent frequency and amplitude modulation for each individual partial track.

A GUI displays the traditional view of the sinusoidal components of the FFT analysis in the browser. The mouse cursor can be used to traverse the FFT display in the frequency domain while listening to the resynthesis of the sound at the cursor location. In addition, the mousewheel can be used to restrict the frequency range and a contrast slider can adjust the opacity of the selected frequency range, defining an amplitude threshold below which partials don't get synthesized.

This enables an intuitive in depth exploration of the analyzed sound in realtime. Any sound generated with these controls can be rendered into a score notation display, displaying either a chord or the pitches of the chord as an arpeggio with annotated pitches in MidiCent and amplitude in dB. In addition the chord can be exported into a SVG file for playpack or for further exploration and manipulation using Inkscape (see Fig. 4).

2.4 SFZ support

SFZ is a soundfont format supported natively with internal sample players defined in *Clamps*. Apart from avoiding the necessity of external MIDI software synths (although their inclusion is fully supported), the main reason for its implementation are the limitations of MIDI when dealing with microtonal polyphony. Other systems, like OpenMusic try to circumvent them by using MIDI channel tuning with multiple MIDI channels, but this hits its limits very quickly, if many sounds within a small interval have to be played simultaneously, as it is frequent for higher partials in spectral music. Using sfz events, the number of simultaneous mictrotonal voices is only limited by the CPU speed, currently well beyond 500 on a standard laptop (see Fig. 5).

For additive synthesis, a sfz with a sine wave table is available.

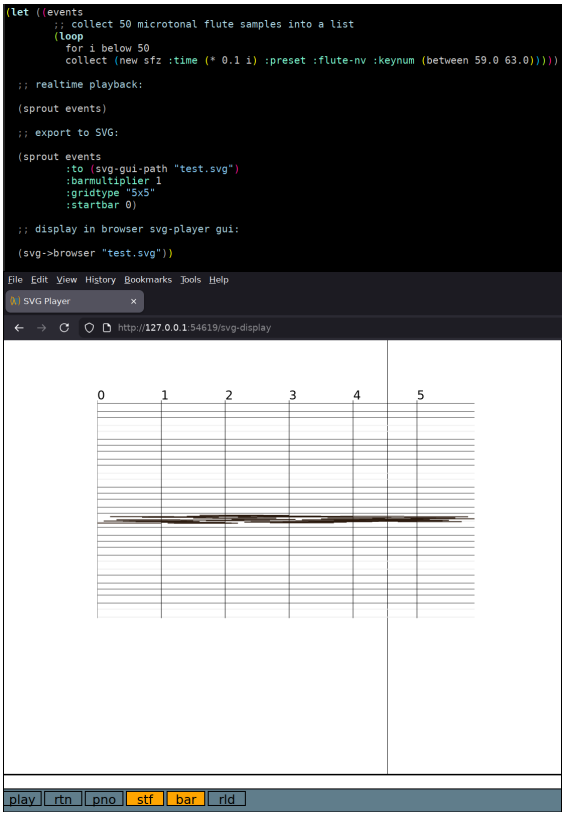


Figure 5. Code example for realtime playback, SVG export and its visualisation of sfz events, using 50 microtonal flute samples. Below the code is the web browser display of the interactive svg-player.

2.5 Poolplayer event engine

Poolevents are another event type added to Common Music. This event type uses a sample based granular synth for realtime output with parameters for the used sample, keynum, amplitude, stretch, attack and release time, start and end positions of the buffer to play, the window width of the grains, the pan setting and the output channels to use (see Fig. 6).

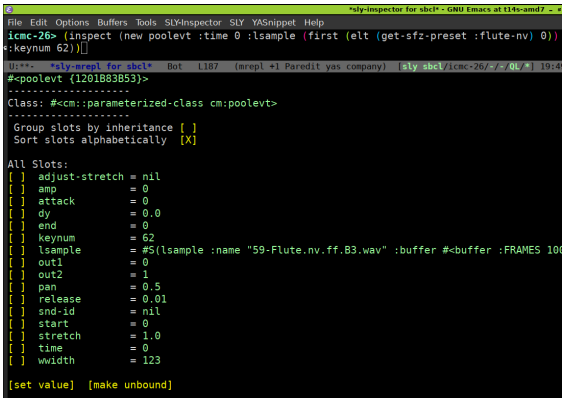


Figure 6. Inspecting an instance of the poolevt class.

For playback of poolevents, a domain specific language (DSL) can be used. The code for a defined preset is displayed in a dedicated emacs textbuffer window. A preset can be thought of like an instrument definition in csound or a synthdef in SuperCollider, but working on the scheduling level of events rather than on the sample level. For each event, all parameters of a poolevt to be played are calculated using a dedicated function for each of the parameters. An additional dtime function calculates the delta time for the next event (see figure).

```
(digest-form-to-preset
 0
 (:bindings ()
  :dtimefn 0.1
  :lsamplefn (elt *pool0* 0)
  :ampfn -6
  :transpfn 0
  :startfn 0
  :endfn 0
  :stretchfn 1
  :wwidthfn 1
  :attackfn 0
  :panfn (random 1.0)
  :releasefn 0.01
  :playfn #'play-buffer-stretch-env-pan-out*
  :outfn (stereo-out)))
```

Figure 7. Example of a preset with default forms for the calculation of the parameters of a sequence of poolplayer events. *digest-form-to-preset* is a macro which compiles these definitions into functions with *x*, *idx*, *dtime* and other, optional arguments as parameters and stores them in arrays to be recalled on the fly in a performance situation.

The body forms of the function definitions are displayed in the preset buffer next to a keyword indicating the name of the parameter (see Fig. 7).

The playback of a preset can be started with a total duration. In this case, all parameter functions will be called within a lexical context, with the bound symbols *x*, indicating the normalized total time into the total duration of the preset, *idx*, an index count of the current event, *dtime*, the time until the next event is scheduled and optional bindings bound at the start of the playback function similar to a *let* specification in Lisp, much like an init time parameter of an instrument definition in Csound.

When a preset is started, it uses temporal recursion for calculating the parameters for each event (see Fig. 8). If a preset is called with the total duration set to NIL, the preset will run indefininetely, until it is stopped by using a special variable for each playback engine. In this case, the parameter functions are called with a x value of 0.

Like in other live-coding situations, the functions can be changed by typing into the emacs buffer and can get recompiled on the fly, reflecting their change immediately, while the preset is running.

Using interpolation functions, any other algorithmic means and MIDI or other controllers, very sophisticated trajectories or developments of parameter evolution are possible with very concise and short code while the preset is running.

The presets are allocated in structures, containing their source code and the compiled deltatime and parameter functions and can be saved in arrays, allowing presets to be hot swapped in realtime, with the contents of the emacs preset buffer exchanged and updated immediately. Between sessions, the source code of the presets can be saved to disk and later retrieved and recompiled.

With this infrastructure it is possible to build very elegant and powerful realtime performance instruments in a hybrid

```
(apply
 #'preset-play
 `(:preset 0
   :dur 4.5
   :gl ,*pool0*
   :dtimefn (n-exp (interp x 0 0 1 1) 0.03 1)
   :transpfn (between -5.0 5.0)
   :ampfn (n-lin (interp x 0 1 0.1 0 0.7 0 1 1)
                 (between -6 6) -32)))
```

Figure 8. Example call to the realtime poolplayer using preset 0 with a duration of 4.5 seconds. *preset-play* calls a recursive function evaluating the parameters of one poolevent on each recursion. The call above overwrites the following default parameter functions of the preset: Played samples are randomly selected from *pool0*, a list of preloaded lsamples. The deltatime (rhythm) decelerates from 0.03 to 1 second over the total duration, the transposition varies randomly between -5 and 5 halfsteps, offset by a midicontroller with ccnum 0 between 0 and 10 halfsteps and the amplitude increases from -32 dB between 0.1 and 0.7 of the total duration to random values between -6 and 6 dB and then decreases back to -32 dB.

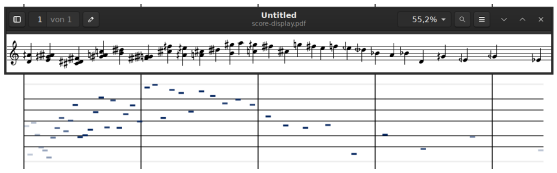


Figure 9. Example of the SVG output of the above call to the poolplayer and its corresponding score-display view. The midicontroller was moved from minimum to maximum and back to minimum during the 4.5 seconds of the performance.

between live-coding and pre-organisation. As the mechanism is modular, other event types for different synths and parameter sets can be defined by the user (see Section 3).

As an additional bonus, a realtime performance can be captured and saved as a SVG file on the symbolic level, allowing for later replay, visualisation, access and manipulation of the data using Inkscape or algorithmic means on the data set (see Fig. 9).

2.6 Online documentation and help system

In agreement with the extensive set of tools to get online help for Common Lisp's internal functions, Code inspection and Debugging infrastructure in current Lisp IDEs, *Clamps* makes use of Common Lisp's builtin capabilities for code inspection and description by generating the HTML documentation of its API directly from the docstrings of the Lisp function definitions etc., using the org format as intermediary format. A custom multipage html exporter written in emacs lisp exports the org files into HTML, complete with cross references, example code and images, using a a minimal custom markup in the docstrings of the Lisp source files. Inspired by the Common Music Dictionary, which is fully incorporated into the documentation system, the index of the *Clamps* dictionary currently contains ca. 600 functions, variables, structs and class entries, directly accessible with a keyboard shortcut from within any lisp source file. For displaying the locally stored documentation, the internal webserver of *Clamps* is used.

3. EXAMPLE: DEFINING AND INTEGRATING A NEW EVENTTYPE

Defining and integrating a custom event type into *Clamps* consists of three parts:

- Definition of the DSP routines to perform the event in realtime.
- Definition of the Common Music class and its realtime play and SVG export functions
- Definition of the import function for SVG

An example of a simple sine wave glissando with amplitude, duration, start- and end-frequency is given.

3.1 Definition of the DSP routines

The DSP code consists of two parts:

- Definition of a new UGen
- Definition of the synth which plays the sound

The synth definition is mandatory, the UGen definition is only necessary, if the UGen doesn't yet exist in *Incudine*. The Ugen definition is not strictly necessary in this use case, as it could be inlined in the dsp! definition, it is given here for completeness and also to demonstrate the ease of defining new UGens in Common Lisp without having to resort to a definition/compile/restart cycle using an alien programming language, like C/C++.

The Ugen is called *sinegliss**. *define-vug* accepts a name, a list of parameters, an optional docstring and default values and a body as arguments.

dsp! defines the synth. It evaluates its body forms, expanding the code of the UGens, which are implemented as Macros and returns the compiled binary lisp function to be called when playing the synth.

```
(in-package :scratch)

;;; Ugen definition:

(define-vug sinegliss* (f1 f2 dur amp phase)
  (* amp 0.1 (sin (+ (* +twopi+ (phasor (line f1 f2 dur #'free))) phase))))

;;; dsp ("synth") definition

(dsp! sinegliss (freq1 freq2 amp dur)
  "High precision sine wave oscillator with glissando from F1 to F2 ,
duration DUR, amplitude AMP and PHASE."
  (:defaults 440 1660 0.2 1)
  (foreach-frame
    (stereo (sinegliss* freq1 freq2 dur amp 0))))

;;; playing the synth:

(sinegliss 440 1880 0.5 1 :head 200)


;;; playing the synth:

(sinegliss 440 1880 0.5 1 :head 200)
```

Figure 10. Definition of the UGen and DSP of a sinegliss. The last line shows a call to play a note. *:head* is a keyword to add the synth instance to a certain group of the audiograph, similar to *SuperCollider*.

3.2 Definition of the Common Music class and its export functions

defobject defines a new event class in *Common Music*. Like *defclass* of CLOS it takes a name, a list of superclasses and the definition of slots. In addition, init parameters and event-streams can be defined for the event type.

```
(in-package :cm)

(defobject sinegliss (event)
    ((startkey :initform 60 :accessor sinegliss-startkey)
     (endkey :initform 60 :accessor sinegliss-endkey)
     (amplitude :initform 0.5 :accessor sinegliss-amplitude)
     (duration :initform 1 :accessor sfz-duration))
  (:parameters time startkey endkey amplitude duration)
  (:event-streams))

;;; realtime engine play function:

(defmethod write-event ((obj sinegliss) (to incudine-stream) scoretime)
  (declare (ignore to))
  (with-slots (startkey endkey amplitude duration) obj
    (let ((time (+ (rts-now) (* *rt-scale* scoretime))))
      (at time #'scratch::sinegliss (ou:mtof startkey) (ou:mtof endkey) amplitude duration))))

;;; play an event to the output:

(sprout (new sinegliss :time 0 :startkey 69 :endkey 93 :duration 1.5))

;;; svg export function:

(defmethod write-event ((obj sinegliss) (fil svg-file) scoretime)
  (with-slots (startkey endkey amplitude duration) obj
    (let* ((x-scale (x-scale fil))
           (stroke-width 0.5)
           (color "#f00000")
           (chan 200)
           (line (let ((x1 (float (* x-scale scoretime) 1.0)))
                   (make-instance
                    'svg-ie::svg-line
                    :x1 x1
                    :y1 (float startkey 1.0)
                    :x2 (+ x1 (float (* x-scale duration) 1.0))
                    :y2 (float endkey 1.0)
                    :stroke-width stroke-width
                    :opacity (float amplitude 1.0)
                    :stroke-color color
                    :attributes (format nil ":type sinegliss")
                    :id (new-id fil 'line-ids)))))
      (svg-file-insert-line line chan fil))))
```

Figure 11. Definion of the class and its export functions.

write-event is the method called by the playback or export functions of *Common Music*/*Clamps*. It's a multimethod, dispatching on the event type and the output type. Two methods, one for realtime output (*incudine-stream*) and one for SVG output (*svg-file*) are defined.

With these definitions in place, events can be created, played and exported to SVG:

```
(let ((events
       ;;; collect 50 instances of a sine gliss:
       (loop repeat 50
          for time below 3 by 0.05
          for startkey from 60 by 0.8
          for endkey = 94 then (- endkey 0.2)
          collect (new sinegliss :time time :startkey startkey :endkey endkey :duration 0.5))))

;;; play events:

  (sprout events)

;;; export events to svg file:

  (sprout events
          :to (namestring (clamps:svg-gui-path "sinegliss.svg"))
          :showgrid nil :piano-roll-vis nil :staff-system-vis nil :bar-lines-vis nil
          :barmultiplier 1 :startbar 0)

;;; display events in svg-player:

  (svg->browser "sinegliss.svg" :reload t))
```

Figure 12. Playing, exporting and displaying the new event type

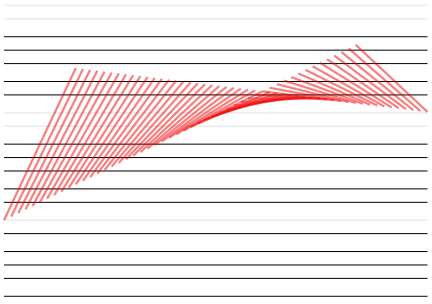


Figure 13. Exported SVG containing 50 sinegliss events.

3.3 Definition of the import functions

To facilitate editing the SVG and import it back into *Clamps*, an import callback function has to be defined and installed. The callback function will be invoked with a property list parsed from the SVG file. It contains the *:time, :duration, :keynum, :amplitude* and *:y2* properties plus all properties from the *:attributes* tag of every SVG element. *add-svg-assoc-functions* installs the callback invoked by the *import-events* function for SVG files of *Clamps* on the *:type* attribute given as the first element of the list submitted to *add-svg-assoc-functions*.

```
;;; import from svg

(defun svg->sinegliss (&rest args)
  "recreate a sinegliss from the coords of the svg element."
  (apply #'make-instance 'sinegliss
         (list* :startkey (getf args :keynum)
                :endkey (getf args :y2)
                (ou:get-props-list args '(:time :amplitude :duration)))))

;;; install the callback

(add-svg-assoc-fns `((sinegliss . svg->sinegliss)))

;;; import and play svg file:

(sprout (import-events "/tmp/test.svg"))
```

Figure 14. Definition of the import callback function and installing it into *Clamps*.

With these definitions in place, a full integration of the sinegliss type is accomplished and sinegliss events can be edited, its endpoints or opacity changed and used along MIDI-, sfz- or poolevents in the same SVG document. Even score-display will work, using the start frequency of the glissando as pitch. If full integration with *Clamps*' lilypond export is desired, a write event method specializing on the *fomus-file* output type can be added.

4. COMPONENTS OF *CLAMPS*

4.1 External components

Apart from the older Common Lisp systems for non-realtime work, like *Common Music*, *FOMUS* and *ATS*, *Clamps* relies heavily on two prerequisites to achieve its goals, *Incudine* and *CLOG*.

4.1.1 Incudine

Conceived and realized by Tito Latini and actively maintained by him. Incudine development started in 2013. It is a full fledged DSP programming environment with builtin MIDI and OSC support, nearly entirely written in Common Lisp.[7] It is one of the declared goals of Incudine to realize everything in Lisp, including the building blocks of the DSP engine, rather than relying on precompiled binaries. The processing loop of a dsp synth is done blockwise and Unit Generators can be defined to return a block of samples or on a sample-by-sample basis. Unit Generators are Lisp Macros and Synths are Functions. This approach enables the aforementioned change to the DSP code and its recompilation in realtime, even while synths are running, without the need to restart the system. Another advantage is the direct accessibility of the source code of all parts of Incudine, jumping to it with a single keystroke while working with the system, enabling an in depth understanding of its code-base.

To be competitive with other systems, great care must be taken to generate fast DSP code. With today's Lisp compilers, it is possible to get up to speed by declaring the types of the used symbols/variables and by avoiding

[7] There is only an exceptionally small footprint of glue code to the *Jack* Audio and MIDI API written in C, but even the code of the scheduler thread with its non-blocking ring-buffer is entirely written in Common Lisp.

garbage collecting code in Unit Generators and Synths. For all critical data structures, like tables, buffers, etc., Incudine uses external C-Arrays which can be directly accessed from within Lisp.

4.1.2 CLOG

The *Common Lisp Omnificent GUI* is a Common Lisp Framework to build graphical user interfaces using web technology. The GUI runs in standard web browsers. Some parts of CLOG are written in Javascript, but in general it is not necessary to write Javascript when working with CLOG, although a good knowledge of Javascript, HTML and CSS is helpful when using it. There are several advantages in using web browsers for building GUIs in *Clamps*: As the GUI is accessed using a webpage URL typed into the builtin browser, devices can be linked to the main program running the DSP code by Wireless Routers without any need for special configuration on the device and without the need to install additional software. For live performances it is thus very easy to have interactive GUIs for performers, using score display, buttons or any interactive components on Tablets or Mobile Phones. *Clamps* uses a mechanism guaranteeing that all GUIs are synchronized at all times: Any interaction or animation on any device is reflected on all other devices.

4.1.3 Common Music

Developed by Rick Taube between 1991 and 2014[8]. First developed in Common Lisp, it was moved to S7 scheme for version 3. *Clamps* uses the latest Common Lisp version 2.12. Common Music is an object oriented environment originally for the production of algorithmic music. Its main components are:

- A sophisticated class hierarchy for timed event types, ranging from MIDI events or csound score events to container types like sequences
- Different output types, ranging from MIDI or csound score files to notation or realtime output
- A process abstraction for the formal and temporal organization of events
- Powerful recursive item and pattern streams (cycle, line, stochastic, functional, markov,...)
- A collection of utility functions for music processing dealing with keynums, rhythms, frequencies, etc.

The advantage of this approach is a unified interface for defining events, organizing them into sequences and sending them to notation, MIDI, CLM, score files or to the realtime output using the same commands.

4.1.4 FOMUS

A notation system developed by David Psenicka in 2005, written in Common Lisp, implementing sophisticated constraint based algorithms for rhythmic interpretation, note spelling, part grouping and voice distribution of the input data. FOMUS can output to different formats (MusicXML, lilypond, CMN and MIDI). For *Clamps*, only the lilypond backend is supported.

4.1.5 ATS

Originally developed by Juan Pampin in the late 1990s using Common Lisp for the analysis, transformation and (re)synthesis of audio files using FFT based algorithms similar to SPEAR. In addition to the analysis of sinusoidal components performed by SPEAR, it also captures and synthesizes the residual noise component of the analyzed audio files using an additional filtered noise. Originally outputting to *CLM* in non-realtime, the original Code has been revitalized and the synthesis engine has been completely rewritten in *Incudine* for *Clamps* to work in realtime.

4.2 Internal Components

The following Components were written over the course of the last 15 years as Common Lisp Packages and recently adapted and integrated into *Clamps*:

4.2.1 Incudine Extensions

incudine-dsps

DSPs for buffer playback/recording and sample playback with granular stretching/transposing, an infrastructure for audio buses with bus DSPs for any number of channels, various levelmeter DSPs, some basic Unit Generators (VUGs) adapted to work with a blocksize > 1, vbap/mdap Ugens and miscellaneous basic envelopes and function tables.

incudine-bufs

Infrastructure dealing with loading, registering and finding buffers. It ensures that buffers are unique and not inadvertedly loaded repeatedly to keep the memory footprint low. In addition a search mechanism is implemented to look for files in user definable paths.

ats-cuda

DSPs for resynthesizing ATS analysis data in realtime.

4.2.2 Common Music Extensions

cm-incudine

Compatibility layer to bind the original MIDI, OSC and real-time scheduling related functions of CM 2 to their equivalent in Incudine.

cm-fomus

Thin wrapper to integrate FOMUS into cm. Most parts were written by Rick Taube for Common Music 2 and got adapted to *Clamps*.

cm-poolevt

Eventclass and infrastructure for sample-based granular playback.

cm-poolplayer

Infrastructure for Live-Coding based players of sample events.

cm-sfz

Eventclass and infrastructure for SFZ based realtime playback using incudine. Also implements a path based search infrastructure for SFZ definitions.

cm-svg

Infrastructure to import and export CM events from and to SVG files.

cm-utils

Package defining a generalized process mechanism called rt-proc, based on coroutines, allowing arbitrary lisp iteration contructs and multiple waits in its body and miscellaneous utility functions.

4.3 CLOG Extensions

clog-dsp-widgets

Implementation of various GUI elements for realtime interactive work or display, like Sliders, Number Boxes, Buttons, Toggles, Radiobuttons, VU-Meters, Scopes, etc. in a browser including a sophisticated infrastructure for connecting the state of Common Lisp objects with the GUI parts that were written in JavaScript to do client side rendering. The package not only facilitates the creation of sophisticated GUIs nowadays taken for granted in computer music environments, but allows for collaborative performances with multiple synced GUIs using standard hardware like Mobile Phones or Tablets without the need to install specialized programs.

clog-midi-controller

Implementation of the GUI part of various MIDI controllers.

ats-cuda-display

A Gui to interactively explore and play Sounds analyzed with ATS in realtime in the frequency domain.

cm-svg.rts

GUI for the interactive display and playback of SVG files exported from CM.

4.4 Other Extensions

cuda-usocket-osc

A small compatibility layer to make OSC work on OSX.

cl-plot

Plotting extension using gnuplot.

cl-refs

Package dealing with the state of variables of an application and their synchronization, similar to Javascript packages like *React* or *Vue*. It uses a declarative syntax to express the relation between stateful objects, like GUI objects on screen, running DSP synths, MIDI controller state, etc., facilitating a very easy to use connection between events and state handling.

cl-midi-control

Infrastructure for midi-ports and MIDI IO, capturing the state of MIDI input and classes for different midi-controllers.

cl-sfz

Base package to parse and load sfz files into incudine, used for *Clamps*'s internal sample-player.

5. CONCLUSIONS

Contrary to common belief, Common Lisp can be used for realtime performance. Utilizing the results of the last 30+ years of research on algorithmic composition, adding SVG import and export and different event types, it is shown that a very flexible and extendable interface for realtime and non-realtime work can be built. The expressive power of Lisp enables a unique workflow for the user and opens up tremendous possibilities with exceptional code transparency and malleability all the way down to the lowest implementation level. With extensive online documentation and a very powerful IDE *Clamps* is a competitive platform for the professional work of composers and performers alike.

Acknowledgments

The development of *Clamps* was only possible because major building blocks of it were available as open source projects. First and foremost I'm indebted to Tito Latini for his groundbreaking work on *Incudine*. It can not be overstated, what an achievement this package establishes and I'm very grateful for the help, he provided while developing DSP modules used in *Clamps*. Also *CLOG* played an important role in being able to implement the GUI parts of the program and, of course *Common Music* by Rick Taube, which not only accompanied my professional work for the last 33 years, but laid the foundations of the class, import and export system, which *Clamps* relies on. Many thanks also to David Psenicka and Juan Pampin for writing and publishing *FOMUS* and *ATS*.

There are many more fantastic packages integrated into *Clamps* and I'm grateful to all their programmers for making them available. And last not least I'm extremely grateful to the inventors of Lisp and all the people who have contributed to it in the last 60+ years to make it such a fantastic piece of software, which continues to bring a lot of joy into my life.

6. REFERENCES

[1] O. Finnendahl, "Clamps" 2025. [Online]. Available: https://codeberg.org/ormf/clamps

[2] T. Latini, "Incudine", 2013. [Online]. Available: https://incudine.sourceforge.net/

[3] D. Botton, "Clog", 2020-25. [Online]. Available: https://github.com/rabbibotton/clog

[4] H. K. Taube, "Notes from the metalevel: Introduction to algorithmic music composition", *Taylor & Francis*, 2005.

[5] J. Bresson, "Sound processing in openmusic", *DAFx*, 2006.

[6] M. Klingbeil, "Software for spectral analysis, editing, and synthesis", *ICMC*, 2005.

[7] J. Pampin, "Ats: a lisp environment for spectral modeling", *ICMC*, 1999.

[8] H. K. Taube, "Common music," 1991-2014. [Online]. Available: https://commonmusic.sourceforge.net/

Real-Time, Low-Latency, High Resolution Audio Spectral Analysis: Phase Matters

Alexandre R. J. François
Interactions Intelligence
alex@interactionsintelligence.org

ABSTRACT

This paper introduces an original approach to computing a spectral representation of audio signals, with high temporal and frequency resolution and high amplitude accuracy, in real-time and with low latency. Applying techniques from phase vocoders to make use of phase information, a new tracking resonator model extends the original Resonate model while retaining its iterative formulation and computational efficiency. A bank composed of frequency tracking resonators constantly self-tunes to the contents of the input signal, rendering the precise tuning of the resonators irrelevant, as long as the bank offers an appropriate coverage of the frequency range of interest for the target application. Self-tuning banks form the basis for an analysis technique that produces, in real-time, for each input sample, a list of uniquely identified and precisely tracked frequency components present in the input signal, together with their correct amplitudes. High temporal and frequency resolution spectrograms illustrate the spectral analysis of real musical signals in a familiar format. The detailed representations produced can potentially improve the quality and accuracy of any traditional application. They also offer promising prospects for real-time, low-latency applications such as accompaniment and improvisation systems. Encouraging initial synthesis experiments also motivate further investigation.

1. INTRODUCTION

This paper introduces an original approach to computing a spectral representation of audio signals with high temporal and frequency resolutions and high amplitude accuracy, in real-time and with low latency. The method applies phase vocoder techniques to extend the *Resonate* [1] model. The focus remains on harmonic analysis with the ultimate goal of applicability to real-time, low latency scenarios. The paper is organized as follows. Section 2 gives a quick overview of the uses of phase vocoder principles that inspired this work. Section 3 summarizes the *Resonate* model in the context of instantaneous frequency computation, first in a single resonator, then in a resonator bank. Section 4 extends this model to a frequency tracking resonator and its application in self-tuning resonator banks. Such banks form the basis for an analysis technique that produces, in real-time, for each input sample, a list of uniquely identified and precisely tracked frequency components present in the input signal, together with their correct amplitudes. High temporal and frequency resolution spectrograms illustrate the spectral analysis of real musical signals in a familiar format. Section 5 presents some initial synthesis results. The paper concludes with a summary of contributions and directions for future research and applications.

Copyright: ©2026 Alexandre R. J. François et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License 3.0 Unported, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

2. PHASE MATTERS

Spectral analysis for audio applications typically focuses on amplitude or power spectrum plots or spectrograms, treating the phase data as a computational by-product. Yet the phase carries important information, as evidenced in the phase vocoder literature.

The phase vocoder is primarily used for time-scale modifications and pitch transpositions of audio signals. As an analysis-synthesis technique, the phase vocoder produces an output signal that is identical or similar to the input signal, by way of a representation that is suitable for meaningful time-frequency manipulations. Portnoff [2] laid the foundations for its practical use in digital signal processing with an implementation based on the Fast Fourier Transform (FFT). Moorer [3] and Dolson [4] explored its use for musical applications.

Phase information is crucial in both analysis and synthesis. The analysis stage computes, from the input signal, a spectral representation consisting of the time varying amplitudes and instantaneous frequencies, by estimating phase time differentials (temporal phase consistency). The synthesis stage makes use of the inverse Fourier Transform (iFFT) to produce an audio signal similar to the input signal from the possibly modified frequency domain representation produced in the analysis stage. If no modification occured, because of the properties of the FFT and iFFT, the output signal is mathematically identical to the input signal (i.e. the phase vocoder is an identity system). In case of modifications, phase constraints must be enforced in order to generate a high quality output signal.

The analysis phase of the FFT-based formulation actually applies the Short-Time Fourier Transform (STFT) to the input signal to produce a frame-based frequency domain representation. This comes at the cost of added mathematical complexity, which compounds in the synthesis phase [5]. The STFT has the advantage of being computationally efficient, but at the cost of significant constraints. For example, the frequency bands are fixed, distributed on

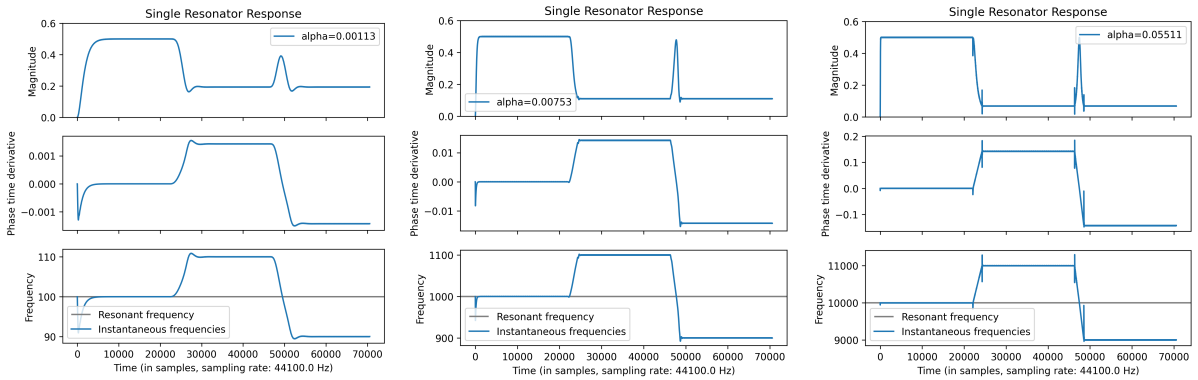


Figure 1. Magnitude, phase difference and estimated instantaneous frequencies over time in response to a sinusoidal signal of fixed amplitude 1, whose frequency varies around the resonator's resonant frequency: 100-110-90 Hz, 1000-1100-900 Hz and 10000-11000-9000 Hz; 0.05s transition duration.

a linear scale (while humans perceive frequencies on a logarithmic scale) and dictated by the window size and input signal sampling rate. Furthermore, the sliding window approach of the SFFT introduces ambiguity in the phase estimation, which necessitates extra processing (phase unwrapping). These and other characteristics inherent to the FFT pose challenges that affect computational complexity, quality of the analysis and of the synthesized output signal. Audio signals produced by phase vocoders exhibit characteristic features such as "transient smearing" and a number of effects referred to as "phasiness." Various analyses and mitigation techniques, in the context of musical applications, are the subject of a substantial body of work (see e.g. [6, 7, 8, 9]).

In contrast to the FFT, *Resonate* builds on a resonator model that accumulates the signal contribution around its resonant frequency in the time domain using the Exponentially Weighted Moving Average (EWMA). A compact, iterative formulation of the model computes a magnitude and phase update at each input signal sample, requiring no buffering and involving only a handful of arithmetic operations. Banks of resonators, independently tuned to perceptually relevant frequency scales, compute an instantaneous, perceptually relevant estimate of the spectral content of an input signal in real-time. Both memory and per-sample computational complexity of such a bank are linear in the number of resonators, and independent of the number of input samples processed, or duration of processed signal. Since the resonators are independent, their resonant frequencies or time constants can be set arbitrarily, and all per sample computations can be parallelized across resonators.

These properties suggest a much simpler, efficient and possibly novel application of phase vocoder techniques, the first results of which are the subject of this paper.

3. INSTANTANEOUS FREQUENCIES

This section summarizes the *Resonate* model originally described in [1], in the context of instantaneous frequency computation, first in a single resonator, then in a bank of independently tuned resonators.

3.1 Single Resonator

Each *Resonate* resonator (indexed k), characterized by its resonant frequency $f_k = \frac{\omega_k}{2\pi}$, is described by a complex number R_k whose amplitude captures the contribution of the input signal component around frequency f_k. Equation 2 shows the recursive update formula for R_k by way of a phasor P_k, applied for each sample x of a real-valued input signal $x(t) \in [-1, 1]$, regularly sampled at sampling rate sr. $\Delta t = 1/sr$ is the sample duration.

$$P_k(t) = P_k(t - \Delta t)e^{-i\omega_k \Delta t} \quad (1)$$

$$R_k(t) = (1 - \alpha_k)R_k(t - \Delta t) + \alpha_k x(t)P_k(t) \quad (2)$$

The parameter $\alpha_k \in [0, 1]$ dictates how much each new measurement affects the accumulated value; it can be expressed as a function of the system's time constant, set heuristically as a function of the resonant frequency f (intuitively inversely proportional to the frequency). The formula in Equation 3 is a reasonable heuristic for the frequency range of interest in audio applications (20-20000 Hz).

$$\alpha_k = 1 - e^{-\Delta t \frac{f_k}{log(1+f_k)}} \quad (3)$$

A smoothed state $\tilde{R}_k$ is produced by applying the EMWA to R_k with parameter β_k to dampen power and phase oscillations, as shown in Equation 4. In practice, $\beta_k = \alpha_k$ yields satisfactory results.

$$\tilde{R}_k(t) = (1 - \beta_k)\tilde{R}_k(t - \Delta t) + \beta_k R_k(t) \quad (4)$$

This formulation is consistent with the first steps in the filterbank interpretation of the phase vocoder analysis as described by Dolson in [4], namely heterodyning followed by lowpass filtering. The amplitude at time t of a *sinusoidal input signal of same frequency as the resonant frequency* is directly proportional to the magnitude of the complex number $\tilde{R}_k(t)$.

At each time step, the phase difference $\Delta\phi(t)$ between the current and previous value of $\tilde{R}$ provides an estimate of the phase's time derivative, to compute the corresponding instantaneous frequency, as shown in Equation 5.

$$f_k(t) = f_k(t - \Delta t) + \frac{\Delta\phi_k(t)}{2\pi\Delta T} \quad (5)$$

Using the property that the phase of a complex number multiplied by the conjugate of another complex number is the phase difference between the two numbers yields Equation 7 for $\Delta\phi_k(t)$, which involves only one principal value

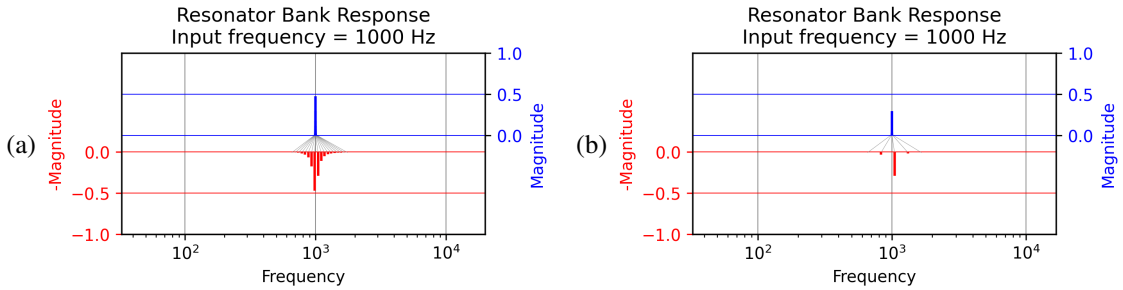


Figure 2. Steady state magnitudes for banks of resonators tuned at geometrically spaced frequencies, in response to a sinusoidal signal of fixed amplitude (1) and frequency (1000 Hz). The bottom graph plots the opposite of each resonator's magnitude at the resonator's resonant frequency position on the x-axis; the top graph plots the resonators' magnitudes at the instantaneous frequency. (a) 112 frequencies from 32.7 to 19910.2 Hz, (b) 28 frequencies from 32.7 to 16742.4 Hz. Sampling rate: 44100 Hz.

argument computation. Because $\Delta\phi_k(t)$ is computed for each input sample, there is no concern of phase unwrapping, a crucial necessary step in FFT-based vocoder analysis when the hop length is greater than 1 (or when explicitly computing and subtracting phases).

$$D_k(t) = \tilde{R}_k(t)\overline{\tilde{R}_k}(t - \Delta t) \quad (6)$$

$$\Delta\phi_k(t) = Arg(D_k(t)) \quad (7)$$

Finally, applying the EMWA with time constant γ_k to the estimated frequency yields the stabilized estimate of Equation 8. In practice, $\gamma_k = \alpha_k/2$ yields satisfactory results.

$$\tilde{f}_k(t) = f_k(t - \Delta t) + \gamma_k \frac{\Delta\phi_k(t)}{2\pi\Delta T} \quad (8)$$

The complex numbers P, R and $\tilde{R}$ capture the full state of the resonator. Updating the state at each input signal sample only requires a handful of arithmetic operations. Equations 6 and 8 add a few more for instantaneous frequency estimation. The argument computation of Equation 7, typically obtained with the `arctan2` transcendental function, is the only source of significant additional computational complexity and potential source of significant increase in processing time. Fortunately this operation is supported efficiently on modern hardware, and the overall per input sample update time remains of the same order as for the basic resonator in reference implementations, several orders of magnitude better than real-time.

Figure 1 shows plots of the magnitude, phase difference and estimated instantaneous frequencies over time in response to a sinusoidal signal of fixed amplitude 1, whose frequency varies around the resonator's resonant frequency, with a 0.05s transition duration. In each case, the instantaneous frequencies computed by the resonator correctly track the actual input signal's frequency. As should be expected, convergence takes longer for lower frequencies, as the resonator dynamics tuning must (and do) take into consideration the laws of physics. The phase time derivative estimate is proportional to the difference between the resonator's resonant frequency and the input sinusoid's frequency (the property used for instantaneous frequency estimation), but so is the magnitude of the resonator. Indeed, for a fixed input signal amplitude, the resonator's response is (by design) maximal if the input's frequency is the same as the resonant frequency, and decreases with the distance between input and resonant frequencies. This is evidenced for example by the peaks in the magnitude plots that coincide with the second input frequency transition, as the input frequency value first moves towards the resonant frequency and then away. Consequently, correctly estimating the actual amplitude of the input signal component is not as straightforward as computing the magnitude of the resonator, but rather should in addition take into account the instantaneous frequency estimate [5].

3.2 Resonator Banks

Banks of resonators as described above, independently tuned to perceptually relevant frequency scales, compute an instantaneous estimate of the spectral content of the input signal, including not only magnitudes but also phase time derivative and instantaneous frequency estimates. The overall update computation time for each input sample remains of the same order as for the basic resonator bank (without instantaneous frequency computation overhead) in reference implementations, well within real-time performance on modern hardware.

The plots in Figure 2 show steady state magnitudes for banks of resonators tuned at geometrically spaced frequencies, in response to a sinusoidal signal of fixed amplitude (1) and frequency (1000 Hz). The bottom graph plots the opposite of each resonator's magnitude at the resonator's resonant frequency position on the x-axis; the top graph plots the resonators' magnitudes at the instantaneous frequency position. Resonators that have significant magnitude, i.e. those whose resonant frequency is in the vicinity of the input signal's frequency, agree on the correctly estimated instantaneous frequency. For the purpose of estimating instantaneous frequencies, the actual resonant frequencies in the bank become irrelevant, as long as their density and distribution resonably covers the range of interest. For musical application, a density of 12 resonators per octave (corresponding to 1 semi-tone intervals) covering the range 20-20000 Hz seems reasonable. As illustrated in Figure 2b, a much sparser bank performs just as well in the case of a single frequency input.

The relevant resonators have different magnitudes however, as the distance from the input frequency and each resonator's resonant frequency is different (by construction). The resonator whose resonant frequency is closest to the actual input frequency has the largest magnitude (clos-

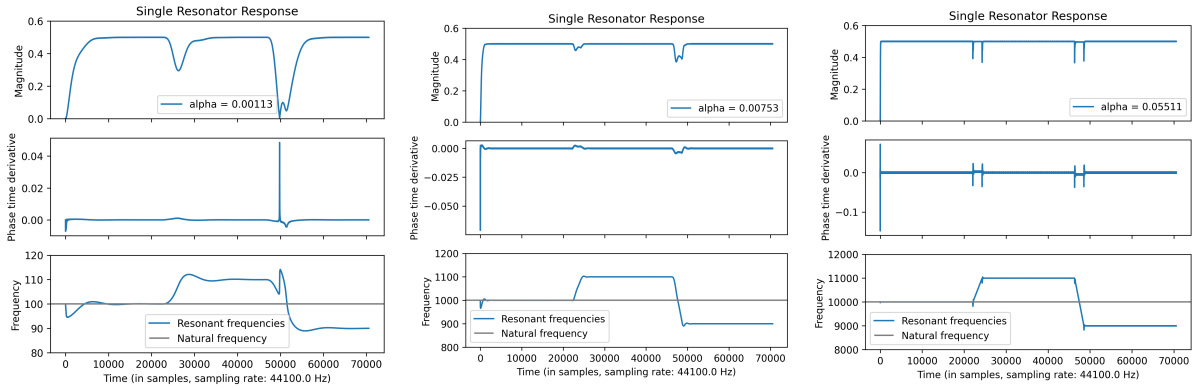


Figure 3. Magnitude, phase difference and resonant frequency over time in response to a sinusoidal signal of fixed amplitude 1 whose frequency varies around the tracking resonator's natural frequency (transition duration 0.05s).

est to the correct value for the unit amplitude input), only directly proportional to the input signal's component amplitude if the resonator's resonant frequency matches the input's instantaneous frequency. The next section builds on this observation to expand the model in a direction resolutely inconceivable with FFT-based approaches.

4. FREQUENCY TRACKING

In FFT-based phase vocoders, the analysis buffer length and the input signal sampling rate dictate the density and distribution of frequency bins. By construction, frequency bands remain constant. The *Resonate* model imposes no such constraint; its dynamic nature and efficiency make the frequency tracking approach developed in this section possible and practical.

4.1 Frequency Tracking Resonator

The frequency tracking resonator expands the basic resonator model by allowing the resonant frequency to change over time: in the absence of significant information (i.e. below a set magnitude threshold), the resonant frequency remains constant, equal to the resonator's natural frequency. In the presence of significant response, however, the resonant frequency tracks the estimated instantaneous frequency.

Resonant frequency tracking is easily achieved by substituting the adaptive phasor of Equation 9 for the fixed angular velocity phasor of Equation 1. Angular velocity changes are stabilized by applying an EWMA with time constant γ_k. In practice, $\gamma_k = \alpha_k/2$ yields satisfactory results. The remainder of the model update logic remains unchanged. From a computational complexity point of view, this adaptation does not add expensive operations, and performance remains well within real-time limits in reference implementations on modern hardware.

$$\omega_k(t) = \omega_k(t - \Delta t) + \gamma_k \frac{\Delta\phi_k(t)}{\Delta T}$$
$$P_k(t) = P_k(t - \Delta t)e^{-i\omega_k(t)\Delta t} \quad (9)$$

Strictly speaking, a change in the resonant frequency should also trigger a consistent update in the dynamics parameters α_k, β_k and γ_k. However if the tracked frequency remains in the vicinity of the resonator's frequency, such an update is not necessary in practice.

Figure 3 shows plots of the magnitude, phase difference and resonant frequencies over time in response to the same sinusoidal signals as those of Figure 1. As the input frequency changes, the system's resonant frequency tracks the estimated input frequency. Consequently, the phase time derivative remains close to 0, and the resonator's magnitude remains close to the maximum value (0.5 for an input signal of amplitude 1).

Here again, convergence takes longer for lower frequencies. However, convergence time from one frequency to another should be inversely proportional to the distance between the two frequencies. A resonator whose natural frequency is closer to the input frequency should converge faster in response to a step signal. Tracking of a smoothly changing frequency depends on the rate of change in the input frequency and the resonator's dynamics, but should be almost instantaneous.

4.2 Self-tuning Resonator Bank

A bank composed of frequency tracking resonators constantly self-tunes to the contents of the input signal.

The overall update computation time for each input sample remains of the same order as for the basic resonator bank in reference implementations, well within real-time performance on modern hardware. As an informal reference point, a C++ implementation that makes use of concurrency consistently updates a bank of 112 resonators in well under 2 μs on platforms ranging from iPad A16 to M4Pro MacBook Pro. Vectorized implementations taking advantage of Single Instruction Multiple Data (SIMD) architectures where available bring the per-sample update duration down to under 650 ns on the same platforms. The open source package `Oscillators`[1] offers Swift and C++ reference implementations of the algorithms described in this paper.

The plots in Figure 4 show steady state magnitudes for banks of resonators tuned at geometrically spaced frequencies, in response to a sinusoidal signal of fixed amplitude (1) and frequency (1000 Hz). The bottom graph plots the opposite of each resonator's magnitude at the resonator's natural frequency position on the x-axis; the top graph plots the resonators' magnitudes at the resonator's (tracked) resonant frequency.

[1] `github.com/alexandrefrancois/Oscillators`

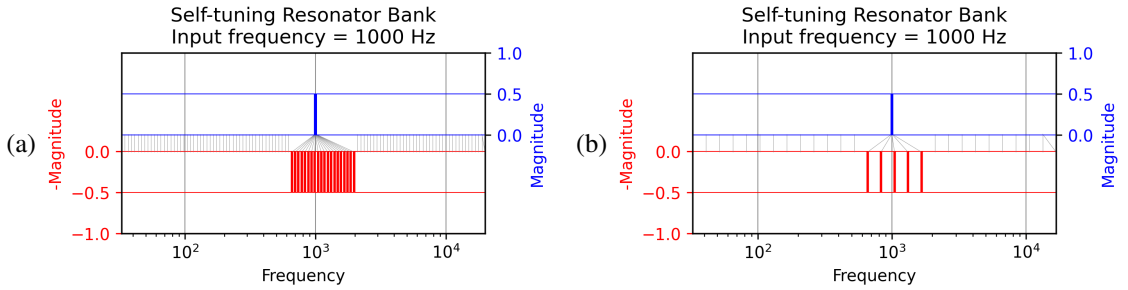


Figure 4. Steady state magnitudes for banks of tracking resonators tuned at geometrically spaced natural frequencies, in response to a sinusoidal signal of fixed amplitude (1) and frequency (1000 Hz). The bottom graph plots the opposite of each resonator's magnitude at the resonator's natural frequency position on the x-axis; the top graph plots the resonators' magnitudes at the resonator's (tracked) resonant frequency. (a) 112 frequencies from 32.7 to 19910.2 Hz, (b) 28 frequencies from 32.7 to 16742.4 Hz. Sampling rate: 44100 Hz.

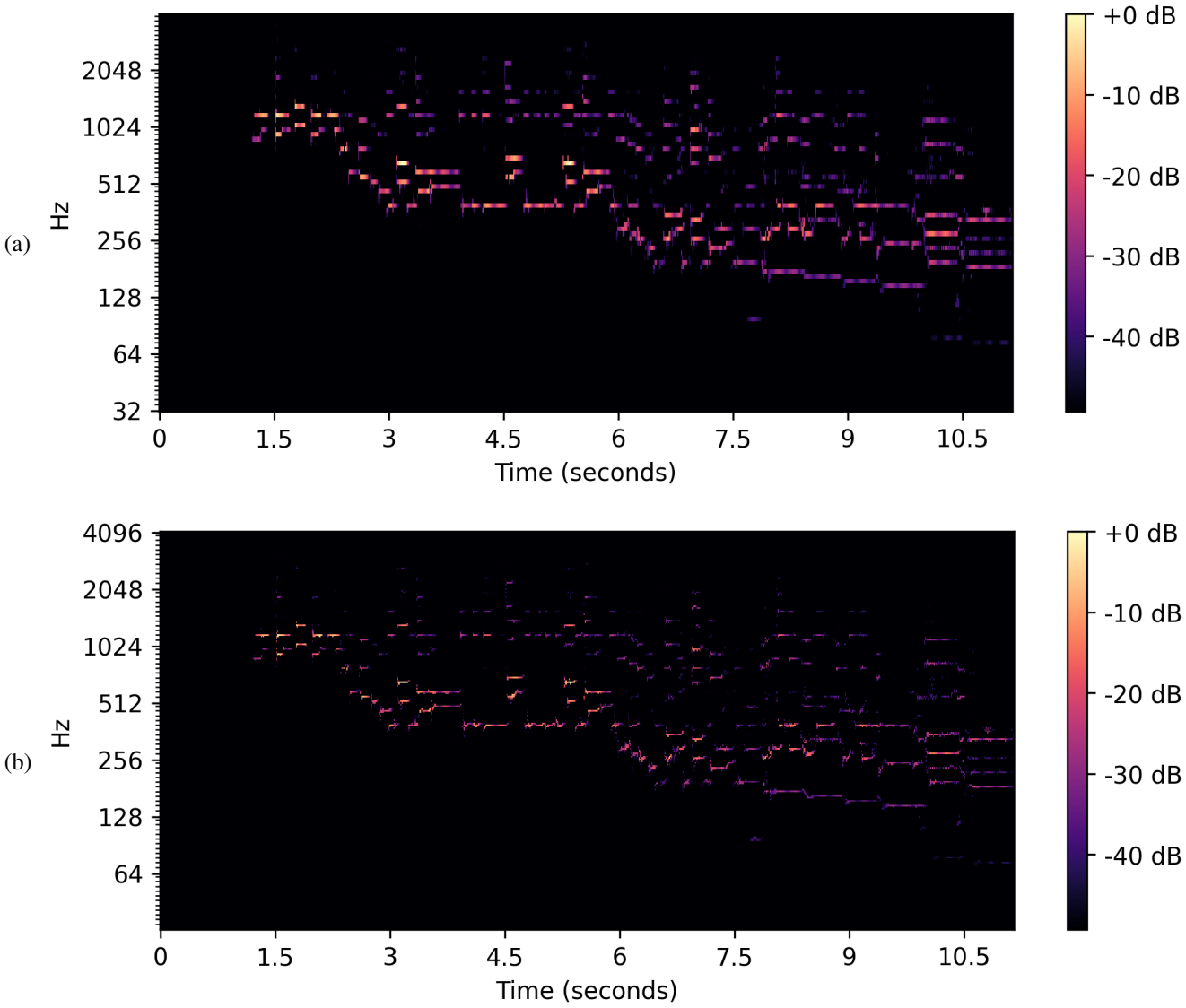


Figure 5. Tracking power spectrogram of a short musical excerpt (first 11.15s of *Who's Loving You* by The Jackson 5; electric piano). Sampling rate: 44100 Hz, 84 resonators with geometrically spaced natural frequencies in the range 32.7-3950.7 Hz, display hop length: 512 samples, 84 (a) and (b) 252 (b) display lines.

As the system adapts, the magnitude of each resonator remains close to the maximum. When processing a more natural signal, the magnitude of the resonators will track the amplitude of the frequency component in the input signal. With this approach, the actual tuning (natural frequencies) of the resonators in the bank does not matter, as long as the bank offers an appropriate coverage of the frequency range of interest. The bank continuously self-tunes to the content of the input signal. As already observed, optimal density and distribution of natural frequencies may vary for different applications. Figure 4b illustrates how a much sparser bank can satisfactorily adapt to at least a single frequency input signal.

As nearby resonators converge towards the same frequency tracking the input signal's dominant component frequency in a given region, the output of the resonator whose natural frequency (from which the dynamics parameters are set) is closest to the tracked frequency is given precedence. Consequently, the dynamics of the resonators need not be updated as a function of the tracked frequency. The outputs from the other overlapping resonators are discarded, as they all converge to the same frequency and magnitude and are therefore redundant. Let $\mathcal{T}(t)$ denote the set of indices of the tracking resonators thus selected at each time.

Analysis of an audio signal by a self-tuning resonator bank as described here produces, in real-time, for each input sample, a list of uniquely identified and precisely tracked frequency components present in the input signal, together with their correct amplitude. The result is a spectral model of the input signal exhibiting high temporal resolution (input signal sample rate), high frequency resolution and high amplitude accuracy.

Note that the computations also include the phase of the tracking components; correct preservation of temporal phase coherence across component handover is not critical for analysis purposes, but a crucial area of future improvements for synthesis (see Section 5).

4.3 Super-resolution Tracking Spectrograms

To illustrate the output of the self-tuning resonator bank processing of real signals in a familiar format, Figure 5(a) shows a power spectrogram computed off-line in a Python environment and rendered with Librosa's [10] `specshow` function. For each rendered time slice, the output of each tracking resonator is rendered in the line corresponding to its current resonant frequency. The lines correspond to the logarithmically distributed natural frequencies of the bank's resonators (12 lines per octave). In case of tracked frequency overlap between nearby resonators, the value at the line is set to the maximum power. (This is equivalent to keeping the power of the resonator whose resonant frequency is closest to its natural frequency.) Such a spectrogram is however a low fidelity rendering of the available instantaneous frequency information, which is computed with much higher accuracy. Figure 5(b) shows the same data rendered as a spectrogram with triple the original frequency resolution (36 lines per octave). In both cases, the display hop length of 512 samples does not reflect the high temporal resolution of the computed data, which comprises one estimate per input sample.

Figures 6 and 7 show screenshots of power spectrograms of musical passages, computed and rendered in real-time using the Oscillators app [2], with a display hop length of 64 samples, to better illustrate the high temporal resolution of the computed spectral representation. The number of frequency lines in the spectrograms is triple the number of resonators in the bank. Still images of such spectrograms necessarily cover shorter durations of content (the Resonate website links to video captures of real-time demonstrations) but reveal more details. Figure 6 features solo violin with fast notes, vibratos and glissandi (from *Paganini's Caprices*). Figure 7 illustrates fine harmonic content tracking and capture of rhythmic transients in rock music (Queen's *Under Pressure* and *Another One Bites The Dust*).

The information computed seems suitable for real-time, low-latency MIDI event generation from the input signal, a promising potential application currently under investigation. Yet, these visual representations also highlight the limitations of a fixed-size grid to represent and use the output of the proposed model, which consists, for each input sample, of a variable size list of arbitrary frequency-amplitude pairs. In particular, optimal use of such dynamic unstructured data may call for a different architecture than for example the prevailing fixed-size tensor that underlies current machine learning architectures.

5. SYNTHESIS

The traditional phase vocoder relies on the iFFT for the synthesis phase, so that in the absence of parameter modifications, the re-synthesized signal is identical to the input signal. Possible modifications include frequency shift and time modification. A similar transform can be achieved with *Resonate*, albeit without the identity property, which would require impractical additional constraints on the frequency density and distribution in the resonator bank. Applying the inverse phasors of Equation 10 to individual tracking resonators $\tilde{R}_k(t)$ yields the complex numbers $S_k(t)$ of Equation 11, from which an audio signal corresponding to the input signal's component can readily be produced. The reverse phasor can optionally introduce a frequency shift ratio fs and a different target sampling rate $sr_s = 1/\Delta t_s$ (time manipulation). A frequency ratio of 1 keeps the frequency unchanged, values greater or lower than 1 result in higher or lower frequency, respectively. A sampling rate equal to the input's sampling rate keeps the timing/speed, a greater or lower sampling rate results in a shorter or longer duration output signal, respectively.

$$P_k^{-1}(t) = P_k^{-1}(t - \Delta t_s)e^{i\omega_k(t) fs \Delta t_s} \tag{10}$$

$$S_k(t) = \tilde{R}_k(t) P_k^{-1}(t) \tag{11}$$

Equation 12 produces an audio signal $s(t)$ which captures salient features of the input signal. The signal is constructed from non overlapping tracking resonators, so that individual components occur only once in the sum. Consequently, vertical phase coherence is not a concern and there is no need for phase locking [6]. However, horizontal phase discontinuities will occur when the dominant resonator in a frequency region changes, potentially causing some clicks in the output signal.

[2] `alexandrefrancois.org/Oscillators`

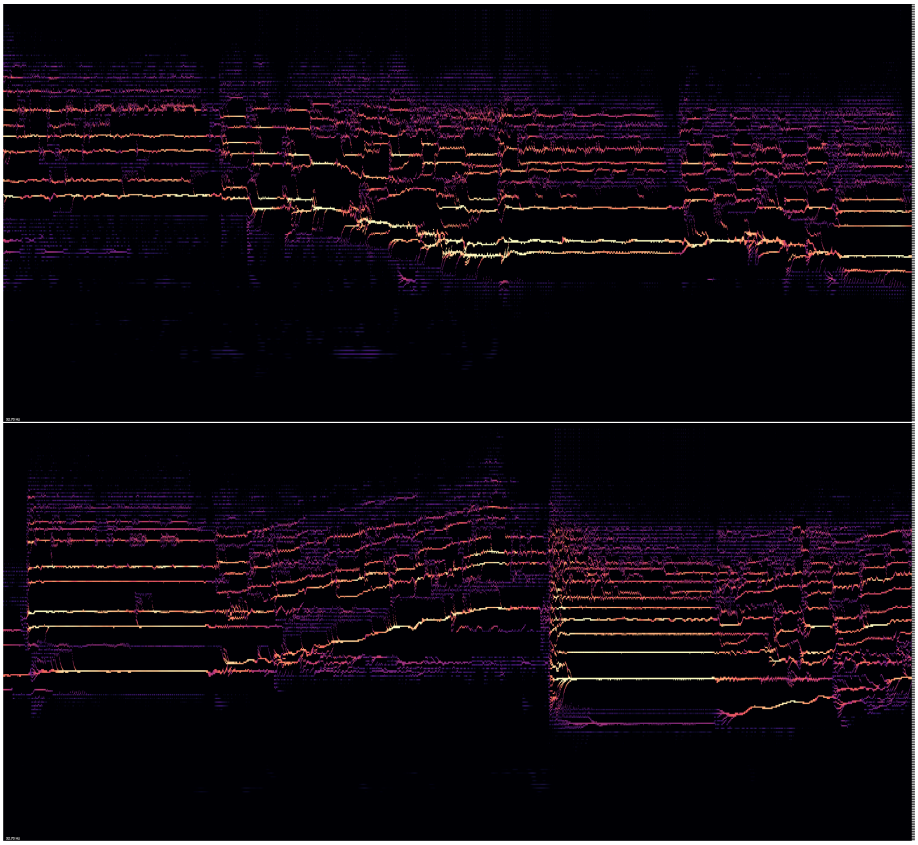


Figure 6. Tracking power spectrogram of short solo violin excerpts from the *Paganini Caprice no.24 in A Minor*. Sampling rate: 44100 Hz, 112 resonators, with geometrically spaced natural frequencies in the range 32.7 to 19910.2 Hz, display hop length: 64 samples, 336 display lines.

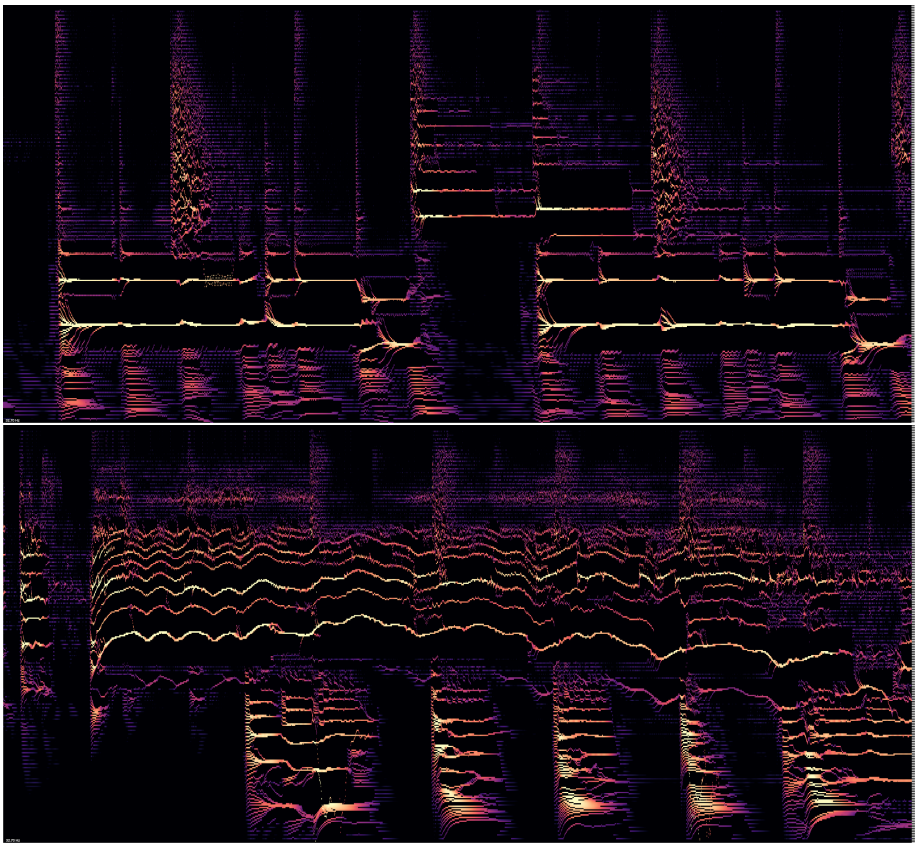


Figure 7. Tracking power spectrogram of short excerpts from Queen's *Under Pressure* (top) and *Another One Bites The Dust* (bottom). Sampling rate: 44100 Hz, 112 resonators, with geometrically spaced natural frequencies in the range 32.7 to 19910.2 Hz, display hop length: 64 samples, 336 display lines.

$$s(t) = KRe(\sum_{k \in \mathcal{T}(t)} S_k(t)) \quad (12)$$

A complete study and evaluation of synthesized signals is out of the scope of this paper, but early experiments with and without frequency shifts and time modifications are encouraging. See the Resonate project website [3] for examples.

Informally, the synthesized audio seems to capture the harmonic content of the input signal. It also exhibits transient smearing (due to the dynamics of the frequency tracking) and some other artifacts likely due to the partial nature of the reconstruction. In these first experiments, the clicks caused by horizontal phase discontinuities are mediated with simple smoothing. If more research is needed to improve the quality of the synthesized audio, the method seems promising with possible applications in real-time accompaniment and improvisation systems, for example.

6. SUMMARY AND FUTURE WORK

This paper described an original approach to computing a high temporal and frequency resolution, and high amplitude accuracy spectral representation of audio signals, in real-time and with low latency.

Applying techniques from phase vocoders to make use of phase information, a new tracking resonator model extends the original *Resonate* model while retaining its efficiency and characteristic properties. A bank composed of frequency tracking resonators constantly self-tunes to the contents of the input signal. With this approach, the actual tuning of the resonators in the bank does not matter, as long as the bank offers an appropriate coverage of the frequency range of interest for the target application.

The extended model formulation builds on the same compact, iterative approach, which affords computing an update at each signal input sample, requiring no buffering and involving only a handful of arithmetic operations, and adding one single transcendental function computation per resonator. Straightforward reference implementations exhibit real-time performance on modern hardware, while vectorized parallel implementations taking advantage of SIMD architectures further reduce computation time, and therefore latency.

Self-tuning banks form the basis for an analysis technique that produces, in real-time, for each input sample, a list of uniquely identified and precisely tracked frequency components present in the input signal, together with their correct amplitude. High temporal and frequency resolution spectrograms illustrated the analysis result of real musical signals in a familiar format.

The detailed representations produced can potentially improve the quality and accuracy of any traditional applications, for example in Music Information Retrieval, tonal analysis and visualization to name a few. They also offer promising prospects for real-time, low-latency applications such as accompaniment and improvisation systems, via real-time audio synthesis or MIDI event generation. Encouraging initial synthesis experiments, with and without time and frequency manipulations, invite further investigation and explorations.

The results presented here remain preliminary and more work is needed to improve the quality of the analysis and synthesis outputs. Optimal use of the dynamic unstructured data produced may require a departure from, for example, the prevailing fixed-size tensor that underlies current machine learning architectures. Another interesting potential area for future research is the application of machine learning techniques to optimize the values of model parameters, such as the density and distribution of natural frequencies in self-tuning banks. To facilitate wider community contributions to the development of the model, in addition to resources already referenced above, the open source module `noFFT` [4] provides python and C++ implementations of *Resonate* functions and Jupyter notebooks with the code used to produce the figures in the paper.

7. REFERENCES

[1] A. R. François, "Resonate: Efficient Low Latency Spectral Analysis of Audio Signals," in *Proceedings of the International Conference on Computer Music*, Boston, MA, USA, 2025, pp. 251–258.

[2] M. R. Portnoff, "Implementation of the digital phase vocoder using the fast Fourier transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-24, no. 3, pp. 243–248, 1976.

[3] J. A. Moorer, "The Use of the Phase Vocoder in Computer Music Applications," *Journal of the Audio Engineering Society*, vol. 26, no. 9, pp. 717–727, 1978.

[4] M. Dolson, "The Phase Vocoder: A Tutorial," *Computer Music Journal*, vol. 10, no. 4, pp. 14–27, 1986.

[5] D. W. Griffin and J. S. Lim, "Signal estimation from modified short-time Fourier transform," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. IEEE, 1983, pp. 804–807.

[6] M. Puckette, "Phase locked vocoder," in *Proceedings of the 1995 International Computer Music Conference (ICMC)*, 1995, pp. 245–248.

[7] J. C. Brown and M. Puckette, "A High-Resolution Fundamental Frequency Determination Based on Phase Changes of the Fourier Transform," *Journal of the Acoustical Society of America*, vol. 94, no. 2, pp. 662–667, 1993.

[8] J. Laroche and M. Dolson, "Improved phase vocoder time-scale modification of audio," *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 3, pp. 323–332, 1999.

[9] Z. Průša and N. Holighaus, "Phase Vocoder Done Right," in *25th European Signal Processing Conference (EUSIPCO)*. IEEE, 2017, pp. 1–5.

[10] B. McFee *et al.*, "librosa/librosa: 0.10.2.post1," May 2024. [Online]. Available: doi.org/10.5281/zenodo.11192913

[3] `alexandrefrancois.org/Resonate`

[4] `github.com/alexandrefrancois/noFFT`

Magnetic Memory Rushnyk

Jocelyn Ho
The University of Sydney
jocelyn.ho@sydney.edu.au

Sofy Yuditskaya
De Vinci Higher Education, Paris
sofy.yuditskaya@devinci.fr

Margaret Schedel
Stony Brook University
margaret.schedel@stonybrook.edu

ABSTRACT

In what ways can two separate but traditionally gendered types of memory—symbolic embroidered memory encoded in ritualistic textile objects and magnetically encoded memory in historic aeronautical computing—intersect to create a performative decoding instrument? This paper introduces Magnetic Memory Rushnyk, a woven score-instrument that synthesizes symbolic embroidered memory with hand-woven magnetic memory. The project investigates the convergence of ritual textile embroidery (historically used to carry social and generational meaning) with the "core memory" techniques developed for early computing and aerospace contexts. By embedding magnetic elements into a traditional East Slavic rushnyk (ritual cloth), we create a single performative object where weaving serves as a material practice of memory-making. The installation is activated through a custom-built electromagnetic interface, allowing performers to "read" the cloth through touch, gesture, and sound. This work extends the Women's Labor artistic research project, moving from the sonification of domestic tools to the performative decoding of ritual objects, proposing a model of memory that is materially durable and physically traversable.

1. INTRODUCTION

Magnetic Memory Rushnyk is a woven score-instrument that brings together the domain of symbolic embroidered memory and hand-woven magnetic storage. The project asks what happens when ritual textile embroidery, used for centuries to carry specific social, protective, and generational meanings, is combined with the material techniques developed to store digital information in early computing and aerospace contexts. These traditions, though separated by context and era, meet in a single performative object that treats weaving as a fundamental technical practice of memory-making.

This project extends the textile score/instrument practice established in *Women's Labor*, a feminist artistic research project that interrogates gendered labor through embedded instruments and accompanying textile scores [1]. In contrast to the housework-based tools developed elsewhere in the project, which sonify the labor of maintenance through hanging or ironing laundry, *Magnetic Memory Rushnyk* draws on ritual textile objects. These objects are deeply domestic and created "in-house," yet their inscriptions are activated through ceremonial performance rather than cleaning or repair.

The *rushnyk* functions here not merely as a decorative covering, but as a binding and mediating technology. By integrating magnetic materials into the weave, the project transforms the textile into a readable archive. As a score, the rushnyk is read through touch, proximal gesture, and electromagnetic induction, making memory materially present and performable. This paper details the theoretical framework of patterned inscription that underpins the work, the historical lineage of the project, and the technical realization of the installation.

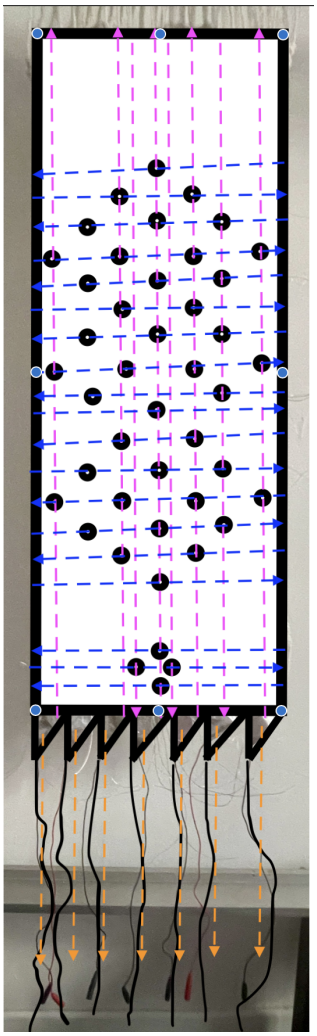


Figure 1. Wiring Diagram Level 1

Copyright: ©2026 Ho et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License 3.0 Unported, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

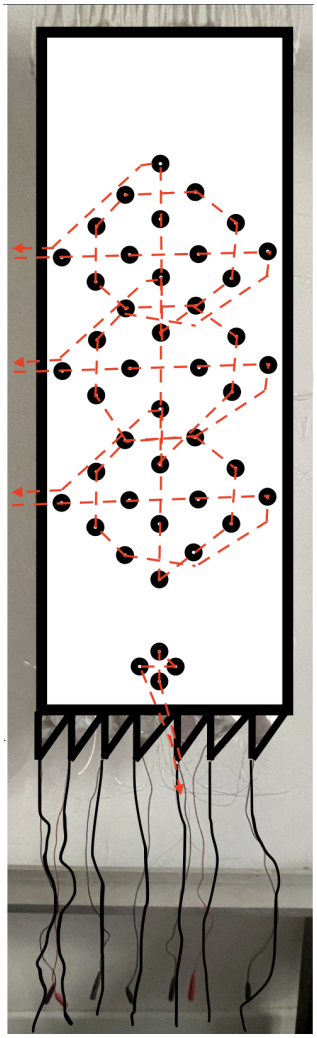


Figure 2. Wiring Diagram Level 2

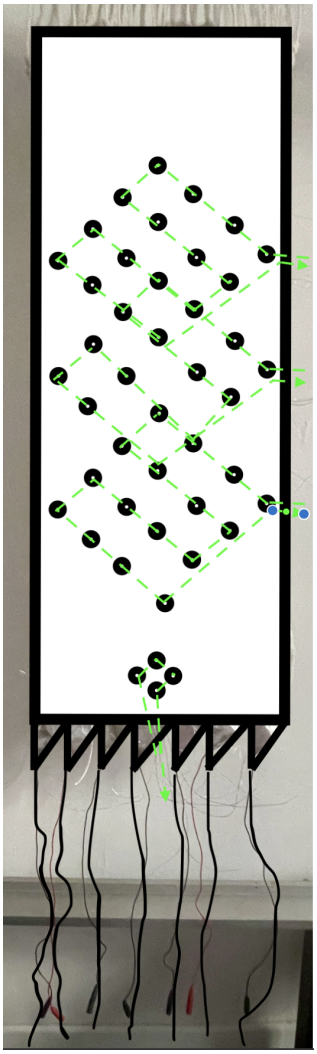


Figure 3. Wiring Diagram Level 3

2. PATTERNED INSCRIPTION AND PERFORMATIVE MEMORY

Across diverse ritual, technological, and artistic practices, memory has rarely been treated as an abstract repository to be accessed on demand. Instead, it has been stabilized through patterned inscription: meaning committed to matter through repetition, structure, and skilled labor. In such practices, memory is a durable configuration whose persistence depends on material continuity and embodied engagement. Pattern does not merely represent meaning; it conditions interpretation, action, trust, and even political change over time. In these contexts, form emerges through ongoing engagement between maker and material rather than being imposed symbolically—a process Tim Ingold describes as one of "correspondence" [2]. Memory, in this sense, is not stored and retrieved, but rather maintained through practice.

2.1 Ritual Stability

Ritual textiles offer a clear example of this logic. A *rushnyk* is a ceremonial cloth used across East Slavic cultures in weddings, funerals, and religious rites, where it functions as a binding and mediating object—marking thresholds, sealing social contracts, and carrying generational memory through repeated use. Meaning is carried not through isolated symbols but through compositional patterning: mirrored ends, borders, axes, and rhythmic repetition that are passed down through generations.

As Franklin Sciacca observes, even when specific imagery shifts in response to political or historical change, "the compositional structure of the rushnyk remains remarkably stable," allowing meaning to endure through pattern rather than specific iconography [3]. Memory here is enacted through handling, placement, and circulation, remaining inseparable from the material form that carries it.

2.2 Woven Logic in Computing

Early computational memory systems instantiate a related commitment to pattern. Magnetic core rope memory, key to early aerospace computing, encoded programs by physically weaving wires through magnetic cores, producing memory as a spatial configuration rather than an abstract data structure. Figure 6 shows an example of weaving computer core memory from 1956 as photographed by Ansel Adams while on assignment for IBM. Figure 7 shows an example of Magnetic Core Memory purchased by one of the authors on Ebay.

James Tomayko emphasizes that this physical permanence



Figure 4. Magnetic Memory Rushnyk, Courtesy of Yuditskaya

was central to its function: once woven, programs were "immune to corruption," capable of surviving the extreme conditions of spaceflight [4]. Reliability in this context depended directly on the stability of the woven structure itself. The program was not "soft" ware; it was hard-wired into the textile matrix of the memory module. Figure 9 shows a wiring diagram from [5] which we followed to create our memory core.

2.3 The Gendered Technician

The durability of such patterned memory depends wholly on skilled labor, and this labor has been historically gendered. Rushnyk-making positioned women as agents of symbolic inscription and community cohesion, while magnetic memory fabrication relied on women's textile expertise to produce uncorruptable computational structures.

The authority of these systems rested on embodied verification and collective care, even as the framing of this labor often obscured its technical status. Rosner et al. describe the work of core memory weavers as demanding "an extraordinary level of precision and care" [6], while Nakamura cautions that presenting electronics manufacture as mere cultural continuity masked its recognition as skilled industrial labor [7]. Across these contexts, weaving functions not as metaphor but as *techne*: knowledge produced through material engagement, repetition, and care.



Figure 5. Example of playing Magnetic Memory Rushnyk with Embroidery Hoop, Courtesy of Yuditskaya

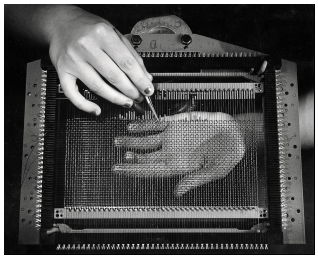


Figure 6. 'Hands weaving magnetic-core memory, IBM, Poughkeepsie, New York,' 1956. Photograph by Ansel Adams.

3. LINEAGE WITHIN WOMEN'S LABOR

The concerns regarding patterned inscription, labor, and performative activation discussed above are not abstract; they are developed within *Women's Labor*, a long-term artistic research project that creates playable instruments and textile scores from domestic and ritual materials. Across the project, everyday objects are treated as sites where gesture, labor, and social meaning are materially encoded and made available for interpretation through performance.

3.1 History of Women's Labor Project

Women's Labor is a feminist artistic project that interrogates unpaid care labor by repurposing housework tools into embedded digital musical instruments. The project transforms quotidian gestures of housework—ironing, folding, hanging laundry—into embodied musical gestures that interrogate gender inequality in the division of domestic labor.

By transforming these gestures into sites of sonic and performative expression, *Women's Labor* brings forms of labor that are historically feminized and rendered invisible into public focus. Domestic work is not treated as an abstract concept, but as a concrete, material practice whose rhythms, repetitions, and physical demands are rearticulated as musical and physical performative acts.

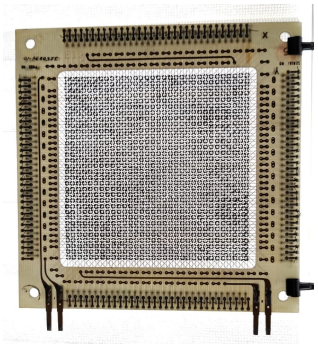


Figure 7. Front of Magnetic Core Memory used for inspiration. Courtesy of Schedel



Figure 8. Dress with Rushnyk presented at birth. Courtesy of Yuditskaya

3.2 Evolution of Textile Scores

One of the primary corollaries of the *Women's Labor* embedded instruments is the development of various types of textile scores, in which textiles function simultaneously as material artifacts, notational devices, and interfaces for performance. *Magnetic Memory Rushnyk* follows this specific line of thinking, evolving from previous iterations:

- **The Embedded Iron:** In earlier works, this instrument functioned as a sentient eye. Color and texture of fabrics were scanned and decoded into sound through inter-sensory color–sound mappings using machine learning [8].

- **Housework Lock (her) Down:** A special white apron (associated with domestic service and feminine propriety) served as a charged fabric score [9]. This unassuming apron encoded misogynistic attitudes encapsulated by nineteenth-century texts that were voiced out when played with Embedded Iron. Through performance, these attitudes were not merely represented but actively decoded into sound, fragmented, and manipulated through the granular synthesis and performer's gestures. Playing the fabric became an act of critical reading, where hidden assumptions embedded in domestic material culture were rendered audible, then disrupted.

- **Housework Commons:** The nine woven eye scores created for this piece crystallized the fabric-score approach into a set of discrete, interpretable textile artifacts. Each score is associated with a specific "attitude" of the eyeball drawn from Eleanor Georgen's 1893 manual *The Delsarte System of Physical Culture*, where rhetoric is taught to women through empowering body language—an especially pointed resiting of expressive authority given that rhetoric was historically coded as a male domain [10].

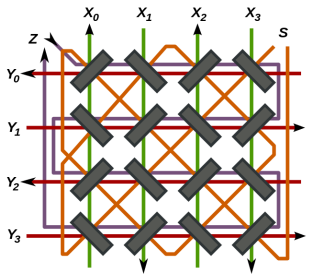


Figure 9. Diagram of a 4x4 plane of magnetic core memory. Diagram by Tetromino/Wikimedia Commons.

In performance, these scores function not as fixed notation, but as affective cues inviting gestural response. *Magnetic Memory Rushnyk* extends this method: textile form becomes a legible site where meaning is materially encoded. However, unlike previous works based on domestic objects, this piece engages a tradition with fixed meanings established by an existing community. Here, the interpretation valorizes the agency of the *creatrix*, carrying undercurrents of lineage and protective fear. The work resists the historical invisibility of domestic labor, highlighting individual agency while maintaining the material commonalities of traditional women's craft.

The project draws direct inspiration from a specific archival artifact: an embroidered cloth produced for Author 1 by her mother to mark her birth. This object creates a direct material lineage, functioning as a Perednik—etymologically "on the front"—rather than a standard rushnyk ("hand-item"). While both objects rely on identical techniques of counting threads and embedding symbolic logic, their distinction lies in their spatial activation: the rushnyk traditionally sanctifies architectural thresholds (icons, windows, corners), whereas the Perednik marks the living body itself as a sacred site [11]. We adopt this somatic orientation, treating the magnetic memory not as a static archive of data, but as a protective, resonant layer that must be navigated in close physical proximity to the performer's body.

4. MAGNETIC MEMORY RUSHNYK

Magnetic Memory Rushnyk is the newest work to emerge from this lineage. Conceived as a wall-sized textile installation, it extends the project's exploration of textile scores by making memory itself the primary site of inscription and interpretation. Oversized magnetic elements are woven into a long textile field, deliberately scaling encoding structures so that patterns normally hidden at electronic or microscopic levels become perceptible to the body. Memory is rendered as structure: something that can be seen, traced, and sonically encountered.

4.1 Material Construction

The textile incorporates bands of magnetized material woven directly into the natural fiber structure. These bands are constructed using magnetic rings and ferrous threads, woven into the warp weft of the fabric. This construction mimics the logic of magnetic core memory grids but scales them to the human hand.

By weaving these materials, the memory is physically locked into the textile matrix. The "bit" is both a voltage state and a physical presence. The visual pattern of the weave correlates to the magnetic data contained within it, allowing the performer to visually anticipate the sonic density of different regions before they are played.

4.2 Sensing and Sonification

The textile is read using a custom-built electromagnetic induction coils (hoop) held by the performers. These hoops function as pick-ups, similar to guitar pickups but designed for spatial navigation. As the performer moves the hoop over the surface of the work, the spatial variations in the magnetization of the woven bands induce small electrical currents in the coils.

This sensor data is streamed in real-time to a sound synthesis and processing environment. There changes in magnetic intensity, polarity, and the speed of the performer's gesture are mapped to synthesis parameters:

- **Flux/Intensity:** Mapped to filterxzl cutoff frequencies and amplitude, requiring the performer to maintain intimate proximity to the cloth to sustain sound.
- **Gesture Speed:** The velocity of the hoop's movement drives the density of granular synthesis textures, linking physical effort directly to sonic density.
- **Magnetic Polarity:** Distinct magnetic orientations in the weave trigger shifts in harmonic pitch sets, derived from deconstructed folk polyphony.

The mapping is deliberately coarse rather than optimized for clean data retrieval. It preserves noise, electromagnetic drift, and signal discontinuity as audible traces of the material interaction. In our construction of *Magnetic Memory Rushnyk* we followed the wiring diagram from Figure 9 and repeated it 3 times creating a standard *Tree of Life* motif rushnyk without the embroidered symbology but preserving the nodes and structure through magnet placement [3]. The colors of wire and magnet chosen are black and white referencing the figures of Dolya and Nedolya, traditional goddesses of weaving, fortune and misfortune in Slavic folklore [**?**]. Figures 1, 2, and 3 show our wiring diagram in layers for clarity. Nodes are activated by sending current through various combinations of lines on the matrix, allowing us to activate nodes in sequence or in clusters, thus creating a visual, woven score that is also an instrument as observed by Enrique Thomas in 2023 [**?**].

4.3 The Performative Experience

The installation operates as a score-instrument: a single object in which material structure simultaneously organizes interpretation and produces sound. Performers engage the textile through touch, proximity, and movement. Temporal form emerges through traversal rather than execution, as performers navigate the textile's length, density, and resistance.

In this sense, the installation functions as an "epistemic tool" for memory itself, shaping how memory is encountered and understood through interaction [12]. By enlarging and exposing the structures of encoding, the installation foregrounds memory as something that can be apprehended through bodily presence rather than inferred through abstraction. The textile wall functions as a field for navigation: movement along its length, pauses at dense regions, and retracing of paths all contribute to the unfolding of musical form.

5. CONCLUSION

Magnetic Memory Rushnyk integrates symbolic embroidery and magnetic encoding into a large-scale, performative textile installation that foregrounds materiality, labor, and legibility. By situating ritual textile practices and early computational memory within a shared framework of patterned inscription, the work advances a performative model of memory grounded in correspondence, commitment, and interpretation.

This project contributes to the field of electronic instruments by challenging the paradigm of electrically amplified interaction. Instead, it proposes a friction-based engagement where the performer must labor to retrieve sound, mirroring the labor required to weave the memory into place. By amplifying the invisible forces of magnetism through physical traversal, the system renders the abstract concept of "digital memory" into a tangible, resisting weight.

Our work serves as a critical act of recovering historical labor, drawing a direct line between the protective magic of the domestic ritualist and the precision engineering of the core memory weaver. In doing so, it suggests that the "soft" technologies of the domestic sphere were never truly distinct from the "hard" technologies of computation; both rely on the rigorous, patterned organization of matter to hold meaning against the flow of time. Memory here is not abstracted or hidden; it is encountered through touch, movement, and sound, remaining inseparable from the material and labor that give it form.

6. REFERENCES

[1] J. Ho, M. Schedel, and S. Yuditskaya, "Housework Commons," in *Proceedings of NIME 2025*, 2025.

[2] T. Ingold, *Making: Anthropology, Archaeology, Art and Architecture*. Routledge, 2013.

[3] F. Sciacca, "Ukrainian Rushnyky: Binding Amulets and Magical Talismans in the Modern Period," *Folklorica*, vol. 17, pp. 1–35, 2013.

[4] J. Tomayko, *Computers Take Flight: A History of NASA's Pioneering Digital Fly-By-Wire Project*. NASA, 2000.

[5] Kaushik Patowary. (2020, Feb.) That Time When Computer Memory Was Measured in Kilograms. Accessed: 2026-01-04. [Online]. Available: https://www.amusingplanet.com/2020/02/that-time-when-computer-memory-was.html

[6] D. K. Rosner, S. Shorey, B. Craft, and H. Remick, "Making Core Memory: Design Inquiry into Gendered Legacies of Engineering and Craftwork," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2018.

[7] L. Nakamura, “Indigenous Circuits: Navajo Women and the Racialization of Early Electronic Manufacture,” *American Quarterly*, vol. 66, no. 4, pp. 919–941, 2014.

[8] J. Ho, M. Schedel, and M. Blessing, “Revaluing Women’s Labor through Material Engagement with Musical Instruments Built from Domestic Tools,” in *Proceedings of ISEA 2020*, 2020.

[9] J. Ho and M. Schedel, “Housework Lock (her) Down,” *Women and Music*, vol. 25, no. 1, pp. 26–32, 2021.

[10] E. Georgen, *The Delsarte System of Physical Culture*. New York: The Butterick Publishing Company, 1893.

[11] L. P. Benetti and M. S. Barros de Held, “The Slav Influence on the Russian Folk Dress,” in *Textiles, Identity and Innovation: In Touch*, 1st ed. CRC Press, 2020.

[12] T. Magnusson, “Of Epistemic Tools: Musical Instruments as Cognitive Extensions,” in *Proceedings of the International Computer Music Conference*, 2009.

Vibrato Matching for Modulation Control and Blending in Sound Mixtures

Jeremy Hyrkas
University of California San Diego
jhyrkas@ucsd.edu

ABSTRACT

In sound mixtures of more than one musical source, different vibrato patterns act as a cue that multiple sources are present for both human listeners and source separation algorithms. Matching the vibrato patterns of the signals in the mixture reduces the perception of multiple sources, particularly when the sources play in unison. This work introduces the vibrato matching algorithm, which first suppresses vibrato in a target signal and then transfers vibrato from a source signal to the target. An existing vibrato suppression algorithm is combined with a new algorithm for vibrato transfer, which imparts frequency modulation and amplitude modulation to the harmonics of the target signal, and amplitude modulation onto the spectral envelope of the non-harmonic residual component. Examples demonstrate the algorithm's utility as a vibrato control mechanism and as a tool for blending sound sources. Matching vibrato degrades the performance of source separation algorithms, suggesting a similar degradation in listeners ability to detect the presence of multiple sources.

1. INTRODUCTION

Vibrato is a musical technique in which the pitch of a sustained note is modulated, and is often accompanied by a modulation in sound level. Signals with natural vibrato therefore feature both frequency and amplitude modulation (FM/AM). The presence of different vibrato patterns improves the ability of listeners to count the number of instruments in a mix, even when the instruments play in unison [1, 2]. This finding motivated the development of unison source separation algorithms that use modulation patterns to isolate sources [3, 4].

Given its perceptual salience, precise control of a signal's vibrato can be useful. Vibrato can be manipulated using sinusoidal modeling [5], where the AM and FM of harmonic partials and AM in the spectral envelope are analyzed and altered prior to resynthesis. Recent efforts focus on manipulating signals directly, either to remove existing vibrato [6] (*vibrato suppression*) or to apply vibrato patterns from one signal to another [7, 8] (*vibrato transfer*).

This study combines and expands these approaches to introduce the *vibrato matching* algorithm. Vibrato suppression [6] is reviewed in Section 2. Section 3 proposes an update to the vibrato transfer algorithm [8] that expands its AM transfer. Vibrato matching, the combination of vibrato suppression and transfer, is evaluated in Section 4 as both a means to change the vibrato pattern of a signal and as a tool for matching vibrato patterns in sound mixtures. It is also shown that by vibrato matching reduces the quality of unison source separation. This reduction suggests that matching vibrato patterns of unison signals can be used as a creative tool to obscure the presence of multiple instruments in a sound mixture.

Copyright: ©2026 Jeremy Hyrkas et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License 3.0 Unported, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

2. VIBRATO SUPPRESSION

The goal of vibrato matching is to analyze a source signal with vibrato $s(n)$ and to impart its modulation patterns to a target signal $t(n)$. Any existing vibrato in $t(n)$ should be suppressed before applying new vibrato to avoid interference of the modulation patterns. Previously proposed vibrato suppression methods are briefly reviewed below and used as a pre-processing step to vibrato transfer.

2.1 Vibrato Filter

Previous works on vibrato control use a *vibrato filter* [5, 6] $v_f(n)$ in various stages of analysis and synthesis. The vibrato filter is a high-order FIR lowpass filter with a cutoff frequency lower than typical vibrato rates (i.e., 2 Hz). The filter is applied using forward-backward filtering to create a zero-phase filter. This filter is used on frequency or amplitude envelopes to suppress variations caused by vibrato.

2.2 Suppressing FM caused by vibrato

The signal $\omega_i^t(n)$ contains the instantaneous fundamental frequency of $t(n)$. $\omega_i^t(n)$ can be derived using peak-picking in the short-time Fourier transform (STFT) of $t(n)$ or from the analytic signal of the first harmonic isolated using a bandpass filter [7].

$$\overline{\omega}_i^t(n) = \omega_i^t(n) \circledast v_f(n) \tag{1}$$

is the instantaneous fundamental frequency of $t(n)$ after removing frequency deviations caused by vibrato [6].

Using these signals, a time-varying demodulating delay

$$D_d(n) = \sum_{l=0}^{n} \left(\frac{\omega_i^t(l)}{\overline{\omega}_i^t(l)} - 1 \right) \tag{2}$$

is constructed. The signal $t\big(n - D_d(n)\big)$ contains very little pitch modulation when $t(n)$ contains light vibrato. For heavier vibrato extent, the *recursive delay*

$$D_{dd}(n) = D_d\big(n - D_d(n)\big) \tag{3}$$

is more effective. The signal

$$\widehat{t}(n) = t\big(n - D_{dd}(n)\big) \tag{4}$$

contains no audible FM, but may still contain AM [7].

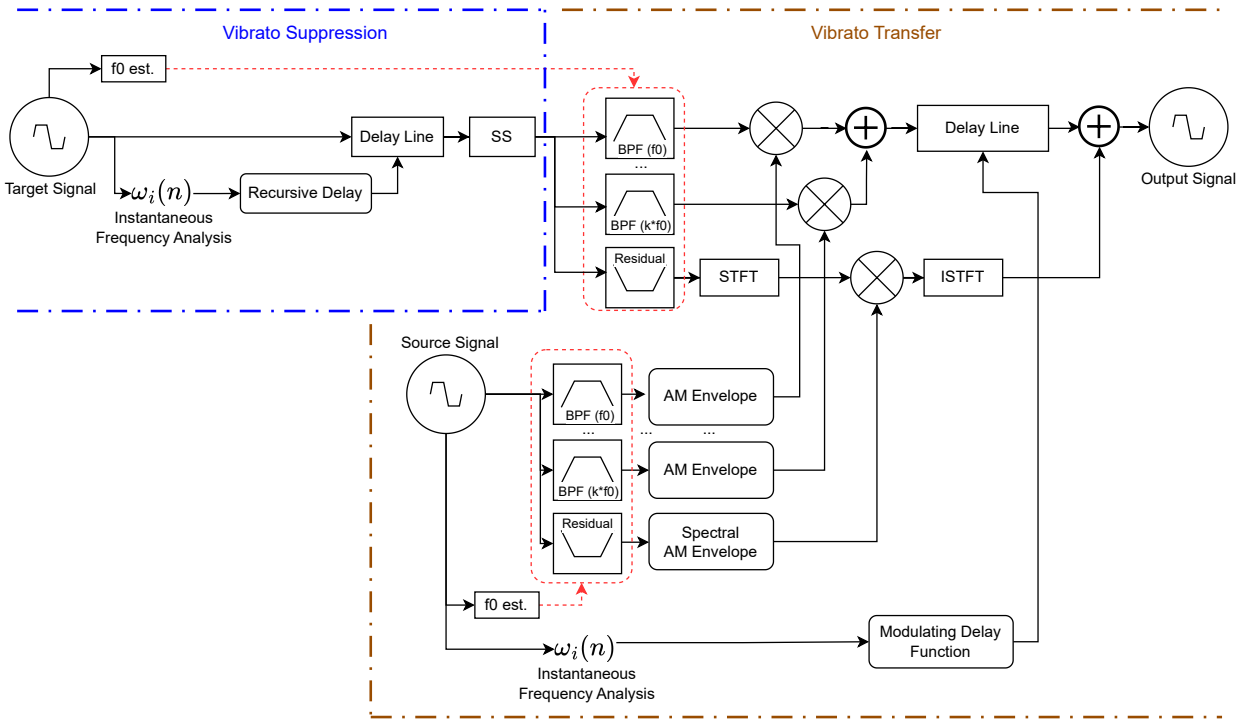


Figure 1. Signal flow for the vibrato matching algorithm. A target signal $t(n)$ has its original vibrato suppressed and is then modulated using AM and FM patterns from a vibrato source signal $s(n)$.

2.3 Suppressing AM caused by vibrato

To suppress AM in $\widehat{t}(n)$, *spectrogram smoothing* (SS) [6] is performed.

$$\widehat{T}(\omega, m) = \mathrm{STFT}\big(\widehat{t}(n)\big) \tag{5}$$

is used to derive

$$E_T(\omega, m) = |\widehat{T}(\omega, m)| \ , \tag{6}$$

the magnitude spectrogram of $\widehat{t}(n)$. A vibrato filter is applied along the m axis for each frequency ω to derive

$$\overline{E}_T(\omega, m) = E_T(\omega, m) \circledast v_f(m) \ , \tag{7}$$

the magnitude spectrogram without vibrato-induced AM. Finally, the ratio of these spectrograms is used to derive

$$\overline{t}(n) = \mathrm{ISTFT}\Big((\widehat{T}(\omega, m) \cdot \frac{\overline{E}_T(\omega, m)}{E_T(\omega, m)}\Big) \ , \tag{8}$$

a version of the target signal with no vibrato-induced FM or AM. Vibrato suppression using the recursive delay and SS is depicted in the upper left portion of Figure 1.

3. VIBRATO TRANSFER

The FM patterns from a source signal with vibrato $s(n)$ can be transferred to the vibrato-suppressed target signal $\overline{t}(n)$ using a time-varying delay line. Previous work performed this transfer in real-time with the addition of a simple AM envelope [8]. In real acoustic vibrato, AM patterns are often different in each harmonic [9] and across the spectral envelope of the non-harmonic *residual* portion of the signal [5]. Realistic vibrato transfer therefore requires modulating the amplitudes and frequencies of the harmonics of $\overline{t}(n)$ and applying AM to the residual signal.

3.1 Transferring AM to the harmonics

Vibrato transfer begins by deriving AM envelopes from the harmonics of $s(n)$ and applying them to corresponding harmonics in $\overline{t}(n)$. Using the fundamental frequency f_0^s of $s(n)$ [1], the kth harmonic of $s(n)$ is isolated using

$$h_k^s(n) = s(n) \circledast \mathrm{BP}_k^s(n) \ , \tag{9}$$

where $\mathrm{BP}_k^s(n)$ is a Butterworth bandpass filter centered at frequency $k \cdot f_0^s$.

The root-mean-square (RMS) envelope of each harmonic is computed as

$$E_k^s(n) = \mathrm{RMS}\big(h_k^s(n)\big) \tag{10}$$

over sliding windows and interpolated to signal length.

$$\overline{E}_k^s(n) = E_k^s(n) \circledast v_f(n) \tag{11}$$

is the envelope of $h_k^s(n)$ without vibrato-induced AM.

$$\mathrm{AM}_k(n) = \frac{E_k^s(n)}{\overline{E}_k^s(n)} \tag{12}$$

yields the AM envelope of the kth harmonic (i.e., the time-varying amplitude such that $E_k^s(n) = \mathrm{AM}_k(n) \cdot \overline{E}_k^s(n)$). It is useful to set $\mathrm{AM}_k(n) = 1$ until the first peak in $\overline{E}_k^s(n)$ so that the signal's attack is preserved. Figure 2 shows an AM envelope derived from a corresponding RMS envelope.

With AM envelopes derived for each harmonic in $s(n)$, the harmonics of $\overline{t}(n)$ are now isolated as

$$h_k^t(n) = \overline{t}(n) \circledast \mathrm{BP}_k^t(n) \ , \tag{13}$$

[1] Known a priori or determined using an f_0 detection algorithm

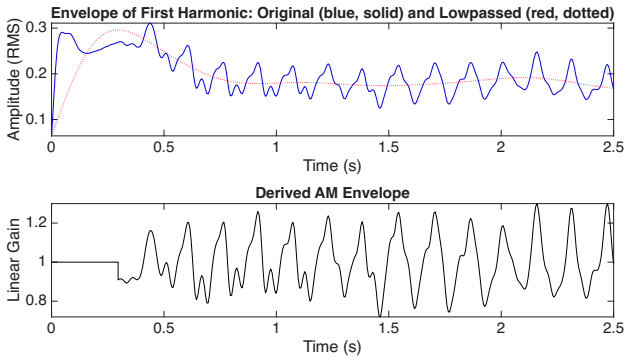


Figure 2. Deriving an AM envelope from an RMS envelope.

where $\mathrm{BP}_k^t(n)$ is a Butterworth bandpass filter centered at frequency $k \cdot f_0^t$. Each harmonic is amplitude modulated

$$\widetilde{h_k^t}(n) = h_k^t(n) \cdot \mathrm{AM}_k(n) \ , \tag{14}$$

after which the amplitude modulated harmonics are summed to create

$$\widetilde{h_t}(n) = \sum^{k} \widetilde{h_k^t}(n) \ . \tag{15}$$

3.2 Transferring FM to the harmonics

As in Section 2.2, $\omega_i^s(n)$ contains the instantaneous fundamental frequency of $s(n)$ and

$$\overline{\omega}_i^s(n) = \omega_i^s(n) \circledast v_f(n) \tag{16}$$

is the instantaneous fundamental frequency without vibrato.

$$D_m(n) = \sum_{l=0}^{n} \left(1 - \frac{\omega_i^s(l)}{\overline{\omega}_i^s(l)}\right) \tag{17}$$

is a time-varying modulating delay that causes the same FM pattern present in $s(n)$ [6]. Using this delay function on $\widetilde{h_t}$ yields

$$h_t^*(n) = \widetilde{h_t}\big(n - D_m(n)\big) \ , \tag{18}$$

a signal comprised of the harmonics of $t(n)$ with identical FM and AM pattern as the harmonics of $s(n)$.

3.3 Transferring AM to the residual signal

AM in the non-harmonic residual in signals with vibrato can differ across the frequency spectrum [5], so it is best imparted in a frequency-dependent manner.

The residual signal of $s(n)$ is captured as

$$r_s(n) = s(n) \circledast \mathrm{BS}_1^s(n) \circledast \mathrm{BS}_2^s(n) \circledast \ldots \circledast \mathrm{BS}_k^s(n) \ , \tag{19}$$

where each $\mathrm{BS}_k^s(n)$ is a Butterworth bandstop filter centered at frequency $k \cdot f_0^s$. $\mathrm{BS}_k^s(n)$ rejects the kth harmonic in $s(n)$ and is complementary to $\mathrm{BP}_k^s(n)$ in Section 3.1.

The STFT of the residual signal

$$R_s(\omega, m) = \mathrm{STFT}\big(\mathrm{r_s(n)}\big) \tag{20}$$

is divided into d equally spaced and non-overlapping frequency regions across ω. For each frame m, the frequency with maximum magnitude in each region

$$\omega_d^*(m) = \underset{\omega * \in [\omega_d, \omega_{d+1})}{\arg\max} \ |R_s(\omega^*, m)| \tag{21}$$

is used to create the spectral envelope

$$\mathrm{SE}_{\mathrm{dB}}(d, m) = 20 \log_{10}\Big(|R_s\big(\omega_d^*(m), m\big)|\Big) \tag{22}$$

in decibels. A vibrato filter is applied across the m dimension for each region to create

$$\overline{\mathrm{SE}}_{\mathrm{dB}}(d, m) = \mathrm{SE}_{\mathrm{dB}} \circledast v_f(m) \ , \tag{23}$$

the spectral envelope without vibrato-induced AM.

The spectral envelopes are converted to linear amplitude

$$\mathrm{SE}_s(d, m) = 10^{\mathrm{SE}_{\mathrm{dB}}(d,m) \cdot \frac{1}{20}} \tag{24}$$

$$\overline{\mathrm{SE}}_s(d, m) = 10^{\overline{\mathrm{SE}}_{\mathrm{dB}}(d,m) \cdot \frac{1}{20}} \ , \tag{25}$$

and a spectral AM envelope is constructed as

$$\mathrm{AM}_s(d, m) = \frac{\mathrm{SE}_s(d, m)}{\overline{\mathrm{SE}}_s(d, m)}. \tag{26}$$

Finally, $\mathrm{AM}_s(d, m)$ is linearly interpolated across d back to the original cardinality of $R_s(\omega, m)$ to create $\mathrm{AM}_s(\omega, m)$.

The residual of $\bar{t}(n)$ is isolated using

$$r_t(n) = \bar{t}(n) \circledast \mathrm{BS}_1^t(n) \circledast \mathrm{BS}_2^t(n) \circledast \ldots \circledast \mathrm{BS}_k^s t(n) \ , \tag{27}$$

where each $\mathrm{BS}_k^t(n)$ is a Butterworth bandstop filter centered at frequency $k \cdot f_0^t$. The spectral envelope of $r_t(n)$ is modulated using the spectral AM envelope as

$$r_t^* = \mathrm{ISTFT}\Big(\mathrm{STFT}\big(r_t(n)\big) \cdot \mathrm{AM}_s(\omega, m)\Big) \ . \tag{28}$$

Finally, the modulated residual is added with the modulated harmonics to create

$$t^*(n) = h_t^*(n) + r_t^* \ , \tag{29}$$

the target $t(n)$ with its vibrato matched to the source $s(n)$. The signal flow of the vibrato transfer portion of the vibrato matching algorithm is depicted on the bottom and right portions of Figure 1.

4. APPLICATIONS OF VIBRATO MATCHING

Potential applications of vibrato matching are showcased here, both as a tool for modifying the vibrato of a target signal and as a method of controlling modulation in mixtures of signals. Accompanying audio, as well as supplemental additional figures and sound examples from each section, are available on the supplemental website.

4.1 Modifying Vibrato in a Target Signal

Vibrato matching allows a target signal's vibrato to be precisely matched to a source. This modification may be useful when the target's vibrato is unsatisfactory, which could be caused by a poor recorded performance or the use of a synthesis technique that lacks realistic vibrato.

Figure 3 shows vibrato matching applied to vocal signals, in which the source signal has a higher vibrato extent than the target. The rate and depth of vibrato in the target are matched to those in the source. Figure 4 depicts vibrato matching between instrument types. A saxophone signal has its vibrato matched to that of a violin, creating the sound of a woodwind instrument with string-like vibrato.

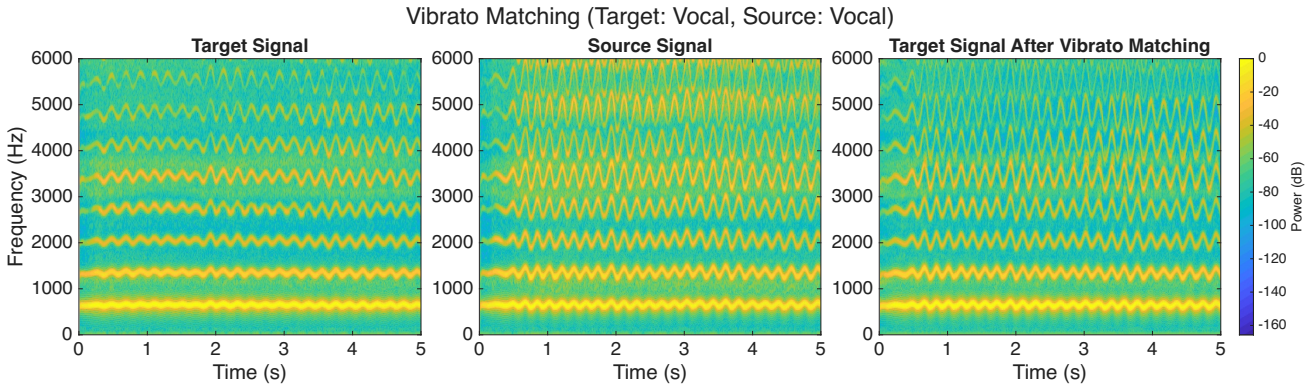


Figure 3. Spectrograms of a vocal with vibrato (left), a vocal with different vibrato (middle), and the left signal after its vibrato was suppressed and vibrato from the middle signal was transferred onto it (right).

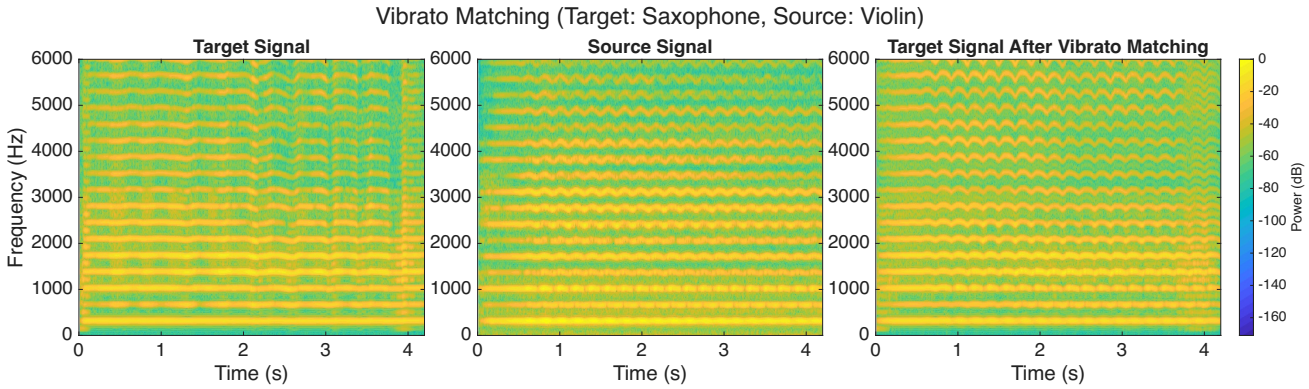


Figure 4. Spectrograms of a saxophone with vibrato (left), a violin with vibrato (middle), and the saxophone after its vibrato was suppressed and vibrato from the violin was transferred onto it (right).

4.2 Modulation Control in Sound Mixtures

Vibrato matching can be used when mixing signals with vibrato as both an engineering tool and a creative audio effect. Figure 5 demonstrates a mix of the source and target vocal signals from Figure 3. In the top spectrogram, partials from each source are visible due to their different modulation patterns. Sonically, the mix contains rough beating as the signals fall in and out of tune with each other.

After vibrato matching, the spectrogram appears to contain only one signal even though two are present. The vibrato matched mix sounds similar to a double tracked vocal performance with no inconsistencies in vibrato pattern between takes. This example demonstrates vibrato matching's usefulness as an audio engineering tool when double tracking performances with vibrato.

Figure 6 depicts a more creative approach. The vibrato pattern from a flute is transferred to a vocal signal, imparting AM in the residual noise that is uncommon in vocal vibrato. In this example, vibrato matching is used as a sound design tool to influence a listener's perception of sonic grouping. When the vibrato matched signals are mixed, the signals modulate parallel, creating a sense of shared movement [2] even though two sources are perceived due to their different pitches and timbre.

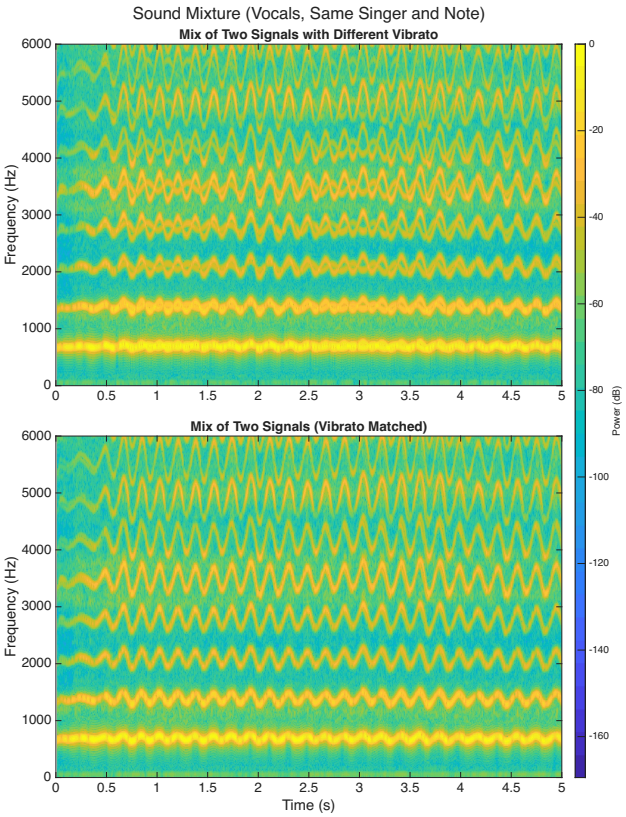


Figure 5. Spectrograms of two vocal signals in unison, before and after matching vibrato patterns.

[2] Grouping elements based on the perception of shared motion is described by the Gestalt principle of Common Fate. The Common Fate Transform used in unison source separation is named for this principle.

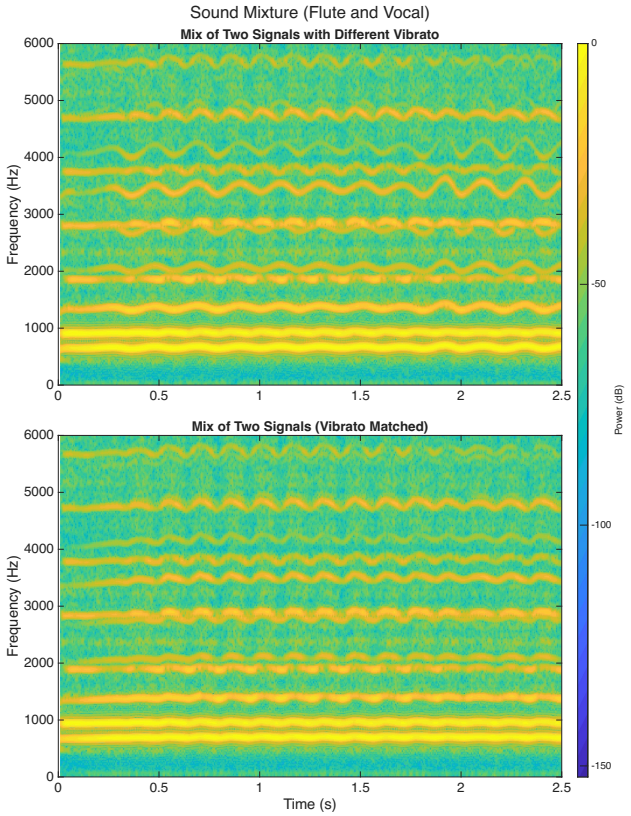


Figure 6. Spectrograms of a vocal signal and a flute signal separated by a perfect fifth, before and after matching vibrato patterns.

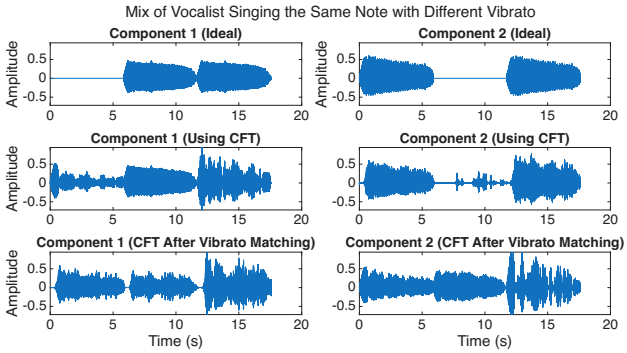


Figure 7. Unison source separation on a mixture of two vocal signals, before and after matching vibrato patterns.

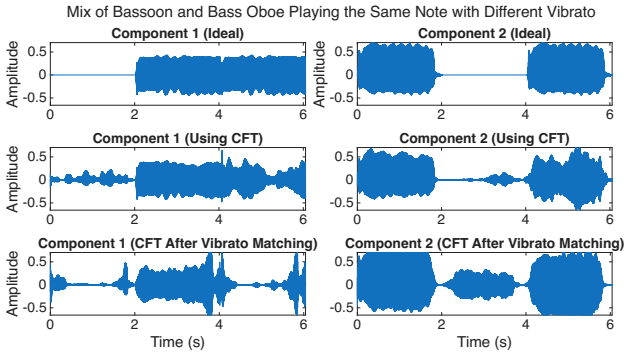


Figure 8. Unison source separation on a mixture of bass oboe and bassoon, before and after matching vibrato patterns.

4.3 Sound Blending and Source Separation

Different vibrato patterns aid in detecting the presence of multiple sources in a mix for both listeners and unison source separation algorithms. If matching the vibrato patterns of unison signals degrades the quality of source separation, it stands to reason that it may also degrade a listener's ability to perceive multiple instruments. Vibrato matching could therefore be used as a sound blending tool in which multiple instrument sounds are mixed and perceived as one sound source.

To test the effect of vibrato matching on source detection, the source separation algorithm based on the Common Fate Transform (CFT) [4] was used on several examples of two unison signals before and after vibrato matching. This algorithm was previously shown to be highly effective at separating unison signals based on their FM and AM patterns. Each signal analyzed by the algorithm contains three events: one source in isolation, then the second source in isolation, and finally both sources simultaneously. This sequencing follows examples from the CFT's original evaluation in an attempt to maximize the separation quality.

Figure 7 shows source separation of the vocal mixes from Figure 5. The mixes feature a vocalist singing the same note with different vibrato rates. The ideal outputs of the source separation algorithm are shown in the top row. The middle row demonstrates the effectiveness of source source separation using the CFT. The first source has noticeable leakage, while the second source is well isolated. The bottom row shows that vibrato matching completely degrades the CFT's ability to separate these sources. This result is congruent with the bottom spectrogram in Figure 7, in which the two sources visually appear as one.

A mix of bassoon and bass oboe playing in unison is shown in Figure 8. The bottom row again shows that vibrato matching reduces the effectiveness of source separation when compared to the original mix in the middle row. The second source is split between the two separated components (see the middle third of each component). It is clear from the final third of each component that the second component extracted from the vibrato matched mix contains the majority of each source's signal instead of only one source. This indicates that the separation algorithm cannot unmix the sources even when primed with each source in isolation.

Spectral features other than modulation play a role in unison source separation. Nevertheless, Figures 7 and 8 indicate a reduced ability for a machine (and perhaps a human) listener to detect the presence of multiple sources in a unison mix after vibrato patterns in each source are matched.

5. CONCLUSION

Vibrato matching can be used to first suppress existing vibrato in a signal and then transfer vibrato from another signal. Previously proposed methods for vibrato suppression are sufficient, but extensions to previous methods for vibrato transfer are necessary for vibrato matching. While the prior vibrato transfer algorithm uses a simple AM envelope that is applied to the entire signal, in vibrato matching different AM patterns are applied per-harmonic and across the spectral envelope of the residual signal. This method of AM transfer, while more complex, is consistent with the

complex nature of AM in natural vibrato [5, 9]. The source code for vibrato matching and original audio files are available online on the supplemental website [3] , as well as additional figures and audio examples from Sections 4.1– 4.3.

Vibrato matching can be used as a method for modifying vibrato in a signal in isolation. Use cases include changing existing vibrato by example, transferring vibrato between instrument classes to create instrument sounds not possible acoustically, or augmenting physical modeling synthesis. Matching modulation patterns between sources also offers unique creative opportunities when sources are mixed together. While listening tests have not yet been conducted to confirm that using vibrato matching on unison signals reduces a listener's ability to perceive multiple sources, source separation algorithms inspired by similar listening tests show a reduced capacity for separation after modulation patterns are matched. Mixes of vibrato matched signals can be used to create hybrid instrument sounds from recordings of different instrument types, or for creative control of the perception of shared movement in a mix.

Acknowledgments

The author would like to thank Professors Tamara Smyth, Miller Puckette, and Tom Erbe for their invaluable insights. Additional thanks to the DMA students in Performance at the Music Department of the University of California San Diego who generously provided examples of their playing with and without vibrato.

[3] `https://jeremyhyrkas.com/ICMC2026`

6. REFERENCES

[1] F.-R. Stöter, M. Schoeffler, B. Edler, and J. Herre, "Human ability of counting the number of instruments in polyphonic music," in *Proceedings of Meetings on Acoustics*, vol. 19, no. 1. Acoustical Society of America, 2013.

[2] M. Schoeffler, F.-R. Stöter, H. Bayerlein, B. Edler, and J. Herre, "An Experiment about Estimating the Number of Instruments in Polyphonic Music: A Comparison Between Internet and Laboratory Results." in *Proceedings of the 14th International Society for Music Information Retrieval Conference (ISMIR)*, 2013.

[3] F.-R. Stöter, S. Bayer, and B. Edler, "Unison Source Separation." in *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, 2014.

[4] F.-R. Stöter, A. Liutkus, R. Badeau, B. Edler, and P. Magron, "Common fate model for unison source separation," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016.

[5] A. Roebel, S. Maller, and J. Contreras, "Transforming vibrato extent in monophonic sounds," in *Proceedings of International Conference on Digital Audio Effects (DAFx)*, 2011.

[6] J. Hyrkas and T. Smyth, "Vibrato suppression by time-varying delay and spectral magnitude demodulation," in *Proceedings of Meetings on Acoustics*, vol. 58, no. 1. Acoustical Society of America, 2025.

[7] ——, "On Vibrato and Frequency (De)Modulation in Musical Sounds," in *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, 2024.

[8] J. Hyrkas, "Real-time implementation of vibrato transfer as an audio effect," in *Proceedings of the International Computer Music Conference (ICMC)*, 2025.

[9] M. Zhang, M. Bocko, and J. Beauchamp, "Measurement and analysis of musical vibrato parameters," in *Proceedings of Meetings on Acoustics*, vol. 23, no. 1. Acoustical Society of America, 2015.

Building Loopers

A Pedagogical Framework for Teaching Creative Software Design
Through Iterative Tool Construction in Max, gen~, and RNBO

Jeff Kaiser
University of Central Missouri
jkaiser@ucmo.edu

Gregory Taylor
Independent
gtaylor@rtqe.net

ABSTRACT

This paper introduces the ideas behind our open-access project "Building Live Loopers in Max."[1] The project presents a hybrid pedagogical and technical framework in which students learn signal processing concepts by constructing live-looping tools in Max, gen~, and RNBO. By engaging with buffer operations, timing structures, playback manipulation, and parameter mapping, students develop technical fluency and musical understanding simultaneously. We introduce a sequence of modular, step-by-step looper designs, a color-coded instructional method for visualizing patcher development, and a cross-environment workflow that reinforces transferable programming habits.

Our coursework is designed to be sufficiently open-ended that students, while grounded in familiar musical contexts, are encouraged to exercise curiosity and explore creative directions beyond the methods presented. Drawing on Dehaene's work on curiosity and Eagleman's writing on relevance, the design aims to engage intrinsic motivation and support students in forming novel connections and actively experimenting with musical ideas. This approach positions looper construction as a bridge between creative music-making and computational thinking, supporting both performance and pedagogical outcomes.

1. INTRODUCTION

Live looping is a widely adopted musical practice spanning experimental, classical, and popular music. Historically, techniques such as Terry Riley's *Time-Lag Accumulator*, Eno and Fripp's *Frippertronics*, and Pierre Schaeffer's tape-based experiments have established repetition (through the use of loops) as a generator of musical structure. Looping has also been used by a variety of artists, including Alvin Lucier, Pauline Oliveros, Gordon Mumma, Steve Reich, and David Torn, as well as Laurie Anderson, Pamela Z, and William Basinski. Looping has found love in popular music among artists like KT Tunstall, Imogen Heap, Reggie Watts, Ed Sheeran, St. Vincent, and many more. Modern software environments have expanded historical practices into flexible, real-time systems capable of complex signal processing manipulations.

While looping is pervasive in performance, its pedagogical potential remains underexplored. Frequently, computer-music education presents signal processing abstractly, or students encounter loopers as pre-built devices without insight into their underlying mechanisms.

We propose that building loopers—rather than merely using them—offers a robust pedagogical framework, in which students experience core signal processing concepts directly, while simultaneously developing musical thinking, strengthening their patching fluency, and cultivating thoughtful UI design.

To help achieve these goals, this project introduces a modular, cross-environment looper curriculum that integrates Max, gen~, and RNBO, emphasizing iterative design, conceptual reinforcement, and creative experimentation.

2. BACKGROUND AND RELATED WORK

2.1 Musical Repetition and Looping

Repetition is a central component of music cognition and the formation of musical expectations (Margulis 2013; Huron 2013; Nettl 2015; Patel 2008). Repetition lies at the heart of many musical forms, including rounds in folk music, call-and-response in blues and gospel, sonatas in classical music, trance-based musics, and minimalism, illustrating how repeated events generate structure and anticipation. Technology-based looping systems formalize this process: they capture sonic material, replay it, add to it, and manipulate it in time, producing patterns that create expectations, or even play against them.

Historically, early tape-based systems enabled performers to explore tape loops with feedback, overdubbing, and layering. Now, digital environments, including Max, gen~, and RNBO, enable more precise control of timing, creative overdubbing/playback methods, and flexible transformations, such as time-stretching, reversal, and sonic modulation.

2.2 Patching Pedagogy

Teaching any type of programming environment, whether visual or text-based, risks emphasizing theoretical derivations over creative sonic feedback, leaving students with abstract knowledge that may be disconnected from musi-

Copyright: © 2026 Jeff Kaiser and Gregory Taylor. This is an open-access article distributed under the terms of the Creative Commons Attribution License 3.0 Unported, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

[1] The complete open-access project, including the manuscript-length text and all step-by-step patchers, is freely available at: http://technophony.com/loop/

cal practice. While numerous papers have explored patch-based instruction in Max and Pd, highlighting iterative experimentation and project-based learning, few studies position looping as a core teaching element. By building looper patchers incrementally, students encounter signal processing concepts in musical, applied contexts, bridging technical understanding with musical thinking.

2.3 Iterative and Cross-Environment Approaches

Cross-environment workflows, where students recreate similar patchers in Max, gen~, and RNBO, expose learners to thinking and considering differences in execution models, signal versus message rates, and environment-specific optimizations. Iterative construction, accompanied by visual cues for newly introduced objects, reinforces pattern recognition and procedural thinking, echoing the principles of musical repetition that are held dear by loopers.

3. PEDAGOGICAL FRAMEWORK

Our framework is structured around three key principles:

3.1 Iterative Construction

Students follow a patch-along model, incrementally building each looper in different environments. New components in each patcher are highlighted with a color-coded system that provides a navigational aid as successive tutorials introduce greater architectural and procedural complexity.

- **Red**: Audio-processing objects/operators
- **Yellow**: Timing and scheduling
- **Blue**: Adjustable parameters
- **Gray**: Previously established elements

This approach makes the changes in the patcher easy to follow and encourages students to retrace steps when debugging or modifying the patcher.

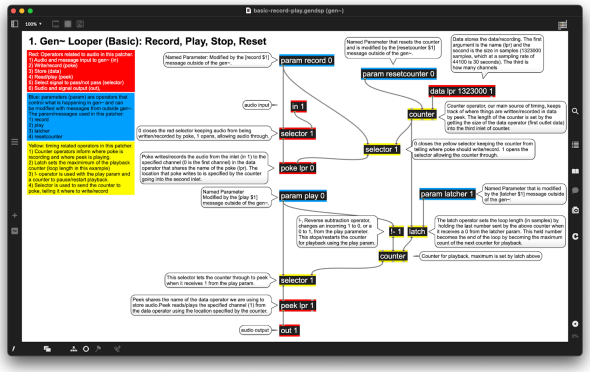


Figure 1: Step 1 in our first gen~ looper.

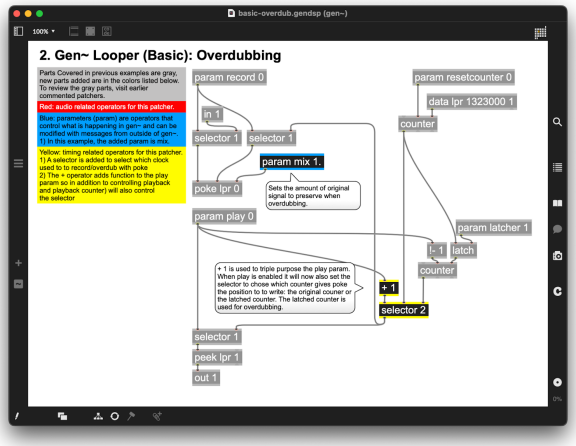


Figure 2: When we add overdubbing, previously added operators from step 1 are greyed out, bringing focus to the new operators being added.

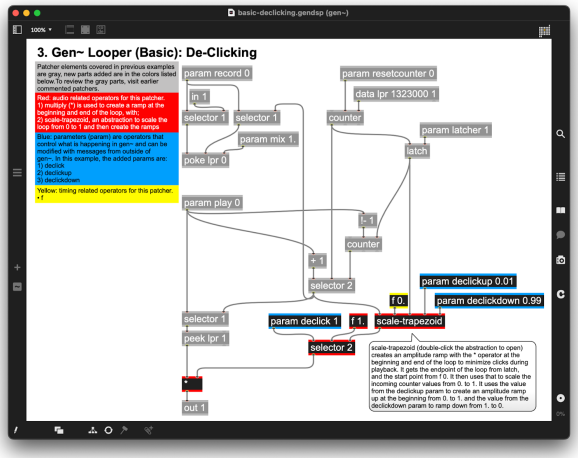


Figure 3: We add a form of declicking, greying out operators in step 1 and 2.

3.2 Cross-Environment Reinforcement: Repetition as Cognitive Scaffold

After mastering the creation of basic loopers in Max, students reconstruct the same conceptual architecture in gen~ and RNBO, using the affordances specific to those environments while critically engaging with the constraints, differences, challenges, and benefits that each environment offers. This process highlights differences in:

- Signal vs. message rate operations
- Memory access and buffer management
- Parameter mapping strategies
- Performance optimization
- Portability and exportability of the patcher (Raspberry Pi, VST/AU plugins, Max externals, et cetera).

Such iterative, cross-environment patching strengthens conceptual knowledge, working to ensure that learners develop robust technical models applicable beyond a single software environment and cultivate programming

habits—all while reflecting on the tools we think (and create) with (Turkle, 2011).

3.3 Curiosity

Our goal is to design the coursework to be sufficiently open-ended that students, while grounded in familiar musical contexts, are encouraged to exercise curiosity and explore creative directions beyond the methods presented in this project. Drawing on Stanislas Dehaene's writing on learning and the value of curiosity, as well as David Eagleman's writing on relevance, we aim to engage students' intrinsic motivation and curiosity, allowing them to form novel connections and experiment actively with musical ideas (Dehaene 2020; Eagleman 2020). In practice, the approach of this project supports a flipped classroom model, where students acquire technical knowledge outside the classroom and bring project ideas for collaborative exploration, error correction, and sharing within the classroom environment.

4. CURRICULUM DESIGN AND TECHNICAL IMPLEMENTATION

Our looper building curriculum consists of three modular stages, each implemented with multiple fully functional patchers.

4.1 Foundational Loopers (Max)

- Mimics tape-based looping with fixed-length tapin~/tapout~
  - Basic control: length, record, overdub, playback, feedback control, mixing, clear
  - Tape emulation (degradation, pitch shifting) and other signal processing techniques are introduced into the feedback loop. (These processes can be abstracted and used later in other contexts.)
- Multitrack Looper, with buffer~
  - Basic control: variable length, record, overdub, playback, feedback control, mixing, clear
  - Adds speed, direction,
  - Develop synchronous and asynchronous tracks for looping
  - Amplitude-shaping and windowing to declick
  - Creating abstractions

4.2 gen~

- Overdubbing looper
  - Contains elements of a tape-style looper, featuring multitrack and overdub capabilities.
  - Basic control: variable length, record, overdub, playback, feedback control, mixing, clear
  - Can be solely overdub or multitrack, or combine both
  - Adds declicking, shortening/dividing the loop, playback speed, playback direction, auto-insert, and randomization of playback
- Varispeed overdubbing looper
  - Developed with Graham Wakefield
  - Unique looper in the project, introduces codebox to address read/play head location problems when overdubbing with varispeed and direction differences

4.3 RNBO

- Tape-style Looper
  - Similar to Foundational Looper in Max, but uses delay~/feedback~
- Multitrack Looper
- Overdubbing Looper
- Embedding gen~ options in RNBO
- Exporting of patchers for plugin or embedded use

Each stage, with the exception of the varispeed gen~ looper, builds upon the previous one, reinforcing both technical concepts and creative applications, while expanding on the strengths and possibilities inherent in each environment.

5. EVALUATION

Our evaluation is based on classroom observations conducted with students enrolled in an introductory Max course within a university-level music technology program:

- **Engagement**: Students were highly engaged. Looping was enjoyable and encouraged curiosity, as well as the further exploration of creative possibilities (Dehaene, 2020; Bonawitz et al., 2011). Students enjoyed discovering the glitches inherent in the limitations of each environment and using them to creative ends (Cascone 2000).
- **Fluency**: Students demonstrated improved ability to reason and transfer knowledge from easier basic loopers into more complex patchers. For example, from tapin~/tapout~ to exploring buffer~ behavior, message-based to signal-based timing, sonic manipulation, and parameter mapping.
- **Creative Application**: Many students adapted loopers for their own performance projects, developing personal designs with their own routing structures, modulation, and highly personalized UIs.
- **Challenges**: For some students, negative metacognition (e.g., '*I don't know how to code'*) created a persistent perception of a steep learning curve even before the course began. Although the visual patching environment partially alleviated these concerns, the underlying issue persisted for some students. Additional challenges arose from the one-semester course structure and the time required to engage meaningfully with the increased complexity introduced when transitioning patchers to gen~ and RNBO.

6. DISCUSSION

The hybrid framework we use integrates musical thinking with the development of patching skills. Loopers, in this sense, serve as both pedagogical objects and performance tools, illustrating how iterative design, repetition, and cross-environment exploration reinforce learning. By situating signal processing principles within creative practice, students gain a deeper, more tangible understanding of abstract concepts, facilitating retention and applicability.

7. CONCLUSIONS

This project is about live looping—a musical practice that we realize attracts fierce advocates and determined skeptics in equal proportions.

For skeptics, looping can seem reductive: a practice that risks flattening musical nuance, encouraging predictability, and linear development. Some might see it as a shortcut that substitutes mechanical repetition for musical development, while others critique what they perceive as aesthetic clichés that have grown around loop-based performance. These concerns—whether philosophical, stylistic, or practical—form a critique that advocates might take into consideration.

For advocates, however, looping is about the power, joy, and discipline of working with repetition and patterned time, as well as the inherent creative possibilities and challenges that arise from it. Live looping is more than a technique—it becomes a way of thinking musically that emphasizes both structure and freedom.

Constructing live-looping tools in Max, gen~, and RNBO provides pedagogical, technical, and creative benefits as students simultaneously acquire fundamental signal processing knowledge while exploring musical repetition, pattern, and expectation. Our modular, iterative framework encourages cross-environment thinking, repeated engagement with core concepts, and creative adaptation. This approach demonstrates how tool-building can serve as an effective bridge between computational thinking and expressive practice in computer-music education.

8. REFERENCES

[1] Margulis, Elizabeth Hellmuth. *On Repeat: How Music Plays the Mind*. Oxford University Press, 2013.

[2] Huron, David. “A Psychological Approach to Musical Form: The Habituation–Fluency Theory of Repetition.” *Current Musicology* 96 (2013).

[3] Nettl, Bruno. *The Study of Ethnomusicology: Thirty-Three Discussions*. University of Illinois Press, 2015.

[4] Patel, Aniruddh D. *Music, Language and the Brain*. Oxford University press, 2008.

[5] Turkle, Sherry, ed. *Evocative Objects: Things We Think With*. MIT Press, 2011.

[6] Dehaene, Stanislas. *How We Learn: Why Brains Learn Better Than Any Machine ... for Now*. Penguin, 2020.

[7] Eagleman, David. *Livewired: The Inside Story of the Ever-Changing Brain*. Doubleday Canada, 2020.

[8] Bonawitz, Elizabeth, Patrick Shafto, Hyowon Gweon, Noah D. Goodman, Elizabeth Spelke, and Laura Schulz. “The Double-Edged Sword of Pedagogy: Instruction Limits Spontaneous Exploration and Discovery.” *Cognition* 120, no. 3 (2011): 322–30.

[9] Cascone, Kim. “The Aesthetics of Failure: ‘Post-Digital’ Tendencies in Contemporary Computer Music.” *Computer Music Journal* 24, no. 4 (2000): 12–18.

Ponticello: An Interactive Conducting System for Mixed Music Performance

Nikolaus Knop
University of Music and Theatre Munich
nikolaus.knop@stud.hmtm.de

ABSTRACT

In composed music that combines acoustic instruments with electronic processing or fixed media, synchronizing acoustic and electronic layers remains a persistent challenge. The use of click tracks, while technically effective, significantly restricts the performers' freedom to expressively shape musical time. This paper presents Ponticello, a system that addresses the synchronization problem by inferring the ensemble's tempo from a video stream of the conductor in real-time. Instead of the ensemble being beholden to a fixed digital click track, the computer follows the flexible pulse indicated by the conductor, which already functions as a shared temporal reference for the human performers. Although the idea of interactive conducting systems is not new —it has been researched since the 1970s — research has largely focused on applications that simulate instrumental performances based on MIDI scores, which limits their applicability to the performance of mixed music. To support a broad range of compositional strategies for mixed music, Ponticello instead models the electronic part as a timeline of temporally extended, electronic processes whose playback tempo is continuously controlled by the conductor. The system has proven sufficiently reliable and accurate in rehearsal and concert settings across multiple conductors.

1. INTRODUCTION

In most music written for human performers, the free and spontaneous shaping of musical time is a central means of musical interpretation. Ritardandi and accelerandi, whether microscopically small or clearly perceptible, account for much of this music's expressive power. They allow phrases to breathe and musical form to be articulated organically.

In the context of composed mixed music, however, which combines acoustic instruments with fixed media or electronic processing, temporal flexibility is often constrained when precise synchronization between the live performers and the playback of the electronic performance score is required. (The term *electronic performance score*, in this paper, refers to the representation of the electronic part of the composition. Analogous to instrumental parts for human musicians, it specifies which actions or processes the computer should perform and when these should occur.)

In common practice, synchronization is typically achieved using click tracks: the musicians hear a metronome through headphones and synchronize themselves with the computer in this way. This demands attention that would otherwise be directed toward one's own playing and interaction within the ensemble, and it allows little to no freedom in shaping musical time.

At the core of this problem lies a discrepancy between two different concepts of time: on the one hand, the elastic time of human perception; on the other hand, the wall clock time of the computer. Synchronization via a click track resolves this discrepancy in favor of the computer: the ensemble adapts to the computer's fixed temporal grid.

Ponticello reverses the direction of synchronization. Instead of requiring performers to align their playing to a fixed grid, the electronic system adapts to the flexible pulse indicated by the conductor, which already functions as a shared temporal reference for human performers. The conductor's gestures are captured by a camera and analyzed in real time to infer tempo information. This information is used to control the playback speed of an electronic performance score executed by the computer. The electronic performance score is modeled as a timeline of processes specifiable in the SuperCollider language and is editable through a graphical user interface.

By extending the role of the conductor to include the control of the electronic system, the computer becomes capable of following tempo variations in real time. In this way, the electronic performance score is no longer anchored to wall clock time, but to the elastic musical time indicated by the conductor. This allows for precise coordination without constraining the performers' freedom of expressive timing.

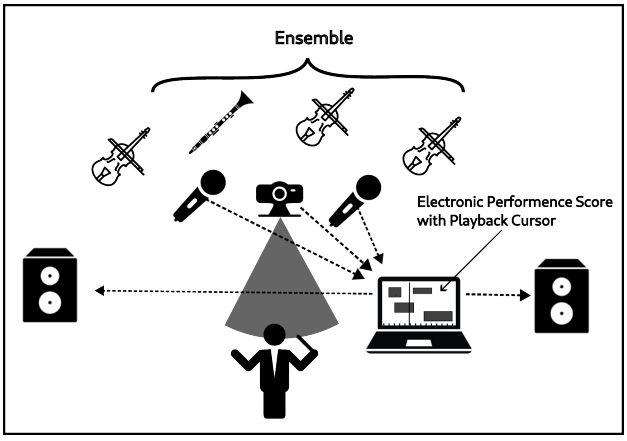


Figure 1. Performance setup

The system is, at the present stage, limited to the temporal coordination aspects of conducting. Other aspects associated with conducting, such as dynamics or articulation, are not addressed.

Copyright: ©2026 Nikolaus Knop. This is an open-access article distributed under the terms of the Creative Commons Attribution License 3.0 Unported, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

2. RELATED WORK

Research on interactive conducting systems has been conducted since at least the 1970s, spanning a wide range of technical approaches [1]. To situate the specific contribution of the present work, it is helpful to categorize this body of research according to both its intended purpose and its underlying technical design.

2.1 Categorization of Existing Systems

2.1.1 Application domains

According to Lee et al., interactive conducting systems can generally be grouped into three main application domains [1]. Most research has been devoted to applications in which the user conducts an entire virtual orchestra. Lee refers to this use case as a "personal orchestra." Early examples of this include Max Mathews' Radio Baton [2], which enabled users to control the playback tempo and dynamics of a MIDI score using custom hardware.

Closely related to this use case are pedagogical applications, in which interactive conducting systems are used to support the learning of conducting technique by providing visual or auditory feedback on gesture accuracy and expressivity. (For an example of this, see [3]).

The present work focuses on a third application domain identified by Lee et al., namely the integration of the computer into an ensemble of human musicians. In this context, interactive conducting systems can be used to coordinate fixed media playback or live processing with human performers.

2.1.2 Input modality

Early research in interactive conducting systems predominantly employed electronically augmented batons incorporating accelerometers and gyroscopes to capture conducting gestures. Nakra et al. [4] extended this approach by employing a wearable garment with multiple embedded sensors to capture not only conducting gestures but also physiological signals, such as muscle tension. More recent research has explored computer vision-based systems that infer gesture information directly from video input. *Ponticello* adopts this approach, employing body landmark detection based on machine learning.

2.1.3 Representation of the Electronic Performance Score

While some earlier systems have taken the approach of dynamic time-stretching of a single audio file (e.g., [5]), most systems relied on MIDI-based score representations, in which conducting gestures control the timing of discrete note events [6]. While effective for the "personal orchestra" use case, these approaches significantly constrain the range of compositional strategies available in mixed music. For example, they do not support techniques such as automation, LFO-based modulation of processing parameters, or live extraction of control data from properties of acoustically produced sounds.

This is where the present approach departs fundamentally from earlier research. Rather than specifying discrete events, the electronic performance score is modeled as a collection of temporally extended electronic processes, which can be specified using the SuperCollider language. The execution tempo of these processes is continuously adapted according to the tempo indicated by the conductor. Compared to MIDI-based representations, this approach supports a broader range of compositional strategies and interactions between human performers and the computer.

2.2 Audio-based Synchronization Systems

The problem of synchronizing an electronic score with live performers has previously been addressed by audio-based score-following systems, most notably *Antescofo* [7]. The primary difference with the present system is, of course, that *Antescofo* derives timing information from the acoustic output of the human performers through machine listening, whereas *Ponticello* is based on real-time analysis of conducting gestures. As a result, the latter does not require detailed knowledge of the instrumental parts or reliable detection of pitches or onsets, but only information about meter and reference tempi for different sections of the piece.

This makes the two systems suitable for different musical scenarios. While *Antescofo* is particularly well suited for solo or small-ensemble works with electronics, especially in situations where no conductor is present or where the use of a conductor is impractical or aesthetically undesirable, *Ponticello*, by contrast, is intended for ensemble contexts in which a conductor is naturally integrated into the performance.

Because synchronization does not depend on the precise execution of the instrumental parts, *Ponticello* is also well suited for works with dense, complex, or non-standard instrumental writing that may be difficult for audio-based systems to follow reliably. In addition, the system can be used for pieces containing "electronics solo" sections, which benefit from conductor-shaped expressive timing.

2.3 Machine Learning in Interactive Conducting Systems

Since Brecht and Garnett [8], the use of machine learning has been explored for the analysis of conducting gestures. Ilmonen [9] continued this line of research, although their system still relied on heuristic components. More recently, Lee [10] presented a machine-learning-based approach to conductor following. Whereas earlier research mostly relied on *feedforward* neural networks, *Ponticello* employs a *recurrent* architecture, to explicitly model temporal dependencies in conducting gestures.

2.4 Positioning of the Present Work

Prior work in interactive conducting systems has largely focused on MIDI-based electronic performance scores, limiting the applicability of these systems in mixed music performance contexts. *Ponticello* combines tempo inference based on machine vision with a SuperCollider-based, process-oriented electronic performance score model, supporting a broad range of compositional strategies in mixed music. Compared to earlier sensor-based approaches, setup complexity and hardware requirements are reduced. This makes it a practical option for musical contexts that are not well served by audio-based synchronization systems.

3. THE ELECTRONIC PERFORMANCE SCORE

The electronic performance score is modeled as a timeline of *score objects* specifying temporally extended processes. Each score object references a *SynthDef*, which encapsulates the synthesis or signal-processing code executed during playback. In SuperCollider, SynthDefs define a signal-processing graph composed of unit generators (*UGens*). Each UGen performs a specific digital signal processing function, such as oscillation, filtering, or physical modeling, and outputs a stream of values. UGens are connected by using the output of one as the input for another, or by reading from and writing to global *buses*.

SynthDefs define a set of parameters that expose the controllable aspects of the signal processing. Each score objects maintains a set of controls that are assigned to the parameters of its associated SynthDef.

In *Ponticello*, controls can take several forms:

- constant values,
- control or audio buses,
- automation curves defined over score-time.
- SuperCollider expressions evaluated at runtime. When an expression evaluates to a UGen, it is used to modulate the corresponding parameter.

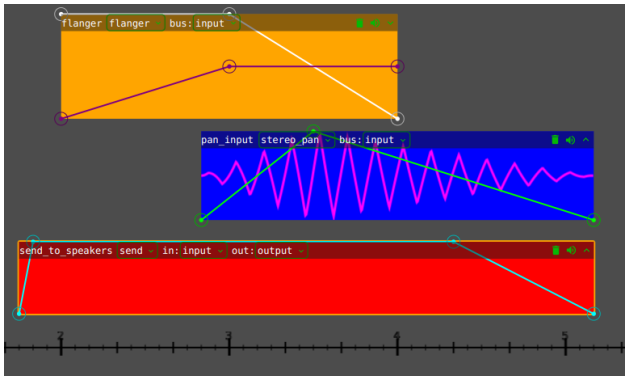


Figure 2. Section of an example electronic performance score as displayed in *Ponticello*'s graphical user interface. The tempo grid defines a 3/4-meter. Over the course of one measure, the depth of a flanger process is gradually increased. In the course of measure 3, a pan modulation process is introduced, whose modulation amplitude rises quickly and then decreases more slowly. The send object routes audio from the `input` bus to the `output`, allowing the live-processed signal to be played back through loudspeakers. The amp parameter is automated to fade-out the electronics at the end of the phrase.

In addition to regular score objects, the score contains one or more *tempo grids* that define the meter and reference tempo for the piece or individual sections of it. These grids establish a mapping from bars, beats and their subdivisions to *score-time*, which is measured in seconds and corresponds to the horizontal axis of the score. This provides the composer with orientation within the electronic performance score, relating its temporal positions to corresponding locations in the performers' parts. Tempo grids are not intended as prescriptive tempo indications for performance, but function solely as a reference, allowing the conductor to choose the performance tempo freely.

Through *Ponticello*'s graphical user interface, the user can define and edit SynthDefs, create score objects referencing these SynthDefs, reposition and resize them within the timeline, and assign or modify control mappings. In this way, *Ponticello* combines the capabilities of SuperCollider's sound synthesis model with direct interaction through a graphical score. Like in traditional DAWs, the electronic performance score thus functions both as a compositional medium and as a performance interface.

3.1 Playback of the Electronic Performance Score

During performance, the tempo grids provide a reference against which the pulse indicated by the conductor can be interpreted by the system. A playback cursor traverses the score. Its speed is continuously adjusted according to the ratio between the tempo indicated by the conductor and the reference tempo defined by the currently active tempo grid.

When the cursor reaches the start of a score object, a corresponding *Synth* instance is created on the SuperCollider server. This instance remains active until the cursor reaches the end of the score object or playback stops, at which point the Synth is released.

The ordering of Synth nodes within the SuperCollider node tree is determined by the vertical position of the corresponding score objects, making the signal-processing order visually apparent and directly editable.

When creating and manipulating the digital score, the composer operates in terms of *score-time*, either directly or through a tempo grid. The *physical time*, on the other hand, that is required to traverse a given region of score-time, depends on the decisions of the conductor during performance. For example, if the reference tempo specified by the score is 100 BPM and the conductor indicates a tempo of 80 BPM at a given moment, the playback speed is adjusted to $\frac{80}{100} = \frac{4}{5} = 0.8$, stretching score-time by that factor. In this way, the electronic performance score becomes *elastic* during performance.

4. TEMPO INFERENCE

The tempo inference system is implemented as a real-time pipeline composed of three layers, operating on a continuous video stream of the conductor.

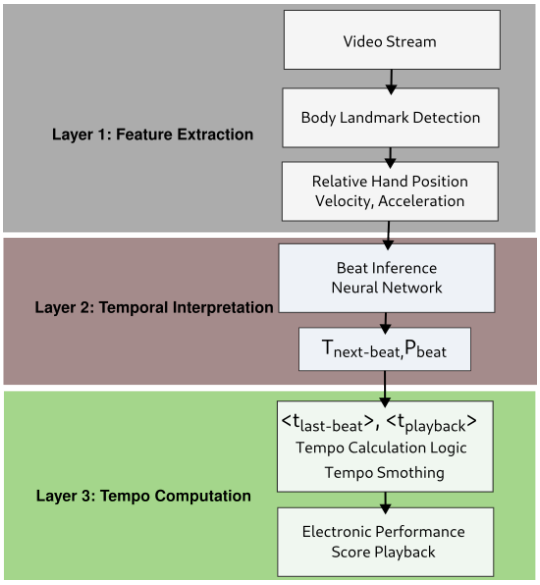


Figure 3. Overview of the tempo inference pipeline

Each video frame is first processed by a computer vision module to extract relevant body landmarks of the conductor. From these landmarks, the position, velocity, and acceleration of the conductor's right hand relative to the torso are computed, resulting in a representation suitable for temporal analysis.

The second layer constitutes the core musical intelligence of the system: a sliding window of extracted features is passed into a recurrent neural network, which is trained to predict two variables:

- $T_{next-beat}$, the estimated physical time interval until the conductor indicates the next beat.
- P_{beat}, the probability that the current frame corresponds to a beat.

The third layer is responsible for adjusting the playback speed of the digital score. It maintains two state variables:

- $t_{last-beat}$, the score-time position of the last recognized beat within the digital score.
- $t_{playback}$, the current position of the playback cursor within the digital score. It is continuosly updated based on the playback speed Δt computed by the tempo inference mechanism, making the two variables mutually dependent.

Based on these variables and the predictions of the neural network, the system continuously adjusts the playback speed of the digital score. At each video frame, the system computes the time interval from the current playback position to the next beat of the active tempo grid. The playback speed is then derived as the ratio between this score-time distance and the time interval to the next indicated beat as predicted by the neural network. To reduce noise, the playback speed is smoothed using an exponential filter, with the smoothing factor α configurable by the user.

Algorithm 1: Tempo inference pipeline.

$seq \leftarrow []$ `//sliding window of feature vectors`
$\Delta t \leftarrow 1$
for each received video frame **do**
  $(x, y) \leftarrow$ estimate relative right hand position
  $(\Delta x, \Delta y) \leftarrow$ calculate velocity
  $(\Delta^2 x, \Delta^2 y) \leftarrow$ calculate acceleration
  append $(x, y, \Delta x, \Delta y, \Delta^2 x, \Delta^2 y)$ to seq
  `//`N `denotes the feature window length`
  if $len(seq) \neq N$ **then**
    $(P_{beat}, T_{next-beat}) \leftarrow$ NN.predict(seq)
    `//`$tempo_{ref}$ `refers to the tempo defined by the active tempo grid`
    $t_{next-beat} \leftarrow t_{last-beat} + \frac{60}{tempo_{ref}}$
    $\Delta t_{next-beat} \leftarrow t_{next-beat} - t_{playback}$
    $\Delta t \leftarrow \frac{\Delta t_{next-beat}}{T_{next-beat}} \cdot \alpha + \Delta t \cdot (1 - \alpha)$
    remove first item from seq
    $dt_{phys} = wall\_clock\_time() - timestamp$
    if $P_{beat} \geq threshold$ **and** $dt_{phys} > min\_beat\_dur$ **then**
      $t_{last-beat} \leftarrow t_{next-beat}$
      $timestamp \leftarrow wall\_clock\_time()$
    end
  end
end

5. IMPLEMENTATION DETAILS

5.1 Programming Languages and Frameworks

The tempo inference system is implemented in Python using the Python API of Google's *MediaPipe* library, which provides pre-trained models for body landmark detection.

Ponticello's electronic performance score model and graphical user interface is implemented in *Kotlin*, a general-purpose, JVM-compatible programming language.

Communication between the performance score model, the tempo inference process, and SuperCollider is handled via OSC. The system communicates not directly with the synthesis server (*scsynth*), but with an *sclang* process to which it sends SuperCollider expressions for evaluation.

5.2 Dynamic Audio Time Stretching

To synchronize the playback of audio buffers with the conductor's gestures, the system must be able to adjust the buffer's playback rate continuously while preserving pitch. This is achieved through granular time stretching, using SuperCollider's *Warp1* unit generator.

A playback cursor moves through the audio buffer at a rate determined by the conductor's tempo. Rather than reading the buffer sequentially at a fixed speed, the system continuously extracts short grains from the buffer at the current cursor position. These grains are then recombined to produce a continuous audio signal.

This method allows audio buffers to expand or contract in real-time in response to conducting gestures without altering pitch. As a result, playback remains tightly synchronized with the ensemble even under significant tempo variation. (Granulation parameters, such as grain duration and density, provide compositional control over the texture of the resulting sound.)

5.3 Elastic Audio Recording

In some mixed music pieces, audio material recorded during the performance is reintroduced at later points in the piece. This presents an additional challenge with regards to synchronization when musical time is shaped freely by the ensemble. If recorded material is later replayed under a different conducted tempo, just time-stretching the audio material according to the new tempo fails to preserve the relative timing relationships.

To address this, the system captures a mapping between score-time and physical time while recording audio to be used during playback. During recording, audio is written into a buffer normally, while a continuous mapping from score-time to physical time is simultaneously stored in a separate buffer.

Algorithm 2: Recording the time mapping buffer

$t_{score} \leftarrow 0$;
$t_{phys} \leftarrow 0$;
for each audio frame **do**
  $T_{phys}[t_{score}] \leftarrow t_{phys}$
  $t_{score} \leftarrow t_{score} + \Delta t$
  $t_{phys} \leftarrow t_{phys} + \frac{1}{\text{audio framerate}}$
end

During playback, score-time advances according to the currently conducted tempo, and the grain index into the audio buffer is determined via the stored mapping. From the current score-time position the corresponding physical-time position is interpolated, which is then used to extract grains from the audio buffer.

Algorithm 3: Playback of live recorded buffers

$t_{score} \leftarrow 0$
for *each audio frame* **do**
  $t_{phys} \leftarrow$ interpolate $T_{phys}(t_{score})$
  move grain position cursor to t_{phys}
  $t_{score} \leftarrow t_{score} + \Delta t$
end

As a result, the relative temporal structure of the recorded material is preserved, while the overall playback adapts smoothly to the current conducted tempo.

This mechanism is implemented using two custom unit generators written in C++, one capturing the mapping from score-time to physical time in a separate buffer during recording and the other reading from that buffer and calculating the appropriate grain positions in the audio buffer during playback.

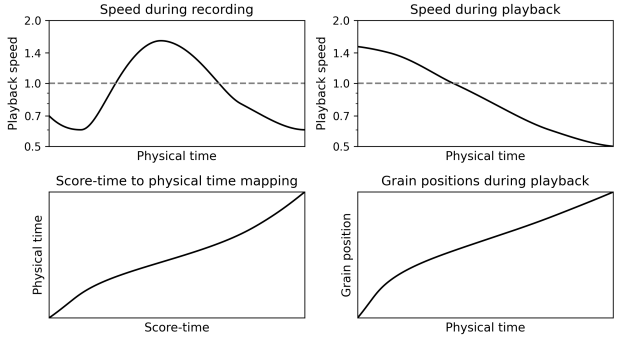


Figure 4. The example shows two tempo curves at the top: one captured during the recording of an audio buffer and the other during playback. The lower-left graph depicts the time-mapping function derived from the recorded tempo curve. The lower-right graph shows the grain index into the audio buffer during playback.

5.4 Neural Network Architecture

The model takes as input a sequence of feature vectors extracted from consecutive video frames and outputs a pair of values $(P_{beat}, T_{next-beat})$, as defined in section 4. During training, target values are derived from manually annotated beat positions in the corresponding video recordings.

The choice of feature window length involves a trade-off. Increasing the window size allows the model to incorporate a broader temporal context, but also results in increased training time and requires more frames to be accumulated before the first prediction can be produced during performance. Experiments indicate that a window length corresponding to approximately five seconds of conducting gesture data is sufficient for optimal results in most contexts.

Beat prediction from postural features is inherently a temporal inference problem. Recurrent neural networks are therefore well suited to this task, as they can model temporal dependencies in sequential feature data.

Among recurrent neural network architectures, Gated Recurrent Units (GRUs) offer a practical trade-off between model capacity and training effort. Their gating structure avoids the instability and vanishing-gradient problems commonly encountered with simple recurrent networks. Compared to more complex architectures such as LSTMs, GRUs require fewer parameters and thus typically less training data to achieve reliable performance,

6. DISCUSSION

6.1 Use in Rehearsal and Concert Situations

Through the system's graphical user interface, the user can select the trained neural network model file (.pth) used for beat inference during performance and configure associated parameters such as the smoothing factor, beat confidence threshold, and minimum beat distance.

Before starting the performance, the conductor can also select the desired starting point in the digital score by positioning the playback cursor at a specific bar. This allows sections of the piece to be rehearsed without restarting from the beginning. The system is initiated using a MIDI foot pedal. Following this trigger, the conductor conducts one full bar in advance, allowing the system and the musicians to establish a common tempo before the acoustic performance begins and the electronic playback starts.

6.2 Tempo Inference Latency

In practice, the system's real-time latency depends almost entirely on the frame rate of the input video stream. The additional latency introduced by body landmark extraction, neural network inference, and the tempo calculation is negligible by comparison. When using a standard USB video camera operating at 30 frames per second, the resulting latency is approximately 35 ms. This latency does not correspond to a constant temporal offset between the system and the performers, but instead reflects the time required for the system to respond to tempo changes indicated by the conductor, making it acceptable in a musical context.

6.3 Reliability and precision

In both rehearsal and performance, the system has demonstrated stable and reliable behavior, maintaining consistent synchronization between the ensemble and the electronic processes over extended musical passages. Given sufficient training data, the system very rarely misses beats, making false negatives highly improbable.

To reduce the occurrence of false positive beat detections, the system allows specifying a minimum time interval between consecutive beats. This parameter can be adjusted by the user according to the maximum tempo expected during performance.

The system achieves its highest accuracy for conducting styles that are represented in the training data, which currently comprises approximately three hours of video material. Initial tests indicate, however, that adaptation to new conductors or conducting styles can be achieved with relatively little additional data: approximately twenty minutes of annotated video material are sufficient to fine-tune the model for reliable operation.

7. CONCLUSION

This paper has presented an electronic performance score model and a system that supports synchronization between live performers and the playback of an electronic performance score, without constraining the ensemble's control over musical timing. By following the conductor's tempo, the electronic performance score becomes elastic during performance, in direct analogy to the temporal flexibility found in the interpretation of music notated for human musicians. Just as meter and rhythm in an instrumental score often provide a structural framework rather than a rigid prescription, the electronic performance score defines a relative temporal structure whose realization in physical time is shaped continuously by the conductor.

Rather than treating artificial intelligence as a replacement for human creativity, the approach employs neural networks as a mediating layer between human musical gestures and computational models. Advances in computer vision and machine learning facilitate the implementation of reliable, camera-based conducting interfaces that avoid the need for specialized or expensive hardware. This makes the technology accessible beyond institutions equipped with custom performance technology.

Ponticello has been used both in rehearsal settings and in concert performances and has proven sufficiently reliable and accurate across multiple conductors. It can be adapted with relatively little effort to conducting styles not represented in the core training data.

This opens up possibilities for composing new works of mixed music that require both tight synchronization between acoustic and electronic layers and the freedom for performers to expressively shape musical time. It also allows for the adoption of existing mixed music repertoire that can benefit aesthetically from a more flexible treatment of musical time.

8. FUTURE WORK

Future work will focus on expanding the training dataset to include a broader range of conducting styles. As the diversity of conducting styles included in the training set increases, the system's ability to generalize across different performers is expected to improve correspondingly.

The system's reliability may be further improved by considering the position of beats inside the measure. First beats inside a measure are typically indicated in a clearly distinct way by most conductors and thus should establish recovery points when beats are missed.

Another area of investigation concerns the integration of gestures commonly used in rehearsal. Conductors sometimes employ specific gestures to stop an ensemble, return to a previous rehearsal point, or restart from the beginning. Extending the system to recognize such gestures would improve its practical usability in rehearsal situations.

While the current system focuses on the temporal coordination aspects of conducting, conducting gestures convey additional musical information that could be incorporated into the model. Future extensions could aim to infer parameters such as dynamics and articulation from gesture features, allowing electronic processes to react not only to the conductor's tempo but also to other parameters indicated through conducting gestures.

9. REFERENCES

[1] K. Lee, M. Junokas, and G. Garnett, "A Review of Interactive Conducting Systems: 1970-2015," in *Proceedings of the 2015 International Computer Music Conference*, Utrecht, The Netherlands, 2016, pp. 485–491.

[2] M. V. Mathews, "The radio baton and conductor program, or, pitch, the most important and least expressive part of music," *Computer Music Journal*, vol. 15, pp. 37–46, 1991.

[3] G. Garnett, F. Malvar-Ruiz, and F. Stoltzfus, "Virtual Conducting Practice Environment," in *Proceedings of the 1999 International Computer Music Conference*, Beijing, China, 1999, pp. 371–374.

[4] T. M. Nakra, T. Machover, and R. W. Picard, "Inside the conductor's jacket: analysis, interpretation and musical synthesis of expressive gesture," 2000. [Online]. Available: https://api.semanticscholar.org/CorpusID:17902572

[5] J. O. Borchers, E. Lee, W. Samminger, and M. Mühlhäuser, "Personal orchestra: a real-time audio/video system for interactive conducting," *Multimedia Systems*, vol. 9, pp. 458–465, 2004.

[6] S. Bergen and T. Lokki, "Conductor Follower: Controlling sample-based synthesis with expressive gestural input," 2012. [Online]. Available: https://api.semanticscholar.org/CorpusID:20546548

[7] A. Cont, "ANTESCOFO: Anticipatory Synchronization and Control of Interactive Parameters in Computer Music." in *Proceedings of the 2008 International Computer Music Conference*, Belfast, Ireland, 2008, pp. 33–40.

[8] B. Brecht and G. E. Garnett, "Conductor Following," in *Proceedings of the 1995 International Computer Music Conference*, Banff, AB, Canada, 1995, pp. 185–186.

[9] T. Ilmonen and T. Takala, "Conductor Following With Artificial Neural Networks," in *Proceedings of the 1999 International Computer Music Conference*, Beijing, China, 1999, pp. 367–370.

[10] M. Lee, "Deep Neural Network based Music Source Conducting System," in *Proceedings of the 2018 International Computer Music Conference*, Daegu, South Korea, 2018, pp. 17–20.

Factor Analysis of Similarity in the Same Orchestral Piece

Minami Kojima
Ochanomizu University
kojima.minami@is.ocha.ac.jp

Rafael Ramírez-Meléndez
Pompeu Fabra University
rafael.ramirez@upf.edu

Takayuki Itoh
Ochanomizu University
itot@is.ocha.ac.jp

ABSTRACT

There have been conventional research comparing multiple performances of the same classical music piece, where many of them primarily discussed the similarity between performances but rarely delved into why those performances are similar. Furthermore, conventional similarity analysis has dealt with the similarity of single features in classical music, such as timbre, tempo, and loudness. However, since actual audiences perceive these features compounded without separating them, relying solely on single features is insufficient to fully represent musical style. To overcome this challenge, our study uses a deep learning model (VGGish, specialized in audio feature extraction) that captures high-level timbral and textural features, in addition to acoustic features (timbre, loudness, tempo), targeting the same orchestral piece. Subsequently, based on those features, we define five grouping criteria: (1) same orchestra, (2) same conductor, (3) same country of orchestra, (4) same nationality of conductor, and (5) same teacher of the conductor. We then evaluate the clustering performance for each criterion. A group exhibiting high clustering performance suggests mutual similarity among the performances within that group, leading us to conclude that the corresponding criterion represents a major factor influencing performance style. The results in this paper show that conductor identity consistently yields the strongest clustering for tempo-related features, while orchestra identity dominates timbral similarity in specific movements. We demonstrate that metadata-driven factors explain similarity beyond purely perceptual or affective similarity measures.

1. INTRODUCTION

The performance of classical music, particularly orchestral pieces, exhibits significant interpretive diversity even within the same piece, stemming from the combination of the orchestra and the conductor. In recent years, the spread of music streaming services has made a vast number of recorded works easily accessible to listeners. For example, approximately 560 performances of Beethoven's Symphony No. 7 alone are available on Apple Music. This situation makes it increasingly difficult for listeners to independently find a performance within the same piece that matches their preferences, posing a crucial challenge in music recommendation technology.

Copyright: ©2026 Minami Kojima et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License 3.0 Unported, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Against this background, studies concerning performance recommendation for identical pieces has been advancing. As a prior study, Grekow [1] calculated the similarity between performances of the same piece using musical emotion recognition, a technique that automatically estimates the emotions perceived by the listener upon hearing the music, and recommended similar recordings. However, calculating similarity for all hundreds of orchestral performances, where a single piece can last several dozen minutes, is not realistic from a computational cost perspective.

Therefore, this study explores the possibility of performance recommendation utilizing not only performance similarity but also recording metadata. If we can investigate the factors influencing performance similarity, we can also apply this information to adjust parameters in a performance recommendation system, leading to an expected improvement in recommendation accuracy.

The objective of this study is to analyze what factors influence performance similarity. Specifically, this study measures performance similarity based on different grouping criteria, such as same orchestra, same conductor, and same teacher, to analyze the factors that significantly impact the performance style. We define performance style as consistent patterns in tempo, loudness, and timbre across recordings.

2. RELATED WORK

Research on the similarity comparison of the same musical piece has been conducted using various methods. Studies focusing on piano performance similarity include the Performance Worm [2], [3], which visualizes changes in tempo and loudness as a moving trajectory over time, and research that automatically searches and optimizes inter-performance algorithms using evolutionary algorithms [4]. Sapp [5] proposed a pyramid-shaped similarity map that simultaneously visualizes the global and local similarity, specifically, which segment and at what temporal scale, another performance most closely resembles a standard performance, based on changes in tempo and loudness. Thus, while a wide variety of original methods have been proposed for piano pieces, studies targeting orchestral music remain scarce. This is due to the lower availability of MIDI data, the basis for feature extraction, in orchestral music compared to piano music, and the complexity arising from influencing factors. For piano music, the factor influencing performance similarity is the "performer only," whereas orchestral music involves the complex combination of two elements: the "conductor and the orchestra ensemble."

Existing research dealing with orchestral music includes Liem's study [6], which analyzed timbral differences and similarities by analyzing spectrograms using Principal

Component Analysis (image analysis). Further work includes the study of Grekow [7], which determined similarity using musical emotion recognition techniques, and the study of Yanchenko et al. [8], which used Hierarchical Multidimensional Scaling (HMDS) to clarify systematic similarities between orchestras not only within a single piece but also across movements and different symphonies.

Conventional studies often focus similarity comparison on single acoustic features, such as timbre, loudness, or tempo. However, since actual audiences perceive these features in a complex, integrated manner rather than separately, relying on a single feature alone is insufficient to adequately represent musical style. Therefore, in addition to conventional acoustic features, we utilize features derived from VGGish, a deep learning model. VGGish is well-suited to our task because it produces fixed-length embeddings that summarize long-term spectral–temporal statistics learned from large-scale audio data, enabling robust comparison of orchestral recordings despite local timing fluctuations and expressive micro-variation. By capturing higher-level timbral and textural characteristics beyond hand-crafted features, VGGish embeddings provide a complementary representation that is sensitive to orchestral sound and identity while remaining computationally efficient for large collections. However, because VGGish embeddings are temporally aggregated and trained for general audio classification rather than expressive music analysis, they may not sufficiently capture local timing, articulation, and phrase-level dynamics that are central to musical interpretation, and should therefore be interpreted as capturing global acoustic similarity rather than detailed performance expressivity. Furthermore, previous studies have left challenges regarding the insufficient validation of the applicability of their results to other performances due to a limited number of analyzed recordings, and also the lack of systematic analysis concerning the factors that influence performance similarity. Therefore, this study collaboratively analyzes large-scale orchestral performance data and metadata to derive hypotheses about stylistic transmission of performance.

3. METHODS

3.1 Selection and Collection of Musical Pieces

This study selected the target pieces based on the criteria of recording diversity and the presence of a clear rhythmic structure. The clear rhythmic structure is required to ensure the stability and accuracy of the Dynamic Time Warping (DTW) algorithm, which will be discussed later. Based on these conditions, We collect 55 performances of Beethoven's Symphony No. 7 (with multiple recordings from 8 conductors and 8 orchestras) and 71 performances of Tchaikovsky's Symphony No. 5 (with multiple recordings from 9 conductors and 8 orchestras) from Apple Music and YouTube. These two pieces are also selected to enable a comparative analysis of performance tendencies. Beethoven's symphony exhibits the clear structural traits characteristic of the Classical era, whereas Tchaikovsky's symphony is a Romantic work rich in emotional and coloristic expression. We analyze how performance similarity differs between these two works with distinct musical characteristics. To ensure the uniformity of performance tendencies, we have restricted the collected recordings to those made after the year 2000. This is based on two reasons: the potential for the orchestra's unique performance style to change due to generational shifts in members when there is a large difference in recording dates, and the acoustic discrepancies inherent in older recordings due to differing recording technologies. Additionally, recordings using period instruments or recordings performed by chamber orchestras are excluded from the analysis to unify the acoustic characteristics for comparison.

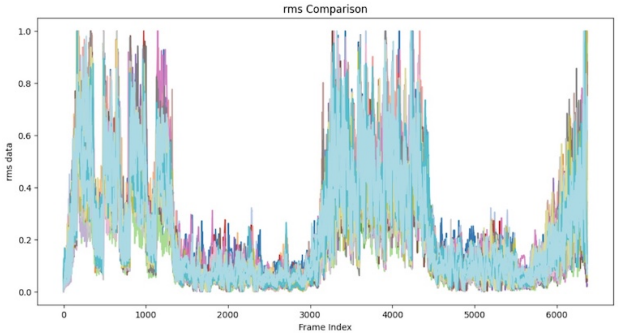


Figure 1: RMS energy profiles for multiple performances synchronized via DTW. The high degree of alignment in the temporal evolution of energy magnitude (vertical axis) confirms the accuracy of the DTW-based synchronization across the frame series (horizontal axis).

3.2 Dataset Construction

Next, we obtain five elements that could influence performance similarity as features for the analysis dataset. These features are: (1) orchestra name, (2) conductor name, (3) orchestra's country of origin, (4) conductor's nationality, and (5) the conductor's teacher.

3.3 Beat Alignment Across Performances

In classical music performance, the tempo can fluctuate significantly even within the same piece. If the analysis proceeds without accounting for such tempo differences, a problem arises where the exact correspondence between musical passages cannot be accurately identified, despite the pieces being identical. Therefore, in this study, we perform temporal alignment between performances using Dynamic Time Warping (DTW). DTW is a technique for measuring the similarity between two time-series data, which makes it possible to establish an accurate correspondence even between performances with differing tempos.

Figure 1 illustrates the temporal evolution of Root Mean Square (RMS) values calculated from multiple recordings of the same piece after the application of DTW. The vertical axis represents the energy magnitude, while the horizontal axis indicates the frame count. Given that the peaks and the timing of energy increases in the RMS, a metric representing acoustic energy, align consistently across the various recordings, it is demonstrated that the time-axis synchronization via DTW was performed with high precision.

We adopt the Python Sync Toolbox package [9] for the

DTW processing. The reasons for selecting this package are primarily twofold.

First, Sync Toolbox does not merely implement a DTW function; it provides a comprehensive synchronization pipeline that integrates a series of processes, ranging from feature extraction and alignment to visualization and evaluation. This allows for the overall efficiency of the research process to be streamlined. Second, Sync Toolbox includes implementations of Multiscale DTW (MsDTW) and Memory-Restricted MsDTW (MrMsDTW). These methods achieve high-precision synchronization performance while significantly improving computational efficiency and memory usage compared to conventional DTW. Since this study involves handling numerous long-duration music recordings, spanning several tens of minutes, these computational optimizations are essential, leading to the decision to adopt this package.

Furthermore, this study used MIDI recordings from Kunstderfuge.com [10] as the standard performance source to align the temporal axis of each performance. The study then applied DTW between the standard MIDI source and each performance recording to establish temporal correspondence. After DTW application, the corresponding frame number in each performance recording is compiled for every single frame of the reference MIDI audio, and this correspondence relationship is saved as a CSV-formatted table. This process allowed for the comparison of features across all performance recordings, regardless of their differing tempos, based on the unified musical time axis provided by the MIDI audio frames.

3.4 Feature Extraction

We use two types of features: acoustic features and VGGish feature derived from deep learning.

3.4.1 Acoustic Feature

For acoustic features, three types of features related to volume, timbre, and tempo are acquired. The features for volume and timbre are extracted using librosa, a Python library specialized in acoustic analysis and feature extraction.

1. Loudness

   Root Mean Square (RMS) : RMS is a feature that represents the average energy of a sound. We normalize this feature to ensure fair comparison by eliminating the influence of volume differences between recordings.

2. Timbre

   1) Mel-Frequency Cepstral Coefficients (MFCC): MFCCs are features that efficiently represent the spectral envelope of a sound by transforming the frequency spectrum to reflect human auditory characteristics (Mel scale). We utilize 13 dimensions for these features.

   2) Spectral Flatness: Spectral flatness is a feature of how flat a signal's power spectrum is, reflecting the noisiness or timbral roughness of the sound. A high value indicates a noise-like sound, while a low value suggests a tonal or periodic sound with a clear harmonic structure.

   3) Spectral Centroid: Spectral centroid is a feature that indicates where the center of gravity of the spectrum is located. It is generally used as a measure of the brightness of the sound.

   4) Spectral Rolloff: Spectral Rolloff is a feature defined as the frequency below which a specific percentage of the total spectral energy is contained.

   5) Spectral Bandwidth: Spectral bandwidth is a feature that represents the spread of the spectrum's distribution. A large value indicates that the sound contains a wide range of frequency components.

3. Tempo

   We measure the amount of time (in frames) by which the segment from MIDI audio frame i to $i + W - 1$ is stretched or shrunk in performance recording j. This warping rate is calculated by dividing the difference between the maximum and minimum frame numbers of the performance recording within the segment by the reference MIDI audio frame width (window size W).

   The warping rate $\lambda_{i,j}$ is defined as:

   $$\lambda_{i,j} = \frac{P_j[i + W - 1] - P_j[i]}{W}$$

   Here, P_j represents the corresponding frame number of MIDI frame i in performance j, W is the segment size used for calculating the warping rate, and i indicates the starting frame of the segment. A value of $\lambda_{i,j} \approx 1$ indicates tempo consistency with the reference MIDI; a value > 1 indicate acceleration, and avvalue < 1 indicate deceleration.

After processing the acoustic features, this study generates the following three types of two-dimensional vectors for use in comparing the similarity between performances:

1. Loudness / Tempo

   A two-dimensional vector that integrates features related to loudness and tempo.

2. Timbre

   A two-dimensional vector obtained by compressing the timbre features (17 dimensions) using Principal Component Analysis (PCA).

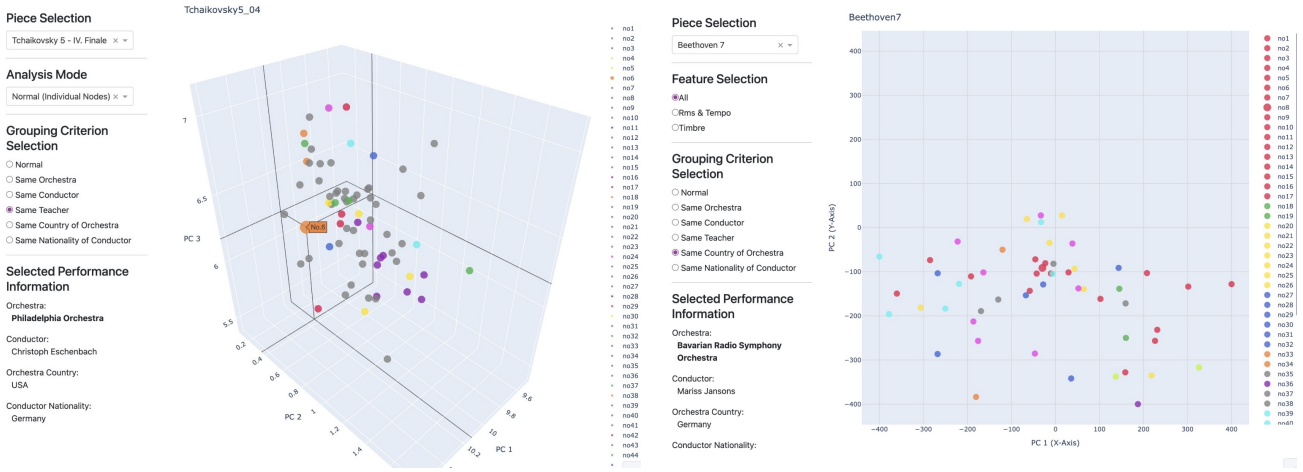
3. Tempo

   A two-dimensional vector obtained by compressing all features (19 dimensions)using PCA.

3.4.2 VGGish Feature

We adopt VGGish [11] for feature extraction using deep learning. VGGish is based on a Convolutional Neural Network (CNN) and has significantly lower computational requirements compared to large-scale transformer-based models, offering superior inference efficiency on a CPU. The audio sources analyzed in this study are extensive in total duration, consisting of dozens of performances, with each movement lasting approximately ten minutes. Therefore, we determined that VGGish is the most appropriate model from the perspective of low computational load and practicality.

Figure 2: Example of Visualization System Usage (Left: VGGish Features, Right: Acoustic Features)



In the extraction process, the audio signal is first converted to 16 kHz mono and then segmented into 0.96-second intervals. Each interval is converted into a log-Mel spectrogram, which serves as the input to the VGGish model. The inference results in a 128-dimensional acoustic embedding obtained from each input interval. Since VGGish interprets the embeddings as capturing the overall acoustic similarity rather than detailed performance representation, we average the embeddings from all frames obtained for each performance and compile them into a 128-dimensional vector without adjusting the beats between performances using DTW. Subsequently, this 128-dimensional vector is progressively compressed, first to 10 dimensions using Uniform Manifold Approximation and Projection (UMAP), and then to 3 dimensions using PCA. This final three-dimensional vector is used for the similarity comparison between performances.

3.5 Visualization

Based on the acquired features, we visualize the similarity between performances. We developed the visualization system using Python Dash for the user interface and Plotly for rendering the graphics, where each coordinate point on the chart represents a single performance. This system is positioned as an exploratory tool to grasp data trends, rather than as a performance evaluation tool.

Users can flexibly change the visualization conditions using the panel on the left. Specifically, the analysis piece can be selected using "Piece Selection," and the grouping setting based on five types of metadata can be chosen using "Grouping Criterion Selection." For example, if the "Same Conductor" is selected, the coordinates of performances by the same conductor are colored the same. Since three types of acoustic feature data exist, the data type can be switched using "Coordinate Data Selection." Furthermore, a function is implemented to allow users to view the corresponding metadata for a performance in the "Selected Performance Information" by clicking on its coordinate point.

4. ANALYSIS METHOD

The main analytical objective of this study is to identify which feature and grouping criterion yield the closest similarity among performances, thereby determining the factors that most significantly influence performance style.

To quantitatively evaluate this broad trend, we employed the Davies-Bouldin Index (DBI) [12], because it explicitly balances intra-cluster compactness against inter-cluster separation using a pairwise cluster similarity formulation, making it well-suited for evaluating clustering induced by predefined metadata categories rather than learned partitions. Unlike the Silhouette score, which assumes point-wise cluster assignment quality and is sensitive to uneven cluster sizes, DBI provides a more stable assessment when clusters vary in cardinality, as is the case for conductor- and orchestra-based groupings. DBI is an index used to evaluate the quality of clustering; lower values indicate higher quality clustering, signifying both high density within the clusters and high separation between clusters. Consequently, a group exhibiting a lower DBI value can be interpreted as representing a major factor that influences performance similarity.

Furthermore, the more specific similarity between individual performances is analyzed in detail using the visualization system.

5. ANALYSIS (DBI DATA)

In this section, we present the results obtained from the DBI data and propose a hypothesis derived from these findings.

5.1 Table Format

In this study, we define five grouping criteria for factors influencing performance similarity: (1) same orchestra, (2) same conductor, (3) same country of orchestra, (4) same nationality of conductor, and (5) same teacher of conductor. In the figures and tables, these criteria are abbreviated as (1) orchestra, (2) conductor, (3) country (orchestra), (4) nationality (conductor), and (5) teacher (conductor), respectively.

5.2 Comparison of Results by Symphony

First, we compare the first movement of Beethoven's symphony with all movements of Tchaikovsky's symphony

Figure 3: DBI Evaluation for Beethoven's Symphony No. 7 and Tchaikovsky's Symphony No. 5

(a) Beethoven's Symphony No. 7 DBI Evaluation

Class.	Feature	Grouping Criterion	DBI
Acoustic Feature	timbre	conductor	2.281
	all	conductor	2.356
	timbre	orchestra	2.715
	tempo_rms	teacher(conductor)	2.737
	all	teacher(conductor)	4.535
VGGish Feature	–	conductor	3.459
	–	teacher(conductor)	4.160
	–	country(orchestra)	5.809

(b) Tchaikovsky's Symphony No. 5 DBI Evaluation

Class.	Feature	Grouping Criterion	DBI
Acoustic Feature	all	conductor	3.853
	timbre	conductor	3.991
	all	orchestra	4.186
	tempo_rms	conductor	4.294
	all	teacher(conductor)	4.780
VGGish Feature	–	orchestra	4.034
	–	conductor	4.168
	–	country(orchestra)	4.875

Figure 4: DBI Evaluation for Tchaikovsky's Symphony No. 5 Movements 1 and 2.

(a) Movement 1 DBI Evaluation

Class.	Feature	Grouping Criterion	DBI
Acoustic Feature	timbre	conductor	2.928
	all	conductor	3.684
	all	orchestra	3.740
	tempo_rms	conductor	5.258
	timbre	teacher(conductor)	5.519
VGGish Feature	–	conductor	3.981
	–	orchestra	4.082
	–	teacher(conductor)	5.148

(b) Movement 2 DBI Evaluation

Class.	Feature	Grouping Criterion	DBI
Acoustic Feature	timbre	orchestra	2.866
	all	orchestra	2.946
	tempo_rms	conductor	3.442
	all	teacher(conductor)	3.962
	all	conductor	4.327
VGGish Feature	–	conductor	3.790
	–	country(orchestra)	3.927
	–	nationality(conductor)	4.888

Figure 5: DBI Evaluation for Tchaikovsky's Symphony No.5 Movements 3 and 4.

(a) Movement 3 DBI Evaluation

Class.	Feature	Grouping Criterion	DBI
Acoustic Feature	tempo_rms	teacher(conductor)	3.265
	all	conductor	3.355
	tempo_rms	conductor	3.644
	tempo_rms	nationality(conductor)	3.803
	timbre	conductor	4.395
VGGish Feature	–	orchestra	2.954
	–	conductor	3.742
	–	country(orchestra)	4.069

(b) Movement 4 DBI Evaluation

Class.	Feature	Grouping Criterion	DBI
Acoustic Feature	all	teacher(conductor)	2.991
	timbre	teacher(conductor)	3.051
	all	conductor	4.046
	timbre	conductor	4.091
	all	orchestra	4.769
VGGish Feature	–	orchestra	3.733
	–	teacher(conductor)	4.907
	–	country(orchestra)	5.129

(using the average of the values for each movement). Regarding the acoustic features, both symphonies commonly showed good grouping performance when clustered by same conductors (timbre and all features). The average of the top five DBI values for Beethoven's Symphony was 2.925, while the average for Tchaikovsky's Symphony was 4.221. This means that the Beethoven recordings exhibit higher similarity between performances. This observation support the difference that Beethoven's music, belonging to the Classical era, has an established fundamental performance style, whereas Tchaikovsky's music, belonging to the Romantic era, allows for greater interpretive freedom and wider expressive range. In the analysis using VGGish features, we observed a difference that the group showing the highest similarity for Beethoven's Symphony was the same conductor, whereas the same orchestra ranked first for Tchaikovsky's Symphony. This suggests that in the performances of Tchaikovsky, the orchestra's inherent acoustic characteristics are strongly reflected as high-level features. However, the specific cause of this difference cannot be identified at the present time. The same country of orchestra consistently ranked third in clustering performance for both symphonies. However, a comparison of the DBI showed that Tchaikovsky's clusters (4.875) had a higher performance than Beethoven's (5.809), means that this grouping factor explains the similarity in Tchaikovsky's performances more effectively. This result suggest a tendency for country-based similarity to be higher in Tchaikovsky's Symphony, which has a distinctive Russian musical character.

5.3 Comparison of Results by Movement

We compare the four movements of Tchaikovsky's symphony. Regarding the acoustic features, the similarity of performances clustered by the same conductor (all features) was high across all movements. Furthermore, the top-ranking elements of tempo/loudness features were consistently factors related to the conductor, such as the same conductor or the same nationality of conductor. This result substantiates that tempo and loudness are strongly controlled by the conductor rather than the orchestra.

In terms of movement-specific characteristics, factors related to the conductor ranked highly in Movements 1, 3, and 4, whereas Movement 2 was unique in that the top two elements were same orchestra (timbre and all features). This suggests that because Movement 2 is a piece that includes many solos and small ensemble sections the orchestra's inherent timbre and ensemble quality, rather than the conductor's instructions, strongly determine the performance's individuality. Additionally, while Movements 1, 2, and 4 had all features or timbre ranking highly, Movement 3 had tempo / loudness listed in three of the top five positions. This indicates that Movement 3, being a light waltz, proceeds at a flowing, relatively constant fast tempo with little rhythmic fluctuation, likely resulting in small differences in tempo and loudness across performances.

Next, concerning the VGGish features, the same conductor ranked first in Movements 1 and 2, while the same orchestra ranked first in Movements 3 and 4. Movements 1 and 2 are characterized by a prevalence of minor keys, a calm tempo, and a deep, heavy sound, while Movements 3 and 4 are energetic, bright, and expansive. It is uncertain whether this musical difference relates to the VGGish results.

Furthermore, calculating the average of the top three VGGish results for each movement yielded 4.40, 4.20, 3.58, and 4.58 for Movements 1 through 4, respectively, indicating that Movement 3 had the highest cluster performance. This suggests that the musical difference - where Movements 1, 2, and 4 are emotional and dramatic, while Movement 3 is light and dance-like—is reflected in the DBI values, indicating that expression is more easily standardized in the latter.

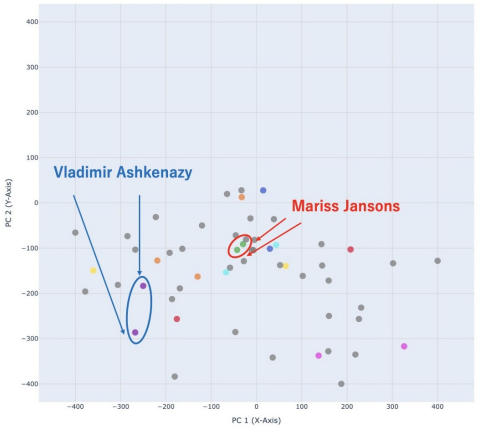
6. ANALYSIS (VISUALIZATION)

Next, we introduce the analysis results from the visualization system. The visualization system is used to verify the concrete similarity between specific performance groups. This is because even if a particular group achieves high cluster performance as evaluated by the DBI, the DBI value alone cannot determine whether that performance is uniform across all elements within the group or varies among elements.

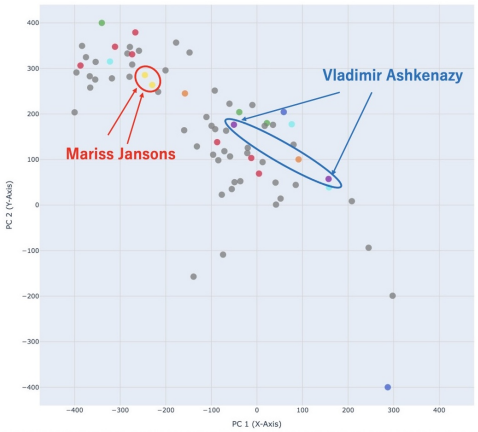
Focusing on the acoustic features, we analyze the similarity among individual conductors on the visualization screen for the "All Features" group classified by "Same Conductor", as this grouping yielded high cluster performance in terms of DBI across both symphonies. We first focused on the performances of Ken-ichiro Kobayashi, who has seven recordings of Tchaikovsky's symphony. Observing the performances of the 4th movement of Tchaikovsky's Symphony, the recordings were clearly separated into two clusters. The performances clustered tightly in the upper left were recordings by Japanese orchestras, while the cluster on the right consisted entirely of recordings by European orchestras. This suggests that even performances by the same conductor may show different similarities depending on the orchestra's country.

We also compared the performances of conductors Mariss Jansons and Vladimir Ashkenazy, both of whom have multiple recordings of both symphonies. For Jansons' performances, the two audio sources of Beethoven were recordings with the same orchestra, while the two audio sources of Tchaikovsky were recordings with different orchestras. While the similarity is naturally high between performances by the same orchestra, a very high similarity was also observed even between performances with different orchestras. Conversely, Ashkenazy's two recordings for both symphonies were with different orchestras (Philadelphia Philharmonic Orchestra and NHK Symphony Orchestra for both symphonies). In both symphonies, the coordinates of the two performances were positioned relatively far apart, confirming that they exhibit different performance styles. There are two potential reasons why a conductor might perform differently depending on the orchestra. The first is the possibility that the conductor intentionally changes the performance style based on the orchestra's characteristics. The second is the possibility that the conductor gives similar directions, but the orchestra's response differs, resulting in a different performance. In the case of Ashkenazy's recordings, since the comparison is between American (Philadelphia

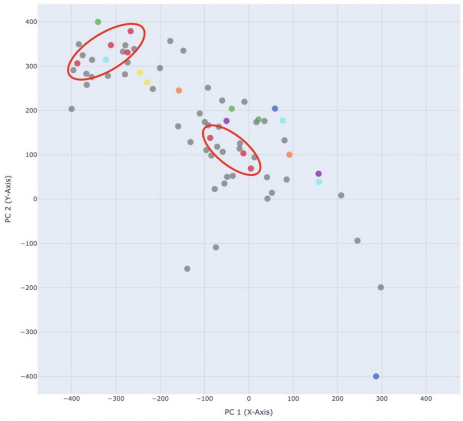
Figure 6: Visualization screen for acoustic features (all features)



(a) Difference of Multiple Recordings by Ashkenazy and Jansons in Beethoven's Symphony No. 7, Movement 1



(b) Difference of Multiple Recordings by Ashkenazy and Jansons in Tchaikovsky's Symphony No. 5, Movement 4



(c) Two Clusters of Kenichiro Kobayashi's Performances in Tchaikovsky's Symphony No. 5, Movement 4

Philharmonic) and Japanese (NHK Symphony) orchestras, it is also possible that the inherent characteristics of the orchestras differ, but determining the primary cause requires expert opinion.

7. CONCLUSIONS AND FUTURE WORK

This study analyzed the factors influencing performance similarity among performances of the same piece by the same orchestra. This study offers a unique contribution to the field by using a deep learning model (VGGish) in addition to conventional acoustic features for the extraction of features. Furthermore, our originality is demonstrated not only by the extensive verification of its applicability across a diverse and expanded dataset of recordings but also by its transition beyond the mere identification of similarities to a rigorous investigation of the underlying causal relationships behind them. Additionally, this study is relevant to MIR evaluation by demonstrating how clustering quality metrics can be used to assess the explanatory power of different feature representations and metadata factors in performance analysis, beyond retrieval accuracy alone. It contributes to explainable music recommendation by identifying interpretable factors, such as conductor and orchestra identity, that systematically shape performance similarity, enabling recommendation systems to justify suggestions using musically meaningful criteria rather than opaque similarity scores. Finally, the findings are pertinent to performance pedagogy and analysis, as they provide empirical evidence of how stylistic traits are transmitted and differentiated across conductors, orchestras, and movements, offering data-driven insights that complement traditional musicological and pedagogical perspectives.

We obtained several findings through the current analysis. First, although the similarity was generally high within the Same Conductor group for both VGGish and acoustic features, this did not apply to all conductors, and variability in similarity was observed among individual conductors. Furthermore, a piece with clear musical structures and established performance styles, such as from the Classical period, showed generally high similarity, whereas piece allowing for a wider range of expression exhibited lower similarity. We also note that the similarity was high for the Same Orchestra group in passages where ensemble precision is crucial. However, some phenomena observed cannot be fully explained at this stage. An example of this is the comparison of VGGish features across movements in Tchaikovsky's Symphony: the group showing the highest similarity was the Same Conductor for Movements 1 and 2, but shifted to the Same Orchestra for Movements 3 and 4. The interpretation of such complex phenomena requires

the opinion of experts, such as professional conductors.

There are two main challenges for future work. The first is to increase the amount of metadata used for analysis. We currently measured similarity based on five metadata items, but potential additions include the place where the conductor studied. This is because we believe that differences in the place of study may create differences in musical schools of thought, which in turn lead to differences in performance styles.

The second challenge is the verification of the impact of non-uniform recording environments on features, and devising countermeasures for this non-uniformity. To promote the uniformity of the recording environment, we limited our analysis to performances recorded after the year 2000, excluding those from eras when recording technology was underdeveloped. As shown by the study by Yanchenko [8], it is well-known that significantly old recordings cause changes in timbre features, but the extent to which recording environments differ among comparatively new performances affects the features is currently unverified. If the difference proves substantial, countermeasures for the non-uniformity of the recording environment will be necessary. As demonstrated by the research of Patynen et al. [13], where differences in hall acoustics are known to influence audience perception of dynamics and power, the acoustic environment is important. Although technical approaches to address this non-uniformity is the acoustic matching method [14], which naturally transforms audio to sound as if it were recorded in a different acoustic environment. We suppose that this issue could be resolved by unifying the acoustics using such a technology before feature extraction. However, implementation is currently difficult because no corresponding model tailored for classical music exists at this time.

Going forward, we aim to conduct the same analysis on different musical pieces to verify whether the current results and trends are consistent, or if new tendencies emerge. For example, we would like to investigate whether clustering performance based on metadata such as the orchestra's country of affiliation or the conductor's country of origin becomes higher when analyzing works by composers such as Dvořák, Debussy, and Ravel, whose music reflects strong elements of Nationalism or distinct regional characteristics.

Acknowledgments

We want to thank the Japan Student Services Organization through the 2025 Student Exchange Support Program, the Maria de Maeztu Units of Excel-lence Programme CEX2021-001195-M, (MICIU/AEI /10.13 039/501100011033) and the IMPA project PID2023-152250 OB-I00 (MCIU/AEI/10.13039/501100011033/ FEDE)

8. REFERENCES

[1] J. Grekow, "Music Recommendation Based on Emotion Tracking of Musical Performances," in *Recommender Systems for Medicine and Music*. Cham, Switzerland: Springer, Studies in Computational Intelligence, 2021, pp. 167–186.

[2] J. Langner and W. Goebl, "Representing expressive performance in tempo-loudness space," in *Proceedings of the 10th Anniversary Conference on Musical Creativity (ESCOM)*, Liège, Belgium, 2002.

[3] S. E. Dixon, W. Goebl, and G. Widmer, "The Performance Worm: Real time visualisation of expression based on Langner's Tempo-Loudness Animation," in *Proceedings of the 2002 International Computer Music Conference (ICMC)*, Gothenburg, Sweden, 2002, pp. 361–364.

[4] S. T. Madsen and G. Widmer, "Exploring Similarities in Music Performances with an Evolutionary Algorithm," in *Proceedings of the Eighteenth International Florida Artificial Intelligence Research Society Conference, Clearwater Beach, Florida, USA*, I. Russell and Z. Markov, Eds. AAAI Press, 2005, pp. 80–85.

[5] C. S. Sapp, "Comparative Analysis of Multiple Musical Performances," in *International Society for Music Information Retrieval Conference*, 2007.

[6] C. C. S. Liem and A. Hanjalic, "Comparative Analysis of Orchestral Performance Recordings: An Image-Based Approach," in *International Society for Music Information Retrieval Conference*, 2015.

[7] J. Grekow, "Comparative Analysis of Musical Performances by Using Emotion Tracking," in *Foundations of Intelligent Systems (Lecture Notes in Computer Science)*, 2017, pp. 175–184.

[8] A. K. Yanchenko and P. D. Hoff, "Hierarchical multidimensional scaling for the comparison of musical performance styles," *The Annals of Applied Statistics*, vol. 14, no. 4, Dec. 2020.

[9] M. Müller, Y. Özer, M. Krause, T. Prätzlich, and J. Driedger, "Sync Toolbox: A Python Package for Efficient, Robust, and Accurate Music Synchronization," *Journal of Open Source Software*, vol. 6, no. 64, p. 3434, 2021.

[10] "Kunst der Fuge – Free Classical MIDI files," https://www.kunstderfuge.com/, 2002.

[11] D. Tao, "VGGish: Keras Implementation," https://github.com/DTaoo/VGGish.

[12] D. L. Davies and D. W. Bouldin, "A Cluster Separation Measure," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-1, pp. 224–227, 1979.

[13] J. Pätynen, S. Tervo, P. W. Robinson, and T. Lokki, "Concert halls with strong lateral reflections enhance musical dynamics," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 111, no. 12, pp. 4409–4414, 2014.

[14] I. Martin-Salinas, G. Piñero, J. A. Belloch *et al.*, "Enhanced U-Net Architectures for Accurate Room Impulse Response Generation via Differential-Phase Learning," *Journal of Audio, Speech, and Music Processing*, vol. 2025, no. 39, 2025.

Tree Rings: Ecological Memory and Linguistic Traces in an Immersive Dome Composition

Yu-Chia Kuo
Department of Music Research, McGill University
yu.kuo@mail.mcgill.ca

ABSTRACT

Tree Rings is a site-specific dome composition that weaves ecological recordings, linguistic material, and generative 3D forms into a layered audiovisual environment. Granular and spectral processing emphasize microscopic textures, while VBAP spatialization and text-to-3D diffusion produce concentric structures that expand toward landscape-scale processes. Treating environmental sound and language as parallel acoustic and cultural archives, the work frames ecological memory as an immersive, temporally scaled experience, offering a metaphor-driven approach to spatial sound and generative visual design within research-creation contexts.

1. INTRODUCTION

Tree rings, visible concentric layers formed by seasonal vascular growth, encode ecological memory across decades or centuries. These growth rings serve as biological imprints of ecological time, inscribed with variations in width, density, and hue that reflect annual cycles of temperature, rainfall, and environmental stress [1]. Trees thus become living archives, recording not only natural rhythms but also human-induced changes such as urbanization and deforestation. As layered biological forms, tree rings offer a potent metaphor for memory, ritual, and transformation. This stratified temporal structure, where environmental history accumulates as layered traces, finds a sonic parallel in environmental recordings, which similarly capture ecological processes and human activity as acoustic archives.

Environmental recordings have long served as material for electroacoustic and soundscape composition, as well as data-driven sonification and ecoacoustic analysis [2, 3, 4]. Fulldome environments extend these approaches by enabling immersive spatial listening, where multichannel audio and large-scale projection create conditions for environmental storytelling, temporal layering, and collective perceptual engagement [5]. Across these practices, ecological materials appear in diverse roles, from sources for scientific visualization to atmospheric components within narrative and experiential works, demonstrating the broad interpretive range afforded by contemporary spatial media.

Tree Rings extends these possibilities by integrating ecological sound and language within a dome-based spatial framework. The work weaves together three material threads: environmental recordings as traces of natural time, linguistic material as cultural memory, and the dome architecture as a medium for spatial translation. Guided by the metaphor of tree rings, the installation unfolds in two sections that progress from microscopic to macroscopic scales, structuring sound, image, and text as concentric temporal layers. Environmental recordings are processed through granular and spectral approaches to highlight biological rhythms and fine textural detail. In parallel, selected textual phrases centered on trees, landscapes, and memory serve as prompts for generative 3D visual synthesis, producing constellation-like structures tracing across the dome.

This project explores a compositional structure in which biological traces, cultural memory, and immersive technology converge as concentric flows of ecological time. Our work examines the dome as both a site of immersion and a medium for spatializing the relationship between environmental and human time, proposing an experiential mode of ecological listening shaped by translation, resonance, and spatial entanglement.

Copyright: ©2026 Yu-Chia Kuo. This is an open-access article distributed under the terms of the Creative Commons Attribution License 3.0 Unported, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

2. RELATED WORK

2.1 Environmental Sound, Language, and Cultural Memory

Environmental sound has been framed in acoustic ecology and soundscape studies as part of communicative systems embedded in social and ecological relations [6, 7]. Soundscape ecology extends this view by analysing biophony, geophony, and anthropophony across spatial and temporal scales, positioning recordings as temporal traces of ecological processes and human activity [8]. Collective listening research further emphasizes the contribution of shared auditory experience to situated memory and place-making [9]. Within such contexts, language can also function as sonic and cultural material through its rhythmic and phonetic qualities. Drawing on these perspectives, *Tree Rings* treats environmental sound and language as parallel acoustic archives, reconfigured via spatial and temporal transformation within the dome.

2.2 Immersive Dome Environments as Spatial Frameworks

Hemispherical full-domes have evolved from specialized visualization platforms into immersive environments that support complex perceptual and collective experiences, characterized by curved projection, wide fields of view, and co-present audiences [10, 11]. In artistic and ecological



Figure 1. Performance documentation from *Tree Rings* at the Satosphère, Société des arts technologiques (SAT), Montréal. Photographed by Ash KG.

contexts, domes and planetaria have been used to combine multichannel sound with large-scale imagery for environmental storytelling and public engagement [12, 5]. Large spherical infrastructures such as the Allosphere further demonstrate how spatialized audio and data-driven imagery can render scientific or artistic material as inhabitable virtual worlds [13]. Within this lineage, our work approaches the dome as a site-specific spatial framework in which sound and generative imagery unfold as layered, experiential forms.

2.3 Generative Systems in Audiovisual Composition

Generative and algorithmic approaches in electronic arts provide frameworks in which rule-based processes, stochastic behaviors, and varying degrees of autonomy shape evolving sonic and visual forms [14, 15]. In performance contexts, live coding foregrounds processual authorship and the dynamics of human–machine co-creation, with an emphasis on artists' real-time engagement with generative structures [16]. Recent AI-driven audiovisual performances extend these paradigms by incorporating machine agents that respond to environmental or linguistic input, thus forming multimodal, co-adaptive compositional systems [17, 18].

Within this landscape, text-conditioned generative methods offer ways to transform linguistic material into spatial or visual structures as interpretive, culturally inflected cues. Our project employs linguistic input in this manner, generating 3D constellations that embody mnemonic traces integrated with processed environmental sound. The resulting audiovisual forms serve as expressive artifacts contributing to the work's temporal and cultural layering.

3. TECHNICAL AND ARTISTIC DESIGN

Tree Rings was developed as a site-specific immersive work for the Satosphère dome at the Société des arts technologiques (SAT) in Montréal. This section outlines the sound materials, sonic structuring, spatialization strategy, and AI-mediated visual system, in which ecological sound and linguistic memory are organized into intertwined layers within the dome environment.

3.1 Sound Materials

The piece is built from environmental recordings and synthesized textures designed to mirror different layers of ecological time. Source material includes boreal and temperate forest soundscapes from northern Canada and Quebec, as well as experimental recordings made with geophones, omnidirectional seismic sensors attached to large pieces of dead wood (Figure 2). These sensors capture low-frequency vibrations transmitted through the material, including subtle friction, internal stress, and structural resonances that are typically absent in conventional air-coupled recordings. Additional sessions involved gently striking and rubbing the wood to produce transient gestures, which were subsequently stretched, fragmented, and recombined employing granular and spectral processing.

3.2 Sonic Structure and Temporal Scaling

The work, approximately eight and a half minutes in duration, is structured into two sections that trace a shift from microscopic to macroscopic scales of ecological time. This structural progression mirrors the tree-ring metaphor: mov-



Figure 2. Experimental recordings captured using geophones attached to large pieces of dead wood.

ing from interior, cellular processes toward expansive forest and atmospheric environments.

In the opening section, the focus is on microscopic activity. Short fragments of the environmental recordings are processed through granular and spectral techniques, with grain size, density, and playback position continuously modulated. The resulting clouds of sound foreground fine details such as crackling, rustling, internal knocks, and small-scale motion that loosely evoke processes such as cell division in the cambium layer or the outward formation of new rings. Low-frequency components captured by geophones anchor these textures, creating the impression that sound emerges from within the wood.

A transition passage forms the work's main point of intensification. Granular textures increase in density and agitation before merging with storm-like forest recordings. This buildup gradually dissolves into a quieter field recording of flowing water, marking the perceptual shift into the second section. The second section expands to macroscopic ecological scales. Materials from the earlier recordings are reorganized into broader ambient layers, with forest soundscapes, resonant swells, and slowly evolving spectral clusters that imply canopies, weather systems, or distant infrastructural presence. These evolving textures articulate a sense of ecological time unfolding outward, from interior growth to landscape-level processes.

3.3 Spatialization Strategy in the Satosphère Dome

Tree Rings was composed for the Satosphère, an 18-metre fulldome equipped with a 93-loudspeaker multichannel system and five low-frequency channels. Spatialization is implemented using Vector Base Amplitude Panning (VBAP) with the SpatGRIS/ServerGRIS spatialization system [5]. Virtual sources are positioned in three dimensions and rendered over loudspeaker triplets distributed across multiple vertical rings. Trajectories and parameter automation are programmed in Ableton Live and transmitted via Open Sound Control (OSC), enabling real-time control of azimuth, elevation, distance, and diffusion. Figure 3 illustrates the spatialization setup, including the loudspeaker layout as visualized in ServerGRIS and the SpatGRIS interface used for spatial trajectory control in Ableton Live.

The spatial layout extends the work's temporal scaling.

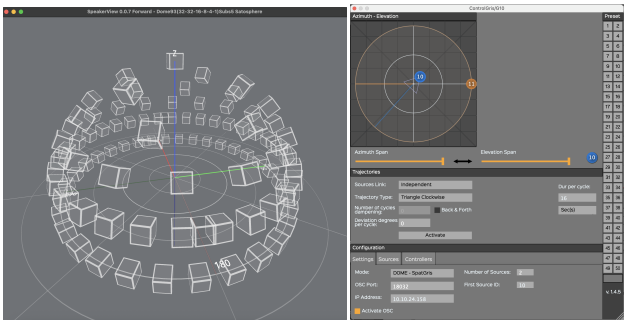


Figure 3. Spatialization setup for the Satosphère dome. Left: Loudspeaker layout of the Satosphère dome as visualized in ServerGRIS. Right: SpatGRIS interface in Ableton Live for spatial trajectory control.

Textures associated with microscopic activity remain compact and inward-oriented, articulated as small-radius trajectories and subtle elevation changes. As the composition expands toward macroscopic ecological scales, spatial gestures widen: decorrelated layers are dispersed across the dome's stacked rings, forming broad ambient bands and slowly rotating clusters. Motion remains generally smooth and unhurried, supporting an enveloping sound field suited to the audience's reclining listening posture.



Figure 4. Dome projection of AI-generated three-dimensional forms produced through text-to-3D diffusion. Photographed by Ash KG.

3.4 Visual Synthesis and Audiovisual Coupling

Text input in any language is first normalized via automatic translation and then processed with OpenAI Shap-E, a text-to-3D diffusion model, to generate AI-driven three-dimensional forms projected across the dome (Figure 4). Rather than serving as a semantic representation, language functions as a generative trigger, with AI-generated assets emerging from interaction within the system's generative logic. Audiovisual coupling is achieved through real-time audio analysis implemented in Max/MSP, where spectral and band-energy descriptors are mapped to visual parameters governing motion, density, and transformation. Continuous parameter modulation is combined with threshold-based triggering, allowing the visuals to reflect both gradual changes and discrete events in the sonic field. Language, sound, and generative imagery operate as parallel material layers that converge into a layered audiovisual environment of ecological and cultural memory.

4. DISCUSSION

Composing for the Satosphère dome highlighted several challenges specific to VBAP-based spatialization in large-scale three-dimensional environments. While horizontal motion was rendered with clear spatial continuity, elevation-based trajectories were perceived as less salient, particularly in comparison to prior experiences with higher-order Ambisonics (HOA) systems [19, 20, 21]. This effect can be attributed to the discrete nature of VBAP elevation rendering, in which vertical motion is approximated by successive transitions among loudspeaker triplets distributed across multiple vertical rings. In such configurations, changes along the Z-axis are often perceived as variations in spatial spread rather than explicit vertical displacement.

These constraints directly informed the spatial organization of sound materials. Audio tracks were classified based on spectral content, temporal density, and dynamic behavior, and spatialized using distinct strategies to reduce masking and maintain perceptual continuity. Broadband and textural sounds were preferred for elevation-based motion, while more articulated trajectories were assigned to horizontal movement. Trajectories were composed conservatively, emphasizing slow, continuous motion to avoid perceptual discontinuities caused by triplet switching [21]. Audience posture also influenced spatial perception, as the reclined listening configuration of the Satosphère likely reduced sensitivity to vertical localization cues. In the context of the Labos Immersifs residency, the audience comprised primarily artists, researchers, and technologists familiar with immersive environments.

5. CONCLUSION

Tree Rings demonstrates how ecological sound, linguistic memory, and immersive dome environments can be integrated into a layered compositional structure shaped by metaphor. By treating environmental recordings and language as parallel acoustic and cultural archives, the work proposes an experiential mode of ecological listening rooted in temporal scaling and spatial translation. Future development includes adapting the piece into a fixed-media Ambisonic version and refining the text-to-3D visual synthesis pipeline for broader use in research-creation contexts involving cultural archives, spatial sound, and generative audiovisual systems. Additionally, we plan to examine audience responses through structured observation and post-experience interviews, focusing on the relationship of listeners' cultural backgrounds and ecological dispositions to their perceptions of immersive environmental narratives.

Acknowledgments

Tree Rings was created collaboratively by Yu-Chia Kuo and Florence-Delphine Roux (sound design), and Time Wei and Pia Vidal (visual design). Special thanks to Time Wei for providing details on the visual synthesis pipeline discussed in Section 3.4. This work was produced during the Labos Immersifs residency at the Société des arts technologiques (SAT), Montréal, with support from Arts hybrides et cultures numériques (AADN).

6. REFERENCES

[1] H. Fritts, *Tree rings and climate*. Elsevier, 2012.

[2] P. Vickers, "Sonification and music, music and sonification," in *The Routledge companion to sounding art*. Routledge, 2016, pp. 135–144.

[3] A. De Campo, "Toward a data sonification design space map," in *Proceedings of the international conference on auditory display*, 2007, pp. 342–347.

[4] A. Zanella, C. Harrison, S. Lenzi, J. Cooke, P. Damsma, and S. Fleming, "Sonification and sound design for astronomy research, education and public engagement," *Nature Astronomy*, vol. 6, no. 11, pp. 1241–1248, 2022.

[5] R. Normandeau, O. Bélanger, C. Lengelé, and D. Ledoux, "SPATGRIS/SERVERGRIS, Creative Tools for 2D and 3D Sound Spatialization." in *ICMC*, 2018.

[6] B. Truax, "Soundscape, acoustic communication and environmental sound composition," *Contemporary music review*, vol. 15, no. 1-2, pp. 49–65, 1996.

[7] H. Westerkamp, "Linking soundscape composition and acoustic ecology," *Organised Sound*, vol. 7, no. 1, pp. 51–56, 2002.

[8] B. C. Pijanowski, A. Farina, S. H. Gage, S. L. Dumyahn, and B. L. Krause, "What is soundscape ecology? An introduction and overview of an emerging new science," *Landscape ecology*, vol. 26, no. 9, pp. 1213–1232, 2011.

[9] K. Norman, "Listening together, making place," *Organised Sound*, vol. 17, no. 3, pp. 257–265, 2012.

[10] S. Schnall, C. Hedge, and R. Weaver, "The Immersive Virtual Environment of the digital fulldome: Considerations of relevant psychological processes," *International Journal of Human-Computer Studies*, vol. 70, no. 8, pp. 561–575, 2012.

[11] C. Hedge, R. Weaver, and S. Schnall, "Spatial learning and wayfinding in an immersive environment: the digital fulldome," *Cyberpsychology, Behavior, and Social Networking*, vol. 20, no. 5, pp. 327–333, 2017.

[12] K. C. Yu, K. Williams, D. Neafus, L. Gaston, and G. Downing, "Gaia Journeys: a museum-based immersive performance exploration of the Earth," *International Journal of Digital Earth*, vol. 2, no. 1, pp. 44–58, 2009.

[13] J. Thompson, J. Kuchera-Morin, M. Novak, D. Overholt, L. Putnam, G. Wakefield, and W. Smith, "The Allobrain: An interactive, stereographic, 3D audio, immersive virtual world," *International Journal of Human-Computer Studies*, vol. 67, no. 11, pp. 934–946, 2009.

[14] A. Dorin, "Generative processes and the electronic arts," *Organised Sound*, vol. 6, no. 1, pp. 47–53, 2001.

[15] N. Collins, "Generative music and laptop performance," *Contemporary Music Review*, vol. 22, no. 4, pp. 67–79, 2003.

[16] A. R. Brown and A. Sorensen, "Interacting with generative music through live coding," *Contemporary Music Review*, vol. 28, no. 1, pp. 17–29, 2009.

[17] K. Tatar, P. Pasquier, and R. Siu, "REVIVE: An audiovisual performance with musical and visual AI agents," in *Extended abstracts of the CHI conference on human factors in computing systems*, 2018, pp. 1–6.

[18] C. Erdem, B. Wallace, and A. R. Jensenius, "CAVI: A coadaptive audiovisual Instrument–Composition," in *International Conference on New Interfaces for Musical Expression*. PubPub, 2022.

[19] F. Zotter and M. Frank, *Ambisonics: A practical 3D audio theory for recording, studio production, sound reinforcement, and virtual reality*. Springer Nature, 2019.

[20] L. Gandemer, G. Parseihian, C. Bourdin, and R. Kronland-Martinet, "Perception of surrounding sound source trajectories in the horizontal plane: A comparison of VBAP and basic-decoded HOA," *Acta Acustica united with Acustica*, vol. 104, no. 2, pp. 338–350, 2018.

[21] G. Marentakis, F. Zotter, and M. Frank, "Vector-base and ambisonic amplitude panning: A comparison using pop, classical, and contemporary spatial music," *Acta Acustica united with Acustica*, vol. 100, no. 5, pp. 945–955, 2014.

The History of Japanese Electroacoustic Music for Piano from the Perspective of Media Genealogy

Hyunmook Lim
Tokyo University of the Arts
s2323914@ms.geidai.ac.jp
lim21@iamas.ac.jp

ABSTRACT

This paper examines the history of compositions for piano and electronics in Japan through the lens of media genealogy. While the development of modern Japanese electronic music emerged nearly in parallel with its European counterparts, it has often been perceived as lacking a distinctive trend or unified stylistic coherence, unlike the established traditions of France's Musique concrète or Germany's Elektronische Musik. To address this, the author categorizes the historically inconsistent trajectory of Japanese electronic music by focusing on works for piano and electronics, tracing the genealogy of specific media that have emerged within the Japanese context. In response to the ICMC2026 theme, "Innovation, Translation, Participation," this study provides a detailed analysis of technological innovation through media genealogies, offers a new translation of this historical narrative, and explores the processes of artistic participation that have shaped Japan's electronic music history.

1. INTRODUCTION

The period between the 1940s and 1950s represented a critical turning point for electronic music, which was predominantly centered in Europe and America. The advent of tape recorders and oscillators, along with the establishment of professional electronic music studios dedicated to experimental exploration, transformed electronic music into a sustainable and significant artistic practice with enduring creative possibilities. Following the pioneering efforts of European institutions such as Pierre Schaeffer's Studio d'essai in France and the WDR Studio für Elektronische Musik in Germany, these practices were promptly introduced to Japan.

Notably, in 1953, following his return from studies in Paris, Toshiro Mayuzumi (黛敏郎) composed Japan's first musique concrète work, "X, Y, Z". In 1954, Mayuzumi further introduced the concepts of the WDR Studio, accompanied by Makoto Moroi (諸井誠)'s translation of articles by Robert Beyer and Herbert Eimert [1]. Since then, numerous Japanese compositions inspired by European contexts have emerged. While early works primarily utilized reel-to-reel tape manipulation and oscillator-generated sounds, composers gradually began incorporating traditional acoustic instruments into their electronic compositions for public performance[1].

In the 21st century, the composition and performance of electroacoustic works are no longer a novelty, and an increasing number of performers now include such works in their core repertoire. However, in contrast to Euro-American scholarship, active research on the history and repertoire of domestically composed electroacoustic music remains limited in Japan. Although some studies exist, the field has yet to develop a comprehensive scholarly discourse or achieve sufficient international recognition. To address this gap, the author aims to capture the historical characteristics of Japanese electroacoustic music for piano by categorizing significant works through a genealogy of specific media.

The rationale for focusing on the piano is twofold. First, it stems from the author's background as a trained classical pianist. Second, there is a scholarly conviction that analyzing works for the piano—the quintessential instrument of European-American classical music—constitutes an essential first step toward a comprehensive understanding of electroacoustic music history. Early landmark compositions such as John Cage's "Imaginary Landscape No. 1" (1939), Pierre Schaeffer's "Études de bruits" (1948), and Karlheinz Stockhausen's "Kontakte" (1958–60), all of which incorporate the piano, serve as compelling evidence supporting this focus.

This paper examines the comprehensive history of Japanese electroacoustic music for piano, encompassing both the pre- and post-emergence eras of computer music. Rather than interpreting this history as a single, linear progression, this study aims to reconstruct it through multifarious chronological timelines shaped by the various media that have emerged throughout Japan's history. By adopting this multidimensional perspective, the author seeks to re-evaluate the perceived lack of a distinctive trend or unified stylistic coherence in Japanese compositions—framing it not as a deficiency, but as a complex manifestation of media genealogy. This perspective will be further explored in the Discussion section.

Copyright: © 2025 Hyunmook Lim. This is an open-access article distributed under the terms of the Creative Commons Attribution License 3.0 Unported, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

[1] Prior to 1953, an early electroacoustic experiment was conducted by Saburo Tominaga (富永三郎), as exemplified by his work "Passacaglia" (1952). However, this individual endeavor did not lead to a broader establishment of electroacoustic music composition in Japan at the time.

2. MEDIA AND WORKS

2.1 Transforming Piano Sound

2.1.1 Ring Modulator

It is reasonable to claim that the history of Japanese electronic music began with the establishment of the NHK Electronic Music Studio. While the studio operated in a provisional phase until 1955, it was officially founded in 1956 by repurposing the observation room situated at the rear of the audience in the former NHK Hall [2]. Through active collaboration between engineers and composers, the NHK Electronic Music Studio emerged as a focal point for electronic music in Japan, producing a substantial number of significant works in the history of Japanese electronic music.

One notable feature of the NHK Electronic Music Studio was its collaborative production method, which involved close cooperation among composers, producers, and engineers. Composers proposed ideas for their compositions and consulted with engineers on how to realize the desired sounds. Following these consultations, engineers custom-built equipment specifically for a work based on each composer's requests through ongoing discussions [3].

In 1967, in partnership with the engineer Shigeru Sato (佐藤茂), Toshiro Mayuzumi developed a specially modified piano called the "Multified Piano (マルチピアノ)". The term "Multified Piano" refers collectively to the piano itself along with various auxiliary devices designed to modulate the piano's sound in real time. This setup involved installing pickup microphones on each string of the piano, which were connected to equipment such as ring modulators, tape delays, and variable filters to modulate the sound. Mayuzumi utilized this "Multified Piano" to compose "Campanology for Multified Piano (マルチピアノのためのカンパノロジー)". This work aimed to modulate the piano sound into a timbre inspired by a bell [4], with the two ring modulators playing a particularly crucial role in achieving this effect.

Although Multified Piano was destroyed the following year, the ring modulator continued to be frequently employed to modulate piano sounds in early live electronics music in Japan. In 1969, Hidenao Ito (伊東英直) composed "Anneau d'Intégrite (整域)". This work was composed using graphic notation, designed for a single piano to be interpreted freely and improvisationally. At the time of its concert premiere in Tokyo, real-time modulation of the piano sound through a ring modulator was performed simultaneously [5]. While "Campanology for Multified Piano" was premiered privately within the NHK Electronic Music Studio, a distinguishing feature of "Anneau d'Intégrite" was the utilization of a ring modulator during a public concert performance. In the same year, Mieko Shiomi (塩見允枝子) also adopted the ring modulator to modulate the sounds of three pianos in her composition "Amplified Dream 1 (夢の増幅 1)" [6]. On the other hand, with the advent of new media such as the three-head tape recorder or electronic piano, the utilization of ring modulators gradually declined starting in the 1970s.

2.1.2 Tape Recorder

In 1950, Tokyo Telecommunications Engineering Corporation (東京通信工業社), the predecessor of Sony, released Japan's first commercially available tape recorder, the "Type G (G 型)" [7]. However, electroacoustic works for piano and electronics utilizing tape recorders remained scarce at that time.

The first significant Japanese work in this genre was "Concertant 4 Movements with Piano and Electronic Sound (ピアノと電子音響に依る協奏的四章)" (1958) by Hideko Ohki (大木英子), featuring electronic sounds produced by the sound engineer Matsuo Ono (大野松雄). In this piece, sound samples such as the striking of an iron plate with a hammer were recorded and then manipulated by altering their playback duration and speed. "Concertant 4 Movements" represents Japan's earliest attempt to integrate electroacoustic techniques—utilizing both oscillator-generated sounds and the methodologies of musique concrète—with live piano performance.

The tape recorder began to demonstrate more diverse capabilities around 1972. That year, Technics released the world's first three-head tape recorder, the RS-279U [8], followed in 1973 by a series of three-head models including the Nakamichi 1000 (Nakamichi Research Institute), the TC-6150SD (Sony), and the D-4500 (Lo-D). Unlike existing two-head models, which combined recording and playback functions into a single head, the three-head design separated these functions. This innovation allowed for simultaneous playback of the sound being recorded, enabling users to monitor the recording in real time.

The composer who promptly exploited the characteristics of the three-head tape recorder was Somei Satoh (佐藤聰明). He composed "Litania" in 1973, immediately following the introduction of these models. Satoh conceived a work featuring rapid tremolo techniques played with both hands, ranging from single tones to clusters. To achieve a smoother tremolo effect than humanly possible, Satoh utilized the three-head tape recorder [9]. Due to the fixed distance between the recording and playback heads, the monitored sound is played back a few milliseconds after being recorded. Satoh exploited this feature to create a delay effect, playing back the piano sound with a slight real-time lag. Consequently, the delayed sound inserted between the chords played by the pianist produced a more delicate and fluid tremolo.

In 1976, Takashi Satoh (佐藤隆) composed "Fade Away". This piece was written for three pianos, each performing the same score with staggered timing to create a cumulative musical fade-away effect. Alternatively, the parts for the first and third pianos could be pre-recorded and overdubbed, allowing the second piano to perform live alongside the tape. While "Fade Away" primarily involved pre-recorded playback, it can be interpreted as a fixed-media version of the methodology employed by Somei Satoh in "Litania", where the tape recorder was used to delay the piano performance in real time. Furthermore, "Fade Away" allows for considerable performance flexibility. The work can be realized not only with pianos but also with any percussion instruments or diverse forms of electronic media.

2.1.3 Computer

In Japan, the development of electronic computers began early, as exemplified by the completion of FUJIC in 1955, the nation's first electronic computer [10]. Despite this early progress, the utilization of electronic computers for musical composition was not yet a subject of discussion during the 1950s. In the 1960s, influenced by the "ILLIAC Suite", Tomoaki Sekine (関根智明) and Taiga Hayashi (林大雅) composed "TOSBAC Suite" (1962) using Toshiba's TOSBAC computer [11]. However, computer-aided composition had yet to become widespread in Japan at that time.

This landscape shifted with the advent of personal computers. Following the commercial success of the Apple II in 1977, personal computers began to see widespread adoption from the late 1970s onward. In the 1980s, Japanese composers studying abroad started to explore composition using these systems. Masahiro Miwa (三輪眞弘), while active in Germany, composed "Duoklav II" in 1987—marking the first work by a Japanese composer for piano and computer. This piece is a duo between a piano and computer-generated sounds. During performance, the musician operates a program developed by Miwa on an Atari 260ST computer while simultaneously playing the piano. In 1988, Takayuki Rai (莱孝之), while researching electronic music in the Netherlands, composed "Four Inventions" by utilizing features such as frequency modulation and ring modulation on the DMX-1010 signal processing computer.

In 1990, the programming language Max from IRCAM was commercialized. In 1992, Suguru Goto (後藤英) composed "Transformation II", the first piece by a Japanese composer for piano and electronics developed within the Max environment. In this work, two tam-tams were placed around the piano to capture resonances through microphones; these sounds were then spatially moved across four speaker channels utilizing Max's MIDI control functions.

Although a limited number of Japanese composers had already utilized IRCAM's ISPW environment and Max, it was especially after the development of MSP in 1997 that computer music composition using Max became widespread across Japan. Since the integration of MSP, computers have become a central and representative medium for facilitating experimental electroacoustic music, primarily centered around the Max/MSP environment.

From the late 1990s onwards, a variety of tools for composing music for piano and electronics emerged beyond Max. In Japan, a new generation of composers began adopting these alternative approaches in the 2010s. For instance, in 2013, Chiku Komiya (小宮知久) composed "Existence No.√-x" utilizing Pure Data, while Toshiya Watanabe (渡辺俊哉) collaborated with Takumi Ikeda (池田拓実) to compose "Vanishing Point" utilizing SuperCollider. Over time, the methods of computer music composition have become increasingly diverse. Composers no longer rely on a single environment; rather, they select and employ compositional approaches that best suit their individual needs and creative contexts.

2.2 Expanding Piano

2.2.1 Electric/Electronic Piano

Attempts to extend the traditional sound of the piano using electricity, such as the Clavecin électrique (1759), have existed since before the 20th century. However, the introduction of electric pianos in Japan did not occur until the 1960s. In 1962, Nippon Columbia released the first Japanese electric piano, the "Elepian (エレピアン)" [12]. Despite this, electronic music composition utilizing electric pianos remained uncommon throughout the 1960s.

Later, in 1973, Roland launched Japan's first electronic piano, the EP-10, which allowed performers to switch between piano and harpsichord sounds [12]. The advent of the EP-10 facilitated the creation and performance of sophisticated works utilizing electronic circuits to expand tonal possibilities, significantly advancing electroacoustic music composition in Japan.

In the same year, 1973, Shigeaki Saegusa (三枝成彰) promptly composed "∞⌒N=1 Vn (Pn) a∋P０[2]". This work employs a piano, a prepared piano, a combo organ, and an electronic piano, arranged so that the four keyboard instruments surround the performer. The sound of each instrument is amplified via microphones and spatially reproduced through six loudspeakers. Saegusa's work is a representative example of early experiments combining the piano with electronic instruments during the formative period of the electronic piano.

In 1974, Roland released the "EP-30", the world's first electronic piano equipped with touch sensitivity [12]. This enabled the control of dynamics solely through the per-former's touch, allowing for richer musical expression. Nevertheless, for a while, composers continued to utilize both electric and electronic pianos according to their specific needs. For example, Yori-Aki Matsudaïra (松平頼暁) composed "Erixatone" in 1979, which utilized three electric pianos and clavinets, each employing a different tuning system. To create a complex sonic texture, each instrument was precisely detuned—specifically to -1/3 tone, well-temperament, and +1/3 tone—and performed simultaneously.

Throughout the late 20th century, electronic pianos evolved from simple analog circuits to digital, MIDI-compatible systems. This progression enabled the instrument to demonstrate creative potentials surpassing those of the traditional piano. For example, in 2003, Yoshibumi Fujiwara (藤原嘉文) composed "Vision II" for electronic piano. This work utilized the sequencer function of the electronic piano, which was described as "impossible to perform on a conventional piano" [13]. The composition aimed to "simultaneously advance a highly mechanical ostinato and expressive real-time performance" [13], embodying a coexistence between machine and human. Although electric and electronic pianos originally emerged as extensions of the acoustic piano, they continue to evolve as media that constantly redefine their own functional boundaries.

[2] In European-American contexts, this work is also commonly known as "Baire's Theorem".

2.2.2 MIDI Keyboard

In 1983, the commercial success of Yamaha's DX7, alongside the miniaturization of digital synthesizers and the introduction of the MIDI standard, marked the beginning of the era of MIDI keyboards. Although the advent of MIDI keyboards allowed for more experimental usage of electronic instruments, few documented works in Japan initially made serious utilization of MIDI keyboards. Composer Yuriko Hase Kojima (小島有利子) analyzed this phenomenon, attributing it to the fact that contemporary composers primarily perceived MIDI-equipped electronic instruments as tools for popular music; consequently, they demonstrated limited interest in applying MIDI technology to contemporary classical music composition [14].

The utilization of MIDI as a creative medium among Japanese composers began to gain momentum in the 1990s. In 1992, Suguru Goto employed Max to control MIDI signals in "Transformation II", managing speaker channel transitions. During the same year, Masahiro Miwa employed MIDI to control an automated piano performance in "Gesänge des Ostens (東の唄)". Concurrently, Shigenobu Nakamura (中村滋延) composed "Interface Concerto", which utilized a MIDI keyboard, a computer, multiple MIDI sound modules, and a specially developed device called "Hyper N-5". This device was designed to detect the input from the MIDI keyboard and generate an automated performance accordingly [15].

MIDI keyboards began to attract attention not only as "instruments" for performing traditional timbres but also as "controllers" for manipulating diverse electronic signals. Masahiro Miwa and Nobuyasu Sakonda (佐近田展康) formed the duo "Formant Brothers" in 2000. Their project involved synthesizing artificial voices using oscillator-based frequencies, enabling real-time articulation through performance. Initially, they utilized Max patches performed on a MIDI keyboard[3] divided into two zones: the upper register for selecting phonemes, and the lower register for controlling pitch and volume [16]. This configuration allowed for the live performance of artificial speech.

Furthermore, MIDI keyboards were occasionally utilized to control visual information in addition to auditory data. In 2017, Koji Takahashi (高橋宏治) composed "17 Etudes". Developed in collaboration with Ayane Shindo (進藤綾音), this work for video, electronic sounds, piano, and MIDI keyboard explores the intricate relationship between image and sound. For example, in Etude No. 1, "Move Quickly", video elements are displayed or cleared based on keystroke recognition. In Etude No. 6, "Count", specific keys trigger video segments of varying sizes while simultaneously playing synchronized audio samples of numbers being counted in different languages. In Etude No. 14, "Censor Beep", specific keystrokes trigger a visual distortion effect on the video, accompanied by sine wave tones.

[3] The Formant Brothers have recently shifted from using a MIDI keyboard to a MIDI accordion as their primary interface for artificial voice synthesis.

2.2.3 Player Piano

While electric pianos, electronic pianos, and MIDI keyboards expanded the piano's timbre and functionality, these devices generally presupposed performance by a human player. Consequently, the stage almost invariably featured a human performer at the instrument. This paradigm was fundamentally altered by the advent of the player piano.

Early player pianos, which utilized piano rolls and pneumatic mechanisms, reached full electronic realization in 1982 with Yamaha's development of the "PC-1 Piano Player". The PC-1 operated in conjunction with an acoustic piano, utilizing floppy disks for the recording and automatic playback of performance data; it was also capable of being triggered via MIDI signals. In 1987, Yamaha introduced the "Disklavier", which operated independently of floppy disks and featured sensors attached to the hammers, enabling the real-time recording and automated playback of live performances [17]. This digitization of recording and playback allowed for seamless integration with computers, facilitating the utilization of player pianos within the domain of computer music.

The most notable early example of exploiting the characteristics of the digital player piano was by Masahiro Miwa. In 1992, he composed "Gesänge des Ostens" for two player pianos and one pianist. In this piece, one player piano was utilized to output the pianist's live performance as MIDI data, while the other was controlled by computer algorithms to perform pre-recorded or generated MIDI data [18], thereby enabling a collaborative interaction between live and automated performances.

While Miwa's works employed player pianos to extend the possibilities of the instrument and human performance, the automated elements remained fundamentally under the control of the human composer. This technological dilemma found new solutions in the 2020s through the widespread adoption of artificial intelligence. Since 2011, IRCAM has been developing "Somax2", a Max object that utilizes AI to enable generative, improvisational performances without direct human intervention [19]. This development marks a significant advancement toward autonomous musical improvisation by machines.

Since the development team led by Gérard Assayag introduced Somax2 to Japan in 2024, compositional experiments utilizing this technology have emerged within the country. In 2024, Ko Sahara (佐原洸) composed "Antiphonie", a work that juxtaposes a strictly notated score with free improvisation. In this piece, Somax2 learns from the live piano performance to generate improvisations on a player piano. During the latter half of the composition, the performer engages in spontaneous improvisation, fostering a dialogue with the AI. In 2025, Masatsune Yoshio (由雄正恒) composed "Empathic Machines (イシンデンシン)", a work in which a pianist's brainwaves are detected and interfaced with Somax2 to drive improvisation on a player piano. Beyond mere human control, player pianos paired with Somax2 and AI represent an autonomous, "thinking" medium that further extends the boundaries of instrumental and human performance.

2.3 Integrating Piano and Speaker

2.3.1 Piano as an Object

Following Erik Satie's "Musique d'ameublement", there have been numerous attempts to detach the piano from its conventional musical context, allowing it to shift into non-musical domains as a kind of still life or sound object. Through such attempts, the piano loses its traditional significance as an instrument for performing established classical repertoire and is instead recontextualized as a motionless object, akin to those found in still-life painting or installation art. In this manner, the piano becomes an object within non-performative acts, simultaneously functioning as a medium that dissolves the boundaries between music and non-music, as well as performance and non-performance.

Experiments that exclude the piano from its conventional context to treat it as a situated object have long attracted the interest of certain Japanese composers. Takehito Shimazu (嶋津武仁) composed "Territory (領域)" in 1985. This work was scored for "modulated natural sounds and objectified piano," where the concept of the "objectified piano" reflects an intention to treat the instrument not as a traditional musical tool, but as a sound-emitting object. In this piece, the pianist freely arranges and performs 17 fragments utilizing frequency modulation and reverberation. Concurrently, modulated samples of natural sounds, such as rain, are played back through speakers. The piano thus occupies a position equal to that of the speakers, acting as an object that produces sound.

In 1991, Mamoru Fujieda (藤枝守) composed "Piano Cocoon". This work drew inspiration from African-derived instruments such as the kalimba or xylophone, many of which produce noise by resonating with the instruments' natural bodies. For instance, gourds frequently utilized as resonators in these instruments often feature small holes covered with spider-silk membranes, colloquially known as "cocoons" [20]. Fujieda electronically interpreted this concept of the instrument's "membrane" in his composition. During the premiere, the physical movements of the performer were detected via sensors, and the captured signals were converted into electronic sounds by a computer. These sounds were then output through speakers placed directly on the piano strings. Simultaneously, pieces of aluminum foil were placed on the strings, vibrating in response to the speaker-triggered string resonances to produce noise. In this setting, the aluminum foil functions similarly to traditional vibrating membranes, while the piano acts as an object that facilitates these vibrations.

In 2022, Kazutomo Yamamoto (山本和智) composed "Example of Resignation i (放棄の実例 i)". In this work, the pianist performs a notated section of the score and subsequently exits the stage. Following the departure, a Max patch generates electronic sounds randomly in real time for several minutes, outputting them through speakers fixed toward the piano. In response to speaker output, the piano produces sympathetic string resonances, presenting itself as a singular sound object independent of human agency.

2.3.2 Transducer

In Japan, the utilization of transducers[4] in the field of electroacoustic music for the piano began to gain traction from the 1990s onward. This development marked a departure from the traditional electroacoustic model, which regarded the piano and the speaker system as separate and distinct media. By attaching transducers directly to the piano, the instrument and the transducer were conceived as a single integrated unit, thereby transforming the piano into a medium capable of emitting electronic sounds.

In 1999, Masahiro Miwa utilized transducers in his composition "18 sai-no-shi (18 さいのし)" for piano and two pianists. In this work, a laptop computer was placed on the piano to simultaneously display performance instructions to the pianists. The computer also recognized the piano sound and responded by triggering samples of spoken poetry and soprano singing, which were output through transducers attached inside the piano [21]. This setup allowed the piano to act as a resonant body for the electronic sounds, blending human performance with electronic sound sources within a single acoustic instrument.

In 2019, Keita Matsumiya (松宮圭太) composed "Shitatari (したたり)". This work involved attaching two transducers to the soundboard of the piano. Sampled sounds, derived from field recordings of flowing rivers and dripping spring water, were electronically manipulated and played back through the piano's body by triggering vibrations via the attached transducers [22].

In the same year, Kotoka Suzuki (鈴木琴香) composed "Blue, Rising". In this piece, two transducers were attached directly to the piano strings, causing them to vibrate. Suzuki incorporated materials from historical compositions by Maurice Ravel, Franz Liszt, and others. Consequently, the piano and the transducers become integrated as a unified instrument, simultaneously serving as a symbolic reference to the musical past [23].

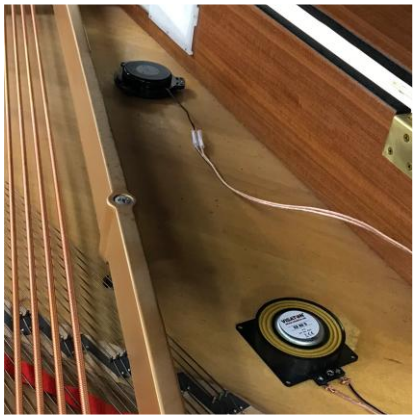


Figure 1. Transducers attached to the soundboard of the Piano for the piece "Shitatari" by Keita Matsumiya [22].

[4] In this paper, the term "transducer" is defined as an electroacoustic device that converts electrical audio signals into mechanical vibrations. These vibrations are transmitted to and radiated by a solid surface, effectively causing the surface to function as a speaker diaphragm. While the device is also known by alternative names such as "audio exciter" or "vibration speaker," this study consistently utilizes the term "transducer" in accordance with its technical specifications.

2.3.3 Wireless Speaker

Attempts to integrate the piano and speaker—such as treating the piano as a sound-emitting object or utilizing transducers—have long persisted in Japan. However, these approaches were often accompanied by a fundamental dilemma: on stage, both the piano and the speaker system remained perceived as separate, distinct media. Not only are conventional speakers visually prominent, but even transducers attached inside the piano are connected by cables, allowing the audience to discern the speaker's presence through the wiring.

To address this dilemma, composers began to devise methods to render speakers as "invisible" as possible by concealing them from the audience's view. In order to focus attention on the integrated acoustics of the piano and speaker, it became essential to prioritize the concealment of cables, thereby preventing the audience from being consciously aware of the electronic sound source.

Compositional practices prioritizing speaker invisibility were pioneered by Yasunoshin Morita (森田泰之進). Beginning in 2014, Morita composed the series "ReincarnatiOn Ring (音／輪)" for traditional instruments and multiple discarded iPods. In 2016, he composed "ReincarnatiOn Ring VI" for piano and electronics played back from a second-hand iPod[5].

At the premiere of "ReincarnatiOn Ring VI", a small wireless Bluetooth speaker paired with a smartphone was hidden around the piano, creating the effect of a "sound with an unknown origin" [24]. This allows the audience to perceive sounds emitting from concealed locations within the venue while the pianist performs. The pre-recorded piano sounds specified in the score are played back through the speaker during the live performance. In this work, the vintage audio equipment is "reincarnated" as an instrument, with the pianist acting as the "mediator" of this reincarnation process [25].

In 2024, Takuma Itoh (伊藤巧真) composed "Sange II-Orzbau_5". At the premiere, electronic sounds were produced by installing Bluetooth speakers inside the piano, which were remotely controlled by the composer from the audience. While Morita's work still required the performer's direct manipulation of the equipment, the remote control via Bluetooth in "Sange II-Orzbau_5" enabled the speakers to be entirely hidden from the audience's field of view, facilitating a more seamless emission of "sound with an unknown origin."



Figure 2. At the time of the premiere rehearsal of "Sange II-Orzbau_5". The Bluetooth speaker was installed inside the Piano. Photograph by the Author.

[5] In this work, the composer also permits the use of various other types of second-hand audio players.

3. DISCUSSIONS

This paper attempted a comprehensive analysis of the fragmented and inconsistently unified history of Japanese electronic music by classifying the media utilized in electroacoustic works for piano and electronic sound according to specific criteria. Through this framework, the study categorized and analyzed historical compositions that employed each of these media types.

As discussed in the introduction, research on the history of Japanese electronic music remains scarce in terms of unified perspectives. The field has often been viewed under the premise that a vast diversity of works exists, which do not conform to singular trends or practices. From the perspective of piano and electronics, the history of Japanese electronic music is better understood not as converging toward a unified movement—as observed in Europe—but rather as a sequence of challenges posed by the emergence of specific media. Individual composers responded to these challenges by creating distinct works based on their unique artistic interests. By examining period-specific media and their corresponding works, it becomes possible to understand how each era in Japan was both reflected in and shaped by the advent of new technologies and their subsequent influence on compositional practice.

While the analysis presented here does not claim to offer an exhaustive summary of the entire history of Japanese electronic music, it successfully organizes and classifies the literature of electroacoustic music for piano and electronics based on the media introduced in each era. Furthermore, this study does not intend to trace Japanese electronic music history as a linear progression driven solely by technological development. Instead, it seeks to offer a multi-perspective view of this history across past and present by reconnecting fragmented or overlooked compositions dispersed over time. By re-examining "lost media" or even "failed media" from a contemporary standpoint, we can illuminate musical values that were underrecognized at the time and trace how their functions have been inherited or transformed in subsequent generations. Moreover, by analyzing various media through diverse case studies, this study draws attention to previously unexplored dimensions of these technologies. By illuminating a single medium or work from multiple angles, the history of Japanese electronic music can be analyzed in a more multidimensional manner.

For instance, Toshiro Mayuzumi's "Campanology for Multified Piano" originally employed a flexible distribution of sound via five loudspeakers. In the 2017 reconstructed performance realized by Hideaki Isobe (磯部英彬), speaker panning was implemented in a circular motion using a DJ controller. This reinterpretation can be understood as an extension of the manual channel panning from the original performance, reimagined through contemporary interfaces such as DJ controllers or MIDI-based technologies. Similarly, the ring modula-

tor—a core medium in the original composition—can be viewed as a precursor to modern implementations through effector programs or virtual studio technologies (VSTs) within digital audio workstations (DAWs). Moreover, the work's use of improvisatory performance with pre-recorded materials anticipated the potential of contemporary AI-driven systems like Somax2, suggesting a continuity between early electroacoustic improvisation and present-day computer-based practices.

In another example, Shigenobu Nakamura's "Interface Concerto" utilized a custom-developed MIDI device, the "Hyper N-5", which generated autonomous performances by continuously varying musical parameters—such as pitch, rhythm, and dynamics—based on pre-recorded numerical data. The automated functions of the Hyper N-5 can be understood as an alternative means to the automated performance of player pianos from the same era, while simultaneously anticipating the development of modern sequencers and Max-based environments. Furthermore, the inclusion of hardware sound modules, such as the "E-mu Proteus/2" and "KAWAI K4r", anticipated the shift toward software-based environments such as VST, marking a transition to the simplified yet high-quality soundscapes of contemporary digital production.

Approaching the history of Japanese electronic music in such a multidimensional manner fosters a closer relationship with individual electroacoustic works often regarded as isolated. It also broadens the scope of Japanese electronic music studies by situating them within wider media-historical frameworks. The case studies discussed in this paper, when interpreted through the genealogies of their respective media, serve as a foundation for a stereoscopic understanding of Japanese electronic music history. Furthermore, this perspective holds the potential to extend toward future research on electroacoustic works for other instruments. By illuminating forgotten media of the past to re-examine contemporary history, this study bears significant value within the framework of media archaeology.

Organizing and categorizing historical Japanese works for piano and electronics also elucidates the inherent vulnerability of past electroacoustic compositions and their respective media. The current state of these historical works is precarious; numerous compositions are difficult to reperform because their original media—such as Multified Piano or Hyper N-5—have been lost, damaged, or rendered obsolete. This situation underscores the ontological dependence of electroacoustic music on specific media, a stark contrast to traditional classical compositions written for acoustic instruments. Adopting a multidimensional view and re-examining the history of Japanese electronic music provides vital clues for resolving this dilemma—as demonstrated by Isobe's successful reconstruction of Mayuzumi's lost composition from a perspective of modern technology. In conclusion, this multidimensional view provides a foundation for future research from alternative perspectives.

4. CONCLUSIONS

In the contemporary music scene, many new works vanish after their premieres, quickly replaced by the next wave of compositions. In an era saturated with cutting-edge technology, re-examining past electronic music is essential; much like classical music from the past prompts us to rethink the present, historical electronic works allow us to scrutinize our current era through a historical lens. Ultimately, illuminating the past serves to re-evaluate the artistic and social value of these works by redefining the dynamics between compositions, performers, and audiences today.

The framework proposed in this paper—organizing Japan's dispersed electronic music history across multidimensional temporal layers—is an integral part of this verification process. Interpreting past technical constraints from a modern perspective is not merely an act of nostalgia; it is an active "music as dialogue" that transmits the inherent structures and functions of media to the next generation.

As no creation exists in a vacuum, every contemporary work references the past, whether through personal memory or social convention. Engaging with historical works provides more than just performance material; it offers potential references for new techniques and societal relationships. For instance, Kotoka Suzuki's "Blue, Rising", which utilizes a piano with vibrating speakers, serves as a "symbolic reference to the past," [23] illustrating how history is integrated into modern practice.

Similarly, unraveling neglected Japanese historical works and their media is not a mere exhibition of legacy. It is a re-examination of the relationship between works and the sociocultural context of their creation, fostering new connections with our own era. By rethinking past "failures" or trial-and-error processes, we can reconstruct modes of engagement with art and technology—ones unattainable through modern media alone.

In the same way that the live electronics in Toshiro Mayuzumi's work anticipated the development of later computer music, and Shigenobu Nakamura's explorations anticipated future MIDI practices, reconstructing past technical inquiries provides multifaceted clues for the evolution of media. By re-evaluating these historical trial-and-error processes as meaningful creative assets, this research serves as a vital "starting point" that evokes new imagination for future musical creation.

Acknowledgments

The author expresses sincere gratitude to all the composers who generously shared their scores and relevant materials, and for providing invaluable advice throughout this research. Special thanks are extended to Professor Suguru Goto and Mr. Koji Kawasaki for their dedicated supervision and guidance. This research was also supported by a grant from the Horita Educational Foundation.

5. REFERENCES

[1] M. Moroi, “The World of Electronic Music [電子音楽の世界],” *The Art of Music* [音樂藝術], vol. 12, no. 6, pp. 40–45, 1954.

[2] K. Kawasaki, *Japanese Electronic Music Second Edition*. Aiikusha Publishing, 2009.

[3] K. Miyashita, *NHK Electronic Music Studio: The role of audio engineers*. Tokyo University of the Arts, 2018.

[4] S. Sato, *The Beginnings of Japanese Electroacoustic Vol. 4: SHIGERU SATO works ＜Diversification of production and computer music＞* (Liner Notes), Sound 3, 2004.

[5] Y. Matsudaïra, “Graphic Notation and Performance [図形楽譜と演奏],” *The Art of Music* [音樂藝術], vol. 28, no. 6, pp. 26–31, 1970.

[6] M. Shiomi, *Cross Talk / Intermedia (Program Notes)*, *Million Concert*, 1969.

[7] M. Kimizuka, “ Historical Development of Magnetic Recording and Tape Recorder,” *Report on Systematization of Technology* [技術の系統化調査報告], vol. 17, pp. 183-273, 2012. https://sts.kahaku.go.jp/albums/abm.php?d=3254&f=abm00010779.pdf&n=073.pdf.

[8] T. Morimoto, S. Keishu, and A. Fujikawa, “Technics Cassette Deck RS-279U,” *Radio Science* [電波科学], vol. 491, 1973.

[9] S. Satoh, “Litania,” *Live Electronic Music in Japan* [日本のライブ・エレクトロニクス音楽], enginebooks - difference, 2021.

[10] B. Okazaki, “Japan's First Electronic Computer FUJIC [わが国初めての電子計算機 FUJIC],” *Journal of Information Processing*, vol. 15, no. 8, pp. 624-632, 1974. http://museum.ipsj.or.jp/guide/pdf/magazine/IPSJ-MGN150808.pdf.

[11] T. Shimazu, “The History of Electronic and Computer Music in Japan: Significant Composers and Their Works,” *Leonardo Music Journal*, vol. 4, pp. 102-106, 1994.

[12] J. Kitaguchi, “Systematized Survey of the Technical Development of Electronic Musical Instruments,” *Report on Systematization of Technology* [技術の系統化調査報告], vol. 26, pp. 1-90, 2019. https://sts.kahaku.go.jp/albums/abm.php?d=3254&f=abm00010829.pdf&n=110.pdf.

[13] T. Akaishi, “The World of Digital Piano Performance: Exploring New Possibilities of Expression with Electronic Piano – In the Case of Audio-Visual Works”, *JSCM Exhibition of Contemporary Music 2003*, pp. 35–45, 2003.

[14] Y. Hase Kojima, “Masahiro Miwa’s “Gesänge des Ostens”: Intersection of Technology and Tradition,” *Proceedings of the Electroacoustic Music Studies Network Conference*, Florence, 2018. http://www.ems-network.org/spip.php?article463.

[15] H. Lim, “Historical Performance Practice in Electroacoustic Music through Restoring Lost Media - “Interface Concerto” by Shigenobu Nakamura as a Case Study,” *Proceedings of the Electroacoustic Music Studies Network Conference*, Paris, 2025. https://zenodo.org/records/19145375.

[16] N. Sakonda, ““Brother’s realtime vocal synthesis performing system”: the overview and background,” *Nagoya University of Arts and Sciences, School of Media and Design / Research Bulletin 2013*, vol. 6, pp. 21-34, 2013. https://nufs-nuas.repo.nii.ac.jp/record/660/files/vol.06_Research_03.pdf.

[17] G. Litterst, *The History of Disklavier, Part 1*. Yamaha, 2017. https://hub.yamaha.com/pianos/p-history/celebrating-30-years-of-disklavier-the-history-of-disklavier-part-i/.

[18] M. Miwa, *Gesänge des Ostens (Speaking Booklet)*, Fontec, 1998.

[19] G. Assayag, L. Bonnasse-Gahot, and J. Borg, “Cocreative Interaction: Somax2 and the REACH Project,” *Computer Music Journal*, vol. 46, no. 4, 2022.

[20] M. Fujieda, “Piano Cocoon,” *Ikebukuro Cyber Café*, pp. 12-14, 1991.

[21] M. Miwa, *18 sai-no-shi*, The Nippon Foundation Library, 2000. https://nippon.zaidan.info/seikabutsu/2000/00727/contents/026.htm.

[22] K. Matsumiya, “Research on hybrid instruments through "Shitatari" for piano and electronics (2019),” *Journal of Japanese Society for Sonic Arts*, vol. 11, no.3, pp. 34-38, 2019. https://data.jssa.info/paper/2019v11n03/8.Matsumiya.pdf.

[23] K. Suzuki, *Blue, Rising – Piano + Transducers*, Kotoka Suzuki, 2019. https://www.kotokasuzuki.com/order/i9q6hnleub4chf6nj0ezi7l2qm0jvt.

[24] Y. Morita, *about ReincarnatiOn Ring VI*, Private Edition, 2023.

[25] M. Yamauchi, “A Ring of Resonance Woven from Gagaku Instruments: Discussing ReincarnatiOn Ring II — An Interview with Member Yasunoshin Morita”, *New Composer,* Vol. 14, pp. 20-23, 2019.

Beyond Musique Concrète: Perceptual Morphing via Audio Latent Embeddings Manipulation

Xiangbin Lin *
Central Conservatory of Music
xiangbinlin@mail.ccom.edu.cn

Du Huang *
Central Conservatory of Music
duhuang@mail.ccom.edu.cn

Qi Qian †
Central Conservatory of Music
qqi@ccom.edu.cn

Maosong Sun
Tsinghua University
sms@tsinghua.edu.cn

ABSTRACT

This paper proposes "Neural Musique Concrète", a compositional paradigm that reinterprets the Quantized Audio Latent Embedding produced by Neural Audio Codecs (NACs) as a malleable digital "Sound Object" (L'Objet Sonore) amenable to direct artistic intervention. While end-to-end generative AI has dramatically accelerated music production, it confines creators to prompt-level interaction, effectively reversing the long-standing trend toward increasingly fine-grained control over acoustic micro-structures. To restore this creative agency, we introduce Latent Manipulation Functions (LMFs)—weighted summation with independent time-varying coefficients—that operate directly on the continuous latent space, enabling "Perceptual Morphing": the deep semantic and acoustic fusion of heterogeneous sound materials beyond waveform-domain superposition. The framework is validated through the electronic composition "He" (Harmony), whose three compositional phases (spectral fusion, stochastic granular scattering, and order-chaos coupling) demonstrate that tensor-based latent editing supports structurally complex musical forms; a complementary survey of five NACs further establishes the requirements for codec applicability. Our results indicate that direct manipulation of NAC latent embeddings effectively bridges the high fidelity of modern AI systems with the fine-grained compositional control central to avant-garde electronic music.

1. INTRODUCTION

In recent years, driven by advances in deep learning, the application of Generative Artificial Intelligence in music has expanded rapidly. The new AI music generation systems have demonstrated reliable "end-to-end" capabilities, often requiring only textual prompts or brief audio conditioning to automatically generate structurally complete and realistically musical audio.

This trend toward high automation has prompted critical reflection on subjectivity and control in musical creation. In the evolutionary history of music technology, the creative paradigm has undergone a leap from the macroscopic to the microscopic: evolving from the era of **Symbolic Representation**, where composers were constrained by musical scores and could only define pitch, duration, and dynamics; to the era of **Signal Representation**—represented by *Musique Concrète* and electronic music—where creators could work directly at the waveform level to sculpt timbre and acoustic micro-structures. However, while the current "Prompt-to-Music" model significantly enhances production efficiency, it to some extent deprives creators of the ability to intervene in sonic details, posing a challenge that threatens to reverse the long-standing evolutionary trend toward increasingly fine-grained creative control.

To better understand this issue, we provide an in-depth deconstruction of the underlying architectures of current mainstream AI music generation models (such as MusicLM [1], MusicGen [2]). Analysis indicates that despite model variations, their core workflows adhere to an "Encode-Predict-Decode" paradigm: first, employing Neural Audio Codecs (NACs) for Audio Tokenization to map high-dimensional continuous waveform signals into low-dimensional discrete tokens or continuous **Latent Representations**; subsequently, utilizing Transformers or Diffusion Models to learn the temporal distribution of these representations and generate new sequences.

This paper posits that the **Audio Latent Embedding** within this architecture is not merely intermediate data but a highly plastic creative medium. By designing specific **Latent Manipulation Functions**—such as weighted summation with time-varying interpolation coefficients—creators can bypass the "black-box" limitations of generative models and directly edit the acoustic properties and semantics of the sound. From the theoretical perspective of Pierre Schaeffer, the Audio Latent Embedding constitutes the digital **"Sound Object" (*L'Objet Sonore*)** of the AI era. Based on this premise, this study aims to explore direct editing methods based on the NAC latent space, proposing a compositional paradigm we term **"Neural Musique Concrète"**—a continuation and innovation of the aesthetics of *Musique Concrète* within the context of artificial intelligence. Figure 1 illustrates the main operational flow of the proposed framework.

*Equal contribution.
†Corresponding author.

Copyright: ©2026 Xiangbin Lin et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License 3.0 Unported, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

2. RELATED WORK

Before the dominance of autoregressive transformers, Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs) established the foundational paradigms for latent space manipulation. Yee-King [3] provided a broad overview of how latent spaces can be navigated for musical purposes. Seminal works such as GANSynth [4] demonstrated that audio could be synthesized by sampling from a high-dimensional continuous latent space (z). Similarly, the Latent Timbre System [5] utilized VAEs to learn and traverse timbral frames of audio.

A key contribution of these generative models was the demonstration of *latent interpolation*, where traversing the path between two latent vectors results in a smooth, nonlinear morphing of timbre and pitch. This paved the way for treating the latent space as a continuous manifold for sonic exploration. Further advancements, such as RAVE (Real-time Audio Variational autoEncoder) [6], successfully adapted these concepts for real-time interaction, allowing artists to navigate the latent space during live performance.

Parallel to latent-space methods, *corpus-based concatenative synthesis* approaches such as CataRT [7] and the Fluid Corpus Manipulation toolkit (FluCoMa) [8] enable composers to navigate and recombine large sound collections through acoustic descriptor spaces. These systems select and juxtapose audio grains based on spectral, temporal, or perceptual features, offering fine-grained signal-level control. However, because they operate on pre-segmented audio units rather than learned representations, the resulting transformations are fundamentally recombinative—producing mosaics of existing material rather than perceptually novel hybrid timbres.

More recently, diffusion-based models have been applied to audio editing. Systems such as AudioLDM [9] and AUDIT [10] employ text-conditioned latent diffusion to perform style transfer and content modification. While powerful, these approaches mediate creative intent through natural language prompts, inheriting the same "black-box" limitation of prompt-to-audio generation: the composer specifies *what* should change but cannot directly control *how* the latent representation is transformed.

Despite these diverse approaches, the GAN/VAE-based latent space methods discussed above often suffer from limitations relevant to our goals: 1) **Global vs. Local Control:** In global latent models (e.g., GANSynth), the latent vector encodes attributes of the entire audio clip. Manipulating the vector affects the segment holistically, making it difficult to perform precise, time-localized editing (e.g., changing the timbre at $t = 2s$ without affecting the rest). 2) **Entanglement:** While techniques exist to disentangle pitch and timbre, unstructured latent spaces often exhibit complex correlations where modifying one attribute inadvertently alters others.

To address fidelity and structural issues, the field has increasingly adopted Neural Audio Codecs (NACs). Unlike the global latent vectors in early GANs, the Quantized Latent Embedding (z_q) in NACs is frame-aligned, providing high temporal resolution for local editing. Furthermore, since z_q exists in a continuous vector space (post-quantization), it inherently supports arithmetic latent manipulation functions.

While recent studies [11] emphasize the novelty of exploring generative latent spaces, they primarily focus on "generative" capabilities (i.e., sampling new sounds from a distribution). In contrast, our work focuses on "transformative" capabilities, treating the Quantized Latent Embedding as a concrete digital sound object. We propose that by applying explicit manipulation functions (e.g., interpolation, vector arithmetic) to z_q, we can achieve precise control over acoustic and semantic features—such as timbre morphing and texture densification—while maintaining the structural integrity provided by the residual vector quantization architecture.

3. AUDIO TOKENIZATION

In modern generative audio systems, audio tokenization converts continuous, high-dimensional waveform signals into compact, discrete representations amenable to autoregressive modeling. This process is fundamentally distinct from traditional symbolic representations (such as MIDI) as it encodes acoustic characteristics directly from the raw signal.

3.1 Neural Audio Codecs

The core technology enabling this transformation is the Neural Audio Codec (NAC). Architectures such as EnCodec [12] and SoundStream [13] adopt a convolutional autoencoder framework coupled with a Residual Vector Quantizer (RVQ). Formally, let $x \in \mathbb{R}^T$ denote the input raw audio waveform, where T is the total number of samples. The process consists of three primary stages: encoding, quantization, and decoding.

The **Encoder** network $\mathcal{E}$ maps the input waveform x to a latent representation z in a lower-dimensional space:

$$z = \mathcal{E}(x), \quad z \in \mathbb{R}^{D \times N} \tag{1}$$

where D represents the channel dimension of the latent space, and N represents the number of time frames, with $N \ll T$. This downsampling significantly reduces the temporal resolution while preserving the semantic and acoustic content of the signal.

The **Quantizer** $\mathcal{Q}$ then discretizes this continuous latent representation z. Unlike standard scalar quantization, NACs utilize *Residual Vector Quantization* (RVQ) to minimize the reconstruction error. RVQ employs a hierarchy of M vector codebooks, denoted as $\mathcal{C} = \{\mathcal{C}_1, \ldots, \mathcal{C}_M\}$. The quantization is performed iteratively: the first quantizer approximates the latent vector, and each subsequent quantizer approximates the residual error of the previous step. Let $r_0 = z$. For each quantization stage $j \in \{1, \ldots, M\}$, the residual r_{j-1} is mapped to the nearest codebook vector $\mathbf{e}_k^{(j)} \in \mathcal{C}_j$:

$$\hat{z}_j = \mathbf{e}_k^{(j)} \quad \text{s.t.} \quad k = \underset{i}{\operatorname{argmin}} \|r_{j-1} - \mathbf{e}_i^{(j)}\|_2 \tag{2}$$

The residual is then updated as $r_j = r_{j-1} - \hat{z}_j$. This hierarchical structure allows the system to capture information in a coarse-to-fine manner, where early layers capture fundamental structures (e.g., pitch, rhythm) and later layers capture fine acoustic details (e.g., timbre, noise).

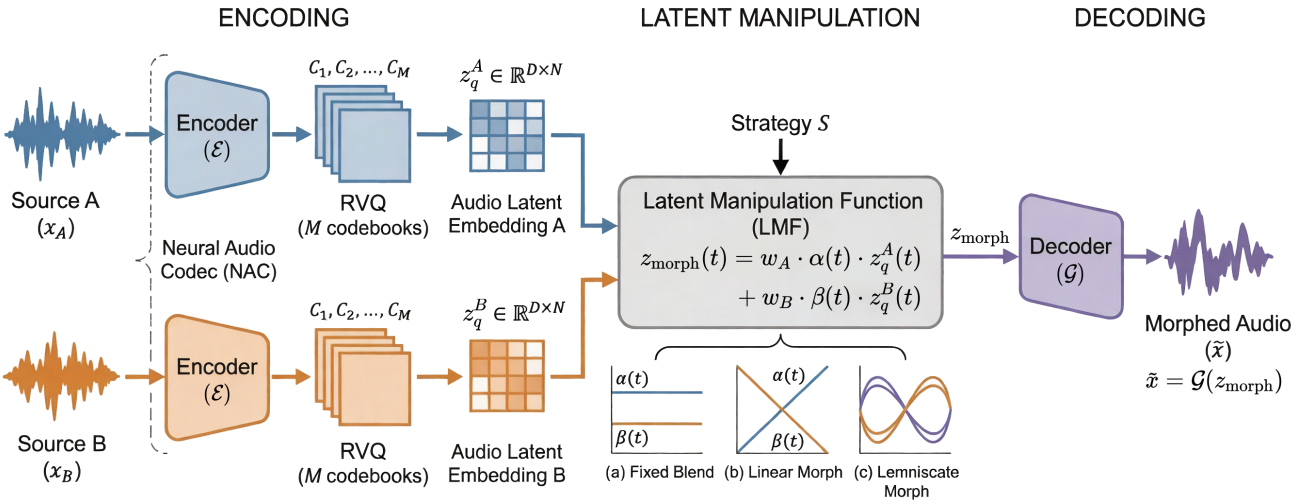


Figure 1. **Perceptual morphing framework overview.** Sources A and B are encoded by a Neural Audio Codec (NAC) into Audio Latent Embeddings z_q^A and z_q^B. The Latent Manipulation Function (LMF) computes $z_{\text{morph}}(t) = w_A \cdot \alpha(t) \cdot z_q^A(t) + w_B \cdot \beta(t) \cdot z_q^B(t)$ under one of three interpolation strategies (Fixed Blend, Linear Morph, or Lemniscate Morph). The decoder $\mathcal{G}$ reconstructs the morphed waveform $\tilde{x}$.

Finally, the **Decoder** $\mathcal{G}$ reconstructs the audio waveform $\hat{x}$ from the quantized representation z_q:

$$\hat{x} = \mathcal{G}(z_q) \approx x \tag{3}$$

The entire network is trained end-to-end using a combination of reconstruction loss and adversarial loss to ensure high perceptual quality.

3.2 Audio Latent Embedding

A critical distinction must be drawn between the symbolic tokens used for sequence modeling and the physical embeddings used for audio reconstruction. While generative models (e.g., MusicGen) operate on the sequence of discrete indices (tokens), our research focuses on the Quantized **Audio Latent Embedding (ALE)**, formally denoted z_q.

The discrete tokens S are merely the indices pointing to entries in the codebooks. For a given time step t, the token set is $S_t = \{s_t^1, s_t^2, \ldots, s_t^M\}$, where s_t^j is the integer index chosen from the j-th codebook. However, these indices are categorical and non-continuous, rendering them unsuitable for direct signal processing operations such as interpolation or mixing.

Conversely, the Quantized Latent Embedding z_q is the vector summation of the selected codebook entries across all RVQ layers. Formally, at time step t, the embedding is defined as:

$$z_q(t) = \sum_{j=1}^{M} \mathbf{e}_{s_t^j}^{(j)} \tag{4}$$

where $\mathbf{e}_{s_t^j}^{(j)}$ is the distinct vector retrieved from codebook $\mathcal{C}_j$ at index s_t^j.

This vector z_q possesses unique properties that distinguish it from both the raw input z and the discrete tokens S:

- **Vector Space Arithmetic:** Unlike categorical tokens, z_q resides in a continuous vector space (albeit a discretized manifold). This allows for arithmetic operations, such as linear interpolation between two distinct sound objects ($z_q^{new} = \alpha z_q^A + (1-\alpha) z_q^B$), which is mathematically valid and acoustically meaningful.
- **Decoupled Representation:** Due to the residual nature of the summation in Equation (4), the embedding z_q is structurally layered. The coarse components contribute to the primary "shape" of the vector, while fine components add detailed nuances.

Therefore, the ALE (z_q) is a concrete, manipulatable digital sound object. It encapsulates the high-level semantic information required by the decoder to reconstruct the audio, while offering a mathematical structure amenable to continuous editing functions.

4. LATENT MANIPULATION FUNCTIONS

In this section, we formalize the methodology of modifying audio content through direct operations on ALEs (z_q). We define these operations as *Latent Manipulation Functions* (LMFs). Unlike symbolic editing which mainly alters music semantic structure, or signal processing which alters waveforms and dramatically changes the resulting acoustic perception, LMFs operate on the continuous latent space of the semantic and acoustic representations, enabling a novel form of "Perceptual Morphing".

4.1 Mathematical Formulation of Latent Manipulation

Let $z_q \in \mathbb{R}^{D\times N}$ denote the quantized latent embedding of an audio segment, where D is the embedding dimension and N is the number of time frames. We define a manipulation function $\mathcal{F}$ parameterized by a control variable λ:

$$\tilde{z}_q = \mathcal{F}(z_q^{(1)}, z_q^{(2)}, \ldots, z_q^{(k)}; \lambda) \tag{5}$$

The modified embedding $\tilde{z}_q$ is then passed to the decoder to synthesize the new audio $\tilde{x} = \mathcal{G}(\tilde{z}_q)$.

4.2 Morphing via Weighted Summation

The most fundamental yet expressive LMF is the weighted summation between two distinct audio sources. Let z_q^A and z_q^B represent the quantized latent embeddings of two acoustically contrasting sources. We define the generalized morphing function as:

$$z_{\text{morph}}(t) = w_A \cdot \alpha(t) \cdot z_q^A(t) + w_B \cdot \beta(t) \cdot z_q^B(t) \quad (6)$$

where $w_A, w_B \in \mathbb{R}^+$ are source weight coefficients controlling the overall intensity of each source, and $\alpha(t), \beta(t) : [0, N] \to [0, 1]$ are time-varying interpolation functions governing the temporal evolution of the mixture.

The functions $\alpha(t)$ and $\beta(t)$ are defined as *independent*—the framework imposes no constraint requiring $\alpha(t) + \beta(t) = 1$. This relaxation of the normalization constraint introduces additional creative freedom: when $w_A \cdot \alpha(t) + w_B \cdot \beta(t) > 1$, the resulting embedding exhibits amplified feature intensity; when the sum falls below unity, features are attenuated. In the special case where $w_A = w_B = 1$ and $\beta(t) = 1 - \alpha(t)$, Equation (6) reduces to standard linear interpolation.

While Equation (6) superficially resembles a cross-fade in signal processing, the perceptual outcome in the latent space is fundamentally different. In the waveform domain, summing two signals ($x_A + x_B$) produces superposition—both sounds are heard simultaneously but remain perceptually distinct. In the latent space, however, the decoder $\mathcal{G}$ is a highly non-linear function: the input z_{morph} often corresponds to a point on the learned manifold that represents neither Source A nor Source B, but a hybrid "sound object" exhibiting characteristics of both—achieving *perceptual fusion* rather than mere superposition.

This fusion capability is structurally supported by the Residual Vector Quantization architecture of z_q. Since $z_q = \sum z_{\text{codebook}}^{(i)}$, the embedding encodes hierarchically layered information. Weighted summation thus simultaneously interpolates coarse structures (pitch, rhythm) and fine details (timbre, noise). The decoder, trained to map latent distributions to perceptually realistic audio, acts as a *harmonizer*, resolving the mathematical discontinuities into a coherent auditory experience.

4.3 Interpolation Strategies

The compositional behavior of the morphing function in Equation (6) is determined by the choice of $\alpha(t)$ and $\beta(t)$. We define three interpolation strategies that span a range of artistic intentions, from static textures to dynamic cyclical transformations.

4.3.1 Fixed Blend

The simplest strategy assigns constant values to both interpolation functions:

$$\alpha(t) = c_1, \quad \beta(t) = c_2, \quad c_1, c_2 \in [0, 1] \quad (7)$$

This produces a time-invariant hybrid texture whose timbral character is determined by the ratio $c_1 : c_2$. For example, non-normalized configurations (e.g., $c_1 = 0.75$, $c_2 = 0.55$) allow fine-tuning the perceptual balance between the two sources without constraining their contributions to sum to unity.

4.3.2 Linear Morph

A time-varying strategy creates a smooth, unidirectional transition from Source B to Source A:

$$\alpha(t) = \frac{t}{N}, \quad \beta(t) = 1 - \frac{t}{N} \quad (8)$$

where N is the total number of latent frames. This is the only strategy satisfying the normalization constraint $\alpha(t) + \beta(t) = 1$, producing a gradual crossfade in the latent domain. The perceptual result, however, differs qualitatively from a waveform-domain crossfade: as α traverses from 0 to 1, the decoder generates a continuous morphing trajectory through the latent manifold, yielding intermediate timbres absent from either original source—a sonic sensation of "becoming" rather than "switching".

4.3.3 Lemniscate Morph

Inspired by the cyclical interplay of opposing forces, this strategy maps the interpolation coefficients onto a lemniscate (figure-eight) trajectory:

$$\alpha(\theta) = \frac{1 + \sin\theta}{2}, \quad \beta(\theta) = \frac{1 + \sin\theta\cos\theta}{2} \quad (9)$$

where $\theta = 2\pi t/N$ sweeps one full period over the audio duration. The resulting trajectory in the (α, β) plane traces a self-intersecting loop within $[0, 1] \times [\frac{1}{4}, \frac{3}{4}]$, producing asymmetric oscillations between the two sources. Unlike the Linear Morph, α and β are not complementary: at certain phases, both coefficients are simultaneously elevated (constructive reinforcement) or simultaneously diminished (mutual attenuation), creating dynamic energy fluctuations that evoke a continuous dialectical exchange between opposing sonic identities. This non-normalized interpolation strategy serves as the primary morphing mechanism in the composition *"He"* discussed in Section 6.

4.4 Implementation

The proposed framework is implemented using the Descript Audio Codec (DAC) [14] as the neural audio codec backbone. We employ the 44.1 kHz, 16 kbps pretrained configuration, which produces quantized latent embeddings with dimension $D = 128$ across $M = 18$ residual vector quantization layers. [1] For a 30-second stereo input at 44.1 kHz, the encoder yields a tensor $z_q \in \mathbb{R}^{C \times D \times N}$ with shape $[2, 128, 2584]$, achieving approximately $512\times$ temporal compression ($44100 \times 30/2584 \approx 512$).

Algorithm 1 details the complete perceptual morphing pipeline. A key design decision is *per-channel independent processing*: for stereo audio, each channel is encoded, manipulated, and decoded separately, preserving the spatial imaging of the original sources while applying the morphing function uniformly across channels.

[1] The pretrained weights used in this work correspond to the 44.1 kHz / 16 kbps checkpoint released with DAC v1.0.0 (`weights_44khz_16kbps.pth`), whose configuration (D=128, M=18) differs from the default architecture described in [14] (D=1024, M=9, ∼8 kbps).

Algorithm 1 Perceptual Morphing via Audio Latent Embedding Manipulation

Require: Audio sources x_A, x_B; weights w_A, w_B; strategy $\mathcal{S}$
Ensure: Morphed waveform $\tilde{x}$

for each channel $c \in \{1, \ldots, C\}$ **do** ▷ Encode
  $z_q^A[c] \leftarrow \mathcal{Q}(\mathcal{E}(\text{preprocess}(x_A[c])))$
  $z_q^B[c] \leftarrow \mathcal{Q}(\mathcal{E}(\text{preprocess}(x_B[c])))$
end for
$N \leftarrow$ number of time frames in z_q^A
if $\mathcal{S} =$ Fixed Blend **then** ▷ Interpolation coefficients
  $\alpha(t) \leftarrow c_1, \quad \beta(t) \leftarrow c_2$
else if $\mathcal{S} =$ Linear Morph **then**
  $\alpha(t) \leftarrow t/N, \quad \beta(t) \leftarrow 1 - t/N$
else if $\mathcal{S} =$ Lemniscate Morph **then**
  $\theta \leftarrow 2\pi t/N$
  $\alpha(\theta) \leftarrow (1 + \sin\theta)/2$
  $\beta(\theta) \leftarrow (1 + \sin\theta\cos\theta)/2$
end if
for each channel $c \in \{1, \ldots, C\}$ **do** ▷ Morph & Decode
  $z_{\text{morph}}[c] \leftarrow w_A \cdot \alpha \cdot z_q^A[c] + w_B \cdot \beta \cdot z_q^B[c]$
  $\tilde{x}[c] \leftarrow \mathcal{G}(z_{\text{morph}}[c])$
end for
return $\tilde{x}$

5. APPLICABILITY OF NEURAL AUDIO CODECS

The latent manipulation framework described in Section 4 operates on the ALE (z_q) produced by a Neural Audio Codec. While the methodology is codec-agnostic in principle, its practical effectiveness depends on several codec-specific properties. This section identifies the requirements that a NAC must satisfy to support latent manipulation, surveys representative codecs against these requirements, and provides guidance for practitioners seeking to apply this framework with alternative codecs.

5.1 Requirements for Latent Manipulation

We define eight requirements, grouped by criticality.

5.1.1 Essential Requirements

These must be satisfied for the framework to function:

- **R1 – Continuous Latent Accessibility.** The codec must expose z_q as a continuous vector, not merely discrete token indices, so that arithmetic operations such as the weighted summation in Equation (6) are mathematically valid.
- **R2 – Encode-Manipulate-Decode Pipeline.** The codec architecture must permit extracting z_q from the encoder, applying external tensor operations, and feeding the modified embedding back into the decoder without retraining or re-quantization.
- **R3 – High-Fidelity Reconstruction.** Music applications demand high perceptual quality, which in practice requires a sampling rate of at least 44.1 kHz. Codecs operating at speech-oriented rates (8–24 kHz) discard spectral content critical to musical timbre.
- **R4 – Deterministic Decoding.** The decoder should produce a deterministic output for a given input embedding. Codecs employing stochastic generative decoders (e.g., diffusion models or flow matching) may compromise the reproducibility of the manipulation outcome, as repeated decoding of the same z_{morph} could yield perceptually different results, undermining the creator's precise control over the sonic output.

5.1.2 Quality Factors

These affect the fidelity and granularity of manipulation:

- **R5 – Decoder Robustness.** Manipulated embeddings (e.g., z_{morph} with $\alpha + \beta \neq 1$) may lie outside the manifold of quantized embeddings encountered during training. The decoder must tolerate such off-manifold inputs without severe artifacts.
- **R6 – Temporal Resolution.** The number of latent frames N per unit time determines the finest granularity of time-localized editing. A higher frame rate (lower compression ratio) enables more precise temporal control.

5.1.3 Extension Factors

These broaden the applicability of the framework:

- **R7 – Real-Time Capability.** If the encoder and decoder can operate in real time, the framework extends to live performance and interactive exploration of the latent space.
- **R8 – RVQ Layer Separability.** The hierarchical structure of RVQ allows selective manipulation of coarse layers (pitch, rhythm) independently from fine layers (timbre nuance), enabling layered compositional strategies.

5.2 Comparative Analysis

Table 1 evaluates five representative codecs that satisfy the essential sample-rate requirement (R3). The selection spans the foundational codec used in this work (DAC [14]), two established architectures (EnCodec [12], VRVQ [15]), and two recent music-oriented designs (DisCodec [16], CoDiCodec [17]).

All five codecs satisfy R1 (continuous latent accessibility). DAC, EnCodec, DisCodec, and VRVQ share a convolutional encoder-decoder backbone with RVQ, providing a straightforward encode-manipulate-decode pipeline (R2). Among these, DisCodec achieves the highest embedding dimension ($D = 1024$), offering a richer representation space, and was specifically trained on music data (MTG-Jamendo). VRVQ extends the DAC framework with an adaptive bitrate mechanism that varies the number of active codebooks per frame, potentially enabling content-aware manipulation depth.

CoDiCodec represents an alternative paradigm: it unifies continuous and discrete representations through Finite Scalar Quantization (FSQ) with a novel dropout mechanism, and employs a consistency-model decoder rather

Codec	SR (kHz)	Bitrate (kbps)	Dim	CB	R1	R2	R4	R7	Domain
DAC [14]	44.1	16	128	18	•	•	•	△	General
EnCodec [12]	48.0	3–24	128	2–16	•	•	•	△	Music
DisCodec [16]	44.1	8	1024	8	•	•	•	△	Music
CoDiCodec [17]	44.1	2.38	512	FSQ*	•	△	×	×	Music
VRVQ [15]	44.1	Var.	1024	Var.†	•	•	•	△	General

Table 1. Comparison of Neural Audio Codecs for latent manipulation suitability. Column abbreviations — SR: sampling rate (kHz); Dim: latent embedding dimension; CB: codebook count or quantization scheme. Requirement columns show R1, R2, R4, R7 only; R3 is reflected in SR, and R5, R6, R8 are discussed in the text. Symbols: • fully met, △ partially met, × not met.

* Finite Scalar Quantization: each latent dimension is independently rounded to a fixed number of discrete levels, replacing explicit vector codebooks with implicit combinatorial codes.

† VRVQ adaptively varies the number of active RVQ codebooks per frame via a learned importance mask; the effective codebook count is content-dependent.

than a deterministic convolutional decoder. While this architecture natively provides continuous embeddings (R1), the encode-manipulate-decode pipeline (R2) requires further validation, as the decoder's behavior on externally manipulated embeddings has not been characterized. Furthermore, the stochastic nature of the consistency-model decoder does not guarantee deterministic output (R4).

Regarding decoder robustness (R5), systematic evaluation across codecs remains an open research question. In our experiments with DAC, non-normalized interpolation under the Lemniscate Morph strategy (Equation 9), where $\alpha + \beta$ ranged from approximately 0.35 to 1.65, produced musically meaningful results without severe artifacts, suggesting a degree of tolerance in the learned decoder.

5.3 Practical Guidance

For practitioners seeking to apply latent manipulation, codecs with a convolutional encoder-decoder backbone and standard RVQ quantization—such as DAC, EnCodec, and DisCodec—currently offer the most direct compatibility with the framework described in this paper. The choice among them involves trade-offs: EnCodec provides the highest sampling rate (48 kHz) and native real-time streaming in its causal configuration; DisCodec offers a larger embedding dimension and music-optimized training; and DAC balances general-domain versatility with well-documented latent access APIs. Emerging architectures such as CoDiCodec point toward future possibilities but require further investigation of decoder robustness under external embedding manipulation.

6. CASE STUDY: ELECTRONIC MUSIC COMPOSITION "HE" (*HARMONY*)

To validate the theoretical framework and technical methodology proposed in the previous sections, we present the electronic music composition *"He"* (《龢》) as a practical case study. This work is an experimental vehicle for exploring the deep fusion mechanisms of heterogeneous sound materials via the direct manipulation of ALEs.

6.1 Conceptual Framework: The Dialectic of "Yue" and "He"

The conceptual foundation of the work is derived from the etymology of the Chinese character "龢" (Harmony). The character is structurally composed of "龠", an ancient wind instrument symbolizing artificially constructed musical order, and "禾", meaning grain, symbolizing nature and raw vitality.

In the context of our technical framework, we establish a semiotic mapping between these philosophical concepts and our computational tools:

- **The "He" (禾, Nature):** Corresponds to the **ALEs**. Generated by the DAC encoder, these embeddings represent the raw, unstructured potential of sound materials in the high-dimensional latent space.
- **The "Yue" (龠, Artificial Structure):** Corresponds to the **LMFs**. These functions act as the logic of artificial intervention, reshaping the topological relationships of the tensors within the latent space.

The composition process is thus defined as the structured application of LMFs upon ALEs, seeking a synthesis between algorithmic logic and sonic nature.

6.2 Phase I: Spectral Fusion via Continuous Interpolation

The opening section initiates the interaction between two contrasting long-form sound sources, metaphorically representing the cosmological dimensions of "Earth" and "Sky".

- **Source A (Earth):** Characterized by dominant low-frequency components, exhibiting a heavy, stable acoustic footprint.
- **Source B (Sky):** Characterized by high-frequency brightness, exhibiting a light, spectral morphology.

Technically, the system applies the **Lemniscate Morph** strategy (Equation 9) as the primary interpolation mechanism. The time-varying coefficients $\alpha(t)$ and $\beta(t)$ trace a self-intersecting loop in the parameter space, producing asymmetric oscillations between the two sources via weighted summation in the latent domain (Equation 6). As $\alpha(t)$ and $\beta(t)$ evolve, the energy envelope of the low-frequency source and the spectral texture of the high-frequency source intertwine. Unlike waveform-domain mixing which merely superimposes amplitudes, the decoder's non-linear mapping transforms this latent-space arithmetic into a composite timbre that possesses the "gravity" of Earth and the "luminosity" of Sky simultaneously,

achieving a deep perceptual fusion where the resulting sound object is distinct from either original source.

6.3 Phase II: Granular Scattering via Stochastic Resampling

As the composition progresses, the auditory texture shifts from continuous morphing to discrete fragmentation, reflecting the physical imagery of scattered grains (禾). In this phase, the inputs are transient-rich, fragmented sound materials. The Latent Manipulation Function shifts its operational logic from smooth interpolation to a Stochastic Discrete Resampling Mechanism.

The algorithm performs micro-slicing on the ALEs along the temporal dimension (N). By randomly reordering these latent slices, the inherent linear coherence of the source audio is deconstructed. This manipulation generates a sound field characterized by high density, non-continuity, and instability. Acoustically, it simulates the chaotic yet detailed behavior of particles in physical space, establishing a structural counterpoint to the stable fusion observed in Phase I.

6.4 Phase III: The Coupling of Order and Chaos

The final stage aims for a macro-structural integration, seeking a dynamic balance between the opposing forces of "High vs. Low" and "Order vs. Chaos". The system accepts parallel inputs from the previously generated continuous fusion (Phase I) and the fragmented granular texture (Phase II). The Latent Manipulation Function employed here acts as a Feature Embedding/Mapping operator.

Specifically, the operation preserves the spectral skeleton and temporal continuity of the Phase I material while embedding the transient features of the Phase II material into its internal texture. This creates a juxtaposition of "elongation" and "fracture" within the latent representations. The resulting auditory form achieves a coupling of continuity and discreteness, manifesting a state of dynamic equilibrium.

Through these three evolutionary stages, "*He*" (《穌》) demonstrates that the editing of ALEs offers a novel compositional pathway. While Phase I directly instantiates the weighted summation framework formalized in Section 4, Phases II and III illustrate broader latent manipulation strategies—stochastic resampling and feature embedding—that extend beyond the specific mathematical formulation presented in this paper, pointing toward a richer taxonomy of LMFs to be developed in future work. In particular, the perceptual outcomes of latent morphing differ qualitatively from waveform-domain mixing: listeners perceive neither Source A nor Source B, but genuinely novel hybrid timbres that could not be obtained through conventional signal processing. This confirms that the decoder's non-linear mapping transforms arithmetic operations in the latent space into semantically meaningful acoustic transformations, validating the proposed methods as effective tools for expanding the boundaries of sound fusion in avant-garde electronic music.

7. CONCLUSIONS

This paper addresses the challenge of diminishing creative agency in the era of "end-to-end" AI music generation. By deconstructing the architecture of modern generative models, we identified the **Audio Latent Embedding (ALE)**—derived from Neural Audio Codecs—as a key medium for artistic intervention. Unlike discrete tokens used for sequence modeling, the ALE functions as a manipulatable digital "Sound Object", bridging the gap between symbolic representation and physical acoustics.

We established a technical framework based on **Latent Manipulation Functions (LMFs)**, enabling direct editing of acoustic and semantic features through operations such as vector interpolation and arithmetic. This methodology moves beyond simple signal processing, enabling a form of "Perceptual Morphing" where heterogeneous sound materials undergo deep structural fusion within the latent space.

The practical viability of this approach was validated through the electronic music composition *"He"* (《穌》). The case study demonstrated that tensor-based manipulation effectively supports complex compositional logic: from the continuous spectral fusion of contrasting timbres to the stochastic scattering of granular textures, and finally to the dynamic coupling of order and chaos. These results confirm that operating on the latent space allows for a coherent synthesis of algorithmic structure (龠, "Yue") and raw sonic potential (禾, "He").

In summary, this research proposes a shift from prompt-based generation to direct latent manipulation. By treating the ALE as a plastic creative medium, we provide a pathway that preserves the high fidelity of modern AI systems while restoring the fine-grained control central to the aesthetics of Musique Concrète.

We acknowledge that this restoration of agency is partial: the topology and expressiveness of the latent space are ultimately shaped by the codec's architecture, training data, and optimization objectives—factors that remain outside the composer's direct control. Nevertheless, compared to prompt-based generation where creative intent is mediated entirely through natural language, latent manipulation shifts the mode of artistic intervention from *describing* desired outcomes to *sculpting* the internal representations of sound, representing a significant recovery of the micro-structural agency that defines the tradition of electroacoustic composition.

8. FUTURE WORK

Future work will explore selective manipulation of individual RVQ layers to enable hierarchical editing of coarse structures independently from fine timbral details, and systematic evaluation of decoder robustness under off-manifold embeddings across different codecs. We also plan to investigate real-time implementations for live performance contexts and to conduct formal perceptual listening tests that quantitatively validate the morphing quality of the proposed framework.

9. REFERENCES

[1] A. Agostinelli, T. I. Denk, Z. Borsos, J. Engel, M. Verzetti, A. Caillon, Q. Huang, A. Jansen,

A. Roberts, M. Tagliasacchi, M. Sharifi, N. Zeghidour, and C. Frank, "MusicLM: Generating Music From Text," arXiv preprint arXiv:2301.11325, 2023.

[2] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Défossez, "Simple and Controllable Music Generation," in *Advances in Neural Information Processing Systems*, vol. 36, 2023.

[3] M. Yee-King, "Latent Spaces: A Creative Approach," in *The Language of Creative AI*, C. Vear and F. Poltronieri, Eds. Cham: Springer International Publishing, 2022, pp. 137–154.

[4] J. Engel, K. K. Agrawal, S. Chen, I. Gulrajani, C. Donahue, and A. Roberts, "GANSynth: Adversarial Neural Audio Synthesis," in *International Conference on Learning Representations*, 2019.

[5] K. Tatar, D. Bisig, and P. Pasquier, "Latent Timbre Synthesis: Audio-based Variational Auto-Encoders for Music Composition and Sound Design Applications," *Neural Comput. Appl.*, vol. 33, no. 1, pp. 67–84, Jan. 2021.

[6] A. Caillon and P. Esling, "RAVE: A Variational Autoencoder for Fast and High-Quality Neural Audio Synthesis," arXiv preprint arXiv:2111.05011, 2021.

[7] D. Schwarz, G. Beller, B. Verbrugghe, and S. Britton, "Real-Time Corpus-Based Concatenative Synthesis with CataRT," in *Proceedings of the 9th International Conference on Digital Audio Effects (DAFx-06)*, Montreal, Canada, 2006.

[8] P. A. Tremblay, G. Roma, and O. Green, "Enabling Programmatic Data Mining as Musicking: The Fluid Corpus Manipulation Toolkit," *Computer Music Journal*, vol. 45, no. 2, pp. 9–23, Jun. 2021.

[9] H. Liu, Z. Chen, Y. Yuan, X. Mei, X. Liu, D. Mandic, W. Wang, and M. D. Plumbley, "AudioLDM: Text-to-Audio Generation with Latent Diffusion Models," in *Proceedings of the 40th International Conference on Machine Learning*, 2023.

[10] Y. Wang, Z. Ju, X. Tan, L. He, Z. Wu, J. Bian, and S. Zhao, "AUDIT: Audio Editing by Following Instructions with Latent Diffusion Models," *Advances in Neural Information Processing Systems*, vol. 36, pp. 71 340–71 357, Dec. 2023.

[11] K. Tahiroğlu and L. Wyse, "Latent Spaces as Platforms for Sonic Creativity," in *Proceedings of the 15th International Conference on Computational Creativity (ICCC)*, 2024.

[12] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, "High Fidelity Neural Audio Compression," *Transactions on Machine Learning Research*, 2023.

[13] N. Zeghidour, A. Luebs, A. Omran, J. Skoglund, and M. Tagliasacchi, "SoundStream: An End-to-End Neural Audio Codec," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 495–507, 2022.

[14] R. Kumar, P. Seetharaman, A. Luebs, I. Kumar, and K. Kumar, "High-Fidelity Audio Compression with Improved RVQGAN," in *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds. arXiv, 2023.

[15] Y. Chae, W. Choi, Y. Takida, J. Koo, Y. Ikemiya, Z. Zhong, K. W. Cheuk, M. A. Martínez-Ramírez, K. Lee, W.-H. Liao, and Y. Mitsufuji, "Variable Bitrate Residual Vector Quantization for Audio Coding," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2025.

[16] L. A. Lanzendörfer, F. Grötschla, A. Dellali, and R. Wattenhofer, "Neural Audio Codec for Latent Music Representations," in *Audio Imagination: NeurIPS 2024 Workshop AI-Driven Speech, Music, and Sound Generation*, Dec. 2024.

[17] M. Pasini, S. Lattner, and G. Fazekas, "CoDiCodec: Unifying Continuous and Discrete Compressed Representations of Audio," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2025.

Designing responsive rhythms utilizing the Impulse Pattern Formulation (IPF)

Simon Linke
ligeti center
University of Applied Sciences
22081 Hamburg, Germany
simon.linke@haw-hamburg.de

Rolf Bader
Institute of Systematic Musicology
University of Hamburg
20354 Hamburg, Germany

Robert Mores
ligeti center
University of Applied Sciences
22081 Hamburg, Germany

ABSTRACT

Impulse Pattern Formulation (IPF) is an analytical modeling approach for synergetic systems motivated by research on musical instruments. It describes the nonlinear coupling of system components as the interaction between individually propagating, exponentially damped impulse trains. Due to this general approach, the IPF has been successfully applied to topics other than musical instruments and is hypothesized to be capable of modeling the entire process of musical perception and performance in the future. This work investigates how the IPF can be applied as a compositional tool that reproduces fundamental musical behavior by modelling the synchronization of musicians to an external rhythm. The derived model is systematically examined by analyzing its behavior when coupled to numerically designed and carefully controlled rhythmical beat sequences. Thus, in the future, the IPF model can be applied, e.g., to replace drum machines and click tracks with more musical and creative solutions.

1. INTRODUCTION

The Impulse Pattern Formulation (IPF), described in Section 2 is a nonlinear recursive equation derived by Bader [1] initially to model the dynamical behavior of musical instruments and has been successfully applied to do so (e.g., [2, 3, 4]). Like many other nonlinear recursive equations (e.g., the logistic map or the sine circle map), the IPF is a relatively simple equation capable of mimicking nonlinear, chaotic behavior. It typically requires minimal computational resources, enabling its application to intricate dynamic systems with many interconnected components. Thus, the IPF was further transferred to systems other than musical instruments. Bader [5] derived an IPF model of the human brain and thus was able to model the perception of musical large-scale form [6]. Further, it was shown that the IPF can be used as a modeling approach for human interaction and sociopolitical questions, e.g., city planning [7].

However, even though the IPF is capable of describing different kinds of self-organized processes, it is not proven that the entire process of musical perception and production can be modeled with the same equation. Yet, the acoustics of musical instruments and human perception have already been modeled. Thus, this paper should provide a first overview of how the IPF can be applied in music production, not only as a synthesizer but also as a compositional tool that reproduces fundamental musical behavior by focusing on human musical rhythm synchronisation.

In comparison to other models (e.g. [8, 9]), the IPF is applied in the time domain and thus has proven to carefully recreate transient behavior (see, e.g., [3, 4]), which is crucial when modeling rhythm synchronization. Nevertheless, the IPF was never applied to the interaction of a musician with others. Thus, this work investigates how the IPF can model the synchronization of a musician to an external rhythm.

A common approach to add a human feel to electrically or digitally generated rhythms is to add slight random deviations to every beat usually using pink or mor general 1/f-noise (see, e.g., [10, 11, 12, 13]). Still, the results are not 100 % convincing (see, e.g., [14, 15]). More complex and sometimes even nonlinear models have been proposed in the past (see, e.g., [16, 17, 18, 19]), but they were somewhat too specialized or theoretical for musical applications. Thus, this paper proposes a more straightforward solution.

Based on the IPF, a metronome-like algorithm is developed. It reacts to changes in the tempo of musicians, similar to what a real musician would do. It can reproduce the tempo of a musician but also show minor imperfections, which are responsible for a more human-like feel. This study compares the IPF's results when reacting to an artificially generated tempo track to the behavior of humans tapping along with a metronome as investigated in the literature (e.g., [20, 21, 17, 22]). However, one can imagine using the IPF model as a humanized drum machine. By varying a limited set of system parameters, it can be used to replace a real musician or to act as a creative character on its own.

2. IMPULSE PATTERN FORMULATION

The Impulse Pattern Formulation (IPF) is an abstract top-down modeling approach to describe the transient behavior of arbitrary, coupled systems. It was initially introduced to model any musical instrument with one universal set of mathematical equations to compare different musical instruments, focusing on stable tone production and transient behavior [1] .

A musical instrument is often described as a generator acting on a resonator (e.g., [23]). The IPF describes them more generally as self-organized systems: A musical in-

Copyright: ©2026 Simon Linke et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License 3.0 Unported, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

strument is a system acting upon itself, consisting of mutually coupled subsystems. A single subsystem is excited by certain impulses, which is evident for plucked-string or percussion instruments, but it is also true for sustained instruments like, e.g., violins or brass instruments (see, e.g., [2, 1]). These impulses are transferred to neighboring subsystems, and from there again to neighboring subsystems, and so on. Further, when transferring from one subsystem to another, a fraction of the original impulse gets reflected and finally returns to its point of origin, where it might cause any changes.

To make this a bit more ostensive, we choose the simple example of a reed instrument, e.g., a saxophone. It consists of two subsystems: a reed and a tube. The reed sends out impulses that are answered by the tube. The answer is just a callback of the impulses sent out by the reed[1, p.286]. The same impulse that was sent out just got reflected at the end of the tube. Due to the valve-like behavior of the reed, the returning impulse allows a new impulse to enter the tube. Thus, the change of the system is recursively caused by the system itself. Further, it is assumed that the impulses need a certain amount of time to travel from one subsystem to another, during which they get exponentially damped. Bader [1, pp. 286-291] describes this time-dependent propagation of impulse trains as a recursive equation similar, to other general nonlinear modelling approaches e.g. the logistic map (see, e.g., [24]). Thus, the IPF can be derived as:

$$g_+ = g - ln\left(\frac{1}{\alpha}\left(g - \sum_{k=1}^{n} \beta_k e^{g-g_{k-}}\right)\right). \qquad (1)$$

The overall system state is represented by g and is related to the amplitude and the periodicity of a signal (further explanations are given by [2]) g_+ is the subsequent system state, and g_{k-} are the precedent states. There is no precise time interval between those system states, as it is the time until a new event occurs. Thus, when modeling musical instruments, this is usually the period of the fundamental frequency T_0.

α is the input strength of the system. In the example of the saxophone, this would relate to the playing pressure. As a musician can vary the playing pressure during tone production, α could be time-dependent. Nevertheless, when modeling, for instance, plucked string instruments, α refers to the force used to strike the string, which can not be changed once the string is struck.

As stated above, a musical instrument usually consists of multiple coupled subsystems. For instance, when describing a guitar, these are the strings, bridge, top plate, ribs, neck, and bottom plate. k refers to the number of included subsystems. Their specific influences on the produced sound are determined by β_k. Usually, they are termed reflection points, as they describe the strength of backtracking impulses when reflected at a particular subsystem. However, in reality, their influence not only relies on impedance changes but also on internal damping and phase relations due to different acoustical path lengths. Thus, they indirectly represent a musical instrument's geometry and material and are usually not time-dependent. Nevertheless, on wind instruments, they also represent time-dependent fingerings (see, e.g., [25]).

Choosing an initial value g_0, Equation (1) can be iteratively calculated. Depending on the system parameters α and β_k, the IPF can diverge or converge to a stable limit after several iteration steps. Further, the IPF can show chaotic behavior like bifurcations. The bifurcation scenario can be quite complex. The simplest case ($\beta_k = 0\ \forall k \in \mathbb{N}$) is shown in Fig. 1. Here, 2500 iteration steps were performed, and the last 500 values are plotted in dependency of α.

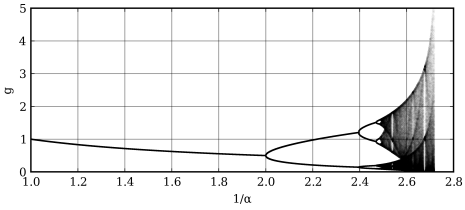


Figure 1. Bifurcation scenario of the IPF with one reflection point α.

The figure becomes more vivid when referring again to the given example of a reed instrument: Low playing pressure, resulting in high values $1/\alpha$, is represented on the right side of the chart. The shown unstable behavior results in noisy sounds. Increasing the playing pressure α results in bifurcations. Here, multiple frequencies can be heard at the same time. Further increase of pressure leads to stable states, resulting in regular periodic motion. The point of this sudden transition is called the first bifurcation point α_c. See [26] for an in-depth mathematical discussion of the boundaries and limits of the IPF.

As explained above, the IPF is not as fixed as traditional generator-resonator models. Any subsystem can be the point of observation. Then α describes the strength of impulses sent out by the observed subsystem while potential β_k describes the impact of other subsystems. Further, the number of reflection points can also vary depending on the research question. While many questions can be answered just by focusing on the most important subsystems, an IPF model can be further detailed by using more reflection points β_k, which implies caring about small details of the instrument geometry or inhomogeneities inside the material. Such scalability is possible within musical instruments but also between entities of instruments.

3. APPLYING THE IPF ON BEAT SYNCHRONISATION

3.1 Deriving an IPF model

One possible solution to model the rhythmic synchronization between two musicians is to emulate their neuronal processes. Thus, the IPF brain model [5] would be an appropriate choice. However, it requires control of lots of (exhibitory and inhibitory) reflection points and time-dependent adaptation due to neuroplasticity. Haken [27] states that an immense number of microscopic components (e.g., muscles or neurons) usually relate to macroscopic behavior when explaining human movement. Even though this relation is often not understood, synergetic behavior can be modeled with a minimal set of order parameters. As in this study, the overall behavior of the musician should

be modeled rather than synchronization processes in the brain, and as introduced in Sec. 2, the IPF and the number of β_k are scaleable depending on the research question. Thus, unnecessary complexity can be avoided. Therefore, we gain a model that can be applied straightforwardly and provides an intuitive understanding of the underlying mechanics.

Every musician is represented by a single subsystem. A particular musician must be chosen as the point of observation, while the others are represented by additional β_k. In Sec. 2 g was described to represent the periodicity of a musical instrument, and the period length of the fundamental frequency determines the time between two iteration steps of the IPF. The same is true here, even though it is the much lower frequency of a constant musical meter.

When modeling a single musician, represented by a single subsystem, the most simple form of the IPF can be used:

$$g_+ = g - ln\left(\frac{g}{\alpha}\right), \tag{2}$$

While the temporal duration g yields from the musicians tempo, α yields from the target tempo as determined intrinsically by the musician or by an external input,such as a metronome, drum computer, or another musician who is unable or unwilling to react to tempo changes. Equation (2) converges against $g = \alpha$, when α is chosen in range sufficient for stable behavior [26]. Thus, the tempo of the IPF converges against the external tempo. In this manner, this approach can be used to model the synchronization of a musician to an external rhythm similar to the tapping experiments presented in the literature (see, e.g., [3, 4]).

The IPF is an iterative equation, and according to Section 1, the time between two iteration steps is not fixed. When modeling musical instruments, it refers to the fundamental frequency.

As described above, the time between two iteration steps of the IPF refers to the period between two beats. However, if these are quarter notes, eighth notes, or something different is not precisely defined and depends on the application. It might be influenced by technical restrictions (e.g., the grid of a hardware sequencer) but also by aesthetic conventions of certain musical genres. As this time defines how fast the IPF is updated and, thus, how fast it converges to an external tempo, it is a crucial parameter. In this study, we relate the value to eighth notes, which, usually medium fast, has proven to be a good compromise. Every time eighth notes are played, a new value g is calculated whereby the new g determines the length of the eighth note and thus the period until the IPF is calculated again.

To use Equation (2), the parameter α has to be calculated based on the external input. Thus, every eighth note, the time between two beats/iterations of the IPF (determined by g) is compared with the time between two succeeding beats of the external input $T_{M,i}$ and one iteration of the IPF is performed. The IPF relies on eighth notes, but it is unclear which note value relates to $T_{M,i}$, the error ΔT between these tempi can be calculated using the modulo operation:

$$\Delta T \equiv T_{M,i} \pmod{g} \tag{3}$$

Thus, α is the IPF target tempo, which equals the external tempo or its integral multiple. It is determined by the system state g corrected by the calculated error:

$$\alpha = g + \Delta T \tag{4}$$

A further restriction has to be made, as when the external input plays dotted notes or triplets, the IPF should synchronize to the fundamental meter. A straightforward solution to this problem is to ignore ΔT, which are longer than a thirty-second note. This approach is also reasonable from a physical point of view, as, e.g., coupled oscillators do only synchronize if their eigenfrequencies are sufficiently similar (see, e.g., [28]). Another alternative approach to this problem is discussed in Section 3.3.

Finally, the parameters g and α must be scaled according to the stable region of the IPF described in detail by [26]. Fast tempos relate to small values of α resp. g. Then, a maximum tempo relates to the first bifurcation point α_c of the IPF. The IPF is not capable of reproducing faster tempos as it becomes chaotic and finally diverges. This maximum tempo is not restricted by the motoric and technical skills of a musician. It must be produced by the IPF only. A coupled musician can play half-time, quarter-time, or even lower due to the above-mentioned modulo operator. Thus, acoustical thoughts can motivate the maximum tempo: Frequencies above 20 Hz are perceived as fused sounds, while lower frequencies are perceived as single beats. Thus, playing with 1200 bpm (beats per minute) would result in a low, buzzing sound instead of a distinguishable rhythm. Consequently, playing a quarter speed of 300 bpm would result in a buzzing sound when playing sixteenth notes, which seems to be the maximum tempo that can be applied in a meaningful musical manner. If τ is a tempo in bpm based on quarter notes,

$$\alpha = \frac{\frac{300\ bpm}{\tau} - 1}{18} + 0.5\,, \tag{5}$$

would relate 300 bpm to α_c. Further, if the tempo is 30 bpm, $\alpha = 1$ resp. $g = 1$, as the same scaling is applied to g and α. Hence, the IPF is restricted to the unit interval for most tempi.

3.2 Investigating the system behavior

In the past, various studies were conducted in which participants tapped along to audible presented rhythmical sequences (e.g.,[20, 22, 10, 21]). Applying similar sequences to the IPF allows a comparison with the literature and, thus, a first comparison between the IPF and human behavior Artificially created impulse trains are used as the input signal for the IPF. These so-called click tracks represent eighth notes played by a musician or a metronome. All tempi in the following sections are given in bpm based on quarter notes.

In most of the following Sections, it is helpful to focus on one tempo to maintain an overview. Therefore, a tempo must be chosen which feels most natural for most people. Several studies have been conducted to find this so-called *spontaneous motor tempo* (e.g., [29, 30]) In the following, 120 bpm is chosen, which matches the 2 Hz found by van Noorden [31], as they have the same magnitude as the values given by most of the other studies.

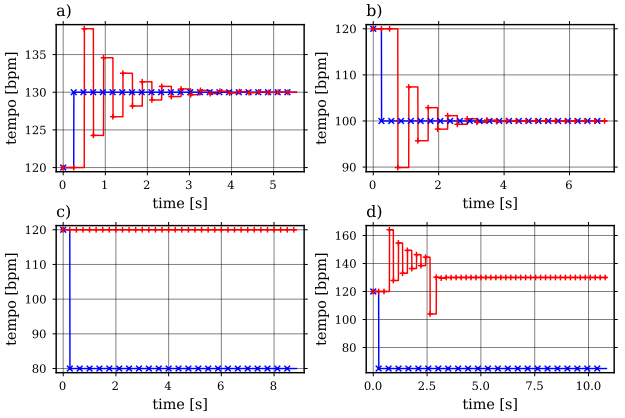


Figure 2. IPF reacts to four different step changes in tempo. Blue lines refer to the tempo of the click track. Red lines correspond to the tempo of the IPF. The single crosses represent single beats. Audio examples of these four cases can be found at [32]

3.2.1 *Discrete step changes*

The first tempo changes which are systematically investigated are step changes. Examples of the reaction of the IPF to those sudden, discrete changes in tempo are shown in Figure 2. The IPF and the click track always start at 120 *bpm*, the *spontaneous motor tempo* introduced above. In general, three different reactions can occur:

- The IPF converges to the tempo of the click track (see Figure 2 a) and b))
- IPF is not effected by the click track, as the change in tempo was larger than a thirty-second note (see Figure 2 c))
- IPF converges to a multiple tempo of the click track (double-time), due to the modulo operation described by Equation (3) (see Figure 2 d))

If the IPF synchronizes to the click track, it does not follow the changes instantaneously. Thus, there is always a transient oscillation. This is similar to humans' reaction to sudden changes in tempo, where often a considerable *overshoot* has been reported in the literature (see e.g. [20, 17, 33, 22]).

However, the IPF shows more distinct oscillations in contrast to those cases. Listening to the audio examples provided by [32], this transition sounds familiar and natural, even though it may remind of a relatively inexperienced musician. Systematically increasing and decreasing the tempo in the same range (100 different values) and measuring the IPF's convergence time to $\pm 2\%$ (the threshold for conscious awareness of a tempo [20, 22]) reveals a mean value of 4 and a standard derivation of 1.46, which may be high but still in good accordance with the literature [20, 22, 17].

It does not have to be a disadvantage that not all tempo fluctuations of the click track are followed by the IPF, as those fluctuations may refer to dotted notes or syncopations, and the IPF should follow the meter of the click track. In Figure 2 c), the click track is slowed down from 120 *bpm* to 90 *bpm*. Hence, a two-against-three rhythm occurs, as the IPF does not synchronize to the new tempo. Two-against-three rhythms can be produced using dotted notes or triplets, so this example shows that the IPF is not affected by those and remains to the fundamental meter.

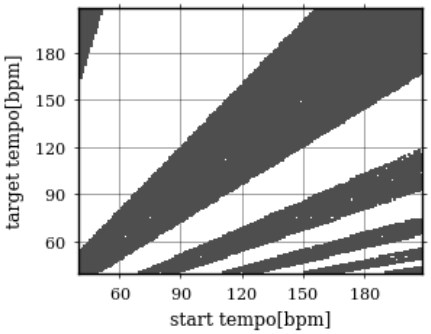


Figure 3. Ability of the IPF to synchronize if the click track performs a step change from a tempo denoted at the abscissa to a tempo denoted at the ordinate. Gray regions correspond to synchronization.

To give an overview, when synchronisation, and thus a change in tempo, occurs, every combination of start and target tempo is modeled. The results are shown in Figure 3. The IPF synchronizes in a broad area around the initial tempo and some smaller regions related to the dividers provoked by Equation 3. Further, synchronization can be detected when the tempo changes from very slow to swift tempi. This area shows the limitation of the synchronization threshold for changes larger than a thirty-second note. For instance, thirty-second notes at a quarter speed of 40 *bpm* could also be described as a sequence of 320 *bpm*, which equals eighth notes at a quarter speed of 160 *bpm*. Figure 3 shows this is precisely the threshold for synchronization at large steps to swift tempi.

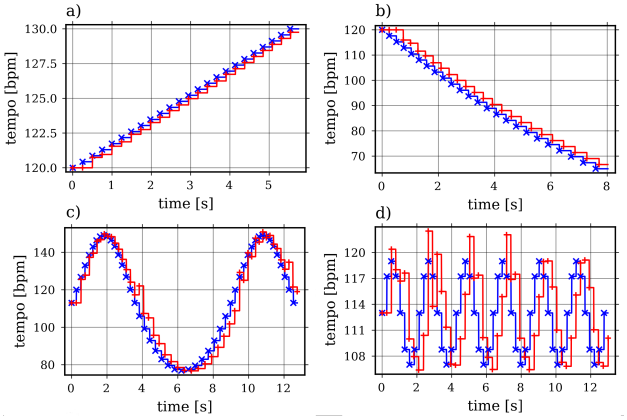


Figure 4. The IPF reacting to the same changes in tempo as shown in Figure 2 when changing the tempo linearly or sinusoidal during 24 beats instead of step changes. Blue lines refer to the tempo of the click track. Red lines correspond to the tempo of the IPF. The single crosses represent single beats. Audio examples of these four cases can be found at [32]

3.2.2 *Global perturbations*

The examples discussed in Section 3.2.1, above, are somewhat artificial. They assume the tempo of the click track to be constant and that tempo changes happen suddenly.

When several musicians play together in an ensemble, the tempo is changed at a somewhat steady rate due to musical expression or fluctuates due to the musicians' rhythmical skills (see, e.g., [34, 33]). Thus, this section investigates the IPF's reaction to linear and sinusoidal tempo changes and noisy click tracks.

Figure 4 shows the same examples as Figure 2, but now the tempo change is applied linearly and sinusoidal during 24 beats, instead of suddenly from one beat to another. It is striking that the IPF is always capable of following the click-track changes. Nevertheless, some minor phase differences $\Delta\Phi$ between the two signals are conspicuous, especial at higher modulation rates.

The examples presented in Figure 5 show different types (Brownian noise vs. white noise) and amounts (1 % and 5 %) of noise. A first observation shows that small amounts of noise and slow fluctuation lead to minor minor deviations between the IPF and the click track. In contrast, substantial and sudden fluctuations increase this error. Similar to the results shown in Figure 2, the IPF is likely to *overshoot*, especially at large changes of tempo.

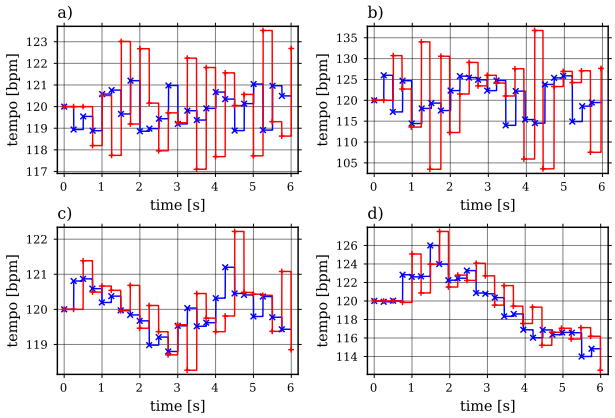


Figure 5. IPF adapting to a noisy click track: the upper line (a) and b)) shows white noise, and the lower line (c) and d) shows Brownian noise. On the left (a) and c)), the fluctuation is ± 1 %, and on the right (b) and d)) ± 5 %. Blue lines refer to the tempo of the click track. Red lines correspond to the tempo of the IPF. The single crosses represent single beats. Audio examples of these four cases can be found at [32]

3.3 Polyrhythmic tempo detection

In some musical applications, it might be wanted that the IPF synchronizes to a broader range of tempos (not like shown in Figure 3) or that the bandwidth of possible tempos for the IPF is reduced to prevent large tempo changes. Both can be done by allowing polyrhythmic beats.

Handel [21] investigated how people tap along polyrhythmic patterns. Applying models of damped resonating oscillators on their data shows that most people prefer tapping to sub-patterns with a frequency close to 120 bpm [31]. The approach can be inverted and transferred to tempo detection of the IPF model. Not the actual tempo of the click track is taken into account, but a different tempo closer to 120 bpm, which builds up a regular polyrhythm to the click track. Therefore only minor changes need to be applied to the tempo detection method introduced in Section 3.1:

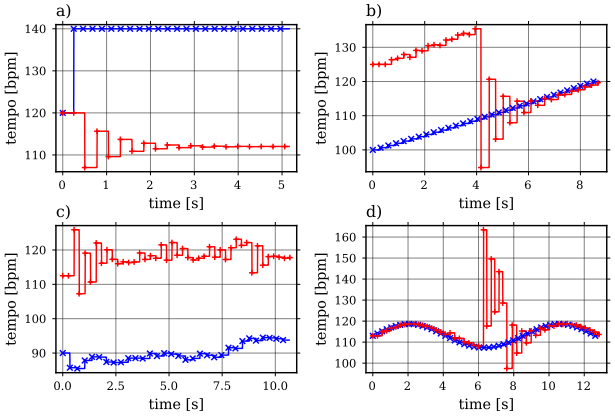


Figure 6. Several scenarios shown in Figures 2, 4 and 5 applied to polyrhythmic tempo detection: a) step change from 120 to 140 bpm, b) linear change from 100 to 120 bpm, c) 5 % Brownian noise added to a 90 bpm click track and d) sinusoidal modulation with a period length of 32 eight notes varied ± 6 bpm around 113 bpm. Blue lines refer to the the tempo of the click track, red lines corresponds to the tempo of the IPF. The single crosses represent single beats. Audio examples of these four cases can be found at [32]

- First, appropriated polyrhythmic relations have to be chosen. Here, the ratios used by Handel [21] (2/3, 3/4, 2/5, 3/5, and 4/5) as well as their non-polyrhythmic counterparts (1/1, 1/2, 1/3, 1/4, and 1/5) are chosen, which corresponds to a Farey sequence of order 5 (see, e.g., [27], [35, pp. 515]).
- In Section 3.1 the tempo was investigated by calculating the time difference between two consecutive beats of the click track. Now, this difference is multiplied by each of the factors chosen above as well as their reciprocal.
- The result closest to 250 ms (which relates to eighth notes at 120 bpm) is chosen as the new tempo.
- This way, differences larger than a thirty-second note must no longer be ignored.

Combining this tempo detection with the IPF model derived in Section 3.1 and determined by Equation (2), the IPF is capable of synchronizing in a polyrhythmic manner. Examples of different applications of the polyrhythmic tempo detection are shown in Figure 6. All Examples are related to four against five polyrhythms. In a) the click track suddenly jumps from 120 bpm to 140 bpm. Instead of following this increase (like the algorithm in Section 3.2 would have done), the IPF slows down to 112 bpm resulting in a $4/5$ ratio. In b), the tempo of the click track is increased linearly from 100 bpm to 120 bpm. Here the IPF starts with a $4/5$ ratio at 125 bpm and follows the increase of the click track with the same ratio. However, in contrast to Figure 4, the linear increase is a bit noisier. The prediction of the click tracks tempo changes does not work as well as in Figure 4, either. As soon as the click track reaches 110 bpm, the IPf starts to synchronize in a 1/1 ratio. Similar results can be seen in d). The IPF can still follow the sinusoidal fluctuations around 130 bpm, but as soon as the click track drops below 110 bpm, the IPF tries to synchronize with a $4/5$ ratio. As the settling time

of the IPF is too long, the click track is above 110 bpm again, before the IPF has reached a $4/5$ ratio and the IPF synchronizes at a 1/1 ratio again. Finally, in c), the click track fluctuates around 90 bpm, and the IPF follows with a $4/5$ ratio around 112.5 bpm. The IPF can follow perturbations caused by Brownian noise, but it seems to overreact in some cases.

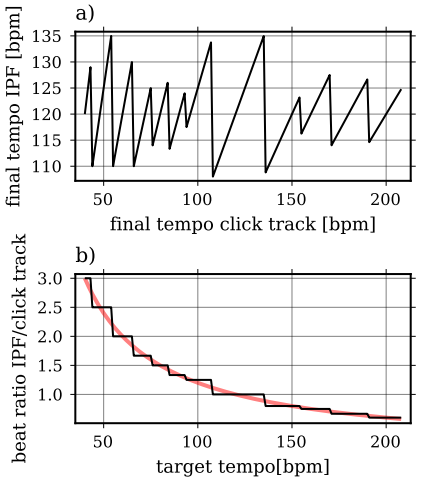


Figure 7. IPFs behavior on different step functions. The start tempo is always 120 bpm, and the final tempo T_F of the click track (32 eighth notes long) is always displayed at the axis of abscissas. a) shows the corresponding final tempo of the IPF and b) shows the related ratio of final tempos of IPF over click track (which corresponds to the different polyrhythmic ratios). The red line represents $120/T_F$, which is true if no synchronization happens.

The behavior of this tempo detection can be systematically investigated by applying different step changes in tempo, similar to Section 3.2.1. The results are shown in Figure 7. Focussing on b) one can see the different polyrhythmic ratios as discrete steps in the ratio of the different final tempos. The red line indicates that the click track and the IPF always synchronize. Further, a) shows that now the tempo range of the IPF is significantly reduced to $108 - 135$ bpm and sudden steps in tempo occur when the polyrhythmic ratio is changed. The bandwidth of the IPF could be extended again if the number of possible polyrhythmic ratios is reduced. Further, the favorable 120 bpm in the tempo detection can be easily changed to any arbitrary value to shift the rhythmical center, if this is helpful for specific musical applications.

4. DISCUSSION

Applying the IPF to the rhythmical synchronization of musicians is a straightforward modeling approach that leads to convincing results, even in the most simple form defined by Equation (2), as can be perceived when listening to the provided audio examples [32] or comparing the numerous plots to the discussed literature. Similar to human behavior, the IPF adapts to step changes in tempo after a short, chaotic transition of approximately four beats and is capable of predicting regular changes in tempo and quickly adapts to them, as discussed in Section 3.2.2. Applied to noisy signals, the IPF shows a mixture of those two behaviors. However, the required algorithm for tempo detection and the scaling of the system parameters can have a crucial impact on the results gained by the IPF.

Even though tempo synchronisation could be achieved straightforwardly, there are always significant phase shifts between the beats, the click track, and the IPFs. This can be easily resolved by adding a second β, which compensates for phase issues. This approach will be discussed in detail in future publications. However, when applied to real-time music scenarios, it has been shown that these phase differences are not a critical issue. On the one hand, they would occur anyway due to imperfections in today's computer systems, with a small latency in ADCs or numerical calculations. Further, they are usually compensated unconsciously by musicians interacting with an IPF system.

Nevertheless, the IPF has been proven to be a promising tool for modeling human musical rhythm synchronization. Thus, it is likely that it can be applied to other aspects of the perception and production of music and, one day, might model the entire process. However, the straightforward formulation and the very low computational costs make this model also a promising tool for real-time application.

Drum machines or computer-based click tracks are standard in today's music and used in recording studios and live concerts. These solutions are convenient for musicians and recording engineers and allow techniques (like, e.g., combining pre-recorded samples with live music) that are hardly possible when the musicians choose the tempo in a self-organized manner. Further, it has been shown that certain musical genres and pieces have a fixed, very precise tempo most pleasing for the audience (see, e.g., [36]).

Nevertheless, in other situations, those fixed tempi are perceived as a bit artificial (or even dull) because of the fixed beat and the lack of minor imperfections that inevitably occur when human musicians perform independently. Those deviations may be intended due to musical expression (e.g., agogic accents) or unintended due to human imperfections. As explained in the Introduction, they are often simulated using long-range correlated noise, which is not always suitable (see, e.g., [15]). However, the proposed IPF model can be used to replace drum machines and click tracks with more musical and creative solutions in live music and recording scenarios. By fine-tuning the modeling parameters, one can decide whether to model the realistic behavior of a human musician or to create novel, unfamiliar systems with a distinct, characteristic manner of rhythmical fluctuation and synchronization.

Acknowledgments

This research was funded under the Program "Innovative Hochschule" (innovative university) by the Federal Ministry of Research, Technology and Space (BMFTR) of Germany and the City of Hamburg (Grant No. 13IHS232C) .

5. REFERENCES

[1] R. Bader, *Nonlinearities and synchronization in musical acoustics and music psychology*, ser. Current research in systematic musicology. Berlin [et al.]: Springer, 2013, vol. 2.

[2] S. Linke, R. Bader, and R. Mores, "The Impulse Pattern Formulation (IPF) as a nonlinear model of musical instruments," in *Proceedings of the International Symposium on Music Acoustics 2019 - ISMA 2019*, M. Kob, Ed., Berlin, 2019, pp. 336–345. [Online]. Available: http://pub.dega-akustik.de/ISMA2019/data/ISMA_proceedings_all.pdf

[3] ——, "Influence of the supporting table on initial transients of the fretted zither: An impulse pattern formulation model," in *180th Meeting of the Acoustical Society of America*, ser. Proceedings of Meetings on Acoustics. ASA, 2021, p. 035003.

[4] S. Linke, R. Mores, and R. Bader, "Describing minimum bow force using Impulse Pattern Formulation (IPF)–an empirical validation," in *Proceedings of Meetings on Acoustics*, vol. 49, no. 1. AIP Publishing, 2022.

[5] R. Bader, "Impulse pattern formulation (IPF) brain model," 2022.

[6] ——, "Modeling Temporal Lobe Epilepsy during Music Large-Scale Form Perception Using the Impulse Pattern Formulation (IPF) Brain Model," *Electronics*, vol. 13, no. 2, p. 362, 2024.

[7] R. Bader, S. Linke, and S. Gernert, "Nonlinear dynamical social and political prediction algorithm for city planning and public participation using the impulse pattern formulation," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 34, no. 9, 2024.

[8] M. Caetano and X. Rodet, "A source-filter model for musical instrument sound transformation," *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 137–140, 2012.

[9] A. Klapuri, "Analysis of Musical Instrument Sounds by Source-Filter-Decay Model," *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*, vol. 1, pp. 53–56, 2007.

[10] C. A. Coey, A. Washburn, J. Hassebrock, and M. J. Richardson, "Complexity matching effects in bimanual and interpersonal syncopated finger tapping," *Neuroscience letters*, vol. 616, pp. 204–210, 2016.

[11] D. L. Gilden, "Cognitive emissions of 1/f noise," *Psychological review*, vol. 108, no. 1, pp. 33–56, 2001.

[12] G. C. van Orden, J. G. Holden, and M. T. Turvey, "Self-organization of cognitive performance," *Journal of experimental psychology. General*, vol. 132, no. 3, pp. 331–350, 2003.

[13] H. Hennig, R. Fleischmann, A. Fredebohm, Y. Hagmayer, J. Nagler, A. Witt, F. J. Theis, and T. Geisel, "The nature and perception of fluctuations in human musical rhythms," *PloS one*, vol. 6, no. 10, p. e26457, 2011.

[14] N. Yamada, "Nature of variability in rhythmical movement," *Human Movement Science*, vol. 14, no. 3, pp. 371–384, 1995.

[15] H. Hennig, R. Fleischmann, and T. Geisel, "Musical rhythms: The science of being slightly off," *Physics Today*, vol. 65, no. 7, pp. 64–65, 2012. [Online]. Available: https://www.researchgate.net/profile/Holger-Hennig/publication/257020003_Musical_rhythms_The_science_of_being_slightly_off/links/56b3a0f408ae1f8aa45350dc/Musical-rhythms-The-science-of-being-slightly-off.pdf

[16] H. Haken, J. A. Kelso, and H. Bunz, "A theoretical model of phase transitions in human hand movements," *Biological cybernetics*, vol. 51, no. 5, pp. 347–356, 1985.

[17] M. H. Thaut, R. A. Miller, and L. M. Schauer, "Multiple synchronization strategies in rhythmic sensorimotor tasks: phase vs period correction," *Biological Cybernetics*, vol. 79, no. 3, pp. 241–250, 1998. [Online]. Available: https://idp.springer.com/authorize/casa?redirect_uri=https://link.springer.com/article/10.1007/s004220050474&casa_token=be_x34xdjouaaaaa:f2t0tiwtwz3ifj9hl0xj1xidr06qaditz9xeeiuaurxgr70ergo0eh1bfjlogkr

[18] O. D'Huys, T. Jüngling, and W. Kinzel, "On the Interplay of Noise and Delay in Coupled Oscillators," in *Control of Self-Organizing Nonlinear Systems*. Springer, Cham, 2016, pp. 127–145. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-28028-8_7

[19] W. Just, P. M. Geffert, A. Zakharova, and E. Schöll, "Noisy Dynamical Systems with Time Delay: Some Basic Analytical Perturbation Schemes with Applications," in *Control of Self-Organizing Nonlinear Systems*. Springer, Cham, 2016, pp. 147–168. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-28028-8_8

[20] J. A. Michon, *Timing in temporal tracking*. Assen: Van Gorcum, 1967. [Online]. Available: http://www.jamichon.nl/jam_writings/1967_ttt_psyfor.pdf

[21] S. Handel and J. S. Oshinsky, "The meter of syncopated auditory polyrhythms," *Perception & Psychophysics*, vol. 30, no. 1, pp. 1–9, 1981.

[22] B. H. Repp, "Processes underlying adaptation to tempo changes in sensorimotor synchronization," *Human Movement Science*, vol. 20, no. 3, pp. 277–312, 2001.

[23] N. H. Fletcher and T. D. Rossing, *The physics of musical instruments*, 2nd ed. New York, NY: Springer, 2010.

[24] M. J. Feigenbaum, "Quantitative universality for a class of nonlinear transformations," *Journal of Statistical Physics*, vol. 19, no. 1, pp. 25–52, 1978.

[25] S. Linke, R. Bader, and R. Mores, "Multiphonic modeling using impulse pattern formulation (ipf)," 2022.

[26] ——, "The impulse pattern formulation (IPF) as a model of musical instruments—Investigation of stability and limits," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 29, no. 10, p. 103109, 2019.

[27] H. Haken, C. E. Peper, P. J. Beek, and A. Daffertshofer, "A model for phase transitions in human hand movements during multifrequency tapping," *Physica D: Nonlinear Phenomena*, vol. 90, no. 1-2, pp. 179–196, 1996. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0167278995002359

[28] A. Pikovsky, M. Rosenblum, and J. Kurths, *Synchronization: A universal concept in nonlinear sciences*, ser. The Cambridge nonlinear science series. Cambridge: Cambridge University Press, 2001, vol. 12.

[29] C. E. Collyer, H. A. Broadbent, and R. M. Church, "Preferred rates of repetitive tapping and categorical time production," *Perception & Psychophysics*, vol. 55, no. 4, pp. 443–453, 1994.

[30] R. Parncutt, "A Perceptual Model of Pulse Salience and Metrical Accent in Musical Rhythms," *Music Perception*, vol. 11, no. 4, pp. 409–464, 1994.

[31] L. van Noorden and D. Moelants, "Resonance in the Perception of Musical Pulse," *Journal of New Music Research*, vol. 28, no. 1, pp. 43–66, 1999.

[32] S. Linke, R. Bader, and R. Mores, "Sound examples of an Impulse Pattern Formulation model synchronizing to different click tracks," 2021. [Online]. Available: https://zenodo.org/records/5758991

[33] B. H. Repp, "Sensorimotor synchronization: a review of the tapping literature," *Psychonomic Bulletin & Review*, vol. 12, no. 6, pp. 969–992, 2005. [Online]. Available: https://link.springer.com/article/10.3758/bf03206433

[34] ——, "Diversity and commonality in music performance: an analysis of timing microstructure in Schumann's Traeumerei," *The Journal of the Acoustical Society of America*, vol. 92, no. 5, pp. 2546–2568, 1992.

[35] J. H. Argyris, G. Faust, M. Haase, and R. Friedrich, *An Exploration of Dynamical Systems and Chaos: Completely Revised and Enlarged Second Edition*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015.

[36] G. Datseris, A. Ziereis, T. Albrecht, Y. Hagmayer, V. Priesemann, and T. Geisel, "Microtiming Deviations and Swing Feel in Jazz," *Scientific reports*, vol. 9, no. 1, p. 19824, 2019.

Extending Xenakis: From Architectural Geometry to Sonification of the Philips Pavilion

Changda Ma
Georgia Institute of Technology
cma326@gatech.edu

Sunshiyu Wang
Georgia Institute of Technology
swang3214@gatech.edu

Canting Zhu
Georgia Institute of Technology
czhu348@gatech.edu

Alexandria Smith
Georgia Institute of Technology
alexandria.smith@gatech.edu

ABSTRACT

Architecture and music have been linked through proportion and temporal structure, yet architectural geometry is rarely viewed as a source of generative music. Revisiting Xenakis' one-directional transformation from string glissandi in Metastaseis to the ruled surfaces of the Philips Pavilion, we invert this workflow and sonify the completed Pavilion as a temporal composition. We reconstruct the Pavilion as nine ruled surfaces, extract their governing ruling lines, and subdivide each surface into structural lines and spatial sampling points. Four evenly spaced ruling lines per surface generate continuous string glissandi, while 3,357 sampled points develop five density-based energy blocks and a sparse brass and woodwind subsequence. Implemented in Python, the system produces MIDI rendered in Ableton Live, accompanied by a real-time 3D visualization that reveals architectural motion, stasis, and structural contrast through sound and image. In general, this work paves the way for the transfer of architectural geometry as a performable musical structure, extending Xenakis's architectural and musical thinking to sonification and interactive music practice.

1. INTRODUCTION

"Music is liquid architecture; architecture is frozen music." This famous phrase, attributed to Johann Wolfgang von Goethe, describes a structural analogy between architecture and music [1]. However, contemporary architectural practice reflects architecture as a medium for transmitting music, rather than as a generator of musical structure. Few studies investigate how architectural geometry itself can be translated into musical logic.

A foundational exploration linking architecture and music at the level of form generation traces back to the work of Iannis Xenakis. In *Metastaseis*, Xenakis projected the mathematical hyperbolic paraboloid onto the string glissandi using a graphic notation of time and pitch [2]. Therefore, the slope, continuity, and density of lines in the hyperbolic paraboloid projection determine the pitch change, continuity, and ensemble density [3]. Together with the modernist architect Le Corbusier, Xenakis translated music composed of glissando patterns into architectural form in the *Philips Pavilion* [4, 5], where the musical structure informed the architectural form. However, the translation remained unidirectional: the architecture appeared as a static outcome rather than an active musical agent.

Recent advances in architectural parametric design [6] allow architectural elements such as surfaces, vertices, and ruled geometries to be accessed as structured and data-driven representations rather than fixed objects [7, 8]. This transformation makes architectural geometry suitable for sonification. Sonification, as a method for translating data into non–speech sound [9], provides a new acoustic dimension for architectural parametric design.

This study explores the sonification of architecture by reversing the historical process established by Xenakis. Rather than reproducing Xenakis's instrumentation, our approach takes the completed *Philips Pavilion* and maps its geometry back into sound. Using parametric architectural modeling, we extract the ruled-surface geometry of the *Philips Pavilion* as spatial data for sonification. We map ruled lines to continuous string glissandi, use variations in point density to define energy-based blocks of stasis, and employ selected sampling points to generate the structure through sparse brass counterpoints. By mapping architectural geometry into sound, this work reframes architecture as an active generative instrument and extends Xenakis's architectural–musical thinking through contemporary computational methods.

2. BACKGROUND AND LITERATURE REVIEW

In the eighteenth century, Gotthold Ephraim Lessing distinguished architecture as a spatial art and music as a temporal art [10]. While early twentieth-century movements and Le Corbusier's concepts challenged this separation by introducing narrative or perceptual time [11–13], architectural space remained fundamentally static. Even as composers such as Stockhausen explored the spatial dynamics of sound [14], architecture continued to serve primarily as a passive container, lacking generative feedback from musical organization. Xenakis's practice achieved a breakthrough in the relationship between music and architecture. By transferring the logic of musical structure to architectural form, he used mathematics to generate the geometric

Copyright: ©2026 Changda Ma et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License 3.0 Unported, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

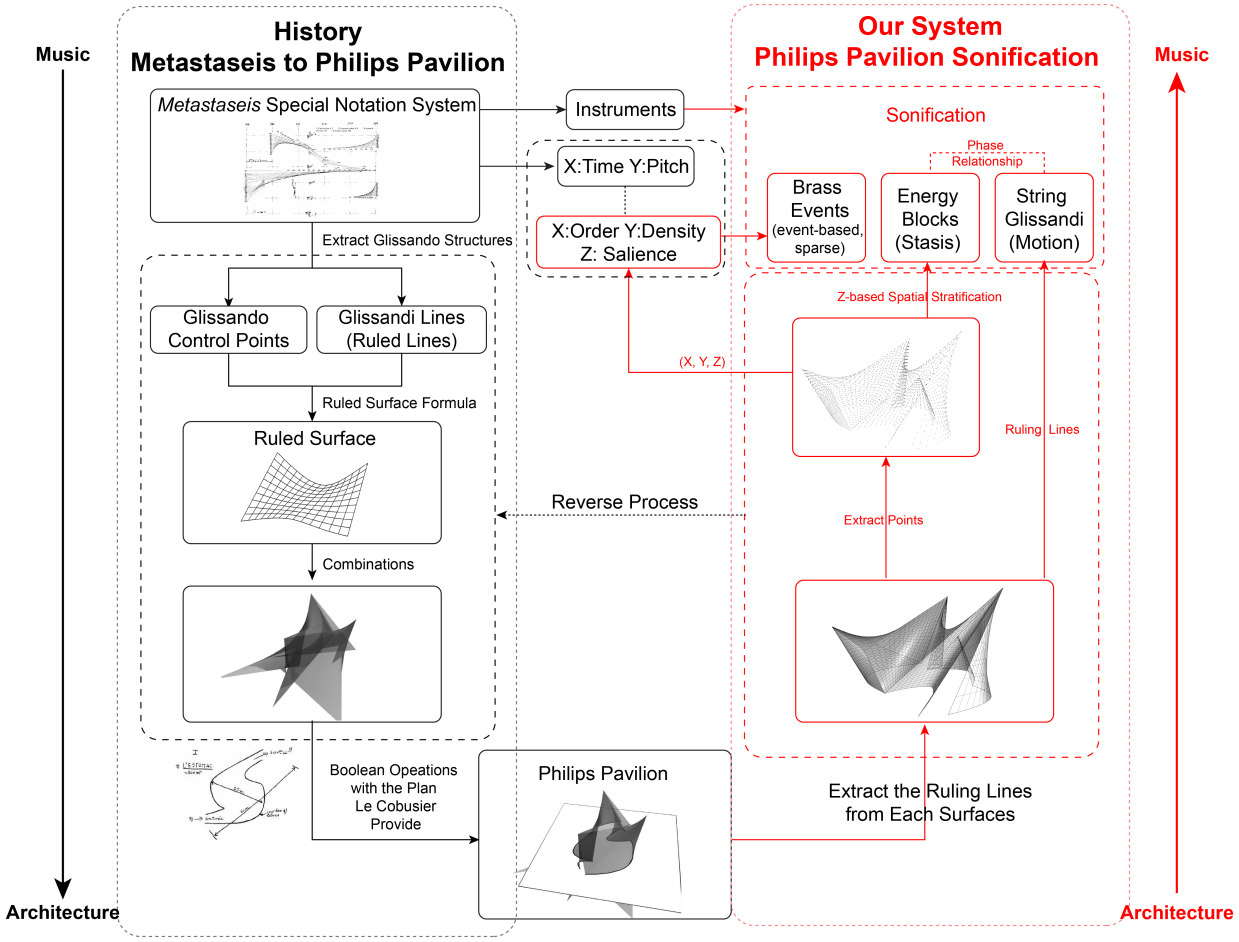


Figure 1. System workflow illustrating the inversion of Xenakis' historical music to architecture process.

form of the *Philips Pavilion* [15]. Rigorous mathematical logic was first used to establish a bridge between musical structure and architectural design principles [16]. However, due to the absence of computational tools for extracting architectural geometry as quantifiable data, this relationship could not be operated in reverse. Architectural form remained an endpoint rather than a generative source of music.

Despite this methodological limitation, the theoretical discourse has recognized the bi-direction of Xenakis's vision. Beyond Anglophone literature, French scholarship emphasizes the reciprocal dialog inherent in Xenakis's interdisciplinary practice. Studies by Nathalie Boulouch [17] and Makis Solomos [18] frame the *Philips Pavilion* not only as a static artifact derived from music, but as a disciplinary crossing where sonic and architectural thought remain entangled.

Although advances in computational design have made architectural geometry available as data, few studies have explored architectural geometry itself as a generative source for sonification. Several studies have drawn inspiration from Xenakis to sonify architectural or urban forms.

Parthenios [19] adopted Xenakis's *Metastaseis*–to–*Philips Pavilion* music-architecture mapping strategy to sonify architectural facades in urban contexts. Although spatial dimensions and surface density were assigned to musical parameters, this research used correspondences without preserving the historical and structural logic from the architectural–musical synthesis of Xenakis.

Torresan et al. [20] demonstrated how structural vibration data can be sonified by translating measured dynamic responses, such as acceleration and modal behavior, into sound. However, this approach represents a system-response sonification based on engineering data rather than a sonification of the architectural geometric structure itself.

In contrast, research on the sonification of architectural geometric structures remains relatively scarce. This gap motivates our system, which implements ruled-surface lines and point distributions as musical control structures.

3. SYSTEM DESIGN

3.1 History-based System Overview

Drawing inspiration from Xenakis's compositional approach in *Metastaseis* and its architectural realization in the *Philips Pavilion*, our system revisits this historical transformation between sound and space. In *Metastaseis*, Xenakis used the distinctive glissando notation system to extract the control points and projection lines of the time-pitch traces for each glissando. To transform these linear sounds into the design logic of a spatial surface, he applied the mathematical principles of the ruled surface, which was shown in Formula 1. A ruled surface is formed by a family of geriatrics sweeping along a directrix [21]. Thus, surfaces can

be generated entirely by lines.

$$S(u,v) = (1-v)\,\boldsymbol{r}_1(u) + v\,\boldsymbol{r}_2(u), \qquad u,v \in [0,1] \quad (1)$$

where $\boldsymbol{r}_1(u)$ and $\boldsymbol{r}_2(u)$ are the two ruling curves, and linear interpolation between them defines the surface geometry.

Through the selection and organization of glissandi with varying directional tendencies, Xenakis constructed multiple ruled surfaces that together defined the spatial and structural logic of the *Philips Pavilion* [16]. This process establishes linear trajectories as a shared structural unit between music and architecture. Building on this principle, our system reverses this historical workflow by treating the completed architectural geometry of the *Philips Pavilion* as a generative source for musical structure, using extracted lines and point distributions to drive motion, stasis, and event-based behaviors in sound.

3.2 System Implementation: Extending Xenakis–Inverting the History Process

We adopt the principles through which *Metastaseis* was transformed into the *Philips Pavilion* and invert this workflow to design our "Extending Xenakis" system. From the history to our system, the full workflow of our system is illustrated in Figure 1.

Based on the ruled surface formula, we deconstruct the nine ruled surfaces of the Pavilion by extracting their ruling lines and applying parametric sampling points along these lines. The spatial points (x, y, z) generated by this procedure were used as data sources for our sonification process. Although *Metastaseis* was processed within a two-dimensional time–pitch framework, its logic lies not in a fixed coordinate system but in the continuous transformation of geometric form into sound. Following this principle, we reinterpret the sampled spatial coordinates as a multidimensional control space for the musical structure, rather than as a direct parameter mapping. Within this framework, we extract ruling lines to generate continuous string glissandi that establish large-scale motion and convergence in pitch. We organized the distributions of points, stratified along the vertical dimension, into energy-based blocks that represent moments of relative stasis following the completion of the glissandi. In parallel, we select a reduced subset of points to generate sparse brass and woodwind events, forming a secondary, event-based layer that strengthens the structural endpoints of the string process. By applying the same instruments used in *Metastaseis*, the geometric form of the *Philips Pavilion* was sonified as a temporal, audible inversion of the historical process through which Xenakis transformed music into architecture. [1] .

3.3 Architecture Geometry Processing

We reconstructed the geometry of the Philips Pavilion in Rhino based on archival drawings and historical documentation. Using parametric modeling in Grasshopper, we rebuilt the *Philips Pavilion*'s ruled surface system and extracted the nine ruled surfaces from the envelope. For each given ruled surface $S(u,v)$, we recovered two governing ruling lines by the inversion process of Eq. 2, and Figure 2 was shown as the result. These ruling lines correspond to the generative trajectories that Xenakis associated with musical glissandi.

$$\begin{cases} \mathbf{r}_1(u) = S(u,0), \\ \mathbf{r}_2(u) = S(u,1), \end{cases} \qquad u \in [0,1]. \quad (2)$$

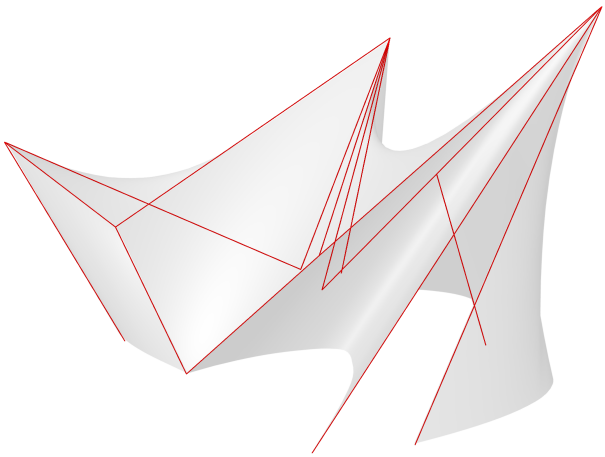


Figure 2. Ruling lines extracted from the Pavilion surface via the inverse ruled-surface process.

To capture continuous variations in curvature, orientation and spatial gradient, we first subdivided each ruled surface into 20 interpolated structural lines between its two governing ruling lines (Figure 3). This dense line sampling provides a high-resolution geometric representation of each surface and paves the way for subsequent sonification processes.

From these 20 structural lines, we selected four evenly spaced lines per surface as the geometric backbone of the string section, yielding a total of 36 structural lines across the Pavilion. For each selected line, we extracted its start point, end point, and total length, which together define the path and duration of each string glissando, as shown in Figure 3.

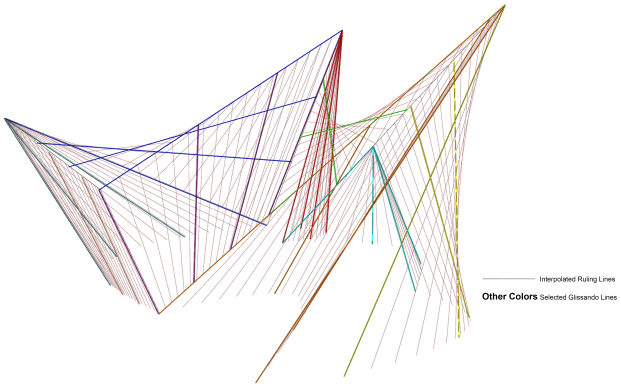


Figure 3. Interpolated structural lines generated between the ruling lines of each surface.

In addition, we sampled 20 evenly distributed spatial points along each structural line, yielding a total of 3,357 spatial

[1] Code is available at: https://github.com/WangSun725/Extending-Xenakis-Architecture-to-Midi-.git

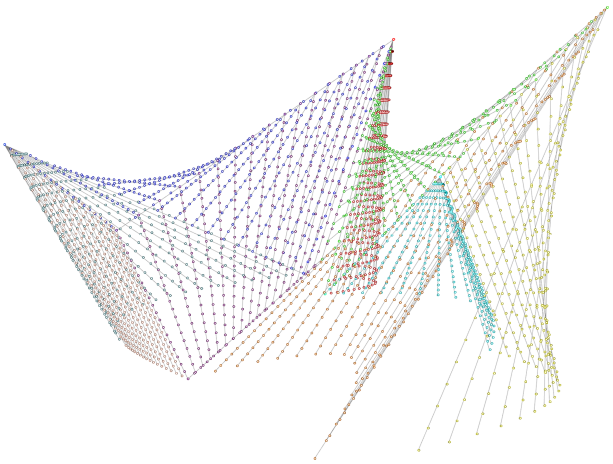


Figure 4. Spatial sampling points extracted from the Pavilion's nine ruled surfaces, colored by surface groups.

points (Figure 4) that form the input dataset for energy-based blocks and event-based brass sonification.

Each sampled point was labeled according to its ruled surface, resulting in nine surface-based point groups. Our system preserves geometric coherence across the sonification process and applies a consistent temporal organization derived from architectural form.

3.4 Sonification Mapping

Our sonification system is designed both to generate a musical piece that represents architectural characteristics and to mirror Xenakis's architectural to musical logic. We create a set of algorithms to transform data extracted from architectural structures into a multi-layer MIDI composition organized in two musical phases.

We organize the composition in a manner analogous to *Metastaseis*, beginning with a continuous string glissando phase derived from ruled-surface line segments, followed by discrete energy blocks driven by spatial point distributions. We then select a reduced subset of sampling points to generate a brass and woodwind subsequence, mapping spatial positions to MIDI note events to enrich the musical texture.

The entire implementation is fully deterministic and is executed in Python, producing MIDI files that can be imported into Ableton Live for orchestration, positioning, mixing, and final rendering.

3.4.1 String Sonification

Our sonification output primarily focuses on string instruments. We divided the string materials into two sections: a "Glissando" section, followed by a "Block" section. In the "Glissando" section, we extract four line segments from each architectural surface, yielding a total of 36 lines. Each line segment is mapped to an independent string voice, resulting in an ensemble consisting of 12 violins, 8 violas, 8 cellos, and 8 double basses. We initialize all string voices in a common pitch class (G), adjusting the octave placement according to the pitch range of each instrument.

Each string instrument performs a sustained glissando, with MIDI pitch information derived directly from the geometric properties of its corresponding line segment. For each line segment, we compute its total geometric length from its start and end points in three-dimensional space. We normalize line lengths and apply a nonlinear shaping strategy to preserve perceptual contrast and avoid extreme temporal values, as shown in Eq. 3.

$$\hat{L}_i = \tilde{L}_i^{\gamma}, \tag{3}$$

where $\gamma > 1$ biases the distribution towards shorter durations. The final glissando duration T_i is obtained by mapping $\hat{L}_i$ to a bounded temporal range, as shown in Eq. 4.

$$T_i = T_{\min} + \hat{L}_i(T_{\max} - T_{\min}), \tag{4}$$

where $T_{\min}$ and $T_{\max}$ define the minimum and maximum allowed glissando durations.

We render pitch evolution as a continuous glissando rather than as a sequence of discrete note steps. For each line segment, we interpret the vertical displacement between its start and end points as a directional pitch tendency, scaled to the vertical extent of the Pavilion geometry. This vertical variation determines the range and direction of pitch movement, allowing each string voice to trace paths through the pitch space. We realize this pitch motion using continuous pitch bending, so that changes in height are perceived as uninterrupted glissandi rather than stepped intervals. Throughout the duration of each glissando, the pitch is updated gradually and evenly over time, maintaining perceptual smoothness and preserving the continuous character of the string texture.

At the end of the "Glissando" section, each string voice maintains the final pitch reached at the end of its path. After a manually defined one second rest, the subsequent string "Block" section begins, and all string instruments remain at this final pitch and organize synchronized tremolo gestures, forming a sequence of energy blocks with coordinated onsets and offsets.

Block durations are derived from the architectural spatial point data. We divide the Pavilion geometry into five equal-width sections along the global vertical axis and assign each sampled point to a block index based on its z-coordinate. For each block, we compute the number of points as a proxy for the local structural density. These values are linearly mapped to block durations within a predefined temporal range. When all blocks contain equal numbers of points, a uniform midpoint duration is assigned. The five blocks are arranged in order, separated by fixed rests, and all string voices end simultaneously at the end of the section.

3.4.2 Brass and Woodwind Subsequence

To enrich the musical texture and fully represent the architectural to musical transformation, we convert a subset of spatial points into nine MIDI tracks assigned to brass and woodwind instruments, forming a secondary structural layer synchronized with the string texture. Unlike the string section, which is derived from ruled lines and represents continuous geometric trajectories, the brass and woodwind layer is derived from discrete point samples and is designed to articulate localized structural moments as punctuated events rather than sustained motion. In this way, the two layers have a distinction presented in the architectural data:

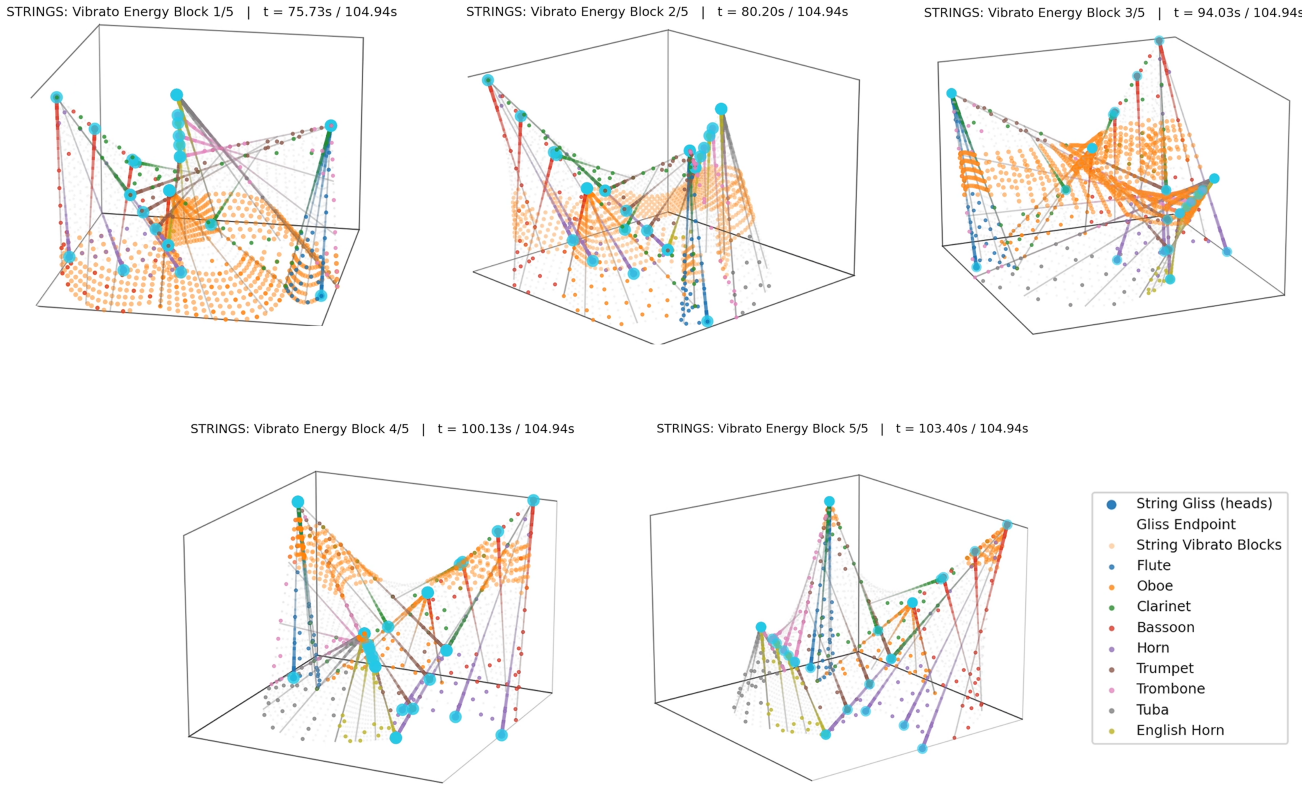


Figure 5. Real-time visualization during the energy block phase.

lines are rendered as continuity, whereas points are rendered as discreteness.

The brass and woodwind section consists of flute, oboe, clarinet, bassoon, horn, trumpet, trombone, tuba, and English horn. A reduced set of architectural points is sampled from the full point dataset and distributed across the nine instruments. This reduction is a structural control strategy that also remains consistent with the instrumental logic inherited from *Metastaseis* [22]. If the entire point dataset were converted into note events, the density of the piece would saturate the musical texture and obscure the distinction between the continuous, surface-based motion of the strings and the event-based articulation of the point layer. By limiting the number of active points, the brass and woodwind subsequences remain subordinate to the string section, while preserving the presence of discrete geometric features within the whole sonification.

Each point is represented by a three-dimensional coordinate (x_i, y_i, z_i) and mapped to a MIDI note event through a structured spatial to musical transformation. We use the vertical coordinate z_i to encode pitch information by normalizing it relative to the global vertical extent of the dataset and mapping it to a bounded pitch range. For brass and woodwind instruments, the resulting pitch values are discretized and quantized to a diatonic pitch set in G major, yielding discrete MIDI note numbers. This discrete pitch mapping distinguishes the brass and woodwind layers from the continuous pitch trajectories used in the string section.

We determine the temporal placement of each event from the horizontal coordinate x_i. Each x_i is linearly mapped to an onset time t_i within the global timeline defined by the string section and quantized to a fixed rhythmic grid. The remaining coordinate y_i modulates the event density and instrumental distribution rather than directly encoding the pitch or time. To maintain perceptual clarity, we encode a minimum inter-onset interval within each individual instrument track.

During the rest portions of the string section , we further thin brass and woodwind activity through probabilistic downsampling, reducing event density while preserving continuity across the full timeline. Through these combined mappings and constraints, the brass and woodwind layer functions as a supporting and complementary component that punctuates and reinforces the structural endpoints of the overall sonification process.

3.5 Real-Time Visualization

We implemented a real-time sonification visualization developed in Python using matplotlib's 3D toolkit and FuncAnimation for real-time animation, with video export performed via FFMpegWriter [23].

Rather than introducing an independent visual timeline, we drive all visual events from the same global time reference used in the sonification system. Therefore, spatial activation, musical structure, and animation playback remain synchronized throughout the composition.

The visualization operates on two architectural datasets from the Pavilion geometry: ruled-surface line segments and spatial sampling points. These datasets correspond to the string glissandi, energy blocks, and brass events introduced in our sonification system.

Before the animation, the architectural geometry of the *Philips Pavilion* is presented as a low-opacity, light-gray point cloud, together with 36 glissandi ruling lines shown

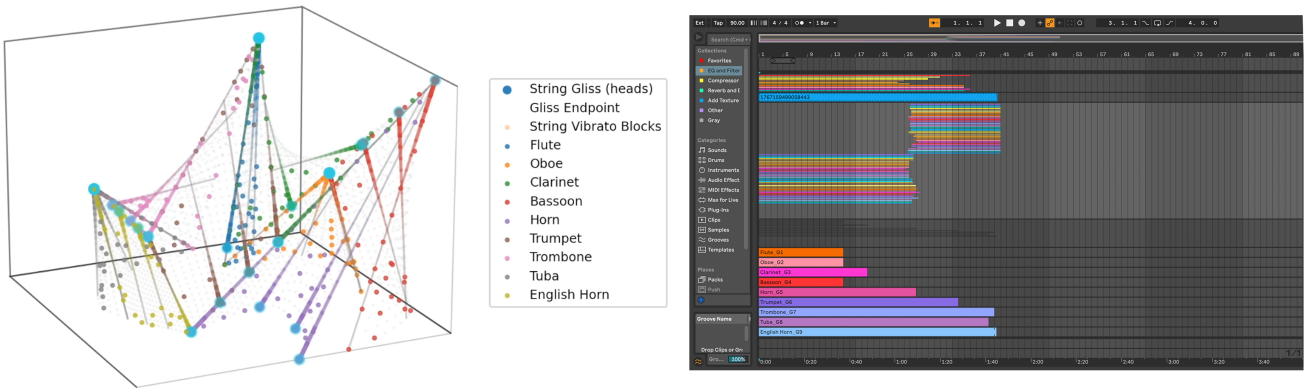


Figure 6. Real-Time Sonification Visualization System

in a muted, inactive state. These background elements remain visible throughout the visualization, providing a spatial context and the form of the Pavilion. Each ruled-surface group maintains a consistent color identity across phases, so the same architectural groups remain legible whether they are shown as lines, blocks, or event points.

During the opening animation, each string glissando is visualized as a colored traversal from the start point to the end point of its corresponding line segment. A bright moving head and a short trailing trace make the direction and continuity of motion perceptible, while the gray scaffold remains a stable reference of the geometry. In this phase, the endpoints are kept visually suppressed, so the attention is focused on the collective motion rather than the destination. The glissando motion process is shown in Figure 7.

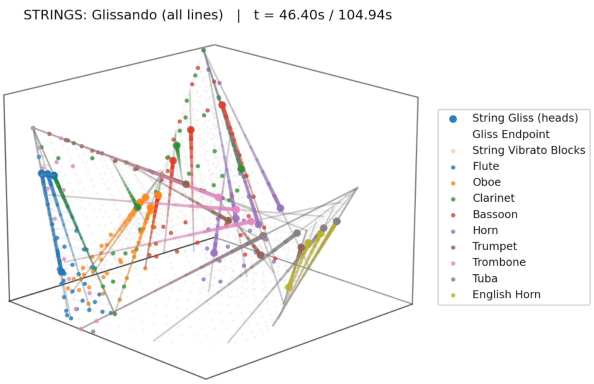


Figure 7. Real-time visualization during the string glissando phase.

After all glissandi are complete, we insert a brief gap in which endpoints appear as dim, steady markers, showing that the string texture has arrived and is holding its final pitch state. The visualization is then processed into the five energy blocks: points are grouped into five strata along the vertical dimension and activated as dense, colored fields that organize moments of relative stasis. Through the "Block" section, the endpoints remain highlighted and pulsed, presenting the idea that the blocks unfold while the string process remains anchored at its accumulated endpoint state, as shown in Figure 5.

In parallel with the glissandi and energy block phases, we activate a reduced subset of sampling points as discrete brass and woodwind events. These appear as small, color-coded point triggers whose onsets follow the beat-quantized onset equation (Eq. 5), forming a sparse secondary layer that contrasts with the continuous string texture, as shown in Figure 7 and Figure 5.

$$\text{onset\_beats} = \text{round}\left(\frac{y_{\text{norm}} \cdot \text{total\_beats}}{\text{grid}}\right) \cdot \text{grid}. \quad (5)$$

During the animation, we maintain a slow and continuous camera rotation at a fixed elevation, allowing the three-dimensional organization of the Pavilion to be perceived without interrupting temporal continuity. The whole real-time visualization is shown in Fig 6 and video output. [2]

4. CONCLUSION

This study proposes an architectural sonification system focused on the *Philips Pavilion* created by Xenakis's *Metastaseis*. Through parametric geometric analysis, we revisited the generative logic behind Xenakis's transition from glissando structures to ruled surfaces in *Metastaseis*. We then reversed this process to sonify the *Philips Pavilion*'s geometry. Using parametric modeling, we reconstructed the Pavilion as nine ruled surfaces and extracted their governing ruling lines. We subdivided each surface into interpolated structural lines and selected four evenly spaced lines per surface as the backbone of the string section, yielding 36 line segments that drive continuous string glissandi. In addition, we sampled spatial points across the ruled surfaces and organized these point distributions into two complementary layers, comprising five energy blocks stratified along the vertical dimension to organize moments of stasis after the glissandi, and a reduced subset of points that triggers sparse brass and woodwind events to strengthen structural moments within the timeline. By adopting the instrumentation associated with *Metastaseis*, our system renders the Pavilion as a time-based musical form, allowing architectural structure to be perceived through motion, accumulation, and event-driven contrast. To more clearly visualize our architecture sonification system, we also developed a real-time visualization platform. The Pavilion appears as a low-opacity point cloud with inactive structural lines, while musical events activate geometric elements in sequence. Colored traversals reveal the unfolding

[2] Real-time visualization video output is available at: `https://youtu.be/zSj_I4n7Yqg`

of string glissandi along architectural trajectories; dense point activations visualize the energy blocks; and discrete point triggers mark brass and woodwind events. Together, the audiovisual system frames architectural geometry not as a static outcome but as an active generative instrument whose structure can be heard and viewed through time.

Our study not only reinterprets Xenakis's music-architecture process in reverse, but also suggests a new direction for exploring the auditory dimension of architectural design and interactive performance. With contemporary parametric tools, architecture can be viewed as a dataset for sonification. Similarly, sonification gives architects a new design perspective beyond visual intuition. Thus, architects can uncover structural logic, rhythmic relationships, and form potential at the auditory level, which were obscured in traditional geometric presentations.

However, there are several limitations in our study. From an architectural data perspective, the ruled surfaces were extracted using uniform interpolation and equivalent division of sampling points to ensure continuity in sonification. However, the ruled surfaces are complex and fixed-density structural lines and sampling points struggle to capture local curvature variations, which can weaken and simplify certain geometric features in the sonification results. From a sonification perspective, although we have restored the instruments used in *Metastaseis*, applying the same instrumental system universally cannot fully express the distinctive features of other architectural styles or explore broad musical possibilities. From an interaction perspective, our sonification system visualized audiovisual elements but lacked user experience features common to interactive architectural systems.

In future work, we aim to explore further possibilities for architectural mapping strategies, extend the sonification system to more architectural cases, and enable the audiovisual representation of the *Philips Pavilion* to interact with users in immersive environments such as VR/AR. Therefore, users can hear the architectural sonification results as they navigate the space, deepening the experiential connection between architecture and music.

5. REFERENCES

[1] R. Howat, "Debussy, Ravel and Bartók: Towards some new concepts of form," *Music & Letters*, pp. 285–293, 1977.

[2] I. Xenakis, "Metastaseis: 1953/54," London, 1967, facsimile score.

[3] R. Wannamaker, "Mathematics and Design in the Music of Iannis Xenakis," *Xenakis Matters*, pp. 127–41, 2012.

[4] E. Sikiaridi, "The Architectures of Iannis Xenakis," *Technoetic Arts*, vol. 1, no. 3, pp. 201–207, 2003.

[5] I. Xenakis, C. Vreedenburgh, A. Bouma, F. Ligtenberg, and H. Duyster, "The Philips Pavilion at the 1958 Brussels World Fair," *Philips Technical Review*, vol. 20, no. 1, pp. 1–36, 1958.

[6] R. Hudson, "Strategies for parametric design in architecture," *Civ. Eng*, vol. 274, pp. 77–81, 2010.

[7] W. Jabi, *Parametric design for architecture*. Laurence King Publishing, 2013.

[8] O. Tessmann and A. Rossi, "Geometry as interface: parametric and combinatorial topological interlocking assemblies," *Journal of Applied Mechanics*, vol. 86, no. 11, p. 111002, 2019.

[9] B. N. Walker and M. A. Nees, "Theory of sonification," *The sonification handbook*, vol. 1, pp. 9–39, 2011.

[10] K. Speidel, ""Telling in Time" Extended: Temporality in the "Spatial" Arts; How Single Pictures Convey Stories Suspensefully," *Poetics Today*, vol. 41, no. 4, pp. 669–704, 2020.

[11] H. R. Brown, *Physical relativity: Space-time structure from a dynamical perspective*. Oxford University Press, 2005.

[12] J. E. Cutting, "Representing motion in a static image: constraints and parallels in art, science, and popular culture," *Perception*, vol. 31, no. 10, pp. 1165–1193, 2002.

[13] S. Giedion, *Space, time and architecture: the growth of a new tradition*. Harvard University Press, 2009.

[14] A. Heathcote, "Liberating sounds: philosophical perspectives on the music and writings of Helmut Lachenmann," Ph.D. dissertation, Durham University, 2003.

[15] S. Sterken, "Music as an art of space: interactions between music and architecture in the work of Iannis Xenakis," *Resonance: Essays on the intersection of music and architecture*, pp. 21–51, 2007.

[16] J. Clarke, "Iannis Xenakis and the Philips Pavilion," *The journal of architecture*, vol. 17, no. 2, pp. 213–229, 2012.

[17] N. Boulouch, "Séverine Bridoux-Michel, Le Corbusier, Iannis Xenakis: un dialogue architecture/musique," *Critique d'art. Actualité internationale de la littérature critique sur l'art contemporain*, 2020.

[18] M. Solomos, "Le Diatope et La Légende d'Eer de Iannis Xenakis," *Musique, instruments, machines. Autour des musiques électroacoustiques*, pp. p–161, 2006.

[19] P. Parthenios, S. Petrovski, N. Chatzopoulou, and K. Mania, "Reciprocal transformations between music and architecture as a real-time supporting mechanism in urban design," *International Journal of Architectural Computing*, vol. 14, no. 4, pp. 349–357, 2016.

[20] C. Torresan, G. Bernardes, E. Caetano, and T. Restivo, "The singing bridge: sonification of a stress-ribbon footbridge," in *International Conference on ArtsIT, Interactivity and Game Creation*. Springer, 2022, pp. 359–373.

[21] S. N. Krivoshapko, "Kinematic surfaces with congruent generatrix curves," *RUDN Journal of Engineering Research*, vol. 24, no. 2, pp. 166–176, 2023.

[22] G. Read, “Extending the tonal resources of wind instruments: Some contemporary techniques,” *Music Educators Journal*, vol. 63, no. 1, pp. 50–55, 1976.

[23] J. Hunter and D. Dale, “The matplotlib user’s guide,” *Matplotlib 0.90. 0 user’s guide*, vol. 487, 2007.

Cincinnati Center for Computer Music 2026

Hefang Ma
University of Cincinnati
mahf@mail.uc.edu

Jingyu Luo
University of Cincinnati
luoj5@mail.uc.edu

Paul Francis
University of Cincinati
francipc@mail.uc.edu

Mara Helmuth
University of Cincinnati
helmutmm@ucmail.uc.edu

Sangbong Nam
Seoul National University of Education /
University of Cincinanti
namsg@ucmail.uc.edu

Wei-Huai Chen
Unviersity of Cincinnati
chen2wu@mail.uc.edu

ABSTRACT

The Center for Computer Music continues to function as an active site for research, composition, performance, and pedagogy in computer music. During the current period, research at the Center has focused on algorithmic and AI-assisted composition systems, real-time sonification and data-driven sound synthesis, beat tracking and performance synchronization, and music programming environments developed by faculty and students. These projects emphasize structured workflows, probabilistic processes, real-time analysis, and their application within compositional and interactive performance contexts.

Creative output includes electroacoustic works, live electronics, spatialized sound, and multimedia performances presented at CCM as well as at national and international venues.

1. INTRODUCTION

Following the previous studio report presented at ICMC 2024 in Seoul, Korea [1], the Center for Computer Music at the College-Conservatory of Music, University of Cincinnati, has continued to demonstrate significant growth and impact in both artistic and research activities. High-level musical projects developed by students and faculty, together with advanced research initiatives, have supported newly joined participants while also reaching musicians and researchers beyond the institution.

During the current period, activities at the Center have emphasized technology-driven music creation and interdisciplinary artistic exploration. Research and creative work have addressed a range of topics spanning algorithmic composition, real-time sonification, interactive systems, and music programming, alongside performances and educational initiatives that integrate computation with compositional and performance practice. In addition, the Center is equipped with virtual reality (VR) hardware, enabling exploratory work in music and virtual environments [2].

Copyright: © 2025 Ma et al. This is an open-access article dis- tributed under the terms of the Creative Commons Attribution License 3.0 Unported, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

2. RESEARCH PROJECTS

2.1 AI-Assisted descriptor-driven system for Generative Orchestration

Hefang Ma has developed Numina, an AI-assisted Descriptor-Driven system designed to support composer-centered orchestral composition. Numina derives perceptual and structural tendencies from audio or score input and maps them into a multi-stage orchestration workflow that separates analysis, planning, and realization. The system integrates probabilistic orchestration procedures adapted from prob() function, incorporating bounded sampling, adjustable concentration, and explicit random seeding to support controlled variation and reproducibility. Numina generates multi-track orchestral MIDI output for a double-wind orchestra, enabling iterative exploration of orchestral structure and behavior across extended formal sections.

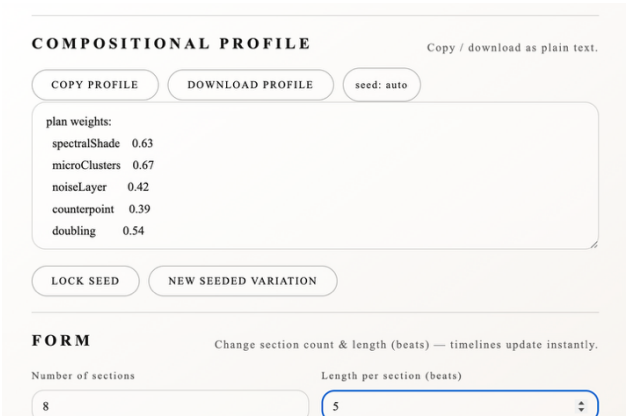


Figure 1. Numina browser-based interface showing the planning layer and compositional profile for composer-guided orchestral structure and probabilistic realization.

2.2 Real-Time Sonification and Interactive Data-Driven Sound System

Mara Helmuth has expanded ongoing research on pulsar sonification by extending previously published works, including From Orion to Cassiopeia and Letting Pulsars Sing: Sonification with Granular Synthesis [3] into a real-time interactive system. Building on methodologies presented in prior publications, this research developed Sonic Pulsings, an interactive application and musical work using Max 9 and rtcmix~, that generates sound directly from pulsar data through data-driven granular synthesis. The expansion of sonification of pulsar parameters includes more diverse granular sounds indicating pulsar types and galactic locations.

2.3 AI-Assisted Composition

Sangbong Nam's research focuses on case studies of AI-based music composition. This work contributed as an author to *Understanding AI Technologies for Artistic Creation: Concepts and Case Studies*, published by the Arts Council Korea, where he examined projects such as *Beethoven X: The AI Project* (2021), *Silicon for Orchestra and Artificial Intelligence* (2022), the *Ricercar Project* (2019–), as well as *OuchAI* and *Treatise AI Improvisation* (2023–25). These case studies explore compositional workflows, system design, and modes of human–AI collaboration in contemporary music practice.



Figure 2. Book cover image from Understanding AI Technologies for Artistic Creation: Concepts and Case Studies (Arts Council Korea,2025)

2.4 Real-Time Beat Tracking and Performance Synchronization

Wei-Huai Chen has adapted the BEAST (Beat Tracking Streaming Transformer) model from Chang and Su 2024 [4] for real-time beat tracking in live performance contexts by redesigning it as a fully block-wise streaming and forward-predicting inference system. This approach enables incremental audio processing from microphone input and supports zero-latency beat and tempo estimation for interactive musical applications.

To address the persistent structural latency inherent in the BEAST model, this work utilizes BEAST's activation function and then applies Madmom's [5] online Dynamic Bayesian Network (DBN) to identify beats within a short-term memory window (≈5 seconds). The key innovation is a predictive beat extrapolation strategy built upon these detected beats. The system first applies a sub-beat collapse procedure to obtain a cleaned, temporally consistent beat sequence. Using the most recent inter-beat interval, it then computes tempo-bounded prediction windows—corresponding to the current, next, and next-next beats—under an allowable ±40% tempo fluctuation. For each window, a forward projection curve is formed by multiplying a rectangular tempo-constraint mask with an exponential peak function centered at the expected beat location to simulate human prediction. These components are summed to create a continuous projection profile. This projection enables zero-latency beat estimation under real-time conditions.

2.5 Bell Language Music Programming

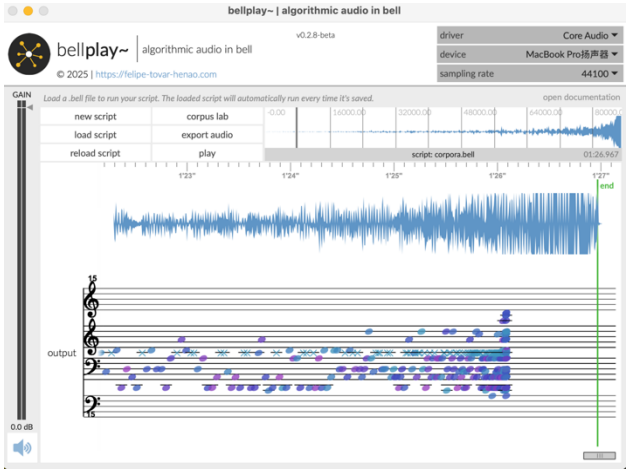


Figure 3. Displays the note distribution at the end of the second part on bellplay~.

Jingyu Luo has developed an electroacoustic composition, *Motif for* Ono, inspired by Yoko Ono's conceptual art and Foucauldian is course theory [6], using the bell programming language [7] and the bellplay~ environment. The work processes vocal material through techniques including audio slicing, FFT-based granular synthesis, pitch quantization, and corpus organization driven by database-based parameter control. The project also integrates interactive visual elements created in TouchDesigner, synchronized with audio analysis, and reflects an approach that combines electronic music programming with conceptual and interdisciplinary artistic frameworks.

3. PROGRAM/COURSES

3.1 Music Programming Course

The Music Programming course has supported many students in developing their own compositional languages and instruments. The course provides a rich music programming learning environment, including studies in RTcmix and related systems. Within this class, X developed a Max/MSP patch based on Short-Time Fourier Transform (STFT) analysis for spectral sound transformation, enabling frequency-domain analysis and spectral content processing.

3.2 Intro to Electronic Music Course

This course was offered during the current period and attracted students from multiple majors across the institution. The curriculum introduced foundational tools such as DAWs and VCV Rack, while also emphasizing the study of electronic music history and current trends in electronic music. For the final project, students developed works in musical genres of their choice, allowing them to explore individual interests and creative directions. Guest artists presented talks and shared their artistic practices with the class.

4. CONCERT AND PERFORMANCE

4.1 Sonic Explorations

Every fall and spring semester, the *Sonic Explorations* concert presents electronic works by students, faculty, alumni, and guests, with frequent emphasis on multichannel sound and works combining performers and electronics. The concert reflects the breadth of creative activity within the CCM computer music community, encompassing fixed media, live electronics, spatialized sound, and multimedia approaches, and highlighting ongoing experimentation and interdisciplinary practice. World premiere performances included *Him, in the City: Episode II* (2025) by Hsin-Yi Tseng, *Projection* (2025) by Brandon Wilhelmsen, *Heliopause* (2025) by Zach Graham, *Motif for Ono* (2025) by Jingyu Luo, and *Infinity* (2025) by Siyuan Kang. Additional works presented were *veronica* (2021) by Wen Wen Van Der Wende, along with faculty compositions *Awaken* (2010) and *Mugung: Between Stillness and Motion* (2025) by Sangbong Nam.

4.2 Recitals and Other Works

Mara Helmuth will present a faculty composition recital at CCM on January 27, 2026, featuring guest performers Patti Cudd, Esther Lamneck, and Andrea Vos-Rochefort, together with CCM cellist Benjamin Hughes.

Hefang Ma's new orchestral work *I–A*, developed using the Numina application, will be read by the CCM Philharmonia in February 2026. In addition, her doctoral recital in February 2026 will include two electronic works, *Ex* and *V*.

Motif for Ono by Jingyu Luo will perform at the 2025 CCF China Computational Art Conference; MOXsonic 2026, and Electronic Music Festival 2026. *Sheltered* by Wenwen Van Der Wende is scheduled for performance on March 1, 2026, at the University of Massachusetts Amherst, performed by Taylor Hestvik.

Felipe Tovar-Henao presented several recent works during this period, including ludus vocalis (2024–25) for fixed multimedia and Machina Rasa (2025) for amplified trio, commissioned by the New Downbeat Collective. These works were presented at venues and events including SEAMUS, the Performing Media Festival, CCM Moveable Feast, and other national and international concerts.

5. AWARDS AND RECOGNITION

In 2025, Mara Helmuth was awarded a Guggenheim Fellowship for a project involving algorithmic composition for instruments and computer.

Hefang Ma's orchestral and electronic work *I* received an Honorable Mention at the Vienna New Year's Concert International Competition. Jingyu Luo's *Motif for Ono* advanced to the final round in the Music Technology Design Category of the International Electronic Music Competition.

6. BEYOND THE CENTER

Mara Helmuth was in residence during her 2024–2025 sabbatical at the Virginia Center for the Creative Arts in Mt. St. Angelo, Virginia, and at Wassard Elea in Ascea, Italy. She also undertook a residency with Esther Lamneck (tárogató and real-time composition), during which they recorded source samples and video, researched controllers, and composed new works. Additional field recordings were conducted in Italy and Paris. One outcome of this period was *Wind Blown Rain*, by Helmuth and Lamneck, with video by Alfonso Belfiore.
Helmuth presented performances at the Society for Electro-Acoustic Music in the United States (SEAMUS), ICMC 2024 and 2025, the Japan Society for Sonic Arts Meeting 2024, and the New York City Electroacoustic Music Festival 2024 and 2025.

6.1 Visiting Scholar

Alumnus Professor Sangbong Nam (Seoul National University of Education) is visiting the Center during the current period. During his visit, he participated in research exchange with faculty and students and presented his work as part of the Sonic Explorations concert series and the Composition Symposium. Additionally, he presented a talk 'Musical Authorship in the Age of AI' at the AI and Emerging Technology Symposium at the University of Cincinnati in Feb., 2026, contributing to ongoing dialogue within the Center's research and performance activities

7. CONCLUSIONS

The activities of the Cincinnati Center for Computer Music during this period encompassed a wide range of performances, research projects, and educational initiatives involving faculty and students. Ongoing research in AI-assisted orchestration, real-time sonification, beat tracking, and music programming resulted in new systems, compositions, and performance practices. Concerts and presentations at CCM and international venues highlighted continued experimentation in electroacoustic music, multimedia integration, and interactive systems. Educational programs supported the development of technical and creative skills across multiple disciplines.
In addition, Helmuth contributed to international scholarly exchange through invited presentations, including a virtual talk at the Symposium on Archiving and Re-Performing Electroacoustic Music (AREM) at Freie Universität Berlin in November 2024, addressing sustainability and long-term performability in interactive computer music [10]. Visiting artists and scholars further enriched the Center's activities, supporting dialogue across institutions and international contexts.
Together, these activities reflect the Center's continued engagement with contemporary computer music research, composition, and performance on both local and international levels.

8. REFERENCES

[1] K. McAuliffe, F. Tovar-Henao, M. Helmuth, Y. Mu, C. Jacobson, and H. Ma, "Cincinnati Center for Computer Music 2024," Proceedings of the 2024 International Computer Music Conference, Seoul, Korea, 2024.

[2] M. Helmuth, Y. Mu, K. McAuliffe, G. Choi, C. Jacobson, A. McFarlane, and W. Lyu, "Cincinnati Computer Music 2022," Proceedings of the 2022 International Computer Music Conference, Limerick, Ireland, 2022, pp. 145–148.

[3] John Simon Guggenheim Memorial Foundation. (2025). Fellows. Retrieved from https://www.gf.org

[4] M. Helmuth, "A composer's view on sustaining interactive music," invited virtual talk, AREM Symposium on Archiving and Re-Performing Electroacoustic Music, Freie Universität Berlin, Germany, Nov. 2024.

[5] M. Helmuth and E. Lamneck, "Composer's note: Music for tarogato, computer and visuals: Burren Wind," Sonic Arts Today, vol. 16, no. 3, pp. 46–50, 2024.

[6] C.-C. Chang and L. Su, "BEAST: Online joint beat and downbeat tracking based on streaming transformer," arXiv preprint arXiv:2312.17156v3, 2024.

[7] D. Mirka, Metric Manipulations in Haydn and Mozart: Chamber Music for Strings, 1787–1791. Oxford University Press, 2009.

[8] P. Vuust, O. A. Heggli, K. J. Friston, and M. L. Kringelbach, "Music in the brain," Nature Reviews Neuroscience, vol. 23, pp. 287–305, 2022.

[9] S. Böck, F. Korzeniowski, J. Schlüter, F. Krebs, and G. Widmer, "madmom: A new Python audio and music signal processing library," Proceedings of the 24th ACM International Conference on Multimedia, pp. 1174–1178, 2016.

[10] M. Heydari and Z. Duan, "Don't look back: An online beat tracking method using RNN and enhanced particle filtering," IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2021.

[11] S. Mills, Discourse. Routledge, 1997.

[12] A. Agostini and J.-L. Giavitto, "bell, a textual language for the bach library," 2018. Retrieved from http://repmus.ircam.fr/_media/giavitto/2019_agostini_giavitto_bell.pdf

De/Re:Generation: Exploring DIY Cassava-Starch Bioplastic Interfaces with EFT-Based Touch Sensing in an Interactive Sound Installation

Adriano Claro Monteiro
Federal University of Goiás
adrianomonteiro@ufg.org

Rafaela Blanch Pires
Federal University of Goiás
rafaela.pires@ufg.com

ABSTRACT

This paper reports the design, implementation, and preliminary validation of a tangible interface that combines electrical field tomography (EFT), support vector machines (SVM), and cassava-starch bioplastics for an interactive sound installation. The system addresses two main challenges: 1) creating low-cost, large-area touch surfaces with flexible geometries, and 2) integrating biodegradable materials into electronic interfaces while preserving sufficient electrical and mechanical stability for real-time performance. It consists of bioplastic interfaces, custom hardware for multiplexed current injection and voltage measurement, and a software pipeline for signal conditioning and SVM-based touch classification. Results show that the system can reliably distinguish a vocabulary of touch gestures on irregular bioplastic objects, while also revealing limitations related to long-term stability, calibration, and object-specific training. Finally, the paper discusses its integration into De/Re:Generation, a sound installation where bioplastic sculptures operate both as scenographic elements and as interactive surfaces within a vibro-acoustic environment inspired by the cicada life cycle.

1. INTRODUCTION

Electrical Impedance Tomography (EIT) and Electrical Resistivity Tomography (ERT) are tomographic techniques that reconstruct the internal distribution of an object's electrical impedance or conductivity (resistivity) from boundary current injections and voltage measurements. While EIT relies on alternating current (AC) excitation, ERT uses direct current (DC) or low frequency AC in scenarios where reactance is negligible [1]. In Electrical Field Tomography (EFT), related principles are used, but the emphasis lies on measuring distortions in an externally applied electric field [2]. Primarily used in applications such as industrial monitoring, medical or geophysical imaging [1], these techniques have recently been repurposed for tactile sensing in human–computer interaction (HCI) [2–12].

In parallel, a growing body of artistic and design practices has explored biomaterials for performative and scenographic ends, particularly in contexts that foreground ecological perspectives [13,14]. Cassava-starch bioplastics, in particular, can be fabricated with low-cost, do-it-yourself (DIY) methods, and have been used in costumes, installations, and performative objects that dramatize processes of degradation and regeneration [15, 16].

The research presented here intersects these fields by investigating cassava-starch bioplastics turned into EFT-based touch interfaces for interactive sound environments as part of the broader artistic research project De/Re:Generation.

At the core of our artistic project lies a poetics which thematizes the cicada, whose life cycle oscillates between long, silent periods underground and brief, intense eruptions of sound and emergence. The work draws on this alternation between invisibility and hyper-presence to think about degeneration-regeneration cycles and auditive perspective of non-human life. The bioplastic interfaces echo cicada exuviae: fragile, hollow envelopes that simultaneously signal absence, transformation and continuation. Coupled with vibrating surfaces and low-frequency sound fields, these forms invite the audience to inhabit both the subterranean and aerial dimensions of the cicada's world, blurring boundaries between body and environment, tactile an aural modality of sound perception, inorganic and organic technological apparatus. In this sense, the interface does not simply control the installation, as a metaphor for the cicada's skin, it occupies a central role in shaping a vibrational-tactile mode of environmental perception and communication.

An earlier publication [17] addresses the development of this artistic research and discusses its sonic and aesthetic goals, while the present paper focuses on the technical details of the bioplastic interface and its applications.

The paper is structured as follows: subitems of section 1 introduce EIT-based and Bio-HCI interfaces, and cassava-starch bioplastic. Section 2 details the design and implementation of the proposed system: bioplastic interface, electronic hardware and software. Section 3 presents a preliminary object-and-task-specific validation. Section 4 discusses the system's applications in artistic works.

Copyright: © 2026 Adriano C. Monteiro and Rafaela B. Pires This is an open-access article distributed under the terms of the Creative Commons Attribution License 3.0 Unported, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

1.1 EIT/EFT- Based Tactile interfaces

Compared to conventional array-based tactile sensors, such as resistive or capacitive grids, EIT approaches (and EIT-related methods) trade spatial resolution for material and geometric flexibility [3–6]: a small number of boundary electrodes can capture distributed changes in conductivity caused by touch, pressure, or deformation without requiring a dense internal wiring matrix. However, they typically exhibit lower spatial and temporal resolution or limited ability to distinguish fine-grained pressure intensities compared to high-density arrays [3, 6]. Several electrode drive patterns (injection and measurement schemes) have been proposed to improve spatial and temporal resolution [18], for instance, optimized stimulation patterns to reduce the number of current injections [7], or adaptive measurement strategies to focus acquisition on regions of interest [8].

One of the pioneering works on EIT-based tactile sensing employed pressure-sensitive conductive rubber as the sensing medium [9]. Subsequent research on sensing material has, for example, explored conductive textiles to improve elasticity and reduce hysteresis [10], elastomer matrices doped with carbon nanofillers [11], and porous polymer foams infused with ionic liquids [12]. These approaches depend on high-cost industrial materials, as well as highly specialized facilities and technical expertise, which makes them impractical for more DIY-based artistic practice.

Electrick, an EFT-based system proposed by Zhang et al. [2] has demonstrated how everyday objects and surfaces can be transformed into touch interfaces by painting or laminating them with conductive coatings and using a small number of electrodes to measure field perturbations. While this system employs synthetic, industrially produced conductive materials, it suggests a more general design space in which flexible conductive substrates can envelop arbitrary geometries.

Our work builds upon these insights by investigating whether DIY cassava-starch bioplastics can serve as EFT sensing media and how their material specificities impact tactile sensing and interaction design.

1.2 Bio-HCI and DIY cassava-starch bioplastics

Recent HCI research has increasingly explored bioplastics and other biomaterials as substrates for tangible and embedded interfaces [19–21], e-wearables and textiles [22], contributing to an emerging area often referred to as Biological HCI (Bio-HCI). Environmental concerns strongly motivate such initiatives, as mainstream HCI making and prototyping still heavily rely on plastics produced from fossil fuel, metals, and other non-renewable materials with poor end-of-life degradability [23, 24]. Biomaterials have been created from living or once-living organisms including plants (e.g. starches, agar, cellulose), animals (e.g. gelatin), bacteria, fungi (e.g. slime molds, mycelium) and microbial cellulose, and have been processed in several forms, such as plastics, foams, dyes, pastes, leathers and clays [19–21, 25, 26].

In industrial contexts, bioplastics often incorporate additives that stabilize mechanical and barrier properties for market applications; however, these additives can reduce biodegradability and perpetuate some of the environmental issues associated with conventional plastics.

In this research, we do not focus on industrially produced bioplastics, but rather on simple, homemade recipes [26] using ingredients easily found in supermarkets. DIY approach affords artists and designers direct control over material composition and aesthetic qualities. It also enables small-scale and low-cost production.

When heated, cassava starch granules undergo gelatinization and polymerization, and after five to seven days of natural drying, the material forms a plastic-like network. Cassava-starch bioplastics exhibit distinctive behaviors: they shrink and warp as they dry, respond strongly to ambient humidity, and can easily crack, crumble, or mold over time [15, 16]. From an ordinary engineering, industrial, or consumer perspective, such instabilities may be seen as limitations. However, in artistic contexts they become performative features that dramatize temporal processes of degradation, fragility, and regeneration. Previous works have used cassava bioplastics for costumes and sculptural pieces that intentionally degrade, releasing seeds or transforming over time [15, 16].

For the De/Re:Generation project, these bioplastics were cast into rounded sculptures that visually echo cicada nymphs and exuviae through their color, translucency, and texture, operating simultaneously as scenographic elements and sensing surfaces [17]. Like cicada shells, they easily crack, crumble, and decompose over time, supporting a poetics of cyclical life, decomposition and regeneration, that resonates with theoretical frameworks that foreground composting as material and semiotic practices [27].

2. SENSOR SYSTEM DESIGN

The main goal of our system is to transform sculptural bioplastic objects into tangible controls for sound installations, without embedding rigid sensor grids in their surface. Instead, a small number of metallic electrodes are attached along the periphery or the backside of each object, and EFT measurements are used to infer where and how the object is being touched.

2.1 Bioplastic Interfaces

Two types of bioplastic interfaces were used. The first consisted of small, thin and flexible circular samples (18cm diameter) with approximately homogeneous material composition and thickness (~1mm), designed for testing the electronic system. The second, developed as part of our artistic work, consisted of hemispherical sculptural shells made from bioplastic and sisal fibers that varied in size and exhibited heterogeneous distributions of material composition and thickness across their surface (typically 1–5 mm, with local blobs reaching up to 10 mm).

Bioplastics were fabricated by heating aqueous mixtures with a weight ratio of 1 part cassava starch, 2.5 parts tap water, 0.25 (1/4) parts glycerin, 0.15 (3/20) parts white vinegar until gelatinization, then casting the hot gel into molds lined with thin plastic film. A brownish coloration was obtained by substituting tap water with liquid

coffee in the same proportion. Once demolded and dried, the pieces exhibited varying degrees of coloration and translucency and, in the case of the bioplastic shells, thickness and warping. The material can last for years if properly preserved, although it may deform due to variations in air humidity and temperature. When placed in a compost bin, it degrades in approximately one week.

Related works have incorporated conductive additives such as metals or carbon-based composites (e.g. graphite, activated charcoal and graphene) into biomaterials to increase conductivity [19–21, 28, 29]. These additions also alter aesthetic properties that are critical for our artistic applications, notably color, translucency, and stiffness. Interestingly, our basic bioplastic recipe described above exhibited low but sufficient conductivity (on the order of a few megaohms), enabling reasonable classification of touch patterns without compromising the desired material aesthetics.

Eight metallic electrodes were attached to the bioplastic interfaces for EFT-sensing. For the circular samples, alligator clips connected to flexible wires were distributed along their perimeter (approximately 45° apart from one another), with tips pointing toward the center of the disc (see Figure 1).

For the bioplastic shells, handmade metallic electrodes, made from aluminum wire and shaped as spirals (approx. 15mm in diameter), were attached to the inner surface of the hemisphere using a thin layer of fresh bioplastic applied over each electrode. The electrodes were positioned to maximize coverage of the shell's area while keeping cables hidden from the audience's main line of sight, and were connected via flexible wires to the analog front-end circuitry. Two rings of four electrodes were positioned along the height of the hemisphere, approximately at 30° and 60° elevation angle and azimuthally rotated 45° relative to one another, so as to distribute the electrodes more evenly and to minimize regions of reduced sensitivity caused by larger spans of bioplastic between electrodes.



Figure 1. On the left, the large bioplastic shell interface; on the right, the small circular sample interface. Objects' relative sizes are not preserved.

2.2 Electronic Hardware Architecture

Our hardware architecture is implemented on a custom PCB (Figure 2) and acts as a shield for a NodeMCU module hosting an ESP32 microcontroller (Espressif Systems). The design goal was to provide a simple and low-cost EIT/ERT/EFT front-end that could be easily integrated into the bioplastic sculptures while minimizing cabling and external components. The ESP32 proved to be a suitable low-cost choice, providing numerous GPIO pins, 12-bit analog-to-digital converters (ADC), 8-bit digital-to-analog converters (DACs), built-in Wi-Fi and Bluetooth connectivity.

For current injections we adopted a Modified Howland Current Source (MHCS) as implemented by Creegan et al. [30]. The MHCS is built around an MCP6022 IC (Microchip Technology) containing two operational amplifiers. It is driven by a voltage signal provided by one of the ESP32's DACs ports which is stabilized trough a voltage buffer using a second MCP6022. The buffered signal is then converted into an output current by the MHCS circuit and routed to the selected electrode pair.

Voltage measurement between electrodes is performed by an INA128 (Texas Instruments) instrumentation amplifier configured in differential mode. INA128's output is routed to one of the ESP32's ADC channels, which digitalizes the amplified potential difference for each measurement configuration.

Both MHCS and INA128 are implemented along with THP resistor connectors allowing the replacement of key resistors. In MHCS, a load resistor tunes the output current to the impedance range imposed by the bioplastic touch interface. In the INA128 stage, the gain-setting resistor adjusts the amplification factor so that the resulting signal fits the dynamic range of the ESP32's ADC.

Routing of current injection and measurement paths to the electrodes is handled by four CD74HC4051E, eight channel analog multiplexer/demultiplexer ICs. This configuration supports up to eight electrodes attached to a given bioplastic object and enables flexible selection of drive and sense electrodes under firmware control. Together, the NodeMCU-ESP32, MHCS, INA128 and multiplexers form a compact EIT/ERT/EFT measurement module that can be embedded in or near the sculptural interfaces and communicates wirelessly with the host computer as described in Section 2.3.

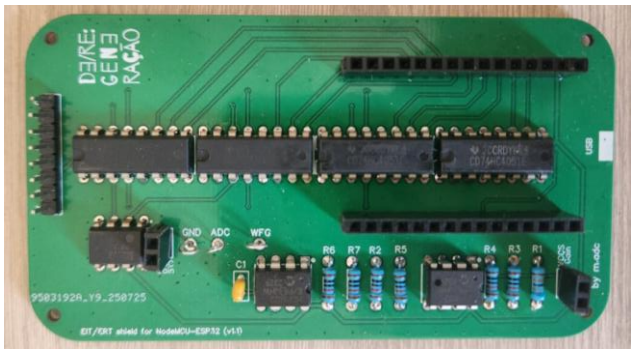


Figure 2. Customized NodeMCU-ESP32 shield implementing the EIT/ERT/EFT circuit described here.

2.3 Software Pipeline and SVM Classification

The software associated with the sensor is split between ESP32 firmware and a Max/MSP patch running on a computer. The ESP32 program is responsible for current injections, electrode switching, and data acquisition. On the host side, Max/MSP handles data reception, signal conditioning, SVM training and real-time touch pattern classification.

2.3.1 Data Acquisition and Conditioning

Among the various current injection and measurement schemes described in the EIT literature we adopted an approach called "opposite method", which injects current between a pair of diametrically opposite electrodes in the electrode array [18]. This configuration improves the resolution of interference patterns in the central region of the electrode ring, which is particularly relevant for touch gestures over the central area of the bioplastic object. For a configuration with 8 electrodes, this method yields a feature vector of 40 differential voltage measurements per full scan cycle.

In our implementation, an electric field is established in the material by injecting a DC excitation generated by one of the ESP32's DAC channels at 3.3V. This voltage drives the MHCS, which produces a constant output current whose magnitude is tuned according to the impedance of the bioplastic interface. Because geometrical and material properties of each cassava-starch bioplastic object differ significantly from one piece to another, a MHCS load resistor is chosen on a per-object basis through empirical tests, ensuring an appropriate current range for each interface.

To reduce electrodes switching artifacts, we introduce small delays in the firmware when stepping through the measurement configurations. A delay of approximately 100µs is inserted after each change of the injection electrode pair to allow the current to settle, and a shorter delay of about 2 µs is used when switching the measurement electrode pair to avoid artifacts caused by the analog multiplexers.

The resulting measurement vectors are transmitted wirelessly to the host computer via Wi-Fi using the UDP protocol. In Max/MSP, incoming data packets are passed through a moving-average filter and then baseline-corrected. A baseline vector is obtained by sampling moving-averaging vectors acquired in a known no-touch condition and subsequent difference vectors are obtained by subtracting this baseline. The resulting baseline-corrected vectors are used both for SVM training and for real-time classification.

2.3.2 SVM-Based Gesture Classification

Following the strategy adopted by Zhang et al. [2] for image-free EIT/EFT interaction, we rely on support vector machines (SVM) to learn and classify patterns of interference in the electric field produced by different touch patterns on the bioplastic interface, bypassing the tomographic image reconstruction step. For this purpose, we use an SVM implementation provided by the ml.lib, a machine learning library for Max/MSP [31].

Training parameters are adjusted empirically for each bioplastic object, reflecting variations in material properties and electrode layout. For each distinct touch gesture, we typically record around 40 consecutive measurement vectors, grouped into 3 to 5 slightly different realizations of the same gesture (e.g., small variations in exact finger/hand position). These labeled vectors form the training set for the SVM. For the artistic applications described in Section 4, the SVM were configured with an RBF kernel, $\gamma = 0.025$, and $C = 1$, however, object-specific calibration proved necessary to achieve robust classification when facing differences in the objects' geometrical and material properties, as well as under non-stationary conditions caused by material degradation, humidity, or minor mechanical damage.

After training, the classifier operates in real time on incoming feature vectors. To reduce computational cost, we downsampled the feature-vector rate to 20ms. To mitigate spurious misclassifications (typically brief output fluctuations following state transitions) we applied a simple temporal smoothing stage: a percentage filter defined by a sliding window over the most recent classification outputs computes the proportion of each class, and an output is accepted as a valid event only if its class exceeds a predefined percentage threshold *P* within the window.

3. SYSTEM VALIDATION

Given the heterogeneous, non-standardized, and intrinsically unstable nature of DIY cassava-starch bioplastic objects, we adopted a validation strategy for the proposed EFT–SVM interface that privileges qualitative assessment of artistic suitability, using quantitative measures only as object-and-task-specific indicators rather than as the basis for comprehensive, standardized benchmarking. We therefore concentrated on exploratory experiments that probed the discriminative capacity of the system and its behavior over time using an object design and touch gesture vocabulary aligned with our artistic proposal.

3.1 Experimental setup

We tested both interface formats described in Section 2.1: 1) a small circular sample (18 cm in diameter) with approximately homogeneous thickness (~1 mm) and composition; and 2) a large heterogeneous hemispherical shell, approximately 55 cm in diameter. Results from the circular interface served as proof of concept and baseline for comparison with tests on the more complex shell interface conceived to our artistic work. Tests were conducted in an environment with temperature ranging from 25°C to 30°C and relative humidity from 60% to 75%. For the circular interface, training data were obtained using one and two-fingertip touches, whereas for the hemispherical interface they were obtained using three-fingertip touches.

For the circular samples, four experimental setups were conducted

Setup A (10 classes): **peripheral single-point touches**. Classes included: no-touch; eight outer points evenly spaced on the circumference between adjacent electrodes (approx. 45° apart); and one central point.

Setup B (18 classes): single-point touches at two radii. Classes included: no-touch; eight outer points evenly spaced on the circumference between adjacent electrodes; eight inner points located at the mid-radius on the same radial lines; and one central point.

Setup C (18 classes): higher-resolution perimeter. Classes included: no-touch; sixteen outer points evenly distributed along the circumference, alternating between

positions aligned with the electrodes and between adjacent electrodes; and one central point.

Setup D (10 classes): **multi-finger gestures**. Classes included: ten multi-finger touch patterns combining different regions and numbers of fingers on the disc.

On the hemispherical shell, we adapted Setups A and B (named A.1 and B.1) in which the outer and inner points were defined at the elevations of the lower and upper electrode rings (approximately 30° and 60°), and the central point was located at the apex of the hemisphere. We added four experimental setups to test higher touch position resolutions, multitouch identification and SVM capacity for dimensional generalization:

Setup E (34 classes): higher-resolution, single-point touches at two radii. Classes included: no-touch; sixteen lower points evenly spaced around the hemisphere at approximately 30° elevation; sixteen intermediate points evenly spaced around the hemisphere at approximately 60° elevation; and one top point (90° elevation).

Setup F (22 classes): higher elevation resolution, single-point touches. Classes included no-touch; twenty points defined by five elevation levels (near outer edge, 30°, 45°, 60°, 75°) and four evenly spaced azimuthal angles (0°, 90°, 180°, 270°); and one top point (90° elevation).

Setup G (20 classes): multitouch. Classes included no-touch; ten four-finger touch gestures (two hands allowed), with finger positions defined by the locations in Setup B.1; and five three-finger and five two-finger gestures formed as subsets of the four-finger gestures.

Setup H (34 classes): dimensionally independent classification. Two independent SVMs were used, one for azimuth (eight evenly spaced azimuthal angles around the hemisphere) and one for elevation (30°, 45°, 60°, 75°). Both classifiers also included no-touch and the top-point class**.**

After training, performance tests were repeated at 15-minute intervals over one hour, with no baseline correction, to evaluate temporal drift under continuous operation. The SVM training data (classes and associated feature vectors) from setups A.1, B.1, E, F and G were saved and used for intermittent performance tests over the subsequent 48 hours, with baseline correction applied, to evaluate the system/interface sensitivity to mid-term changes in environmental conditions.

3.2 Results and Discussion

Across all experiments with the circular sample, the system reliably discriminated the tested positions. Errors manifested mostly as rapid class fluctuations lasting a few milliseconds after some state transitions; these were suppressed by using a threshold of ≥60% for the percentage filter added to the system output. In Setup D, no relevant classification errors were observed.

For Setups A–C, the dominant (although sporadic) misclassification occurred between central touch and no-touch, reflecting the relatively smaller field changes produced by contacts at the center compared with the more pronounced perturbations caused by peripheral touches. In Setup B, occasional misclassifications were observed between inner and outer points on the same radial line, whereas in Setup C they tended to occur between adjacent peripheral locations. In both cases, these errors were most likely caused by partially overlapping field perturbations from nearby touch positions, given the relatively large finger contact area compared to the small bioplastic surface.

For tests with the larger interface, A.1 and B.1 achieved 100% correct classification with a percentage filter threshold (P) of ≥50%. Notably, single-finger touches were sufficient for correct classifications. Lower thresholds produced brief post-transition fluctuations, typically between certain touched positions and the no-touch state, and, in B.1, between adjacent elevation levels. In Setups E, F and H transient state fluctuations persisted even when higher thresholds were used. Considering only touch classes, Setup E showed transient fluctuations in ~36% (12/33) of classes with P = 8, and ~40% (13/33) with P = 6; Setup F showed ~20% (4/21) with P = 8, and ~33% (7/21) with P = 6; and Setup H showed ~27% (9/33) with P = 8, and ~48% (16/33) with P = 6. As in Setups B and C, most fluctuations occurred between adjacent azimuthal positions in Setup E and adjacent elevation levels in Setup F, suggesting overlapping field perturbations in higher resolution mappings.

For the multitouch test (Setup G), when trained with uncorrelated touch patterns, the classifier achieved 100% accuracy. When trained on subset-related patterns, misclassifications occurred between four-finger and three-finger gestures: 3/5 (60%) of three-finger touches were classified as their four-finger superset, and 1/10 (10%) of four-finger touches were classified as their three-finger subset. All two-finger subsets were correctly discriminated from their superset, and vice versa.

In all experimental setups classification results remained stable over one hour of continuous operation. For mid-term tests with the larger interface, single-touch, lower resolution mappings (A.1 and B.1) kept stable, whereas higher resolution (E, F) and multitouch (G) mappings degraded. In E and F, elevation information collapsed as misclassifications occurred between classes sharing azimuthal positions. In setup G, part of the multitouch patterns ceased to be recognized.

Overall, our results showed achievable touch-position detection resolutions, in the geometries investigated, on the order of two to three times the electrode count per dimension. Setups with moderate class counts and coarser spatial distinctions (such as A, B, C, D, A.1 and B.1) remained robust, whereas attempts at finer-grained single-point location discrimination (as in E, F and H) presented less reliable results. Over short-time spans, classification instability was mostly observed during brief post-transition intervals and could be mitigated by temporal filtering (at the expense of real-time responsiveness) or potentially reduced through finer parameter tuning and improved training procedures. Regarding mid-term performance, the persistence of lower resolution mappings is particularly relevant to our artistic context, in which sculptural interfaces were installed and reactivated multiple times over the course of an exhibition (see Section 4). Moreover, results from Setup H suggest that an SVM can generalize azimuthal or elevation information regardless

of its dimensional counterpart, indicating the possibility of parametrically independent, continuous gesture mappings by applying regression methods, as proposed by Zhang et al. [2].

Comparing the performance of the two interfaces, and within the sensitivity of our experiments, the inclusion of sisal fibers in the final shell sculptures did not appear to significantly affect the system's response. Instead, the dominant factors were the electrode count per bioplastic area, and touch-position resolution.

These results indicate that the EFT–SVM approach, even when applied to unstable and low conductive bioplastic materials, can support a range of touch vocabularies without requiring tomographic image reconstruction. At the same time, they highlight trade-offs among mapping resolution, object size, electrode count and material drift. From an artistic perspective, the system's qualitative performance and its capacity to accommodate material variability are sufficient to enable expressive use.

4. ARTISTIC APPLICATIONS

4.1 First prototype

In its first artistic application, the EFT–SVM–bioplastic interface was deployed as a prototype sound installation during a residency by the authors at Ligeti Zentrum in Hamburg, Germany, serving as a proof of concept during the technical stage of the research.

A suspended sculpture composed of two bioplastic shells equipped with 16 electrodes (8 each) functioned as the main tangible controller for the initial prototype test. Five touch patterns combining both shells were trained and mapped to parameters controlling a real-time synthesized soundscape based on physical models of cicada calls [32] and other insects [33], diffused over a 30-loudspeaker immersive system.

Touch patterns on the bioplastic sculpture triggered changes in sound source presence, spatial trajectories, and synthesis parameters, allowing visitors to "stir" the sound field from more realistic to more abstract soundscapes.

As can be seen in Figure 3, the first prototype was assembled and hand-soldered on a phenolic perfboard. The implemented circuit followed the same design described in Section 2.2, except that we used four 16-channel multiplexer/demultiplexer modules containing CD74HC4067 ICs, providing a 16-electrode EFT system whose electrodes were evenly distributed between the two bioplastic shells in the sculpture.

4.2 De/Re:Generation installation

The final artistic work of the De/Re:Generation project was premiered at Centro Cultural UFG in Goiânia, Brazil (Figure 4). For this work, the interface was scaled up to the dimensions described in Section 3 and integrated into a more complex installation that combines: multiple bioplastic sculptures, including two shells used as interfaces; an array of 48 vibration motors coupled with spiral-shaped exciters and diverse resonant surfaces (glass, wood, metal, and acrylic), forming a wall-mounted vibro-acoustic field (see Figure 5); and subwoofers reproducing low-frequency sine waves to create beating patterns. A custom PCB was designed as a 16-motor shield for NodeMCU-ESP32 modules, enabling algorithmic control of the vibration motors via Wi-Fi. The work was presented both as an interactive installation and as an instrumental environment for site-specific music and scenic performances.



Figure 3. First prototype of the bioplastic sculptural interface presented at the Ligeti Center in Hamburg (February 2025) as part of an immersive interactive sound system.



Figure 4. Spectators interacting with the bioplastic interfaces mounted on white plinths in installation presented at Centro Cultural UFG, Goiânia, October 2025.

A Max/MSP patch was created to algorithmically control the 48-motor array and generate rhythmic, spatial, and dynamic vibro-acoustic patterns. When performed as an instrumental environment, the overall system was controlled using a computer connected to the ESP32 motor shields. When presented as an installation, visitors were invited to handle the bioplastic sculptures, discovering how touch patterns modulate the vibrating walls.

Two bioplastic interfaces were available for public interaction, each mapped to five single-hand touch positions: the first triggered variations of a set spatial–rhythmic patterns manifested in the motor array (random pointillism plus cicada-like rhythms; groups with regular

pulses; rotating patterns; sequence repetition; and steady sound masses), and the second selecting the resonant surfaces associated with these patterns, consequently changing their timbre. Low-resolution touch mappings were chosen to facilitate exploratory user interaction and to improve robustness (see Section 3) during the one-month exhibition period.



Figure 5. Detail of one of the installation walls, showing several modules of vibrating motors and resonant surfaces using different materials.

5. CONCLUSION AND FUTURE WORK

This paper presented the design, preliminary object-specific validation, and application of a tangible interface that combines EFT, SVM-based classification, and cassava-starch bioplastics for an interactive sound installation. The work demonstrates that the proposed EFT–SVM system can employ DIY biomaterials, such as cassava bioplastics, as viable sensing media, despite strong material instabilities and heterogeneity. These characteristics require a mode of use that engages with transience, customization and object-specific adjustments, which, in the case of an artistic application, can directly shape the formal and poetic dimensions of the work.

Future work includes further developments across all dimensions of the project: bioplastic composition, fabrication and design; more sophisticated hardware circuitry for EFT/EIT/ERT; extended use of machine learning techniques, including methods for capturing continuous touch gestures; applications in new artistic contexts considering system designs whose technical features are informed by associated artistic processes.

Acknowledgments

We acknowledge the School of Music, the Institute of Scenic Arts, and the MediaLab of the Federal University of Goiás for hosting this research; the Ibermúsicas program and Funarte for financial support; the Ligeti Center for receiving us as visiting researchers during part of this project; and the Cultural Center of UFG for hosting the *De/Re:Generation* exhibition and performances.

6. REFERENCES

[1] A. Adler, D. Holder, “Electrical Impedance Tomography: Methods, History and Applications” (2nd Ed.) CRC Press, 2023.

[2] Y. Zhang, G. Laput, and C. Harrison, “Electrick: Low-Cost Touch Sensing Using Electric Field Tomography,” *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI ’17)*, Denver, CO, USA, 2017, pp. 1–14.

[3] H. Chen, X. Yang, P. Wang, J. Geng, G. Ma, and X. Wang, “A large-area flexible tactile sensor for multi-touch and force detection using electrical impedance tomography,” *IEEE Sensors Journal*, vol. 22, no. 7, pp. 7119–7129, 2022.

[4] S. Russo, R. Assaf, N. Carbonaro and A. Tognetti, "Touch Position Detection in Electrical Tomography Tactile Sensors Through Quadratic Classifier," in *IEEE Sensors Journal*, vol. 19, no. 2, pp. 474-483, 15 Jan.15, 2019.

[5] D. Silvera-Tawil, D. Rye, M. Soleimani and M. Velonaki, "Electrical Impedance Tomography for Artificial Sensitive Robotic Skin: A Review," in *IEEE Sensors Journal*, vol. 15, no. 4, pp. 2001-2016, April 2015 .

[6] F. Visentin, P. Fiorini, and K. Suzuki, “A Deformable Smart Skin for Continuous Sensing Based on Electrical Impedance Tomography”. Sensors, v.16, n.1928, 2016.

[7] H. Lee, K. Park, J. Kim, and K. J. Kuchenbecker, “Piezoresistive textile layer and distributed electrode structure for soft whole-body tactile skin,” *Smart Materials and Structures*, vol. 30, no. 8, p. 085036, Jul. 2021.

[8] K. Park, H. Lee, K. J. Kuchenbecker and J. Kim, "Adaptive Optimal Measurement Algorithm for ERT-Based Large-Area Tactile Sensors," in *IEEE/ASME Transactions on Mechatronics*, vol. 27, no. 1, pp. 304-314, Feb. 2022.

[9] Y. Kato, T. Mukai, T. Hayakawa and T. Shibata, "Tactile Sensor without Wire and Sensing Element in the Tactile Region Based on EIT Method," *SENSORS, 2007 IEEE*, Atlanta, GA, USA, pp. 792-795, 2007.

[10] A. Yao, C. L. Yang, J. K. Seo, and M. Soleimani, “EIT-Based Fabric Pressure Sensing,” *Computational and Mathematical Methods in Medicine*, vol. 2013, no. 1, 405325, 2013.

[11] H. Lee, D. Kwon, H. Cho, I. Park, and J. Kim, “Soft Nanocomposite Based Multi-point, Multi-directional Strain Mapping Sensor Using Anisotropic Electrical Impedance Tomography,” *Scientific Reports*, vol. 7, no. 1, Jan. 2017.

[12] J.-B. Chossat, H.-S. Shin, Y.-L. Park, and V. Duchaine, “Soft Tactile Skin Using an Embedded Ionic Liquid and Tomographic Imaging,” *Journal of Mechanisms and Robotics*, vol. 7, no. 2, 021008, May 2015.

[13] T. Beer, *Ecoscenography: An Introduction to Ecological Design for Performance*. Singapore, Palgrave Macmillan, 2022.

[14] S. Pantouvaki, I. Fossheim, and S. Suurla, "Thinking with costume and material: a critical approach to (new) costume ecologies," *Theatre and Performance Design*, vol. 7, no. 3–4, pp. 199–219, Oct. 2021.

[15] R. B. Pires and F. G. O. Júnior, "Experimentação com bioplásticos de amido de mandioca: uma colaboração entre designers de moda e produtores das artes da cena para o uso de materiais econômica e ambientalmente sustentáveis," *in Proceedings of the 15º Colóquio de Moda*, Porto Alegre, 2019.

[16] R. B. Pires, "The performative aspects of biomaterials: Biodegradable manioc (cassava) starch costumes for performances of regeneration" *Studies in Costume & Performance*, vol. 10, no. 1, pp. 11–35, 2025.

[17] A. C. Monteiro and R. B. Pires, "De/Re:Generação: desenvolvimento de uma obra multimídia baseado em cigarras e bioplásticos como interfaces de toque," *in Proceedings of the Congresso da Associação Nacional de Pesquisa e Pós-Graduação em Música (ANPPOM),* Campo Grande, Brazil, 2025.

[18] P. Kauppinen, J. Hyttinen, and J. Malmivuo, "Sensitivity Distribution Visualizations of Impedance Tomography Measurement Strategies," *International Journal of Bioelectromagnetism*, vol. 8, no. 1, pp. 63–71, 2006.

[19] M. Koelle, M. Nicolae, A. S. Nittala, M. Teyssier, and J. Steimle, "Prototyping Soft Devices with Interactive Bioplastics," *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology (UIST '22)*, Bend, OR, USA, 2022, pp. 1–16.

[20] F. Bell, M. Alistar, and L. Buechley, "Integrating biomaterials and interactive technologies: practice-based perspectives on the growth of biodesign within human-computer interaction", *Biotechnology Design*, vol. 4, e3, pp. 1–12, 2025.

[21] E. S. Lazaro Vasquez, N. Ofer, S. Wu, M. E. West, M. Alistar, and L. Devendorf, "Exploring Biofoam as a Material for Tangible Interaction", *Proceedings of the 2022 ACM Designing Interactive Systems Conference (DIS '22)*, Virtual Event, Australia, 2022, pp. 1525–1539.

[22] S. Guridi, M. Iannacchero, and E. Pouta, 'Towards More Sustainable Interactive Textiles: A Literature Review on The Use of Biomaterials for eTextiles', in *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, Honolulu, HI, USA, 2024.

[23] E. S. Lazaro Vasquez, H.-C. Wang, and K. Vega, "The Environmental Impact of Physical Prototyping: a Five-Year CHI Review," *Proceedings of the Self-Sustainable CHI Workshop (SelfSustainableCHI 2020)*, Honolulu, HI, USA, 2020.

[24] É. Bozó *et al*., "Bioplastics and Carbon-Based Sustainable Materials, Components, and Devices: Toward Green Electronics," *ACS Applied Materials & Interfaces*, vol. 13, no. 41, pp. 49301–49312, 2021.

[25] F. Bell, N. Ofer, and M. Alistar, "ReClaym our Compost: Biodegradable Clay for Intimate Making," *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems (CHI '22)*, New Orleans, LA, USA, 2022, art. no. 454, pp. 1–15.

[26] M. Dunne. *Bioplastic cook book*. 2018. Retrieved December 26, 2025 from https://library.ucsd.edu/dc/object/bb5094430n/_2_1.pdf.

[27] D. J. Haraway, *Staying with the Trouble: Making Kin in the Chthulucene*. Durham: Duke University Press, 2016.

[28] O. Oluwasina, A. Aderibigbe, S. Ikupoluyi, O. Oluwasina, and T. Ewetumo, "Physico-electrical properties of starch-based bioplastic enhanced with acid-treated cellulose and graphene oxide fillers," *Sustainable Chemistry for the Environment*, vol. 6, p. 100093, 2024.

[29] A. Amri, H. Rahmana, S. P. Utami, R. S. Iriyanti, Z. T. Jiang, and M. M. Rahman, "Conductive composites of tapioca based bioplastic and electrochemical-mechanical liquid exfoliation (EMLE) graphene," in *Proceedings of the 2nd International Conference on Oleo and Petrochemical Engineering (ICOOPChE 2017)*, Pekanbaru, Indonesia, 2017, art. no. 012026.

[30] A. Creagan, J. Bradfield, S. Richardson, L. S. Johns, K. Burrowes, H. Kumar, P. M. F. Nielsen, and M. H. Tawhai, "A wearable open-source electrical impedance tomography device," *HardwareX*, vol. 18, 2024.

[31] J. Bullock and A. Momeni, "ml.lib: Robust, Cross-platform, Open-source Machine Learning for Max and Pure Data," *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2015.

[32] S. Matsumura, "Synthesis of cicada chirping sound by physical modeling synthesis," *Journal of Sonic Arts*, vol. 14, no. 2, pp. 9–12, 2022. Available: https://data.jssa.info/paper/2022v14n02/2.Matsumura.pdf. Accessed: Dec. 08, 2025.

[33] A. Farnell, *Designing Sound*. MIT Press, 2010.

Studio Report: Laboratory of Advanced Music Production, Senzoku Gakuen College of Music

Takeyoshi Mori
Laboratory of Advanced Music Production
Senzoku Gakuen College of Music
t-mori@senzoku.ac.jp

ABSTRACT

The Laboratory of Advanced Music Production at Senzoku Gakuen College of Music explores new forms of artistic expression through the integration of music and technology. Our mission is to expand the possibilities of music creation, performance, and listening beyond the framework of traditional music research and practice. The Laboratory brings together artists, engineers, and researchers in a collaborative environment that integrates education, creative production, research, and international exchange.

The facility supports immersive and technology-driven artistic work, including multichannel and spatial audio systems, multi-projection environments, and motion-capture technologies. These systems can be flexibly combined through a dedicated high-speed media network within the college campus, enabling large-scale and experimentally oriented productions. This studio report introduces the structure, facilities, and activities of the Laboratory, outlining its research and creative directions in the field of computer music. By sharing our work with the international community, we aim to contribute to the ongoing development of technologically informed musical expression.

1. INTRODUCTION

Senzoku Gakuen College of Music offers a wide range of programs that cover many musical and performing-arts fields, including classical music, jazz, film and game music, musical theatre, dance, ballet, voice acting, and media arts production. Students are therefore able to engage with music not only as sound, but also as a multidisciplinary art that intersects with movement, theatre, and visual expression. This laboratory was established in 2025 within this diverse educational environment.

We view the coexistence of these many artistic disciplines as one of the College's central strengths. By connecting these practices with contemporary music technology, we seek to explore new possibilities for music creation, performance, and listening that go beyond the framework of traditional music research and production. In doing so, we aim to develop innovative forms of artistic expression and novel ways of experiencing music in technologically rich and immersive contexts.

Copyright: © 2025 Takeyoshi Mori. This is an open-access article distributed under the terms of the Creative Commons Attribution License 3.0 Unported, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

2. FACILITIES AND TECHNICAL INFRASTRUCTURE

The Senzoku Gakuen College of Music campus hosts an extensive range of specialized audio and media facilities that support the activities of this laboratory. These include five recording studios of different scales, a performance space equipped with a 7.1.4-channel loudspeaker system, and a motion-capture studio.

2.1 Recording Studios

The main recording studio is centered around an SSL Duality console (48-channel input). The system is based on a Mac Pro running Pro Tools, with Genelec 8040A and 1035B monitors for stereo reproduction, and musikelectronic geithain RL901K loudspeakers for 5.1 surround monitoring. A range of outboard signal processors is also installed to support advanced recording and mixing workflows.

Two mid-size recording studios are each equipped with an SSL AWS 924 Delta console (24-channel input). Monitoring is provided by RL901K and Focal Trio6 Be studio monitors. The associated recording booths can also function as small theatre spaces and are equipped with Auro-3D 13.2-channel playback systems.



Figure 1. B305 MA Studio (mid-size studio used primarily for mixing and post-production)

In addition, two multichannel studios are built around Avid S6 consoles and are used for MA work and Dolby Atmos production. Both studios feature 7.1.4-channel monitoring systems using Genelec The Ones loudspeakers, and each includes an adjacent recording booth that

allows for flexible use across production, research, and teaching.

2.2 Performance Space

The main working space of the Laboratory is "M-Studio," a large performance environment equipped with a 7.1.4-channel spatial audio system. The room is approximately 15 m × 15 m with an 8 m ceiling height. All audio systems are integrated through a Dante network, and the space also supports three-channel multi-projection video and a large-scale theatrical lighting system. The studio features a motorized rigging system with multiple lighting battens, allowing loudspeakers, projection equipment, and stage lighting to be positioned and reconfigured freely according to the specific needs of each project.

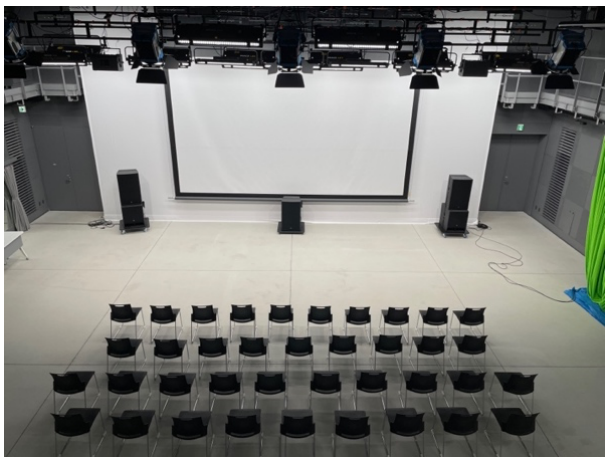


Figure 2. M-Studio (flexible performance space)

2.3 Networked Media Infrastructure

A dedicated high-speed fiber-optic media network links multiple studios and performance facilities across the campus. Audio is distributed primarily using the Dante system, while video is carried over two channel SDI connections, with additional ethernet links for data and control. This infrastructure enables integrated operation between remote spaces, including the motion-capture studio (equipped with OptiTrack Flex 13), a 1,200-seat concert hall, and several rehearsal rooms. As a result, large-scale, interdisciplinary, and technologically complex projects can be realized across the campus.

3. EDUCATIONAL PROGRAMS

Although this laboratory is an independent research organization within the college and is not formally attached to any specific course, it has a particularly close relationship with the Music Design Course, as many of its activities involve the application of music technology.

The Music Design Course is one of the largest programs at Senzoku Gakuen College of Music, with more than 500 students including both undergraduate and graduate students. The course is broadly divided into two main fields of specialization: composition/arranging and recording/production. In addition to these core areas, students are able to study a wide variety of technology-related subjects through elective courses, including media programming with Cycling'74 Max, 3DCG production, and video and stage work using motion-capture systems and interactive technologies.

Students from both the Music Design Course and the newly launched Media Arts Course, which specializes in visual production, actively participate in the Laboratory's projects. Through involvement in these research and creative activities, students gain practical experience in both the artistic and technical dimensions of music technology, and are able to explore new forms of musical expression that integrate sound, visual media, and interactive technologies.

4. RESEARCH AND CREATIVE ACTIVITIES

This section outlines the research directions and creative activities currently undertaken by the Laboratory.

4.1 Research Themes

The Laboratory focuses on three closely related areas: practice-based research that integrates immersive, sensing, and AI technologies; the exploration of creative artistic expression through music and media production; and the development of international research and creative networks. Through these activities, the Laboratory aims to connect technological innovation with artistic practice, fostering new forms of computer music creation and performance.

4.2 Creative Projects

A major project carried out by the Laboratory is *47Trees*, an immersive media-theatre work created in collaboration with the Department of Music at Stanford University. The music and overall concept were developed by Jarosław Kapuściński. The work integrates sound, video, and live performance in a way that allows the audience to experience the performance space as an immersive environment. The Laboratory contributed primarily to the spatial-audio design using a multichannel loudspeaker system, the 4K multi-projection system used as a stage-design element, and the streaming of immersive media content.

This project also involved the active participation of undergraduate and graduate students from the Music Design Course, who were responsible for spatial-audio programming using Cycling'74 Max 9 and video generation using TouchDesigner. Through collaboration with musicians and artists including Jarosław Kapuściński, the project provided a valuable opportunity for students to apply knowledge acquired through coursework in a practical creative setting, resulting in a strong educational outcome.



Figure 3. A scene from *47Trees*

Other recent projects include an industry–academia collaboration with NeSTREM Live and Korg Live Extreme, in which an orchestral performance in the college concert hall was transmitted in real time through the dedicated high-speed fiber-optic media network to S-Studio, which is connected to M-Studio. There, recording, mixdown, and live streaming in Dolby Atmos were carried out, while the same data were simultaneously sent to M-Studio to create a real-time listening and viewing environment in Dolby Atmos. This project also provided undergraduate students in the recording and production field of the Music Design Course with an opportunity to learn advanced techniques for Dolby Atmos recording and streaming in a practical setting, demonstrating how the Laboratory's activities contribute directly to university education.

In this live-streaming project, two distribution platforms, NeSTREM Live and Korg Live Extreme, participated in technical cooperation. The system used the Live Extreme live encoder and delivery infrastructure, while the distribution stream was received and played back through the NeSTREM Live application and the Live Extreme application or a web browser. The collaboration between these two platforms, which would previously have been regarded as competitors, was highly innovative and provided useful knowledge for future collaborative projects.

4.3 Outreach and Collaboration

The Laboratory is also engaged in a wide range of outreach and collaborative activities, including guest lectures, workshops, concerts, and participation in international conferences.

In 2024, Hans Tutschku of Harvard University was invited to present Senzoku Electroacoustic Music Concert, which featured an electroacoustic music concert utilizing an immersive audio system. The concert also included fixed-media works by graduate students in electroacoustic music that had received awards in an international competition and were newly produced in Dolby Atmos for this occasion. In addition, audiovisual works by faculty members of Senzoku Gakuen College of Music, including the author, were presented through the same immersive sound system.



Figure 4. Flyer for Senzoku Electroacoustic Music Concert 2024

In 2025, Jon Balke of the Norwegian Academy of Music was invited to lead Senzoku Electroacoustic Music Concert. The concert included creative works using his "Spektrafon" system, as well as student compositions and performances, such as an improvisation for piano and AI-driven electronics created by a graduate student from the Music Design Course.

It also hosted a guest lecture by Mari Kimura (University of California, Irvine), an internationally active violinist and computer music researcher, focusing on creative applications of her sensor-based performance interface "MUGIC".

In addition, the Laboratory participates in international academic exchanges, including collaborative events with the Music and Image Trans-disciplinary Program at the Taipei National University of the Arts, and presentations at the Forum of AI Music Production held at Tainan University of Technology.

Through these activities, the Laboratory aims to foster experimental artistic practices that integrate sound, visual media, performance, and interactive technologies in an international context.

5. FUTURE DIRECTIONS

The Laboratory of Advanced Music Production will continue to develop research and creative activities that connect music with technology. In particular, we aim to further expand our work in areas such as immersive audio, interactive systems, and AI-based creative practices, and to apply these technologies in practical artistic contexts.

We will also strengthen our collaboration with overseas institutions and researchers, creating more opportunities for international exchange and joint creative work.

At the same time, we will ensure that the Laboratory remains a place where education, research, and artistic production naturally come together, providing an environment in which students and artists can continue to explore new possibilities for music expression with technology.

6. REFERENCES

[1] Senzoku Gakuen College of Music:
https://www.senzoku.ac.jp/music/en
Accessed: March 30, 2026.

[2] The Laboratory of Advanced Music Production:
https://www.senzoku.ac.jp/music/laboratory/lamp/
Accessed: March 30, 2026.

[3] 47Trees Homepage: https://47trees.com
Accessed: March 30, 2026.

Co-Composing with Plants: Early Experiments in Bio-Responsive Score Design

Solomiya Moroz
Independent
info@solomiyamoroz.com

Nicolo Merendino
University of Hong Kong of Science and Technology
nicolom@hkust-gz.edu.cn

Massimo Sterlino
Independent
max@sterlinux.it

ABSTRACT

This paper presents a novel compositional system that positions plants as active, agential collaborators. We developed a custom IoT sensor device to measure a plant's biophysical state, including electrical activity, light, and humidity, and stream this data in real time to a bespoke software environment. Unlike commercial bio-sonification devices that generate ambient sound, our system translates biophysical fluctuations into the structural elements of a live musical score. The project is grounded in posthumanist and new materialist frameworks, which embrace interspecies entanglement. Here, collaboration is reconceived as a non-hierarchical network: the plant influences algorithmic score generation, software mediates the data, and human performers interpret the live-generated notation, creating a continuous feedback loop. This approach challenges traditional paradigms of human-environment interaction, proposing a relational and interdependent creative process. The system also serves as the technical foundation for an in-development chamber opera, where musicians perform by interpreting a score generated in real time by their more-than-human partners.

1. INTRODUCTION

Over the years, plant-to-sound technologies have transitioned from niche experiments to commercially accessible tools, with devices like *PlantWave* [1], *Music of the Plants* [2], and *PlantsPlay* [3] leading the market. These systems typically convert changes in bio-electrical data initiated by TENS sensor (Transcutaneous Electrical Nerve Stimulation) [4] electric current flows into pre-mapped MIDI notes, either integrating with standard DAWs or generating sound through proprietary engines. Notably, many such devices can be DIY-built using basic sensors [1], blurring the lines between scientific instrumentation, artistic toolkits, and agricultural monitoring systems.

In our project, we developed a custom sensor array that measures electrical activity (via an ECG-analog circuit), light exposure, and humidity. These data are transmitted via an IoT module to a real-time music software environment. Unlike commercial applications that prioritize ambient sound generation, our system treats plants as active collaborators in the compositional process. By translating biophysical fluctuations into structured pitch and rhythmic data, the plant becomes a living score generator—enmeshed in a feedback loop with musicians who interpret and respond to its signals, whose sound vibrations in turn stimulate further sensor activity. This approach challenges conventional hierarchies in human-environment interaction, contributing to a relational and interdependent creative process.

Additionally, we situate this study within the broader context of sensor-based art and music projects, providing a comparative analysis of our biofeedback device and its use in real-time score generation, while also positioning the research within wider theoretical discussions on plant collaborative behaviour and posthumanism. The system presented here is an open, evolving framework that invites multiple theoretical interpretations: as an extended embodied network and as a posthumanist, agential subject. While these interpretations offer productive lenses, they currently operate within a predominantly anthropocentric model. We address this tension in the discussion, seeking to understand how the ecological concern raised by our research might be more adequately addressed in future iterations. The work is structured around the following central research question: *Can plants function as artistic collaborators—not merely as passive subjects, but as agential, more-than-human partners?* To address this question, we present:

1. a theoretical framework that underpins the research
2. an account of the artistic implementation
3. a description of the technical system implementation
4. a discussion of the findings and their broader implications

2. THEORETICAL FRAMEWORK

In literature on plant collaborative behaviour, it has been established that plants form mutually beneficial relationships with mycorrhizal networks to secure nutrients, forage for resources, ward off danger, and generate reciprocal gains [5]. This relationship, known as a mycorrhizal symbiosis, is one of the world's most vital partnerships [6], [7]. This embodied plant behaviour exemplifies the 4Es of cognition, transcending the definition of cognition as exclusively neuronal but based on Embodied, Embedded, Enacted, and often Extended process [8], [9], [10].

The first three principles of the 4E framework are relatively straightforward: cognition is Embodied (it requires a body), Embedded (it is inseparable from its environment), and Enacted (it expresses itself through action in the world). The fourth principle, Extended cognition, is the most applicable to our discussion on plants' behaviour. It proposes that cognitive processes can occur, in part, beyond the physical body of the organism [11]. Drawing a direct analogy to the

Copyright: ©2026 Solomiya Moroz et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License 3.0 Unported, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

extended cognitive network of plant-mycorrhizal symbiosis, our project positions collaborative, extended behavior as the theoretical core of its bio-feedback system for artistic collaboration with plants.

Additionally, the question of plants' artistic collaboration is based on the work of artist-researchers Reardon-Smith [12] and Donohue [13], who critique rigid boundaries between human/nonhuman and organic/digital actors. Their work aligns with Donna Harraway's [14] seminal challenge to anthropocentrism (*"Do our bodies end at our skin?"*) and with Beth Coleman [15] and Russell Legacy [16], who interrogate materiality and "reality" in hybrid digital-physical ecosystems.

These perspectives collectively anchor our project within posthumanist, new materialist, and techno-feminist discourses, which favour interspecies entanglement. Collaboration is thus reconceived as a non-hierarchical network: plants reacting to sound vibrations influence the algorithmic score, and performers reinterpret this output, establishing a cyclical feedback loop as plants are subjected to new sound inputs. This model diverges sharply from traditional performance frameworks, where technology and ecology are subordinate to human control.

2.1 Related Work

2.1.1 Plant data sonification projects

Devices employing transcutaneous electrical nerve stimulation (TENS) electrode systems are widely used in electronic plant music for generating electronic plant sound. Projects like, *Data Garden Group's* 2012 Philadelphia Museum of Art sound installation, created around the announcement of their MIDISprout device [1] uses a TENS-based system. The group devised an imaginary "bioband" of four synthesized parts (lead, rhythm, bass, and ambient pads), producing layered, diatonic and pentatonic textures that were highly repetitive [17]. On the second day's recording, audible tempo shifts in the lead synthesizer may indicate that MIDI data from the plant interface was used not only for pitch but also to trigger tempo changes, that can flexibly control rhythmic speed. While the installation was conceptually innovative and notable for its early role in a then-nascent field, the resulting music may seem monotonous to some listeners because of its limited harmonic and structural variety [4].

Other artistic projects like Augustine Leudar's *The Code*, and *El Bosque Encantado* use plants' bioelectrical changes for in situ data sonifying installations [4]. Leudar worked primarily with the idea that the pathways of electrical signals transmitted through a subterranean root and fungal network could be made audible [18]. Although not conceived as a data sonification project, this installation renders the inaudible signals from the tree-root ecosystem audible by transposing them into the human hearing range [18]. This data is also mapped to synthesized sounds and played live in situ, where the environmental data itself becomes a perceptible sonic element [18]. Of additional relevance is the deployment of fungal networks, analogous to those observed in mycorrhizal symbioses, as a conceptual node for signal exchange. These networks were modeled as intermediaries for tracking interactions among distinct electrical signals within the arboreal network, which in turn informed and constrained the decision-making processes governing the ambisonic spatialisation of the work [18].

2.1.2 Music Score precedents

Composer Mamoru Fujieda has incorporated plants into his music for nearly three decades. His initial work began in collaboration with media artist Yuji Dogane, resulting in the 1994 album *Ecological Plantron*—an audio document of a Tokyo gallery installation [4]. Dogane's "Plantron" device converted bioelectrical plant signals into MIDI notes using Fast Fourier Transform (FFT).

Fujieda's later cycle, *Patterns of Plants* (1995–2011), is scored for ensembles of Japanese instruments in various tunings and has also been adapted for piano. Fujieda expressed a desire to hear plants' "daily activities" as "their voices" [19]. However, the biosignals in these works are not presented directly but are highly mediated—freely filtered and interwoven with diverse musical traditions, including Japanese chamber music, gamelan, and Western postminimalism.

2.1.3 Artistic precedents

Recently, there have been several artistic projects looking specifically at agential behaviour of plants and what it can tell us what they might be experiencing. Notably, a project, created by Maria Castellanos and Alberto Valverde, titled *Other Intelligences (OI)* (2022) [20]. The project involves creating a networked system of houseplants across different domestic locations, connected via the internet. Over time, researchers observed that these plants communicated their state by sharing sonic signals and responding to each other, as well as to the presence of humans in the room. This interaction was evidenced by their responses to various stimuli and by the observable effect stimuli from other, remotely connected plants had upon them. This work also produced interesting qualitative findings about how humans can be affected through their interaction with more-than-human elements. Particularly noticeable was the effect on human perceptions of plant agency. This project aims to engage with and build upon these findings as part of the artistic project described below.

2.2 Creative Project Description

Our project moves beyond utilitarian roles to explore extended embodied plant behaviour, reconceiving its agency within the domain of sound and musical practice. We specifically investigate how plant agency functions in sound-producing environments and how the interplay of sound and human interaction influences plant behavior while reciprocally shaping human musicianship.

This work incorporates elements from the aforementioned projects, including bioelectric and environmental sensing. However, it diverges in its primary output: we do not engage in direct data sonification. While sonification remains an option within the broader chamber opera project, the core function of our sensor data is to drive real-time score generation. However, unlike Fujieda's filtered and intermodulated approach, our system generates musical phrases and structures in real time within the computer-assisted composition environment of *bach* [21]. Here, the incoming

sensor data is mapped directly to pitch material within predefined registral limits, and rhythmic organization, creating a dynamic compositional framework rather than a direct sonic translation.

Additionally, our experimentation with the biofeedback system device provides the conceptual foundation for a chamber opera currently in development by Solomiya Moroz. In this work, musicians act as interpreters of plant signals by performing from a score generated in real time.

Musically, the opera merges a traditionally scored composition with material generated in real time by the plants themselves. An onstage installation of living plants responds to environmental stimuli—light, heat, and sound vibrations—emitting bio-electrical signals. These signals are processed by the bio feedback score system, which then subtly infiltrates and modifies the performance altering the musicians' score, and influencing the electronic soundtrack.

This interplay forms an interdependent ecosystem. The central thematic paradox asks: If humanity struggles to understand its own signals, how can we hope to interpret the subtle communications of nature? The opera invites us to consider whether we will learn to attune to these messages before crisis is unavoidable, exploring how we might form deeper understanding within our human and ecological relationships. The live score is hosted in a Max/MSP environment as a computer-aided composition and is streamed to musicians over a Wi-Fi network.

3. TECHNICAL EXECUTION

3.1 Process

The bio-feedback device was developed through an iterative research-through-prototyping process aimed at transforming a living plant into an Internet of Things (IoT) device. Initial phases focused on literature review and exploratory measurements of plant bio-electrical signals using non-invasive techniques. Early prototypes were intentionally simple, validating signal detectability, frequency bands of interest (25 Hz), and responsiveness to stimuli. Subsequent iterations progressively integrated signal conditioning, multi-sensor acquisition, local storage, real-time transmission, and robustness improvements driven by benchmarking and artistic use-case requirements (computer music performance).

3.2 Hardware

The hardware architecture is an easily accessible, low-cost, low-power embedded system built around an ESP32 Wroom microcontroller [22], which performs 12-bit ADC sampling at 100 Hz and enables wireless communication. Bioelectrical signals are captured with Ag/AgCl electrodes and conditioned by an AD8232 analog front-end, originally designed for biopotential measurements. This core plant signal is contextualized by environmental sensors: a photoresistor (with a 100 k resistor voltage divider) for ambient light and an FC-28 soil moisture sensor with an LM393 comparator. A custom 4th-order passive RC low-pass filter (cutoff 25 Hz) attenuates mains interference. For data management, a MicroSD card module provides local storage, controlled by dedicated REC/STOP push buttons, and the entire assembly is powered by a 5V USB supply. This design ensures faithful signal reconstruction while supporting both real-time transmission and local data logging. The system is designed to foster modularity and interoperability .

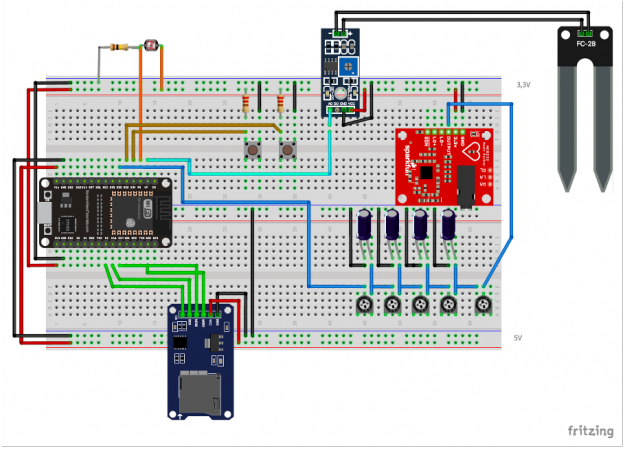


Figure 1. Schematic of the device

List of components:

- ESP32 Wroom microcontroller
- AD8232 biopotential amplifier
- Ag/AgCl electrodes
- Photoresistor + 100 k resistor (voltage divider)
- Soil moisture sensor (FC-28 with LM393)
- Passive RC low-pass filter (4th order)
- MicroSD card module
- Push buttons (REC / STOP)
- USB Power supply (5 V)

3.2.1 Schematic

The complete schematic integrates the AD8232 output into an ESP32 ADC pin, environmental sensors into parallel ADC channels, a passive low-pass filter upstream of sampling, and SPI connections for microSD storage. Control buttons manage session state (see Figure 1).

3.2.1.1 Code

Firmware is written in C++ and leverages FreeRTOS on the ESP32's dual-core architecture. A producer–consumer model is implemented: one real-time task samples sensors at a fixed rate, while a secondary task handles data storage and transmission, preventing timing jitter. Data are structured into fixed records and serialized as CSV and JSON.

3.3 IoT connection

The device functions as a full IoT node, publishing sampled data in real time to a remote infrastructure via its integrated ESP32 Wi-Fi module. Data is transmitted using the MQTT publish-subscribe protocol to a central Mosquitto broker, where a Python subscriber service ingests the JSON-formatted data and stores it in a MongoDB NoSQL database for downstream analysis. For local redundancy, data is

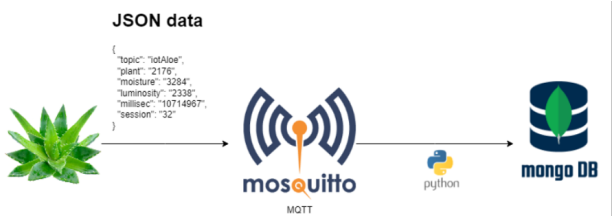


Figure 2. Architecture of IoT Communication

also archived in CSV format (see fig. 4). This continuous streaming of plant and environmental data enables its live integration into interactive systems, such as musical performances, while also feeding dedicated analysis pipelines.

3.3.1 Architecture

The ESP32 publishes sensor packets to dedicated MQTT topics. A Python-based subscriber listens to these topics and writes incoming messages into a MongoDB database, where each sample corresponds to a document. This architecture supports scalability (multiple plants, multiple topics), real-time access, post-processing, and integration with machine-learning or multimedia systems (see fig. 2).

The final result is a robust, non-invasive IoT system that enables a living plant to function as an active data-producing device (Figure 3). The system reliably captures low-frequency bio-electrical signals generated by the plant, correlates them with environmental parameters such as light intensity and soil moisture, and makes the resulting data available both locally and remotely in real time. Through careful signal conditioning, synchronized sampling, and a multitasking embedded architecture, the device guarantees temporal consistency and data integrity, even during long recording sessions.

Beyond data acquisition, the system enables real-time streaming of plant signals over the network, allowing them to be immediately consumed by external software systems. This makes the plant directly usable as an input interface for interactive applications, including computer music performance, live score generation, sound synthesis, and multimedia installations. At the same time, the structured storage of data in both CSV files and a NoSQL database supports offline analysis, and visualization.

3.4 Software and the Visual Interface

The live data stream is routed via Python to a Max/MSP interface, where sensor data is parsed into three distinct parameters: ”plant” (bio-electrical activity), ”moisture”, and ”luminosity” as pictured in fig. 2. This data drives a real-time score-generation environment for live musicians.

For the live score generation, we used the *bach* ecosystem for computer aided composition (CAC)[21] package objects. The system built in CAC allocates rhythmic, pitch, and instrumental material within predefined instrumental register limits. The Max patch is designed to distribute incoming data across six instrumental parts visible in the *Streamed Score Example*, Fig. 5.

Initially, pre-recorded ”plant” data from CSV files was used for testing. Once the live IoT device was operational, the system utilized all three data streams, each duplicated, to generate a prototype score for rehearsal and development.



Figure 3. Plant with Device

plant2

plant	moisture	luminosiy	millisec	session
985	4095	92	504260	113
1033	4095	91	504270	113
1008	4095	93	504280	113
1067	4095	92	504290	113
1042	4095	89	504300	113
1104	4095	89	504310	113
1090	4095	90	504320	113
1173	4095	81	504330	113
1200	4095	89	504340	113
1319	4095	87	504350	113
1360	4095	87	504360	113

Figure 4. Sampled Data Example from Plant 2

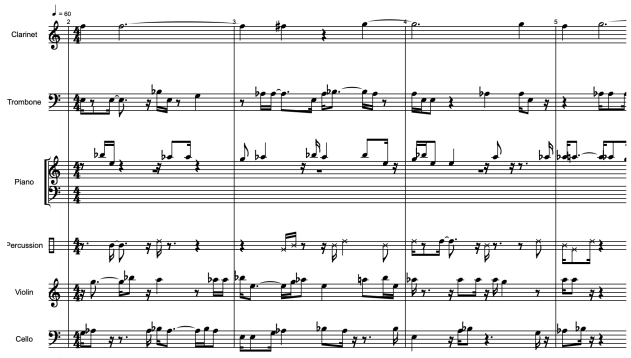


Figure 5. Streamed Score Example

In future iterations, this setup will be scaled to incorporate at least two live devices for full-score generation.

3.5 Programming within Max/MSP

The score generation is a creative decision-making space where the composer sets up specific rhythmic and instrumental relationships using the incoming data as pitch material (see fig. 6). This creates a flexible, evolving environment for this collaborative human and more-than-human musical work. While the collaboration occurs with plant-based impulses, the composer ultimately shapes this data into a musical score that could be meaningful for the interpreting musicians. The patch manages data flow through three essential stages:

1. Timed Input: A *metro* object controls the initial data input to subsequent *bach* objects, cycling every 4 seconds (aligned with 4 measures at 60 BPM). This interval is adjustable.

2. Rhythmic and Polyphonic Processing: A distributed polyphonic section allows for the composition of specific rhythmic groupings and rests. A dedicated *bach.join* subpatch then consolidates the six processed data streams into a single, unified score for the six instruments (see fig. 7).

3. Networked Output and Refinement: The completed score is sent to a networked score space via Mira [23], where it is distributed to the performers' iPads for live use. In this final space, additional performance parameters—including articulations, tempo, and phrasing—can be adjusted.

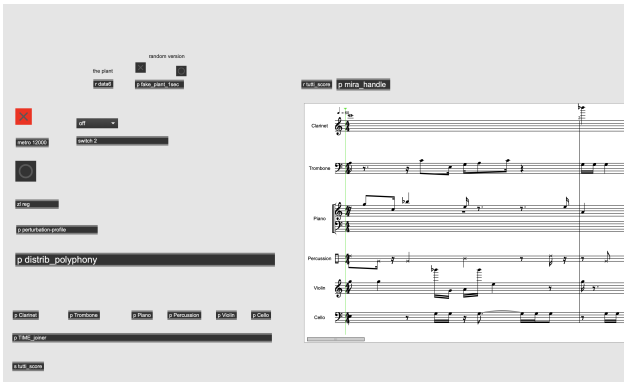


Figure 6. Main interface in CAC

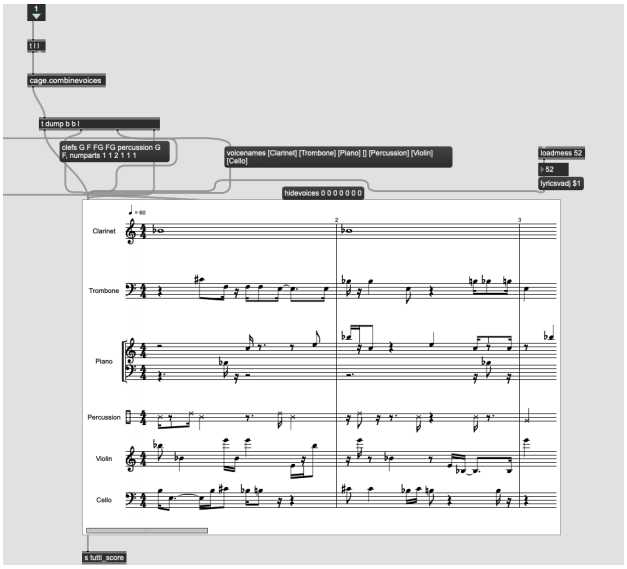


Figure 7. Inside bach.join subpatch

3.6 Interpreting and Collaboration

The scoring system was tested by six musicians during its development in several rehearsal sessions. It required continuous adjustment to reach a stage where the real-time score was sight-readable and manageable to scroll in real time. Most sessions utilized pre-recorded plant data from previous sessions to ensure a smooth rehearsal process; therefore, the live feedback loop between musicians and plants remains to be fully tested. Similarly, empirical data on musicianship is currently lacking and would require follow-up studies, such as surveys and in-depth interviews,

to document performers' experiences with the scoring system. However, we have tested the system with separate sound files in order to determine whether changes in sound source volume and quality have an influence on the plants' electrical activity. It was noted that softer string sounds corresponded to lower electrical signal activity (in the range of 1600–2000), whereas louder sounds from noise and bass-focused music returned values between 1900 and 2400. Additionally, changes were observed under different lighting conditions: signals measured during full lighting, as well as in dim and dark conditions, were also affected, with brighter lighting conditions returning higher electrical signal measurements. This data is stored on CSV files and supports our hypothesis about the feedback nature of our system.

4. DISCUSSION

4.1 Collaborative and Ethical Considerations

The system presented here is an open, evolving framework that invites multiple theoretical interpretations: as an extended embodied network and as a posthumanist, agential subject. While these interpretations offer productive lenses, they currently operate within a predominantly anthropocentric model. The plant itself did not consent to its role as a musical agent; it is merely responding to its environment as it naturally would. This mirrors a foundational ecological truth: nature is not dependent on humanity, but humanity is inextricably dependent on nature.

This tension leads to a critical ethical and practical question: could such a system be designed not only to extract data but to provide reciprocal benefits to the plants, thereby granting them a form of agency within their own ecosystems?

While the direct benefits of our system to the plants require further empirical study, related research suggests this is a plausible direction. The work on networked plant intelligence [20] demonstrates that plants can form selective, sustained connections, similar to 'friendships', through enhanced communication in a sensor-mediated environment. This indicates a capacity for decision-making and relational dynamics within non-neuronal networks [24], aligning with theories of extended plant cognition.

4.2 Relationship to Prior Work

In contrast to projects which use TENS-based systems, our device employs an electrocardiography (ECG) sensor to measure bioelectrical signals at the leaf surface. It operates as a passive system with high input impedance, designed to minimize perturbation of the measured tissue. TENS, by comparison, is conceptually inverse in function: rather than performing measurements, it delivers electrical impulses into the tissue, thereby introducing current, altering the physiological state of the biological system, and eliciting a response to the applied stimulus. When applied to plants, a TENS device does not monitor intrinsic, spontaneous electrical activity; instead, it imposes an exogenous electrical stress that evokes a reactive response from the plant. In contrast, our system does not actively modulate or interfere with tissue behavior, but rather records the plant's endogenous electrical activity as it occurs spontaneously.

In addition to the bioelectrical sensor, the system integrates three environmental sensors to simultaneously track light and moisture conditions. This multi-modal data provides flexibility, allowing the system to map various ecosystem states to different musical parameters. For instance, in addition to score generation, the patch can be programmed to generate distinct soundscape environments based on these real-time inputs.

In line with other works reviewed, our approach similarly utilizes bioelectric and environmental sensing. The key point of departure, however, lies in its output. Rather than focusing on direct data sonification like Leudar's in site sound installation [18], our primary objective is to channel this sensor data into real-time score generation. Our method also distinguishes from Fujieda's aesthetic of filtered and intermodulated signals. Instead, within the bach computer-assisted composition environment, sensor input is structured into musical phrases by directly mapping it to pitch (within defined registral bounds) and rhythmic parameters. The result is not a direct translation of data into sound, but a dynamic, generative compositional framework.

4.3 Current Limitations

Our current prototype is a physically open system on a breadboard without a PCB, making it susceptible to minor disturbances such as sound or dust. A custom 4th-order passive RC low-pass filter (cutoff 25 Hz) attenuates mains interference, though the data can still require a brief settling period after connection before stabilizing.

Another current research limitation of the project is the lack of systematic data on the musicians' experience, which prevents conclusive findings regarding the effectiveness and subjective impact of the bio-feedback loop. We plan to address this through larger-scale experiments beyond creative academic projects, which are scheduled to commence later this year.

4.4 Future Developments

To address the aforementioned device limitations, future research and development activities could focus on integrating higher-performance electrodes and enhancing the signal amplification and filtering circuitry to obtain signals with reduced noise and improved fidelity. In addition, the system could be migrated to a custom-designed device [25], including a printed circuit board (PCB) and other features to enable more robust, compact, and reliable device implementation [26]. Such implementation can allow the setup to be implemented in on-site outdoor performances and installations.

Future major improvements could involve implementing machine learning to train the system to recognize patterns in live plant data streams in real time. For this we can already use samples of data available in CSV files (fig.4) before testing in real time. A faster, adaptive response to computational changes would enhance the visual score interface and enable the programming of more complex interactions within the patch's overall structure, such as dynamic adjustments to pitch, rhythm and soundscapes. This development could further entangle our system and its artistic applications within posthumanist and more-than-human discourses, the implications of which would require

systematic examination and critical unpacking. A similar level of analytical rigor would be necessary when collaborating with musicians, in order to understand how these entanglements influence the human–computer interaction (HCI) dimensions of the system.

Our project's most significant future challenge lies in shifting from an anthropocentric *interface* to a truly reciprocal *collaboration*. This would involve designing feedback mechanisms where the system's outputs, whether sonic, environmental, or data-driven, are tailored to create conditions beneficial to the plant's own flourishing, thus closing the ethical and ecological loop opened by our research.

5. CONCLUSION

We have presented a novel IoT-based biofeedback device and system that enables living plants to function as active collaborators in live score generation, designed for integration within a broader chamber opera project. The system moves beyond existing bio-sonification tools by positioning the plant not as a passive signal source for ambient sound, but as an agential participant in a relational creative network.

From a technical perspective, the prototype demonstrates a functional pathway from biophysical signal acquisition to structured musical notation. The custom hardware and software pipeline successfully captures plant bio-electrical activity, contextualizes it with environmental data, and translates these multi-modal streams into a dynamic, performable score within a Max/MSP environment. This workflow provides a tangible model for more-than-human collaboration in computer music.

Theoretically, the project engages with frameworks of extended cognition and posthumanist performance, proposing a model where agency is distributed across plant, algorithm, and human performer. This challenges anthropocentric creative hierarchies and opens a space for transdisciplinary inquiry into the nature of collaboration, agency, and interspecies communication.

While this work is an early contribution, it establishes a foundation for several critical lines of future inquiry. These include refining the hardware for robustness, exploring machine learning for real-time pattern recognition, and—most importantly—investigating whether such systems can be designed to offer reciprocal benefits to the plants themselves, potentially granting them greater agential influence within their ecosystems. Ultimately, this project reframes the plant not as an instrument to be played, but as a potential partner in a shared, interdependent creative process.

6. REFERENCES

[1] S. Cusumano, "MIDISprout," https://github.com/electricityforprogress/MIDIsprout, accessed: 2025-12-27.

[2] C. Backster, "Music of the Plants," https://www.musicoftheplants.com/, accessed: 2025-12-27.

[3] E. Taori, "Plants Play," https://plantsplay.com/, accessed: 2026-01-02.

[4] P. V. Miller and C. Cox, "Music from plant biosignals: A conceptual and analytical orientation," *Music Theory Online*, vol. 30, no. 1, 2024.

[5] A. G. Parise, C. Marín, F. Tamagnini, M. Tibbett, and B. J. Pickles, "How mycorrhizal fungi could extend plant cognitive processes," *Symbiosis*, vol. 96, no. 2, pp. 105–120, 2025.

[6] W. Remy, T. N. Taylor, H. Hass, and H. Kerp, "Four hundred-million-year-old vesicular arbuscular mycorrhizae." *Proceedings of the National Academy of Sciences*, vol. 91, no. 25, pp. 11 841–11 843, 1994.

[7] M. I. Bidartondo, D. J. Read, J. M. Trappe, V. Merckx, R. Ligrone, and J. G. Duckett, "The dawn of symbiosis between plants and fungi," *Biology letters*, vol. 7, no. 4, pp. 574–577, 2011.

[8] F. Calvo Garzón, "The quest for cognition in plant neurobiology," *Plant signaling & behavior*, vol. 2, no. 4, pp. 208–211, 2007.

[9] M. Dawson, "Embedded and situated cognition," in *The Routledge handbook of embodied cognition*. Routledge, 2014, pp. 59–67.

[10] A. Newen, L. De Bruin, and S. Gallagher, *The Oxford handbook of 4E cognition*. Oxford University Press, 2018.

[11] A. Clark and D. Chalmers, "The extended mind," *analysis*, vol. 58, no. 1, pp. 7–19, 1998.

[12] H. Reardon-Smith, "Sounding kin: A queer-feminist thinking of free improvisation," *thesis*, 2021.

[13] M. S. Donohue, "Mythologising the social in experimental improvisation," Ph.D. dissertation, The University of Huddersfield, 2022.

[14] D. Haraway, "Encounters with companion species: Entangling dogs, baboons, philosophers, and biologists," *Configurations*, vol. 14, no. 1, pp. 97–114, 2006.

[15] B. Coleman, *Hello avatar: rise of the networked generation*. Mit Press, 2011.

[16] R. Legacy, "Glitch Feminism: A Manifesto," 2020.

[17] D. Garden, "Quartet: Live at The Philadelphia Museum of Art," https://datagarden.bandcamp.com/album/quartet-live-at-the-philadelphia-museum-of-art, accessed: 2026-01-02.

[18] A. Leudar, "Surrounded: a series of sound installations that combine plant electrophysiology and 3D sonic art," *Leonardo*, vol. 51, no. 5, pp. 517–523, 2018.

[19] M. Fujieda, "Tuning of Patterns of Plants," *Arcana IV: Musicians on Musicians*, pp. 140–150, 2009.

[20] A. Ramos, M. Castellanos, and E. Ganuza, "Other Intelligences: Investigating the Plant-Human Relationship in Domestic Spaces," *Philosophies*, vol. 9, 2024.

[21] A. Agostini and D. Ghisi, "Bach: An environment for computer-aided composition in max," in *ICMC*, 2012.

[22] N. Merendino, A. Rodà, and R. Masu, ""Below 58 BPM," involving real-time monitoring and self-medication practices in music performance through IoT technology," *Frontiers in Computer Science*, vol. 6, p. 1187933, 2024.

[23] Cycling74, "Mira for Max," https://cycling74.com/products/mira, accessed: 2026-01-02.

[24] J. Lee, M. Segundo-Ortin, and P. Calvo, "Decision Making in Plants: A Rooted Perspective," *Plants*, vol. 12, no. 9, 2023. [Online]. Available: https://www.mdpi.com/2223-7747/12/9/1799

[25] N. Merendino and A. Rodà, "Defining an open source CAD workflow for experimental music and media arts," in *Proceedings of the 10th International Conference on Digital and Interactive Arts*, 2021, pp. 1–6.

[26] N. Merendino, M. Bettega, A. P. Melbye, J. D. Sullivan, A. Rodà, and R. Masu, "Sustainable digital fabrication in nime: Nine sustainability strategies for dmi production," in *Proceedings of the international conference on new interfaces for musical expression*, 2024, pp. 266–274.

Acoustic Interactive Sand Tray Therapy System: An Embodied Interface for Multisensory Sound Interaction

Yunze Mu[†*]
School of Music
University of Louisville
yunze.mu@louisville.edu

Lorna E. Segall[†]
School of Music
University of Louisville
lorna.segall@louisville.edu

Zhixin Xu[†*]
ICCI
Shanghai Jiao Tong University
zhixin.xu@sjtu.edu.cn

ABSTRACT

Foundational work in computer music and gestural interface design has emphasized embodied and tangible interaction as a central component of expressive musical systems [1, 2]. This paper introduces the Acoustic Interactive Sand Tray System, an innovative interface that translates the continuous physical manipulation of sand into real-time auditory feedback. Utilizing an overhead depth-sensing camera and the YOLO v11 object detection model, the system captures complex surface geometries and identifies physical artifacts within the tray. We ex-tract perceptually salient features, such as surface flat-ness and regional elevation, which are transmitted via OSC to a hybrid sound engine implemented in Max/MSP, Unity, and RTcmix (RTcmix and WebRTcmix).

While the system supports interdisciplinary applications, this paper focuses on its technical architecture, specifically the sensing pipeline and the many-to-many mapping strategies that link material deformation to sound syn-thesis. By prioritizing material affordances over symbolic control, the system facilitates exploratory, low-cognitive-load engagement, allowing users to intuitively shape "sound worlds" through tactile interaction. The Acoustic Interactive Sand Tray System contributes a robust framework for material-based sound control, demonstrating the potential for non-rigid, natural inter-faces to foster immersive and embodied musical experiences.

1. INTRODUCTION

Most contemporary digital music systems rely on screen-based graphical interfaces that require users to translate physical intent into abstract parameter manipulation. While such interfaces offer precision, they often impose cognitive and perceptual constraints that limit immediacy and embodied engagement. Foundational work in computer music has long identified this disconnect between human action and digital sound control as a central challenge for expressive interaction [3].

In response, research in new musical interface design has increasingly explored embodied, tangible, and material-based interaction as alternatives to screen-centric paradigms [1, 4]. These approaches emphasize physical engagement, gesture, and bodily intuition as primary modes of musical control. Rather than treating the body as an input device for symbolic commands, embodied systems position physical interaction as an integral component of musical meaning.

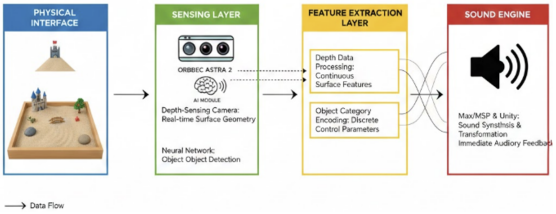


Figure 1: The Acoustic Interactive Sand Tray System dataflow.

Natural materials provide particularly rich affordances for embodied interaction. Sand, in particular, supports shaping, smoothing, excavation, and construction without predefined gestures or symbolic representations. Its continuous and non-rigid nature enables exploratory interaction driven by tactile intuition rather than learned technique. From a music technology perspective, this presents both challenges and opportunities for sensing and mapping.

This paper presents the Acoustic Interactive Sand Tray System, an interactive sound system that augments sand-based interaction with real-time auditory feedback. By tracking surface deformation and object placement within a sand tray, the system enables users to influence sound through material manipulation rather than explicit musical control. Although the system has been explored in multiple applied contexts, this paper is limited to a discussion of the system's technical design, sensing pipeline, and sound-mapping strategies.

1.1 Embodied and Tangible Musical Interfaces

The role of embodiment in musical interaction has been widely theorized and documented. Cadoz and Wanderley articulated instrumental gesture as a foundational concept in digital instrument design, emphasizing the inseparability of physical action and sound production. Subsequent work has expanded this framework within the context of sensor-based and tangible musical interfaces, particularly through the New Interfaces for Musical Expres-sion (NIME) community.

Tangible musical interfaces typically employ physical ob-

[†] These authors contributed equally to this work and share first authorship.

Copyright: ©2026 Yunze Mu[†] et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License 3.0 Unported, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.*

jects, surfaces, or deformable elements as primary con-trol structures. These systems often improve accessibility and expressivity by grounding sound control in bodily action. However, many tangible systems rely on discrete sensors or predefined gestural vocabularies, which may constrain exploratory interaction.

1.2 Material-Based Interaction

Material-based interfaces extend tangible interaction by emphasizing the expressive affordances of physical mate-rials themselves. Prior work has explored deformable surfaces, textiles, clay, and fluid media as sound controllers, highlighting their capacity for ambiguity, continuous transformation, and non-symbolic interaction. Such systems often prioritize exploratory engagement over precision, aligning with research that values provisionality and embodied sense-making in musical practice.

Sand represents an extreme case of a continuously deformable material. Its surface geometry is unstable, resolution varies with compaction, and interactions resist segmentation into discrete gestures. Rather than treating these characteristics as limitations, the Acoustic Interactive Sand Tray System embraces them as expressive af-fordances.

1.3 Mapping Strategies in Computer Music

Mapping, the relationship between user action and sonic response, has been identified as a core design problem in digital musical instruments. Hunt, Wanderley, and Para-dis formalized mapping as a critical determinant of expressivity, arguing that perceptual transparency and responsiveness are essential for meaningful interaction [5]. Wessel and Wright further emphasized the importance of "intimate control," in which small physical actions pro-duce perceptually significant sonic results.

Subsequent research has highlighted the effectiveness of feature-based and many-to-many mappings for supporting exploratory interaction, particularly in systems where precise gesture recognition is neither feasible nor desirable [6]. The present system adopts this approach by ex-tracting perceptually salient spatial features from material interaction and mapping them to continuous sonic trans-formations.

2. SYSTEM DESIGN AND IMPLEMENTATION

2.1 System Overview

The Acoustic Interactive Sand Tray System consists of four primary components:(see fig 1)
(1) Physical Interface: A shallow sand tray containing fine-grain sand and a curated set of small physical objects.
(2) Sensing Layer: A depth-sensing camera (e.g., Orbbec Astra 2) mounted above the tray captures real-time surface geometry. The depth-sensing part is written in C and processed data is sent to Max/MSP and Unity through OSC. A neural network–based object detection module identifies selected physical objects placed within the tray. The object detection layer is based on the open-source neural network framework written in C and CUDA by Ali Farhadi called Darknet and the object detection algorithm you only look once (YOLO v11) [7](see fig 2). . This is particularly critical for the Sand Tray system to maintain a high frame rate, ensuring that the latency between physical object placement and auditory response remains below the threshold of perceptual lag.

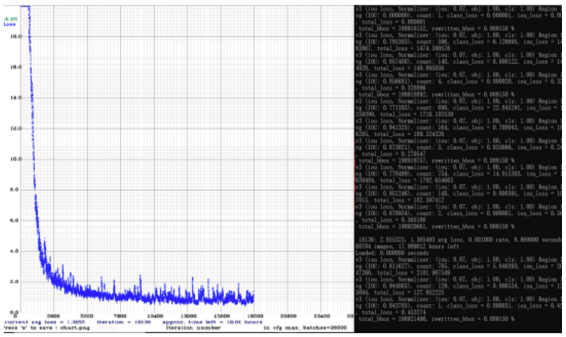
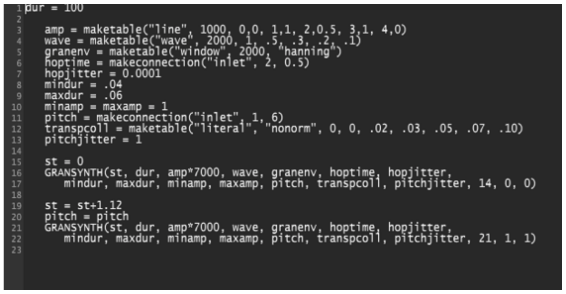


Figure 2: Training the object detector using Darknet through command line. For the chart on the left, the blue line shows the current average loss (which could be considered the current accuracy of the object detector). The average loss number going lower means a better result.

(3) Feature Extraction Layer: Depth data is processed to extract continuous surface features, while object categories are encoded as discrete control parameters.
(4) Sound Engine: Extracted features are mapped to sound synthesis and transformation processes implement-ed in Max/MSP and Unity, producing immediate auditory feedback.

RTcmix and the cloud-based WebRTcmix API has been used in the Sound Engine, taking parameters from the former layers and mapping them to the sounds.(see fig 3)



```
dur = 100

amp = maketable("line", 1000, 0,0, 1,1, 2,0.5, 3,1, 4,0)
wave = maketable("wave", 2000, 1, .5, .3, .2, .1)
granenv = maketable("window", 2000, "hanning")
hoptime = makeconnection("inlet", 2, 0.5)
hopjitter = 0.0001
mindur = .04
maxdur = .06
minamp = maxamp = 1
pitch = makeconnection("inlet", 1, 6)
transpcoll = maketable("literal", "nonorm", 0, 0, .02, .03, .05, .07, .10)
pitchjitter = 1

st = 0
GRANSYNTH(st, dur, amp*7000, wave, granenv, hoptime, hopjitter,
   mindur, maxdur, minamp, maxamp, pitch, transpcoll, pitchjitter, 14, 0, 0)

st = st+1.12
pitch = pitch
GRANSYNTH(st, dur, amp*7000, wave, granenv, hoptime, hopjitter,
   mindur, maxdur, minamp, maxamp, pitch, transpcoll, pitchjitter, 21, 1, 1)
```

Figure 3: RTcmix code for taking parameters from the object detector and generating sound.

2.2 Surface Contour Features

Two primary surface features are extracted:
(1) Surface Flatness: Computed as variance in depth values across the tray, capturing the overall smoothness or turbulence of the surface.(see fig 4)
(2) Central Elevation: Tracks the relative height of the central region, enabling detection of peaks and depressions.

These features are intentionally coarse-grained, prioritizing perceptual salience over fine spatial precision.

2.3 Mapping Strategies

Using object detection techniques(see fig 5), the system recognizes predefined physical objects. Each object cate-

gory functions as a high-level control token, enabling shifts in sound behavior or synthesis topology without requiring explicit user commands.

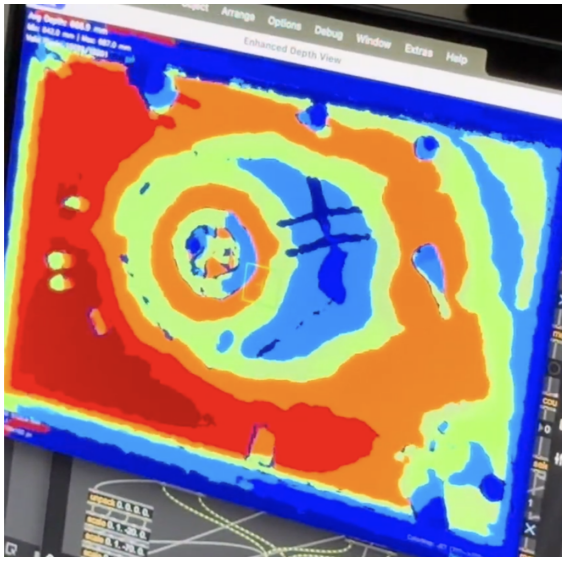


Figure 4: Visualization example using the surface flatness information.

Mapping strategies were designed to emphasize immediacy, perceptual coherence, and exploratory affordance:
(1) Surface Flatness → Timbral Density: Smoother surfaces produce stable and sustained textures, while irregular surfaces introduce spectral modulation and complexity.
(2) Central Elevation → Harmonic or Spatial Focus: Raised central features increase harmonic intensity or spatial prominence, while depressions reduce presence or diffuse sound outward.
(3) Object Categories → Sound Worlds: Detected objects select or modify synthesis processes, enabling transitions between contrasting sonic behaviors.

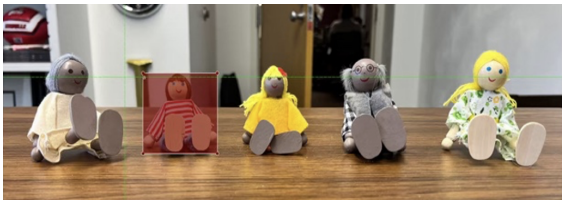


Figure 5: One of the object detection examples.

Unlike many sensor-based musical interfaces that prioritize discrete gesture recognition, this system intentionally embraces material ambiguity, using perceptually grounded surface features to support continuous and exploratory sound interaction.Table 1 shows Some examples of how parameters control sounds.

3. PRELIMINARY EVALUATION AND USE CASES

The system has been tested in exploratory sessions with university students, composers, and workshop participants.

Table 1: Examples of how parameters control sounds

Sound Category	Control Parameter	Mapping Logic
Discrete Events (e.g., Apple bite)	YOLO v11 Confidence Score	Trigger when possibility is $>$ 90%
Ambient Texture (e.g., Drone)	Global Surface Flatness	Continuous modulation of Gain and Cutoff
Harmonic Content	Regional Central Elevation	Spectral centroid shift based on peak height
Composite Scenes	Object Count / Density	Global synthesis topology transformation

Users typically establish causal relationships be-tween material manipulation and sonic response without formal instruction, engaging through shaping, smoothing, and spatial exploration rather than goal-oriented performance.(see fig 6)



Figure 6: Facilitated exploratory sessions using the Acoustic Interactive Sand Tray System in a university setting.

In facilitated sessions conducted in collaboration with a certified music therapist, the system functioned as an open-ended sound-making environment rather than a prescriptive tool. These sessions served as use cases for evaluating interaction robustness and mapping clarity, rather than assessments of therapeutic outcomes.

4. DISCUSSION AND LIMITATIONS

From the perspective of musical interface technology design [8], the system occupies a region of the design space that emphasizes embodied material interaction and perceptual coherence, rather than high-resolution gesture recog-

nition or symbolic control. The Acoustic Interactive Sand Tray System demonstrates how continuously deformable materials can function as expressive interfaces for sound interaction. By prioritizing material affordances and perceptually meaningful features, the system supports exploratory engagement without requiring musical train-ing or symbolic control.

Limitations include sensitivity constraints imposed by sand's physical properties and depth sensing resolution. Fine-grain gestures may be obscured, and mapping parameters require careful calibration to balance responsive-ness with sonic stability.

In addition, the current implementation relies on a single overhead depth-sensing camera. As a result, occlusion may occur when users lean forward and temporarily block the camera's field of view, leading to intermittent loss of surface data.

Future iterations may address this limitation through multi-camera systems, multi-modal sensing, or adaptive mapping strategies to enhance robustness and expressivity.

5. CONCLUSION AND FUTURE WORK

This paper presented the Acoustic Interactive Sand Tray System, an embodied interactive sound system that translates material interaction into real-time sonic trans-formation. By integrating depth sensing, feature extraction, and perceptually grounded mapping strategies, the system proposes an alternative paradigm for musical interaction centered on continuous engagement with a natural material.

From a music technology perspective, the system demonstrates how continuously deformable materials can function as expressive interfaces without relying on discrete gesture recognition or symbolic control. The design emphasizes perceptual coherence, exploratory affordance, and low cognitive overhead, aligning with current re-search in embodied and tangible musical interfaces.

By lowering the barrier to expressive sound creation, the system not only serves artistic purposes but also opens new avenues for inclusive musical participation in therapeutic and community contexts.

Future work will explore expanded sensing modalities, including multi-camera systems configurations and multimodal sensor fusion, as well as adaptive mapping strategies that respond dynamically to user behavior. Formal user studies will further investigate interaction patterns and expressive potential across compositional, education-al, and experimental contexts. Beyond its current implementation, the system suggests broader possibilities for incorporating natural materials into computer music inter-face design.

6. REFERENCES

[1] C. Cadoz and M. M. Wanderley, "Gesture–Music," in *Trends in Gestural Control of Music*. IRCAM, 2000.

[2] M. M. Wanderley and M. Battier, Eds., *Trends in Gestural Control of Music*. IRCAM, 2000.

[3] D. Wessel and M. Wright, "Problems and prospects for intimate musical control of computers," *Computer Music Journal*, vol. 26, no. 3, pp. 11–22, 2002.

[4] A. Tanaka, "Musical performance practice on sensor-based instruments," in *Trends in Gestural Control of Music*, M. M. Wanderley and M. Battier, Eds. Paris, France: IRCAM, 2000, pp. 389–405.

[5] A. Hunt, M. M. Wanderley, and M. Paradis, "The importance of parameter mapping in electronic instrument design," *Journal of New Music Research*, vol. 32, no. 4, pp. 429–440, 2003.

[6] S. Fels, A. Gadd, and A. Mulder, "Mapping transparency and expressiveness," *Organised Sound*, vol. 7, no. 2, pp. 205–214, 2002.

[7] G. Jocher and J. Qiu, "YOLO11: Real-Time Object Detection and Beyond," *arXiv preprint arXiv:2410.17725*, 2024.

[8] D. Overholt, "The musical interface technology design space," *Organised Sound*, vol. 16, no. 2, pp. 217–226, 2011.

The Singing Skin: An Audience-Centered Biofeedback System for Musical Interaction Based on Galvanic Skin Response

Eun Ji Oh, Jun Woo Beck, and Alexandria Smith
Georgia Institute of Technology
{eoh61, jbeck85, alexandria.smith}@gatech.edu

ABSTRACT

Music can evoke measurable physiological responses, yet these responses have been predominantly explored from the performer's perspective in interactive and biofeedback-based music systems. In contrast, the sonification of audience physiology remains relatively underexplored in live music contexts. We present The Singing Skin, a real-time, audience-centered biofeedback system for live performance that integrates listeners' physiological responses into musical control. The system measures galvanic skin response (GSR) and uses the phasic component of the GSR signal as an index of moment-to-moment audience engagement. This phasic GSR–based control signal is normalized and mapped to the rhythmic subdivision of a monophonic lead line generated by a wavetable synthesizer. Rather than directly modifying tempo or pitch, the control signal modulates the cutoff rate of a low-pass filter, producing an indirect pacing effect that influences perceived musical drive and energy. The system is demonstrated in a live performance setting involving a violinist and a listener equipped with GSR sensors. This work contributes a novel approach to audience-inclusive musical interaction by extending audience physiology as an active control source in live music performance.

1. INTRODUCTION

Music has the strong power to evoke emotional and arousal-related processes, which can be measured through physiological signals. Over the years, several artists have explored innovative ways to make physiology audible. Early biofeedback works, such as Alvin Lucier's *Music for Solo Performer* [1] and Richard Teitelbaum's synthesizer experiments [2], utilized performers' biosignals as control sources for sound production. David Rosenboom's *On Being Invisible* further advanced this approach by sonifying the performer's electroencephalography (EEG) alpha activity [3], while later projects, like Pamela Z's *BodySynth* [4], expanded performer-centered mappings using electromyography (EMG) data. Collectively, this body of work has primarily focused on biosignals from the performer's perspective.

In contrast, *audience*-centered biofeedback remains relatively underexplored in the realm of computer music, despite the inherently interactive nature of live performance. We therefore propose *The Singing Skin*, a novel audience-centered biofeedback system for live music that incorporates listeners' physiology directly into the performance. Our focus is on galvanic skin response (GSR), specifically its phasic component, as an indicator of moment-to-moment audience engagement. The system establishes a real-time feedback loop in which both performers and audiences can perceive and respond to each other's physiological states. We demonstrate this framework through a live performance, offering an exploratory performer–audience biofeedback framework that positions audience physiology as a primary musical control element and provides a basis for the implementation of audience-inclusive biofeedback systems.

Copyright: ©2026 Eun Ji Oh et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License 3.0 Unported, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

2. BACKGROUND

2.1 Physiological Signals

Physiological signals are involuntary bodily responses that arise when an individual is exposed to external stimuli or experiences fluctuations in internal states. These unconscious responses are regulated by the autonomic nervous system (ANS), which maintains homeostasis in the human body and supports functions (e.g., stress regulation and blood pressure control) [5]. The ANS is composed of two complementary branches: the sympathetic nervous system (SNS) and the parasympathetic nervous system (PNS). The SNS and PNS work together to regulate bodily function, but generally operate in opposing ways. The SNS is associated with "fight-or-flight" responses [6], preparing the body for action during emergency situations or physical exertion by increasing blood flow, heart rate, and muscle activity. In contrast, the PNS is activated during quiet, resting conditions to conserve energy and restore bodily balance, promoting relaxation by decreasing muscle activity and reducing cardiovascular activation.

Various physiological signatures reflecting ANS activity can be measured through different biosignals, including electrocardiography (ECG; German: *Elektrokardiogramm*, EKG), electromyography (EMG), galvanic skin responses (GSR; also referred to as electrodermal activity, EDA), and electroencephalography (EEG). Each biosignal provides distinct indicators of SNS or PNS activation. For instance, heart rate (HR) and heart rate variability (HRV) are the indicators commonly derived from ECG signals. Increased HR and lower HRV are typically associated with heightened arousal and SNS activation, whereas decreased HR and higher HRV reflect PNS dominance [7, 8]. In the case of GSR signals, both tonic skin conductance level (SCL) and phasic skin conductance responses (SCR) are widely

regarded as direct indicators of SNS activation. SCL represents baseline arousal levels, whereas SCR reflects rapid, stimulus-evoked changes in sympathetic activity [7]. The choice of physiological indicators in the research varies depending on the goals of the study and the specific facets of autonomic functioning that researchers seek to examine.

2.2 Biofeedback Systems in Music

Music is closely linked to emotional and arousal-related processes. The link between physiological changes during musical activities and emotional arousal is well-established by empirical evidence [9, 7, 10, 11]. Based on this connection, various artistic and technological approaches have aimed to incorporate physiological responses within creative and performative contexts. A pioneering example is Alvin Lucier's *Music for Solo Performer* (1965), in which the composer utilized real-time EEG brain signals from the performer to activate percussion instruments on stage, thereby transforming internal physiological activity into musical expression [1]. Similarly, Richard Teitelbaum explored the use of biosignals to manipulate electronic synthesizers in the late 1960s [2]. In these early systems, biosignals primarily served to extend the performer's intentional control, with physiological data used as an additional means of instrument control during performance.

Building on this foundation, David Rosenboom introduced *On Being Invisible* (1975), which shifted the focus from controlling instruments to explicitly revealing the performer's internal states through sound using biosignals [3, 12, 13]. In this work, the performer's EEG alpha-band activity was sonified as a sustained tone [12, 14]. Because alpha-band brain activity is commonly associated with relaxed or meditative states, this sonification made aspects of the performer's internal cognitive state perceptible through sound. Accordingly, Rosenboom approached physiological activity not as a control signal, but as a core sound material shaping the musical work [15, 16]. A similar approach appears in Pamela Z's *BodySynth* [4], a performative interface in which the performer's muscle activity was used to manipulate electronic sound sources. Wearable sensors attached to the performer's body captured muscle activity and mapped it to Musical Instrument Digital Interface (MIDI) control signals, enabling real-time control of samplers and synthesizers [17].

These biofeedback-based music systems have focused on the performer, positioning biosignals as either control inputs or sonic material produced by the performer. In comparison, *audience* physiology has received relatively little attention in performance settings. However, musical outcomes in live performance are continuously shaped by interactions between performers and audiences. Accordingly, audience physiological responses can be understood not only as measures of reception, but also as active elements within an interactive feedback loop, through which performers and audiences mutually shape the sonic and formal structure of a musical work.

2.3 Audience-Centric Biofeedback Systems

To the best of our knowledge, audience biosignals have rarely been systematically integrated into music performance contexts. Nevertheless, several related explorations in performance art and dance have incorporated audience physiology as a means of fostering reciprocal engagement between the stage and spectators. An early example is Ruth Anderson's *Centering* (1979), which brought collective GSR signals from the audience into the structure of a live dance performance [18]. In this work, GSR signals from four observers were sonified as pitch height while they viewed the performance. This design formed an interdependent loop in which audience responses influenced the sonic outcome and, in turn, shaped performers' reactions [1] .

More recently, the *Boiling Mind* project (2021) employed real-time HRV and EDA data from the audience to generate audiovisual elements that shaped the performance environment in a live dance context [19]. Audience physiological responses were used as staging elements, influencing sound and visual components that interacted with the choreographers. Subsequent analysis of the performance revealed a relationship between increased HRV—reflecting PNS activation—and event-related phasic peaks in EDA associated with moments of heightened audience engagement and musical crescendos [20].

Related approaches have also appeared in installation-based works. For example, *Phantom Undulations* (2023) [21] incorporated biodata from both artists and visitors as musical elements, repositioning visitors from passive observers to active contributors in the creation of the artistic work. In this installation, artists' respiration, heartbeat, EDA, and HRV were sonified as timbre, duration, pitch, loudness, and sonic texture, respectively, while visitors' physiological responses, such as EDA and heartbeat, were also incorporated into the evolving musical artifact.

Building on these precedents, we propose a biofeedback system for music performance that integrates audience physiology as an integral component of the musical work itself. Grounded in collaboration across psychophysiology, music cognition, and computer music, our system transforms listeners' real-time physiological responses into an additional musical layer that is audible to both performers and audiences. We adopt a physiologically meaningful indicator informed by empirical findings on musical engagement during music listening. The indicator is mapped to rhythmic modulation—a musical dimension shown to shape perceived drive and temporal engagement in music—allowing physiological dynamics to function as an audible musical process within performance. The proposed system operates as an augmentation of performance texture by modulating expressive musical parameters within a fixed jazz standard, rather than generating novel musical material or dictating structural shifts. To evaluate this framework, we conducted an exploratory case study involving a violinist and a single listener equipped with GSR sensors. Through this design, the system establishes a performer-audience feedback loop in which performers respond musically to the audience's physiological state, while the audience, in turn, responds to the unfolding musical performance.

This work offers a conceptual contribution by demonstrating how empirically grounded insights from music cognition and psychophysiology can be translated into musically meaningful, real-time interactive systems within computer music practice. By transforming largely implicit and un-

[1] https://www.encyclopedia.com/arts/dictionaries-thesauruses-pictures-and-press-releases/anderson-evelyn-ruth

conscious physiological responses into audible musical processes, the proposed approach enables both performers and co-present listeners to intuitively perceive and interpret each other's affective and physiological engagement with the music. This initial feedback loop in the current study provides a scalable foundation for future research aimed at integrating collective audience biosignals into a unified, shared musical response. Additionally, while the present work focuses on GSR-based modulation of rhythmic structure as one instantiation, the approach is not limited to this mapping and can be extended to other empirically validated physiological indicators to control a wide range of musical parameters (e.g., pitch or loudness) within interactive biofeedback music systems. Moreover, beyond concert performance, such biofeedback systems may also support communication in broader music-centered interaction contexts—including music therapy and education—by making otherwise unobservable internal responses perceptible and shareable through sound during musical engagement.

3. SYSTEM DEVELOPMENT

3.1 Physiological Indicator for Musical Engagement

In designing the proposed system to support a performer-audience feedback loop, we focus on physiological signals capable of capturing moment-to-moment changes in audience engagement. Following prior audience-centered performance works, we employ galvanic skin response (GSR) in our system. Within the GSR signal, we make a distinct design choice to focus specifically on a phasic component, which we use as a skin conductance response (SCR)-related indicator of musical engagement.

This design choice is further supported by empirical studies in music cognition. Prior research has shown that heightened musical arousal, as reflected in acoustic features such as tempo, loudness, and timbral brightness, is associated with increased autonomic nervous system (ANS) responses in listeners, including elevated SCR. For example, faster or accelerating tempi [22, 23], increased sound intensity [24], and brighter timbres [25] have been shown to correlate with greater SCR values. Moreover, studies examining the relationship between physiological responses, musical arousal, and listener engagement in live music concert have found that SCR closely tracks sympathetic arousal and is linked to higher self-reported engagement, with faster-tempo passages consistently eliciting elevated SCR values [26, 27].

Accordingly, our system maps a phasic GSR–based control signal, representing SCR-related activity, to MIDI continuous controller (CC) values that are intended to reflect relative changes in arousal in a musically perceptible manner. Increases in this control signal result in shorter note durations in a monophonic lead synthesizer, indirectly conveying a sense of increased pace and drive without explicitly altering tempo. Using this phasic GSR signal as a real-time modulator thus enables both performers and listeners to perceive changes in audience engagement through expressive variations in the melodic line. In this way, audience engagement can be incorporated as an active, real-time component of the performance structure.

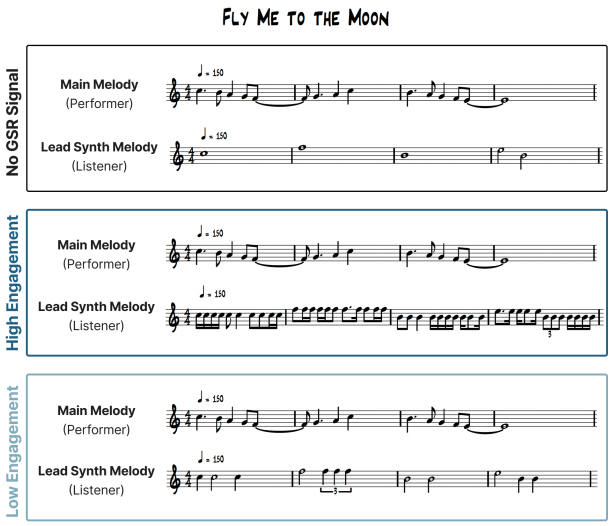


Figure 1. Musical design example illustrating three output conditions of the lead synthesizer. The top panel shows the default setting without an audience control signal, while the middle and bottom panels represent high and low audience engagement, respectively, expressed through increased or reduced rhythmic subdivision at a fixed tempo.

3.2 Musical Design

The musical design of the proposed system aims to make audience engagement perceptible through a musically coherent and interpretable manner. In prior research, musical engagement has been shown to be closely associated with changes in arousal, particularly via listeners' perception of musical pace and drive, which is strongly influenced by tempo. However, directly modifying tempo during live performance can be musically disruptive and may interfere with the performer's expressive control and phrasing. Rather than altering global tempo, we therefore adopt an indirect but musically meaningful strategy to convey changes in perceived pace by manipulating the rhythmic subdivision of a melodic line. For example, higher arousal levels are expressed through shorter note durations, while lower arousal levels result in more sustained melodic gestures. An example output of this musical design is shown in Figure 1.

This approach allows changes in engagement to be perceived as variations in rhythmic density and momentum, while preserving the notated tempo and the performer's intentional timing. By treating physiological influence as an expressive cue rather than an explicit command, the system supports musical interpretation. To implement this design, the melodic line is generated using a wavetable synthesizer (Serum, Xfer) [28], whose low-pass filter cutoff frequency rate is modulated by a phasic GSR–based control signal. As the magnitude of this control signal increases, the rate of the Low-Frequency Oscillator (LFO) modulating the filter cutoff frequency increases accordingly. This modulation perceptually subdivides a sustained note into multiple shorter articulations, creating the impression of increased rhythmic activity without altering pitch or tempo.

We chose this synthesis-based approach over direct note triggering or tempo modulation because it enables smooth, continuous transitions between states of engagement. As a result, higher levels of audience engagement are conveyed musically through more rapid rhythmic subdivision. This

design allows physiological arousal to shape perceived pacing while remaining embedded within a musically coherent structure.

3.3 Overall System Architecture

The system architecture is divided into two primary computational units (see Figure 2): a data acquisition and processing unit (PC 1) and a sound generation unit (PC 2). The signal flow begins with GSR signals from the music listener, captured by an amplifier and transmitted to PC 1.

In the PC 1 environment, GSR signals are streamed into a Python-based processing pipeline via the Lab Streaming Layer (LSL) [29]. This pipeline consists of three sequential stages: (1) GSR pre-processing to obtain a phasic-dominant signal, (2) listener-specific calibration and sample-wise min—max normalization, and (3) extraction of a musically anchored feature using a sliding window.

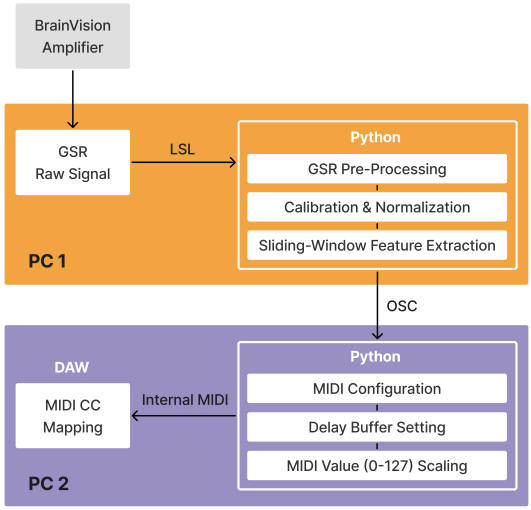


Figure 2. System architecture and signal flow of the proposed audience-centered biofeedback system for music performance. GSR: galvanic skin responses; LSL: Lab Streaming Layer; OSC: Open Sound Control; MIDI: Musical Instrument Digital Interface; CC: continuous controller; DAW: Digital Audio Workstation.

The normalized control signal is then transmitted to PC 2 via Open Sound Control (OSC) [30]. On PC 2, a second Python environment translates the physiological data into musical control parameters. This stage includes (1) setting a delay buffer to align physiological feedback with musical phrasing, (2) MIDI configuration, and (3) scaling the normalized control signal into MIDI continuous controller (CC) messages. Finally, the control data are sent via an internal MIDI connection to the digital audio workstation (DAW), where MIDI mappings modulate synthesizer parameters in real time.

3.4 Biosignal Acquisition and Processing

3.4.1 Acquisition

GSR was recorded using a BrainVision *actiCHamp* amplifier (Brain Products GmbH; see Figure 3). Using the BrainVision LSL Connector free software, the raw GSR stream was forwarded in real time to the performance computer (PC 1) via the LSL at a sampling rate of 10,000 Hz. This sampling rate corresponds to the minimum rate selectable in the BrainVision LSL Connector and was used for acquisition prior to downstream filtering and resampling. The same PC 1 ran a Python environment for online processing, ensuring low-latency transfer and a unified time base.

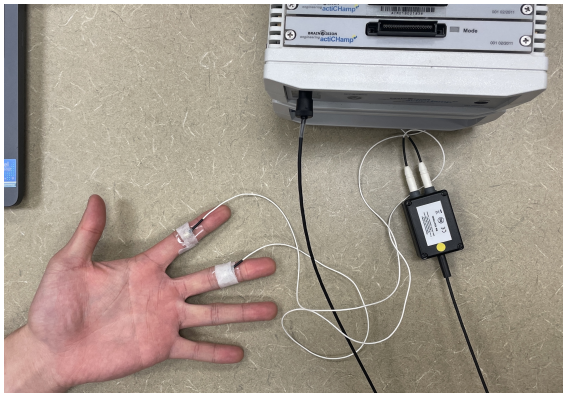


Figure 3. Setup of galvanic skin response (GSR) recording using BrainVision Ag/AgCl electrodes attached to the listener's hand.

3.4.2 Real-Time Preprocessing

In this work, the phasic component of the GSR signal is used as an index of skin conductance responses (SCR)–related activity. To emphasize this component, the incoming GSR stream was filtered online using a fourth-order Butterworth band-pass filter (0.01–3.0 Hz), implemented as a causal filter with second-order sections (SOS) (`sosfilt` in SciPy [31]). This filtering attenuates very slow tonic fluctuations as well as high-frequency noise, yielding a phasic-dominant GSR signal suitable for real-time control. The filtered signal was subsequently downsampled to 20 Hz, which provides sufficient temporal resolution for SCR-related dynamics while reducing computational load.

3.4.3 Calibration and Normalization

At the start of each run, 30 s of baseline data were collected to establish a listener- and session-specific operating range. Based on this calibration window, minimum and maximum values were estimated using percentile statistics, and subsequent samples of the phasic-dominant GSR signal were linearly mapped to the [0,1] range via min–max normalization. This sample-wise normalization yields a bounded control signal that is robust to inter-individual differences and facilitates stable inter-process communication (e.g., OSC transmission to PC 2).

3.4.4 Feature Extraction for Musical Engagement

Rather than performing model-based SCR decomposition or discrete event detection, we estimate SCR-related activity by computing a musically anchored control feature from the normalized, phasic-dominant GSR signal. Specifically, a sliding mean statistic is applied over a fixed analysis window defined in musical time.

Prior to performance, the tempo (beats per minute; BPM) is specified by the user. The analysis window spans four bars at the current tempo—a phrase-scale unit commonly used in musical practice—while the hop size corresponds to

a thirty-second note (32nd note). Expressing both window and hop sizes in musical units, rather than absolute time, ensures that the resulting control feature remains aligned with musical phrasing and automatically rescales with tempo changes. At the 20 Hz analysis rate (50 ms per sample), hop updates are quantized to the analysis grid. When the duration of a 32nd note falls below 50 ms at higher tempi, the hop size is clamped to a single analysis sample. The resulting windowed control signal provides a smooth, tempo-aligned estimate of SCR-related activity, which is transmitted in real time to PC 2 via OSC for musical mapping.

3.5 Sound Generation

Although the proposed system is not tied to a specific musical genre, its sound generation design assumes the presence of an existing musical layer that can be continuously modified in real-time. This assumption allows audience physiology to be integrated into an ongoing performance without interrupting musical continuity. Accordingly, the system operates on pre-existing musical material, consisting of a pre-determined harmonic progression and a MIDI-based melodic line prepared in advance within a DAW.

The computed audience physiology values described in Section 3.4 are used as a MIDI control signal to represent moment-to-moment fluctuations in audience engagement as an explicit musical control parameter. Building on prior findings that SCR activity is closely associated with tempo-related musical features during heightened engagement, the system links physiological values indirectly to perceived rhythmic pace. Rather than directly altering the notated tempo, the system modulates rhythmic density by dynamically subdividing the rhythm of an existing melodic line. To realize this rhythmic modulation in a temporally coherent and perceptually meaningful manner, the following subsections describe the sound generation pipeline implemented on PC 2.

3.5.1 MIDI Synchronization

Before processing the incoming normalized phasic GSR–based control signal, the system establishes a synchronized temporal environment with the DAW. The Python script running on PC 2 listens for MIDI Transport messages (Start, Stop, and Clock) sent by the DAW. This configuration supports that the system's internal processing clock is synchronized with the tempo defined in the DAW. By conforming to the DAW's transport status, the system prevents temporal data drift and maintains musical coherence, ensuring that the physiological feedback loop begins and ends precisely with the musical performance.

3.5.2 Delay Buffer Setting

To account for the listener's perceptual response latency, a delay buffer is applied to the incoming control signal. Rather than mapping physiological changes to sound parameters instantaneously, the system is designed to reflect the listener's response to the musical phrase they have just heard. Accordingly, a sliding time window (i.e., delay buffer) is used to temporarily store the normalized phasic GSR–based control signal. Instead of reading the signal immediately, the system accesses the buffer with a fixed temporal offset corresponding to the desired delay duration. This design ensures that the current musical output is driven by the audience's physiological response to the preceding musical segment, aligning the feedback mechanism with perceptual and musical timescales.

3.5.3 Biosignal-to-MIDI Value Mapping

After temporal alignment through buffering, the delayed control signal is converted into discrete musical control values suitable for real-time synthesis. Specifically, the normalized control signal in the range $[0.0, 1.0]$ is mapped to MIDI values in the range $[0, 127]$ using a transfer function with an adjustable exponent:

$$y = x^{\gamma} \cdot 127,$$

where x denotes the normalized control signal, y the resulting MIDI value, and γ a sensitivity parameter that determines the curvature of the mapping function.

Setting $\gamma = 1.0$ yields a linear mapping, in which output values are directly proportional to the input. Values of $\gamma < 1.0$ produce a concave (logarithmic-like) curve that increases sensitivity to lower input values, whereas values of $\gamma > 1.0$ generate a convex (exponential-like) curve that reduces sensitivity in the lower range. The resulting continuous values are quantized into standard 7-bit integers to ensure compatibility with the MIDI protocol.

This flexible mapping function allows the system to accommodate inter-individual variability in physiological dynamics. For listeners exhibiting a narrow range of control signal fluctuations, smaller values of γ expand subtle variations into a broader range of MIDI outputs, preventing the musical response from sounding static. Conversely, for listeners with a wider dynamic range, larger γ values help prevent saturation at the upper end of the MIDI scale, preserving musically usable control resolution.

3.5.4 Transmission to MIDI Port

Following value mapping, the quantized MIDI control values are encapsulated into standard MIDI CC messages. Each message is assigned to a predefined MIDI channel and CC number specified within the script. The resulting data stream is transmitted via a virtual internal MIDI port (e.g., IAC Driver or loopMIDI) to the external sound generation environment.

3.5.5 Sound Modulation Strategy

The transmitted MIDI control messages serve as an intermediate representation that enables routing of the audience physiology–derived control signal to synthesis parameters within the sound engine (see Figure 4). In the present system, this control stream modulates the rate of an LFO.

The LFO is routed to modulate the cutoff frequency of a low-pass filter applied to the lead synthesizer voice. As the LFO oscillates, it periodically reveals and masks high-frequency transients that define the articulation of rhythmic pulses. The MIDI control values directly determine the frequency of the LFO oscillation, thereby shaping the perceived rhythmic subdivision of the sound.

This design results in the following musical behaviors:

- **High Engagement (Fast LFO Rate):** As the control value increases, the LFO rate accelerates, leading to more frequent articulations and a denser rhythmic

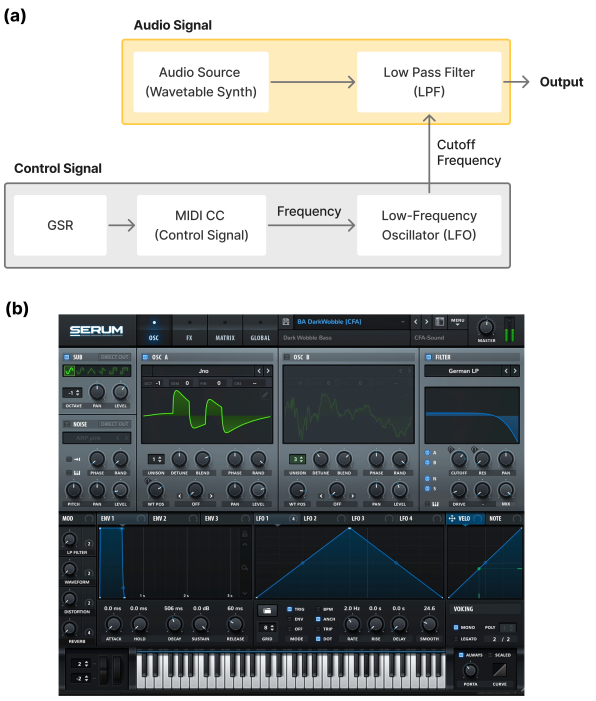


Figure 4. (a) Structure of the lead synthesizer used as the sound engine. (b) Screenshot of the Serum wavetable synthesizer (Xfer).

subdivision. This is perceived as increased musical drive and energy.

- **Low Engagement (Slow LFO Rate):** When the control value is low, the LFO modulates the filter at a slower rate. This produces fewer articulations and a more sustained rhythm, perceived as a relaxed musical pace.

Through this mechanism, variations in audience engagement are translated into changes in perceived musical pace without altering the notated tempo or pitch. This synthesis-based approach was chosen over direct note triggering or tempo modulation because it enables smooth and continuous transitions between engagement states. As a result, physiological arousal shapes the perceived pacing of the music while remaining embedded within a musically coherent and performer-controlled structure.

4. LIVE PERFORMANCE DEMONSTRATION

4.1 Music Selection

To demonstrate the system in a musical context, the jazz standard *Fly Me to the Moon* was selected. A jazz standard was selected to leverage the improvisational flexibility of the genre. The nature of jazz improvisation—operating within established harmonic and rhythmic rules while allowing for spontaneous variation—is an ideal format for reflecting listener-driven modulations. This framework allowed the performer to respond spontaneously to the evolving lead line. It facilitated a dynamic musical interplay, avoiding the conflict that often arises with rigid, pre-determined structures. Furthermore, this flexibility ensures that the listener's physiological feedback is integrated into the performance in a way that remains musically coherent, minimizing the risk of the system generating output that conflicts with the harmonic or rhythmic context.

4.2 Performance Setup

The performance setup involved one violinist and one listener. The performance structure comprised three components: a physiology-driven lead line, a live violin improvisation, and a backing track. For the lead synthesizer line, a pre-sequenced MIDI clip was provided, and the audience's physiological feedback modulates the rhythmic subdivision by modulating the rate of a Low Frequency Oscillator (LFO), which in turn controls the filter cutoff frequency. Over the backing track, a violin performed a melody and improvisation.

Galvanic skin response (GSR) sensors were attached to the fingers of the listener's non-dominant hand to minimize motion artifacts. A 30-second GSR baseline was then recorded to establish the listener's physiological range for normalization. Following this calibration, the normalized phasic GSR–based control signal was transmitted from PC 1 to PC 2 to verify that the Open Sound Control (OSC) connection and data scaling were functioning correctly.

Both the listener and the violin performer monitored the sound through separate headphones to ensure real-time perception of the modulated physiological feedback. The acoustic violin signal was captured using a microphone and routed through an audio interface (PreSonus AudioBox iOne) to the digital audio workstation (DAW), where it was processed and recorded using Pro Tools (Avid) [32]. All audio signals, including headphone monitoring and microphone input, were transmitted via wired connections, with the audio interface connected to PC 2 via USB.

4.3 Performance and Recording

Before the session, the listener was instructed to listen comfortably, while the violinist was asked to improvise freely in musical response to the evolving synthesizer. The tempo was set to 150 BPM according to the performer's preference, and the biosignal processing script was initialized prior to the performance. The session started with the activation of the DAW transport, which simultaneously triggered the backing track playback and the recording systems.

Once the backing track began, the violinist performed an improvisation lasting approximately three minutes. After an initial four-bar delay, the physiology-driven lead line started to modulate its rhythmic subdivision based on the listener's phasic GSR–based control signal. The four-bar delay was implemented to align the physiological response with the musical structure. During the performance, both the violinist and the listener were able to hear the physiological feedback in real-time. This auditory loop allowed the violinist to respond musically to the listener-driven modulation, while the listener simultaneously perceived changes in the sound corresponding to their own physiological responses.

For subsequent data analysis, both the acoustic violin audio and the MIDI CC automation data were recorded simultaneously in the DAW, while the normalized phasic GSR–based control signal was saved as a `.csv` file. Figure 5 presents the real-time normalized phasic GSR signals captured during the system demonstration.

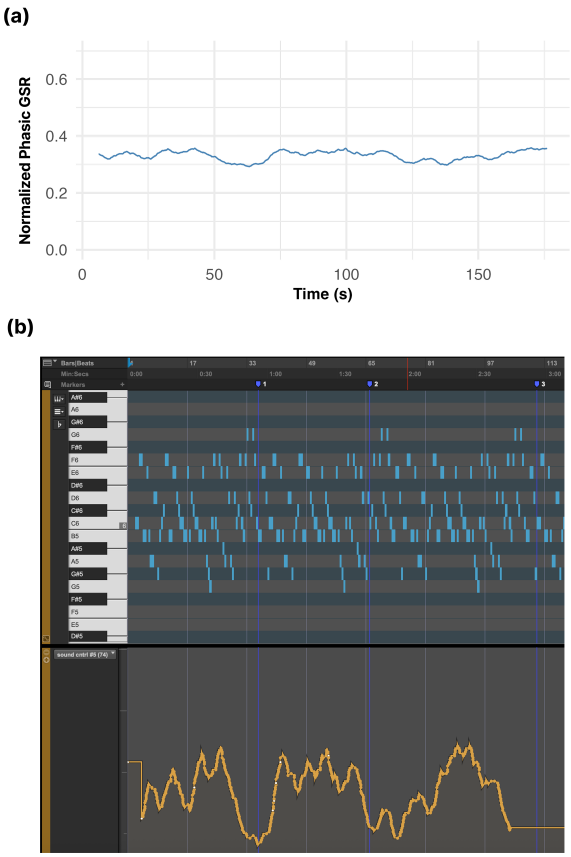


Figure 5. Real-time audience physiology–derived control signals during the system demonstration. (a) Min–max normalized phasic-dominant GSR signal over the full musical excerpt (time in seconds). (b) Digital audio workstation (DAW) view illustrating MIDI continuous controller (CC) data derived from the control signal and mapped to the Low-Frequency Oscillator modulates the filter cutoff frequency of the lead synthesizer line during the initial segment of the performance.

5. CONCLUSION AND FUTURE WORK

This work introduced an audience-centered biofeedback musical system that incorporates listeners' physiological responses into a real-time performance setting. Unlike previous approaches that primarily relied on performer-generated biosignals, our system utilizes galvanic skin response (GSR), extracting phasic skin conductance response (SCR)–related activity and mapping its moment-to-moment fluctuations to musical parameters. Through a exploratory live music demonstration, we showed that this approach can support a functioning feedback loop between performers and audiences.

By transforming physiological data into musical expression, the proposed system suggests a shift in the listener's role from passive observer toward active participant and co-creator in the performance. Audience engagement, expressed through unconscious physiological responses, shapes the rhythmic and structural evolution of the music, blurring the conventional boundary between stage and audience. While the framework was demonstrated in a jazz improvisation context, the underlying system architecture is a scalable and may be adapted to other musical genres and performance settings, including scenarios involving collective audience interaction.

Despite these promising outcomes, several limitations remain to be addressed in future work. First, the normalization of GSR signals poses inherent challenges due to significant inter-individual physiological variability, which can affect the consistency and robustness of mapping these signals to musical elements. Second, the limited scale of the current demonstration—conducted as an exploratory case study with a single listener—highlights the need for larger-scale evaluations to validate the system's effectiveness across diverse audience groups. Furthermore, while the system enables real-time interaction, it currently functions by modulating expressive parameters within a fixed musical form rather than serving as a shared compositional agency that fundamentally alters the work's structural evolution.

Future research will focus on extending this framework toward collective biosignal integration and exploring more generative mappings that grant the audience a more formative role in the compositional process. One practical challenge is the requirement of a calibration period prior to the performance, which is necessary to establish an individual physiological baseline but may interrupt the flow of a concert setting. Additionally, further investigation is needed to assess how reliably SCR-related activity reflects momentary physiological responses to music. Future systems could incorporate additional biosignals, such as heart rate variability (HRV), respiration, or electroencephalography (EEG), to provide a more comprehensive representation of audience engagement. Finally, controlled user studies are needed to evaluate whether audience-centered biofeedback enhances engagement for both performers and listeners, and to better understand its influence on live musical interaction.

6. REFERENCES

[1] V. Straebel and W. Thoben, "Alvin Lucier's music for solo performer: experimental music beyond sonification," *Organised Sound*, vol. 19, no. 1, pp. 17–29, 2014.

[2] E. R. Miranda and A. Brouse, "Interfacing the brain directly with musical systems: on developing systems for making music with brain signals," *Leonardo*, vol. 38, no. 4, pp. 331–336, 2005.

[3] D. Rosenboom and D. Rosenboom, *On being invisible*. Music Gallery Editions, 1977.

[4] Z. Pamela and T. Sellar, "Parts of Speech," *Theater*, vol. 30, no. 2, pp. 58–65, 2000.

[5] L. K. McCorry, "Physiology of the autonomic nervous system," *American journal of pharmaceutical education*, vol. 71, no. 4, p. 78, 2007.

[6] W. B. Cannon, *The wisdom of the body*. Norton & Co., 1939.

[7] D. C. Fowles, "The three arousal model: Implications of Gray's two-factor learning theory for heart rate, electrodermal activity, and psychopathy," *Psychophysiology*, vol. 17, no. 2, pp. 87–104, 1980.

[8] T. F. o. t. E. S. o. C. t. N. A. S. o. P. Electrophysiology, "Heart rate variability: standards of measurement, physiological interpretation, and clinical use," *Circulation*, vol. 93, no. 5, pp. 1043–1065, 1996.

[9] V. N. Salimpoor, M. Benovoy, G. Longo, J. R. Cooperstock, and R. J. Zatorre, "The rewarding aspects of music listening are related to degree of emotional arousal," *PloS one*, vol. 4, no. 10, p. e7487, 2009.

[10] N. S. Rickard, "Intense emotional responses to music: A test of the physiological arousal hypothesis," *Psychology of music*, vol. 32, no. 4, pp. 371–388, 2004.

[11] G. H. Zimny and E. W. Weidenfeller, "Effects of music upon GSR and heart-rate," *The American journal of psychology*, vol. 76, no. 2, pp. 311–314, 1963.

[12] D. Rosenboom, "Method for producing sounds or light flashes with Alpha brain waves for artistic purposes," *Leonardo*, vol. 5, no. 2, p. 114, 1972.

[13] ——, "The Performing Brain," *Computer Music Journal*, vol. 14, no. 1, pp. 48–66, 1990.

[14] B. Lutters and P. J. Koehler, "Brainwaves in concert: the 20th century sonification of the electroencephalogram," *Brain*, vol. 139, no. 10, pp. 2809–2814, 2016.

[15] D. Rosenboom, "Extended Musical Interface with the Human Nervous System: Assessment and Prospectus," *Leonardo*, vol. 32, no. 4, pp. 257–257, 1999.

[16] ——, "A Program for the Development of Performance-Oriented Electronic Music Instrumentation in the Coming Decades: "What You Conceive Is What You Get"," *Perspectives of New Music*, vol. 25, no. 1/2, pp. 569–583, 1987.

[17] G. E. Lewis, "The virtual discourses of Pamela Z," *Journal of the Society for American Music*, vol. 1, no. 1, pp. 57–77, 2007.

[18] A. Lockwood. Hearing a Person—Remembering Ruth Anderson (1928-2019). NewMusicBox, New Music USA. [Online]. Available: https://newmusicusa.org/nmbx/hearing-a-person-remembering-ruth-anderson-1928-2019/

[19] M. Sugawa, T. Furukawa, G. Chernyshov, D. Hynds, J. Han, M. Padovani, D. Zheng, K. Marky, K. Kunze, and K. Minamizawa, "Boiling mind: Amplifying the audience-performer connection through sonification and visualization of heart and electrodermal activities," in *Proceedings of the Fifteenth International Conference on Tangible, Embedded, and Embodied Interaction*, 2021, pp. 1–10.

[20] J. Han, G. Chernyshov, M. Sugawa, D. Zheng, D. Hynds, T. Furukawa, M. Padovani Macieira, K. Marky, K. Minamizawa, J. A. Ward *et al.*, "Linking audience physiology to choreography," *ACM Transactions on Computer-Human Interaction*, vol. 30, no. 1, pp. 1–32, 2023.

[21] D. Hynds, D. Zheng, Y. Zhang, H. Ma, K. Ragozin, G. Chernyshov, J. A. Ward, T. Saito, K. Kunze, and K. Minamizawa, "Phantom Undulations: Remote Physiological Sensing in Abstract Installation Works," in *Proceedings of the Augmented Humans International Conference 2023*, 2023, pp. 367–370.

[22] P. Gomez and B. Danuser, "Relationships between musical structure and psychophysiological measures of emotion." *Emotion*, vol. 7, no. 2, p. 377, 2007.

[23] S. Khalfa, P. Isabelle, B. Jean-Pierre, and R. Manon, "Event-related skin conductance responses to musical emotions in humans," *Neuroscience letters*, vol. 328, no. 2, pp. 145–149, 2002.

[24] H. Egermann, N. Fernando, L. Chuen, and S. McAdams, "Music induces universal emotion-related psychophysiological responses: comparing Canadian listeners to Congolese Pygmies," *Frontiers in psychology*, vol. 5, p. 116059, 2015.

[25] K. N. Olsen and C. J. Stevens, "Psychophysiological response to acoustic intensity change in a musical chord," *Journal of Psychophysiology*, 2013.

[26] A. Czepiel, L. K. Fink, L. T. Fink, M. Wald-Fuhrmann, M. Tröndle, and J. Merrill, "Synchrony in the periphery: inter-subject correlation of physiological responses during live music concerts," *Scientific reports*, vol. 11, no. 1, p. 22457, 2021.

[27] W. Tschacher, S. Greenwood, H. Egermann, M. Wald-Fuhrmann, A. Czepiel, M. Tröndle, and D. Meier, "Physiological synchrony in audiences of live concerts." *Psychology of aesthetics, creativity, and the arts*, vol. 17, no. 2, p. 152, 2023.

[28] Xfer Records, "Serum: Advanced Wavetable Synthesizer," https://xferrecords.com/products/serum, 2014, accessed: 2025-12-19. [Online]. Available: https://xferrecords.com/products/serum

[29] C. Kothe, S. Y. Shirazi, T. Stenner, D. Medine, C. Boulay, M. I. Grivich, F. Artoni, T. Mullen, A. Delorme, and S. Makeig, "The Lab Streaming Layer for Synchronized Multimodal Recording," *Imaging Neuroscience*, vol. 3, p. IMAG.a.136, 2025, open Access. [Online]. Available: https://doi.org/10.1162/IMAG.a.136

[30] M. Wright, "Open Sound Control: an enabling technology for musical networking," *Organised Sound*, vol. 10, no. 3, pp. 193–200, 2005.

[31] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright *et al.*, "SciPy 1.0: fundamental algorithms for scientific computing in Python," *Nature methods*, vol. 17, no. 3, pp. 261–272, 2020.

[32] Avid Technology, "Pro Tools," 2025, accessed: 2025-12-19. [Online]. Available: https://www.avid.com/pro-tools

Emotion-Based Film Music Retrieval with Handcrafted and Deep Models

Lucas Ong
Queen Mary University of London
l.ong@se24.qmul.ac.uk

Ruby Olivia Nagano Crocker
Queen Mary University of London
r.o.n.crocker@qmul.ac.uk

George Fazekas
Queen Mary University of London
george.fazekas@qmul.ac.uk

ABSTRACT

Film music powerfully conveys emotion, yet computational methods for retrieving film tracks that match a target emotional state remain underexplored. This paper presents two approaches for emotion-based film music retrieval using Valence–Arousal (V–A) representations. The models are evaluated on the FME-24 dataset, which provides time-aligned participant-annotated V–A ratings for film music excerpts. The first approach applies k-Means to handcrafted audio features, while the second uses a VaDE model with contrastive learning to align audio and V–A embeddings. Results show that both methods capture emotion-related structure, with the deep model enabling more flexible, fine-grained selection.

1. INTRODUCTION

Music has a powerful ability to shape emotional experience, influencing perception, memory, and behaviour. In multimedia contexts such as film, music guides the audience's emotional interpretation of scenes and enhances narrative immersion [1]. Film music provides a naturalistic and contextually grounded stimulus for studying emotional responses, even in short excerpts, because it carries temporal and narrative cues (pacing, harmonic progression) that participants can mentally follow. In contrast, traditional laboratory settings often include constraints; simplified stimuli, decontextualized excerpts and unfamiliar surroundings which can limit engagement and ecological validity [2]. Selective musical choices can reliably communicate affect in otherwise ambiguous scenes, shaping not only the overall atmosphere but also the perceived emotions and relationships of characters [3]. Understanding and computationally modelling this relationship, specifically how musical features evoke and convey emotion, has become a central challenge within the field of Music Emotion Recognition (MER) [4].

While significant progress has been made in predicting emotion from audio features [5, 6, 7], comparatively less attention has been devoted to emotion-driven retrieval: the task of identifying music that aligns with a desired or target emotional state. This gap is particularly relevant in film, where composers and editors often need to locate emotionally congruent material rather than classify existing tracks. Furthermore, many V-Aailable film–music datasets remain limited in size, scope, and diversity, often relying on royalty-free or artificially composed samples that lack the expressive richness of commercial film soundtracks.

To address these limitations, this paper presents a system for emotion-based clustering and retrieval using the Film Music Emotion Dataset (FME-24) [8]. FME-24 contains excerpts from film soundtracks taken from a range of contemporary movies. Each excerpt is annotated on the Valence–Arousal (V-A) scale by crowd-sourced listeners, providing emotional ratings from non-experts that reflect natural, everyday responses to film music. Each excerpt includes time-stamped V-A scores and associated emotion sentence descriptors, providing a perceptually grounded representation of the emotional content. This dataset allows for both quantitative analysis and retrieval based on perceived emotion. Two computational models are evaluated using FME-24: (1) a feature-based k-Means approach leveraging handcrafted audio descriptors, and (2) a deep generative clustering model that integrates contrastive learning with 2-dimensional V-A embeddings, enabling emotion-informed retrieval of film music using emotional coordinates.

This paper advances emotion-based clustering and retrieval of film music by comparing two modelling approaches and introducing a framework that projects acoustic features into V-A space, enabling retrieval based on emotional content. The effectiveness of the proposed emotion-based clustering and retrieval methods was assessed using a combination of quantitative analysis and a perceptual listening test (4.2).

2. RELATED WORKS

Music has been shown to influence emotional states in therapeutic contexts, including palliative care, stress regulation, and support for dementia patients [9, 10]. Different musical styles can reliably evoke distinct responses: faster, rhythmically driven music tends to increase arousal, while slower or softer music creates calming effects. Music also produces measurable physiological effects, such as reductions in heart rate and cortisol levels [11, 12, 13]. These behavioural and physiological patterns highlight the potential for computational systems to support emotion-based music retrieval in applied contexts such as film and media.

In addition, the influence of music on real-world decisions has become increasingly apparent. Music streaming services such as Spotify [1] use mood and emotion-based recommendations (e.g., Discover Weekly playlists, mood-specific playlists) to guide listening behaviour, this illustrates how computational understanding of music and emotion can impact everyday media consumption [14, 15].

Copyright: ©2026 Lucas Ong et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License 3.0 Unported, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

[1] https://www.spotify.com

Within this context, Music Emotion Recognition (MER) has emerged as an active research area developing computational methods to identify or predict emotion from music [16]. While much work has focused on emotion prediction from audio features [5], comparatively less attention has been given to emotion-driven retrieval, which is particularly relevant for applications such as film scoring, media recommendation, and therapeutic music systems [17, 18]. Film provides a rich context for exploring emotion in music, as selective musical choices can reliably convey affect in otherwise ambiguous scenes, shaping both the overall atmosphere and perceived emotions of characters [3, 1]. A core aspect of MER is emotion representation. One common approach is categorical, assigning music discrete emotion labels [5, 19]. While convenient for classification, categorical representations have limitations: there is no universally agreed-upon set of emotions [20], and coarse categories may overlook subtle emotional nuances.

An alternative approach is dimensional, which represents emotion along continuous scales. Russell's circumplex model of emotion [21] is widely used in MER, employing Arousal (intensity) and Valence (positiveness) axes, typically scaled from -1 to $+1$. Extensions of this model include a third dimension, Potency, representing perceived control [22], but the two-dimensional model is generally preferred for ease of interpretation [23]. Figure 1 depicts this model. MER systems may also distinguish between felt emotions, evoked in listeners, and perceived emotions, intended by composers [20, 6]. This work focuses on perceived emotion, aiming to retrieve film music whose emotional content is recognisable and interpretable by users. While individual listeners may vary in their interpretation, multiple annotations per track were aggregated to capture a consensus emotional value, providing a practical approximation of the intended affective content [24, 20].

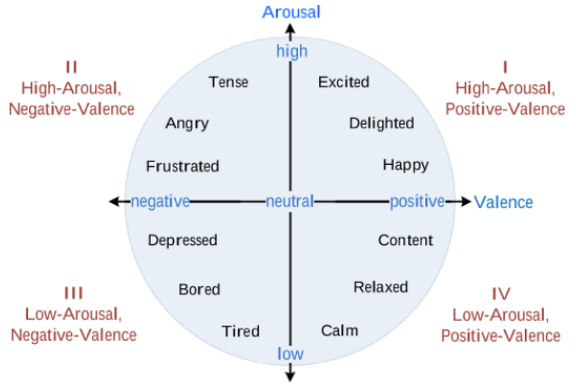


Figure 1. Russell's Circumplex Model of Emotions [21]

MER research has extensively examined audio features such as timbre, rhythm, energy, and melody [5]. Timbral features like spectral contrast and Mel-Frequency Cepstral Coefficients (MFCCs) are particularly effective [6]. Many of these features show consistent relationships with emotion: tempo and loudness are generally associated with Arousal, while mode and harmony typically relate to Valence [25, 26, 27]. However, emotional perception arises from complex interactions between spectral, rhythmic, and harmonic properties [7], and such associations are not absolute due to the subjective and context-dependent nature of emotion in music.

Given the large number of available audio features, incorporating all of them can result in the curse of dimensionality, where excessive and highly correlated features degrade model performance. Dimensionality reduction techniques, such as Principal Component Analysis (PCA), are commonly used to address this challenge, reducing the feature space while preserving most of the variance [28]. For instance, studies applying PCA to music emotion classification have demonstrated that accuracies exceeding 80% can still be achieved [29, 30].

k-Means is a computationally efficient and widely used clustering algorithm in Music Information Retrieval (MIR), suitable for grouping music based on extracted audio features. It has been shown to perform effectively when applied to time-frequency representations of music signals, enabling automatic feature extraction for tasks such as genre prediction [31]. In addition, combining k-Means with both high- and low-level audio features has proven effective for clustering music according to thematic or emotional content [32]. The handcrafted MER system applies k-means clustering, with grid search used to tune k and the silhouette score selecting the optimal configuration.

Recent advances in deep learning have shown that pretrained audio embeddings can outperform handcrafted features for downstream classification tasks. In particular, models such as VGGish and L3–Net, originally developed for Sound Event Detection (SED), have demonstrated excellent performance in music emotion classification [33], and multiple audio embedding models have achieved high accuracies in speech emotion recognition [34]. These results emphasize the advantages of leveraging rich, pretrained representations for emotion-driven analysis.

State-of-the-art deep clustering approaches combine generative models with contrastive learning, making them particularly suitable for film music retrieval, where subtle variations in audio features must be matched to nuanced emotional content.The Variational Deep Embedding (VaDE) model, used in this paper, operates within a Variational Autoencoder (VAE) framework in which multiple Gaussian distributions model latent priors, making it particularly effective for clustering tasks [35, 36]. This approach is integrated with contrastive learning inspired by CLIP, aligning latent audio embeddings with V-A coordinates treated analogously to natural language [37]. By projecting both audio representations and emotion embeddings into a shared latent space, this method enables optimised emotion-based retrieval.

3. METHOD

This section presents the methodological framework used to develop and evaluate two complementary approaches to MER for emotion-based film music retrieval. The first approach uses handcrafted audio features combined with unsupervised clustering to provide a transparent, interpretable baseline for emotion modelling. The second approach uses deep generative and contrastive learning methods, integrating Variational Deep Embedding (VaDE) with V-A alignment to enable richer, data-driven emotional representations.

3.1 FME-24 Dataset

Both methods are trained and evaluated using the FME-24 dataset, which provides film music excerpts annotated for continuous V-A dimensions. These annotations serve as the empirical basis for mapping perceived emotion onto acoustic features, enabling retrieval of music that aligns with specific emotional characteristics.

The FME-24 dataset comprises 300 unique film compositions spanning 21 genres and 156 composers [8], with associated film metadata collected from IMDb for contextual information. For annotation, participants listened to 15-second excerpts and recorded V-A scores whenever they perceived a change in emotion. Across 185 annotators, each labeling 10 excerpts, the dataset contains an average of 4–5 annotations per excerpt. After filtering and correcting for errors, 409 annotated samples from the original 600 were available for this study.

FME-24 was chosen for its explicit focus on emotional dimensions in film music, rather than categorical descriptors such as genre. Previous datasets often rely on older film music (majority pre-2000), which may not reflect contemporary compositional styles, instrumentation, or production techniques [38]. Other datasets have been constructed primarily for genre prediction tasks [39], making them less suitable for emotion-driven tasks. In contrast, FME-24 provides continuous V-A ratings with multiple annotations per sample, capturing within-track emotional variation and offering a robust foundation for training and evaluating both models in this study.

3.2 Handcrafted Features

To utilise the V–A annotations in FME-24, handcrafted audio features were extracted to capture emotion-relevant properties. Each 15-second excerpt was split into three 5-second segments, yielding 1,227 audio samples. Shorter segments ($1-3$ s) often contained little audio, while longer segments (> 5 s) offered limited benefit due to the already small dataset size. Each segment was assigned an ID to track its original excerpt, and audio loading and feature extraction were performed using Librosa [2] .

The features extracted can be broadly categorised into four categories: timbre, rhythm, loudness, and harmony. For a full breakdown of all the features extracted, see Table 1. Where applicable, a hop size of 1024 and FFT points of 4096 were used, providing an appropriate balance between time and frequency resolution. The initial feature set had 72 dimensions per sample (1227×72).

Dimensionality reduction was applied in three steps. First, low-variance features ($variance < 0.15$) and highly correlated features ($correlation > 0.75$) were removed, reducing the set to 35 features (from 72). KernelPCA with an RBF kernel ($\gamma = 0.1$) captured nonlinear relationships, reducing the data to 6 dimensions. RBF-kPCA was preferred over linear PCA for slight improvements in cluster metrics and to account for dataset nonlinearity.

The clustering algorithm used was k-Means (Scikit-Learn [3]). The optimal number of clusters ($k = 7$) was selected by inspecting silhouette scores across different k-values (Figure 2).

[2] https://librosa.org/doc/latest/feature.html
[3] https://scikit-learn.org/stable/index.html

Category	Feature	Aggregation	Dimension
Timbre	MFCCs	Mean	(20,)
	MFCCs	Std	(20,)
	Spectral Centroid	Mean	(1,)
	Spectral Bandwidth	Mean	(1,)
	Spectral Contrast	Mean	(1,)
Rhythm	Tempo	-	(1,)
	Onset Strength	Mean	(1,)
Loudness	RMS	Mean	(1,)
	RMS	Std	(1,)
	Beat RMS	Mean	(1,)
Harmony	Chroma Features	Mean	(12,)
	Chroma Features	Std	(12,)

Table 1. Extracted Audio Features

A two-dimensional visualisation of the resulting clusters is shown in Figure 3. Median aggregation was applied to cluster centroid distances and V-A scores to account for multiple segments per track and multiple emotional annotations per excerpt. This approach, chosen for its robustness to outliers, also prevents duplicate retrievals of the same track. Although retaining full annotation variability could better represent emotional nuance, a single median V-A score per track was used for simplicity.

The aggregated V-A values for each cluster are shown in Table 2. The retrieval component of this approach takes an input V-A score, calculates the Euclidean distance between each of the cluster scores, and returns the closest cluster. Results are ranked according to the distance between the track and the cluster centroid, with the lowest distances ranked at the top. It is worth noting that clusters are relatively close in the V-A space, as discussed further in Section 5.

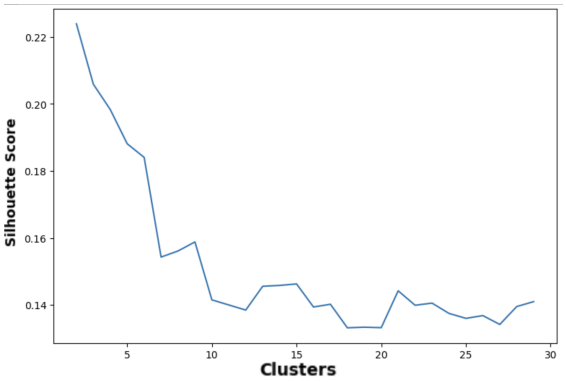


Figure 2. Silhouette Scores Across k for Handcrafted Clusters

Cluster	Arousal	Valence
1	0.13	0.37
2	0.08	0.13
3	0.17	0.18
4	0.18	0.20
5	0.25	0.35
6	0.01	0.07
7	-0.05	0.18

Table 2. Median Arousal-Valence per Cluster

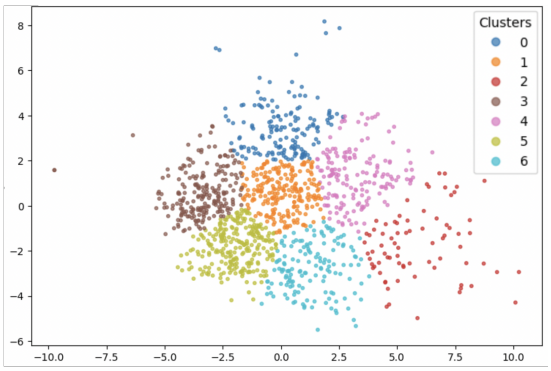


Figure 3. 2D k-Means Clusters (Handcrafted Features)

3.3 VaDE + Contrastive Learning Approach

Unlike the handcrafted feature approach, this method directly learns latent audio representations using deep embeddings. VGGish [4] and OpenL3 [5] were used for feature extraction, with hop sizes matched and audio converted to mono before processing. Each 15-second excerpt produced 15 spectrograms (approx. 1s each), expanding the dataset to 6,135 samples from the original 409. The extracted embeddings (128 VGGish + 512 OpenL3 dimensions) were concatenated into 640-dimensional representations.

In line with [35], the model is implemented in PyTorch [6] and follows a Variational Autoencoder structure with an encoder and decoder. Each consists of three sequential layers (512, 256, 128 neurons) with batch normalization, leaky ReLU activation, and dropout. The encoder outputs a 32-dimensional latent representation along with mean and log-variance parameters for Gaussian reparameterization, while the decoder reconstructs embeddings from this latent space.

To enable same space retrieval, V-A scores were encoded using a separate component, after median aggregation for the multiple scores per track. This encoder contained a linear layer with 64 dimensions, a ReLU activation function and a normalisation layer, executed sequentially and outputting a 32-dimensional latent representation. Both latent spaces were projected through a shared 32-dimensional projection head (linear $\rightarrow$ batch norm $\rightarrow$ ReLU) , shown to enhance representation quality in contrastive learning [40].

The complete loss function used by the model combines a VaDE component and a contrastive component, as seen in Equation 1, where λ controls the impact of the contrastive component of the model on the final weights:

$$L = L_{\text{VaDE}} + \lambda L_{\text{CL}} \tag{1}$$

A complete VaDE loss function can be seen in Equation 2 [35]. This VaDE loss is broken down into two parts: a reconstruction loss and a KL-divergence loss. The former ensures that the output of the decoder component matches the audio embedding input by minimising the Mean Squared Error (MSE). The latter can be further decomposed into latent distribution (KL_z) and cluster divergence (KL_c) terms:

$$L_{\text{VaDE}} = L_{\text{recon}} + \beta\big(L_{\text{KLz}} + L_{\text{KLc}}\big) \tag{2}$$

KL_z pulls each sample x_i toward its assigned cluster c's mean and variance, aligning the encoder's distribution with the Gaussian mixture prior (Equation 3). KL_c matches each sample's cluster assignment probabilities γ_{ic} to the global cluster distribution $p(c)$ over K clusters (Equation 4):

$$L_{\text{KLz}} = \frac{1}{2}\left[\log\frac{\sigma_c^2}{\sigma_{x_i}^2} + \frac{\sigma_{x_i}^2 + (\mu_{x_i} - \mu_c)^2}{\sigma_c^2} - 1\right] \tag{3}$$

$$L_{\text{KLc}} = \sum_{c=1}^{K} \gamma_{ic} \log\frac{\gamma_{ic}}{p(c)} \tag{4}$$

Equation (5) [40], shows the contrastive loss function, it calculates the loss for a given sample i, where all encoded V-A scores and latent audio embeddings have been normalised. The numerator sums the cosine similarities between z_i and embeddings a_j in the same cluster c_i, while the denominator sums similarities across all other samples ($j \neq i$), both inside and outside the cluster. Similarities are scaled by a temperature parameter $\tau = 0.1$ for the final model.

$$L_{\text{CL}}(i) = -\log\frac{\sum_{j\in\mathcal{P}(i)} \exp\big(\text{sim}(z_i, a_j)/\tau\big)}{\sum_{k\neq i} \exp\big(\text{sim}(z_i, a_k)/\tau\big)} \tag{5}$$

Training was conducted for 200 epochs using the AdamW optimizer (learning rate 1×10^{-5}, weight decay 1×10^{-4}), with a OneCycleLR scheduler and gradient clipping to stabilise training. Cluster means were initialised via k-Means on the encoded audio space. Figure 4 shows a 2-D visualisation of the embeddings post-clustering. Six Gaussian priors were used, determined via a silhouette score analysis similar to Figure 2, though the embeddings naturally converged into three dominant clusters. This three-cluster structure may reflect broad emotional groupings or simply dataset imbalance; this could be looked at in future work. Figure 5 plots the loss variables over the training loop.

Projected audio embeddings were precomputed for efficient querying. New V–A inputs were matched using cosine similarity and ranked to retrieve the top-k closest tracks. The sensitivity of retrieval quality to the choice of k was not assessed, representing a limitation of this work. In this system, clusters function mainly as an emotional representation rather than being directly used for retrieval.

4. EVALUATION

Evaluation of the two approaches was conducted using both quantitative and qualitative methods, outlined in the following sections.

[4] https://docs.pytorch.org/audio/main/generated/torchaudio.prototype.pipelines.VGGISH.html

[5] https://pypi.org/project/openl3/

[6] https://pytorch.org/

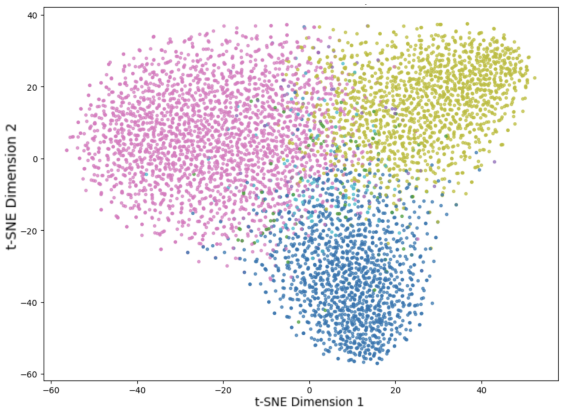


Figure 4. t-SNE of VaDE + CL Latent Space

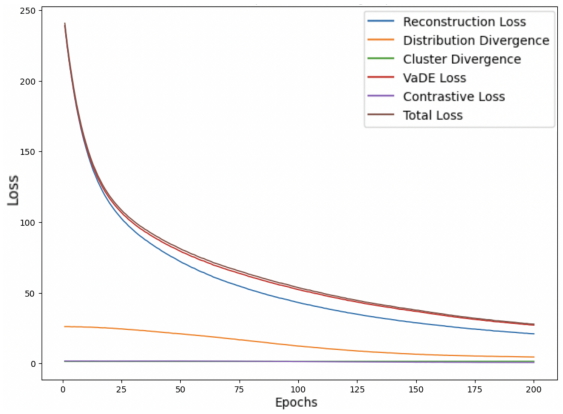


Figure 5. VaDE + CL Training Losses

4.1 Quantitative Metrics

Three metrics were used to evaluate cluster quality: silhouette [41], Davies-Bouldin [42], and Calinski-Harabasz [43]. Each metric quantifies within-cluster cohesion and between-cluster separation. These values can be used to compare clustering quality across different models. The silhouette score assesses how well samples match their assigned cluster compared to others, the Davies-Bouldin index quantifies average inter-cluster similarity (lower is better), and the Calinski-Harabasz index captures the ratio of between- to within-cluster dispersion (higher is better). Together, these metrics offer complementary perspectives on cluster compactness and separability across models.

4.2 Qualitative Metrics

Both models were evaluated through an online user study designed to capture subjective responses to emotion-based retrieval. The survey application was built with a React [7] front-end and Flask [8] back-end, hosted on Firebase [9] and Render [10], respectively. Firebase also handled storage for audio files, film metadata, and survey responses. Figure 6 shows the V–A query input page, while Figure 7 illustrates the results interface.

[7] https://react.dev/
[8] https://flask.palletsprojects.com/
[9] https://firebase.google.com/
[10] https://render.com/

Participants completed two retrieval tasks (each with a different query), selecting the retrieved film music excerpts that best matched their intended emotional input. They could also submit new queries to generate additional results. After each task, participants answered questions on demographics, listening habits, and the role of emotion in their music preferences. The survey was administered twice (once for each model) with $k = 5$ retrieved results per query. Participants answered four questions regarding the emotional relevance and consistency of the retrieved results:

1. How many songs were emotionally relevant to the Arousal–Valence score entered?

2. The songs shown were emotionally similar to the Arousal–Valence score I entered.

3. The songs shown were emotionally similar to each other.

4. The order of songs decreased in relevance to the Arousal–Valence score I entered.

Question 1 used a discrete scale ($0-5$ emotionally relevant songs), while Questions $2-4$ used a five-point Likert scale.

Participants were recruited via academic mailing lists (including Queen Mary's Centre for Digital Music and the School of Electronic Engineering and Computer Science) This approach provided a diverse pool, combining individuals with some musical familiarity and members of the general public, ensuring responses were both informed and broadly generalisable [8].

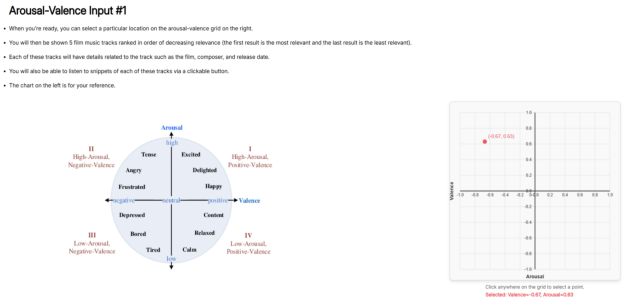


Figure 6. V–A Query Input

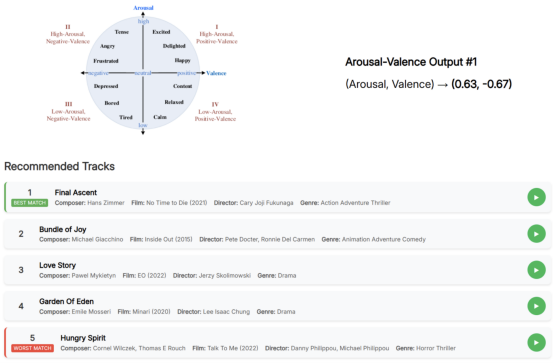


Figure 7. Retrieval Output Interface

5. RESULTS AND DISCUSSION

Results are presented in two parts: Section 5.1 reports quantitative clustering performance, and Section 5.2 summarizes qualitative findings from the user study.

5.1 Quantitative Results

Cluster evaluation metrics are summarized in Table 3. The handcrafted feature model achieved higher silhouette and lower Davies–Bouldin scores, indicating stronger within-cluster cohesion and clearer separation. The VaDE + CL model achieved a slightly higher Calinski–Harabasz score, suggesting denser overall clusters. Visual inspection (Figure 4) shows three dominant VaDE + CL clusters alongside several smaller, less distinct ones.

The lower silhouette and higher Davies–Bouldin scores for VaDE + CL likely result from these smaller, less distinct clusters. Operating in a higher-dimensional latent space (32 features vs. 6 for the handcrafted model) reduces the reliability of Euclidean distance as a similarity measure, as points become more uniformly distant. Nonetheless, relative differences remain informative for comparing cluster quality.

Dataset and embedding limitations constrain the VaDE + CL approach. While VaDE was originally developed for datasets exceeding 10,000 samples [35], it is applied here to only 6,135 excerpts. Short embedding windows (1 s), chosen for compatibility with OpenL3 and VGGish, may fail to represent longer, atmospheric film cues. The aggregated V–A scores of handcrafted clusters (Table 2) occupy a narrow region around the neutral point (0,0), underrepresenting extreme emotions. This compressed emotional range limits cluster separability and reduces both models' ability to learn or retrieve distinct affective patterns despite the broader emotional variation present in the music.

VaDE + CL provides greater retrieval flexibility. Unlike the handcrafted approach, which always returns the same tracks for a cluster, VaDE + CL works across the full latent space. Each V–A input is projected into this space, allowing retrieval of the top-k nearest embeddings. This enables finer selection of emotionally aligned tracks, for example, smoothly interpolating between calm and tense music rather than jumping between fixed clusters.

To better reflect retrieval performance, V–A proximity was also considered as a simple ranking-based measure, where the Euclidean distance between the query and retrieved tracks indicates emotional similarity, with smaller distances corresponding to more relevant results, this can be found in the .ipynb file via the GitHub repository .

5.2 Qualitative Results

The user study (Section 4) was completed by 34 and 26 participants for the handcrafted and VaDE + CL models, respectively, who completed two retrieval tasks and answered four questions regarding emotional relevance and consistency.

Figures 8 and 9 summarize responses to Questions 1 and 3. More handcrafted participants reported $4-5$ emotionally relevant tracks, while 9% more VaDE + CL participants reported none. Around 50% of respondents for both models selected "Mostly Agree" for Question 3, indicating that returned tracks were largely emotionally consistent. This suggests that clustering captures some affective coherence, but retrieval precision is limited: 62% of handcrafted and 73% of VaDE + CL respondents retrieved $0-3$ relevant tracks out of 5.

Small sample sizes and high variability should be noted, but these results provide useful approximations of user-perceived performance.

Model	Metric	Value	Range	Optimal
Handcrafted Features	Silhouette Score	0.15	(-1, +1)	+1
	Davies-Bouldin	1.57	(0, ∞)	0
	Calinski-Harabasz	218.07	(0, ∞)	∞
VaDE + CL	Silhouette Score	0.09	(-1, +1)	+1
	Davies-Bouldin	3.65	(0, ∞)	0
	Calinski-Harabasz	232.35	(0, ∞)	∞

Table 3. Cluster Metrics: Handcrafted vs VaDE + CL

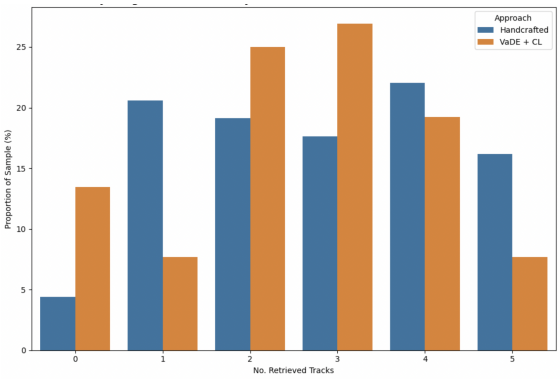


Figure 8. Emotional Relevance Responses - Q1

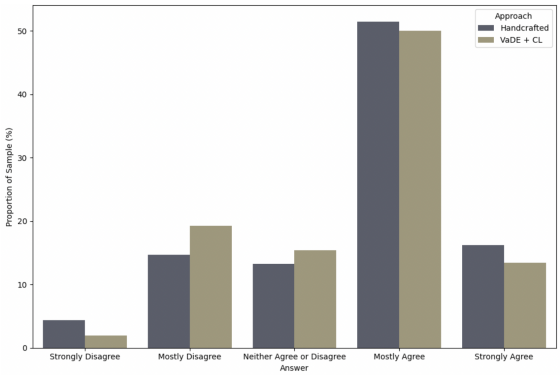


Figure 9. Emotional Consistency Responses - Q3

Quantitative and qualitative results suggest that VaDE + CL allows more flexible retrieval, though the compressed emotional range and dataset constraints appear to limit differentiation of affective patterns. Expanding the dataset or refining V–A representations may help improve learning.

6. CONCLUSION AND FUTURE WORK

This paper investigated emotionally targeted retrieval of film music using Valence–Arousal (V–A) inputs. Two models were developed: a handcrafted k-Means model using timbre, rhythm, loudness, and harmony features, and a VaDE + CL deep generative model with contrastive learning. By aligning audio and V–A embeddings in a shared latent space, VaDE + CL enables retrieval through proximity, allowing finer selection of tracks that match the desired emotional content.

Both models captured affective patterns differently: the handcrafted model as a baseline, and VaDE + CL enabling more flexible, fine-grained retrieval. The main limitation was the small size of FME-24 dataset. Sparse clusters, compressed emotion range, and silent segments further constrained detection of subtle affective differences, highlighting priorities for future dataset expansion and refinement. These approaches can extend beyond film music retrieval. VaDE + CL, in particular, could be applied to alternative auxiliary variables, larger datasets, or other audio domains such as speech. Its generative and latent-space capabilities make it suitable for retrieval, synthesis, and other affective audio tasks beyond the current study.

Another limitation was the treatment of the FME-24 dataset: only 6,135 excerpts were available, with around 409 annotated for V–A. Multiple annotations per track were aggregated into a single value, which likely reduced the emotional diversity the models could learn from. Future work should explore using the full range of annotator responses to capture inter-annotator variance, potentially improving sensitivity to subtle affective differences and enabling richer, more nuanced retrieval. The handcrafted model is restricted to fixed clusters, often returning the same results for repeated inputs, with clusters exhibiting minimal variation in V–A scores. The VaDE + CL model faced challenges with small or sparse clusters, high-dimensional embeddings that reduce the interpretability of distance-based metrics, and short or silent audio segments that produced non-informative embeddings. These factors collectively limit the models' ability to capture and retrieve distinct affective patterns.

The evaluation uses a direct, interpretable measure of retrieval quality based on aggregated V–A annotations, but does not capture temporal variations in emotion, limiting sensitivity to the dynamic changes characteristic of film music. This reflects a trade-off between temporal detail and robust evaluation given the dataset size.

Future work could first focus on enhancing the FME-24 dataset itself. Increasing the number of participants providing V–A annotations would improve the reliability of the aggregated emotional scores, while adding more film music excerpts could reveal broader patterns and improve model training. Combining V–A annotations with written perceived emotion labels could further enrich the representation of emotional nuances, benefiting both clustering and retrieval. Additionally, evaluating the models on other datasets, or on a combined FME-24 and external dataset, could provide opportunities for transfer learning and further improve model performance.

Alongside dataset improvements, methodological refinements could strengthen the models. Pre-processing to remove silent or non-informative audio segments would improve embedding quality, while exploring alternative pre-trained audio embeddings and dimensionality reduction techniques could enhance latent-space clustering and reduce overfitting. Knowledge-based grouping of V–A clusters may help distinguish subtle affective categories. Finally, testing the models on larger or alternative datasets could assess generalizability, and given sufficient data, the generative capabilities of VaDE + CL could be used to create new music aligned with target V–A scores and combined with additional constraints such as style and instrumentation.

This work shows that while both handcrafted and deep learning approaches can capture insightful patterns, deep generative models combined with contrastive learning provide more flexible, fine-grained emotional matching. Future work will focus on expanding the FME-24 dataset and addressing the variance in V–A annotations. Combining these annotations with emotion descriptions could help distinguish more subtle emotional differences in music.

7. REFERENCES

[1] A.-K. Herget, "On music's potential to convey meaning in film: A systematic review of empirical evidence," *Psychology of Music*, vol. 49, no. 1, pp. 21–49, 2021.

[2] M. Tervaniemi, T. Makkonen, and P. Nie, "Psychological and physiological signatures of music listening in different listening Environments—An exploratory study," *Brain sciences*, vol. 11, no. 5, p. 593, 2021.

[3] R. Parke, E. Chew, and C. Kyriakakis, "Quantitative and visual analysis of the impact of music on perceived emotion of film," *Computers in Entertainment (CIE)*, vol. 5, no. 3, p. 5, 2007.

[4] J. Kang and D. Herremans, "Towards Unified Music Emotion Recognition across Dimensional and Categorical Models," *arXiv preprint arXiv:2502.03979*, 2025.

[5] Y.-H. Yang and H. H. Chen, "Machine recognition of music emotion: A review," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 3, no. 3, pp. 1–30, 2012.

[6] M. Barthet, G. Fazekas, and M. Sandler, "Music emotion recognition: From content-to context-based models," in *International symposium on computer music modeling and retrieval*. Springer, 2013, pp. 228–252.

[7] Y. Song, S. Dixon, and M. T. Pearce, "Evaluation of musical features for emotion classification." in *ISMIR*, 2012, pp. 523–528.

[8] R. Crocker and G. Fazekas, "Temporal analysis of emotion perception in film music: insights from the FME-24 dataset," in *Sound and Music Computing Conference Proceedings*, vol. 21, 2024, pp. 247–252.

[9] K. J. Kemper and S. C. Danhauer, "Music as therapy," *South Med J*, vol. 98, no. 3, pp. 282–8, 2005.

[10] D. Aldridge, *Music therapy in palliative care: New voices*. Jessica Kingsley Publishers, 1999.

[11] H.-L. Lai and Y.-M. Li, "The effect of music on biochemical markers and self-perceived stress among first-line nurses: A randomized controlled crossover trial," *Journal of advanced Nursing*, vol. 67, no. 11, pp. 2414–2424, 2011.

[12] T. Ventura, M. C. Gomes, and T. Carreira, "Cortisol and anxiety response to a relaxing intervention on pregnant women awaiting amniocentesis," *Psychoneuroendocrinology*, vol. 37, no. 1, pp. 148–156, 2012.

[13] D. Fancourt, A. Ockelford, and A. Belai, "The psychoneuroimmunological effects of music: A systematic review and a new model," *Brain, behavior, and immunity*, vol. 36, pp. 15–26, 2014.

[14] R. Hou, "Music content personalized recommendation system based on a convolutional neural network," *Soft Computing*, vol. 28, no. 2, pp. 1785–1802, 2024.

[15] M. Velankar and P. Kulkarni, "Music recommendation systems: overview and challenges," *Advances in speech and music technology: Computational aspects and applications*, pp. 51–69, 2022.

[16] D. Han, Y. Kong, J. Han, and G. Wang, "A survey of music emotion recognition," *Frontiers of Computer Science*, vol. 16, no. 6, p. 166335, 2022.

[17] M. J. Silverman, "Songwriting Techniques in Music Therapy," *Music and Music Therapy Interventions in Clinical Practice: A Textbook for Music Therapy Professionals, Clinicians and Researchers*, p. 91, 2025.

[18] J. Bradt, C. Dileo, K. Myers-Coffman, and J. Biondo, "Music interventions for improving psychological and physical outcomes in people with cancer," *Cochrane Database of Systematic Reviews*, no. 10, 2021.

[19] R. W. Picard, E. Vyzas, and J. Healey, "Toward machine emotional intelligence: Analysis of affective physiological state," *IEEE transactions on pattern analysis and machine intelligence*, vol. 23, no. 10, pp. 1175–1191, 2001.

[20] J. A. Sloboda and P. N. Juslin, "Psychological perspectives on music and emotion," *Music and emotion: Theory and research*, pp. 71–104, 2001.

[21] J. A. Russell, "A circumplex model of affect." *Journal of personality and social psychology*, vol. 39, no. 6, p. 1161, 1980.

[22] J. A. Russell and A. Mehrabian, "Evidence for a three-factor theory of emotions," *Journal of research in Personality*, vol. 11, no. 3, pp. 273–294, 1977.

[23] L.-C. Yu, L.-H. Lee, S. Hao, J. Wang, Y. He, J. Hu, K. R. Lai, and X. Zhang, "Building Chinese affective resources in valence-arousal dimensions," in *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: Human language technologies*, 2016, pp. 540–545.

[24] W. F. Thompson and B. Robitaille, "Can composers express emotions through music?" *Empirical studies of the arts*, vol. 10, no. 1, pp. 79–89, 1992.

[25] A. Gabrielsson and E. Lindström, "The Influence of Musical Structure on Emotional Expression," in *Music and Emotion: Theory and Research*, P. N. Juslin and J. A. Sloboda, Eds. Oxford University Press, 2001, pp. 223–248.

[26] Z. Yang, Q. Su, J. Xie, H. Su, T. Huang, C. Han, S. Zhang, K. Zhang, and G. Xu, "Music tempo modulates emotional states as revealed through EEG insights," *Scientific reports*, vol. 15, no. 1, p. 8276, 2025.

[27] L. M. Hofbauer and F. S. Rodriguez, "Emotional valence perception in music and subjective arousal: Experimental validation of stimuli," *International journal of psychology*, vol. 58, no. 5, pp. 465–475, 2023.

[28] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley interdisciplinary reviews: computational statistics*, vol. 2, no. 4, pp. 433–459, 2010.

[29] Y. Xia and F. Xu, "Study on music emotion recognition based on the machine learning model clustering algorithm," *Mathematical Problems in Engineering*, vol. 2022, no. 1, p. 9256586, 2022.

[30] H. Khabiri, M. N. Talebi, M. F. Kamran, S. Akbari, F. Zarrin, and F. Mohandesi, "Music-induced emotion recognition based on feature reduction using PCA from EEG signals," *Frontiers in Biomedical Technologies*, 2023.

[31] J. Wülfing and M. A. Riedmiller, "Unsupervised Learning of Local Features for Music Classification." in *ISMIR*, 2012, pp. 139–144.

[32] Y. Jiang and X. Jin, "Using k-means clustering to classify protest songs based on conceptual and descriptive audio features," in *International Conference on Human-Computer Interaction*. Springer, 2022, pp. 291–304.

[33] E. Koh and S. Dubnov, "Comparison and analysis of deep audio embeddings for music emotion recognition," *arXiv preprint arXiv:2104.06517*, 2021.

[34] O. C. Phukan, A. B. Buduru, and R. Sharma, "A comparative study of pre-trained speech and audio embeddings for speech emotion recognition," *arXiv preprint arXiv:2304.11472*, 2023.

[35] Z. Jiang, Y. Zheng, H. Tan, B. Tang, and H. Zhou, "Variational deep embedding: An unsupervised and generative approach to clustering," *arXiv preprint arXiv:1611.05148*, 2016.

[36] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[37] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PmLR, 2021, pp. 8748–8763.

[38] T. Eerola, "Are the emotions expressed in music genre-specific? An audio-based evaluation of datasets spanning classical, film, pop and mixed genres," *Journal of New Music Research*, vol. 40, no. 4, pp. 349–366, 2011.

[39] B. Ma, T. Greer, D. Knox, and S. Narayanan, "A computational lens into how music characterizes genre in film," *PloS one*, vol. 16, no. 4, p. e0249957, 2021.

[40] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*. PmLR, 2020, pp. 1597–1607.

[41] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987.

[42] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE transactions on pattern analysis and machine intelligence*, no. 2, pp. 224–227, 2009.

[43] T. Caliński and J. Harabasz, "A dendrite method for cluster analysis," *Communications in Statistics-theory and Methods*, vol. 3, no. 1, pp. 1–27, 1974.

Vibrational Analysis of Traditional Persian Kamanche Sound Box: Experimental and Computational Investigation of Structural Modifications

Amir Abbas Orouji
INESC TEC, Faculty of Engineering, University of Porto
Amir.A.Orouji@outook.com

Ayoub Banooshi
IRIB University
ayoubbanoushi@yahoo.com

Gilberto Bernardes
INESC TEC, Faculty of Engineering, University of Porto
gba@fe.up.pt

ABSTRACT

The kamanche, a bowed spike fiddle central to Persian classical music, features a spherical sound box covered with stretched animal skin and is played vertically on the performer's lap. Despite acoustic similarities to the violin, comprehensive research on kamanche acoustics remains limited. This study investigates the acoustic contribution of the sound box to resonance characteristics and tonal quality of the closed-back kamanche, the most prevalent contemporary variant. The research combines COMSOL Multiphysics vibration simulation with experimental validation through impulse response frequency measurements. Investigated modifications include upper and lower hemisphere thickness variations and sound hole area reduction. Results demonstrate that upper hemisphere changes, while preserving internal air volume, substantially affect fundamental resonance patterns, corroborating traditional luthier observations. This study also suggests that the vibration modes 4,5, and especially 7 might be good candidates for maximum contribution to the overall amplification of the string's resonance and the overall sound of the instrument.

1. INTRODUCTION

The kamanche (Persian for "small bow") represents a central bowed string instrument in Persian classical music with over 1000 years of documented history according to UNESCO intangible cultural heritage [1]. The instrument comprises five primary components: neck, spherical sound box, membrane, soundboard, and bridge (see Figure 1). Originally three-stringed, the kamanche was modified in the 20th century to incorporate a fourth string, with common tunings including E-A-E-A, G-D-A-E, or C-G-C-G [1, 2].

Regional variations exhibit significant morphological diversity. Two primary variants exist: closed-back instruments with sealed sound boxes, and open-back variants where the posterior surface is replaced with a metallic support rod [3]. Traditional construction employs hardwoods, predominantly walnut (Juglans regia) and mulberry (Morus) [4, 5].

Sound production results from complex vibroacoustic interactions: bow-string friction dynamics, bridge transmission, membrane vibration, internal air resonance, and sound box coupling [6]. However, the relative acoustic contributions of these components remain poorly understood due to limited academic research and preservation of traditional construction methods through master-apprentice luthier relationships. Master luthiers observe that minor sound box thickness and shape modifications substantially alter the instrument's timbre, yet systematic quantification of these effects is absent from the literature. This study addresses the knowledge gap by characterizing the acoustic properties of standalone sound boxes through an integrated computational and experimental approach, combining Finite Element Modeling (FEM) with traditional luthier expertise to establish relationships between geometric modifications and vibrational characteristics.

The remainder of this paper is structured as follows. Section 2 reviews existing literature and related works. Section 3 details the experimental methodology, which applies FEM. Section 4 reports the simulation findings, and Section 5 concludes the work by analyzing the results and discussing possibilities for further development.

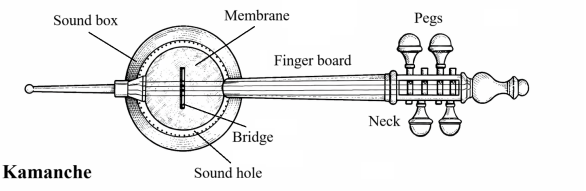


Figure 1. Kamanche illustration: the tuning pegs adjust string tension; an unfretted, violin-like fingerboard runs along the neck; bow-excited string vibrations pass through the bridge to the taut skin membrane atop the sound box, which amplifies and radiates the sound.

2. RELATED WORKS

Apart from the fact that the acoustic properties of wood are anisotropic, they can be drastically different depending on the circumstances in which the tree has grown, the wood was kept, and stored. Mulberry wood is no exception, and even though the anisotropic acoustic properties of mulberry and walnut woods have been studied [7, 8], research

[1] https://ich.unesco.org/en/RL/art-of-crafting-and-playing-with-kamantcheh-kamancha-a-bowed-string-musical-instrument-01286

Copyright: ©2026. Amir Abbas Orouji et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License 3.0 Unported, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

on Iranian musical instruments remains limited. Notable contributions include Mansour's FEM and Boundary Element Modeling (BEM) of the setar, examining bridge mechanics and sound field characteristics, as well as an investigation of acoustic quality indicators for mulberry wood (Morus Alba) used in Persian instruments, including the kamanche [9, 10]. Tronchin et al. [11] introduced the intensity of acoustic radiation parameter for Persian instruments (tar, setar, santoor), revealing specific harmonic ratios in resonant frequencies (3:2 for tar, 2:1 and 6:5 for setar) linked to characteristic timbres.

Thickness variations significantly affect string instrument acoustics. Gough [12] demonstrated that violin plate thickness influences eigenfrequencies and mode shapes, with the frequency ratio between modes 5 and 2 achieving values found in fine Italian violins at approximately 2.5 mm uniform thickness. Stanciu et al. [13] found that decreasing plate thickness by 0.6 mm reduced the first eigenfrequency by approximately 10%, with effects most evident in cavity modes after assembly. Recent work on the Bilbao project violins [14] employed CNC-machined plates and FEM updating to separate intentional design variations from natural wood variability.

Internal air cavities function as Helmholtz-type resonators. Elejabarrieta et al. [15, 16] developed FEM models coupling wooden structure with enclosed air for guitar resonance boxes, showing that enclosed gas type affects modal patterns. For violins, signature modes including cavity mode (A0), C-bout rhomboidal mode (CBR), and main body resonances (B1-, B1+) are essential for sound production [17]. However, Saati [6] concluded that kamanche sound box vibrations were negligible compared to membrane motion through experimental modal analysis using shaker excitation, which is contradictory to the luthiers' understanding.

Master luthier Mr. Ahmadi notes that minor changes in sound box shape and thickness substantially alter instrument sound. To address these conflicting perspectives, this study provides one of the first systematic acoustic analyses of the kamanche sound box and its vibrational characteristics, addressing the knowledge gap in Persian musical instrument acoustics.

3. METHODOLOGY

Our research employed a combined experimental and computational approach to characterize the acoustic behavior of the kamanche sound box. The methodology consisted of experimental modal analysis through impulse response testing, followed by FEM and parametric studies of geometric modifications.

3.1 Finite Element Modeling (FEM)

Computational modeling employed COMSOL Multiphysics [18] Structural Mechanics module for eigenfrequency analysis using axisymmetrical 2D modeling for faster calculation (see Figure 2). Due to limited literature on Persian mulberry wood acoustic properties with musical application, material properties were extracted from Pourtahmasi's data on Iranian wood species [19] [2] .

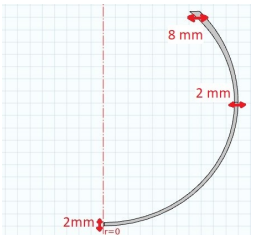


Figure 2. The axisymmetrical 2D model of the soundbox in COMSOL Multiphysics

Property	Value	Unit
Sample Code	C4	-
Density (ρ)	576.58	kg/m³
Longitudinal Velocity (V_L)	4845.24	m/s
Radial Velocity (V_R)	1833.86	m/s
Tangential Velocity (V_T)	2519.70	m/s
Longitudinal Young's Modulus (E_L)	13058.51	MPa
Radial Young's Modulus (E_R)	1880.87	MPa
Tangential Young's Modulus (E_T)	3531.52	MPa
Acoustic Quality Factor (Q_f)	148	-

Table 1. Acoustic properties of *Morus alba* sample C4 derived from work of Pourtahmasi [19].

Among the wood samples Pourtahmasi studied, the sample C4, shown in Table 1, provided the closest experimental correlation to our impulse response. These validated properties served as the reference model for parametric studies despite residual discrepancies between experimental and computational results.

3.2 Modal Analysis

Frequency Response Functions (FRFs) were obtained via impact-excitation testing of a kamanche sound box provided by a master luthier, with a sound-box diameter of 17 cm and a top sound-hole diameter of 10 cm. The sound box was suspended using elastic cords within a rigid frame to approximate free-free boundary conditions (shown in the right image of Figure 3). Impact excitation was employed by a B&K Type 8202 instrumented hammer with a Type 8200 force transducer, while vibrational response was measured using five B&K Type 2222C accelerometers strategically positioned to capture dominant mode shapes, as shown in the left image of Figure 3. Data acquisition utilized a B&K Type 7533 LAN interface connected to a W8 analyzer system.

Frequency response functions were calculated using B&K Pulse LabShop software with a Hanning window and linear averaging. Modal parameter extraction employed LMS TestLab [21] software with stabilization diagram analysis. Modal validation used three convergence criteria: Modal Phase Collinearity (MPC), Modal Phase Deviation (MPD), and Modal Assurance Criterion (MAC), yielding 20 validated modes within 0-3100 Hz.

3.3 Parametric Geometric Studies

Systematic geometric modifications investigated the hypothesis that sound box thickness variations, particularly

[2] For this purpose, the common wood properties manuals, including the Wood Handbook [20] and Bucur's handbook [5], were also studied.

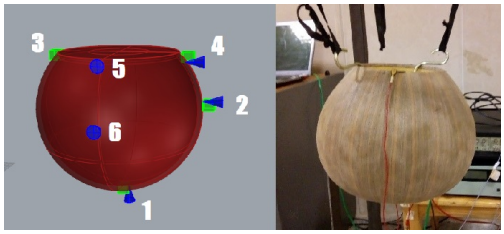


Figure 3. Left: The position of accelerometers (numbers 1,2,3,4 with green color) and hitting points (1,2,4,5, and 6 with blue color) in measuring FRF using impulse response. Right: The free-free impulse response analysis setup for the sound box.

around the upper aperture, influence tonal characteristics toward higher frequencies and enhanced treble content. Since thickness of the sound box changes area of contact with membrane which is one of the critical elements in the overall sound of the instrument [22], four modification scenarios were analyzed with constant internal air volume: (1) upper hemisphere thickness variation from 2 mm to 8 mm, (2) lower hemisphere thickness variation from 2 mm to 8 mm with unchanged upper geometry, (3) uniform thickness reduction from 4 mm to 2 mm throughout, and (4) sound hole area reduction by 50%.

To correlate modal results with actual acoustics, recordings from the completed kamanche were obtained for all four strings tuned to C4 (262 Hz), G4 (391 Hz), C5 (523 Hz), and G5 (781 Hz). Harmonic content analysis established relationships between sound box modal characteristics and spectral response.

4. RESULTS

Experimental modal analysis identified 20 distinct modes within the 0-3100 Hz frequency range (see Figure 4). Comparison with finite element predictions using Pourtahmasi's C4 mulberry wood properties showed a reasonable correlation for lower-order modes, with MPC exceeding 90% and MPD below 20 for most modes, indicating an acceptable identification quality. However, significant frequency deviations were observed, with some modes showing experimental-to-simulation ratios exceeding 2:1. These discrepancies likely arise from material property variability between reference samples and the tested specimen, as well as differences in wood treatment, aging, and moisture content. Despite quantitative differences, C4 properties provided the most appropriate available data for Iranian mulberry wood and were selected as the reference model for parametric studies.

The parametric study results are presented in Figure 5. Upper hemisphere thickening (2mm to 8mm) produced systematic frequency increases of 8-15% in the mid-frequency range (green line). Lower hemisphere thickening yielded moderate shifts primarily below 1500 Hz, with minimal deviation from the reference model (red line). Uniform thickness reduction (4mm to 2mm) caused frequency decreases of 15-30% across all modes (purple line). Aperture closure by 50% produced frequency increases comparable to upper thickening (brown line).

The frequency ranking from these modifications follows: Thinner All $<$ base model $<$ Thicker Bottom $<$ Thicker Top $\approx$ Half-closed. Upper hemisphere modifications consistently produced greater frequency shifts than lower hemisphere changes. Mode shapes remained qualitatively similar across all configurations, with primary differences appearing in the magnitude of frequency shifts rather than modal characteristics. The aperture closure effect was comparable to upper hemisphere thickening, suggesting structural significance beyond its acoustic radiation function.



Figure 4. Impulse response results comparison with the modal analysis simulation of the sound box using Pourtahmasi properties of Mulberry (sample C4).

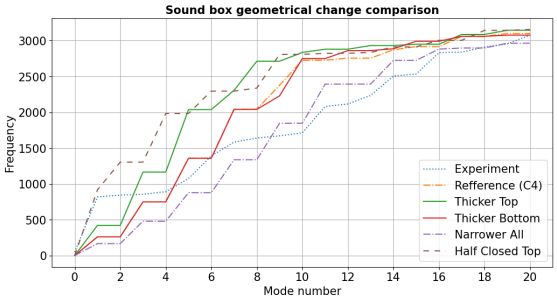


Figure 5. The comparison of modes among different geometrical changes using COMSOL. The first six modes in the sound box with a smaller hole area (Half Closed Top) are the highest among all. An increase in the thickness of the upper hemisphere (Thicker Top) also leads to an increase in the frequency of these modes, while changes in the lower hemisphere (Thicker Bottom) show almost no change compared to the reference. This effect is reversed when the overall thickness is reduced (Narrow All).

Modal-string coupling analysis of a CGCG-tuned kamanche revealed selective acoustic reinforcement. Spectral analysis of individual strings demonstrated that Modes 4 (480 Hz), 5 (720 Hz), and 7 (1050 Hz) provide primary resonant support to the C strings through harmonic alignment, with mode 7 at approximately 1000-1100 Hz serving as the dominant coupling mechanism where multiple string harmonics converge [3]. The sound box exhibited minimal modal support for G strings, particularly at their fundamental frequencies, indicating that the structural design preferentially reinforces the central C strings that define the instrument's characteristic timbre.

5. CONCLUSIONS AND DISCUSSION

This study provides the first systematic acoustic analysis of the kamanche sound box using experimental and computational methods. Key findings validate traditional luthier

[3] Sound recording of each string of the Kamanche used for Spectral analysis can be found in the online supplementary material to this article available at: https://anonymous.4open.science/r/ICMC-2026-files-0E8B/README.md

practices: upper hemisphere modifications significantly affect acoustic behavior, while lower hemisphere changes have minimal impact. This explains why craftsmen focus on upper-body adjustments for tonal refinement. Upper modifications preferentially affect mid-to-high frequency modes, supporting claims that such changes enhance treble and brightness. The acoustic impact of aperture closure matches upper thickening, demonstrating the sound hole functions as a critical structural discontinuity, not just a sound port. Local stiffness changes near this region are amplified due to stress concentration at geometric discontinuities. Consistent mode shapes across configurations indicate modifications alter modal frequencies through stiffness changes rather than fundamentally changing vibrational patterns. For a kamanche tuned to C-G-C-G, modes 4, 5, and 7 show the strongest coupling to the C string. Discrepancies between experimental and computational results reflect challenges in modeling instruments made from natural materials with variable properties. Limited characterization of Iranian wood species constrains predictive accuracy, though qualitative trends provide valuable design guidance. Future research should address: (1) mechanical and acoustic characterization of Iranian woods, (2) coupled fluid-structure-acoustic models, (3) correlation studies linking modal characteristics to perceived sound qualities, and (4) extension to other Persian instruments like tar and setar.

6. REFERENCES

[1] J. During, *The spirit of sounds: the unique art of Ostad Elahi (1895-1974)*. Associated University Press, 2003.

[2] H. Farhat, *The dastgah concept in Persian music*. University of California, 1965.

[3] S. Blum, P. Bohlman, P. Bohlman, and D. Neuman, *Ethnomusicology and modern music history*. University of Illinois Press, 1993.

[4] U. G. Wegst, "Wood for sound," *American Journal of Botany*, vol. 93, pp. 1439–1448, 2006.

[5] V. Bucur, *Handbook of materials for string musical instruments*, 2016.

[6] F. Saati, K. Sepahvand, M. Maeder, K. Pourtahmasi, and S. Marburg, "Modal Analysis of the Persian Music Instrument Kamancheh: Finite Element Modeling and Experimental Investigation." ISMA, 6 2014.

[7] I. Brémaud, J. Gril, and B. Thibaut, "Anisotropy of wood vibrational properties: dependence on grain angle and review of literature data," *Wood Science and Technology*, vol. 45, pp. 735–754, 11 2011.

[8] E. Obataya, T. Ono, and M. Norimoto, "Vibrational properties of wood along the grain," *Journal of Materials Science, Springer*, vol. 35, pp. 2993–3001, 2000.

[9] H. Mansour, A. Kasaiezadeh, S. Arzanpour, and M. Behzad, "Finite Element Modeling of Setar, a Stringed Musical Instrument," in *Volume 15: Sound, Vibration and Design*. ASME, 2009, pp. 591–597.

[10] H. Mansour, S. Arzanpour, H. Alghassi, and M. Behzad, "An Experimental Investigation on the Bridge Effect in Sound Produced by Setar," *ASME International Mechanical Engineering Congress and Exposition, Proceedings*, vol. 15, pp. 491–495, 7 2010.

[11] L. Tronchin, M. Manfren, and V. Vodola, "Sound Characterization through Intensity of Acoustic Radiation Measurement: A Study of Persian Musical Instruments," *Applied Sciences 2020, Vol. 10, Page 633*, vol. 10, p. 633, 1 2020.

[12] C. Gough, "Violin plate modes," *The Journal of the Acoustical Society of America*, vol. 137, no. 1, pp. 139–153, 01 2015.

[13] M. D. Stanciu, F. Dinulică, V. Bucur, V. G. Gliga, S. M. Nastac, and M. Câmpean, "Changing the vibrational behavior of the wooden thin arched plates—The maestro violins experimental study case," *Thin-Walled Structures*, vol. 174, p. 109042, 2022.

[14] A. Meza Pérez, S. Gonzalez, R. Viala, and C. Fritz, "Developing a finite element model for the study of an experimental batch of violins," in *10th Convention of the European Acoustics Association, Forum Acusticum 2023*, 09 2023.

[15] M. J. Elejabarrieta, A. Ezcurra, and C. Santamarıa, "Coupled modes of the resonance box of the guitar," *The Journal of the Acoustical Society of America*, vol. 111, no. 5, pp. 2283–2292, 05 2002. [Online]. Available: https://doi.org/10.1121/1.1470163

[16] A. Ezcurra, M. Elejabarrieta, and C. Santamaría, "Fluid–structure coupling in the guitar box: numerical and experimental comparative study," *Applied Acoustics*, vol. 66, no. 4, pp. 411–425, 2005.

[17] G. Bissinger, "Structural acoustics of good and bad violins," *The Journal of the Acoustical Society of America*, vol. 124, no. 3, pp. 1764–1773, 09 2008. [Online]. Available: https://doi.org/10.1121/1.2956478

[18] "COMSOL Multiphysics." [Online]. Available: https://www.comsol.com/comsol-multiphysics

[19] K. Pourtahmasi and A. Se Golpayegani, "Introducing Mulberry's wood (Morus alba L.) used in bowl shaped musical instruments of Iran," *Le bois: instrument du patrimoine musical*, vol. 29, 2009.

[20] F. P. L. (US), *Wood handbook: wood as an engineering material*. The Laboratory, 1987, no. 72.

[21] "Simcenter Testlab software." [Online]. Available: https://plm.sw.siemens.com/en-US/simcenter/physical-testing/testlab/

[22] A. A. Orouji, "Study of the effects of the shape and volume of Kamancheh's sound box on the output sound and vibration properties of the instrument," Master's thesis, IRIB University, 2019.

BbMuse: A Blackboard-Driven Framework for Real-Time Interactive Music

Fabian Ostermann
TU Dortmund University, Germany
fabian.ostermann@tu-dortmund.de

ABSTRACT

Interactive music systems are frequently built as ad hoc multi-agent architectures with custom communication protocols and project-specific execution models, while recent machine-learning approaches often encapsulate behavior in monolithic, computationally expensive black boxes. This paper revisits blackboard architectures for real-time interactive music generation and argues that composition and musical interaction can be modeled as distributed decision-making processes operating on shared musical state. We introduce BbMuse (BlackBoard MUSic Engine), an open-source, platform-independent Python framework that implements a dataflow-oriented blackboard variant inspired by real-time robotics. System state is encoded as typed representations on a global blackboard, while modules explicitly declare required and provided information, enabling automatic scheduling via topological sorting. As a result, system development becomes incremental and module-focused, since no inter-module dependencies must be specified. Further, the framework supports concurrent execution and demonstrates that real-time performance is possible with Python using native-library acceleration. We provide a growing collection of example projects, discuss diverse use cases and outline future features for learning-based module replacement as well as a GUI editor.

1. INTRODUCTION

Interactive and generative music systems are frequently implemented as multi-agent systems [1], in which autonomous components interact to jointly produce musical output. While efforts toward standardization exist [2], most systems are still designed ad hoc, with custom architectures, bespoke inter-agent communication protocols, and project-specific execution models. As a result, implementations are difficult to understand, reuse, extend, or compare.

At the same time, recent advances in machine learning have led to a contrasting trend: increasingly complex musical behavior is encapsulated within monolithic neural models [3]. While such approaches exhibit impressive results, they often function as black boxes, incur significant computational cost, and are poorly suited for real-time interaction or deployment on end-user hardware.

In this work, we propose revisiting the concept of blackboard systems and transferring it to real-time interactive music generation. Blackboard architectures have a long history and continue to be applied in real-time robotics and control systems. We argue that blackboards are particularly well suited for musical applications, as composition and musical interaction can be viewed as distributed hierarchical decision-making processes.

Based on this idea, we introduce B♭Muse [1], short for *BlackBoard MUSic Engine*, a platform-independent Python-based framework for implementing real-time interactive music systems using a blackboard-driven architecture. B♭Muse is designed to enable developers to combine diverse concepts in a modular manner within a unified execution model and is released [2] under the MIT license for unrestricted use in research, artistic, commercial, and educational contexts. In addition to the core engine, we provide a growing collection of example projects that serve as reusable templates and reference implementations for frequently required modules.

1.1 Motivation

Blackboard systems have proven effective in domains with comparable structural and temporal requirements, most notably real-time robotics [4], where complex behavior emerges from the coordination of multiple interdependent subtasks operating under strict time constraints. Therefore, striking conceptual similarities exist between real-time robotics and interactive music generation. The latter typically involves tasks such as audio and symbolic input analysis, phrase or material selection, harmonic and formal decision-making, and high-precision temporal control for playback. Each task itself may consist of multiple hierarchical subtasks that operate at different temporal and semantic levels. Similarly, robotic systems must integrate perception (e.g., visual object recognition), decision-making, and precise motor control, often in dynamically changing environments. In both cases, system performance depends on the coordinated interaction of heterogeneous processes rather than on a single monolithic model.

Recent work has identified several key challenges for modern AI-based music generation systems, including explainability, interactivity, adaptability, and control [3]. The latter two are directly addressed by blackboard architectures, which were explicitly designed for adaptive decision-making in dynamic environments [5]. Within a blackboard system, individual modules can contribute partial solutions based on shared knowledge, allowing global behavior to emerge from local decisions. Notably, data-driven approaches such as deep learning can be naturally integrated as individual modules, enabling their use in interactive settings while avoiding monolithic system designs.

Copyright: ©2026 Fabian Ostermann. This is an open-access article distributed under the terms of the Creative Commons Attribution License 3.0 Unported, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

[1] Suggested pronunciation: "*bee bee muse*"

[2] B♭Muse and examples are available at https://github.com/fabianostermann/bbmuse. The version described in this paper is archived at https://doi.org/10.5281/zenodo.19221871.

By combining such models with rule-based modules and explicit data representations, the overall system architecture also supports improved transparency and explainability.

This work will show how blackboard-driven interactive music systems offer:

- a divide-and-conquer strategy, reflecting the hierarchical and interdependent nature of musical decision-making,
- independent module development without explicit hard-coded inter-module dependencies,
- reuse of modules across projects without project-specific re-engineering,
- incremental extension of existing systems by adding or exchanging modules,
- adaptive behavior in dynamic contexts through incremental updates to shared state, and
- seamless integration of heterogeneous paradigms, allowing symbolic, rule-based, and data-driven modules to coexist and interact.

2. THE BLACKBOARD ARCHITECTURE

This section introduces the basic theoretical principles of traditional blackboard systems as introduced in [5, 6, 7].

The blackboard model emerged at a time when AI research was dominated by expert systems. Therefore, the first examples of blackboard systems include the HEARSAY-II system [8] for speech understanding and the real-time system HASP [6] that interprets sonar signals for ocean surveillance.

The blackboard model is based on a *metaphor*: a diverse group of experts sit in a room and collaborate using a wall-mounted blackboard as common workspace and only form of communication. The experts are constantly monitoring the blackboard, waiting for an opportunity to contribute partial solutions based on their individual knowledge. Each new contribution may enable further contributions by others. This iterative process continues until a satisfactory solution is reached.

The following terms were established for the three main components present in every blackboard architecture: (1) the *blackboard* itself that holds the structured data, (2) a finite number of *knowledge sources* (KS) that act as expert agents manipulating data on the blackboard, and (3) a *control* instance of some kind (sometimes called control shell) that activates knowledge sources to process data on the blackboard. Control may be implicit, e.g., if the knowledge sources are self-activating (the so-called *opportunistic* behavior) in a fully parallel processing setup. Most commonly, however, a dedicated controller decides which knowledge source to activate next based on the current blackboard state while ensuring fair scheduling. This classic setup is depicted in Figure 1.

The traditional blackboard architecture and its terminology were derived from theories of problem solving. Our approach is more related to control theory [5]. Therefore, our system design is based on a specific modern variant of blackboard systems, which is still used today as state-of-the-art technology in the field of real-time robotics (details follow in Section 4).

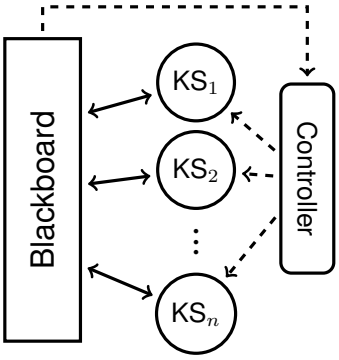


Figure 1. Schematic of a classic blackboard system with multiple knowledge sources (KS)

3. RELATED WORK

Blackboard systems have been applied to a variety of tasks including the aforementioned HEARSAY-II for speech understanding and HASP for real-time ocean surveillance, but also diverse modern applications such as unmanned vehicles [9], smart sustainable cities [10], or real-time legal decision-making for autonomous cars [11]. This section, however, focuses on musical systems that include blackboards as well as approaches that show implicit similarities to the blackboard pattern.

3.1 Blackboard systems for musical tasks

Applying blackboard systems to musical tasks, especially from the field of *music information retrieval* (MIR), is not new. However, such works are comparatively rare and fragmented. In the past, blackboard systems were mainly applied to traditional MIR tasks, most prominently to polyphonic music transcription [12, 13, 14], but also to auditory scene analysis [15] and musical structure analysis [16].

One of the earliest attempts at developing a musical blackboard system is [17]. It presents a theoretical sketch of a blackboard-based music composition system that models music knowledge with the goal of studying musical creativity. Another work is PRAIS, which is primarily about general real-time, distributed (parallel) knowledge-based reasoning. For the purpose of illustration, it includes a schematic and a short description of a hierarchical blackboard system for music generation [18, Fig.1]. [19] reports how blackboards were used for distributed music systems in live performances of the *computer network music* ensemble The Hub. In [20], a musical multi-agent system for interactive real-time accompaniment is designed that uses a blackboard architecture for inter-agent communication. It is fully MIDI-event-based and applies a music ensemble metaphor. The real-time music system from [21] uses a blackboard to share metadata tags about tracks from a pre-analyzed audio database between four musical agents to coordinate playback selection. The most recent work in this list is [22], which uses a blackboard-providing service agent in the context of a hardware quantum multi-agent system for interactive computer music.

Although the references above underline the applicability of blackboards to interactive music generation, none of the systems utilize the blackboard pattern nearly as extensively as we propose.

3.2 Distinction to multi-agent systems

Multi-agent systems (MAS) are a quite common approach to interactive music systems [1]. However, they usually do not incorporate the blackboard design pattern, but focus more on direct message passing and explicit communication [23]. Also, MAS and blackboard systems are not mutually exclusive. Systems may be described as:

- MAS with a blackboard: agents are autonomous, but share a blackboard as a common workspace.
- Blackboard with agents: knowledge sources are implemented as agents, but their autonomy is limited; they behave more like modular processors scheduled by blackboard control.

Two major differences might be identified in the attributes of *state* and *control*. In blackboards, the state is centralized, explicit and shared with all agents, whereas in MAS the state is typically distributed and private to each agent. Therefore, coordination is decentralized. Each agent may have its own decision-making process and thus takes individual actions, whereas blackboard systems usually aim to achieve a global decision. Because of that, a blackboard system as a whole can represent an agent itself in a meta system, which actually is a typical setup in the field of robotics. The line between MAS and blackboard systems, however, will always be blurry.

3.3 User-centered graph-based generative systems

The concept of visual, graph-based design is famously applied to generative music frameworks by Max and Pure Data [24]. The *dataflow programming* paradigm allows for modularity, but lacks declarative semantics. That means that wiring is fully manual and thus does not scale well. Text-based coding features are less important. Their strength lies in intuitive *visual programming*, which provides a lower entry barrier. While, in general, visual programming languages can be designed as general-purpose languages, text-based coding is more flexible, controllable and has stronger debugging capabilities. Therefore, only code makes it possible to efficiently implement any desired musical idea, but at the cost of beginner-friendliness, which of course can also be a highly valuable design goal for a generative music system.

As identified by [25], the interactive nature of systems such as OpenMusic, Max/MSP and Pure Data is based on the existence of a visual GUI to manipulate the structures of their computations at runtime. The same applies to text-based programming with *live coding* frameworks [26]. However, systems especially designed for real-time self-controlled interactive improvisation scenarios such as OMAX [27] or Dyci2 [28] are very domain-specific and do not allow for general-purpose use.

We propose that there is a missing piece that fits the gap between the desired properties to (1) have a flexible framework that has the potential to model every possible musical idea, but, at the same time, (2) bring clarity and manageability to the design of complex systems that pure text-based frameworks for music generation usually do not offer (cf. muspy [29] or SCAMP [30]) and therefore must be tediously achieved, e.g., by developing custom GUIs for every new project.

3.4 Modern machine learning approaches

It is also typical for modern music generators to use a deep learning-based single model approach that learns from huge amounts of example data. However, to date they often lack structural explainability, real-time capability, and flexibility for interactive modification [3]. Therefore, the idea of modularizing such AI models also emerged in the domain, but to date flexible interconnection is limited (e.g., ComfyUI[3]).

A new direction that uses shared knowledge bases on multi-agent LLM reasoning, which has already been applied to automatic music composition [31]. The core idea is quite similar to the blackboard model, but connections are usually not explicitly drawn. However, the use of terms such as "multi-agent conversation" [31] and "shared workspace" [32] could be interpreted as the implicit use of the blackboard pattern. However, classic blackboard systems typically emphasize state-triggered, opportunistic activation of knowledge sources by the controller and typed shared representations, which is usually not the case, except for [33]. And by leveraging existing LLMs, the shared protocol for interconnection between LLM agents is natural language, which makes handling musical objects noisy and inefficient. Interactive control is also limited with this approach.

4. FRAMEWORK DESIGN AND CURRENT FEATURE SET

This section describes the implementation-agnostic design of the B♭Muse framework. Implementation details will be given in Section 5.

The impetus for starting this project was that we found striking similarities between the two challenges of real-time robotics and interactive music generation. In both domains, we cope with complex systems that have to make hierarchical decisions in rapidly changing environments. Robots have to observe, decide, and act. Interactive music has to listen, compose, and play.

Therefore, we searched and found our blackboard architecture role model in a framework that was originally developed by the GermanTeam [34] for the so-called *RoboCup* events, where teams of robots are programmed to participate in a soccer tournament. As with the challenges, the main benefits (see Section 1.1) are also shared by both domains and have already been successfully demonstrated in robotics: the record world champion in robot soccer, *BHuman*, which emerged from the former GermanTeam, is using this exact framework successfully to this day [4].

The core principle is to modularize control behavior by applying a divide-and-conquer strategy. Named *representations* are introduced that live on a global blackboard and hold explicitly structured data. *Modules* that act as the formerly introduced knowledge sources, can then explicitly request access to specific *representations* (called a *require*) and state which representations they want to modify (called a *provide*). By allowing only one providing module per representation, the control flow becomes unidirectional. The controller's main responsibility is to execute the modules in an order that respects the dependencies implied by the *require* requests of the modules.

[3] https://www.comfy.org/

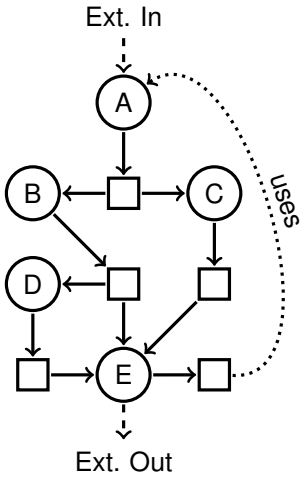


Figure 2. The streamlined blackboard system with modules (circles) and typed representations (boxes)

The formerly opportunistic activation of knowledge sources (cf. Figure 1) is replaced by the unidirectional data flow through modules. Figure 2 shows an exemplary configuration. The blackboard is divided into pieces, the so-called *representations*, which are depicted as boxes in Figure 2, while *modules* are depicted as circles. Outgoing edges pointing to boxes represent *provided* representations. *Require* requests are represented by edges that start at a box and end at a circle. The representations are explicit, structured, and semantically constrained data objects with a well-defined internal structure. They are initialized before first module execution.

With the explicit declaration of *provided* and *required* representations for each module, automatic inference of a valid execution order is possible by determination of the implicit module graph. This is the main desired property of this type of blackboard system, because it becomes very beneficial when the number of modules is high.

The concrete execution order can be calculated by topological sorting. This is only needed once at system start, so we chose Kahn's algorithm [35]. To apply it, we first determine the mapping of each providing module p to a list of consuming modules that require a representation that p provides. The time complexity is $O(V + E)$, where V is the overall number of modules, and

$$E = \sum_{\text{provider } p} (\text{num. of consumers of } p).$$

The execution order for a module graph is not necessarily unique. For Figure 2, valid orders are ABCDE, ABDCE and ACBDE. This ambiguity reveals that parallel execution is theoretically feasible. More about parallelization follows in Section 5.4.

As it is allowed that representations are required by multiple modules, it may also occur that no valid order exists, which means that the module graph has at least one cycle. If so, dependencies are in conflict. It may sometimes be necessary for a program's logic that cycles exist, meaning that at least two modules depend on each other's provided representations. To arrive at a valid execution order, the concept of *use* is introduced. A used representation has no update dependencies in the current iteration, with the trade-off that it may deliver outdated information from previous iterations (cf. dotted line in Figure 2).

Representation provides must be exclusive, meaning that only one providing module is allowed per representation, preventing write conflicts. Also, modules must provide at least one representation to be included in the execution order, which is a useful property for easy deactivation by just emptying the list of *provides* and for module exchange. If a module acts only as an interface to externals (see dashed lines in Figure 2), e.g., sending messages to a MIDI device or receiving from it, and therefore has nothing substantial to contribute to the blackboard, it is intended to simply provide a dummy representation. In practice, such representations are useful anyway, e.g., for logging purposes.

5. IMPLEMENTATION DETAILS

After presenting the implementation-agnostic system design decisions in Section 4, this section provides details of our Python implementation.

5.1 Project management

Since B♭Muse is an engine capable of executing multiple independent projects, projects are organized as self-contained directory structures that contain all project-related files. A directory is considered a valid B♭Muse project if it contains a project descriptor file [4] named `project.bbmuse` located at the directory root. When B♭Muse is launched from within such a directory, the engine automatically discovers and executes the corresponding project. Alternatively, an explicit project directory may be supplied via the command-line interface.

The process of modifying a project is intentionally designed to be agnostic to the core engine implementation. In the current workflow, projects are created and edited by directly modifying project files using a standard text editor [5]. At the same time, this design naturally supports the development of a B♭Muse-specific graphical editor, which can be implemented as an additional layer without requiring changes to the engine core (see Section 8.1).

Besides being an indicator for B♭Muse projects, the project descriptor file also holds all project-specific settings in TOML syntax, including a name and a brief description. By default, files that implement modules and representations are recursively searched in the subdirectories `modules/` and `representations/`, respectively. Internal module and representation names (cf. Figure 4, lines 2–4) are defined by the files' names based on the pattern `<module name>.py` and `<representation name>.py`. The default search paths can be changed inside the project descriptor file. However, the file can also remain empty, as the software design paradigm of *convention over configuration* is used. That paradigm also allows backward compatibility with future releases.

[4] Synonymous terms are: project manifest, sentinel file, or marker file

[5] i.e., B♭Muse projects can be developed using any IDE. For the development of the core engine and the accompanying example projects, VSCodium with Python extensions was used.

5.2 Module implementation and hook functions

Representation and module implementations are provided as Python files. Minimal examples are given in Figures 3 and 4, respectively. Each module declares the representations it requires and provides as lists of representation names which are used for automatic scheduling by the engine. The framework follows the software design paradigm of *inversion of control*. Activation of a module is handled by calling hook functions, which are comparable to system callbacks. They are to be called exclusively by the framework's controller instance and never from inside modules. To distinguish them from local function definitions, they always begin with a leading underscore.

The main function in every module is `_update(bb)`, which is called periodically by the controller respecting the execution order. The mandatory parameter `bb` represents the blackboard. Its content can be read and modified by the module. However, only the representations that a module provides are writable. To restrict write access to required and used representations, the passed object is a so-called *view* instance of the real blackboard. The view only exposes the subset of required, used and provided representations. All representations that are not declared as provided are considered read-only. Value assignment is prevented by overwriting Python's `__setattr__` function.

There are also two optional hooks: `_init()` can be used to initialize modules (e.g., to open MIDI devices, create ports/sockets, or configure external libraries). If global variables are to be assigned from inside functions, the `global` keyword must be used.[6] `_close()` is intended for safe shutdowns (e.g., to free memory, close devices, or save logs to disk). The framework always attempts to close all modules before any program exit, even if caused by runtime errors.

Besides the hook functions, modules are allowed to have local functions to organize subroutines. It is important to note that internally modules and representations are not instantiated as Python classes, but by dynamic imports from user-specified file paths using Python's *importlib*. Modules and representations are therefore loaded as Python *modules*[7], which allows implementations of simple modules to be very short (cf. Figure 4), e.g., there is no need for modules to explicitly inherit from a common base class or to write a class header at all. However, when we load module files following this *metaprogramming* approach, users can use any feature the Python programming language offers without any additional framework code.

Another feature is that representation objects can also define functions. This allows for providing utility functions that can be required by multiple modules and such functions that live on a representation could be modified at runtime by their providing module, thus making full use of *functional programming* in Python.

[6] `global` is a compile-time directive that efficiently reserves the local namespace slot for the given variable. Alternatively, Python's `SimpleNamespace` from *types* may serve as an elegant solution if a module needs many internal variables.

[7] Potential for confusion: modules in Python are a language-specific concept that is not related to modules in B♭Muse.

5.3 Why Python?

A reasonable question is why we chose Python for a framework that is to be applied to real-time generation, as Python is often perceived as slow. The answer is manifold.

Python allows for *rapid prototyping*. New musical ideas can be directly tested with far fewer lines of code than would be needed in C++ or Rust. That aligns well with blackboard architectures, as they are also known to allow for *rapid prototyping* and *exploratory programming* [36]. Also, many programmers are familiar with Python. Therefore, we decided against the design of a domain-specific language that would also have meant we had to develop our own compiler. On the contrary, we inherit all of Python's language features such as list comprehension, generator expressions, context managers and lambda expressions, by applying the aforementioned metaprogramming approach with dynamic imports (cf. Section 5.2) instead. Furthermore, in a blackboard-driven interactive music system, many modules exist that only contain deterministic control logic, probabilistic decisions, input/output callbacks and similar. On a modern machine, all rule-based and chance-based calculations are fast enough to be implemented in pure Python.

If needed, optimization can be applied. Python is often underestimated when it comes to performance optimization. It provides extensive features for the integration of native libraries. Examples include NumPy for numerical CPU-based calculations, PyTorch (and ONNX) for GPU-based inference, OpenAL bindings, and even the integration of custom C++/Rust extensions is strongly supported. Therefore, Python code in high-performance B♭Muse projects can follow the *orchestration* design pattern.

Another major advantage of using Python is that the import of any third-party Python libraries is possible inside B♭Muse modules (cf. Figure 4, line 6), ranging from machine learning and control (PyTorch, TensorFlow, JAX, scikit-learn), probabilistic modeling and optimization (numpy.random, scipy.stats, pymc), and data handling (pandas, polars), to music- and audio-specific tooling such as MIDI and real-time I/O (mido, python-rtmidi, python-osc), signal processing and analysis (librosa, scipy.signal, aubio, madmom), synthesis/playback interfaces (sounddevice, pyaudio, pyfluidsynth), and symbolic music representations and rendering (mido, music21, muspy, pretty_midi, SCAMP, abjad). Concrete plans for a future engine feature including machine learning will be discussed in Section 8.2.

Finally, the installation process[8] is also simple and platform-independent with Python. The B♭Muse core has no dependencies on any third-party libraries. Therefore, any CPython-compatible compiler can run the source code directly. Additionally, we made B♭Muse available via *pip*.[9] Individual dependencies of B♭Muse projects can be managed using virtual environments (e.g., *venv* or *conda*), which is also a common, well-integrated Python workflow.

5.4 Parallel execution with groups

The linear execution order has its advantages, however, in some cases, certain modules need to run faster than oth-

[8] Information for an immediate first-run experience, see the README in the project repository.

[9] Install the PyPI package with the following command:
`pip install bbmuse`

```
value = 5

def _validate():
    # check value constraints
    assert 0 < value < 10
```

Figure 3. Example definition of a minimal representation

```
GROUP = "<group-name>"
USES = [ "<representation-name>", .. ]
REQUIRES = [ "<representation-name>", .. ]
PROVIDES = [ "<representation-name>", .. ]

import <python-package>

internal_value = None

def _init(): # [optional]
    # initialize module state
    global internal_value
    internal_value = 1

def _update(bb):
    # process module logic
    # update data on blackboard
    r = bb.<representation-name>
    r.value = internal_value

def _close(): # [optional]
    # free memory / save to disk
```

Figure 4. Example module implementation showcasing main features

ers. One example is real-time demands such as MIDI event polling and scheduled sending. Another example is that while improvisational elements may need immediate output modifications to really feel interactive, more musically composed elements may benefit from having more calculation time available for planning. For the latter, of course, *time-sliced computation* could be applied. But this would need a lot of project-specific code engineering. Therefore, B♭Muse provides the concept of *module grouping*.

Members of a group are executed in a shared thread. The global execution order is consulted, but executions of modules that are not in the group are skipped. To guarantee execution while preventing conflicts, every module is assigned to exactly one group. A default group is used if no group is explicitly declared. In this setup, faster-running threads will read representations from slower-running threads without any new updates. However, in blackboard systems, that is no problem as module logic should only depend on the latest state of the blackboard. Conversely, slower-running modules may miss intermediate information. However, they will always have the latest information available. If information is cumulative, representations can also be equipped with buffers.

In module files, group membership is declared by adding the string variable `GROUP` (cf. Figure 4, line 1). This allows development to be highly exploratory as configurations can be tested without the need to write complex threading-related code. Internally, concurrent execution of groups is implemented with Python's *threading* module. Note that this does not enable load distribution across multiple CPU cores as Python's standard threading model uses the *Global Interpreter Lock* (GIL). Real multi-core parallelization (in Python called *multiprocessing*) is possible but not yet implemented in B♭Muse. The main problem: Sharing representation objects between processes is costly and may be even less efficient than threading for low to moderate program complexity. The optimal solution would be to instantiate the blackboard in shared memory. However, this would require a robust mechanism for automatic memory allocation, as manual allocation would necessitate extensive project-specific engineering, thereby undermining the plug-and-play nature of B♭Muse. However, native libraries such as NumPy and PyTorch normally release the GIL internally for heavy calculations, such as vectorized array operations, convolutions and linear algebra, as it is common that they are linked against a multi-threaded backend (OpenBLAS/MKL). This workflow of orchestrating native libraries from Python threads with the GIL in place proved to be surprisingly efficient and can be recommended. Alternatively, a module may offload computation by spawning its own separate OS process in `_init()` using Python's multiprocessing. A template of this approach is provided in the supplementary example collection.

5.5 Debugging and (live) performance mode

The engine of B♭Muse can be started in different run modes: debug, normal, perform. The *debug* mode enables detailed logging and activates the optional hook function `_validate()` in representation files. The intended use case: check value constraints and raise assertion errors when invalid values are detected.

The default is running in *normal* mode. When runtime errors occur, exceptions are caught by the controller but an immediate safe shutdown of the engine is triggered, which includes calling `_close()` on all modules.

As B♭Muse is intended for live music performances (see Section 7), the *perform* mode is intended to be enabled for long-running installations or live performances, that can hardly be tested thoroughly beforehand or not fully at all for scenarios where free improvisation is involved. Therefore, the *perform* mode suppresses runtime exceptions, and if group threads crash, they are immediately restarted. Further, the garbage collector is disabled during the main update loop to increase performance.

The Python compiler's own optimization flag `-OO`, however, has proven to be ineffective in our scenario, because it only skips assertions and strips docstrings, and therefore only improves efficiency when debug asserts are heavily used or memory footprint is important. Neither issue is critical here. For future improvements of B♭Muse, we will determine if other engine-based optimization strategies are useful.

Because the code inside modules' update functions is run over and over again, testing *PyPy* as an out-of-the-box alternative to CPython is promising. With its just-in-time hot-path compilation concept, it may help to speed up projects that use a lot of pure Python code. Similarly, Cython can be used to directly write efficient code in modules that is compiled to C. Therefore, the B♭Muse engine is also implemented to be Cython-compatible.

6. TECHNICAL AND PRACTICAL ADVANTAGES

As can be inferred from Section 5, the main advantage of B♭Muse is that it greatly supports the development process

of interactive music systems by automatically resolving and scheduling inter-module dependencies. Modules are replaceable and swappable without re-engineering the full system (even during runtime). The implementation of single modules can be kept short and functionality is therefore easier to comprehend. This again makes modular extension and reuse of existing systems easier. Interactive music systems can be built incrementally and evolve over time.

Beyond these properties, B♭Muse also supports explainability: musical decision-making is decomposed into readable inter-connected subroutines, so that global behavior can be traced back to local contributions, which is a vanishing property in the modern AI landscape. Moreover, single monolithic models are usually far more computationally expensive. Many subtasks in interactive music (e.g., gating, timing, constraint enforcement, deterministic mapping, simple stochastic choice) can often be implemented as lightweight rules or heuristics at substantially lower runtime cost than neural inference. The flexibility of module interchange in our framework is therefore increasingly important considering the modern AI landscape. The ease of swapping different module candidates to determine whether a complex AI model is needed for a specific subtask or if a rule-based solution might achieve similar results in the context of a complex system enables developers to lower not only execution times but also power consumption. As a result, systems can be incrementally analyzed, explained, modified, and improved without explicit restructuring.

7. USE CASES

7.1 Live performance scenarios

B♭Muse's most straightforward application scenarios are interactive live performances and real-time improvisations. It can be run onstage on any decent laptop, analyzing audio input and producing responses via a connected audio interface. For event-based processing, connection to hardware devices can be established over hardware MIDI interfaces (e.g., using mido). Through hardware loopbacks or virtual ports, any software synthesizer, generative frameworks such as Pure Data or SuperCollider, as well as live DAWs (Ableton, Bitwig) can be integrated. Also, running multiple instances of a B♭Muse program works out of the box.

7.2 Embedded/installation contexts

B♭Muse runs on any platform where a Python interpreter is available. This way, a program can be developed and tested, e.g., on a laptop or home computer, and then run as a headless program on embedded devices such as a Raspberry Pi. Even limited CPU performance can be simulated by limiting the frequency of the main update loop.

7.3 Adaptive scoring for interactive non-linear media/video games

Nowadays, most games are developed using a *game engine* that provides game-specific features such as graphics rendering, physics or input handling and also allows compilation for various platforms. While basic sound effects and audio buses are common in modern engines, adaptive music tools are not. Therefore, specialized middleware such as Wwise or FMOD exists. Those solutions excel at real-time parameter-driven playback, layering, and resequencing of pre-rendered musical material, but are not designed to function as generative or compositional models. Their market leadership stems from their availability as native, well-integrated libraries across virtually all major game engines and target platforms.

B♭Muse, in the current state of development, cannot be shipped as a native library. The recommended approach to connect B♭Muse to any game engine is via local networking (socket communication) as a platform-independent method for sending messages between processes. All modern game engines provide network features. The same applies to the B♭Muse side: `socket` is a built-in library in CPython. Therefore, a B♭Muse program can serve as a local music server. A remote network server can be implemented in the same way.

7.4 Miscellaneous

In its current version, the core engine of B♭Muse is not restricted to musical applications. A wide range of applications can potentially be supported, including visual and generative art systems, game AI, procedural content generation, or multimodal interactive installations, as well as any other application that benefits from being modeled with the specific blackboard architecture that B♭Muse provides. The B♭Muse project will be extended by music-specific modules and features. However, we will always keep the core code separate to allow for application to non-musical domains.

8. PLANNED FEATURES

Two major extensions of B♭Muse are currently in development. Both are developed as optional libraries, so that the B♭Muse core has no additional dependencies and can always be run independently.

8.1 Graphical User Interface

As B♭Muse operates on the implicit module graph, this graph serves as an ideal basis for a GUI editor design. The current version already includes a simple static visualization utility that, on request, creates a PNG file in the project's root directory using matplotlib and networkx.

The GUI's main features will be project management (create, configure, modify), module monitoring (run state, execution time, update frequency), intuitive handling of inter-module connections (visualize requires and provides), on-click opening of module and representation files (no need to browse directories), and easy access to module-specific logs and debugging features. Additionally, the GUI may provide a collection of basic template modules (e.g., basic input-output and timing utilities, data transformation, etc.). The goal is a platform-independent lightweight GUI also implemented in Python for reasons of compatibility and easy integration. Therefore, we are considering using the web-based UI framework *NiceGUI*.

8.2 Replacing Modules using Reinforcement Learning

Another motivation for this entire project was a concrete idea for a flexible integration of modern AI models into such blackboard systems, not only as fixed components of individual modules. Specifically, input and output of

modules are well-defined by the *requires* and *provides* statements, and the data structures of representations are explicitly typed. Therefore, this data can be collected and used in supervised or reinforcement learning settings. For example, it allows training a model that mimics a module's original (rule-based) behavior using *behavioral cloning*. Further, we can then take that model and *fine-tune* it based on interactive user feedback, which is a technique known as *reinforcement learning from human feedback*. Both can be performed offline, prior to a live performance, or online, during a performance. Our vision is to provide such features for applying learning algorithms to modules in B♭Muse. Again, the decision to use Python, which currently is the most common language for machine learning libraries, makes it easy to integrate learning features.

9. CONCLUSIONS

We revisited blackboard systems for real-time interactive music generation and presented them as a modular alternative to ad hoc architectures and monolithic black-box models. Based on this idea, we introduced B♭Muse, a platform-independent Python-based framework that implements a blackboard-driven execution model for interactive music systems. By structuring shared state as typed representations and expressing inter-module dependencies via explicit *requires*/*provides* declarations, B♭Muse enables automatic scheduling and straightforward extensibility while remaining compatible with heterogeneous approaches, including data-driven components. Overall, B♭Muse provides a practical foundation for building complex interactive music systems that are easier to understand, reuse, and adapt.

For users without a programming background, the programming skills required to use B♭Muse effectively are currently high. Therefore, the example and template collection will be extended and the GUI feature under development will also increase clarity during project development. However, understanding and applying blackboard-driven system design in general can certainly help developers in the design and implementation of complex interactive music systems.

10. REFERENCES

[1] K. Tatar and P. Pasquier, "Musical agents: A typology and state of the art towards Musical Metacreation," *JNMR*, vol. 48, no. 1, pp. 56–105, 2019.

[2] O. Bown, B. Carey, and A. Eigenfeldt, "Manifesto for a Musebot Ensemble: A platform for live interactive performance between multiple autonomous musical agents," in *ISEA*, 2025.

[3] J.-P. Briot, G. Hadjeres, and F.-D. Pachet, *Deep Learning Techniques for Music Generation*. Springer, 2020.

[4] T. Röfer, T. Laue *et al.*, "B-Human Code Release Documentation 2025," 2025, https://docs.b-human.de/coderelease2025/.

[5] B. Hayes-Roth, "A Blackboard Architecture for Control," *Artificial Intelligence*, vol. 26, no. 3, pp. 251–321, 1985.

[6] H. P. Nii, "The Blackboard Model of Problem Solving and the Evolution of Blackboard Architectures," *AI Magazine*, vol. 7, no. 2, pp. 38–53, 1986.

[7] D. D. Corkill, "Blackboard Systems," *AI Expert*, vol. 6, no. 9, pp. 40–47, 1991.

[8] L. D. Erman, F. Hayes-Roth, V. R. Lesser, and D. R. Reddy, "The Hearsay-II speech-understanding system: Integrating knowledge to resolve uncertainty," *Computing Surveys*, vol. 12, no. 2, pp. 213–253, 1980.

[9] M. E. Shin, T. Kang, and S. Kim, "Blackboard Architecture for Detecting and Notifying Failures for Component-Based Unmanned Systems," *Journal of Intelligent & Robotic Systems*, vol. 90, pp. 571–585, 2018.

[10] H. Orchi, A. B. Diallo, H. Elbiaze, E. Sabir, and M. Sadik, "A Contemporary Survey on Multisource Information Fusion for Smart Sustainable Cities," *Inform. Fusion*, vol. 114, 2025.

[11] L. Szymanski, B. Sniezynski, and B. Indurkhya, "A multi-agent blackboard architecture for supporting legal decision-making," *Computer Science*, vol. 19, no. 4, 2018.

[12] K. D. Martin, "Automatic Transcription of Simple Polyphonic Music," in *3rd Joint Meeting of the Acoustical Societies of America and Japan*, 1996.

[13] G. Monti and M. Sandler, "Automatic Polyphonic Piano Note Extraction using Fuzzy Logic in a Blackboard System," in *DAFx-02)*, 2002.

[14] J. P. Bello Correa, "Towards the automated analysis of simple polyphonic music: a knowledge-based approach," Ph.D. dissertation, Queen Mary, University of London, 2003.

[15] D. Godsmark and G. J. Brown, "A blackboard architecture for computational auditory scene analysis," *Speech Communication*, vol. 27, pp. 351–366, 1999.

[16] C. Fox, "ThomCat: A Bayesian Blackboard Model of Hierarchicel Temporal Perception," in *FLAIRS*, 2008.

[17] O. E. Laske, "Music composition as hypothesis formation: a blackboard concept of musical creativity," *Knowledge-Based Systems*, vol. 3, no. 1, pp. 36–41, 1990.

[18] D. Goldstein, "Extensions to the Parallel Real-Time Artificial Intelligence System (PRAIS) for fault-tolerant heterogeneous cycle-stealing reasoning," in *CLIPS*. Johnson Space Center: NASA, 1991.

[19] S. Gresham-Lancaster, "The Aesthetics and History of the Hub: The Effects of Changing Technology on Network Computer Music," *Leonardo Music Journal*, vol. 8, pp. 39–44, 1998.

[20] R. D. Wulfhorst, L. Nakayama, and R. M. Vicari, "A multi-agent approach for musical interactive systems," in *AAMAS*, 2003, pp. 584–591.

[21] A. Eigenfeldt and P. Pasquier, "Negotiated Content," in *ICCC*, 2001, pp. 27–32.

[22] A. Kirke, "Testing a hybrid hardware quantum multi-agent system architecture that utilizes the quantum speed advantage for interactive computer music," *JNMR*, 2020.

[23] L. F. Thomaz and M. Queiroz, "A Framework for Musical Multiagent Systems," in *SMC*, 2009, pp. 119–124.

[24] M. Puckette, "Pure Data," in *ICMC*, 1996, pp. 269–272.

[25] A. R. J. Francois and E. Chew, "An Architectural Framework for Interactive Music Systems," in *NIME*, 2006, pp. 150–155.

[26] A. F. Blackwell, E. Cocker, G. Cox, A. McLean, and T. Magnusson, *Live Coding: A User's Manual*. MIT Press, 2022.

[27] G. Assayag, G. Bloch, and M. Chemillier, "OMAX-OFON," in *SMC*, 2006.

[28] J. Nika, K. Déguernel, A. Chemla, E. Vincent, and G. Assayag, "DYCI2 agents: merging the "free", "reactive", and "scenario-based" music generation paradigms," in *ICMC*, 2017.

[29] H.-W. Dong, K. Chen, J. McAuley, and T. Berg-Kirkpatrick, "MusPy: A Toolkit for Symbolic Music Generation," in *ISMIR*, 2020.

[30] M. P. Evanstein, "SCAMP: Suite for Computer-Assisted Music in Python," Ph.D. dissertation, University of California, Santa Barbara, 2019.

[31] Q. Deng, Q. Yang, and R. Yuan, "ComposerX: Multi-Agent Symbolic Music Composition with LLMs," in *ISMIR*, 2024.

[32] Y. Zhang, A. Maezawa, G. Xia, K. Yamamoto, and S. Dixon, "Loop Copilot: Conducting AI Ensembles for Music Generation and Iterative Editing," *arXiv:2310.12404 preprint*, 2024.

[33] B. Han and S. Zhang, "Exploring Advanced LLM Multi-Agent Systems Based on Blackboard Architecture," *arXiv:2507.01701 preprint*, 2025.

[34] T. Röfer, J. Brose, D. Göhring, M. Jüngel, T. Laue, and M. Risler, "GermanTeam 2007," in *RoboCup Preproc.*, 2007.

[35] A. B. Kahn, "Topological sorting of large networks," *Comm. of the ACM*, vol. 5, no. 11, pp. 558–562, 1962.

[36] H. P. Nii, "Intro., 2.4. Rapid Prototyping," in *Blackboard Architectures and Applications*. Academic Press, 1989.

Listening Across Spaces: Perceptual Evaluation of an Ambisonics-Based Sound Installation

Felipe Otondo
Arts and Technology Lab
Institute of Acoustics
Universidad Austral de Chile
felipe.otondo@uach.cl

Leonardo Santos
Arts and Technology Lab
Institute of Acoustics
Universidad Austral de Chile
leonardo.santos@alumnos.uach.cl

ABSTRACT

This paper explores how immersive listening to natural soundscapes is shaped by the spaces in which it unfolds. Using second-order Ambisonics field recordings rendered through a third-order Ambisonics decoding scheme, five natural soundscape excerpts were reproduced over calibrated 16-loudspeaker Genelec arrays in two contrasting venues: an acoustically controlled laboratory and an untreated museum gallery. *Listener evaluations addressed presence, envelopment, timbral clarity, stability and depth using a perceptual framework grounded in recent immersive audio literature. The results reveal distinct perceptual profiles across venues, where spatial precision emerges in controlled conditions and reverberation contributes to a more diffuse sense of overall immersion in the museum. The study highlights immersion as a situated experience shaped by sound content, room acoustics, and reproduction conditions, with implications for artistic sound installations and exhibition design.*

1. INTRODUCTION

Immersive audio technologies are increasingly used in artistic, scientific, and educational contexts to create compelling listening experiences that situate audiences inside complex and rich acoustic environments [1]. Despite significant advances in higher-order Ambisonics (HOA), multichannel loudspeaker arrays, and virtual reality platforms, the relationship between physical reproduction conditions and the subjective experience of overall immersion remains insufficiently explored [2]. Ambisonics is particularly well suited to address this gap, as it enables the reproduction of natural field recordings with high spatial fidelity while remaining independent of specific loudspeaker layouts. This independence is especially relevant for sound installations that circulate between acoustic laboratories and public venues, such as museums or concert halls, where loudspeaker placement, room volume and background noise can vary considerably. For artists and engineers working with environmental recordings, understanding how these variables affect listeners' overall immersion is crucial both for technical design and for curatorial decisions.

Copyright: © 2025 First author et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License 3.0 Unported, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

The work reported here forms part of the *Soundlapse* project, a broader research initiative focused on creative and educational applications of natural soundscapes in southern Chile [3]. Seeking to bridge acoustic ecology, immersive audio and creative learning, the project has developed an extensive corpus of multichannel recordings from wetlands, coastal zones and native forests. This material has been applied across diverse contexts, including large-scale bird monitoring, acousmatic concerts, museum sound installations and pedagogical activities with primary school children [4]. This study focuses on a subset of recordings and proposes a structured methodology for evaluating immersive listening in two contrasting venues: an acoustically treated laboratory at Universidad Austral de Chile and the museum *Centro de Interpretación Patrimonial De Todas Las Aguas del Mundo* (CIPTLAM) in Valdivia, Chile, examining how room acoustics and reproduction conditions jointly shape listener ratings across five natural soundscape excerpts. Rather than isolating room acoustics as a single independent variable, the study compares two ecologically valid reproduction contexts representative of how Ambisonics-based sound installations are deployed across laboratory and exhibition settings.

2. IMMERSIVE SOUNDSCAPE REPRODUCTION

2.1 Ambisonics recordings and reproduction

Ambisonics represents the sound field around a listening position as a weighted sum of spherical harmonics, thereby decoupling the sound field description from any specific loudspeaker configuration. In practice, the Ambisonics signal is encoded in an intermediate format (Ambix) and decoded in real time to the specific loudspeaker layout available at each venue. This decoding typically involves computing a matrix of gain coefficients that maps Ambisonics channels to loudspeakers, optionally combined

with frequency-dependent weighting to reduce truncation errors and preserve timbral balance across the listening area.

In the present work, the sound scenes were captured using a second-order Core Sound OctoMic microphone and stored in Ambix format. For playback, these recordings were reproduced using a third-order Ambisonics decoding workflow adapted to each loudspeaker configuration. Accordingly, the recording stage was second order, while the reproduction stage relied on a third-order decoding scheme. Accordingly, the recording stage was second order, while the reproduction stage relied on a higher-order decoding strategy that effectively upscaled the material for loudspeaker rendering in each case.



Figure 1. Second-order Ambisonics OctoMic microphone used for field recordings in this study.

2.2 Immersion, Envelopment and Overall Listening Experience

The term "immersion" has been used in a wide range of contexts, often conflated with related concepts such as envelopment, presence or narrative transportation. Recent work distinguishes between immersion as an objective property of a system—its ability to deliver rich, multisensory stimulation and isolate the user from the physical world—and immersion as a subjective psychological state, characterized by focused attention and a feeling of being surrounded or engaged by the mediated environment [5]. In spatial audio, envelopment is typically defined as a perceptual attribute related to the extent to which sound energy arrives from all directions, whereas immersion refers to a higher-level cognitive and affective state that may or may not correlate directly with envelopment.

The notion of immersion adopted in this study aligns with recent literature that conceptualizes immersive experiences as multidimensional phenomena involving sensory, cognitive and emotional dimensions. Agrawal and Bech describe immersion as the psychological sensation of "being inside" a mediated environment, produced by the alignment of perceptual cues such as spatial coherence, clarity, environmental realism and emotional engagement [6]. In audio-only contexts, immersion is shaped by the interaction between spatial fidelity, listener interpretation, and affective response to the acoustic material [7]. Ambisonics addresses these dimensions as a scene-based spatial audio technique, offering a flexible and portable framework for representing sound fields with high spatial continuity. Its success, however, depends on multiple factors, including the recording order, the geometry and distribution of the loudspeaker array, room acoustics, calibration and equalization. Previous research identified relevant perceptual attributes for evaluating Ambisonics spatialization in controlled environments; here we extend this work to a non-ideal museum space.

3. LISTENING TESTS

3.1 Methodology

Two reproduction spaces were selected to investigate how room acoustics influence perceptual evaluations of immersive soundscape reproduction. The first space, the Arts and Technology Lab at Universidad Austral, is an acoustically controlled environment characterized by a reverberation time of approximately 0.25 seconds. It features a 16.1 loudspeaker array distributed across three vertical layers, allowing near-complete three-dimensional decoding of second-order Ambisonics (Figure 2). In this space, spatial cues—such as localization, depth, envelopment, and the stability of the auditory scene—can be assessed with minimal interference from reflections. The second space, the CIPTLAM museum, presents a contrasting context: an acoustically untreated gallery used for audiovisual exhibitions, with a reverberation time of approximately 1.2 seconds and a single-layer ring of sixteen loudspeakers (Figure 3). Although both venues used 16-loudspeaker Genelec systems, the laboratory employed a multilevel array and the museum a single-layer ring; thus, the comparison reflects realistic playback contexts where room acoustics and reproduction geometry vary together. This distinction is particularly relevant for attributes such as presence and envelopment, which may be influenced by the availability of vertical spatial cues.

To assess how these acoustic spaces affected overall sonic immersion, five perceptual attributes were selected: presence, envelopment, timbral clarity, spatial stability and depth. These attributes align with established definitions: presence (sense of being there), envelopment (sense of being surrounded), timbral clarity (transparency of the scene), stability (robustness of the spatial image under head movement), and depth (perceived distance and near–far structure).

Sixteen trained listeners, all acoustic engineering students with experience in critical listening, participated in the tests. Participants underwent a training phase to familiarize

themselves with the perceptual attributes and a continuous 1–10 rating scale. This ensured consistent internal references and reduced inter-listener variability.

The stimuli consisted of second-order Ambisonics field recordings made in coastal, forest and wetland environments. These recordings were selected for their rich spatial features (e.g., layered ambiences, lateral motion, and vertical activity). The same stimuli were evaluated in both spaces, with playback levels carefully matched and presentation order randomized to minimize contextual bias. Participants rated each attribute immediately after listening, which allowed for fine-grained perceptual comparisons between spaces.



Figure 2. Listening test setup at the Arts and Technology Lab, Universidad Austral de Chile.

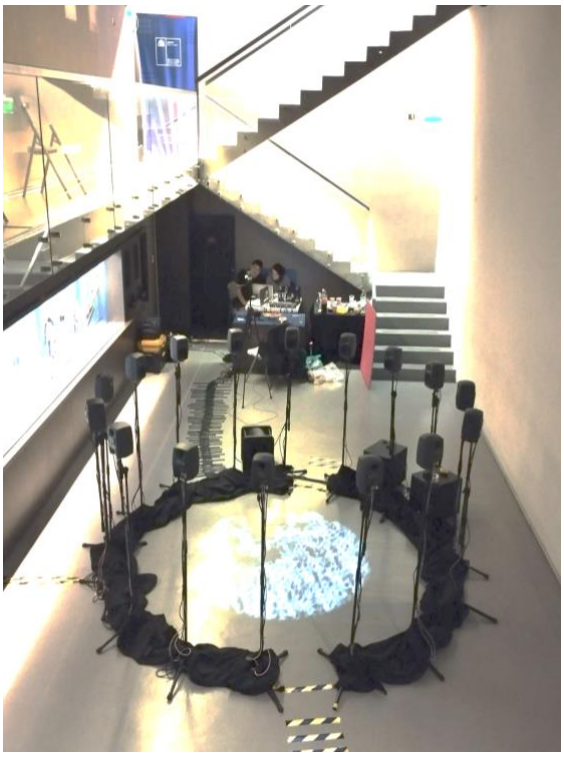


Figure 3. Listening test setup at the CIPTLAM museum in Valdivia, Chile.

By integrating controlled and real-world listening environments, the methodology enabled a nuanced assessment of how room acoustics shape immersive experiences, with insights that informed both the artistic and technical design of the final installation, including decisions related to material selection, playback equalization, moderation of broadband density, and spatial composition strategies better suited to reverberant spaces.

3.2. Results

The analysis of the results draws on both the mean subjective ratings summarized in Table 1 and the ANOVA outcomes reported in Table 2, revealing consistent perceptual differences between the two listening environments and across soundscape types. Overall, ratings were higher in the acoustically controlled laboratory than in the museum for most perceptual attributes, although the magnitude of these differences varied depending on both the attribute evaluated and the characteristics of the recordings. Presence was primarily stimulus-driven and showed limited sensitivity to room acoustics, with Beach (0 m) and Coastal Forest recordings receiving the highest ratings in both environments and Wetland recordings scoring lowest. No meaningful interaction between room and recording type was observed for this attribute.

In contrast, Envelopment exhibited a clearer dependence on the listening space. As shown in Table 1, ratings remained high and stable in the laboratory, whereas in the museum they decreased more noticeably for spectrally broad and laterally active recordings, indicating a moderate interaction with room acoustics. Timbral Clarity followed a similar trend, with consistently higher ratings in the lab and a general reduction in the museum, particularly for dense or broadband soundscapes. Stability was also rated higher in the laboratory, with spatial coherence degrading in the museum across most stimuli, especially for more complex or distant sources. Depth showed the strongest room-dependent effect: recordings containing salient near–far cues achieved substantially higher ratings in the laboratory, while these cues were partially masked in the museum environment. The ANOVA results in Table 2 corroborate these trends, confirming robust room effects for Envelopment, Timbral Clarity, Stability, and Depth, while Presence remained comparatively less affected by the listening environment.

Field recording	**High. forest**		**Beach (0 m)**		**Beach (80 m)**		**Coastal forest**		**Wetland**	
Location	L	M	L	M	L	M	L	M	L	M
Presence	7.7	7.1	8.2	8.0	7.2	6.2	8.1	7.7	6.8	6.0
Envelopment	8.2	7.2	8.5	8.1	7.3	5.9	8.7	8.2	7.6	6.4
Timbral clarity	9.5	8.4	8.8	7.8	8.3	7.1	9.0	8.2	8.7	8.4
Stability	9.3	8.5	7.5	6.9	6.4	5.6	9.2	8.7	7.7	6.9
Depth	9.2	7.3	8.7	6.9	8.3	7.3	8.0	6.1	9.2	8.5

Table 1. Mean subjective ratings (1–10) for each field recording and listening space (Lab (L) and Museum (M)).

Acoustic attribute	Source	F	p-value	η^2 (ges)
Presence	Room	3.9	0.0664	0.0471
	Recording	13.1	**< 0.0001**	0.2236
	Interaction	2.1	0.1131	0.0139
Envelopment	Room	13.5	**0.0022**	0.1327
	Recording	17.3	**< 0.0001**	0.2614
	Interaction	2.7	0.0602	0.0263
Timbral clarity	Room	11.7	**0.0038**	0.0990
	Recording	4.5	**0.0177**	0.0895
	Interaction	2.3	0.1062	0.0146
Depth	Room	48.6	**< 0.0001**	0.3112
	Recording	11.1	**< 0.0001**	0.2134
	Interaction	6.3	**0.0013**	0.0543
Stability	Room	5.6	**0.0313**	0.0697
	Recording	33.5	**< 0.0001**	0.4215
	Interaction	0.2	0.8776	0.0016

Table 2. ANOVA results across venues and field recordings. Statistically significant p-values are shown in bold.

4. CONCLUSIONS AND FUTURE WORK

Overall, the results indicate that immersive soundscape reproduction is strongly shaped by the conditions under which it is presented. Across the two venues studied here, presence proved relatively robust, whereas more fine-grained spatial attributes such as depth, envelopment, timbral clarity, and stability were more sensitive to the combined effects of room acoustics and playback configuration. These findings have important implications for the deployment of Ambisonics-based installations in public and exhibition spaces, where perceptual fidelity may depend not only on the acoustic response of the room but also on how reproduction layouts are adapted to the site.

The differing loudspeaker configurations between the laboratory and the museum represent a relevant limitation of the study, as the use of a three-layer system in the laboratory and a two-dimensional setup in the museum may have influenced perceptual outcomes, particularly for Presence and Envelopment, which are closely linked to vertical spatial cues. Future work should prioritize the implementation of comparable multilevel loudspeaker configurations across listening environments or systematically control and isolate the role of vertical reproduction, in order to better disentangle room-related effects from reproduction system effects and strengthen the robustness of comparative evaluations. In this context, the role of loudspeaker equalization will also be examined as a contributing factor.

In connection with this study, an artistic sound installation was designed and implemented at the museum, taking as a starting point several of the ideas and results of the experiments documented here. Future work will involve exploring the relationship between listening test outcomes and survey responses collected from visitors to the installation. Early results show that high levels of perceived overall immersion and engagement can be achieved in exhibition contexts even when specific spatial attributes such as depth and stability are perceptually reduced by room acoustics. This suggests that affective response, ecological plausibility and curatorial framing play a critical role in how audiences integrate immersive soundscapes as coherent experiences. Further analysis will aim to more explicitly correlate attribute-based listening test results with experiential survey data, strengthening the methodological bridge between controlled perceptual evaluation and artistic practice.

Acknowledgments

This work was supported by the Chilean National Fund for Scientific and Technological Development (FONDECYT, Chile) under Grant 1250388.

5. REFERENCES

[1] J. Paterson and H. Lee, Eds., *3D Audio*. New York, NY, USA: Routledge, 2022.

[2] J. Riionheimo and T. Lokki, "Spatial Sound, Part 1: Exploring the Impact of Listening Room on Perceptual Immersion," J. Audio Eng. Soc., vol. 72, no. 12, pp. 913–929, 2024.

[3] Soundlapse Project, "Soundlapse," Available: `https://soundlapse.net/` (accessed Mar. 2026).

[4] A. Rabello-Mestre and F. Otondo, "Listening to the Anthropocene: A Queda do Céu," *Computer Music Journal*, vol. 46, no. 1–2, pp. 25–39, 2023.

[5] S. Agrawal, A. Simon, S. Bech, K. Bærentsen, and S. Forchhammer, "Defining Immersion: Literature Review and Implications for Research on Immersive Audiovisual Experiences," *Proceedings of the 147th AES Convention*, New York, 2019.

[6] S. R. Agrawal and S. Bech, "Immersion in Audiovisual Experiences," in *Sonic Interactions in Virtual Environments*, M. Geronazzo and S. Serafin, Eds. Cham, Switzerland: Springer, 2023, pp. 319–351.

[7] G. Kailas and N. Tiwari, "An Empirical Measurement Tool for Overall Listening Experience of Immersive Audio," *Proceedings of the 2021 IEEE International Conference on Consumer Electronics (ICCE)*, Las Vegas, 2021.

Computational Analysis of Expressive Tempo in Irish Traditional Dance Music

Mário Pereira
Faculty of Engineering
University of Porto
up201304843@up.pt

António Sá Pinto
INESC TEC, Faculty of
Engineering, University of Porto
asapinto@fe.up.pt

Treasa Harkin
Irish Traditional Music Archive
treasa.harkin@itma.ie

Gilberto Bernandes
INESC TEC, Faculty of
Engineering, University of Porto
gba@fe.up.pt

ABSTRACT

This paper presents a computational study of expressive tempo in Irish traditional dance music, analysing 136 annotated performances of reels and jigs. Using beat-level tempo calculation, predominant-tempo estimation, and deviation-curve analysis, we examine how timing varies across tune types, performance settings, and musical structure. Results show that expressive deviations are generally subtle: reels display a mild deceleration tendency, jigs remain highly tempo-stable, and solo–ensemble and instrument-specific differences are minimal. Phrase-level clustering reveals three characteristic deviation profiles, with strong acceleration occurring only in opening phrases, reflecting common slow-start performance practices. These findings provide, to the best of our knowledge, the first systematic quantitative characterisation of expressive timing in this tradition and highlight how micro-variations emerge from stylistic, technical, and interpretive factors while maintaining overall temporal stability.

1. INTRODUCTION

Irish traditional music is a living cultural practice that has continually transformed while retaining its core identity. Historically, it functioned primarily as social dance music performed in domestic settings, until a major shift occurred between the 1960s and 1980s when the practice moved "from the kitchen to the pub" [1]. Today's session culture represents a new social ritual in which the music has shifted from accompanying dancers to entertaining listeners. This change in function has reshaped performance practice and musical interpretation, as performers are no longer bound by the rhythmic constraints of dance accompaniment [1].

This transformation has been equally paralleled by profound changes in how the tradition is documented, preserved, and transmitted, namely undergoing a significant technological transformation. The shift from oral to media-transmitted channels represents a fundamental change in cultural transmission methods [1]. Several online repositories facilitate tune sharing and preservation. Platforms such as The Session [1] enable musicians to submit new compositions, comment on existing repertoire, and access notated materials. The Irish Traditional Music Archive (ITMA) serves as the primary institutional repository, providing professional curation and scholarly access to traditional music materials.

Access to extensive collections of recorded material creates new opportunities for applying digital-humanities methodologies to systematically identify idiosyncrasies in performance practice. Of particular interest is the exploration of the performative freedom that emerged when the music became decoupled from dance, especially with regard to temporal nuance and tempo flexibility in performance, which we hypothesise to carry expressive significance.

Existing music performance analysis (MPA) research has established robust methodologies for computing expressive performance indicators [2], encompassing computational methods for analysing expressive timing, including inter-beat interval analysis [3] and systematic approaches to timing deviation measurement [4]. Studies have investigated tempo variability and consistency across repeated performances [5], revealing both the flexibility and stability of performers' interpretive choices, while research on rubato patterns has shown how performers systematically deviate from mechanical timing to create musical expression [6].

Analyses of the relationships between timing and dynamics [7] have demonstrated the interdependence of these expressive dimensions, with corpus studies revealing systematic co-variation patterns in recordings [8]. Research on the perceptual correlates of timing decisions [9] has established links between acoustic measurements and listener experience, while work on expressive timing profiles has shown how performers create hierarchical temporal structures that listeners perceive as musically coherent [10]. Additionally, studies examining timing microstructure have revealed performance nuances at the millisecond level that contribute to perceived expressiveness [11], and research on asynchrony in ensemble performance has illuminated coordination mechanisms in multi-performer contexts [12].

Copyright: ©2026 Mário Pereira et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License 3.0 Unported, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

[1] https://thesession.org, last accessed on December 2025.

Tune info			Number of performances					Total
Type	**N**	**Title**	**Ensemble**	**Fiddle**	**Flute**	**Whistle**	**Other**	
Reel	1	Colonel Fraser	8	0	2	1	2	13
	2	Farewell To Ireland	3	2	0	2	2	9
	3	Lucy Campbell	7	1	0	1	0	9
	4	The Bucks Of Oranmore	7	0	0	1	1	9
	5	The Foxhunters	5	2	0	1	0	8
	6	The Mason's Apron	4	1	2	0	0	7
	7	The Morning Dew	5	1	1	1	1	9
	8	The New Mown Meadows	6	1	0	0	1	8
Jig	9	Banish Misfortune	4	1	0	1	1	7
	10	Morrison's	6	1	0	1	3	11
	11	The Banks Of Lough Gowna	2	1	0	1	2	6
	12	The Gold Ring	3	1	1	0	2	7
	13	The Humours Of Ballyloughlin	3	1	2	1	3	10
	14	The Humours Of Bandon	4	0	0	0	0	4
	15	The Lark In The Morning	3	0	0	1	2	6
	16	The Mist Covered Mountain	3	0	0	1	3	7
	17	The Rolling Waves	4	0	0	0	2	6
		Total	77	13	8	13	25	

Table 1: Dataset composition, indicating number of performances per tune and instrumentation. An index (N) was attributed to each tune for later reference.

These methodologies are, however, predominantly developed within Western classical music contexts in which a blueprint (e.g., a score) is available, enabling systematic comparison between notated and performed structures. Applications beyond this tradition remain limited, with relatively few studies addressing performance analysis in improvised music, oral traditions, or non-Western musical practices where score-based comparison frameworks are inapplicable [2]. The present study extends these analytical approaches to Irish traditional music, focusing specifically on expressive timing in reels and jigs, the two most prominent dance types within the tradition. We address three main questions: (1) What are the most salient expressive tempo deviations in performance? (2) Do these deviations differ across performances, dance types, and instrumentation? (3) Where in the musical structure do these nuanced timing variations occur?

To this end, we compiled a corpus of 136 reel and jig performances encompassing a range of performance configurations, with an emphasis on collecting multiple performances of the same tune. Recordings were scraped from The Session repository and subsequently validated by experts in Irish traditional music for representativeness. Following expert approval, the recordings were manually segmented at multiple temporal levels (beats, phrases, and sections). After computing the tempo and tempo deviation curves per tune, statistical indicators of expressive tempo were extracted across hierarchical layers. Comparative analyses were then conducted across (1) tune types, (2) performance configurations (e.g., solo versus ensemble and differing instrumentations), (3) multiple recordings of the same tune, and (4) structural locations within each tune, allowing for a systematic assessment of how expressive tempo varies with context, musical form, and interpretive diversity.

The remainder of this article is organised as follows. Section 2 introduces the analysed dataset, detailing its provenance, relevant characteristics, and annotation procedures. Sections 3 and 4 present the methodological framework, including the extraction of temporal features from the annotations and the techniques employed to infer expressive timing. Section 5 reports the empirical findings, which are subsequently examined and contextualised in Section 6. Section 7 summarises the findings, highlights the limitations of the study and proposes possible future directions.

2. DATASET AND ANNOTATIONS

The dataset adopted in our study was compiled via web scraping from The Session, a community website dedicated to the repertoire of Irish traditional music. The Session hosts a large repository of user-submitted sheet music, and community members frequently share links to performance recordings within comment threads. The initially scraped data was processed and formatted into a structured dataset. Recordings were validated by experts from ITMA, who screened for two criteria: (1) their representativeness within the Irish folk and tradition and (2) variability in instrumentation. In this context, we reduced the study to a selection of reels and jigs [2] due to their relevance in the tradition.

The resulting dataset, represented in Table 1, consists of 8 reels and 9 jigs. Each tune includes between 4 and 13 different recordings, resulting in 72 reel performances and 64 jig performances in total. In addition to its dance type, each performance was also labelled with its performance configuration setting: solo or ensemble. The classification of ensemble was attributed based on whether the mu-

[2] Jigs are typically notated in 6/8 time signature, characterised by two groups of three eighth notes per measure. The rhythmic emphasis typically falls on the first and fourth eighth notes, creating the characteristic ONE-two-three, ONE-two-three pattern. Reels are usually in 4/4 time, though often felt in cut time, with accents on the first and third beats of each measure [13, 14]. These metric differences represent the most salient distinction between the two types and provide a natural starting point for analysis, enabling investigation of whether these structural differences manifest in distinct performative timing patterns.

sicians were playing with any kind of musical accompaniment, e.g., other musicians or backing tracks. Then, within solo performances, selected instruments–fiddle, flute and whistle–were isolated based on higher sample size. The remaining instruments–uilleann pipes, hammered dulcimer, guitar, accordion, mandolin, banjo and khaen–with more residual numbers of performances, were categorised as "other" [3].

Irish tunes feature a single melodic line and usually consist of two sections, A and B, each made up of eight measures [4]. When repeated, these sections yield an AABB structure [13]. This regular structure facilitates the annotation process and provides a robust framework for analysing timing patterns within individual performances, enabling evaluation of how a tune's structure may influence temporal expression.

The restriction to 136 performances was imposed by the intensive manual annotation process required for each performance. Annotations were done using Sonic Visualizer through a systematic procedure. Beats were initially marked by tapping along with the audio in real time and subsequently adjusted to align as precisely as possible with note onsets. Where onsets were ambiguous or absent (e.g., sustained notes), the gap was subdivided equally and verified by ear. When other ambiguities arose, transcriptions from The Session served as reference materials. Following beat annotation, phrase boundaries were marked at eight-measure intervals (equivalent to sixteen beats in both jigs and reels). These were subsequently verified by checking that each segmentation timestamp was aligned with the first onset of each phrase, accounting for occasional missing beats at the start of some recordings.

A methodological constraint was encountered, concerning tune repetitions. During a single performance, tunes are often repeated an undetermined amount of times. To address this, in performances with multiple iterations of the tune, only the first complete iteration was analysed. While this allows for a more consistent analysis, especially within each tune, it also constitutes a limitation, potentially overlooking developments or expressive variations in later repetitions.

Within the first iteration, minor structural variability still persists. Some recordings of the same tune differ by a maximum of two phrases, due to incomplete recordings or different versions of the tune.

All information was compiled into a spreadsheet that will be shared in an online repository [5]. Annotations and relevant metadata for each performance, including links to the original media, are provided here.

[3] Since the eighteenth century, the core instruments of the Irish music tradition have included the fiddle, flute, whistle, harp, and the uilleann pipes. Over time, however, more instruments have been adopted, and the tradition nowadays employs a broad instrumental palette, exhibiting adaptability without undermining historical continuity [13].

[4] There are exceptions to this. A few tunes in the compiled dataset have three or four sections. However, these are also eight bars long and follow the same compositional logic (e.g., a tune with three sections would have an AABBCC structure, and so on).

[5] Repository available at `https://github.com/merio177/icmc2026_irish_expressive_tempo.git`

3. EXTRACTING TIMING INFORMATION

To quantify expressive timing, we extract two primary features from the beat annotations of each performance: a predominant tempo and a tempo deviation curve capturing local fluctuations.

We begin by computing inter-beat intervals (IBIs) between consecutive annotated beats and converting each interval to local tempo in beats per minute ($BPM = 60/IBI$). This yields a local tempo series for each performance.

Then, to establish a predominant tempo against which deviations can be measured, we employ kernel density estimation (KDE), as described by Ellis [15]. Given the sequence of IBIs, KDE approximates the underlying probability density function. The bandwidth parameter, which controls the smoothing applied to the estimate, was selected using Scott's rule, an approach that automatically scales bandwidth based on the sample size and standard deviation. The predominant tempo is defined as the tempo corresponding to the mode of the IBI distribution (i.e., the point of maximum density).

We now compute the deviation curve by subtracting the established predominant tempo from each value in the local tempo series. This transformation yields a deviation series that isolates expressive timing variations from the underlying stable tempo. For each performance, we compute the mean of this deviation series, producing a single global deviation metric that quantifies the overall tendency to accelerate (positive values) or decelerate (negative values) relative to the predominant tempo.

To facilitate qualitative interpretation, tempo curves are visualised as piecewise linear functions with an overlaid exponential moving average (calculated using $alpha = 0.4$ and recursive update) which smooths local irregularities while preserving global shape and directional trends. Figure 1 shows six examples of the resulting graphs. These visualizations provide intuitive verification of the computed predominant tempo and reveal expressive strategies such as gradual acceleration or phrase-specific tempo manipulation. While visual analysis is not scalable to the full corpus, such examples offer interpretive context for the discussion in Section 6.

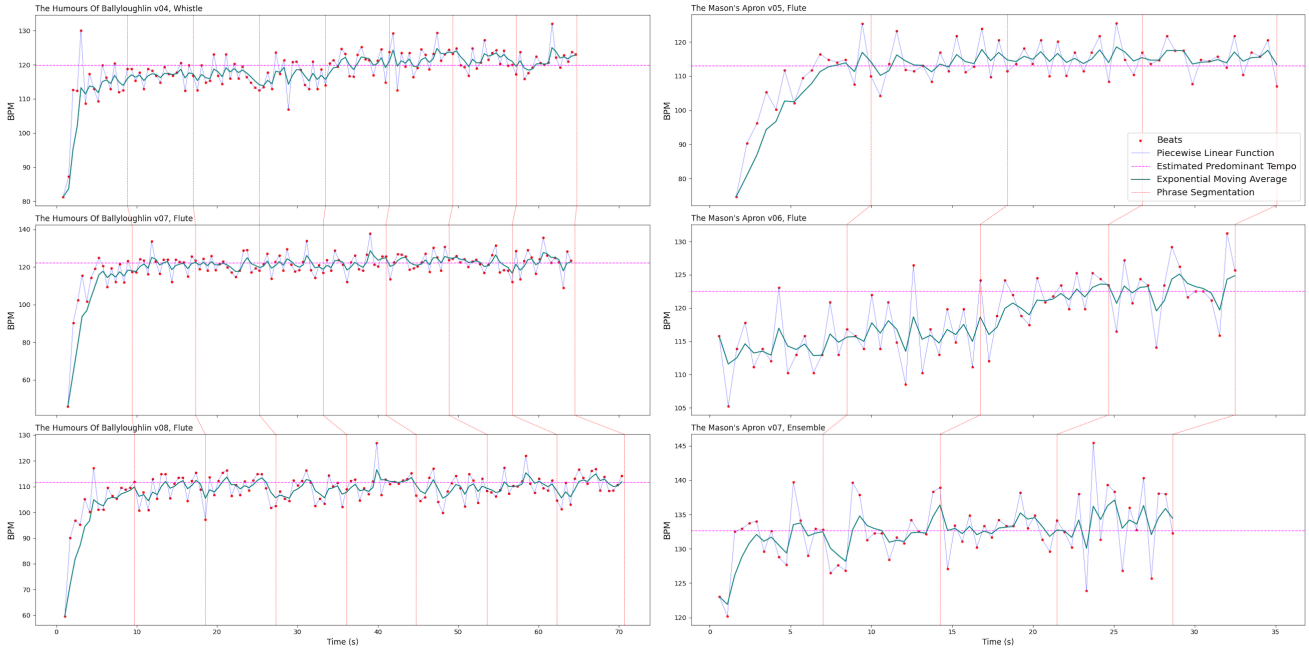
4. STATISTICAL ANALYSIS

The statistical analysis comprises two complementary approaches. First, to address research questions 1 and 2, we compute descriptive tempo deviation statistics across tune types, instrumentation categories, and individual tunes. Second, to address question 3, we apply hierarchical clustering to phrase-level tempo deviation curves to identify structurally localized timing patterns.

4.1 Descriptive statistics

To address questions 1 and 2, we calculated mean deviation values for all performances and grouped in different ways to analyse both commonalities (shared central tendencies) and diversity (between-group differences and within-group outliers).

First, we compared jigs and reels. These tune types differ primarily in metric structure, enabling investigation



(a) (b)

Figure 1: Tempo curves for three performances of a jig (a)–The Humours Of Ballyloughlin (Tune 13)–and a reel (b)–The Mason's Apron (Tune 6). The shorthands after the tunes' titles indicate specific performances (e.g., v01 refers to the first performance listed in the dataset).

of whether metric organization influences expressive timing. Second, we partitioned performances by instrumentation (solo versus ensemble) to examine how this may influence timing stability. In ensemble performances, musicians must coordinate with accompanists, whereas in solo settings, they rely solely on their internal timing. Within solo performances, we further investigate whether instrument-specific factors (e.g., physical constraints, articulation, timbral characteristics) influence timing expression. Finally, we examine whether timing patterns remain consistent across different performances of the same tune, assessing whether expressive timing is tune-specific. This may provide insight into how the musical content influences timing patterns.

4.2 Phrase clustering

To address question 3, we adopt hierarchical agglomerative clustering as a blind, content-driven procedure to identify common arch shapes in tempo deviation. Clustering is performed on the phrases' tempo-deviation curves across the entire dataset. By focusing exclusively on curve-based similarity, the method emphasizes shared tendencies (e.g., ascending/descending or uniform arch-like profiles) while remaining agnostic to configuration factors (e.g., tune type or instrumentation).

Each performance's tempo (BPM) curve was segmented into phrase-level sub-curves according to their corresponding phrase annotation boundaries. Prior to clustering, the sub-curves underwent three preprocessing steps. First, baseline normalization was performed by subtracting the initial BPM from all subsequent values, thereby resetting the origin to zero and isolating relative tempo changes from absolute levels. Second, the curves were smoothed using the same parameters described in Section 3. Third, the curves were resampled via linear interpolation to a common length equal to that of the longest phrase in the dataset, ensuring uniform vector dimensionality in the presence of minor annotation inconsistencies and occasional dropped beats. Although this resampling may introduce slight distortion, it retains the expressive contour of each curve, which is the primary focus of the analysis.

Following preprocessing, phrase-level tempo deviation curves were clustered using hierarchical agglomerative clustering with Ward's linkage [6]. A cut height (distance threshold) was applied to the resulting dendrogram to obtain a discrete partition of the data. Rather than fixing the number of clusters *a priori*, the cut height was iteratively tuned while examining the cluster-wise mean tempo deviation curve as a form of visual validation, ensuring that clusters were both statistically separable and musically interpretable.

To visualize the results, all individual phrase curves within each cluster were plotted alongside their cluster mean, providing a sense of both the consistency and variability within each group. The mean curves serve as representatives of the characteristic timing profiles associated with each cluster, enabling interpretation of the expressive strategies they embody.

[6] Under Ward's criterion, at each iteration the pair of clusters to be merged is chosen to minimize the incremental within-cluster variance (error sum of squares). When applied to raw feature vectors, this procedure is equivalent to operating with (squared) Euclidean distances, and is well suited to recovering musically coherent timing patterns.

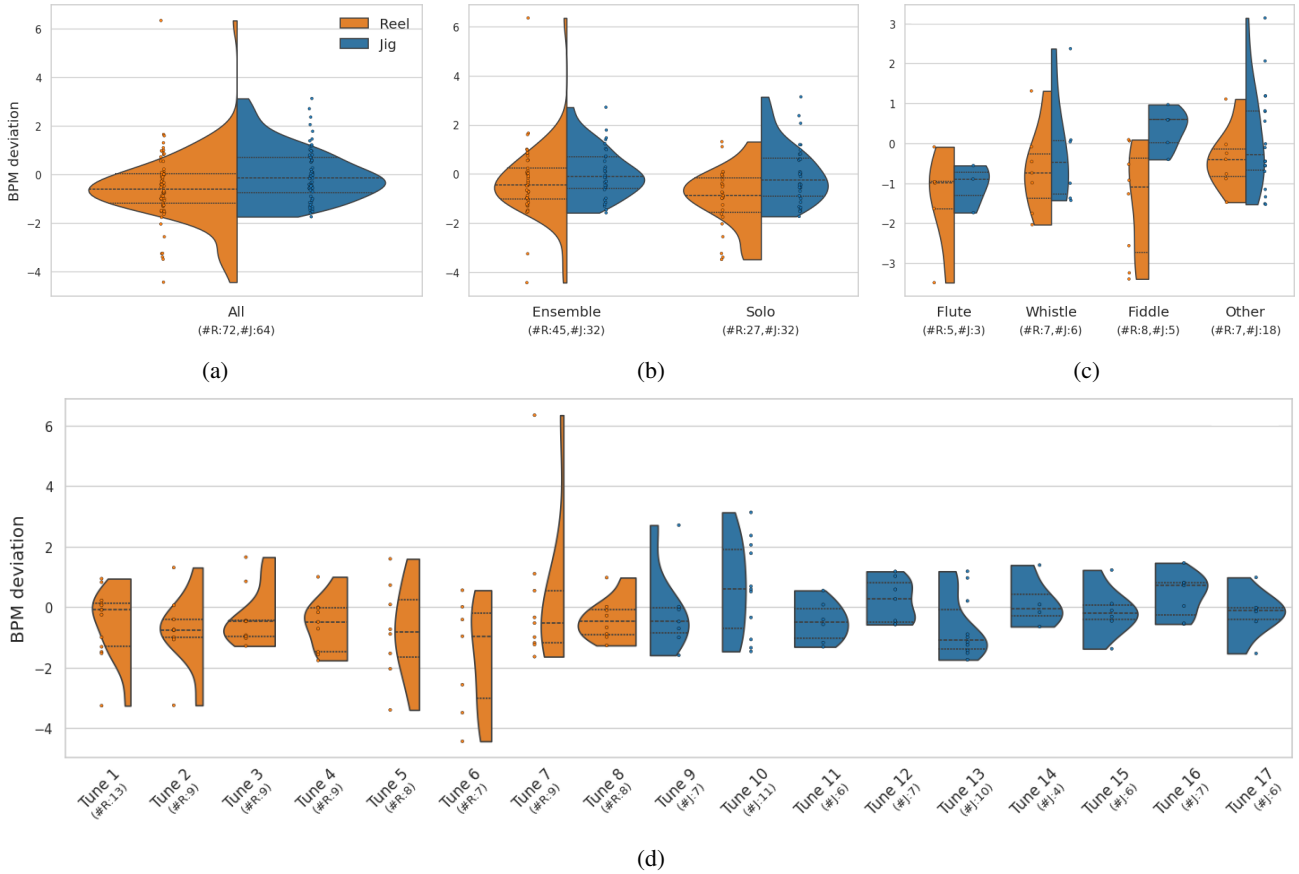


Figure 2: Violin plots showing probability density distributions of mean BPM deviation relative to the predominant tempo, across tune types (a), instrumentation settings (b), solo instruments (c), and individual tunes (d). All graphs show the main distinction between jigs and reels. Sample sizes are indicated below each label, with the shorthands "#J" for jigs and "#R" for reels.

5. RESULTS

This section presents the empirical results of the analysis. Global tempo deviation patterns are summarised using violin plots, showing the distribution of values across the examined conditions (Figure 2). Phrase-level expressive timing is analysed through hierarchical clustering, with Figure 3 illustrating the resulting patterns and their distribution across musical and performance categories. Emphasis is placed on results that characterise the magnitude and organisation of expressive timing.

5.1 Global deviation

Figures 2a–2d present distributions of mean tempo deviation relative to the predominant tempo across several variables, addressing the magnitude of global tempo deviation and its variability across musical contexts. Violin plots show both central tendency and distribution shape, allowing inspection of concentration and outliers.

A comparison between reels and jigs (Figure 2a) shows systematic differences. Jigs exhibit a compact distribution centred close to zero (median = - 0.13 BPM), indicating limited expressive timing relative to the predominant tempo. Reels show a wider distribution with a stronger negative shift (median = - 0.60 BPM), reflecting the presence of performances with sustained deceleration. While many reel performances remain close to zero deviation, a subset departs more clearly from the predominant tempo, a behaviour not observed in jigs.

Solo and ensemble performances show comparable distribution shapes and central tendencies, suggesting that ensemble context does not impose a clear constraint on expressive timing at the global tempo level. Within solo performances, instrument-specific distributions reveal subtle but consistent differences. Flute performances display a pronounced negative bias, with all performances exhibiting negative mean deviations. Fiddle shows comparable negative bias, though primarily for reels. Whistle and remaining instruments display more symmetric distributions centred near zero.

Across the 17 analyzed tunes, distributions are broadly comparable, suggesting that expressive timing strategies are not strongly tune-dependent. The Mason's Apron (Tune 6) departs from this pattern, showing the largest negative shift and the widest spread.

A positive outlier evident in Figures 2a, 2b, and 2d corresponds to performance v02 of the reel The Morning Dew (Tune 7). This pronounced deviation, rendered particularly striking by the prevailing deceleration tendency in the reels distribution, motivates a closer examination of the ensemble's acceleration strategy, undertaken in Section 6.

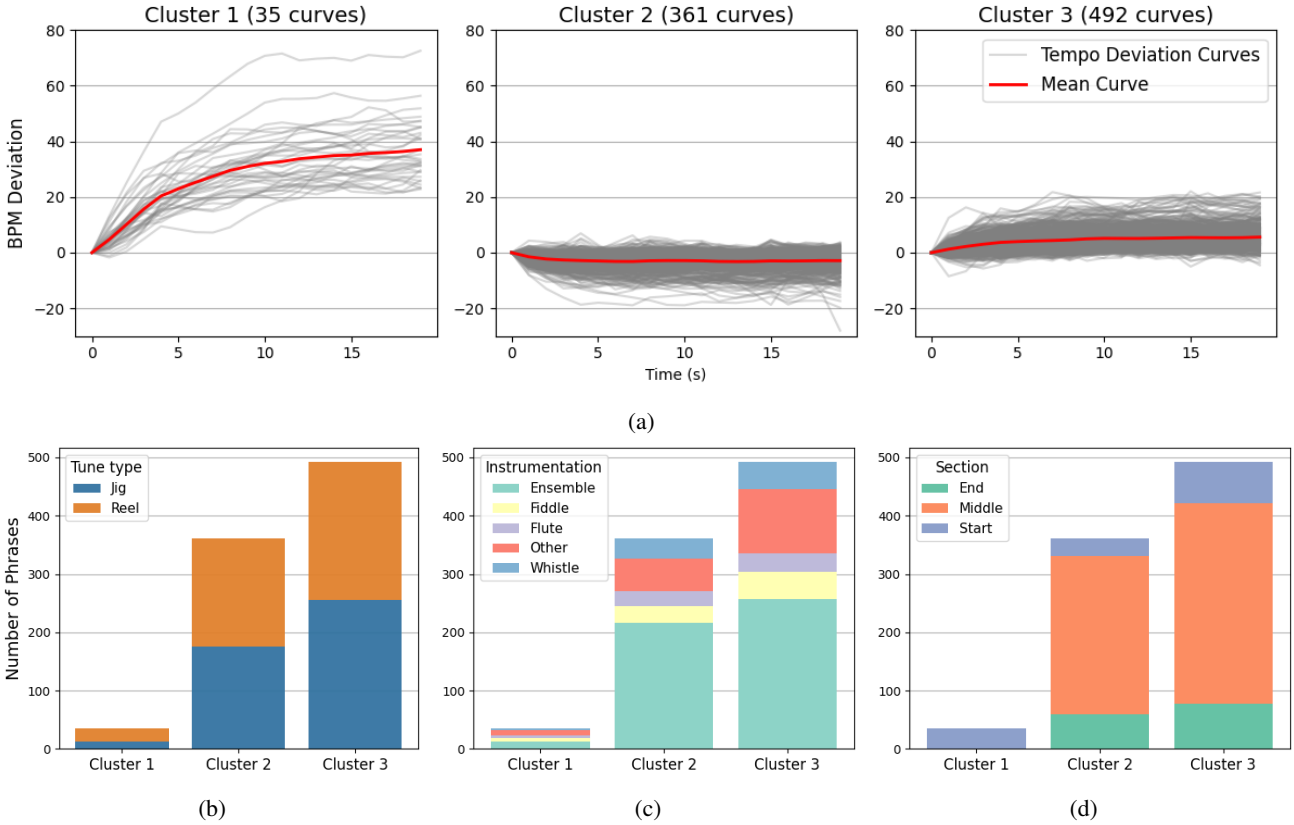


Figure 3: Resulting clusters from hierarchical clustering of phrase-level tempo deviation curves. Each grey line represents a phrase's tempo deviation curve (after the preprocessing described in Subsection 4.2). The mean curves, in red, are interpreted as the tempo deviation tendency of each cluster (a). Distribution of phrase clusters across tune type (b), instrumentations (c) and section (d).

5.2 Phrase deviation patterns

Hierarchical clustering of phrase-level tempo deviation curves resulted in three clusters, which represent three timing patterns (Figure 3a): strong acceleration (Cluster 1), mild deceleration (Cluster 2), and mild acceleration (Cluster 3). Cluster 1 contains 35 phrases and shows a clear increase in tempo across the phrase, with a mean deviation of $+37$ BPM. Cluster 2, with 361 phrases, exhibits a small deceleration (-2.9 BPM). Cluster 3 includes 492 phrases and shows a modest acceleration pattern ($+5.6$ BPM).

Figures 3b and 3c show the distribution of phrase clusters across tune types and instrumentation. Cluster membership is broadly similar across these variables, indicating that they do not strongly determine phrase-level expressive timing patterns.

To examine the structural placement, phrases were divided into three categories: *Start* (initial phrase), *Middle* (all intermediate phrases), and *End* (final phrase). This reflects common performance practice, in which opening phrases establish tempo and closing phrases support formal closure, while middle phrases reinforce the established tempo. The distribution in Figure 3d supports this. All phrases in Cluster 1 occur at the start of performances, consistent with a slow-start strategy in which tempo increases across the opening phrase. Middle and end phrases are distributed between Clusters 2 and 3, indicating that expressive timing in these positions is limited to small deviations.

6. DISCUSSION

In this section, we contextualize the quantitative findings with expert perceptual analysis, addressing the three research questions posed in the Introduction. Specifically, we examine (1) the most salient tempo deviations identified computationally, (2) how these deviations vary across tunes, tune types and instrumentation settings, and (3) the structural locations where timing variations emerge most prominently. The expert review highlighted how performers employ expressive timing in ways that are often intentional, stylistically grounded, and perceptually meaningful to trained listeners. At the same time, the findings indicate that while expressive timing is present across the corpus, most deviations are subtle and often too small to be perceived reliably without analytical tools. This convergence between perceptual and computational evidence reinforces the view that Irish traditional musicians generally maintain a high degree of temporal stability. It also suggests that the measured patterns are stylistically coherent and audibly verifiable.

6.1 Global expressive tendencies

Addressing research questions 1 and 2, the salience and contextual variability of tempo deviations, the statistical analysis between tune types demonstrates a minor tendency toward deceleration in reels, with some perfor-

mances showing more pronounced negative outliers in comparison to jigs with near-zero deviation.

Conversely, the absence of clear differences between solo and ensemble performances suggests that ensemble coordination does not strongly constrain global tempo choices. This result is consistent with the high rhythmic literacy and internalised pulse characteristic of experienced musicians, who often synchronize intuitively even in loosely coordinated session environments.

In solo instrument-specific performances comparisons, the flute displays a noticeable negative deviation bias. The physical demands of the instrument may explain this. Playing the flute for a long time and at speed requires strong lungs and great breath control, and as such, a combination of intricate melodic content with fatigue may produce slight decelerations. With that in mind, it is not surprising that this instrument, of all of them, is the one that shows a more negative deviation. Despite that, these timing drifts are still very subtle and remain within accepted performance norms.

6.2 Tune-specific interpretive behaviours

Several tunes in the corpus have innate characteristics that directly corroborate the quantitative findings. One of these is The Mason's Apron (Tune 6). This is a notoriously intricate tune, often reserved as a showpiece, with lots of opportunities for performers to show off their technique. A section in the second part features a technique called "back-stitching", where three or four notes revolve around a central note. Dense passages like these naturally invite unintended micro-accelerations or decelerations. An acceleration in this section is particularly evident in performance v06 (see Figure 1b, in the third portion of the graph). The first part of the tune also contains lots of rolls, where five notes fit into the space of three. This can also induce timing shifts, which can be heard in performance v07 (Figure 1b). Nevertheless, much of the negative deviation for this tune arises from an intentional performance choice: many players employ a deliberately slow opening phrase, followed by increasing acceleration. Without this introductory stretch, the overall deviation measurement for most recordings would likely be closer to zero or even positive—a pattern consistent with performance practice, as showpiece reels often conclude faster than they begin.

Comparable interpretive explanations apply to other tunes. Colonel Fraser (Tune 1) is another showpiece tune rich in ornamentation. However, even in the recordings with the highest measured deviation, the performers are playing with sufficient control that deviations are imperceptible by ear. These cases illustrate that ornamentation alone does not necessitate tempo drifts; rather, deviations likely depend on performer intention and the technical or expressive functions of specific passages.

Performance v02 of The Morning Dew (Tune 7) exhibits the highest positive deviation value across the entire corpus. This performance is an ensemble in which the musicians gradually accelerate over the entire duration of the performance, as seen in Figure 4, rather than just in the first phrase. This pattern seems to have yielded a lower predominant tempo relative to the whole performance, causing its mean deviation to result in a higher value.

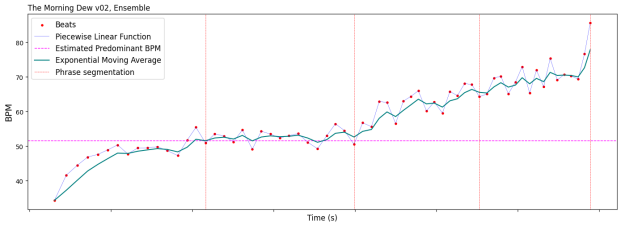


Figure 4: Tempo curve for performance v02 of The Morning Dew (Tune 7). This performance shows a distinct pattern, with consistent acceleration.

In the original recording, the performers repeat the tune five times and maintain this pattern all throughout. The highlights the methodological limitation mentioned in Section 2, regarding the analysis of only the first iteration.

6.3 Phrase-level timing strategies

Concerning research question 3, the hierarchical clustering revealed three phrase-level profiles. Cluster 1 is of particular importance, representing pronounced acceleration patterns found exclusively in opening phrases. This supports both the global deviation findings and the qualitative observations: musicians frequently begin at a slower pace and quickly establish their target tempo within the first phrase. In some cases the first few notes are played slowly but in many cases it is only the first note that is held longer. This is very common practice and serves as a way of getting the audience's attention. It also grounds the performer with their instrument and ensures that their hands are well positioned. As this is an oral tradition, the first few notes of a tune are very important to signal to the audience and to other performers which tune is being played. For example, The Humours of Ballyloughlin (Tune 13) includes performances in which the opening note is intentionally prolonged. This is graphed as a major acceleration, as seen in Figure 1a but musically it is not perceived as such.

7. CONCLUSIONS AND FUTURE WORK

This study presents, to the best of our knowledge, the first systematic, data-driven study of expressive tempo in Irish traditional music, combining beat-level tempo modelling, KDE for predominant tempo estimation and hierarchical clustering of phrase-level deviation curves. By examining 136 annotated performances of reels and jigs, we quantified how timing varies across tune types, instrumentation settings, and musical structure.

The results show that while expressive tempo variation is indeed a component of Irish traditional music performance, it typically remains subtle. Deviations arise from a combination of technical, physiological, and expressive factors, including ornamentation density, phrase initiation strategies, instrument-specific constraints, and performance context. Even cases that deviate significantly in numerical terms, such as extended lead-ins or deliberate slow starts, are readily interpretable within the stylistic framework of the tradition.

These findings establish a quantitative baseline for understanding expressive timing in the tradition and confirm that even statistically detectable deviations are musically

coherent and culturally grounded. In practical terms, they also suggest a role for ear-based validation by expert musicians and for more musically informed expressive synthesis, especially through structured phrase-initial expansion and controlled stabilization rather than undifferentiated random fluctuation.

Several directions for future work emerge from this study's scope and limitations. The most logical continuation is the expansion of the corpus. The annotation of additional performances would provide a more robust foundation for statistical analysis and enable investigation of a broader range of tunes, tune types, and instruments. Given the richness of the Irish traditional repertoire and the diversity of contemporary performance practices, a larger dataset could reveal patterns currently masked by limited sample sizes. Beyond corpus expansion, several analytical dimensions could be incorporated. First, the restriction to first iterations leaves portions of many performances unexplored. Performers often play a tune two or more times consecutively. A longitudinal analysis across consecutive iterations would test whether deviations accumulate, stabilize, or vary in any other way. Second, incorporating performer metadata would enable comparison between them, possibly revealing how individual stylistic choices or different levels of expertise may contribute to expressive timing patterns.

Alternatively, a more in-depth approach could address fine-grained timing adjustments. The present study operates at global and phrase-level resolutions, potentially masking local adjustments. Ornamentation, a defining characteristic of Irish traditional music, involves rapid note sequences (e.g., rolls, cuts, trills) that create opportunities for local timing shifts. A higher resolution analysis, examining timing at the level of individual beats and note onsets within ornamented passages could illuminate whether ornaments systematically induce acceleration or deceleration.

Acknowledgments

Research funded by the project "EA-DIGIFOLK: AN EUROPEAN AND IBERO-AMERICAN APPROACH FOR THE DIGITAL COLLECTION, ANALYSIS AND DISSEMINATION OF FOLK MUSIC" (101086338) under the program Marie Skłodowska-Curie Actions Staff Exchanges (HORIZON-MSCA-2021-SE-01-01).

8. REFERENCES

[1] E. Falc'her-Poyroux, "Irish Music Redefined, An Outsider's Viewpoint," in *Music and the Irish Imagination*, T. Dubost and A. Slaby, Eds. Presses universitaires de Caen, 2013, pp. 29–41. [Online]. Available: http://books.openedition.org/puc/7078

[2] A. Lerch, C. Arthur, A. Pati, and S. Gururani, "An Interdisciplinary Review of Music Performance Analysis," *Transactions of the International Society for Music Information Retrieval*, vol. 3, no. 1, pp. 221–245, Nov. 2020. [Online]. Available: http://transactions.ismir.net/articles/10.5334/tismir.53/

[3] B. H. Repp, "Patterns of expressive timing in performances of a Beethoven minuet by nineteen famous pianists," *The Journal of the Acoustical Society of America*, vol. 88, no. 2, pp. 622–641, 1990.

[4] E. F. Clarke, "Generative principles in music performance," in *Generative Processes in Music*, J. Sloboda, Ed. Oxford: Clarendon Press, 1988, pp. 1–26.

[5] A. P. Demos, T. Lisboa, and R. Chaffin, "Flexibility of Expressive Timing in Repeated Musical Performances," *Frontiers in Psychology*, vol. 7, Oct. 2016. [Online]. Available: http://journal.frontiersin.org/article/10.3389/fpsyg.2016.01490

[6] N. Todd, "A Model of Expressive Timing in Tonal Music," *Music Perception*, vol. 3, no. 1, pp. 33–58, 1985.

[7] A. Llorens, "The Recorded Brahms Corpus (RBC): A Dataset of Performative Parameters in Recordings of Brahms's Cello Sonatas," *Empirical Musicology Review*, vol. 16, no. 1, pp. 124–133, Dec. 2021. [Online]. Available: https://emusicology.org/article/id/4796/

[8] W. Goebl, "Melody lead in piano performance: Expressive device or artifact?" *Journal of the Acoustical Society of America*, vol. 110, no. 1, pp. 563–572, 2001.

[9] D. Fabian and E. Schubert, "Musical Character and the Performance and Perception of Dotting, Articulation and Tempo in 34 Recordings of Variation 7 from J.S. Bach's *Goldberg Variations* (BWV 988)," *Musicae Scientiae*, vol. 12, no. 2, pp. 177–206, Jul. 2008. [Online]. Available: https://journals.sagepub.com/doi/10.1177/102986490801200201

[10] C. Palmer, "Music performance," *Annual Review of Psychology*, vol. 48, pp. 115–138, 1997.

[11] A. Gabrielsson, "Music performance research at the millennium," *Psychology of Music*, vol. 31, no. 3, pp. 221–272, 2003.

[12] P. Keller, "Ensemble performance: Interpersonal alignment of musical expression," in *Expressiveness in Music Performance*, D. Fabian, R. Timmers, and E. Schubert, Eds. Oxford: Oxford University Press, 2014, pp. 260–282.

[13] D. E. Hast and S. Scott, *Music in Ireland: Experiencing Music, Expressing Culture*, ser. Global Music Series. Oxford, New York: Oxford University Press, Jul. 2004.

[14] F. Vallely, Ed., *The companion to Irish traditional music*, 2nd ed. Cork: Cork University Press, 2011, oCLC: ocn772584731.

[15] R. J. Ellis, Z. Duan, and Y. Wang, "Quantifying Auditory Temporal Stability in a Large Database of Recorded Music," *PLoS ONE*, vol. 9, no. 12, p. e110452, Dec. 2014, publisher: Public Library of Science (PLoS). [Online]. Available: https://dx.plos.org/10.1371/journal.pone.0110452

The MMMAudio Computer Music Environment

Sam Pluta
Peabody Institute of the Johns Hopkins University
spluta1@jhu.edu

Ted Moore
Peabody Institute of the Johns Hopkins University
tmoore97@jhu.edu

ABSTRACT

We introduce MMMAudio, a new audio creative coding environment designed to close the gap between instrument building and low-level DSP development while reducing the maintenance burden typical of monolithic, compiled systems. Contemporary computer music languages such as Max, Pure Data, and SuperCollider excel at graph-based instrument design but impose steep barriers when custom DSP is required, pushing users into C/C++ plugin workflows with unfamiliar APIs, build systems, and cross-platform complexities. MMMAudio addresses these issues by centering its programming model on Mojo for high-performance DSP and seamless Python–Mojo interoperability for tooling, AI, and scientific libraries. In MMMAudio, unit generators (UGens) are simple Mojo structs, enabling users to write, test, and distribute new UGens without leaving their code editor or contending with external build pipe-lines. This design simultaneously encourages new DSP creation, leverages Python's mature ecosystem for machine learning and data processing, and exploits Mojo's performance features (e.g., SIMD) for fast, real-time audio processing. We present the system's architecture, programming model, and extension mechanisms.

1. INTRODUCTION

A new software system, called MMMAudio, is in the early stages of development. Its design foregrounds three strengths that we feel set it apart from existing Computer Music Environments (CMEs): 1) It unifies instrument building and DSP authoring under the same workflow and language (Modular's Mojo), letting creators prototype innovative DSP directly within the same environment where they are building instruments. 2) It leverages the existing, well supported programming languages, Mojo and Python, as well as their infrastructure, taking advantage of the many packages in the Python ecosystem. This will allow the devteam to focus on core functionality and DSP, instead of having to maintain multiple layers of tooling: IDE environments, GUI packages, i/o libraries, etc. 3) It embraces industry-leading and foundation-supported AI and analysis tools, connecting to Python and Mojo's mature ML ecosystems rather than requiring teams of developers create entirely new systems for AI training and inference[1, 2].

MMMAudio achieves this by leveraging Mojo-Python interoperability, creating a Python-based control environment with fast DSP written in Mojo. This allows us to fully engage the robust Python ecosystem for everything from GUI to serial communications, use any IDE (from VS Code to vim) for development, and use industry-leading AI tools like PyTorch and sci-kit learn for AI engagement, all-the-while running custom, fast DSP in Mojo. The current state of MMMAudio can be found at:
https://github.com/spluta/MMMAudio.

Copyright: ©2026 Sam Pluta et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License 3.0 Unported, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

2. WHY MOJO

2.1 If it Ain't Broke, Don't Fix It (History)

The leading CMEs all have a similar structure. They split operations between a "slow" but facile scripting language for data manipulation/control structures and a fast, compiled language for audio manipulation and complex math. In his 2002 paper, *Max at Seventeen*[3], Miller Puckette attributes the influence of Max's design to Max Matthews, whose "RTSKED [programming environment] attacked the problem of real-time scheduling of control operations for a polyphonic synthesizer" using "sporadically occurring events that caused state changes in the synthesizer."
"RTSKED modeled the performance as a collection of tasks running in parallel." "This separation of the real-time control problem into separate tasks was a key advance necessary for the computer to act as a musical instrument."

The "Max design," with parallel, "sporadic" operations first controlled audio via MIDI on a completely separate machine, the IRCAM 4X[3]. In 1991, Puckette maintained this split when he added tilde$\sim$ audio processing objects, and the system became Max/MSP. "Max provides a separate data type for audio signals which does not mix smoothly with messages; conversion from audio to messages is problematic, and the scheduler treats control and audio-rate processing separately [3]." Thus, 1991 marked the release of the first Max version that ran a parallelized control system alongside a fixed sample-rate audio graph—all on the same machine.

James McCartney's SuperCollider[4] language originally grew out of a similar split. McCartney had written "Synth-O-Matic which was a non-real-time C-like synthesis programming language" and "a MAX object called Pyrite which contained the interpreter for the language which was extended and used in SuperCollider. Writing SuperCollider involved integrating the language interpreter, garbage collector and function library of Pyrite with the synthesis engine and functions of Synth-O-Matic[5]." While SuperCollider 1 and 2 felt as though the scripting language and the synthesis engine were part of the same basic language, "SuperCollider Server" (often referred to as SC3), the current version in use, more fully separates their functions, go-

ing so far as to make them separate applications: SCLang and SCServer[4].

Like SuperCollider, Max, and Pure Data, MMMAudio follows a control-language/audio-engine split, using Python as the control environment and Mojo for the audio engine. Because Mojo syntax is designed to be Pythonic [6], MMMAudio has a similar "feel" as when using "Max" (control data) and "MSP" (audio~ processes) in the same Max patch. Jumping back and forth between writing code for signal processing and controlling systems is smooth. Furthermore, Python is widely considered to be one of the easiest programming languages to learn, which we believe will be echoed with Mojo as more users learn it. This lowers the threshold-for-entry and puts writing low-level DSP in Mojo just one step beyond Python programming, making MMMAudio an ideal environment for beginners and experts alike.

2.2 The Problem Mojo is Designed to Solve

> "we see Mojo as being an incredible part of the Python ecosystem. We're not looking to break Python or change it, or, quote, unquote, 'fix it.' We love Python for what it is. Our view is that Python is just not done yet... it just doesn't have those features that you would use to do C-like programming. And so if you say, okay, well, I'm forced out of Python into C for certain use cases, well, then what we're doing is we're saying, 'Okay, well, why is that? Can we just add those features that are missing from Python back up to Mojo?' And then you can have everything that's great about Python, all the things that you're talking about that you love plus not be forced out of it when you do something a little bit more computationally intense, or weird, or hardware-y" [6]

In the quote above, Chris Lattner, the author of Mojo, describes the impetus behind creating the Mojo Programming Language[7]. He describes the issue he is solving as this: AI developers train their models in Python using high-level tools such as PyTorch and TensorFlow. These tools provide industry-leading, easy-to-use frameworks for training deep learning models. However, while these frameworks are great for training models, once models are trained, running inference on them at scale requires a different approach. Python is no longer the right tool because it is a scripting language that is not designed for speed. Lattner points out later in the same interview that those most versed in training AI models aren't usually as skilled (or even interested) in coding in performant languages like C++ or Rust. Thus, a whole second team of developers is needed to translate the trainings from Python to compiled languages that can run the trainings at high speed.

While Lattner is specifically discussing issues around AI inference, he might as well be describing the current state of creative coding for audio, where a similar issue exists. Languages like SuperCollider, Max, and Pure Data are outstanding environments for instrument building and interface design, but they (SuperCollider and Max at least) fall short when it comes to DSP implementation.

2.3 C and C++ Based Systems

According to Puckette, Max and Pure Data (and we include SuperCollider as well) provide "a way of combining pre-designed building blocks into configurations useful for real-time computer music performance. This includes a protocol for scheduling control and audio-rate computations, an approach to modularization and component intercommunication[3]." In these systems, the user constructs control graphs and DSP graphs using pre-existing unit generators. These UGens are pre-compiled and each system has a convenient way of plugging the generators together.

However, an issue arises when the user needs to do something that is not possible with existing UGens. When low-level DSP design is desired, the user has to leave their creative coding environment of choice to work in a completely different environment: one with a different set of tools and a different skill-set necessary for development.

Plugins for CMEs are usually written in C or C++ or some combination thereof, and the learning curve goes beyond just having to learn a new language. In order to make a plugin, a musician-turned-developer has to learn:

1. a whole new language: C or C++.
2. the API for the CME
3. compilation tooling such as CMake, Make, etc.
4. differences between Windows and Unix compilers, etc.
5. usually a whole new code editor and development environment
6. distribution strategies, such as GitHub Actions, if you want users to download a build for their machines

The potholes around CMake are large enough, but the sum total of barriers to plugin development for the average creative coder are profound. Even Cycling74's excellent gen [8] environment, while looking like Max, is different enough from Max to create real barriers to entry.

2.4 Enter MMMAudio and Mojo

MMMAudio avoids these problems. MMMAudio code is written in Mojo, and MMMAudio plugins are written in Mojo. If you know MMMAudio, you know Mojo and if you know Mojo, you can make a plugin. Importantly, the user doesn't need to leave the development environment or even the current file to do so. A UGen in MMMAudio is simply a Mojo struct with a suitable "next" function ready to give the next sample when called upon. Further, the entire source code for the MMMAudio project is available inside the programmer's IDE, so starting a new plugin is as simple as finding a plugin resembling what is needed, copying it, and editing the code for new functionality.

3. EXTENDING MMMAUDIO

3.1 Creating New UGens

All CMEs discussed in this paper share a common design. They all allow instrument designers to construct complex structures out of simpler structures using *composition*. SuperCollider users can create Pseudo-UGens out of UGens

and then construct SynthDefs from UGens and Pseudo-UGens [9]. Max and pd users can construct subpatchers from objects and other subpatchers. And gen patchers can contain other gen patchers.

MMMAudio's approach to composition takes inspiration from the Faust Programming Language's[10], in that environment's DSP library is built out of the DSP library itself. Complex single-sample unit generators are built out of simpler single-sample UGens, all of which are written in Mojo. For example, the code below represents a comb filter written in Faust:

```
fb_comb(maxdel, N, b0, aN) = (+ <: delay(maxdel,
↪  N-1),_) ~ *(-aN) : !,*(b0) : mem;
```

Listing 1: Faust's feedback comb filter.

This function builds a comb filter by placing Faust's delay object (which is declared elsewhere in the code-base) directly in its feedback path. There are three important points: 1) the complex UGen is composed of simpler UGens, all written in the same language. 2) A coder can use this language to create single-sample feedback UGens where other UGens can be placed in the feedback path. 3) This new UGen of UGens can be placed inside of other UGens, or infinitely many, that can also operate at single-sample delay. 4) The result is a compiled UGen, not a patch or SynthDef, that is no different from any other UGen.

In the MMMAudio code below (both more verbose and more readable than the Faust example above), we similarly insert a delay into the feedback loop of the Comb UGen. This kind of feedback loop could be constructed in Max (with tapin/tapout) or SuperCollider (with LocalIn/LocalOut), but in both cases the minimum length of the delay time would be the block size, limiting its use.

```
struct Comb[num_chans: Int = 1, interp: Int =
↪  2](Movable, Copyable):
    var world: World
    var delay: Delay[Self.num_chans, Self.interp]
    var fb: MFloat[Self.num_chans]

    fn __init__(out self, world: World, max_delay:
    ↪  Float64 = 1.0):
        self.world = world
        self.delay = Delay[Self.num_chans,
        ↪  Self.interp](self.world, max_delay)
        self.fb = MFloat[self.num_chans](0.0)

    fn next(mut self, input: MFloat[self.num_chans],
    ↪  delay_time: MFloat[self.num_chans] = 0.0,
    ↪  feedback: MFloat[self.num_chans] = 0.0) ->
    ↪  MFloat[self.num_chans]:
        fb_in = input + self.fb * clip(feedback,
        ↪  -1.0, 1.0)
        var out = self.delay.next(fb_in, delay_time)
        self.fb = out
        return out
```

Listing 2: MMMAudio comb filter in Mojo.

Because MMMAudio and Faust (and Pure Data in certain subpatchers) operate sample-by-sample, filters, analog modeling, sample-accurate granulation, feedback systems, and more, are expressible, where their creation in SuperCollider and Max is stunted due to block-size limitations.

3.2 A Library of Composed UGens

Using MMMAudio, larger UGens can be constructed by composition, just like SuperCollider SynthDefs or Max subpatchers, but in MMMAudio these composed UGens are compiled into their own objects, rather than functioning as a linked-together lists of pre-compiled objects. A clear example of MMMAudio's composition style is its PitchShift UGen, which is comprised of many familiar building blocks: a Recorder object with embedded SIMDBuffer, a Dust (to facilitate triggering Grains with time dispersion), and multiple Grain UGens. Each Grain contains a buffer player called Play. Playback of the Recorder's SIMDBuffer inside the Play UGen is controlled by a Impulse phasor, which also drives the Envelope. A PolyTrigger UGen controls polyphonic voice allocation, growing the list of Grains when more voices are needed.

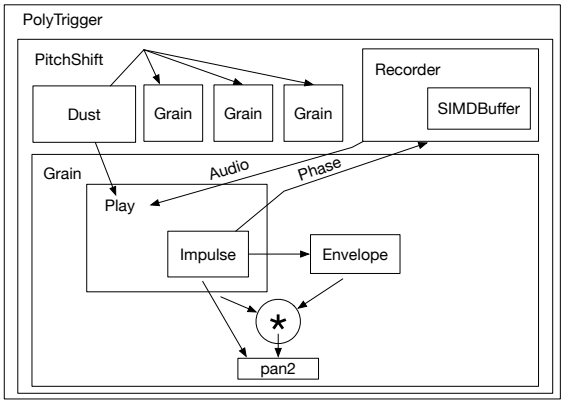


Figure 1: Composition-based construction of the Pitchshift UGen

4. JOINING THE PYTHON ECOSYSTEM

4.1 Mojo-Python Interop

Maybe the biggest advantage of MMMAudio is that it is situated in the Python ecosystem. Mojo, MMMAudio's DSP language, is designed for Python interoperability[11], and Mojo's PythonModuleBuilder allows building MMMAudio graphs directly in Python. When one runs a MMMAudio graph from Python, Mojo's PythonAudioBuilder automatically determines whether the graph has changed since last compiled, and if necessary, compiles a new executable before starting the audio engine. The MMMAudio Graph then runs as a Python external directly in the Python environment where it was compiled, ready to receive control messages from its Python host. If the graph has not changed since its last compilation, it runs instantly without needing to compile.

4.2 Many Many Packages

Being inside the Python ecosystem allows us to take full advantage of the many packages necessary for building a creative coding environment. We use mido[12] for MIDI, hidapi[13] for hidapi bindings, python-osc[14] for Open

Sound Control, matplotlib[15] for plotting, pyautogui[16] for the mouse location, PySide6 (Qt)[17] for gui, etc. We use Python to parse the messages received from these various packages, then send modified messages to MMMAudio's audio engine using the MMMAudio messaging system: a simple key-value system able to send floats, ints, triggers, lists, text (for loading files like audio files and json files), etc, back and forth between Python and Mojo. Beyond control data, we use pyaudio[18] bindings for Portaudio[19] and have access to Numpy[20] and SciPy[21] for data manipulation.

Having these tools available for use was priceless in developing MMMAudio quickly and efficiently. Since these tools are available and are maintained by the Python community, the MMMAudio development team does not have to create and maintain our own bindings (though we do plan to bind portaudio and Mojo in the future). Instead, we look to the open-source community to make these available. Importantly, many of these packages have major foundation/commercial support (Python Foundation, KDE Free Qt Foundation, and Modular), which means they will be open-source and available for years to come.

4.3 Python is the Place for AI and Mojo was Designed for AI

By joining the Python ecosystem we gain access to the most powerful and widely used tools in AI: PyTorch[22] and TensorFlow [23] for deep-learning and scikit-learn[24] for training smaller models, analysis, classification, decomposition, etc. This means that *any neural network that can be run in Python, using industry-leading AI tools, from simple multi-layer perceptrons to transformers, can run in MMMAudio*. Training happens in Python, but inference can be made either on the Python side, or, since Mojo can run Python code internally through its Python interop, directly in the audio-engine. Further, Mojo is designed for GPU use and highly parallelized processes (Modular's MAX inference engine is written in Mojo[25]). This will streamline development of further AI inference and allow us to run neural networks and FFT processes parallelized on the GPU when useful.

4.3.1 Example 1: PyTorch Inside Mojo for Low-level Synthesis Control

Multi-Layer Perceptrons are commonly used in synthesis systems for mapping low-dimensional input vectors to higher dimensional control vectors. These, in turn, manipulate the parameters of complex synthesizers [26]. The following example shows where MMMAudio can excel at this task, giving the user high-dimensional control of low-level parameters.

In MMMAudio, neural network training happens in Python, using standardized AI tools like PyTorch. We can then run inference on our trained network in Python, but because Python code can run directly in Mojo, the ideal approach is to place the PyTorch MLP code directly inside our Mojo synth. In this example, our synth is a modified Freeverb algorithm with added delayed feedback and crossfeedback, sending each channel of its stereo outputs back into both its inputs. For each of the two channels there are 8 lowpass comb filters in parallel followed by four allpass filters in series. These filters expose all of their parameters to the multi-layer perceptron. The MLP controls 70 parameters in total: delay time, feedback, and lowpass frequency for each of the 16 Comb filters, delay time and feedback coefficient for each of the 8 allpass filter pairs, feedback and crossfeedback coefficients for overall feedback of the system, and delay time for each channel of the delay on the feedback. The entire structure is controlled by a 3 dimensional input: 1 slider and 1 xy pad.

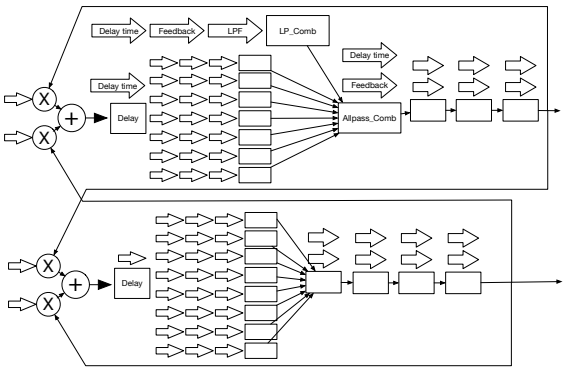


Figure 2: MLP Control of a Single Sample Feedback Network. Each ARROW represents an output from the MLP.

While this structure is technically possible in any CME (or combination of CMEs), our ability to make accessible every parameter of every UGen at single sample feedback, combined with the capacity to embed industry-leading inference engines inside any UGen, makes this approach straight-forward, idiomatic, and accurate in MMMAudio.

4.3.2 Example 2: Audio Descriptors and Python-based Data Processing

As made popular by CataRT[27], a common practice when approaching data science for musical creativity includes analyzing and plotting audio slices for inspection, validation, or performance. Mubu[2] and FluCoMa[1] added machine learning to this approach. Because it is situated in Python, MMMAudio easily replicates and extends these patterns of use by drawing on extensive existing tools such as scikit-learn[24] and matplotlib[15].

The following example uses MMMAudio's Python-based non-real-time slicing and analysis tools as described in Section 5.5. A collection of test audio files (that come with the FluCoMa Toolkit[1]) is sliced using a spectral-flux-based onset detection algorithm. Each slice is analyzed for 14 MFCC coefficients. The non-real-time analysis for each slice takes place in Mojo and is returned to Python as a Numpy[20] array with shape `( Num FFT Frames, 14 MFCCs )` and data type `float64`. Once in Python, the MFCC coefficients of each slice are summarized as the mean of the coefficient over all the FFT frames in the slice. The zeroth coefficient is removed to avoid representations of overall spectral energy (see Listing 3). With each slice now described by a vector of 13 coefficients, umap-learn[28] is used to reduce the dimensionality of the dataset to two dimensions for plotting in matplotlib[15] (see Figure 3). A callback function is used to send the mouse position to a fit KDTree (from scikit-learn[24]), find the slice nearest to the mouse, and send the necessary data for that slice to Mojo for playback.

```
    d = {
        "path": "resources/src.wav",
        "thresh":0.01,
        "min_slice_len":0.1,
        "window_size":1024,
        "hop_size":512,
        "num_coeffs": 14
    }
    y, sr = librosa.load(d["path"], sr=None)
    slices = MBufAnalysis.spectral_flux_onsets(d)
    data = np.ndarray((len(slices)-1, 13))
    for i in range(len(slices)-1):
        start = int(slices[i])
        end = int(slices[i+1])
        d["start_frame"] = start
        d["num_frames"] = end - start
        mfccs = MBufAnalysis.mfcc(d)
        data[i] = mfccs[:, 1:].mean(axis=0)
```

Listing 3: Excerpt of MMMAudio MFCC extraction in Python.

While other CMEs are also able to create this functionality, the strength of MMMAudio is its access to the plethora of data science and machine learning tools in Python. Many more are accessible in Python than what has been ported to existing CMEs. As new machine learning methods and tools become available, they are almost certain to appear in Python, at which point MMMAudio users can immediately experiment with their potential for musical applications.

4.4 IDE Toolchains

It cannot be overstated how important sophisticated IDE toolchains are in modern coding systems. Because MMMAudio uses industry supported languages, we have access to Pylance (or other Python language server) and the Mojo Language Server, which provide essential code intelligence features like autocomplete, error detection, go-to-definition, and symbol search. In addition, both Python and Mojo can be run in Jupyter Notebooks [29], adding an expressive tool for prototyping and live-coding alike.

5. FURTHER SPECIFICS

5.1 Built-in SIMD Processing

Mojo is a language written primarily for GPU-based AI inference. Outside the Fourier Transform and Neural Network inference, vanilla audio DSP has few applications for parallel processing. However, our codebase leverages Mojo's inherent SIMD types, "enabling you to write code that automatically leverages modern CPU vector capabilities while maintaining code clarity and portability[30]." This enables parallel processing in filters, distortions, FFT channels, and algorithmic generators, essentially giving us the second channel of a stereo effect at little or no extra CPU cost.

Data types like SIMDBuffer and Delay store audio in sequential lists of SIMD vectors instead of interleaved arrays. This optimizes reading and writing audio data, since 1 copy or write is needed where 2 would be necessary when using an interleaved buffer. Stereo UGens can output 2D SIMD vectors, and when these vectors get sent down

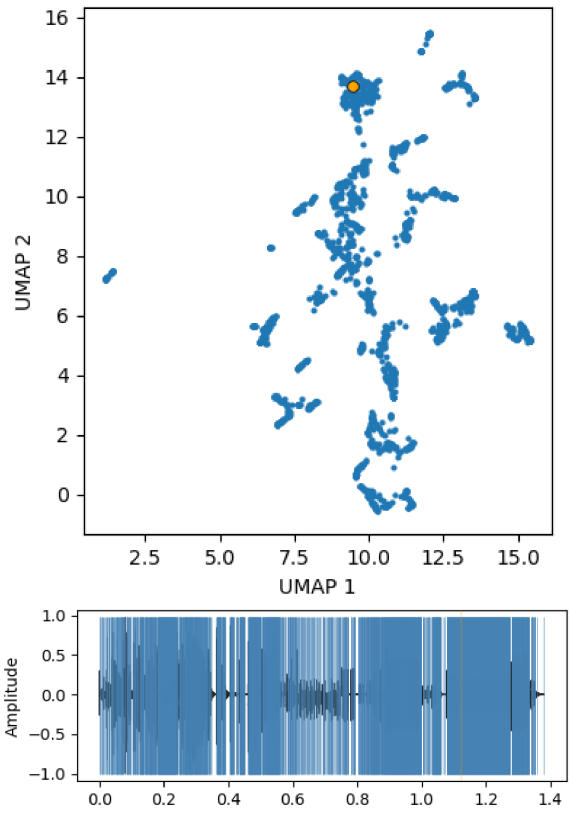


Figure 3: UMAP dimensionality reduction plot of MFCC analyses and corresponding waveform slice visualizer.

the processing chain, MMMAudio UGens written to receive any-dimensional SIMD vectors will process the audio in parallel, maintaining the dimensionality of an input vector as its output.

```
struct OnePole[N: Int = 1](Movable, Copyable):
    var last_samp: MFloat[N]

    fn __init__(out self):
        self.last_samp = MFloat[N](0.0)

    fn next(mut self, input: MFloat[N], coef:
    ↪  MFloat[N]) -> MFloat[N]:
        coef2 = clip(coef, -0.999999, 0.999999)
        var output = (1 - abs(coef2)) * input +
        coef2 * self.last_samp
        self.last_samp = output
        return output
```

Listing 4: a simple one-pole filter, able to process a Float64 or N dimensional SIMD Float64 vector.

This complexity is largely hidden from the user. Because Mojo's Float64 type is a one dimensional SIMD Vector, the user does not need to fully understand SIMD to reap its rewards. They can start learning MMMAudio by thinking in floats. [1]

[1] SIMD vectors do have limitations; in Mojo, they must be sized at a power of 2, and the user should not make SIMD vectors larger than two times the processor's vector size, or the code will be slow[30]. Arm processors like the Apple M1 can process two 64 bit floats in parallel, while Intel chips can process up to eight in parallel[31].

Since stereo is a common audio format, we have chosen SIMD as our default data type. MMMAudio can process larger signals using Mojo's various collections, like List.

5.2 Interpolation

At the core of any audio engine are great sounding oscillators. One of the basic DSP operations MMMAudio provides is a single, robust, wavetable-oscillator struct, `Osc`, with various approaches to interpolation (linear, cubic, quadratic, lagrange, sinc) and oversampling for alias-free high-quality audio synthesis. Any wavetable, from the default base waveforms to commercial 3D morphing wavetables, can be read by `Osc`, and it can interpolate between any number of wavetables, resulting in 3D wavetable lookup. Frequency and phase modulation and phase synchronization are possible as well. This provides a solid foundation for exploration of alias-free timbrel synthesis.

5.3 Oversampling

Additionally, MMMAudio has a built-in Oversampling struct that can be folded into any Unit Generator or synth graph. This struct holds an N-SIMDsample oversampling vector (corresponding to N times oversampling), a function to add SIMD samples to the buffer, and a function to low pass filter the buffer and return a single SIMD vector of output. Mojo's parameter arguments allow the user to bypass the oversampling code completely (at compile-time) or parameterize and inline the oversampling process. This means that any UGen designed as a "standard" 1X process can easily be adapted to run as an oversampled UGen, while maintaining efficiency when the oversampling is bypassed.

5.4 FFT Processing

MMMAudio includes an intuitive and powerful system for implementing custom FFT-based processing and analysis algorithms. The user creates a custom struct (coded in Mojo), with a `next_frame` function to which two lists of `Float64` values are passed: (1) the magnitudes and (2) phases of the most recently processed FFT window. These values can be used to compute an analysis or can be transformed in-place. The output of the custom struct is then passed to an IFFT and eventually returned as a stream of audio samples reflecting the user-implemented spectral processing.

To be used, the custom struct is simply passed to MMMAudio's existing `FFTProcess` struct, which deals with all the necessary plumbing for a Fast Fourier Transform. It buffers the most recently received `window_size` of input audio samples and performs an FFT every `hop_size`. The FFT results are passed to the user-defined struct's `next_frame` function, after which an IFFT can be performed. The audio samples output by this process are buffered similarly to the input so they can stream in-to and out-of `FFTProcess` sample-by-sample like any other UGen. Latency is the necessary `window_size` samples. A user can specify any of the common window shapes (hann, hamming, blackman, rect, sine, kaiser, or triangle) for pre- and/or post-FFT windowing.

Similar to this FFT-based processing system, there is a system for implementing user-defined buffered processes

```mojo
struct FFTLowPass[window_size: Int =
↪  1024](FFTProcessable):
    var lpbin: Int

    fn __init__(out self, world: World):
        self.lpbin = 20

    fn next_frame(mut self, mut magnitudes:
    ↪  List[Float64], mut phases: List[Float64]) ->
    ↪  None:
        for i in range(self.lpbin,(self.window_size
        ↪  // 2) + 1):
            magnitudes[i] = 0.0
```

Listing 5: FFT brick wall low pass filter in Mojo.

that do not require an FFT or that need access to the raw input samples (for example an RMS calculation or the YIN[32] pitch detection algorithm). MMMAudio's `BufferedProcess` struct similarly buffers input and output samples and passes the most recent `window_size` samples to a user's custom struct every `hop_size` samples.

5.5 Audio Analysis

Using the `FFTProcess` and `BufferedProcess` described in Section 5.4, MMMAudio offers, for real-time and non-real-time analyses, many of the common audio descriptors found in other toolkits[1, 2, 33, 34, 35]. These include RMS, pitch (using YIN[32]), spectral centroid, spectral spread, spectral skewness, spectral kurtosis, spectral rolloff, spectral flatness, spectral crest, spectral flux (including a thresholded onset detection), mel-bands, and MFCCs. These analyses have been validated by comparing against two of the common audio analysis toolkits (FluCoMa[1] and Librosa[33]) to ensure that MMMAudio's audio analysis results match.

In order to take full advantage of the many data science and machine learning tools found in Python, it is necessary to have audio-analysis not only on the Mojo side of MMMAudio but also in Python. While Python-based audio analysis packages already exist[33, 34], if audio analyses used in Python are later to be used alongside real-time analyses in Mojo, it is important for both analyses to derive from the *same code functions*. This is because the variety of toolkits contain differences in how certain analyses are computed and variations such as floating-point accuracy can introduce discrepancies that may contaminate data.

MMMAudio takes advantage of Mojo's Python interop[11] to offer non-real-time analyses in Python using the same Mojo code that performs MMMAudio's real-time audio analyses. This ensures that Python-based MMMAudio non-real-time audio analyses will match analyses in the real-time context. To maintain full precision and smoothly integrate with Numpy-based tools, MMMAudio returns the Mojo-based analyses as Numpy data structures with data type `float64`, which can then be used with powerful toolkits such as scikit-learn[24] and SciPy[21].

5.6 Efficiency and Parallelism

A trade-off for MMMAudio's flexible, single-sample, 64-bit design is that it is less performant than SuperCollider on a single CPU. While there is no way to make a fully

legitimate comparison (MMMAudio, Max, and pd are 64 bit internally while SC is 32 bit, etc), on our M2 mac running with a 128 sample block size, MMMAudio was able to play 6000 simultaneous sine wave oscillators vs SuperCollider's 18000. Max was at just under 5000 using mc.cycle and Pure Data around 12000 using the clone object. In all cases, that is a lot of sine waves. MMMAudio's secret weapon is that, as SuperCollider can run multiple servers alongside one sclang instance, by the magic of Python's multiprocessing library, MMMAudio can run multiple audio engines from inside a single Python instance. This allows our M2 to run over 45000 oscillators at one time, all controlled by a single "control-rate" Python instance running in its own process. And getting multiple engines up and running is as simple as creating each as its own variable:

```
from mmm_python import *
m_s = []
for i in range(4):
    m_s.append(MMMAudio(128,
    ↪  graph_name="ParallelGraphs",
    ↪  package_name="examples"))
    m_s[-1].start_audio()
```

Listing 6: creating 4 MMMAudio instances with a `for` loop.

5.7 No Busses

Some CMEs introduce an extra layer of abstraction for splitting, combining, and routing audio to different parts of a program (such as SuperCollider's busses). Because MMMAudio processes audio one sample at a time and because Mojo gives parent structs access to all child member variables as deep into the data structure as desired, MMMAudio has no concept of busses. If data needs to be accessed, it can either be passed into a struct as an argument or accessed directly, not requiring a user to converse with additional data structures. This simplifies reasoning about signal flow and program design, while natively enabling sample-accurate audio routing and mixing over disparate parts of a program.

5.8 Documentation

Through MMMAudio, we aim to foster a community that welcomes many kinds of music making. To support this, we are developing comprehensive documentation and learning resources that are approachable for beginners from a range of backgrounds, while still offering the depth needed to take full advantage of the system. And our library contains detailed and great-sounding examples that demonstrate how different processes function.

Because MMMAudio enables creative decisions at every level—from low-level DSP to instrument design and performance—we expect users to extend it in their own idiosyncratic directions, rather than being funneled into the pre-existing workflows or aesthetics of the authors.

6. CONCLUSIONS

> "Is a specialized computer music language even necessary? In theory at least, I think not."
>
> -James McCartney [4]

We believe that MMMAudio shows an alternative to specialized computer music languages. It leverages Mojo / Python interop to provide a framework for expressive audio coding on par with current CMEs. By joining the Python ecosystem, it is able to take advantage of the vast library of Python packages that provide much of the functionality of the environment.

We came to create MMMAudio because we were exploring the use of neural networks to control highly dimensional synthesis and feedback systems[26]. This led us to build instruments with novel approaches to single sample feedback and DSP. In order to do that, we left our chosen CME to develop new DSP in C++ and Faust. We realized that we needed a system that could accelerate the fusion of machine learning, instrument building, and digital signal processing by making all of these practices happen in one place. MMMAudio is aimed to be this place, encompassing Faust's hard-core DSP world and the Max / SuperCollider / Pure Data instrument-building world, while at the same time allowing direct use of industry leading AI platforms. Using Mojo for both DSP and instrument design and Python for control and machine learning, we bring together these traditionally separate practices under the same development environment, knowledge base, and community.

Acknowledgments

Special thanks to Brad Garton, James McCartney, Miller Puckette, and Chris Lattner.

7. REFERENCES

[1] P. A. Tremblay, O. Green, G. Roma, J. Bradbury, T. Moore, J. Hart, and A. Harker, "The Fluid Corpus Manipulation Toolbox," Jul. 2022. [Online]. Available: https://doi.org/10.5281/zenodo.6834643

[2] N. Schnell, A. Röbel, D. Schwarz, G. Peeters, R. Borghesi *et al.*, "MuBu and friends–assembling tools for content based real-time interactive audio processing in Max/MSP," in *ICMC*, 2009.

[3] M. Puckette, "Max at Seventeen," *Computer Music Journal*, vol. 26, no. 4, pp. 31–43, 2002.

[4] J. McCartney, "Rethinking the Computer Music Language: SuperCollider," *Computer Music Journal*, vol. 26, pp. 61–68, 2002. [Online]. Available: https://api.semanticscholar.org/CorpusID:30952359

[5] ——, "SuperCollider, a New Real Time Synthesis Language," in *International Conference on Mathematics and Computing*, 1996. [Online]. Available: https://api.semanticscholar.org/CorpusID:5976007

[6] L. Fridman and C. Lattner. Chris Lattner: Future of Programming and AI | Lex Fridman Podcast. [Online]. Available: https://www.youtube.com/watch?v=pdJQ8iVTwj8

[7] Modular.com. Mojo: Powerful CPU+GPU Programming. [Online]. Available: https://www.modular.com/mojo

[8] G. Wakefield, "gen~," *unpublished*, 2011.

[9] SuperCollider Help. Writing Unit Generators. [Online]. Available: https://doc.sccode.org/Guides/WritingUGens.html

[10] D. F. Y. Orlarey and S. Letz, "Faust: An efficient functional approach to dsp programming," *unpublished*, 2009.

[11] Modular.com. Python Interoperability. [Online]. Available: https://docs.modular.com/mojo/manual/python/

[12] Mido Developers. mido. [Online]. Available: https://github.com/mido/mido

[13] P. Rusnak. hidapi. [Online]. Available: https://github.com/trezor/cython-hidapi

[14] T. Attwad. python-osc. [Online]. Available: https://github.com/attwad/python-osc

[15] J. D. Hunter, "Matplotlib: A 2D graphics environment," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.

[16] A. Sweigart. pyautogui. [Online]. Available: https://github.com/asweigart/pyautogui

[17] Qt Group. PySide6. [Online]. Available: https://doc.qt.io/qtforpython-6/index.html

[18] H. Pham. PyAudio. [Online]. Available: https://people.csail.mit.edu/hubert/pyaudio/

[19] R. Bencina and P. Burk, "PortAudio - an Open Source Cross Platform Audio API," in *International Conference on Mathematics and Computing*, 2001. [Online]. Available: https://api.semanticscholar.org/CorpusID:32143419

[20] C. R. Harris, K. J. Millman, S. J. van der Walt, *et al.*, "Array programming with NumPy," *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020. [Online]. Available: https://doi.org/10.1038/s41586-020-2649-2

[21] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, *et al.*, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020.

[22] A. Paszke, S. Gross, F. Massa, *et al.*, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-\deep-learning-library.pdf

[23] M. Abadi, A. Agarwal, P. Barham, *et al.*, "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[24] F. Pedregosa, G. Varoquaux, A. Gramfort, *et al.*, "Scikit-learn: Machine learning in Python," *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[25] Modular.com. Modular MAX. [Online]. Available: https://www.modular.com/max

[26] S. Pluta, "Multi-mapped Neural Networks for Control of High Dimensional Synthesis Systems," in *Proceedings of the 2nd Conference on AI Music Creativity*, 2021. [Online]. Available: https://aimc2021.iem.at/papers/

[27] D. Schwarz, G. Beller, B. Verbrugghe, and S. Britton, "Real-time corpus-based concatenative synthesis with catart," in *9th International Conference on Digital Audio Effects (DAFx)*, 2006, pp. 279–282.

[28] L. McInnes, J. Healy, and J. Melville, "Umap: Uniform manifold approximation and projection for dimension reduction," *arXiv preprint arXiv:1802.03426*, 2018.

[29] B. E. Granger and F. Pérez, "Jupyter: Thinking and Storytelling With Code and Data," *Computing in Science Engineering*, vol. 23, no. 2, pp. 7–14, 2021.

[30] Modular.com. SIMD. [Online]. Available: https://docs.modular.com/mojo/stdlib/builtin/simd/SIMD/

[31] Wikipedia. Single instruction, multiple data. [Online]. Available: https://en.wikipedia.org/wiki/Single_instruction,_multiple_data

[32] A. De Cheveigné and H. Kawahara, "YIN, a fundamental frequency estimator for speech and music," *The Journal of the Acoustical Society of America*, vol. 111, no. 4, pp. 1917–1930, 2002.

[33] B. McFee, C. Raffel, D. Liang, D. P. Ellis *et al.*, "librosa: Audio and music signal analysis in python," in *Proceedings of the 14th python in science conference*, vol. 8, 2015.

[34] D. Bogdanov, N. Wack, E. Gómez, S. Gulati *et al.*, "Essentia: An audio analysis library for music information retrieval," in *Britto A, Gouyon F, Dixon S, editors. 14th Conference of the International Society for Music Information Retrieval (ISMIR); 2013 Nov 4-8; Curitiba, Brazil.[place unknown]: ISMIR; 2013. p. 493-8.* International Society for Music Information Retrieval (ISMIR), 2013.

[35] G. Peeters *et al.*, "A large set of audio features for sound description (similarity and classification) in the CUIDADO project," *CUIDADO Ist Project Report*, vol. 54, no. 0, pp. 1–25, 2004.

DANCING CABIRIA: A HYPER-ENVIRONMENT STUDY THROUGH CORPUS-BASED TECHNIQUES

Piero Poli
Conservatorio Statale di Musica Giuseppe Verdi di Torino
piero.poli@conservatoriotorino.eu

ABSTRACT

This paper introduces the concept of hyper-environment — an additional spatial layer superimposed on the choreographic space, where physical movement becomes a means of navigating and activating pre-analyzed sound materials. The work examines Dancing Cabiria, a reenactment in four scenes from Giovanni Pastrone's silent film Cabiria (1914), as a case study to explore performative hyper-environments that employ corpus-based synthesis techniques within a virtual reality framework. Through the use of motion-tracking suits, four choreographies are performed, each one by four dancers whose movements are translated into sound via audio corpora distributed throughout the virtual space surrounding each performer. Each choreography outlines different uses and configurations of this hyper-environment, and allow for the discussion of compositional and instrumental issues such as the scale and density of the corpora, the relationship that emerges between movements width, corpus dimensions, and virtual space volume, and the role of real-time feedback in the design of hybrid instruments for performers.

1. CONTEXT AND INTRODUCTION

Living Cabiria is a research project that investigates how the digital translation of a silent film classic can activate new forms of sensory and emotional participation in the viewer more than a century after its creation. Within this context, *Dancing Cabiria* focuses on the relationship between dance movement, choreographic space, and the construction of immersive sound environments in VR. In this work, the performer's body is not only a source of data for sound control, but becomes the center of a *hyper-environment*: an additional virtual space populated by audio corpora that surround the dancer and can be explored, traversed, and touched through gesture [1] . Through four choreographies created by Letizia Gioia Monda and inspired by four different scenes from the film, the project develops and compares different strategies for mapping corpora in virtual environments in which to immerse the performer, with the aim of understanding how the morphology of movement and the topology of the corpus contribute to performer's kinesthetics, the awareness of a hyper-environment of a solely sonic nature, and to the development of new compositional methods. Designed for virtual reality, these choreographies place the viewer directly inside the same environment as the performers, enabling shared access to the choreographic space and to the hyper-environments developed within it. The article proposes *Dancing Cabiria* as a case study to discuss the design of hybrid corpus-based hyper-environments, at the convergence of composition, digital instrument design, and performance studies. After a brief overview of corpus-based techniques and the concept of hyper-environment, the technical pipeline for motion capture and sonification and the four configurations developed in the project are presented. The final discussion highlights the practical and compositional implications that have emerged and outlines some future developments towards more open, interactive, and real-time-oriented systems.

[1] This term will be investigated in detail in Section 4.

Copyright: ©2026 Piero Poli. This is an open-access article distributed under the terms of the Creative Commons Attribution License 3.0 Unported, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

2. EVOLUTION OF CORPUS BASED TECHNIQUES

Corpus-based synthesis techniques emerged in the late 1990s as a creative extension of concatenative synthesis methods initially developed for speech synthesis, applying segmentation, descriptor analysis, and selection strategies to large sets of recorded sounds [1]. In this context, the work of Diemo Schwarz and the CataRT system played a central role in defining the practice of interactive corpus-based synthesis, in which the exploration of a space of descriptors replaces the direct control of synthesis parameters [2, 3]. Recent projects such as *FluCoMa* [2] integrate machine learning and data mining techniques to enable the creation, manipulation, and programmable exploration of corpora within computer-based composition environments [4].

Corpus-based synthesis is hybrid in nature and exploits a plethora of different techniques and principles, not necessarily developed for sound generation: it shares the content-based approach of granular synthesis, but implements it through analysis (MIR [3]) and cataloguing. With the evolution of technology, increased computing power, processing speed, and more accessible costs, experiments using these types of techniques have taken into account different aspects of the synthesis process [5], also leading to an evolution towards aspects not limited to sound production [6]: corpus-based techniques can be used as a tool for assisted composition, for the rearrangement of audio materials, orchestration, and reorchestration [7], for the construction of

[2] https://www.flucoma.org/
[3] Music information retrieval.

augmented instruments [8], for the development of performances [3], or for the creation of improvisational contexts exploiting haptic feedback [9, 10]; they represent a method of synthesizing sound as much as an exploratory approach to composition.

3. CORPUS-BASED PERFORMANCE PRACTICES

Parallel to their technological development, the performative, compositional, and artistic possibilities of these techniques have generated multiple works, many of which consider the corpus not only as starting material but as a true musical action space that can be explored through gesture. In *Surfing the Waves* [11], Tremblay and Schwarz use the electric bass to navigate through a sound corpus, turning the bass into a medium between instrumental practice and a new method of timbre exploration. Schwarz further investigates possible methods and interfaces for corpus exploration [5, 3]: the corpus space becomes a compositional and performative tool, again with Tremblay in *Rich Contacts* [12]. Spatial and timbre dimensions can also form a close correlation, as analyzed by Einbond in *Spatializing Timbre with Corpus-Based Concatenative Synthesis* [13], where timbre and gestural features are linked to sound distributions in space. In more recent years, artists and researchers such as Tanaka, Visi, Di Donato, and Zbyszyński [14, 15] have exploited the opportunities offered by machine learning to develop mappings between motion capture and concatenative synthesis, controlling timbre transformations of corpora and pushing gesture – corpus mapping toward advanced gestural sonifications. A different perspective is offered by Laczkó's *The Hum* [16], in which a corpus of nearly five thousand recorded sound gestures is projected into a navigable 3D space via UMAP, explored in real time through a virtual camera within a spatial audio setup. These works lay the groundwork for the idea of the hyper-environment proposed here: a three-dimensional corpus-based sound space around the performer, fully navigable through bodily movement in virtual reality.

4. HYPER-ENVIRONMENT

The hyper-environment functions as an additional and invisible sonic context relative to the performer, perceptible exclusively through hearing and integrated with their kinesthetic awareness of the body in space. The idea emerges to develop a compositional method based on the interaction of a performer with their surroundings, where the corpus becomes a navigable space and the gesture–sound correlation takes on spatial and distributed dimensions [4] . This virtual environment — populated by corpora distributed around the performer — acts as a performative narrative instrument: it can evolve during the performance, be shared between performer and audience, and requires active exploration to reveal its internal structure. Unlike the traditional correlation between gesture and sound, the hyper-environment immerses the performer (through sensors that detect their position) in a virtual space shared with the corpus, where the real and sonic dimensions overlap: it is no longer the specific movement that generates a predetermined sound but the position in which that movement is executed within the distributed volume, and only through bodily trajectory the internal structure of this invisible space becomes revealed. As Rebelo observes that sound perception is intrinsically linked to the concept of performance [17], this correlation between physical and virtual spaces takes on an inherently exploratory nature: we are accustomed to associating movement and sound, expecting a visible gesture to correspond to an expected noise. Our ability to predict sound necessarily depends on the perception of the surrounding environment and on our lived experience: in the same way that we become disoriented in an unfamiliar context — lacking knowledge of its affordances and needing to learn through exploration — the hyper-environment induces a similar process both in the performer, who must rediscover the sonic potential of their body within the invisible space, and in the spectator, accustomed to a direct movement – sound correlation and now disoriented by its positional and distributed transformation.

A hyper-environment is configured as an additional sound layer on top of the perceptible acoustic environment, analogous to the spectral stratification of light: we perceive only the visible range of colors, while infrared and ultraviolet rays coexist invisibly and can only be seen through means that enhance our perception. Similarly, the hyper-environment adds layered sound layers — composed of multiple bodies distributed, alternating, or overlapping in virtual space — which the performer explores and activates through movement, revealing a richness of timbre and space that is not immediately accessible but can be experienced via loudspeakers diffusion.

5. MOTION CAPTURE AND SONIFICATION PIPELINE

The technical implementation of Dancing Cabiria is based on a motion capture pipeline that connects full-body tracking to corpus-based sound synthesis within a unified Max environment, using packages as bach [5] , dada [6] , ears [7] , FluCoMA, and SP Tools [8] for analysis and database organization and SPAT [9] for spatialization.

5.1 Motion capture setup

Dancers wear Xsens motion capture suits [10] that stream position and orientation data for a selected set of joints [11] at 240 fps into Unity through the MVN Unity Live Plugin [12] . [13] . To avoid fragmenting the workflow across multiple software environments, this data is then forwarded via OSC

[4] The hyper-environment proposed here is an operational definition specific to this study, to be validated through further performative and perceptual research.

[5] https://www.bachproject.net/

[6] https://github.com/bachfamily/dada

[7] https://www.bachproject.net/ears/

[8] https://rodrigoconstanzo.com/sp-tools/

[9] https://forum.ircam.fr/projects/detail/spat/

[10] https://www.xsens.com/motion-capture/suits

[11] The possible sensors used are: Shoulder R/L, Elbow R/L, Wrist R/L, Hips, Head, Thigh R/L, Knee R/L, Ankle R/L. Additionally, sensors on the gloves are also available, one for each finger.

[12] https://base.xsens.com/s/article/MVN-Unity-Live-Plugin?language=en_US

[13] Inside Unity, each body segment is represented as a GameObject associated with one or more Transform or Rect Transform component which encodes global and local position, rotation, and scale

to Max, where it is cleaned, organized into a structured movement database using the *dada.base* object. Noisy or redundant streams are filtered out, sensor selections are reduced to a subset of joints relevant for each choreography, and all data are normalized to facilitate subsequent mapping to corpus descriptors and virtual space.

5.2 Integrated workflow in Max

To improve data accessibility and streamline the workflow within a unified environment, low-resolution video clips of the Unity mannequin choreographies were synchronized with the automation of the movements within the database, providing a visual correlation for sonification during playback. This structure allowed offline sonification, giving the composer greater freedom to experiment and enabling precise synchronization with the final video footage.

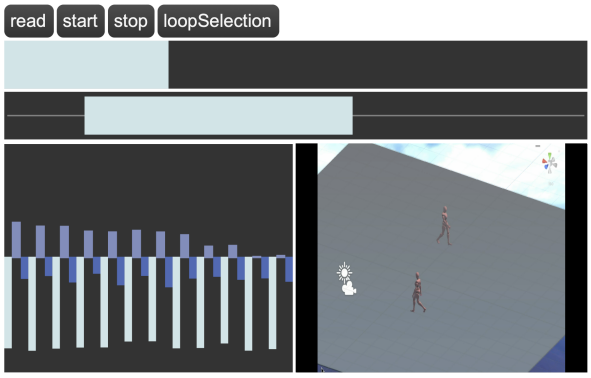


Figure 1. Database interface for handy motion data extraction and video synchronization.

5.3 Corpus creation and analysis

The pre-analysis of the audio material for corpus creation was performed using the FluCoMa library, specifically leveraging novelty [14] and onset [15] detection features, which provided precision and customization in fragmenting the source material into semantically coherent units. Feature extraction and corpus cataloging utilized a comprehensive set of descriptors computed via both FluCoMa objects (for MFCC and chroma) and the *Essentia* [16] analysis library integrated through *ears.essentia* [17] . This combination enabled robust normalization across heterogeneous scales, facilitating the projection of these features into low-dimensional virtual spaces (2D planes or 3D volumes) navigable by performer trajectories.

[14] https://learn.flucoma.org/reference/noveltyfeature/

[15] https://learn.flucoma.org/reference/onsetfeature/

[16] https://essentia.upf.edu/algorithms_reference.html

[17] A typical corpus of Dancing Cabiria is composed of 500-2500 units with duration between 50-2000ms, analyzed by a combination of: 13 MFCC, 12 chroma and multiple spectral descriptors (spectral complexity, spectral energy, spectral centroid, zero crossing rate, pitchyin, strong peak, rolloff, max magnitude frequency, spectral skewness, spectral kurtosis, chords detection, and inharmonicity) via *fluid.bufmfcc* , *fluid.bufchroma* , *ears.essentia* . For high-dimensional descriptors like MFCCs (typically 13+ coefficients), FluCoMa handles the dimensionality reduction via *fluid.umap.*.

5.4 Corpus-based sonification strategies

Sensor data aggregation transforms high-dimensional motion capture streams into low-dimensional control parameters suitable for corpus-based synthesis, addressing the challenge of mapping complex body trajectories to descriptor space navigation. Local movement velocities modulate unit extraction rates — ensuring sound production correlates with gestural dynamics rather than static poses — and the target vector for k-NN search is computed as the mean position of relevant joints [18] . Additional sensor orientations provide post-selection parameter modulation (pitch range, envelope shaping), with optional stochastic perturbations preventing repetitive sequences and maintaining timbre variability across sustained trajectories. Specific implementations of these mapping principles across each of the four choreographies are detailed in Section 7.

5.5 Spatialization and audio rendering

Spatialization was achieved in Max using the SPAT package developed by IRCAM for an octophonic system, adopting two main strategies that link sound position to movement data or the intrinsic distribution of the corpus in the virtual space. In individual volumetric and planar corpus distribution cases (see Sections 7.2 and 7.1), the dancers' hips sensors determine the position of the sound source in the listening space, with the stereo spread calculated based on the distance from a virtual central point (corresponding to the camera); in the shared corpus distribution configuration (see Section 7.4), on the other hand, each unit maintains a fixed position in the octophonic space based on its location in the volume of the corpus, simulating stationary sound objects. The final multichannel audio is then rendered binaurally for the Unity VR environment, preserving the spatial immersiveness of the original system.

6. HYPER-ENVIRONMENT CONSTRUCTION: CORPUS MAPPING TO 3D SPACE

The creation of hyper-environments involves immersing the performer within a virtual 3D space populated by the sound corpus. By leveraging motion-tracking data, the volume of this environment can be calibrated to match various scales according to specific compositional objectives. This space can either faithfully replicate the physical performance environment through 3D photogrammetry or take on arbitrary shapes and proportions for exploratory and narrative purposes.

The corpus units are spatially distributed based on their normalized spectral descriptors (e.g., MFCCs, spectral centroid, chroma). Each spatial axis (x, y, z) is mapped to a specific acoustic attribute or set of parameters. To optimize navigability and ensure an even spread of sounds, the extracted values undergo a quantile normalization process, defined as:

$$x' = F^{-1}(u), \quad u \sim U[0,1] \tag{1}$$

where F represents the empirical Cumulative Distribution Function (CDF) of the descriptors. This transformation

[18] Mean aggregation ensures robustness against noisy single-joint readings and smooth trajectory transitions.

re-maps the original distribution into a uniform space, preventing the formation of overly dense clusters or empty regions that would hinder interaction. To maintain a balance between global spatial uniformity and the original timbral relationships, a linear interpolation is applied:

$$x_{final} = \alpha x' + (1 - \alpha) x_{raw} \quad (2)$$

The weighting factor α allows the composer to tune the distribution, blending the mathematical regularity of the quantile map with the original clustering of the raw descriptors — as shown in fig.2 and fig.3. Finally, to adapt the corpus distribution to the performer's actual movements, the Cartesian coordinates are further refined using manually designed transfer functions for each axis. These functions allow for the targeted compression or expansion of specific regions of the populated space, tailoring the interactive volume to the observed performative needs.

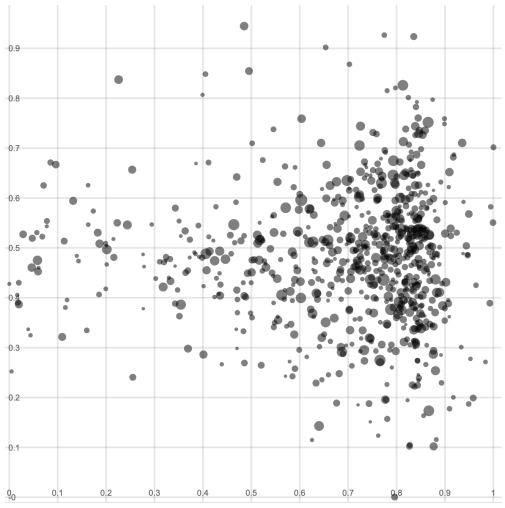


Figure 2. Normalized corpus distribution using values of pitch, inharmonicity, spectral skewness, and spectral energy.

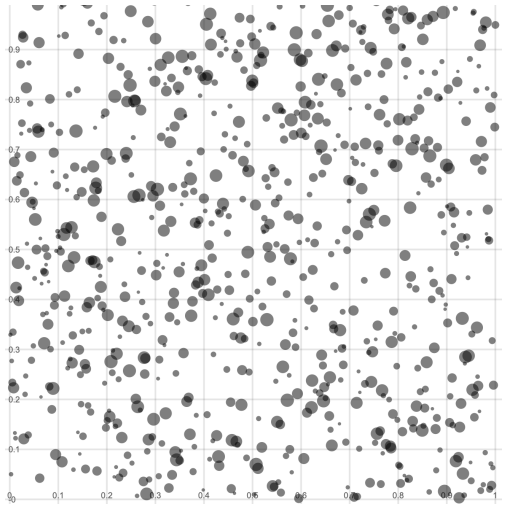


Figure 3. Same normalized corpus of fig.2 warped via quantile transformation.

This iterative approach to space deformation is made possible by the graphical visualization of data, taking inspiration from the redistribution processes proposed by Diemo Schwarz [5] where descriptor spaces are actively navigated as performative instruments. The result of these operations is a reproducible and semantically coherent new environment of sound — the hyper-environment —, where physical proximity in the virtual volume corresponds to timbral similarity between units, immersing the performer within the sound of the corpus itself.

7. FOUR HYPER-ENVIRONMENT CONFIGURATIONS IN *DANCING CABIRIA*

7.1 Volumetric exploration

The choreography, filmed at Parco Dora in Turin, explores the gigantic body motif from Cabiria through an A-B-A structure: the dancers begin with synchronized movements from an outer circle, converge toward the camera in a diachronic manner, reaching an inner circle close to the ground, and then expand again, returning to synchrony. As can be seen in fig. 4, each hyper-environment is configured as a three-dimensional volume that envelops the performer's entire body like a sphere, populated by a corpus distributed according to the maximum extension of the limbs. The average position of each joint group (shoulder-elbow-wrist for arms, thigh-knee-ankle for legs, plus hips) defines the target for k-NN search in the corpus, while the local average speed of these sensors determines the extraction rate of the units, [19] and the rotation of the hands modulates the reproduction pitch range. This configuration produces a 32-voice polyphony per dancer (8 per limb): each limb gesture extracts specific units from the personal corpus surrounding the body, revealing a clear relationship between the scale of body movements, the size of the distributed corpus, and the immersive perception of virtual space.

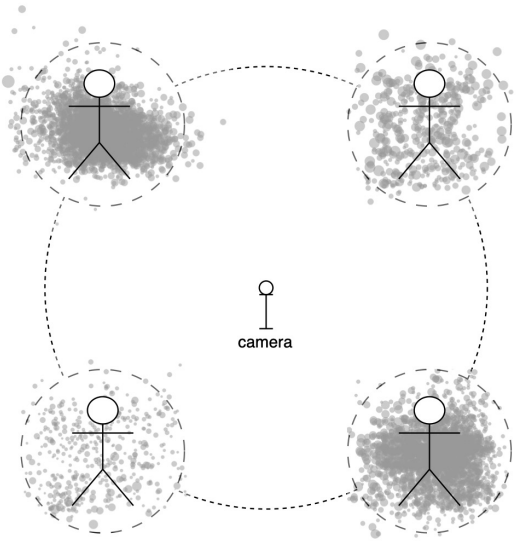


Figure 4. Corpora displacement around each performer.

[19] Throughout this paper, the term "units" refers to segmented sound fragments, following the foundational terminology of corpus-based concatenative synthesis (e.g., "units" and "targets" in Schwarz) [1], rather than more recent variants.

7.2 Planar exploration

Filmed on the Murazzi along the Po river in Turin, the choreography explores the theme of navigation through an A-B-C structure: the dancers trace wide, independent trajectories that converge only in the final section in synchronous movement, amplifying the sensation of motion thanks to the central 360° camera. The hyper-environment is configured as a corpus mapped onto a two-dimensional plane superimposed on the choreographic space, as shown in fig. 5. The hips position determines the target for the extraction of units from the corpus, with stochastic noise added to the position to avoid consequential repetitions of the same units and high corpus dimensions to ensure timbre fluidity and coverage of the entire movement area. The performer navigates extended soundscapes following broad trajectories in space, creating a sound trail in their wake that gradually reveals the characteristics of the environment.

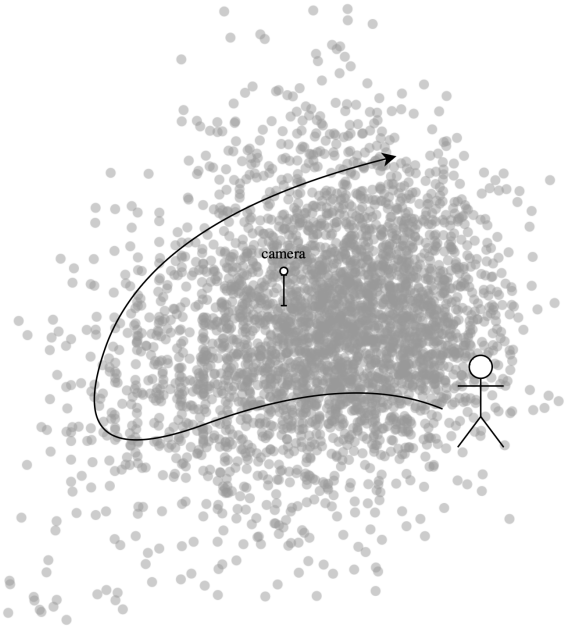


Figure 5. Corpus displacement in a plane for each performer.

7.3 POV exploration

The third choreography, filmed in the courtyard of Palazzo Carpano in Turin, explores the theme of orientalism of Pastrone's work through an intimate A-B structure in pairs: the first section focuses on the slow movements of the hands and arms of two dancers lying down, while the second shifts the focus to another couple, accentuated by the POV and 360° shot. As depicted in fig. 6, the hyper-environment is configured as a virtual sphere around the upper extremities of the performer, populated by vocal nature corpora, with consonant sounds removed via FluCoMa to generate an evanescent and dreamlike sensation. The relative position of the hands defines the k-NN target, their speed modulates the playback rate of the units, the elbows control the SPAT envelope and reverb, while the finger sensors classify additional parameters. The micro-gestures of the extremities reveal gradients of harmonicity-inharmonicity, testing bodily proximity and fine granularity in the exploration of the perimeter corpus.

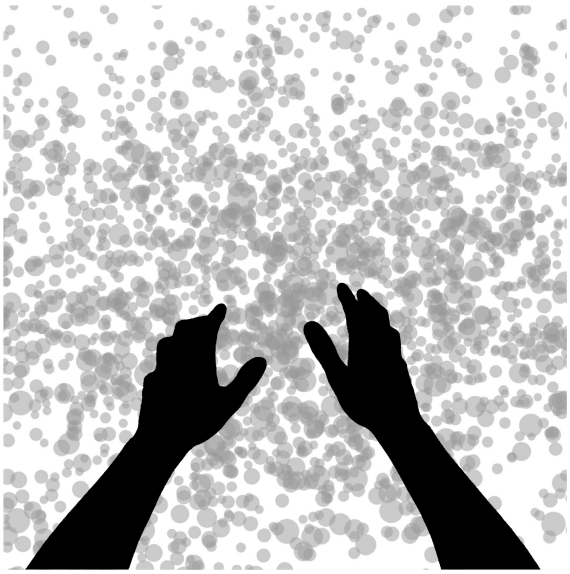


Figure 6. POV corpus exploration.

7.4 Shared planar exploration

The last choreography, filmed in the Gallery of Kings at the Egyptian Museum in Turin, explores the theme of symmetry through an A–B structure: it begins as a canon with a four-beat offset sequence of movements, evolves into the cyclic repetition of a single gesture, and ends with the collective synchronization of the full sequence. The hyper-environment shown in fig. 7 is configured as a space shared by all dancers, populated by a single corpus generated from ambient and vocal sounds recorded during rehearsals, with units of particular short duration (5–50 ms). Each performer's limbs explore the shared environment through a mapping similar to the volumetric exploration's one (see Section 7.1), but in this case the spatialization of each unit depends on its position in the virtual space rather than on the performer's movements. As in the second choreography, noise was added to unit extraction to minimize potential consecutive repetitions of corpus material. Through gestures, this configuration activates a stable, reactive environment of immobile sound objects, probing a shared sound-field space and fostering a common perception in the exploration of the environment.

8. FURTHER CONSIDERATIONS

Dancing Cabiria functioned as a laboratory for testing corpus-based hyper-environment design principles, revealing both strengths and critical issues in the relationship between gesture, corpus distribution, and immersive perception. The following subsections address four aspects emerging from the four configurations: the distribution of performative control across choreographic space, the relationship between movement scale and corpus topology, the perceptual asymmetry between performer and audience, and the technical strategies adopted to ensure timbre fluidity.

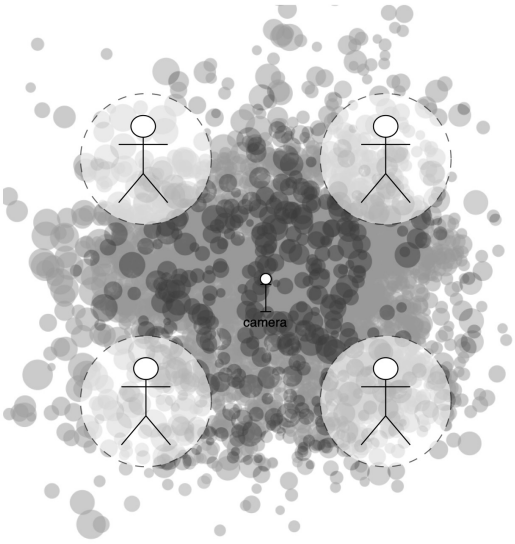


Figure 7. corpus displacement in the choreographic space.

8.1 Shared awareness between performer and audience

The four choreographies outline a continuum of agency that evolves from local precision to global sharing. In volumetric exploration (see Section 7.1), each limb acts as an independent probe within the performer's personal sound space, offering immediate deterministic control but limited timbre surprise. Planar exploration (see Section 7.2) shifts the focus to extended trajectories, slowing corpus navigation and allowing a gradual discovery of sonic materials. POV exploration (see Section 7.3) concentrates this agency in the extremities — hands and fingers — enabling fine-grained navigation of timbral gradients within a high-density corpus region. Shared planar exploration (see Section 7.4) takes these concepts and makes it collective: each individual gesture activates a shared sound field where the perception of one's own contribution intertwines with that of the other dancers.

8.2 Movement scale and corpus topology

Comparative analysis of different movement amplitudes across configurations confirms that corpus topology must adapt to gesture morphology to maintain perceptual continuity. Shared planar exploration (see Section 7.4) (2420 units, 4m x 8m x 2m volume) sustained macro-gestures ($v > 1m/s$), while POV micro-explorations (see Section 7.3) (740 units, 0.5m radius sphere) captured fine gestures ($v < 0.3m/s$). The size of a corpus depends both on the extraction method and on the extent of the virtual space in which it is distributed: micro-gestures require highly dense corpora with many small units (50-150ms, $\Delta MFCC > 0.2$ between neighbors) to capture subtle timbre variations, while longer units (500-1500ms, $\Delta MFCC < 0.1$) need noise injection (*std* = 0.05-0.08) to avoid repetition — a problem common with undersized corpora. Since distances in virtual environments are relative, distribution volume varies with movement scale that traverses it. Ultimately, movement scale determines corpus size and structure: ignoring this relationship results in loss of immersion, as directly observed across the four choreographies.

8.3 Shared disorientation: performer and spectator in the hyper-environment

The hyper-environment generates a dual learning process that involves performer and audience in an asymmetrical yet complementary way. The performer discovers corpus affordances incrementally through movement, since the spatial structure of the hyper-environment is not directly perceivable. The spectator, accustomed to direct gesture–sound correlations, experiences perceptual disorientation: they observe trajectories that activate unpredictable soundscapes, noting movement sequences whose sonic outcome is opaque without prior knowledge of the corpus structure. Volumetric and planar explorations (see Sections 7.1 and 7.2) accentuate this divergence, emphasizing monumental scale or extended paths; POV and shared volumetric explorations (see Sections 7.3 and 7.4) instead create convergence through tactile intimacy and collective synchronicities. This asymmetry between performer and audience could be further explored as a compositional parameter to structure the hyper-environment as a shared perceptual space.

8.4 Technical strategies for corpus fluidity and perceptual smoothness

To ensure timbre fluidity even with relatively small corpora, slightly random variation of audible characteristics proved effective: small adjustments in pitch, envelope, and filtering created perceived homogeneity and masked repetitions, while simultaneous playback of multiple units (polyphony) leveraged natural auditory masking. Descriptor normalization — essential given their heterogeneous scales — facilitated spatial visualization and exploration of corpora, ensuring dimensional coherence. Finally, units duration emerges as a key factor in timbre quality: the same source material, segmented into short units (50 ms) versus long ones (1-2000 ms), yields drastically different results, since analytic features (computed over temporal averages) vary substantially, influencing both synthesis and perception.

9. CONCLUSIONS

The *Dancing Cabiria* case studies revealed the concrete possibility of a compositional approach that integrates corpus-based synthesis techniques to structure three-dimensional exploratory hyper-environments around the performer, shared in real time with the audience. This method transforms the corpus from static sound material into a space navigable through bodily movement, promoting volumetric exploration that accounts for movement scale and morphology, in continuity with existing practices but with particular attention to a shared awareness of the environment.

It is also necessary to note that true real-time feedback between performer and instrument occurred only during testing phases, not in final sonification. Starting from the author's initial experiments, and also during final project design, it became clear that solo configurations hold greater

potential as compositional method — where the hyper-environment functions less as collective instrument and more as dynamic compositional partner, enabling genuine interaction between performer sensation and environment mutation. This distinction reveals hyper-environments' hybrid nature: an instrument for pre-choreographed ensemble performance, where gestures activate stable shared soundfields, or an exploratory composition tool for individual work, where reciprocal adaptation between body movement and sonic space drives emergent musical structure. The offline approach of Dancing Cabiria thus serves as proof-of-concept for real-time systems where such feedback becomes central to the performative process.

10. FUTURE DEVELOPMENTS

This work provided many starting points for further development, as the hybrid nature of hyper-environments represents a crossroads of multiple directions to be explored in the fields of HCI, [20] composition, environment design, instrument design, and new musical interfaces. A critical limitation of purely sonic hyper-environments is the performer's reliance on trial-and-error auditory exploration to discover corpus spatial affordances. Future developments propose a multimodal feedback layer with ambient lighting to indicate local corpus density variations and body vibrotactile vests activating distributed vibrations to signal proximity to timbral boundaries or significant gradients.

This third perceptual layer (sonic + visual + tactile) transforms the hyper-environment from invisible space to consciously navigable instrument. Feedback becomes a structural compositional parameter: hidden clusters without signals for timbral surprises, dense zones marked for controlled exploration.

Additional development of the hyper-environment will focus on dynamic narrative construction: real-time modification of corpora distribution and composition as the performance unfolds, enabling sound narratives that evolve dynamically with the performance trajectory. Layered mixed hyper-environments will be explored, with body areas dedicated to different scales — micro-gestures of the extremities with short, dense units for fine timbre details, extended body volumes with medium units for broad soundscapes, and shared fields with variable durations for smooth transitions — creating multi-layered sound structures that respond to specific gesture morphology. Another point of interest lies in the dynamic transformation of elements within the corpus through the construction of a structure that accounts for physics in virtual spaces, allowing the performer to change individual units by moving them through space like real objects.

Acknowledgments

The author gratefully acknowledges the Conservatorio Statale di Musica Giuseppe Verdi di Torino and the Museo del Risorgimento di Torino, where this research is being conducted as part of a doctoral program, for institutional support and funding, with particular thanks to their supervisor, Andrea Agostini, and Pierre Alexandre Tremblay for their mentorship. The author also thanks Professor Letizia Gioia Monda, project director, for her collaboration and invaluable guidance throughout the development process, and Professor Andrea Valle for the given opportunity, as well as Nicolò Pilon and the Balletto Teatro di Torino for their professionalism.

[20] Human Computer Interaction.

11. REFERENCES

[1] D. Schwarz, "Concatenative sound synthesis: The early years," *Journal of New Music Research*, vol. 35, no. 1, pp. 3–22, 2006.

[2] ——, "A system for data-driven concatenative sound synthesis," in *Digital Audio Effects (DAFx)*, 2000, pp. 97–102.

[3] ——, "Interacting with a Corpus of Sounds," in *Symposium on Sound and Interactivity. Singapore*, 2013.

[4] P. A. Tremblay, G. Roma, and O. Green, "Enabling programmatic data mining as musicking: The fluid corpus manipulation toolkit," *Computer Music Journal*, vol. 45, no. 2, pp. 9–23, 2021.

[5] D. Schwarz, "The sound space as musical instrument: Playing corpus-based concatenative synthesis," in *New Interfaces for Musical Expression (NIME)*, 2012, pp. 250–253.

[6] D. Ghisi and C. Agon, "Real-Time Corpus-Based Concatenative Synthesis for Symbolic Notation," in *Proceedings of the International Conference on Technologies for Music Notation and Representation*, 2016, pp. 7–13.

[7] D. Schwarz, S. Britton, R. Cahen, and T. Goepfer, "Musical applications of real-time corpus-based concatenative synthesis," in *International Computer Music Conference (ICMC)*, 2007, pp. 47–50.

[8] L. Wolstanholme, J. Shier, R. Constanzo, and A. McPherson, "Drum Modal Feedback: Concept Design of an Augmented Percussion Instrument," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2025, pp. 108–115.

[9] L. S. Hayes. (2020) Event Detection and Improvisation: Moon via Spirit. FluCoMa Project, Learn.FluCoMa. [Online]. Available: https://learn.flucoma.org/explore/hayes/

[10] D. Schwarz, "Touch Interaction for Corpus-based Audio–Visual Synthesis," in *Proceedings of New Interfaces for Musical Expression (NIME 2023)*, 2023. [Online]. Available: https://www.imera.fr/wp-content/uploads/2023/06/schwarz-nime-2023-cocavs.pdf

[11] P. A. Tremblay and D. Schwarz, "Surfing the waves: Live audio mosaicing of an electric bass performance as a corpus browsing interface." NIME, 2010.

[12] D. Schwarz, P. A. Tremblay, and A. Harker, "Rich contacts: Corpus-based convolution of audio contact gestures for enhanced musical expression," in *New Interfaces for Musical Expression (NIME)*, 2014, pp. 1–1.

[13] A. Einbond and D. Schwarz, "Spatializing timbre with corpus-based concatenative synthesis," in *ICMC*. Michigan Publishing, 2010, pp. 72–75.

[14] F. G. Visi and A. Tanaka, "Interactive machine learning of musical gesture," in *Handbook of artificial intelligence for music: Foundations, advanced approaches, and developments for creativity*. Springer, 2021, pp. 771–798.

[15] M. Zbyszyński, B. Di Donato, F. G. Visi, and A. Tanaka, "Gesture-timbre space: Multidimensional feature mapping using machine learning and concatenative synthesis," in *International Symposium on Computer Music Multidisciplinary Research*. Springer, 2019, pp. 600–622.

[16] B. Laczkó, "The Hum: 3D Audiovisual Live Performance for Projection and Spatial Audio," in *Proceedings of the 29th International Conference on Auditory Display*, Troy, NY, USA, 2024, pp. 201–203.

[17] P. Rebelo, "Performing space," *Organised Sound*, vol. 8, no. 2, pp. 181–186, 2003.

Music of Changing Lines: Toward a Culturally Situated Approach to the I-Ching

Ling Qi*
Georgia Institute of Technology
lqi60@gatech.edu

Teng Aleksandra Ma*
Georgia Institute of Technology
tma98@gatech.edu

Alexandria Smith
Georgia Institute of Technology
alexandria.smith@gatech.edu

*These authors contributed equally to this work.

ABSTRACT

The I-Ching is one of the most influential texts in Chinese intellectual history, integrating divination, cosmology, and ethical reflection. While Western experimental music, most notably John Cage, has drawn on the I-Ching as a source of chance operation, such appropriations have often detached its formal mechanisms from the interpretive and philosophical processes that give the text meaning. This work, Music of Changing Lines, presents an interactive system that re-centers the I-Ching as a meaning-bearing framework rather than a neutral randomizer. Users perform Wen Wang Fa coin casting, which is accompanied in real time through probabilistic musical processes. The resulting hexagrams and changing lines are interpreted by a large language model, Gemini, in relation to the user's inquiry. This textual interpretation is then translated into a prompt for a generative music model, Lyria, producing a responsive musical realization. By situating AI as an interpretive intermediary rather than a compositional authority, the system foregrounds the I-Ching's ritual, interpretation, and participation as the primary sonic materials. Music of Changing Lines extends process-driven traditions in computer music by demonstrating how generative AI can support participatory, meaning-driven musical processes without prescribing musical structure or replacing human agency.

1. INTRODUCTION

The *I-Ching* [1], or *Book of Changes*, has circulated for more than three millennia as both a divinatory system and a philosophical text. Its hexagrams (Figure 1), generated through ritualized chance procedures such as the Wen Wang Fa coin-toss method, provide a framework for interpreting uncertainty, change, and the unfolding of events. Through centuries of commentary and reinterpretations, most notably within the Confucian tradition, the text's symbols have been read as both oracular signs and ethical or metaphysical principles.

In the twentieth century, the *I-Ching* entered Western experimental music most prominently through the work of John Cage. Cage was introduced to the *I-Ching* by Christian Wolff, whose father published its English translation [3]. Drawn to the use of chance, Cage recontextualized the *I-Ching* as a compositional tool in works such as *Music of Changes* [4] and *Music for Piano 21–52* [5]. He used coin tosses to form hexagrams, which then determined values for musical parameters such as sound type, duration, and dynamics. Despite Cage's prior personal engagement with the text as a source of philosophical and life guidance [6], this approach largely set aside the interpretive and cosmological readings of the hexagrams that traditionally follow the casting process. Hence, while these works marked the first explicit introduction of the *I-Ching* into Western experimental music, they deviated from the *I-Ching*'s philosophical and divinatory context.

In this work, *Music of Changing Lines*, we revisit the *I-Ching* not merely as a mechanism for randomness but as a culturally situated system of meaning. Our creative goal is to design an immersive, interactive divination experience that translates the full casting and interpretive process of the *I-Ching* into a participatory sonic ritual. Rather than generating music solely from rule-based mappings, we integrate the divinatory semantics of the *I-Ching*: the Ben Gua (original hexagram), Zhi Gua (derived hexagram), and

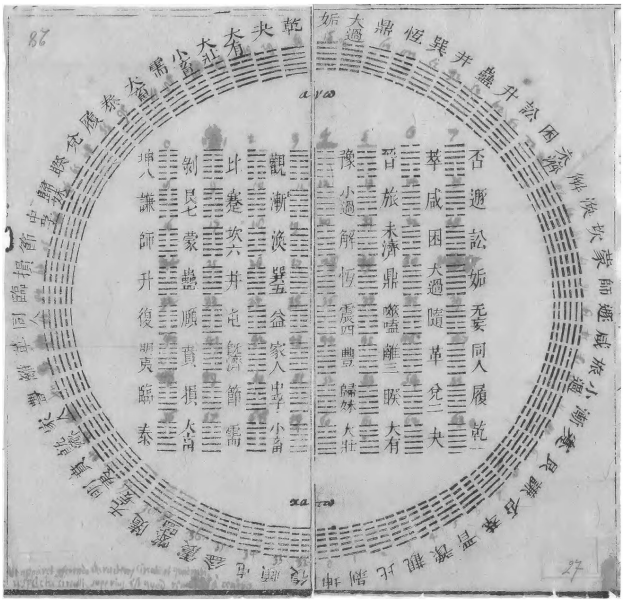


Figure 1: Hexagrams of the *I-Ching* that belongs to German mathematician philosopher Gottfried Wilhelm Leibniz [2].

Copyright: ©2026 Ling Qi et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License 3.0 Unported, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

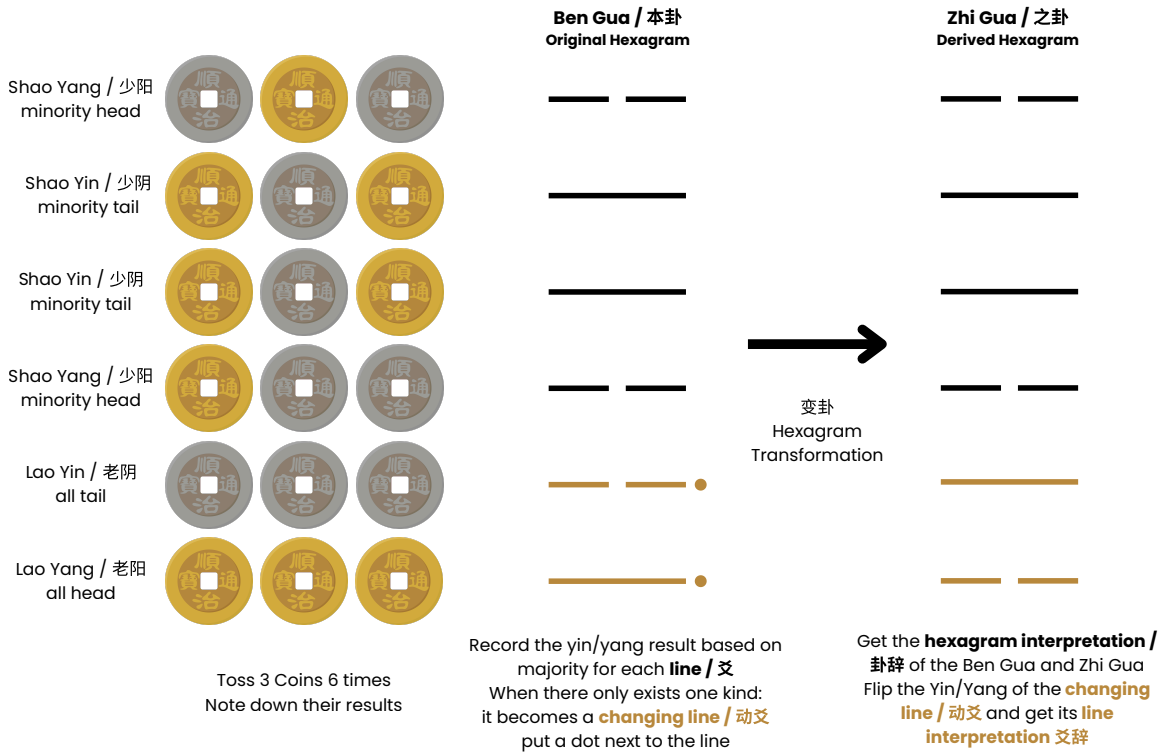


Figure 2: Overview of the Wen Wang Fa coin tossing approach in the *I-Ching*.

any Dong Yao (changing lines) guide the musical affect, structure, and evolution. To realize the personalized interpretation required by the *I-Ching* process, we draw on contemporary machine learning tools for text and music generation. A large language model (Gemini 2.5 Flash [7, 8]) interprets the hexagrams in relation to the user's question, synthesizing elements from the Gua Ci (hexagram interpretation) and Yao Ci (line interpretation). This interpretation is then translated into musical descriptors, such as mood, texture, pacing, instrumentation, which condition a real-time generative music model (Lyria [9]) to produce a responsive sonic representation of the divination outcome. The casting stage retains Cagean chance operations, while the interpretive stage reintroduces the symbolic and philosophical dimensions that Cage bracketed out.

Through this work, we explore how ancient divination practices can be reimagined as interactive music systems, and how computer music AI may function as an interpretive intermediary within a participatory process, instead of a generator of finished artifacts. Rather than approaching the *I-Ching* as a compositional mechanism in the Cagean sense, this work treats computer music as a medium to accompany the divination process. Here, process—rather than musical output—becomes the primary compositional object, and sound functions as a means of articulating uncertainty, interpretation, and becoming.

2. RELATED WORKS

2.1 John Cage and Chance Operation

John Cage's *Music of Changes* [4] stands as a foundational work in the history of chance-based composition. In this work, Cage constructed charts for musical parameters like tempo, duration, dynamics, density, and sound type. Each chart consists of 64 values, corresponding to the *I-Ching*'s 64 hexagrams. Cage used repeated coin tosses to form hexagrams and let chance select values from the charts at each structural point of the composition. His compositional goal was to eliminate personal taste and intention, allowing sound to emerge "free of individual memory and psychology" [6]. Before Cage, postwar avant-garde music was largely defined by serialist and modernist systems that sought total compositional control. Cage's chance operation marked an important turn within twentieth-century experimentalism, influencing Fluxus, Neo-Dada, and later developments in algorithmic and stochastic music [10, 11, 12].

Scholars have since debated the implications of Cage's technique [13, 14, 15]. Jensen [14] argues that despite Cage's genuine fascination with Eastern philosophy, he employed a Westernized, modernist reading of the *I-Ching* that emphasized randomness over its traditional semantic and divinatory meanings. Furthermore, Jensen suggests that Cage's procedures parallel the "chaos game," generating forms that are unpredictable yet still governed by underlying constraints, illustrating that Cage's rejection of intention does not necessarily yield pure indeterminacy.

2.2 The *I-Ching* and Its Westernization

To address the cultural and epistemological gap between Cage's procedural adoption and the *I-Ching*'s original function, it is important to situate the text within its historical foundations. The *I-Ching*, originally written during the Zhou dynasty (1000–750 BCE), serves both as a manual of divination and a philosophical text rooted in Daoist and

Confucian cosmology.

Traditional *I-Ching* divination systems such as Wen Wang Fa rely on coin tossing, a process we illustrated in Figure 2. It begins by casting three coins six times, in order to generate a hexagram composed of yin (broken) and yang (solid) lines, along with any changing lines (*Dong Yao*) that indicate transformation. The original hexagram is called *Ben Gua*, and the transformed hexagram derived from Dong Yao is called *Zhi Gua*. There are 64 possible hexagrams in total, each with a corresponding interpretation (*Gua Ci*). Ben Gua's Gua Ci provides an overarching judgment of the situation, while the *Yao Ci* of any changing lines offers more fine-grained guidance about movement, transition, and the unfolding of events. Interpreters then compare the *Ben Gua* with the *Zhi Gua* (the transformed hexagram derived from the changing lines) to discern their position within a shifting cosmological pattern.

The translation of the *I-Ching* (Princeton University Press, 3rd ed.) by German sinologist Richard Wilhelm, rendered into English by American Jungian psychologist Cary F. Baynes, is the most influential Western version of the ancient Chinese classic [1]. However, this edition reframes the text through a Western intellectual lens, emphasizing the *I-Ching* as a source of personal introspection rather than a ritual system [16]. This is the version that John Cage read, and it likely influenced his understanding of the *I-Ching* as a method of pure chance, stripped of its original symbolism and semantic interpretations.

2.3 Other Musical Transcodings of the *I-Ching*

Apart from John Cage, numerous composers have devised techniques that translate the structural or symbolic features of the *I-Ching* into musical parameters [17, 18, 19]. Chou Wen-Chung's "variable modes" partition the twelve-tone scale into three conjuncts—Heaven, Humans, and Earth—deriving melodic intervals from the cosmic significance of the hexagrams [19]. Zhao Xiaosheng's Tai Chi framework draws on the yin–yang dynamics embedded in the hexagrams to generate pitch materials, producing cyclical musical motion analogous to hexagram transformation processes [19]. Chung Yiu-Kwong's I-Ching Compositional System [17] connects the commentaries (Gua Ci) and individual line statements to pitch organization and the temporal statements trajectory of musical sections. Jeremy Muller's research [18] explores real-time chance procedures in digital contexts, demonstrating how contemporary technologies can replicate Cagean indeterminacy. Collectively, these approaches reveal the wide range of musical possibilities inspired by the *I-Ching*, ranging from symbolic mappings to technologically driven, interactive systems.

3. SYSTEM DESIGN

Music of Changing Lines [1] is implemented as a web application that aims to create an immersive, audio-visual experience of the *I-Ching* divination. Users are guided through three primary stages: intake, casting, and interpretation. The user interface and flow chart of our work are shown in Figure 3.

3.1 Intake Stage

After being introduced to the system and the relevant context of the *I-Ching* process, users are invited to submit a question for consultation, which may range from personal dilemmas to everyday choices. While classical *I-Ching* readings often draw on detailed personal information, our system only requests an optional name, preserving privacy while maintaining a sense of personalization.

3.2 Casting Stage

The Casting Stage operationalizes the chance component of the *I-Ching* divination. Users perform six coin tosses, each consisting of three coins, to form a hexagram. To enhance engagement and personalize the experience, we animate the coin tosses and require the user to initiate each toss via an interactive button. The result is unique to each user and the moment of interaction, embodying the traditional notion that chance is intertwined with both the individual and the universe.

During casting, semantic interpretations are deliberately withheld, following the *I-Ching* principles that forbid premature reading of individual lines. Because this stage carries no interpretive meaning, we use a rule-based, probability driven method, similar to John Cage's chance operation. We sampled taiko drum, koto, shakuhachi, shamisen, flute, and nylon-string guitar (as an approximation of the pipa) from a soundfont library [2], with each coin row mapped to one of these instruments. The instruments were chosen from the library to provide the closest available match to traditional Chinese sounds. After a toss, pitch and duration sequences are generated probabilistically for the corresponding instrument, then looped and layered with previous instruments to produce a cohesive, multi-voiced soundscape. The probability is informed by the coin outcome: for example, a three-head toss (old yang) favors ascending, longer notes, whereas three tails (old yin) favors descending, shorter notes. This allows the music to reflect the dynamics of the universe at the moment of casting, without conveying semantic meaning. All melodies are confined to a pentatonic scale to maintain harmonic coherence and evoke an aesthetic commonly associated with East Asian musical traditions, consistent with the *I-Ching* context. Unlike Cage, who relied on pre-defined charts, we use probability distributions to give more agency to chance while keeping aesthetic constraints that make the music immersive rather than distracting.

3.3 Interpretation Stage

The Interpretation Stage is the main enhancement of our system over Cage's approach, integrating the *I-Ching*'s semantic interpretations. From the coin casting, an original hexagram, any changing lines (Dong Yao), and the resulting transformed hexagram are formed. We retrieve the corresponding context-free interpretations for each hexagram and line. These data, together with the user's inquiry, are sent to *Gemini 2.5 Flash* [7, 8], a Large Language

[1] The source code of our application is available at `https://github.com/LingQi000809/iching-sonification`. A video walk-through of our system can be viewed at `https://vimeo.com/1150693113?share=copy&fl=sv&fe=ci`.

[2] https://github.com/gleitz/midi-js-soundfonts

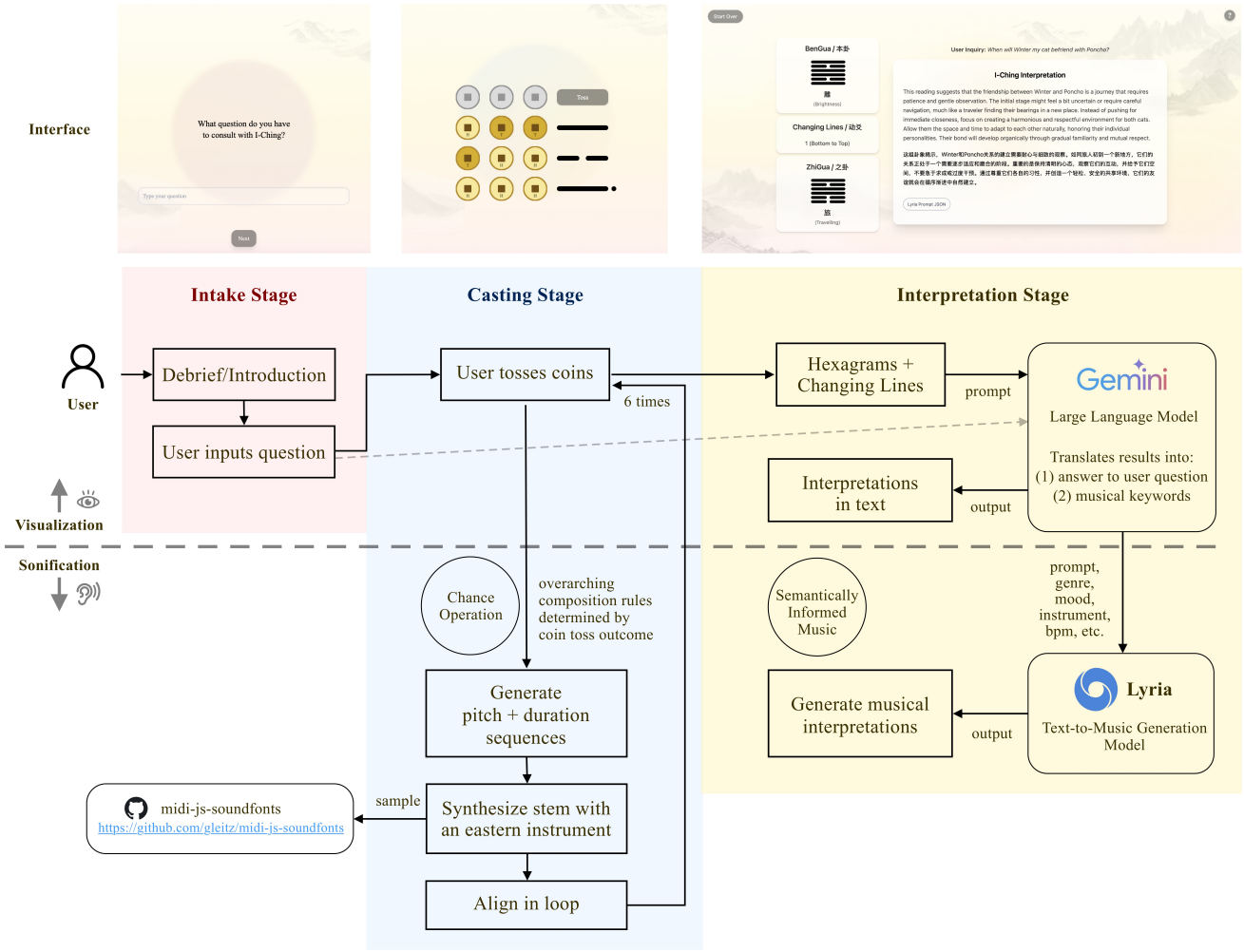


Figure 3: User Interface and Flow Chart for the system, *Music of Changing Lines*

Model (LLM), via Google GenAI's API. The inputs are prompt-engineered so that Gemini generates a reading tailored to the user's question, as well as keywords capturing mood, energy, dynamics, and spatial qualities. The keywords are then passed to Lyria [9], a text-to-music model in the Google GenAI suite. Lyria generates 30–60 second ambient pieces reflecting the hexagram transformations in the context of the user inquiry. A "Lyria Prompt JSON" button allows users to view the exact prompt used for music generation, providing transparency and reproducibility.

Alongside the generated music, the interface displays the hexagrams, changing lines, and textual interpretations as reference, grounding the auditory experience in the canonical *I-Ching* structure.

3.4 Sound Engine

The system's sound engine is implemented using the Web Audio API, with Tone.js providing abstractions for timing, routing, and synthesis. Sound generation operates in two stages: a real-time casting phase and a generative playback phase. During casting, each coin toss activates a loop-based layer corresponding to a hexagram line, rendered with sampler instruments derived from a soundfont library and routed through per-line panning into a shared compressor and reverb. The casting-stage music continues looping while the system generates the divinatory interpretation and music prompt, maintaining sonic continuity during computation. Once the interpretation is complete, the system transitions to a generative playback phase, where a language-model–derived music plan conditions a real-time text-to-music session using Google's Lyria. At any point, a user-initiated reset can stop all active audio processes, disposes of looping layers, and returns the system to a silent initial state.

3.5 User Interface

The interface, as shown on the top of Figure 3, is designed to prioritize minimalism and immersion, echoing the meditative and contemplative nature of the *I-Ching* practice. Users can backtrack, restart, or view instructions at any point through persistent navigation controls and tooltips. Each page features a breathing oracle circle in the background that slowly shifts in color and rhythmically expands and contracts. The design draws inspiration from the character *Samantha* in the film *Her* and Apple's voice assistant *Siri*, imitating an entity that listens and responds. To further reflect the cultural context of the *I-Ching*, subtle overlays of Chinese ink painting textures are incorporated into the background.

Interactive elements are woven into the interface to en-

gage users, including a chatbot-like inquiry system and user-initiated coin tossing. These interactions aim to make the experience personal and meaningful: the user's actions directly influence the result, emphasizing the concept that chance is co-created by the individual and the universe.

3.6 Process-driven Participatory Music Creation with AI

Our system draws on a lineage in computer music that treats musical works as interactive processes rather than fixed scores. In their discussion of *Computers and Future Music* [20], Max Mathews et. al. describe a future in which listeners actively participate in shaping musical outcomes through interaction with computational systems. Our work can be understood as a contemporary realization of this model, implemented with generative AI algorithms.

Rather than influencing musical detail directly, participation in our system occurs through a ritualized question-and-casting interaction. Users pose a question and perform a coin casting according to the *I-Ching*; the resulting hexagram initiates an interpretive process rather than prescribing musical structure. This aligns with Mathews's conductor concept, exemplified by the GROOVE [20], in which humans guide a computational process without specifying individual musical events.

In the interpretation stage of our system, a large language model generates a textual interpretation of the casting outcome and the user's question. In this way, AI serves as an intermediate interpretive step: it generates a textual interpretation of the casting outcome, which is then used to condition the music generation process. The generative music model responds to this prompt to produce a musical realization.

The contribution of this work lies not in new synthesis or generation techniques, but in demonstrating how contemporary generative AI can be integrated into computer music systems as an interpretive interface rather than a compositional authority. By situating AI within a culturally grounded, participatory process, the system offers an alternative model for human–AI music interaction—one that emphasizes meaning-making and engagement over optimization or automation. This perspective extends early visions of participatory computer music into the context of modern AI-driven tools, and suggests new roles for language models in interactive music systems beyond direct control or replacement of human creativity.

4. DISCUSSIONS

In developing this system, we encountered several intertwined aesthetic, technical, and cultural challenges. First, given the *I-Ching*'s complexity—its many divinatory methods, centuries of philosophical commentary, and historical reinterpretations—no single work can capture it in its entirety. Because the *I-Ching* continues to be widely used as a divinatory practice, we focus specifically on the popular approach of Wen Wang Fa as an interactive process. The availability of large language models made it possible to generate interpretations that are specific to each user's consultation, and therefore supports a more unique, dialogic, and interactive experience.

Second, we spent considerable time brainstorming how to implement chance operations musically. Our goal was to preserve the unpredictability associated with Cage's practice while imposing constraints that maintain perceptible structure and continuity. The loop-based design, in which each coin row generates a probabilistic melodic layer, allowed us to keep the local musical rules simple yet produce a perceptible sense of accumulation. As more instruments enter, the sonic texture emerges in parallel with the hexagram, reinforcing the metaphor of lines gradually composing a pattern of meaning.

Finally, model controllability remains an open problem. Although Lyria is designed for conditioned, real-time music generation, we found that its responses were sometimes insensitive to nuanced changes in the textual prompts and did not always align with the intended mood or structural trajectory. This raises broader questions about the limits of current text-to-music systems in supporting fine-grained, semantically grounded control. Addressing these issues is crucial for deepening the connection between divinatory interpretation and sonic outcome.

Future work could extend this system toward community-driven forms that are more closely connected to users' experiences. One direction is to aggregate anonymized user consultations and use recurring or divergent questions, along with their *I-Ching* interpretations, as material for a generative artwork that sonifies collective patterns of inquiry. Another direction would introduce deeper user interaction by allowing participants to reflect on the moods elicited by an interpretation, with these affective responses informing adaptive music generation strategies. Together, these approaches would deepen the dialogic relationship between users, the *I-Ching*, and the resulting music.

From a computer music perspective, this work shows how AI can be integrated into process-driven systems not as an autonomous composer or generator of finished artifacts, but as an interpretive intermediary within a participatory musical process. By foregrounding ritual, interpretation, and interaction, the system extends historical visions of participatory computer music into contemporary discussions on AI and meaning-driven musical systems.

5. CONCLUSION

The *I-Ching* is foundational to Chinese philosophy and cosmology, and its divinatory practices continue to shape how people think about change, uncertainty, and decision-making. While Western experimental musicians such as John Cage have drawn inspiration from the text, their focus on chance has often bracketed out its semantic, ritual, and cultural dimensions. In *Music of Changing Lines*, we attempted to re-balance that relationship by designing an immersive system that guides users through each stage of the Wen Wang Fa divination process and incorporates not only randomness but also meaning.

Our approach situates chance operations in the casting stage, where probabilistic musical processes mirror the unfolding of the hexagram, and then reintroduces semantic depth in the interpretation stage through LLM-driven textual readings and AI-generated music. In doing so, we move beyond treating the *I-Ching* as a generic randomization tool and instead foreground it as a culturally situated,

meaning-bearing framework. By embedding algorithmic and generative music within a participatory process, this work opens avenues for designing computer music systems that prioritize interaction, personalization, and the unfolding of musical experience over fixed artifacts.

6. REFERENCES

[1] Anonymous, *The I Ching or Book of Changes*, R. Wilhelm and C. F. Baynes, Eds. Princeton, NJ: Princeton University Press, 1967, translated by Richard Wilhelm; rendered into English by Cary F. Baynes. Original Chinese text compiled ca. 1000–750 BCE.

[2] F. Perkins, *Leibniz and China: A Commerce of Light*. Cambridge University Press, 2004.

[3] R. Kostelanetz, *Conversing with Cage*, ser. Readers' Guides to Essential Criticism. Routledge, 2003.

[4] J. Cage, "Music of Changes," 1951.

[5] ——, "Music for Piano 21–36, 37–52," 1960.

[6] C. John, *Silence. Lectures and Writings*. Middletown, 1961.

[7] G. Team, R. Anil, S. Borgeaud, J.-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. M. Dai, A. Hauth, K. Millican *et al.*, "Gemini: a family of highly capable multimodal models," *arXiv preprint arXiv:2312.11805*, 2023.

[8] G. Comanici, E. Bieber, M. Schaekermann, I. Pasupat, N. Sachdeva, I. Dhillon, M. Blistein, O. Ram, D. Zhang, E. Rosen *et al.*, "Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities," *arXiv preprint arXiv:2507.06261*, 2025.

[9] L. Team, A. Caillon, B. McWilliams, C. Tarakajian, I. Simon, I. Manco, J. Engel, N. Constant, Y. Li, T. I. Denk *et al.*, "Live music models," *arXiv preprint arXiv:2508.04651*, 2025.

[10] G. Brecht and L. M. Young, "An anthology of chance operations..." *(No Title)*, 1963.

[11] L. Sitsky, *Music of the Twentieth-Century Avant-Garde*. Greenwood Publishing Group, 2002.

[12] B. W. Joseph, "Experimentations: John Cage in Music," 2016.

[13] D. W. Patterson, *Appraising the catchwords, c. 1942-1959: John Cage's Asian-derived rhetoric and the historical reference of Black Mountain College*. Columbia University, 1996.

[14] M. G. Jensen, "John Cage, Chance Operations, and the Chaos Game: Cage and the" I Ching"," *The Musical Times*, vol. 150, no. 1907, pp. 97–102, 2009.

[15] D. W. Bernstein and C. Hatch, *Writings through John Cage's music, poetry, and art*. University of Chicago Press, 2010.

[16] R. J. Smith, "How the Book of Changes arrived in the West," *New England Review*, vol. 33, no. 1, pp. 25–41, 2012.

[17] Y.-K. Chung, *I ching compositional system: the symbolism, structures, and orderly sequence of the sixty-four hexagrams as compositional determinants*. City University of New York, 1995.

[18] J. Muller, "Real-Time Chance Operations: A Technologically Modern Approach to John Cage's 27'10.554" for a Percussionist."," *Percussive Notes Online Research Edition*, vol. 1, pp. 24–31, 2016.

[19] K. Xue and F. Y. Loo, "Transcoding the I Ching as Composition Techniques in Chou Wen Chung, Zhao Xiaosheng and Chung Yiu Kwong," *Musica Hodie*, vol. 19, 2019.

[20] M. V. Mathews, F. R. Moore, and J. C. Risset, "Computers and Future Music," *Science*, vol. 183, no. 4122, pp. 263–268, 1974. [Online]. Available: https://www.science.org/doi/abs/10.1126/science.183.4122.263

Reimagining Athletic Gesture: Transforming Basketball Sound into Narrative Electroacoustic Music

Guanjun Qin
University of Bristol
champion.qin@bristol.ac.uk

Yunxuan Jia
University College London
yunxuan.jia.22@ucl.ac.uk

Neal Farwell
University of Bristol
neal.farwell@bristol.ac.uk

ABSTRACT

This paper presents FMVP, an electroacoustic fixed-media composition that transforms the sounds of basketball into a narrative of doubt, struggle, and redemption. Built entirely from field recordings captured on an indoor court, the work reimagines sport as a metaphor for resilience and creative endurance. Through granular time-stretching, spectral transformation, dynamic filtering, and spatial motion, physical gestures such as dribbling, sliding, and impact are translated into evolving sonic textures and large-scale form. Inspired by the career arc of NBA player Stephen Curry, the composition explores how kinetic rhythms can be reshaped into emotional trajectories, aligning with the conference theme of Innovation, Translation, Participation. Methodologically, the project sits within artistic research, using composition as a mode of inquiry into the relationship between embodied action and sound narrative. The paper discusses the conceptual framework, sound-design process, and structural strategies underpinning FMVP, arguing that everyday athletic environments offer rich material for electroacoustic storytelling and for rethinking how listeners participate in narratives constructed purely through sound.

1. INTRODUCTION

Electroacoustic music has long served as a space for exploring the relationships between sound, movement, and meaning. Since the early practices of musique concrète, composers have investigated the expressive potential of everyday sonic materials and their capacity to convey narrative, emotional, or conceptual significance [1]. Although many works draw on environmental sound, technology, or abstract gesture, the sonic world of sport, which is defined by physical intensity, rhythmic precision, and embodied performance, has remained relatively underexplored within electroacoustic composition and research. Yet sport contains a rich palette of acoustic detail and naturally performative structures that offer substantial potential for sonic reinterpretation.

FMVP (Finals Most Valuable Player) engages directly

Copyright: © 2026 Qin et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License 3.0 Unported, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

with this underdeveloped terrain by transforming the sounds of basketball into a metaphorical narrative of doubt, struggle, and redemption. Rather than documenting a literal game, the work reimagines the physical energy of play as an emotional arc inspired by the career trajectory of NBA player Stephen Curry. His journey from underestimation to achievement illustrates a broader story of perseverance, and this trajectory becomes a structural and conceptual foundation for the composition. In this way, athletic motion, including dribbling, sliding, impact, and acceleration, serves as a generative basis for musical form and expressive development.

The piece also aligns closely with the ICMC 2026 theme of Innovation, Translation, Participation. It introduces an innovative use of sports based field recordings as primary compositional material, translates kinetic gestures into spectral and spatial transformations, and invites listeners to participate in an unfolding sonic drama shaped by physical energy. This exploration raises questions about how embodied actions can inform sound narrative, and how culturally recognisable environments, such as a basketball court, can be reframed as sites for artistic investigation [2].

From a research perspective, FMVP contributes to the field of composition as research by positioning creative practice as a methodological tool for generating insight. The work does not simply use sound to represent movement; rather, it examines how movement can be structurally and metaphorically transformed through electroacoustic processes [3]. By analysing the conceptual foundations, technical procedures, and aesthetic strategies behind FMVP, this paper aims to illustrate how artistic research can bridge physical sensation and sonic imagination, and how gesture driven environments can be reinterpreted in ways that extend the vocabulary of contemporary computer music.

2. CONCEPTUAL FRAMEWORK

The conceptual foundation of FMVP is shaped by two core principles: the use of sound as metaphor and the reinterpretation of physical motion as musical expression. In electroacoustic music, sound-based composition has long explored how everyday sonic materials can carry expressive and symbolic meaning beyond their original contexts [1]. These ideas allow the sonic environment of basketball to function not as a literal documentation of athletic activity but as the basis for an expressive

narrative that moves between preparation, introspection, transition, and competition.

The first principle, sound as metaphor, reframes familiar actions on the basketball court as markers of psychological or emotional states. Dribbling, footwork friction, and impact sounds are treated as materials that can suggest anticipation, focus, resistance, or breakthrough depending on how they are transformed spectrally or spatially. Through this metaphorical lens, the composition uses athletic gestures to evoke internal processes rather than external scenes. Such interpretations resonate with spectromorphological perspectives in electroacoustic music that consider the relationship between gesture, perception, and the evolving shapes of sound [2].

The second principle concerns the translation of physical motion into musical form. Athletic actions carry inherent characteristics of repetition, momentum, tension, and release. These qualities provide structural models for shaping energy, density, and pacing within the piece. As the composition evolves, different phases of practice, reflection, and competitive intensity are interpreted through shifting textures and dynamic development, allowing movement to become a source of musical logic. This approach is consistent with broader discussions of sound-based musical aesthetics, where environmental sounds may be transformed into expressive musical structures through creative listening and compositional design [3].

Rather than presenting each stage as an isolated section, FMVP weaves these conceptual states into a continuous trajectory that moves from disciplined preparation to introspective analysis, from transformation to heightened performance energy. The work does not employ motion capture or quantitative movement analysis; instead, gesture is interpreted perceptually through the sonic traces of physical action recorded in the basketball environment. This conceptual pathway provides a narrative foundation that is later articulated through detailed sound processing and compositional techniques, which are described in Section 4.

3. METHODOLOGY AND MATERIALS

The sound materials for FMVP were developed primarily from an extensive set of field recordings captured inside an indoor basketball gym, supplemented by a small number of synthesizer generated tones. Recording assistance during these sessions was provided by Yunxuan Jia. From the larger body of recorded material, specific sounds were selected based on their clarity, spectral richness, and gestural distinctiveness, allowing them to be effectively transformed and integrated into the compositional structure. These sources form the central sonic foundation of the composition. In addition to these core materials, a brief fragment of sports commentary associated with a well known basketball moment was incorporated only at the end of the piece to heighten the dramatic intensity of the final section. This vocal fragment functions purely as a narrative device and is not part of the main corpus of processed sonic material.

Field recordings were carried out using a ZOOM ambisonic 360-degree microphone, which enabled the capture of a full spatial image of the gym environment, alongside a Rode NT1 microphone connected through a Focusrite fourth-generation audio interface. Approximately three hours of recording were completed inside the gym, whose reflective surfaces produced strong natural reverberation. This acoustic characteristic contributed to the spatial colouring and sense of physical depth present in many of the raw samples. A wide range of microphone positions and movements was explored to obtain varied transient responses, directional cues, and spectral profiles. Microphone heights of approximately 50 cm, 100 cm, and 150 cm were tested, and in several cases the microphone was handheld and moved dynamically with the athlete, including during jumps, to capture a heightened sense of embodied motion. Additional recordings from frontal, lateral, rear, and elevated positions provided a diverse set of spatial perspectives.

The recorded material included dribbling, foot friction, impacts on the backboard and rim, footsteps, breathing, and general room tone. From roughly one hundred minutes of usable content, approximately ten minutes of material were selected based on clarity, dynamic stability, and spectral richness. Selected samples were manually trimmed, denoised when necessary, and categorised according to their functional characteristics.

Editing and transformation were undertaken primarily in Cubase 14 Pro, with additional processing carried out using patch based environments in Pure Data and Max, including buloTOOLS by Neal Farwell, University of Bristol (not publicly released), alongside external plug-in suites including FabFilter, Ozone, Waves, and Soundtoys. A wide range of signal-processing techniques was employed, including granular time stretching, spectral morphing, dynamic filtering, convolution, pitch shifting, transient enhancement, saturation, envelope shaping, and automation-based spatialisation. These procedures allowed the physical qualities of the recorded athletic gestures to be reshaped into musically expressive textures suitable for narrative development.

A significant methodological component involved transforming basketball impacts into percussive elements used in the transitional Section C of the composition. Kick-like sounds were produced by isolating low-frequency dribble samples and emphasising the $90-100$ Hz region through EQ and transient shaping. Snare-like materials were derived from mid- and high-frequency impact samples, while hihat-like textures were created from recordings of impacts against metallic structures

such as railings, with emphasis placed on preserving high-frequency content. This process enabled the work to shift fluidly from environmental realism to beat-driven rhythmic construction.

The completed composition was produced as a stereo fixed-media work at 48 kHz and 24-bit resolution. Mixing relied heavily on automation to shape spatial motion, intensity contours, and timbral evolution. Together, the recording, selection, and processing methods form the technical foundation through which the conceptual and narrative aims of FMVP are realised.

4. SOUND DESIGN AND STRUCTURE

The composition FMVP unfolds through a six-part structure that integrates field recorded basketball sounds, synthetic elements, and narrative driven electroacoustic transformation. Rather than presenting environmental recordings as documentary material, the work emphasises the transformation of recorded gestures into structured musical elements. The design of each section reflects a distinct stage of physical and psychological progression, supported by detailed manipulation of rhythm, spectral density, spatial behaviour, and gesture based development. Although the sections are conceptually differentiated, the work maintains a continuously evolving flow in which material from earlier stages is transformed, recontextualised, or expanded, producing a sense of reflective return and forward momentum.

The opening section, A1, establishes the sonic environment of practice. Raw dribbling gestures, foot impacts, and backboard strikes form the foundation of this segment. These materials are subtly processed using granular techniques implemented in Pure Data, producing fine-textured variations that evoke focus and controlled repetition without obscuring the recognisability of the source. The intention in A1 is to foreground the disciplined, physical nature of training while retaining enough of the original sound signature for listeners to understand the athletic context. The atmosphere is intentionally restrained, characterised by moderate dynamics, limited spectral expansion, and a stable rhythmic base.

B1 introduces the first reflective phase. Material from A1 is altered through time-stretching, filtering, and spectral blurring, generating expanded textures that suggest introspection and memory. Long-form layers derived from Paulstretch processing create elongated harmonic surfaces, while filtering shapes these surfaces into gradually rising trajectories that function as textural risers. Reverb and spectral diffusion expand the spatial field, shifting the focus from external action to internal contemplation. The sonic language becomes softer, more fluid, and less grounded in physical gesture, reflecting the mental processes of reviewing previous attempts, recalling competitive scenarios, and imagining tactical possibilities.

A2 returns to the domain of physical practice, but with markedly increased intensity. The rhythmic gestures are more stabilised, the transients sharper, and the spectral content richer. Additional layers of processed dribble impacts and transient-enhanced footwork sounds reinforce the sense of accumulated skill and heightened control. Elements introduced in A1 are developed further, often through denser layering or expanded spectral shaping. This stage embodies the energy of improved technique and focused determination, functioning as a strengthened counterpart to the initial practice segment.

B2 forms a deeper and more expansive reflective phase. Compared with B1, this section exhibits greater spatial breadth, more pronounced spectral stretching, and thicker layers of elongated textures. The sound world becomes increasingly cinematic, with time-stretched resonances and diffuse harmonics creating a broad and immersive auditory field. B2 prepares the transition into Section C by gradually loosening the connection to the concrete sonic environment, allowing abstracted gestures and spectral residues to emerge from the reflective haze.

Section C marks a pivotal transformation in which environmental sound is reinterpreted as percussive material. Here the compositional goal is to reveal how previously familiar sonic elements can shift into new functional roles, signalling a transition toward heightened momentum. Low-frequency dribble samples are shaped into kick-like impacts through EQ emphasis in the 90–100 Hz region and transient enhancement. Mid- and high-frequency hits become snare-like accents, while metallic railing impacts supply material for hihat-like textures. These percussive elements emerge gradually rather than abruptly, building on traces and colours foreshadowed in earlier sections. Conceptually, this moment represents the shift from internal reflection to decisive action: the sonic environment begins to assemble itself into a rhythmically driven structure, and the listener perceives the consolidation of energy and intention.

The final section, D, presents the highest level of intensity and marks the psychological climax of the work. Rhythmic elements introduced in C evolve into a more assertive beat-driven texture, supported by expanded synthetic layers and spatially wide harmonic content. Additional processed basketball sounds reinforce the sense of force and urgency, while the inclusion of a short fragment of sports commentary (used sparingly and symbolically) enhances the dramatic clarity of the narrative resolution. The spatial field is widened significantly through panning automation and brightened synthetic timbres, contributing to a sense of culmination and release. In this stage, earlier memories, doubts, critiques, and aspirations implied throughout the piece seem to converge, forming a complex internal landscape before returning the listener to the immediacy of action. The section concludes with the most dense and luminous

spectral profile in the work, symbolising a moment of decisive breakthrough.

Across the entire structure, FMVP progresses from controlled physical preparation (A1) to introspection (B1), strengthened practice (A2), deepened reflection (B2), structural transformation (C), and eventual culmination (D). The energy curve follows a trajectory of initial restraint, partial withdrawal, renewed construction, conceptual expansion, transitional acceleration, and ultimate intensity. This interplay of physical gesture, spectral evolution, and narrative pacing forms the core of the work's compositional logic and connects its technical processes with its broader artistic aims.

5. AESTHETIC DISCUSSION

The aesthetic identity of FMVP emerges from the interaction of physical gesture, psychological reflection, and sonic transformation. Rather than treating basketball as a literal subject, the work approaches athletic motion as an expressive resource for structural and emotional development. In this way, familiar sounds are reframed not only as acoustic artefacts but also as narrative agents within an evolving psychological trajectory.

A central strategy in FMVP is the shifting balance between recognisable source material and abstraction. Early sections preserve the identity of the basketball environment, grounding the listener in disciplined repetition and focused bodily practice. As the piece develops, these materials undergo spectral, temporal, and spatial transformation. This movement between concreteness and abstraction parallels the oscillation between physical action and internal reflection that characterises training and competition.

The work also explores how athletic intensity can be translated into electroacoustic dramaturgy. The opening practice gestures, the blurred textures of reflection, the strengthened force of renewed training, and the emergence of rhythmic drive together form a dynamic expressive curve. This curve echoes cycles familiar to athletes, including preparation, analysis, adjustment, and execution, and turns them into sonic form.

A further aesthetic shift occurs in Section C, where environmental sound is reorganised into percussion. Here familiar gestures acquire new functionality: actions associated with practice become instruments of momentum and force. The gradual formation of kick, snare, and hihat materials from basketball impacts marks a decisive turning point in the narrative.

The final section intensifies this trajectory through wider space, increased spectral brightness, and the addition of synthetic material. The brief use of commentary, treated symbolically rather than realistically, reinforces the dramatic resolution. Across the work, FMVP demonstrates how athletic environments can function as rich material for electroacoustic exploration, transforming embodied action into emotional and structural meaning.

6. CONCLUSIONS

FMVP explores how athletic gestures can be transformed into narrative driven electroacoustic form. Using field recorded material from a basketball environment and subjecting it to granular, spectral, spatial, and percussive transformation, the work shows how familiar physical actions can be reinterpreted as expressive musical processes. Its six part structure moves from controlled practice to introspection, renewed force, structural transformation, and heightened intensity.

Rather than documenting a sporting event, the composition reframes the sounds of practice as carriers of emotional and structural meaning. Through shifts between recognisable source bonded material and more abstracted sound processes, the work connects physical training, psychological reflection, and narrative momentum. Future developments may include multichannel expansion, sensor based gesture data, or interactive versions responsive to live movement. In this way, FMVP contributes to ongoing discussions of gesture, embodiment, and the transformation of non musical environments into musical experience.

7. MEDIA LINKS

https://drive.google.com/file/d/1pPAUk2Dch596-Qz5XRqLdg8oUTkQ2f6E/view?usp=share_link

8. ACKNOWLEDGEMENTS

The author gratefully acknowledges support from the China Scholarship Council (CSC).

9. REFERENCES

[1] T. Wishart, On Sonic Art. Amsterdam: Harwood Academic, 1996.

[2] D. Smalley, "Spectromorphology: Explaining sound-shapes," Organised Sound, vol. 2, no. 2, pp. 107–126, 1997.

[3] L. Landy, "On the aesthetics of sound-based music," Organised Sound, vol. 17, no. 1, pp. 11–22, 2012.

!
!

!"#$%!&'()*%#+,*-& %
%

!"#$%&'"(")%!
"#$%&!'())*+,!-*./)0,.12!3)456,1 !
p.rebelo@qub.ac.uk !

*$+,-&.+/01%2!&
"#$%&!'())*+,!-*./)0,.12!3)456,1 !
c.jackson@qub.ac.uk !

"#$%&'"% !

!"#$%&'()"#*"+")&#,"-"'$./")&0#1&#23456#7%"")80#9)(-"*:
0(&;#<"'=&#$"$>()?#(&0#=$*/1&($)#$)#&@"#$++10($)#$=#
(&0#AB&@#1))(-"*01*;#10#2345C#5")&*"#=$*#D)&"*,(0+(.'()1*;#
4"0"1*+@#()#2$%),#1),#E%6@(0#&*1)0(&($)#.*$/.&",#1#
":"-1'%1&($)#$=#"0"1*+@#0+$."#1'$)?0(,"#0(?)(=(+1)&#():
-"0&/")&#()#&"+@)$'$?("0#1),#=1+(G@(0#0&%,($#*".$*&#
0%//1*(0"0#&@"#*1&($)1'"#=$*#&@"0"#+@1)?"0#1),#.*"0")&0#
+%**")&# 0&$&&@1*&# =1+('(&("0# 0%..$*&()?# *"0"1*+@#()#
(//"*0(-"# 0$%),6# +$/.$0(&($)6# ."*=$*/1)+"6# (/.*$-(01:
&$)6# ()+'%0(-"#/%0(+#/1H()?6# ()0&*%/")&#1),# ()&"*1+&($)#
,"0(?)6# -(*&%1'# 1+$%0&(+# ()0&*%/")&06# "+$'$?(+1'# 0$%),6#
.1*&(+(.1&$*;#.*1+&(+"6#1),#/%0(+#0+@$#1*0@(.F

()!*+%&,-.'%*,+ !

78.,! ,1(9.:! 0);:01! 5:<(,),! :*! 0)<)*1! 9)/)4:;=)*1,! 61!
"#$%! 6*9! 0)54)<1,!:*! 18)! 9)/)4:;=)*1! 6*9!)/:4(1.:*! :5!
18)!<)*10)!:/)0!1>)*12!2)60,!:5!)?.,1)*<)!@!

/)!*+$%*%.%*,+"0!',+%12% !

"#$%!>6,!),16A4.,8)9!.*!BCCD!6,!18)!":*.<!#01,!$),)60<8!
%)*10)!61!'())*+,!-*./)0,.12!3)456,1&!4)9!A2!E0:5),,:0!F.G
<86)4!#4<:0*@!H(*9.*I!50:=!'())*+,!6*9!#146*1.<!E8.46*G
180:;.),!644:>)9!5:0!18)!<0)61.:*!:5!6!(*.J()!;82,.<64!6*9!
.*1)44)<1(64!0),:(0<)!5:<(,.*!I*!18)!.*1)0,)<1.:*!A)1>))*!
=(,.<&!1)<8*:4:I2&!,:(*9!6*9!,:<.)12@78)!.*6(I(061.:*!:5!
6! 9)9.<61)9! A(.49.*I! >6,! :55.<.6442! :;)*)9! A2! <:=;:,)0!
K6048).*L!"1:<M86(,)*!:*! 18)! :<<6,.:*! :5! N:*:0602! O)G
I0))! 6>609)9! A2! '())*+,@!78)! .*.1.64! 9)/)4:;=)*1! :5!
"#$%!.,!9) ,<0.A)9!.*!6!BCCP!,1(9.:!0);:01!8.*<)!18)*&!
"#$%! 86,! A(.41! 6! >:049>.9)! 0);(161.:*! 5:0! 0),)60<8! .*!
,:(*9! 6*9! =(,.<@! 78)! J(64.12! 6*9!)?<)44)*<)! :5! 18.,! 0)G
,)60<8! 86,! A))*! 0)<:I*.,)9! A2! 18)! $),)60<8! T?<)44)*<)!
H06=)>:0M! 180:(I8! >:049! J(64.12! :(1;(1,! 6*9!)*/.0:*G
=)*1,@! U*! $TH! BCBR&! 18)! 6,)*1+;6*)4! 8.I84.I81)9!
;06<1.<)!0),)60<8!:(1;(1,!.*!=(,.<!61!'-3!6,!V>:0494)69G
.*IW&!6!,.I*.5.<6*1!;:01.:*!:5!>8.<8!V)=A:9.)9!.**:/61.:*&!
<0)61./)!,<:;)!6*9!0.I:(0!61!18)!8.I8),1!4)/)4@W!)09.,<.G
;4.*602!0),)60<8&!6!M)2!;601!:5!"#$%+,!<:*10.A(1.:*&!>6,!
64,:!8.I84.I81)9!6,!X6*!60)6!:5!;601.<(460!,10)*I18&!>.18!

!"#$%&'()*+ ,+ -.-/+ 01213"1)+ 435+ 6(&7+ &7+ 48+ 1778 4%)&:319
)%&2<)1;+<8;1%+)(1+)1%=7+"+)(1+ Creative Commons Attribution
License 3.0 Unported ?+@(&:(+#1%=&)7+<8%17)%&:)1;+<7
)&"8?+48;+%1#%";<:)&"8+&8+48$+=1;&<=?+#%"A&;1;+)(1+"%&'
7"<%:1+4%1+:%1!&)1;5

!

=6*2!:(1;(1,!Y(9I)9!1:!A)!.*,10(=)*164!.*!9)/)4:;.*I!*)>!
,6069.I=,!6*9!:5!;0:5:(*9!.*54()*<)@W!
$),)60<8! 61! "#$%! 86,! A))*! 5(*9)9! A2! A:9.),! .*<4(9.*I!
18)!T(0:;)6*!-*.:*&!#01,!6*9!N(=6*.1.),!$),)60<8!%:(*G
<.4&!T<:*:=.<!6*9!":<.64!$),)60<8!%:(*<.4&!T*I.*))0.*I!
6*9!E82,.<64! "<.)*<),! $),)60<8! %:(*<.4&! Z4:A64! %864G
4)*I),!$),)60<8!H(*9&!18)!30.1.,8!%:(*<.4!0*!18)!%6*6G
9.6*!$),)60<8!%:(*<.4@!78.,!5(*9.*I!106<M!0)<:09!.*!>:0M!
1)==.*I!;0.=60.42!50:=!18)!601,!.,!060)!6*9!.*!.1,)45!)/.G
9)*<)!:5!)?<)44)*<)!.*!6!-K!6*9!.*1)0*61.:*64!<:*1)?!E@G
5),,:0!E)90:!$)A)4:!>6,!6;;:.*1)9 !O.0)<1:0*!BCBR@!78)!
<)*10)!.,!0(*!A26!,1))0.*I!<:==.11))!,)11.*I!;0.:0.1.),!6*9!
,1061)I2@! %(00)*1! =)=A)0,8.;! <:*,.,1,! :5!!6<69)=.<,!
50:=!6<0:,,!,.?!"<8::4,!61!'())*+,&!B\!E8O!,1(9)*1,!6*9!
B!/.,.1.*I!,<8:460!@

3)!4'"*0*%*1$!

345 677"$1,8"&9%:2#&

78)! ":*.<!]6A ! >6,! 9),.I*)9! 6,! 18)! <)*10)! ;.)<)! :5! 18)!
"#$%!A(.49.*I@!#0<8.1)<1,!N644!346<M!O:(I46,!<:*<)./)9!
:5! 18)! A(.49.*I! 6,! 6! <:0)! 180))! ,1:02!/:4(=)! 1:! 644:>! 5:0!
18)!8.I8),1!,1(9.:!J(64.12!,;61.64!6(9.:!)?;)0.)*<)!1:!6(9.G
)*<),!(;!1:!RBC!,)61)9!:0!,16*9.*I@!*54()*<)9!A2!18)!^),1!
Z)0=6*!E6/.44.:*!61!18)!R_`C!a,6M6!^:049!H6.0&!6!,;8)0.G
<64! 6(9.1:0.(=! >.18! 1)<8*:4:I2! 9)/)4:;)9! 61! 7-! 3)04.*!
6*9!".)=)*,!6*9!=:,1!<0.1.<6442!6*9!.*54()*1.6442&!4)9!A2!
"1:<M86(,)*QB@! #! M)2! 4)I6<2&! ,1.44!)?;)0)*<)! 18)!
":*.<!]6A!1:962!.,!18)!6<:(,1.<6442!106*,;60)*1!I0.9!54::0!
644:>.*I!5:0!0.*I,!:5!4:(9,;)6M)0,!;0:Y)<1.*I!50:=!(*9)0!
18)!6(9.)*<)@!

!

! "#$%&!"#$%"&'()*"+,-"

!

! "##! $%%&'())***+,-.+/0+-1)2#'#/20$#4%2#')'/20)&#5&65)2! 0-22#4%!
68'%

!

The Sonic Lab is installed with 84 loudspeakers across 4 levels. They are configured to enable three spatial systems: surround (48 loudspeakers surrounding the audience at four vertical levels (5th Order Ambisonics, Dolby Atmos, point-source diffusion), height adjustable frontal array (42 loudspeakers for Wavefield Synthesis) and the IKO [3] loudspeaker. The design of the three systems aims to explore the unique spatial affordances of each, including diffuse immersive image with surround ambisonics highly precise frontal localization and distance with Wavefield Synthesis and the object-like 'inside out' characteristic sound of the IKO.

HIGH LOUDSPEAKERS - 8 x Meyer UPM-1P
Dolby Atmos configuration, arranged in two rows front to back. The loudspeakers hang from six of the nine overhead adjustable ceiling panels.

MID HEIGHT LOUDSPEAKERS -15 x Meyer UPJ-1P and 3 x Meyer and UMS-1P subwoofers.
Dolby Atmos configuration. The loudspeakers are suspended from the technical gantry approximately 4m above the ground floor.

GROUND LOUDSPEAKERS - 4 x Genelec 1038B and 4 x Genelec 1037B positioned in an octophonic configuration.
1 x IKO 3D loudspeaker.

Wavefield synthesis array - 42 Genelec 4410. Positioned on a retractable truss at the front of the stage area.

BASEMENT LOUDSPEAKERS - 8 x Meyer UPM-1P
2 x Meyer X-800 subwoofer - at lower ground level (4m below the grid floor).

Mixing Console: A Solid State Logic S500 mixing console is located at the rear of the lab. The console is highly configurable for the various listening systems in the space. There are also multiple I/O rack locations around the lab to allow for live audio processing and recording.

The loudspeaker system is managed using a QSC QSYS system (1 x Core 510i) over a Dante network equipped with Netgear AVLine switches.

An Apple Mac Studio hosting a ProTools, ambisonics and wavefield synthesis system using Reaper and IRCAM's Panoramix. Connectivity is via a DAD Core 256 interface.

The adjoined Control Room is equipped with a Studer Vista mixing console and multicamera video mixing software to allow independent mixing for recording and live streaming.

The Sonic Lab also features a Yamaha Disklavier installed with the Magnetic Resonator Piano[4].

3.2 The Maker Space

A significant expansion of SARC's fabrication facilities supports research in music instrument and interaction and interface design in the context of Virtual Acoustic Instruments. A dedicated space with extraction for the various processes includes a CNC Machine for professional-level prototyping and manufacturing. This allows for precise manufacturing of 3-Dimensional objects out of metal. In combination with the 3-axis CNC machine, for work with harder materials. 9 x 3D printers (including 5 x Bambu Lab P1S 3D).1 x vinal cutter, 1 x laser cutter (bed size: 610mm x 305mm). Industrial CNC machine Mantech Spartan 1313 CNC Router (bed size 1200x1200mm). A desktop CNC Milling machine - Pocket NC V2-50. Soldering stations with PCB manufacturing. Wood working room with hand and power tools. The nearby Interaction Lab is a space where users can setup and test concept builds and installations.

3.3 The SARC Gallery

The SARC gallery, opening in 2026 supports immersive, audio-visual and physical interaction research and artistic practice by providing an intimate space with concealed equipment. The 4.8m × 4.8m gallery is equipped with 4-side video projection and surround sound (16 channel ceiling grid). The Gallery has a programme of curated immersive and interactive artworks throughout the year, also supporting residencies and collaboration with local galleries.

Equipment includes: Projection – 4 x Optoma ZK430ST UHD laser projector. Loudspeakers – 16 x 4430A Genelec, 2 x Genelec 7350APM subwoofers. Lighting – 4 x motorised Elation Chorus Line 16 wash fixtures. AV signals are network-based using Dante and NDI powered with Netgear AVLine, distributed and scheduled using QLab software.



Figure 2. The SARC Gallery

!

!

34C<="&9@'*&>%(,)"&

#!A),,::M)!5.)49!>:0M!/)8.<4)!5:0!,;)<.64.,)9!=(41.<86**)4!
6(9.: C&,(64!0)<:09.*I!6*9!A0:69<6,&!8.,!(*.J()!56<.4!2
<:=A.*) ,!;0:5),,.:*64!=(41.<86**)4!NO!6(9.:C&,(64!<6;G
1(0.*I!6*9!,10)6=.*I!)J(.;=)*1!6,!>)44!6,!6*!.==)0,./)!
;462A6<M!,2,1)=!644:>.*I!5:0!0),)60<8!1),1,&!9.,,)=.*61.:*!
6*9!;(A4.<!)*I6I)=)*1 @78)!"#$%!F:A.4)!>6,!9),.I*)9!
A2!E)90:!$)A)4:&!%06.I!g6<M,.:*!>.18!O.I.164!Z6078)!@!
9),.I*!;0:<),,! 6.=)9!1:!A0.*I!1:I)18)0!(,)6A.4.12!6<0:,,!6!
*(=A)0!:5!9.55)0)*1!,<)*60.:!@8)!/6*!.,!A6,)9!:*!6!=6.*!
,1(9.:!60)6!>.18!6!`@D@R!EF%!4:(9,;)6M)0!,2,1)=@!a*)!6(G
9.:!=.?.*I!;:,.1.:*!6*9!:*)!/.9):!;:,.1.:*!644:>!5:0!1>:!
:;)061:0,!1:!>:0M!1:I)18)0!,860.*I!18)!,:(*95.)49!6*9!6<G
<),,! 1:! 6! 460I)! ,<0))*!@*! :*)! 86*9&! 18)! "#$%! F:A.4)!
5(*<1.:*,!6,!6!;0:9(<1.:*!10(<M!:0!a3!/6*@!78.,!,<)*60.:!
644:>,! "#$%! 0),)60<8)0,! 1:! 0)<:09! ,:(*9! 6*9! =(,.<!
)/)*1,!>.18!8.I8!<86**)4!<:(*1!6<0:,,!6!/60.)12!:5!/)*(),@!
#!,)<:*9!,<)*60.:!,!5:<(,)9!:*!4:*I!9(061.:*!5.)49!0)<:09G
.*I!.*!0)=:1)!4:<61.:*,@!78.,!644:>,!5:0!5.)49!10.;,!:/)0!6!
*(=A)0!:5 !962,@!#!<:=A.*61.:*!:5!A(.41!.*!Uja!6*9!O6*1)!
0)<:09)0,!6,!>)44!6,!6!/60.)12!:5!,;61.64!6*9!,;)<.64.,1!=.G
<0:;8:*),!644:>,!5:0!=(41.<86**)4!4./)!,;61.64!=:*.1:0.*I!
6*9!,10)6=.*I@#*:18)0!,<)*60.:!.,!A6,)9!:*!,:<.6442!)*G
I6I)9!;06<1.<),!6*9!5:<(,),!:*!,.1) C;)<.5.<!.*1)0/)*1.:*,!
644:>.*I!5:0!0)<:09.*I!6*9!4.,1)*.*I!,),,.:*,!:5!(;!1:!5:(0!
.*9./.9(64,!61!6!1.=)@!78.,!6.=,!1:!A0.*I!.==)0,./)!6(9.:!
6*9! 6(9.: C&,(64! ,10)6=.*I! 1)<8*:4:I.) ,! 1:! <:==(*.1.),!
1861!:18)0>.,)!86/)!*:!6<<),,@!H.*6442&!18)!"#$%!F:A.4)!
<6*!6<1!6,!6!=:A.4)!,16I)!>.18!6*!:(19::0!E#!,2,1)=!64G
4:>.*I!5:0!;:;!(;!;)05:0=6*<),!.*!;(A4.<!,;6<)@! !

!

#-OUa d`@D@R!4A2! #1=:,! 4:(9,;)6M)0! <:*5.I! n! RR! ?!
EF%! %.\C! d>.18! E:>)0,:51! 6=;4.5.)0,&R! ?!Z)*)4)<!
`\[C#EF!,(A>::5)0@!! !

#lUO! F7$i! "1(9.: ! 6(9.:! .*1)056<)! >.18! O#0F: F!
=:*.1:0!<:*10:4!6*9!#;;4)!F6<!F.*.@!

B!?!'"]!-Hb! 569)0!<:*10:44)0,!

\!?!":(*9!O)/.<),!bbb!6*9!R!?!":(*9!O)/.<),! "<:0;.:! C&
O6*1)!0)<:09)0,!5:0!0)=:1)!0)<:09.*I@!!

#!; :016A4)!0)<:09.*I!0)>.18!#lUO!F7$i! O6*1)!.*1)0G
56<)!6*9!#44)*!p!N)641!8!<:*,:4)!>.18! O7R[b!Uja!A:?@!!
N:4246*9!.*1)0<:=!,2,1)=@!!

lUOTac!\!?!3.099:I!EDK!E7o!<6=)06!0(**.*I!:/)0!6!
hOU!*)1>:0M!;:>)0)9!A2!A.J(.12!,>.1<8),@!

l.,,:*!=.?.*I!;:>)0)9!(,.*I!lF.?!6*9! 6*!NE!oB!F.*. !
E%!@EC&.*M!PZ!0:(1)0!5:0!0)=:1)!4./)!,10)6=.*I@!!

!

78)!/6*!.,!)J(.;;)9!>.18!5:(0!$:6=)0!4).,(0)!A611)0.),&!
)*6A4.*I!)?1)*9)9! :5&0.9! 0)<:09.*I@! U1! 64,:! 5)61(0),! 6!

;:;; C&;!0::5!>.18!6!9:(A4)!A)9&!6,!>)44!6,!6!106.4)0!5:0!69G
9.1.:*64!,1:06I)!@

!

!"#$%&'*!"#$%"&./0"5'-)2%"

5)!"&%*$%*'!&1$1"&'6!"+-!'&1"7
%*81!9&'"%*'1 !

#1!18)!<:0)!:5!"#$%+,!6<1./.12!:06<1.<0)A6,)9!0),)60<8!
6<0:,,!=(41.;4)!60)6!.*!,:(*9!6*9!=(,.<&!:51)*!.*/:4/.*I!
.*1)09.,<.;4.*602!)4)=)*1,@!#0)6,!:5!,(16.*)9!6<1./.12!9)G
/)4:;!60:(*9!)?;)01.,)!6*9!0),)60<8!.*1)0),1,!:5!6<69)=.<!
,1655!6*9!E8O!,1(9)*1,@!":=)!8.I84.,!81<4(9)!:;)*!6;G
:0:6<8), !1:!<:=:;.,1.:*&!.*<4(9.*I!)4)<10:6<:(,1.<&!:;)*!
,<:0.*I!6*9!.*,10(=)*164!.*<4(9.*I!>:0M!>.18!:(0!0),.9)*1!
)*,)=A4)! fN609! $6.*! ":4:.,1! T*,)=A4)W$! 6*9!)?;)0.G
=)*164!6;:0:6<8),!1:!U0.,8!7069.1.:*64!F(,@!#!M)2!)4)G
=)*1!:5!"#$%+,!>:0M&!A:18!.*!1)0=,!:5!0),)60<8!=)18:9&!
.*1)09.,<.;4.*602!6*9!<0)61.:*!:5!601>:0M!.,!.=;0:/.,61!:*
:51)*!.*/:4/.*I!(*(,(64!:0!A),,:M)!.*,10(=)*1, !6*9!=(,.G
<64!9)/.<),!U*!18.,!<:*1)?1&10(=)*1!6*9!.*1)06<1.:*!9)G
,.I*!5))9,!.*1:!601.,1.<!:(1;(1,6,!9:),!0),)60<8!.*1:!/.01(64!
6<:(,1.<!.*,10(=)*1,&!.*!;601.<(460&!)?;4:0.*I! 18)!.*1)06<G
1.:*! A)1>))*! ;82,.<64! =:9)4.*I! 6*9! ;82,.<64! .*1)056<)!
d,))&! 5:0!)?6=;4)QRS@!*,164461.:*! >:0M! .*/:4/.*I! ,.1)G
,;)<.5.<j,.1)G0),,:*,./)! 6;;0:6<8),! 6*9! .,61.64! 5.)49! 0)G
<:09.*I!:5!)6990),,!)*/.0:*=)*164!.*,(),@!

:) !&1$1"&'6!'.0%.&1 !

"#$%+,!<(41(0)!.,!<8606<1)0.L)9!A2!54)?.A.4.12!.*!0),,:*9G
.*I! 1:! 0),)60<8! :;;:01(*.1.),! 6*9! .*.1.61./),! >8.<8! <6*!
06*I)! 50:=! .*9./.9(64! ;0:Y)<1! 1:! .*1)0*61.:*64! 0),)60<8!
I06*1,@!"#$%+,!E8O!<:8:01!86,!A))*!6!<:0)!,10)*I18!.*!:(0!
0),)60<8!)*/.0:*=)*1@! a(0! E8O! 46A6,0)! 6! 8(A! 5:0?G
<86*I)!:5!.9)6,!6*9!9.,<(,,.:*!4)69.*I!1:!<:446A:061.:*!A)G
1>))*! 601.,1,&0),)60<8)0,6*9! ,<8:460,! 6<0:,,! 9.55)0)*1!
5.)49,@!".*<)!BCB\!>)!86/)!06*!18)!"#$%!"2=;,.(=!9)G
.<61)9!1:!=)=A)0,!;0),)*1.*I!18).0!<(00)*1!>:0M!6*9!5:,G
1)0.*I!<:46A:061.:*@!a*!18)!:<<6,.:*!5!18)!6**./)0,602!.*!

!

$!$%%&'())***+$/292/84#4'#:.6#+0!5:)

%! $%%&'())***+,-.+/0+-1)2#'#/20!6#4%2#')'/20)2#'#/20$)8:&25;8!8-4<
%-2#!)=:&25;8'84<!>-%-2#'!7-49#9!.?!@/4/98/4!""AB@'!-8&&52%84!<!

<65./6!4#%*521!57!8:&25;8'/%854!2#'#/20!6#4%2#!<!84!C#67!%5!9#3
;#65&!%$#!650/6!8:&25;8'#9!/49!#D&#28:#4%/6!:-'8D!'0#4#

!

2024, an archive of 20 years of work has been compiled with over 300 entries across different media, capturing research outputs by academic staff and PhD students. The archive is itself a growing research resource allowing for an understanding of trends in research topics and representing an 'in-house' repository and shared institutional know how. The archive also served as the core resource for a celebratory artwork which took the form of an experimental immersive documentary [6] created by Pedro Rebelo and Frank Delaney entitled "SARC x 20", a 30 minute five-screen film with Higher Order Ambisonic audio[4].

5.1 Impact

SARC's research has always had impact outside academia through participatory and socially engaged projects. Impact takes place across communities through participatory artistic projects in sound and music. The work addresses diversity and inclusion and focuses on communities identifying as disabled[5] and socially disadvantaged communities. These communities are both participants and beneficiaries of projects rooted in research in sound and music. In the context of practice-based work the outcome of a participatory research process is the artwork itself as research output and often a result of participation and co-design.

5.2 Sonorities Festival Belfast

The oldest running festival in Ireland focusing on contemporary, experimental and exploratory music making, Sonorities had its first program in 1981. Since 2004, the Festival has been hosted by SARC while using venues across the city and in collaboration with other promoters. The programme is curated through a combination of open calls, invited acts and activities supported specifically by ongoing research projects and including a one day symposium. Currently directed by Professor Paul Stapleton, the festival runs every two years, with the 2026 edition having received 392 applications from 39 countries across 8 open calls. The 2026 edition runs from 15th-18th April with 11 invited acts, an ongoing streaming collaboration with the IF festival[6], 10 acts as part of Handmade Music, 9 spatial audio works in the sonic lab, 15 exhibitions and installations in the SARC Gallery and other spaces in the city, soundwalks, SARC Mobile suite specific interventions, 11 works as part of a collaboration with Climate Sound Action initiative and 17 symposium presentations.

6. CONCLUSIONS

This studio report summarizes a trajectory of twenty years since the establishment of SARC and the first ICMC studio report from the centre. Although the notion of a 'studio' in the sense of a shared space for sound and music creation is central to a lot of the work, SARC has become a more amorphous structure to allow for interdisciplinary research to emerge in response to challenges and opportunities.

Acknowledgments

We would like to thank the SARC Community, all staff and students who have contributed to this curious, provocative and exciting research culture. To all those who carry the flag of "SARCness" with us and make the world an even more interesting place to listen out for.

7. REFERENCES

[1] 'Sarc: Studio Report - researchr publication'. Accessed: Dec. 16, 2025. [Online]. Available: https://researchr.org/publication/AlcornCR05

[2] S. Williams, 'Osaka Expo '70: The promise and reality of a spherical sound stage', presented at the Insonic, Karlsruhe, Germany, Nov. 2015. Accessed: Dec. 16, 2025. [Online]. Available: http://insonic2015.org/

[3] F. Zotter and M. Frank, 'Compact Spherical Loudspeaker Arrays', in *Ambisonics: A Practical 3D Audio Theory for Recording, Studio Production, Sound Reinforcement, and Virtual Reality*, F. Zotter and M. Frank, Eds, Cham: Springer International Publishing, 2019, pp. 153–170. doi: 10.1007/978-3-030-17207-7_7.

[4] A. McPherson, 'The Magnetic Resonator Piano: Electronic Augmentation of an Acoustic Grand Piano', *Journal of New Music Research*, vol. 39, no. 3, pp. 189–202, Sept. 2010, doi: 10.1080/09298211003695587.

[5] E. O. Néill, M. Van Walstijn, and M. Ortiz, 'Real-time bowing parameter sensing methods for articulated bow-string physical modelling', in *Forum Acusticum/Euronoise 2025, the 11th EAA Annual European Conference on Acoustics and Noise Control Engineering*, European Acoustics Association, 2025. Accessed: Dec. 30, 2025. [Online]. Available: https://www.researchgate.net/profile/Maarten-Van-Walstijn/publication/392706769_Real-time_bowing_parameter_sensing_methods_for_articulated_bow-string_physical_modelling/links/6856f2f1e8fa0f5c282598a1/Real-time-bowing-parameter-sensing-methods-for-articulated-bow-string-physical-modelling.pdf

[6] P. Rebelo and F. Delaney, 'SARC x 20: Strategies in Audiovisual Spatial Composition', in *Music and the Moving Image Conference 2025*, New York, 2025.

[4] https://www.qub.ac.uk/research-centres/sarc/research/SARCx20/

[5] See for example https://www.qub.ac.uk/research-centres/sarc/research/bridgingthegap/

[6] https://improvfest.ca/home/

Supervised Memory: How Machines Can Preserve What We Cannot Hold

Giovanni Roma
Conservatorio Domenico Cimarosa Avellino
giovanni.roma@conservatoriocimarosa.org

Alba Francesca Battista
Conservatorio Domenico Cimarosa Avellino
alba.battista@conservatoriocimarosa.org

ABSTRACT

This paper presents an AI framework for preserving electroacoustic works threatened by technological obsolescence and vanishing performance knowledge. Through supervised annotation as "composition of comprehension," we transform machine learning into active interpretation rather than passive archiving. Our approach employs a two-level vocabulary system distinguishing universal from composer-specific notational elements, enabling systematic knowledge transfer across diverse repertoires. We ground the framework in one implemented reconstruction—Jonathan Harvey's Ricercare una melodia *from incomplete documentation—and outline two further experimental fronts: analyzing context-dependent notation in Stockhausen's* Solo*, and exploring annotation possibilities in Boulez's spatial coordinates. The methodology treats annotation not as neutral transcription but as interpretive translation, where each label embeds aesthetic decisions and performance practice. Harvey's implementation revealed how editorial simplification between 1984 and 2003 editions created cascading performance challenges, validating our recovery of embedded procedural knowledge. The framework progresses from mechanical reproduction through systematic reading to conscious reactivation, establishing foundations for computational preservation while acknowledging fundamental limits. We argue that effective preservation requires not static archives but living traditions maintained by transparent, contestable machine interpretations. This positions AI-based complements as participants in musical preservation rather than mere repositories, preserving both structural relationships and the reasoning patterns that animate them.*

Copyright: © 2025 Roma G. e Battista A. F. This is an open-access article distributed under the terms of the Creative Commons Attribution License 3.0 Unported, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

1. INTRODUCTION

The crisis facing electroacoustic music is not merely technical but existential. Works that defined an era of musical innovation now face extinction as the knowledge required for their realization vanishes with each departing practitioner. Consider Jonathan *Harvey's Ricercare una melodia (*1984) [1]: originally notated with meticulous detail including all four delay levels, routing diagrams, and fader controls for trumpet and tape recorders, yet by 2003 the viola edition [2] retained only the solo line. This editorial simplification exemplifies a broader pattern of epistemic erosion where the procedural intelligence embedded in scores gradually disappears, leaving performers to reconstruct complex electronic architectures from incomplete documentation. This progressive loss extends beyond individual works to threaten an entire musical tradition. The pioneers who collaborated directly with composers like Harvey, Stockhausen, and Boulez carry embodied knowledge that transcends written notation [3]. This practitioner-based knowledge loss has been widely identified as a central preservation problem in live electronic music, where documentation alone cannot substitute for embodied expertise. Understanding which technical choices served aesthetic purposes, recognizing when apparent inconsistencies were intentional features, and knowing how technological limitations became sites of creative invention. These practitioners function as living bridges between the composer's imagination and the work's realization. As they retire, we lose not just technical skills but the interpretive frameworks that make these scores comprehensible.

We propose addressing this crisis through a novel approach: developing AI-based tools for the conscious reactivation of performance practices, designed to integrate the learning process by structuring knowledge in ways comparable to an essay written by a practitioner with embodied expertise, while maintaining transparent, contestable reasoning. Our framework treats these tools not as neutral repositories but as instruments embedded within a living tradition, developed through what we term "supervised annotation as composition of comprehension." This methodology transforms the act of teaching machines into an act of preserving not just symbols but the reasoning patterns and aesthetic priorities

that animate them. [E] At the current stage of this research, we do not claim to have implemented the full system described here. Rather, this article outlines the methodological framework and architectural principles required for such a system, grounded in a detailed analytical reconstruction of Harvey's *Ricercare una melodia.* The practical implementation of the proposed AI-assisted tools therefore remains a direction for future research.

Our contribution emerges from two interconnected research streams. First, we present a detailed reconstruction of Harvey's *Ricercare* that recovers the procedural knowledge embedded in the original 1984 score, demonstrating how editorial simplification creates cascading performance challenges. Second, we develop a theoretical framework for AI-mediated preservation that distinguishes between universal and composer-specific notational elements, enabling systematic knowledge transfer across works and traditions. By combining practical implementation with theoretical rigor, we establish foundations for preserving electroacoustic heritage through computational means while acknowledging what computation cannot capture: the embodied experience, the creative courage, the spiritual aspirations that transcended the page.

2. CASE STUDY: HARVEY'S *RICERCARE UNA MELODIA*

2.1 The Epistemic Gap

Jonathan Harvey's *Ricercare una melodia* exemplifies the challenges outlined above. The work was originally composed in 1984 for trumpet and tape-delay system and published by Faber Music, but it was conceived as a flexible system adaptable to multiple instruments. The catalogue therefore includes contemporaneous versions for flute, oboe, saxophone, cello, and trombone as documented in the publisher's catalogue [4], as well as later editions for clarinet and viola published in 2003 [2].

The 1984 trumpet score presents a comprehensive notational system: five staves displaying the live instrument and four delay iterations, detailed signal routing diagrams, precise fader movements, and technical specifications for the two-recorder tape system. Each delayed voice appears on its own staff, allowing performers to visualize the emerging canonical structure and anticipate harmonic intersections.

The 2003 viola edition reflects a different technological context. Rather than notating the delay structure explicitly, the edition assumes the use of an automated digital patch that manages the delay processes externally. As a result, the score retains only the instrumental line, while the delay staves, routing diagrams, and fader indications disappear. When the original patch became obsolete by 2024, performers were left with a score that no longer contained sufficient information to reconstruct the underlying five-voice canon, forcing them to rediscover the delay architecture through experimental reconstruction.

2.2 Original Architecture and Digital Translation

The work's electronic system employs two four-track tape recorders in an elegant feedback configuration as shown in Figure 1. The first recorder captures the live input, creating a three-second buffer before playback begins on the second machine. The output from track one splits, feeding both the first speaker and track two of the initial recorder, establishing a cascading delay pattern. This spiral continues through all four tracks, generating delays at 3, 6, 9, and 12 seconds that combine with the live voice to create a five-part canon.

Section B introduces radical temporal transformation. Setting the playback speed to half transforms not only pitch—each iteration sounds an octave lower—but fundamentally alters the temporal architecture. The delays expand exponentially: 6000ms, 18000ms, 42000ms, and finally 90000ms. What began as tight canonical imitation becomes vast temporal architecture, with the final delay occurring ninety seconds after the original gesture. This progressive augmentation creates what Harvey envisioned as "fusion and fission"—unified timbres decomposing into constituent elements across time and register.

Our digital implementation [5] pursues a fundamentally semantic approach to score interpretation. Rather than optimizing algorithmic efficiency, the reconstruction serves as a hermeneutic method—extracting and understanding the notational logic embedded in Harvey's score. We deliberately replicated the analog interface with its original tape-machine controls, recognizing that these operational metaphors carry compositional meaning. A 360-second buffer accommodates the extreme delays while independent playback heads simulate multi-track reading, but these technical solutions serve a deeper purpose: through systematic implementation, we uncover the implicit knowledge that links notation to realization. This approach differs radically from software updates or technical optimization; it uses reconstruction as an investigative tool to decode the score's semantic layers and preserve the interpretive intelligence required for future realizations.

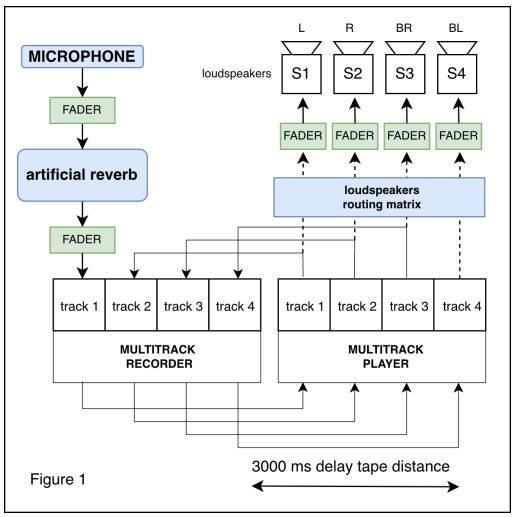


Figure 1. Block diagram of the performance setup redrawn from the original score of *Ricercare una melodia* (1984).

2.3 Spatial Choreography

Harvey's quadraphonic spatialization employs deliberate asymmetry. Rather than sequential speaker assignment (1–2–3–4), the score specifies non-linear routing (1–3–2–4), creating spatial trajectories that disrupt predictable circular motion. The four loudspeakers are assumed to be arranged in a standard quadraphonic configuration, surrounding the audience at the four corners of the performance space. This configuration transforms the performance space into what Harvey called a "theatre of transformation," where sounds emerge from unexpected locations, challenging spatial perception [6].

The original score includes precise fader choreography for measures 34-46, where dynamic control creates active spatial movement. These indications, absent from later editions, reveal Harvey's conception of space as compositional parameter rather than technical afterthought. Each fader movement serves both practical and poetic purposes: balancing the accumulating delays while sculpting the sonic environment.

2.4 Reconstruction as Interpretation

Rebuilding Harvey's system revealed layers of implicit knowledge embedded in the original notation. Seemingly simple indications like "fade" at measure 34 require multiple decisions: which channel fades, over what duration, following which curve? The score assumes familiarity with studio practice circa 1984, when certain procedures were standardized yet undocumented.

Our reconstruction process revealed how the original notational completeness, with all delay voices visible, fundamentally changes performance understanding. The presence of these staves transforms interpretation from guesswork to informed decision-making as shown in the reconstructed algorithm in figure 2.

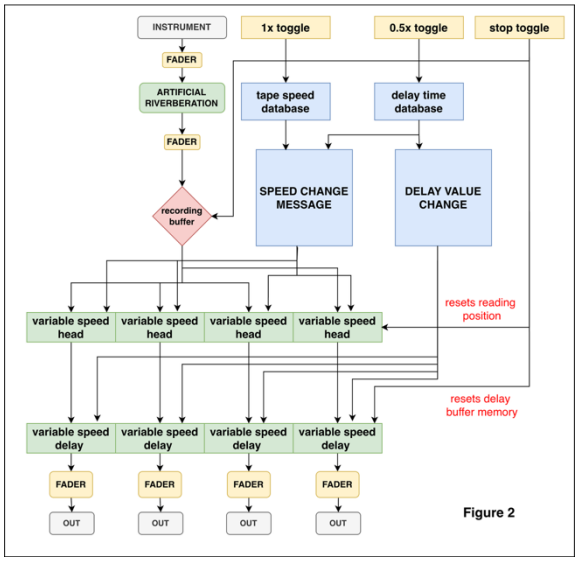


Figure 2. Flowchart for Max-Msp algorithm.

2.5 From Preservation to Discovery

This case study demonstrates how reconstruction functions as research in its own right. Through systematic implementation, we recovered not only technical specifications but also compositional intelligence: Harvey's use of delay as a harmonic generator, his exploitation of mechanical constraints as timbral resources, and his conception of electronic transformation as a spiritual metaphor. The rebuilding process revealed decision trees embedded within the notation—branching interpretive paths that demand aesthetic judgment rather than mechanical execution.

The Harvey reconstruction established several methodological principles foundational to our broader framework. Chief among them is the necessity of complete documentation: the explicit visibility of delay staves transforms performance from approximation to informed comprehension. Moreover, the score contains implicit cultural assumptions—unwritten conventions deciphered only through shared practitioner knowledge. Digital translation must therefore move beyond surface reproduction, preserving the structural relationships that sustain a work's identity across technological transitions.

Most importantly, the reconstruction validated our central hypothesis: the preservation of electroacoustic works depends on active interpretation, not passive updating or archiving. The agent we envision would not merely store Harvey's score but model its operational logic, retaining the procedural knowledge required to realize the work on future technologies while transparently recording the interpretive decisions involved in each realization.

3. THEORETICAL FRAMEWORK

3.1 From Symbols to Constraints: The Necessity of Supervised Learning

The electroacoustic corpus presents a fundamental challenge absent from traditional musical repertoires. While most corpora share interpretation rules across hundreds of similarly structured scores, each electroacoustic work constructs its own notational universe, a unique hybrid drawing on centuries of musical writing. Harvey's *Ricercare* exemplifies this complexity: stemless notes with time brackets recall medieval practice, a traditional 3/4 meter controls the rhythmic flow, and the speaker-routing diagrams introduce notation that is purely electronic. These heterogeneous systems do not merely coexist; they interact, creating work-specific grammars that resist generalization.

This notational heterogeneity makes unsupervised pattern recognition impossible. An agent trained on statistical regularities would fail to recognize that Harvey's stemless notation serves a different function than Stockhausen's, that identical symbols carry different meanings across works, that each score requires its own interpretive key. The solution lies in supervised annotation Human-in-the-Loop (HIL) processes where experts explicitly teach the machine each work's internal logic.

Through HIL annotation, we transform implicit knowledge into explicit constraints. When an expert marks Harvey's time brackets, they're not just labeling symbols

but teaching the agent how medieval notation functions within this specific electronic context. Each annotation embeds interpretive intelligence: this fermata aligns with tape splice points, this dynamic marking controls all five voices simultaneously, this rest creates deliberate gaps in the delay pattern.

Understanding, in our framework, means constructing a constraint network specific to each work [7]. This conception of understanding as relational and procedural rather than symbolic aligns with embodied and process-oriented theories of musical cognition, where meaning emerges from operational relationships rather than static representation. The agent doesn't learn universal rules but work-specific grammars—systems of relationships that govern how symbols interact within that particular score. Given Harvey's indication "fade" at measure 34, the agent must navigate constraints derived from this specific work: precedent patterns established earlier in the score, the technical specifications of the two-recorder system, the documented performance practice for this piece alone.

3.2 Two-Level Vocabulary Architecture

Our annotation methodology employs a systematic distinction between core transferable elements and work-specific symbols. This two-level architecture reflects a central insight: electroacoustic notation operates simultaneously on universal and particular registers.

The core vocabulary encompasses elements that transcend individual works—standard musical notation, common electroacoustic techniques (delay, reverb, spatialization), and established performance practices. These items form the grammatical backbone enabling comprehension across different compositional languages. They provide the stable foundation upon which work-specific innovations build. See figure 3.



Figure 3. Computer Vision Annotation Tool framework (CVAT), set of experimental labels.

The work-specific vocabulary captures the irreducible uniqueness of each composer's notational innovations—symbols that cannot be generalized without erasing their essential meaning. Harvey's particular routing diagrams, Stockhausen's *Formschema* structures, Boulez's numerical spatial coordinates—these must be learned as local dialects within the broader language. The semantic bridge between levels allows the agent to identify analogous structures across diverse works even when their surface notations differ radically.

This architecture enables progressive learning: the agent first masters universal elements, then uses these as scaffolding to comprehend work-specific innovations. Each new work potentially enriches the core vocabulary as initially unique symbols prove transferable to other contexts.

3.3 Annotation as Composition of Comprehension

Given the principle that annotation is interpretive translation, the practical challenge lies in navigating between over-specification (imposing interpretations that close legitimate alternatives) and under-specification (leaving ambiguities that prevent realization). Our solution preserves interpretive plurality: a single symbol can simultaneously carry multiple readings, each tagged with its source and justification. The agent learns not a single "correct" interpretation but a spectrum of legitimate possibilities.

Consider annotating Harvey's delay indications. The same notation might be marked as "structural delay" (emphasizing its role in canonical construction), "timbral transformation" (highlighting its acoustic effects), or "performance cue" (focusing on its execution requirements). These parallel annotations don't compete but complement, building a multi-dimensional understanding that approaches the richness of human interpretation.

The annotation process thus becomes close reading that makes explicit what human interpreters leave implicit. Each interpretive decision is documented, creating an audit trail of reasoning that the agent can traverse and extend. Through accumulated annotations, the agent develops what we might call "synthetic intuition"—the ability to make informed decisions in novel situations based on precedent patterns.

3.4 Operational Understanding and Its Limits

The agent's understanding necessarily differs from human comprehension. It cannot experience the phenomenological dimension of music—the feeling of rightness when a phrase resolves, the tension of an unresolved harmony, the spatial sensation of sounds moving through acoustic space. What it can maintain is structural integrity: identifying and preserving relationships that must hold for the work to maintain its identity.

This is crucially different from statistical pattern matching. The agent doesn't search for regularities across thousands of works to generate statistically plausible outputs. Instead, it identifies work-specific constraints through supervised observation. In Harvey's case, it learns how certain temporal relationships recur throughout the

work, how dynamics follow consistent trajectories through the delay system, how the transformation from Section A to Section B preserves underlying structural proportions despite radical changes in time scale.

These structural invariants serve as the skeleton upon which performance flesh can be added. The agent maintains the bones; human interpreters provide the breath. This division of labor acknowledges both the capabilities and limitations of computational understanding while establishing a productive partnership between human and machine intelligence.

4. Annotation Methodology

4.1 The Progression: From HIL to HOL

While Section 3 establishes the necessity of supervision, the operational workflow follows a distinct simplifying trajectory from Human-in-the-Loop (HIL) to Human-over-the-Loop (HOL).

Initially, the framework demands intense HIL engagement to establish the Core Vocabulary and defining fundamental relationships. However, as the agent masters these universal elements, the supervision dynamics shift. The transition is not cyclical but asymptotic: the "horizontal" layer of common notation becomes increasingly automated, requiring only HOL validation.

The Human-in-the-Loop component does not disappear but becomes surgically focused on the "vertical" dimension of the vocabulary—the work-specific innovations. When approaching a new composition, the agent leverages its established Core knowledge (HOL) while the human expert concentrates exclusively on teaching the novel, composer-specific dialects (HIL). This structure ensures that expertise is not wasted on retraining known concepts but is entirely devoted to the interpretive challenges of the new work.

4.2 Annotation Protocol Technical Implementation

Our annotation protocol, implemented through the Computer Vision Annotation Tool (CVAT) framework (Figure 4), structures the interpretive process through systematic layers. Each symbol receives multiple simultaneous annotations capturing different dimensional aspects: syntactic (what the symbol is), semantic (what it means), operational (how to execute it), and contextual grouping (how it relates to surrounding elements). CVAT is an open-source web-based annotation platform originally developed for computer vision datasets. It is implemented primarily in Python and JavaScript and allows collaborative labeling of visual materials through a browser interface. The software is publicly available and widely used for dataset preparation in machine learning contexts, which makes it suitable for structuring the multi-layer annotation process described here.

For Harvey's Ricercare, this multi-layered approach revealed the interconnected nature of seemingly independent notational systems. A single fermata might be simultaneously tagged as: duration marker (syntactic), tape splice point (semantic), buffer reset trigger (operational), and structural boundary (contextual). These parallel annotations don't compete but accumulate, building comprehensive understanding through complementary perspectives.

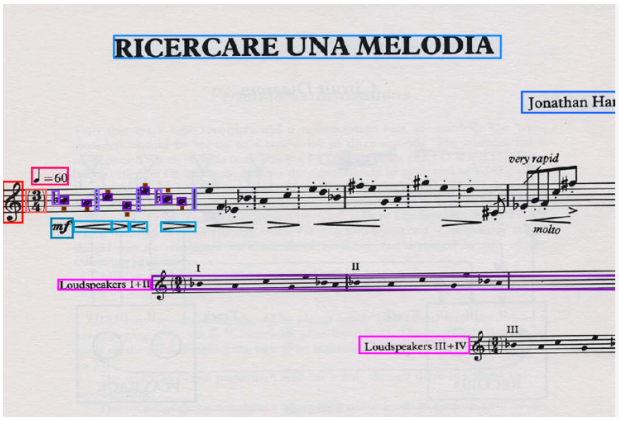


Figure 4. Annotation layers in CVAT framework.

The protocol accommodates interpretive disagreement as data rather than noise. When two annotators mark the same symbol differently, both interpretations are preserved with their justifications. This plurality becomes particularly valuable when dealing with ambiguous notation—instances where the composer deliberately left multiple possibilities open. The agent should learn to recognize and maintain this productive ambiguity rather than forcing premature resolution.

Technical challenges arise from the temporal nature of electroacoustic scores. Unlike traditional notation where symbols occupy discrete positions, electronic instructions often specify processes unfolding over time. Our solution employs temporal anchoring: linking annotations to specific time points while maintaining relationships across extended durations. This allows the agent to understand how a single instruction might govern an entire section's behavior.

A complementary layer aligns these temporal anchors with supervised spectral representations derived from performance recordings—mel-spectrograms, MFCC envelopes, spectral flux and other descriptors strictly relevant to the interpretive context. These recordings, in their final implementation, will be taken from realizations explicitly approved by the composers or by practitioners directly connected to the original performance tradition. The spectral layer is not introduced for classification or stylistic profiling but to reveal how notated processes manifest acoustically: the propagation of delay patterns, the energy profiles of feedback accumulation, the spectral consequences of half-speed playback or the filtering effects resulting from fader choreography. While the present work establishes the annotation protocol, the full system will integrate these curated recordings as an additional interpretive substrate. By synchronizing symbolic annotations with their acoustic outcomes, the agent will learn not only what the score prescribes but how its instructions materialize in sound.

4.3 Distributed Cognition and Knowledge Architecture

The annotation process instantiates distributed cognition-intelligence emerging from the interaction between human expertise and machine memory. Humans contribute pattern recognition, aesthetic judgment, and cultural knowledge; machines provide perfect recall, systematic consistency, and exhaustive cross-referencing. Neither alone possesses complete understanding; together they form a cognitive system exceeding individual capabilities.

This distribution operates across multiple scales. At the micro level, individual annotations embed human interpretation within machine-readable structures. At the macro level, accumulated annotations create emergent understanding—the agent begins to recognize principles no single annotator explicitly taught. Between these scales, the annotation community functions as collective intelligence, with each contributor adding perspectives that enrich the system's interpretive capacity.

The knowledge architecture maintains explicit genealogy for every decision. Each annotation links to its source—whether human expert, automated inference, or community consensus. This traceability serves multiple purposes: enabling revision when new evidence emerges, allowing users to evaluate the authority behind interpretations, and preserving the scholarly apparatus necessary for academic credibility.

4.4 Harvey as Methodological Laboratory

The annotation of Harvey's *Ricercare* revealed the granular complexity hidden within apparently simple instructions. Consider the indication "fade" appearing at measure 34—a single word that opens into a decision tree of interpretive possibilities. Which channel fades? Over what duration does the fade occur? Should it follow a linear or exponential curve? Does it affect only the direct signal or also the accumulated delays? Each answer shapes the sonic result, yet the score provides no explicit guidance.

Our annotation protocol for such moments employs parallel tracking: one annotator might interpret this as a master fade affecting all channels equally, preserving the balance established in previous measures. Another might read it as selective attenuation of the live signal while maintaining delay levels, creating a gradual transition from presence to memory. Both interpretations are preserved, tagged with their justifications drawn from precedent patterns in the score, documented performance practice, and the technical constraints of the two-recorder system.

This multiplicity extends throughout the annotation process. Harvey's fermatas, initially marked simply as sustained durations, revealed themselves through collective annotation as structural markers aligning with tape operations. Multiple annotators independently identified these points as mechanical necessities rather than expressive choices, their convergent observations strengthening the interpretation. Yet where interpretations diverged—whether a particular fermata served expressive or technical purposes—these disagreements were preserved as productive ambiguity.

The audit trail emerging from this process maps not just decisions but reasoning. When the agent encounters similar notational situations, it can traverse this trail, understanding not merely what was decided but why. A fermata in measure 15 was tagged as technical because it aligns with documented splice points; a similar fermata in measure 42 was tagged as expressive because it occurs mid-phrase without technical justification. The agent learns to distinguish between surface similarity and functional difference.

Through Harvey, we discovered that annotation density correlates with notational ambiguity. Passages with explicit technical specifications required minimal annotation—the score speaks clearly. But moments of notational sparseness, where conventional symbols carry unconventional meanings, demanded intensive interpretive labor. The accumulation of annotations around these ambiguous zones creates what we might call "interpretive mass"—regions where human intelligence has been concentrated to resolve underdetermination.

Most significantly, the Harvey annotation demonstrated how work-specific knowledge emerges from accumulated observations. No single annotator possessed complete understanding of the delay system's behavior, but through distributed annotation, patterns emerged that no individual had explicitly recognized. The collective intelligence of the annotation community exceeded the sum of individual contributions, validating our distributed cognition model not as theoretical abstraction but operational reality.

5. Three Experimental Fronts

The methodology developed through Harvey's reconstruction extends beyond a single work, addressing fundamental challenges in electroacoustic preservation. We selected three compositions not to establish a typology but to probe distinct fault lines in how notation encodes electronic practice. While only Harvey has been fully implemented, Stockhausen and Boulez illuminate the terrain ahead, each demanding different extensions of our annotation framework.

5.1 Harvey: Concealed Electronic Grammar

Ricercare una melodia presents deceptive simplicity. Its conventional notation—standard notes, dynamics, articulations—masks an intricate electronic architecture where five voices emerge from one, temporal relationships transform exponentially, and spatial trajectories follow non-linear paths. The surface appears familiar; the behavior is radically different.

Our reconstruction [5] served as methodological proving ground. Through systematic implementation, we discovered how supervised annotation could expose the operational logic that notation merely implies. The analog interface replication proved crucial—not for nostalgic authenticity but because operational metaphors carry semantic weight. When performers see "tape speed: 0.5x," they understand not just a technical parameter but a

compositional transformation where time dilates and pitch descends.

This first implementation confirmed that annotation can bridge the gap between what scores show and what they assume. Harvey's fermatas aren't arbitrary durations but splice points; his dynamics govern not one voice but five; his rests architect silence across accumulating delays. The score's conventional surface becomes legible only when its electronic grammar is reconstructed through supervised observation.

5.2 Stockhausen: Semantic Instability Through Time

Where Harvey conceals complexity beneath conventional notation, Stockhausen's *Solo* [8] presents a different challenge: identical symbols that mean different things depending on when they occur. The work's six *Formschema* create temporal containers ranging from 10.5 to 195.5 seconds. A gesture occupying half of a short cycle becomes a fleeting moment in a long one—not just proportionally different but phenomenologically transformed.

This systematic polysemy will test whether our framework can model context-dependent semantics. The annotation system must track not fixed meanings but relationships: how feedback density changes with cycle duration, how the same crescendo means seconds in one context and minutes in another, how structural position determines interpretive strategy. The agent cannot simply read symbols but must understand which temporal universe it inhabits at each moment.

Stockhausen thus probes the framework's capacity for meta-reading—recognizing when to shift interpretive modes based on higher-order structural markers. This capability, essential for navigating electroacoustic notation's heterogeneous systems, requires the agent to maintain multiple parallel frameworks without conflating them.

5.3 Boulez: Parametric Precision as Challenge

Boulez's *Anthèmes II* [9] inverts Harvey's problem entirely. Nothing hides; everything is specified. Spatial coordinates appear as numbers, trajectories as mathematical functions, transformations as precise algorithms. The notation achieves near-complete determinacy through parametric exhaustiveness.

This explicitness poses its own challenge: can annotation preserve mathematical precision without reducing it to mechanical execution? The framework must encode continuous spatiotemporal data while maintaining performative flexibility—capturing not just where sounds move but how that movement relates to gesture, phrase, and form. Boulez tests whether supervised annotation can handle high-resolution parametric information as musical rather than merely technical data.

These three fronts—hidden grammar, shifting semantics, parametric precision—map the territory our methodology must eventually cover. Harvey demonstrates that annotation can reveal; Stockhausen will test whether it can adapt; Boulez will probe whether it can maintain precision without sacrificing musicality. Together they define not a complete system but a research program, each work exposing different aspects of what preserving electroacoustic intelligence entails.

6. Politics & Boundaries

The works discussed in this study—by Jonathan Harvey, Karlheinz Stockhausen, and Pierre Boulez—were selected for methodological reasons. Each of them presents a particularly explicit relationship between instrumental notation, technological processes, and performance practice, making them suitable case studies for examining how procedural knowledge is embedded in electroacoustic scores. The aim of this article is therefore not to define a representative repertoire, but to analyse works in which the interaction between notation and technological systems is especially well articulated.

6.1 The Cost of Disambiguation

Every annotation is an act of power. When we label a symbol, define a relationship, or establish a reading, we exercise interpretive authority that will propagate through all subsequent interactions with the work. This power carries aesthetic commitments, cultural assumptions, and historical contingencies that shape what the music can become.

Electroacoustic notation frequently preserves intentional ambiguity as compositional strategy. An undefined gesture invites performer agency; an unspecified parameter opens space for contextual interpretation. Our annotation framework must render explicit what was deliberately left open, specify what was purposefully suggestive. This necessary disambiguation is not a flaw but a responsibility to acknowledge.

Our approach—maintaining parallel, coexisting interpretations within the agent's training—transforms disambiguation from reduction to multiplication. The agent doesn't learn a single authoritative version but a spectrum of valid interpretations, each linked to its origin and justification. Disagreement becomes data; contradiction becomes knowledge. The agent can explain why two readings may both be legitimate yet mutually exclusive, preserving the productive tension that uniform interpretation would erase.

6.2 The Canon Problem

By choosing which works to annotate, which interpretations to preserve, we inevitably contribute to canon formation. The agent will know Harvey, Stockhausen, and Boulez in meticulous detail while remaining unaware of composers whose work was never selected for preservation. This selection bias, embedded in training data, risks becoming an algorithmic destiny.

The danger amplifies through the perceived objectivity surrounding AI systems. An agent's interpretation carries computational authority, transforming contingent choices into seeming necessities. What begins as one researcher's selection becomes the tradition, crystallizing particular

visions of electroacoustic music at the moment when plurality matters most.

The framework offers no complete solution, only mitigation strategies. By maintaining transparent documentation of selection criteria, by preserving multiple interpretations rather than canonical readings, by acknowledging gaps in coverage, we at least make our biases visible and contestable. Future communities can add their own selections, creating parallel archives that preserve different traditions.

6.3 What Cannot Be Preserved

Acknowledging limits is essential to honest preservation. What we preserve are structural relationships, procedural knowledge, and explicit reasoning patterns connecting notation to realization. The AI architecture will never understand music as humans do, but it can maintain the technical and structural knowledge necessary for works to remain executable. This is not comprehensive preservation but functional preservation, enough to prevent these works from becoming unperformable, enough to maintain bridges to future practice.

7. CONCLUSION: TOWARD LIVING PRESERVATION

This paper emerges from a concrete problem: reconstructing Harvey's *Ricercare una melodia* revealed how much essential knowledge resides outside the score, in the gap between notation and realization, where editorial simplification can undermine preservation. Our analysis for implementation [5] exposed not only technical specifications but an entire grammar of assumptions, splice points masked as fermatas, routing logic embedded in spatial diagrams, and transformational procedures hidden beneath conventional symbols. Reconstruction thus became a research method, with each implementation decision uncovering a further layer of embedded intelligence.

Building on this practical foundation, we propose a supervised annotation framework that treats preservation as active interpretation rather than passive archiving. The distinction between core and work specific vocabularies reflects the dual nature of electroacoustic notation as both shared language and private dialect as an experimental probe of workability. Likewise, the shift from Human-in-the-Loop to Human-over-the-Loop models a durable division of labour: AI-based tools may gradually automate the universal, "horizontal" grammar, while human expertise continues to decode the "vertical" strata of composer analytical layers derived from the annotation process. Annotation understood as a "composition of comprehension" transforms machine teaching into an act of cultural preservation.

This work defines the conceptual architecture and methodological conditions required for such a system. The three experimental fronts, Harvey's reconstructed grammar, Stockhausen's context dependent semantics, and Boulez's parametric precision, map the methodological landscape ahead. Each work highlights a different challenge in preserving electroacoustic intelligence, from recovering implicit procedures to managing shifting meanings and encoding mathematically defined behaviour.

This phase of the research also evaluates the sustainability of a fully Human-in-the-Loop process for deep annotation. While such an approach allows the capture of detailed procedural knowledge, future developments will explore complementary strategies for scaling the process. These include segmentation techniques for the annotation workflow, geometric annotation layers for graphical notation, and integration with MEI and OMR pipelines for canonical notation. Analytical layers synchronization derived from this process may then be extracted both as structured vector datasets and as knowledge sources for retrieval augmented systems.

Future work will explore the multi-layer CVAT pipeline, model Stockhausen's temporal polysemy, and encode Boulez's spatial mathematics. These developments will test whether supervised annotation can sustain living performance traditions through transparent, revisable machine interpretation.

What we offer here is not a finished system but the conditions for one: a conceptual architecture enabling electroacoustic works to endure not as archaeological remains, but as practices that continue to evolve, held in the dynamic space between human creativity and machine memory, and preserved alongside the scores within existing archival standards.

8. REFERENCES

[1] Harvey, J. 1984. "*Ricercare una melodia*, for trumpet and tape delay system." London: Faber Music.

[2] Harvey, J. 2003. "*Ricercare una melodia*, for viola and electronics." Score and CD-ROM edition. CD-ROM contains Max/MSP 4.x patch originally developed at IRCAM for Macintosh (Mac OS 9/early OS X) using CNMAT OSC externals. London: Faber Music.

[3] Sterne, J. "The preservation paradox in digital audio" in Sound Souvenirs Amsterdam University Press ISBN 97890 8964 1328

[4] Faber Music. Jonathan Harvey – Works Catalogue. Available: https://www.fabermusic.com/we-represent/jonathan-harvey/works Accessed: March 2026.

[5] Roma, G. Battista, A. F. "Reimplementing Harvey's *Ricercare una melodia* for viola and electronics: a Performer-Centric Approach." Manuscript submitted to Computer Music Journal.

[6] Harvey, J. 1999. "In Quest of Spirit: Thoughts on Music." Berkeley: University of California Press.

[7] Godøy, Rolf Inge (2006). Gestural-Sonorous Objects: embodied extensions of Schaeffer's conceptual apparatus. Organised Sound. ISSN 1355-7718. 11(2), p. 149–157. DOI: 10.1017/S1355771806001439

[8] Stockhausen, K. *"Solo: for a melody instrument with feedback".* Wien: Universal Edition, 1966.

[9] Boulez, P. *"Anthèmes II for violin and live electronics".* Wien: Universal Edition, 1997.

CALM: Translating Somatic Experience into Compositional Structure as a Trauma-Informed Methodology

Sophie Rose
Australian Institute of Music and University of Melbourne
rose.s@unimelb.edu.au

ABSTRACT

CALM is a performance work and compositional system that translates bilateral, body-focused movement into sound behaviour and spatial form. Drawing on trauma-informed movement practices, the work treats somatic engagement as a generative compositional constraint rather than a representational or expressive metaphor. CALM deliberately subverts common assumptions that stillness and meditative movement are inherently calming. Instead, it externalizes bodily instability, fragmentation, and heightened internal noise that can arise for some trauma survivors during periods of stillness. Using wearable gestural interfaces, spatial audio, and voice-based synthesis, CALM maps movement rate, bilateral coordination, and physical effort to sonic density, timbre, and spatial distribution. Musical structure emerges through sustained bodily negotiation, constraint, and breakdown rather than through virtuosic control or symbolic gesture. The paper positions CALM as a practice-based methodological framework for translating somatic experience into sound behaviour, contributing to research in embodied computer music, trauma-informed creative practice, and participatory listening contexts.

Keywords: *yoga; trauma-informed practice; embodied interaction; somatic composition; practice-based research.*

1. INTRODUCTION

CALM is a work which constrains interaction through bilateral coordination [1], drawing on yoga as a body-focused trauma-informed movement practice [2] to emphasise left-right bodily engagement and coordination as a means of regulation and grounding. The physicality is captured by glove-based gesture devices that use the size and degree of continuous motions to alter sonic outcomes, representing an internal response to stillness. *CALM* was generated through practice-based research and asks how somatic experience can be translated into compositional structure through trauma-informed practice [3]. Rather than treating bodily movement as a gestural controller layered onto pre-existing musical logic, this work positions bodily experience itself as the generative substrate of composition. It is not designed to represent trauma narratively, nor to sonify emotion directly, but to translate lived somatic conditions – stillness, activation, physical effort and exertion, bilateral coordination, and bodily resistance – into the formal behaviours of sound.

Trauma-informed and body-focused engagement practices such as yoga, mindfulness, and bilateral stimulation[1] recognize the interconnectedness of mind and body and emphasise bodily sensation as a cue for regulation and meaning-making. These practices function on a case-by-case basis and often operate below conscious linguistic processing [4]. They aim to help individuals feel more present by restoring their connection to their bodies. *CALM* draws on this logic not to reproduce therapeutic exercises, but to extract compositional principles from them.

This paper extends my prior work, shifting from a focus on gestural mapping to somatic translation as a trauma-informed methodology [5]. Translation, in this context, is the process of structurally re-articulating somatic experience as sonic behaviour. This translation preserves ambiguity, instability, and nonlinearity, reflecting the way trauma persists in procedural, body-based memory.[2] It reinterprets somatic therapeutic techniques in musical terms as a system of practice, integration, and articulation of the embodied experience. DOI link to the work:

- 10.5281/zenodo.8197911

2. BACKROUND AND CONTEXT

A thorough discussion of therapeutic efficacy is beyond the scope of this work. *CALM* is not a music therapy project, and trauma-informed practices are not universally successful. Personal differences, contextual factors (for example, a personality clash between patient and professional), and bodily histories significantly affect therapeutic outcomes. This section lays out the concepts used in the development of the approach used in *CALM*.

Copyright: © 2026 Rose. This is an open-access article distributed under the terms of the Creative Commons Attribution License 3.0 Unported*, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.*

[1] A synonym for bilateral coordination.

[2] Procedural or body-based memory refers to non-verbal memory encoded through bodily action and sensation rather than narrative recall.

2.1 Trauma-informed Practice and Body-Focused Engagement

Trauma-informed practices were developed within feminist frameworks and began to be integrated into mainstream therapy in the 1980s and 1990s [2]. They involve deliberate attention to bodily sensations arising from psychological reactions rooted in the individual's trauma history and embodied responses to it that persist in the present.

Body-focused practices, developed within therapeutic and movement contexts, can take several forms, such as yoga, dance, and bilateral coordination [1]. Bilateral stimulation uses alternating stimuli across the midline of the body, as in EFT (Emotional Freedom Technique), tapping, and EMDR (Eye Movement Desensitization and Reprogramming).[3] In short, rhythmic alternation between the left and right sides of the body is widely used in therapeutic contexts to support nervous system regulation, reduce dissociation, and scaffold emotional processing.

Body-focused engagement emphasises the importance of bodily sensations as cues in the processing of trauma. The premise is that allowing an individual to reinvest, reconfigure, escape, or titrate traumatic memory through present-moment activity may release psychological damage from procedural memory, reducing some of its immediacy [6]. *CALM* draws on these principles as structural design logic for musical interaction rather than as therapy.

Choreographed movement practices (as in yoga and dance) are a subset of bilateral stimulation, in which synchronized, coordinated movements may cross the body's midline, be rhythmic or repetitive, or be mirrored. *CALM* does not seek seamless engagement but foregrounds moments of friction, resistance, and instability by borrowing these tropes and situating them against the reaction that is purported to alleviate the embodied response. This colours the movements themselves and the expectations placed on gestural language as captured through the datagloves, removing any content-neutral ambiguity from the movement progressions and bodily organisation that influence musical temporality and perceptual coherence [7]. *CALM* departs from a tendency in gestural composition toward semiotically ambiguous sound–action relationships [8], [9].

2.2 Structural Translation

Rather than embedding therapeutic exercises as narrative content, *CALM* translates somatic logic directly into compositional structure. Bodily balance, effort, and limitation determine how musical material is activated, sustained, and transformed. By feeling out the body's movement through balance, tension, and physical limitations, the system frames the sonic output in ways that run counter to general expectations [7]. The system requires coordinated left–right movement to activate and sustain musical material. This design choice intentionally slows interaction, discourages abrupt gestures, and enforces rhythmic regularity.

2.3 Yoga, Stillness, and the Problem of Internal Noise

Yoga is a group of mental, physical, and spiritual practices and philosophical concepts originating from India. It is often recommended to manage trauma and anxiety due to its focus on bodily awareness and meditative movement. However, clinical evidence does not fully support its effectiveness in specific trauma contexts [10], [11]. For some trauma survivors – particularly those with sexual trauma histories – yoga can provoke distress rather than grounding [12], [13]. Characteristically, yoga involves extended periods of stillness; these moments are most likely to trigger traumatic recall, as stillness can mimic freeze responses.[4]

Trauma often slinks around unbidden in the body, so listening inward during stillness may amplify distress rather than soothe [6], [16]. The seemingly benign qualities of yoga can tap into the implicit emotional memories, which are not reliably accessed through will but can be triggered through movement or situational prompts, to trigger an individual's trauma responses when not adequately supervised or safely applied [12], [13]. This is my lived experience. In saying this, one person's experience, while not generalizable, is likely to resonate with others simply because we are all human minds poured into a variety of human-shaped vessels, which influences our embodied experience and psychology [17], [18], [19].

In *CALM*, stillness is therefore equated with death, decay, and loss of control. Physical quietness becomes a sonic space in which internal anguish is audible. Low rates of change in movement amplify scream-like and abrasive sounds. Increased movement quietens the internal sonic landscape, representing freedom, harmony, and progression toward consonance. This inversion deliberately subverts commonly held Western assumptions about yogic practice: that stillness and meditation in specific contexts naturally bring peace.

3. METHODOLOGY

While this work emerged from an autoethnographic [20], [21] trauma-informed [2], [4] practice-based creative practice research, this paper focuses on generalizable compositional principles, making it relevant to a broader computer music community.

3.1 Technology and Mapping

CALM translates somatic experience through a system in which movement rate, implicit physical effort, and bilateral coordination directly shape sonic structure. Positional movement data is captured by the IMUs (inertial measurement units) in MiMU datagloves and transmitted via Wi-Fi to the Glover companion software [22], [23]. Controls are routed as OSC and MIDI information and sent into Max [24] and Ableton Live [25].

[3] These three use the same type of rhythmic alternating left-right touch, visual feedback, physical movement, or sound at different speeds to enact healing processes and/or thought-rewiring.

[4] The four Fs of fight, flight, freeze, and fawn are behavioural responses to traumatic, dangerous or threatening events [14]. In a freeze, an individual may feel unable to move or escape, which can manifest physically as rigidity or limpness [15].

Positional data is used to calculate velocity and quantity in Max; see Figure 1 and Figure 2. These values are converted into MIDI Control Change (CC) data before being passed to Ableton Live, which scales the amplitude of multiple audio layers. Changes in amplitude are proportional to bodily activation through the rate of change.

Gestural emblems [26] (see Figure 3) function as note-on and note-off triggers for effects in Ableton Live, including filter delay, vocoder reverb, ring modulation, and chorus effects. MIDI CCs control pairs of effects in combination with yaw, pitch, and roll data from the microcontrollers.

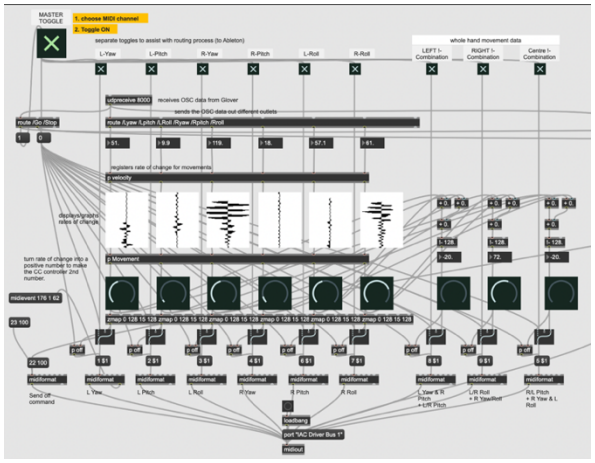


Figure 1. Max-patch calculates the rate of movement across the left and right hands through yaw, pitch, and roll.

While a minimal aspect of yoga, arm- and hand-based flourishes are added during some static poses (such as saddle; see Figure 5). These cues were taken from Pilates (a related form of exercise) to add additional momentum and keep the rate-of-change data consistent, thus keeping the audible material more consistent. This reflects the effort required to maintain or exit poses that is not captured by hand movements ordinarily, foregrounding physical labor. The speed and rate of movements control amplitude and effects across pre-composed audio layers, providing an alternative illustration of trauma's somatic remnants.

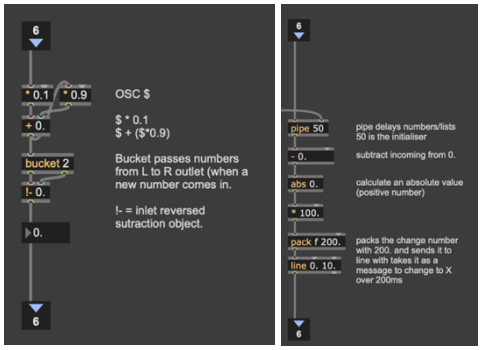


Figure 2. (left) Velocity calculations and (right) Movement sub-patch calculations.

The system privileges macro-movement and sustained or large movements over micro-gestures. This also makes the routing unsuitable for rapid improvisation, high-speed, or high-fidelity gestural virtuosity. The system is not plug-and-play; it requires embodied familiarization. Additionally, the system's transferability requires a redesign rather than replicating it for different devices or differently configured bodies, as these affect the system's behaviour.

This system externalizes bodily instability through resistance, latency, and non-linear response and privileges movement for its own sake rather than aiming for expressive transparency in the narrative sense. The body is translated into sound behaviour rather than encoded symbolically. Movement captured is relative, not absolute. Additionally, inertial sensors accumulate drift over time. While "zeroing" gestures are integrated into the routing, this still requires a certain amount of imprecision to be accounted for in the composition process.



Figure 3. Hand positions used to trigger effects in *CALM*: Puppet hand, OK, Open Hand, Fist, and One Finger Point.

The real-time sound latencies are not hugely disruptive to a performer's ability to execute the work, as they need not monitor audio to control the musical output. However, the metaphorical latency of delayed experience and the non-immediacy of bodily effort, when combined with a non-linear set of changes, introduce unpredictability or unintended sonic outcomes. Even small latencies interrupt the sound coupling in reflexive gesture use and introduce moments of uncertainty or wavering. It requires the performer to attend to the body, rather than chasing the sound, conceptually mirroring somatic awareness in practice. The premise of this work calls for slow interaction and constraint over virtuosity – a latency in action. As aesthetic coding, these small latencies rarely trigger large spikes in the audio but reflect effort over time; building, ebbing, trading off. In *CALM*, the latency is not treated as an error to be eliminated but as a structural characteristic of the somatic translation process.

Although the system's absolute latency is minimal, smoothed response curves introduce a delay between movement and sonic change, which is perceptible at times. This delay sometimes slows interaction, limits impulsive gestures, and supports the gradual modulation central to trauma-informed movement practices. As a result, timing, pacing, and instability become emergent compositional features. By embracing the introduced latencies as an affordance of the system, the resulting resistance mirrors the non-linear, delayed nature of bodily regulation, embedding somatic logic directly into the compositional structure.

Bodily instability functions as an affordance within the system. Micro-instabilities, particularly limb tremors captured by the gloves, introduce small, irregular fluctuations in control data (dissimilar to the added flourishes). At lower movement rates, these can produce perceptible stuttering or rapid oscillation between sonic states (for example, 02:33-02:52 mm:ss). Smoothing processes and latency intermittently amplify these effects at the threshold of detection, leading to momentary disruptions in otherwise continuous modulation. While sustained, controlled movement produces smoother trajectories, these interruptions introduce localized instability that enhances contrast and, at times, increases perceived consonance.

3.2 Composition Structure

The full score for *CALM* is available via the DOI on page 1. The choreography unfolds as a transition and tension between two sonic states: abrasive "stillness" material and more consonant "movement" material. The piece is bookended with traditional beginning and ending positions (child's pose and savasana, respectively), framing the performance within states of low bodily activation. Between these positions, the body progresses through floor-based, crouching, and standing configurations to vary the movement demand and sustain visual and physical intensity.

As performance progresses, increased bilateral coordination and sustained movement enable the emergence of harmonic vocal material, leading to a general shift toward consonance and perceived stability. This trajectory is not linear and the timing and execution vary between performances. Breakdowns in balance or coordination also interrupt this progression, reintroducing abrasive material and fragmenting continuity. The form is therefore shaped through ongoing negotiation between body and system.

3.3 Audio Materials

Two primary audio categories operate within *CALM*: "stillness" and "movement."

"Stillness" audio was produced by destructively editing live and pre-recorded scream samples through hand gestures and physical movement. These sounds were transformed through sampling, looping, granular synthesis, and spatial panning in a quadraphonic environment using the aforementioned tools.[5] Three gestural mediation mappings progressively destroyed samples: one controlling recording and looping, one controlling playback speed and direction, and one altering granular parameters and Euclidean rhythmic structures in HATEFISH RhyGenerator [27].

An additional thirty-eight audio tracks were generated through overlapping live improvisation and disfigured audio. Older audio generated during development was cut and spliced into newer gesturally created tracks, resulting in eight composite tracks spatialized using Ableton's Surround Panner plugin [28]. Hand gestures and IMU data also controlled volume and effects on movement audio, proportionally linked to the intensity of stillness. The constant threat of screaming remains present, even as consonant harmony emerges.

"Movement" audio was created by synthesizing the performer's voice using Native Instruments' Kontakt [29] at every pitch across the performer's range. Three choral vignettes were composed for three to eight voices, with the number of voices increasing over time to increase volume and harmonic density/complexity. Early sections were performed acoustically and blended into the virtual choir. Movement data controlled both amplitude and effects processing on these layers. These materials were similarly spatialized to those in "Stillness."

The choral sound is repeatedly disrupted through digital modulation, preventing it from appearing entirely human-generated. Audio effects were chosen to create an unearthly, digitally mediated choir. This externalizes internal sensation and maintains tension through the persistent threat of abrasive sounds. Consonant harmonies only emerge through bodily stimulation.

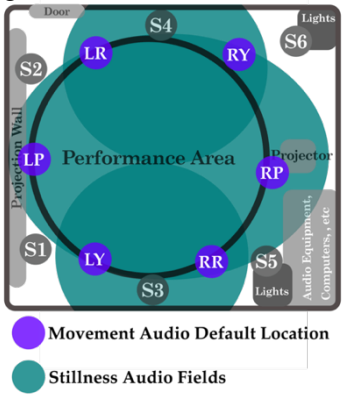


Figure 4. Audio locations in *CALM*.

Audio was placed in a six-channel audio environment.[6] The audience was seated on the rim of the performance area to create a sense of audio immersion with an individualized viewing experience. Figure 4 shows spatial placement and audio fields:

- S1-6 are the loudspeakers,
- Teal is the audio field of abrasive tracks. Their density is concentrated where the audience was most densely clustered.
- Purple are the individual audio tracks assigned to "Movement." These are labelled L and R (Left, Right), and Y/P/R for Yaw (horizontal left-to-right movement about the microcontroller's axis), Pitch (tilt up toward the ceiling or down toward the floor), and Roll (roll left-to-right about the microcontroller's axis).

Table 1 shows an overview of the bilateral sound-movement routings in *CALM*. Yaw, pitch, and roll independently modulate spectral brightness, amplitude, and timbral instability across "stillness" and "movement" audio layers (see 00:25–00:35, mm:ss). Parameter ranges are deliberately constrained to enforce slow interaction, prevent impulsive gestures, and embed trauma-informed somatic logic into the compositional structure.

Bodily Domain	Movement Source	Sonic Target	Parameter	Mapping Range	Compositional Role
Stillness	Left Hand (Rate of Change)	Stillness Audio (Left)	Spectral Brightness	10.1–22.0 kHz	Amplifies abrasive internal sound during physical stillness
	Centre (Cumulative Stillness)	Stillness Audio (Centre)	Spectral Brightness	10.7–22.0 kHz	Reinforces internal noise during low bodily activation
	Right Hand (Rate of Change)	Stillness Audio (Right)	Spectral Brightness	10.1–22.0 kHz	Spatialises internal distress across the sound field

[5] Glover, MiMU, Ableton Live, and Max.

[6] Due to the size of the room and loudspeakers, and availability of matching loudspeakers.

Movement Activation	Bilateral Coordination Threshold	Movement Audio Effects	Device On/Off	G0–C1	Discrete activation enforces intentional, non-reflexive gesture
Movement – Amplitude	Left Hand (Yaw/Pitch/Roll)	Movement Audio (Left)	Track Volume	−∞ dB to −10 dB	Gradual emergence of harmonic material through sustained movement
	Right Hand (Yaw/Pitch/Roll)	Movement Audio (Right)	Track Volume	−∞ dB to −10 dB	Bilateral balance shapes density and temporal pacing
Timbral Instability	Left Hand (Yaw/Pitch/Roll)	Modulation Effects (Vibrato/Ring/Chromatic)	Dry/Wet	0–41%	Introduces controlled instability tied to bodily effort
	Right Hand (Yaw/Pitch/Roll)	Modulation Effects (Vibrato/Ring/Chromatic)	Dry/Wet	0–41%	Maintains tension and prevents expressive transparency
Spatial Smearing	Left Hand (Yaw/Pitch/Roll)	Reverb (Cathedral/Wanderer)	Dry/Wet	0–43%	Expands perceived internal space without loss of control
	Right Hand (Yaw/Pitch/Roll)	Reverb (Cathedral/Wanderer)	Dry/Wet	0–61%	Asymmetric spatialisation reflects bodily imbalance

Table 1. CALM gesture routing summary and purposes.

3.4 Movement Selection and Trauma-Informed Design

Movements incorporated choreographically in *CALM* were selected to provide extramusical benefit without contributing to adverse effects. Trauma triggers can be activated through physical positioning, particularly when movements tap into procedural memory. This is especially relevant for sexual trauma, where areas such as the hips, diaphragm, chest, and abdomen commonly hold chronic tension [30]. In a chronic state, they need to be loosened gradually and consistently. So, while I chose these movements based on their comfort for my body, they transcend my experience and could be helpful for other performers.

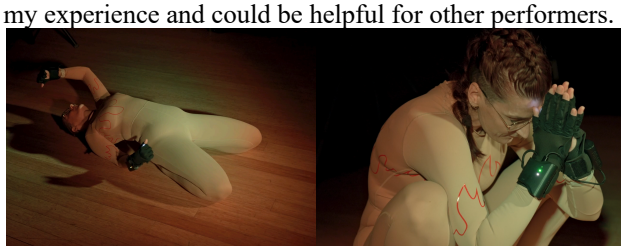


Figure 5. Saddle (left) and malasana (right) poses, taken from video stills.

For example, Malasana (a deep, standing squat; see Figure 5) was used instead of the Happy Baby pose because of its similar physical benefits and reduced vulnerability. Movements were modified based on bodily comfort and capability. I made and still use alterations to the poses to advance my flexibility, providing a direct physical benefit.[7] Saddle pose (Figure 5, left) sometimes requires ungainly exits into the next position. Though aesthetically unattractive, the imperfect execution reinforces the performer's humanity and challenges expectations of grace and virtuosity. It also shows that some poses might be difficult or even impossible for some performers, or even for the same performer on different days.



Figure 6. Mermaid (left) and half-camel (right) poses, taken from video stills.

This approach demonstrates that bodily limitation and variability are not failures but integral to the compositional system. The poses used include child's pose, puppy dog, half camel, camel, saddle, mermaid side stretch, seated star, malasana, up and down dog, warrior two, peaceful warrior, dancer, mountain, and crucifix; see Figure 5 and Figure 6 for examples.

In *CALM*, musical time is not externally imposed but emerges from bodily rhythm. The requirement for bilateral alternation produces elongated temporal forms, longer sound events and a pedaling between two conflicting states, and a reduced gestural density (drawing from a subset of available gestures). This results in a musical texture that resists acceleration and overload.

3.5 Transferability of the System Logic

While the specific mappings, parameter ranges, and system architecture are implementation-dependent and reflect the affordances and limitations of this system, the underlying design principles are intended to be transferable. These include using bilateral coordination as a constraint, translating somatic states into sonic behaviour, and treating latency, resistance, and non-linearity as compositional features. These principles extend beyond this system and can be adapted through different gesture vocabularies and sensor placements across the body.

4. DISCUSSION

The system foregrounds instability, resistance, and ambiguity as productive musical forces. Rather than resolving tension, *CALM* sustains it, reflecting the ongoing nature of bodily negotiation in the process of trauma recovery. When bilateral coordination breaks down, the system responds musically through fragmentation, loss of continuity, or spectral thinning. These moments are not errors but integrated compositional material, allowing breakdown to become an audible structure (see 09:55–10:36 and 02:00–02:11, mm:ss). Works such as Alexander Schubert's *Your Fox's, a Dirty Gold* [31] and Laetitia Sonami's *Dreams of*

[7] For example, a full malasana only became possible after eight months of practising and working through different modifications.

Control, Dreams of Chaos [32] similarly employ technological intervention to introduce instability and moments of aesthetic rupture in pop and experimental performance, but without considering the body's potential for failure.

Within this framework, failures or other errors directly contribute to the sound and performance aspects. Wobbles or collapses generated while maintaining balance feed back into the system as movement. Balancing is an act of bodily attention – of maintaining control – while failure is musical material, feeding into the glitch as a form of resistance [33]. This approach shares affinities with Atau Tanaka's performance work, where effort, force, physical resistance, and touch shape sonic output through electromyogram arm bands [34]. Following Maxine Sheets-Johnstone's assertion that movement is a primary mode of knowing rather than a derivative expression [7], *CALM* treats bodily motion not as an input to be interpreted, but as the generative condition through which musical structure emerges.

Though designed to subvert yoga practice due to the potential negative side-effects for some trauma survivors, *CALM* does not aim to mock or dismiss yoga's (or any other therapeutic practice's) usefulness for others. Nor does it seek to critique well-intentioned advice or the centuries-old cultural practice.[8] It simply externalizes a lived experience that contrasts dominant narratives. For those whose experiences differ, they are sometimes felt as deviant – aberrations or glitches within normalcy [33]. *CALM* provides a small point of resistance; failure to adhere to expectations, particularly in the performance of wellness [33], [35].

From performance perspectives, *CALM* requires sustained bodily awareness and attentiveness. The system discourages virtuosic display and instead prioritizes listening, pacing, and embodied presence (see 02:13–03:15, mm:ss). The voice remains silent,[9] and the performer's soft footsteps do not meaningfully contribute to the sonic environment. The performer's silence functions as a structural restraint rather than an expressive absence. This emphasis on negotiation over control invokes co-adaptation through use, where instability and unpredictability are integral to the instrument's behaviour and inclusion [32], [36].

Participation in *CALM* occurs primarily through listening. Audiences are invited into a somatic logic in which bodily effort shapes sound, fostering empathy through shared perceptual experience rather than didactic explanation. The performer participates through their own embodied regulation and, at times, through the negotiation of balance. This engagement is entirely physical, though distinct from traditional instrumental performance, aligning more closely with movement- and dance-based modes of musical embodiment, inviting audiences to attune to bodily pacing rather than spectacle.

The performance environment is circular; however, in practice, the audience often chooses to sit on two sides of the circle to avoid the projector line. Abrasive audio clusters around these seating positions, maximizing discomfort caused by their proximity to harsh noises in symbolic and perceptual terms. The immersive audio environment creates a sense of sonic envelopment, activating innate sound-localization abilities [37, Ch. 2] and drawing the audience into the shared psychological space that accommodates both positive and negative stimuli.

Anecdotally, the work has resonated with others around who share similar histories. One audience member described the experience as "like watching their own internal meltdown from the outside." Seeing discomfort reflected on stage can evoke mixed responses; however, the open articulation of such experience counters its frequent dismissal by well-meaning observers or outsiders. In this way, *CALM* aims to foster awareness of the trauma's less visible aftereffects, particularly for those whose embodied reactions fall outside the dominant norms, which must often be defended as legitimate lived experience.

CALM does not equate expressivity with freedom. It embeds constraint, regulation, and pacing into the interface, allowing bodily coherence to shape interaction. In this framework, restraint can be as generative as virtuosity, particularly in pedagogical and participatory performance contexts. As Sonami remarked, we so want to "explore and dominate the unknown, yet leave it behind as soon as it becomes domesticated" [32, 00:05:19-00:05:25]. *CALM* shifts focus from augmentation as enhancement towards augmentation as friction [38, p. 349]. The sound systems function as an extension and representation of the body, amplifying instability and limitation rather than overcoming them [7], [39].

CALM operates within a practice-based research framework in which knowledge is generated through doing rather than prior theoretical modelling [3]. Iterative bodily engagement, system failure, and reconfiguration shape the work's compositional structure, demonstrating how somatic experience can organize musical form [40], [41].

5. CONCLUSIONS

CALM reframes somatic experience as compositional structure through bilateral choreography and trauma-informed practice. By translating bodily activation, stillness, and effort into sound behaviour, the work demonstrates how trauma-informed practice can function as a generative methodology rather than a representational theme. Stillness becomes instability and internal harshness, while movement becomes agency through constraints.

As a methodological contribution, *CALM* demonstrates how trauma-informed movement practices can function as generative compositional frameworks, offering pathways for embodied computer music that prioritize lived bodily experience. The system's deliberate constraint (slowed interaction, bounded parameter ranges, and discrete activation) positions latency, resistance, and non-linear response not as errors to be eliminated but as structural characteristics of somatic translation. Breakdown, imbalance, and bodily limitations can become an audible structure rather than a failure in this process.

[8] As a caveat, I always critique the Western tendency to create a supermarket-of-spirituality approach to Eastern cultural and spiritual practices due to the embedded colonial take-and-appropriate attitudes that frequently accompany the attitude.

[9] Unless a groan or other marker of physical exertion escapes the body.

As a piece and method, *CALM* positions embodiment as a generative structural force inside the expressive overlay; it offers an approach for translating somatic knowledge into an interaction design system beyond clinical and fitness contexts, with implications for other embodied systems that prioritize restraint, pacing, and bodily coherence as the material for composition. Future systems may benefit from designing for regulation rather than expressivity, as is seen in some music therapy processes. For example, George Khut's work using children's heart rate monitors to create an embodied feedback loop to reduce pre-surgery anxiety [42].

5.1 Future Work

The movement progression in this work will be reworked into another movement framework: Floor Flow® by Marlo Fisken [43], a subset of Flow Movement®, in which the floor functions as the primary apparatus. This style foregrounds momentum and conscious engagement with the body and emphasises the body's feeling over the outward aesthetic form. The working title for the piece is *FLOW><FOIL.*[10] The yoga-derived movement vocabulary will be reversed and navigated through this dance style, which incorporates circular looping patterns and low-to-the-ground, slithering motions. In contrast to *CALM*, sound will be linked more directly to harmony and rhythm. Constant rates of change will stabilize the pulse and meter, while stillness will cause dissonance, interruption, and rhythmic stuttering.

Future work will also include collaboration with ensembles working in other dance languages (for example, Butoh). These contexts provide further possibilities to investigate how somatic translation operates across a variety of aesthetic, spatial, and temporal logics in relation to ensemble coordination, collective pacing, and spatialization strategies.

6. REFERENCES

[1] O. T. H. Children, "What is 'Bilateral Coordination'?," Occupational Therapy Helping Children. Accessed: May 20, 2023. [Online]. Available: https://occupationaltherapy.com.au/what-is-bilateral-coordination/

[2] M. Curi, "A Short History of Trauma-Informed Care," *IowaWatch: Investigate MidWest Newsroom*, USA, p. 1, Jun. 15, 2018.

[3] L. Candy, *The Creative Reflective Practitioner: Research Through Making and Practice*. UK: Routledge, 2019. doi: https://doi.org/10.4324/9781315208060.

[4] H.-R. Han *et al.*, "Trauma informed interventions: A systematic review," *PLoS One*, vol. 16, no. 6, p. 28, Jun. 2021, doi: 10.1371/journal.pone.0252747.

[5] S. Rose, "CALM: Mapping yoga practice for gestural control to externalise traumatic experiences," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, Mexico City, Mexico: NIME, 2023, pp. 608–611. doi: http://nime.org/proceedings/2023/nime2023_89.pdf.

[6] P. A. Levine, *Trauma and Memory: Brain and Body in a Search for the Living Past: A Practical Guide for Understanding and Working with Traumatic Memory*. USA: North Atlantic Books, 2015.

[7] M. Sheets-Johnstone, *The primacy of movement*, Expanded 2nd ed. in Advances in consciousness research, no. v. 82. Amsterdam ; Philadelphia: John Benjamins Pub. Co, 2011. Accessed: May 15, 2020. [Electronic resource]. Available: https://ebookcentral.proquest.com/lib/unimelb/detail.action?docID=784232

[8] D. Smalley, "Space-form and the acousmatic image," *Organised Sound*, vol. 12, no. 1, pp. 35–58, Apr. 2007, doi: 10.1017/S1355771807001665.

[9] R. Fischman, "A Manual Actions Expressive System (MAES)," *Organised Sound*, vol. 18, no. 3, pp. 328–345, Dec. 2013, doi: 10.1017/S1355771813000307.

[10] H. Cramer, D. Anheyer, F. J. Saha, and G. Dobos, "Yoga for posttraumatic stress disorder – a systematic review and meta-analysis," *BMC Psychiatry*, vol. 18, no. 1, p. 72, Mar. 2018, doi: 10.1186/s12888-018-1650-x.

[11] J. Taylor, L. McLean, A. Korner, E. Stratton, and N. Glozier, "Mindfulness and yoga for psychological trauma: systematic review and meta-analysis," *Journal of Trauma & Dissociation*, vol. 21, no. 5, pp. 536–573, Oct. 2020, doi: 10.1080/15299732.2020.1760167.

[12] P. A. Levine, V. Yalom, and M.-H. Yalom, "Peter Levine on Trauma Healing: A Somatic Approach," 2010. Accessed: Jan. 24, 2023. [Transcript]. Available: https://www.psychotherapy.net/interview/interview-peter-levine

[13] J. L. Herman, *Trauma and Recovery: The Aftermath of Violence - From Domestic Abuse to Political Terror*. New York; USA: Basic Books, 1992.

[14] A. Marschall, "The Four Fear Responses: Fight, Flight, Freeze, and Fawn," Verywell Mind. Accessed: Jul. 11, 2023. [Online]. Available: https://www.verywellmind.com/the-four-fear-responses-fight-flight-freeze-and-fawn-5205083

[15] N. B. Schmidt, J. A. Richey, M. J. Zvolensky, and J. K. Maner, "Exploring Human Freeze Responses to a Threat Stressor," *J Behav Ther Exp Psychiatry*, vol. 39, no. 3, pp. 292–304, Sep. 2008, doi: 10.1016/j.jbtep.2007.08.002.

[16] B. van der Kolk, *The Body Keeps the Score: Mind, Brain and Body in the Transformation of Trauma*. UK: Penguin UK, 2014.

[17] T. Rohrer, "The Body in Space: Dimensions of Embodiment," in *Body, Language, and Mind: Cognitive Linguistics Research*, vol. 1, E. Torres, Ed., in Embodiment, vol. 1. , Germany: Walter de Gruyter GmbH & Co, 2008, pp. 339–378. [Online].

[10] As in a narrative foil – an opposing character where there is a reciprocal relationship to understanding both characters better through their contrasting attributes.

Available: https://api.pageplace.de/preview/DT0400.9783110199116_A19081595/preview-9783110199116_A19081595.pdf
[18] T. Rohrer, “Embodiment and Experientialism,” in *The Oxford Handbook of Cognitive Linguistics*, D. Geeraerts and H. Cuyckens, Eds., 2010, pp. 25–47. doi: 10.1093/oxfordhb/9780199738632.013.0002.
[19] R. M. Sapolsky, *Behave: The Biology of Humans at Our Best and Worst*. USA: Penguin, 2017. [Online]. Available: https://archive.org/details/Behave
[20] T. Spry, *Body, Paper, Stage: Writing and Performing Autoethnography*. California; USA: Routledge, 2011. Accessed: Apr. 07, 2020. [Online]. Available: http://ebookcentral.proquest.com/lib/unimelb/detail.action?docID=843865
[21] D. E. Reed-Danahay, Ed., *Auto/Ethnography: Rewriting the Self and the Social*. in Explorations in Anthropology. New York; USA: Oxford, 1997. Accessed: Sep. 13, 2019. [Online]. Available: https://content.ub.hu-berlin.de/monographs/toc/ethnologie/BV025182531.pdf
[22] MiMU Gloves LTD, “MiMU,” Gestural Interface: Datagloves, 2010. Accessed: Oct. 18, 2021. [Online]. Available: https://mimugloves.com/
[23] MiMU Gloves LTD, *Glover*. (2022). MI.MU Gloves Ltd, UK.
[24] Cycling ’74, *Max*. (2018). C, C++, JUCE. Cycling ’74, USA. [macOS]. Available: https://cycling74.com/
[25] Ableton, *Ableton Live*. (2021). C++. Ableton, Germany.
[26] R. I. Godøy and M. Leman, *Musical Gestures: Sound, Movement, and Meaning*. London, UNITED KINGDOM: Routledge, 2009. Accessed: Apr. 16, 2023. [Online]. Available: http://ebookcentral.proquest.com/lib/unimelb/detail.action?docID=465371
[27] S. Vigni, *HATEFISH RhyGenerator*. (2019). C++. Hornet Plugins, Italy. Accessed: Feb. 01, 2023. [VST; macOS]. Available: https://www.hornet-plugins.com/plugins/hatefish-rhygenerator/
[28] Ableton, *Surround Panner*. (2020). C++. Ableton, Germany. Accessed: Mar. 14, 2022. [MacOS]. Available: https://www.ableton.com/en/packs/surround-panner/#?item_type=max_for_live
[29] Native Instruments, *KONTAKT 7*. (2021). C++. Native Instruments GmbH, Germany. Accessed: Feb. 01, 2023. [macOS]. Available: https://www.native-instruments.com/en/products/komplete/samplers/kontakt-7/
[30] American Psychological Association, “Stress effects on the body,” *American Psychological Association*. in Stress. American Psychological Association, USA, p. 1, Nov. 01, 2018. Accessed: Feb. 01, 2023. [Online]. Available: https://www.apa.org/topics/stress/body
[31] A. Schubert, Giga Hertz Award, ZKM, Karlsruhe. *Your Fox’s, A Dirty Gold*, (2011). Accessed: Mar. 31, 2026. [For solo performer with voice, motion sensors, electric guitar and live-electronics]. Available: https://www.youtube.com/watch?v=nJ7Ygns2f9Q
[32] L. Sonami, “Dreams of Control, Dreams of Chaos,” presented at the NIME 2014, 2014. Accessed: Nov. 02, 2020. [Online]. Available: https://www.youtube.com/watch?v=nEkufXDb-Ro
[33] L. Russell, *Glitch Feminism: A Manifesto*. Verso Books, 2020.
[34] A. Tanaka, *닻올림 연주회 dotolim concert series_187 Atau Tanaka 아타우 다나카*, (Aug. 28, 2024). Accessed: Aug. 28, 2024. [Online Video]. Available: https://www.youtube.com/watch?v=q37YXNPOzY8
[35] R. James, *Resilience & Melancholy: Pop Music, Feminism, Neoliberalism*. USA: Zero Books, 2015. [Online]. Available: https://archive.org/details/resiliencemelanc0000jame_x6n7
[36] Stelarc, *Exoskeleton*. 1998. Accessed: Apr. 01, 2026. [Walking Machine]. Available: http://stelarc.org/_activity-20227.php
[37] J. Blauert, *Spatial Hearing: The Psychophysics of Human Sound Localization*. The MIT Press, 1996. doi: 10.7551/mitpress/6391.001.0001.
[38] P. Z, “A Tool is a Tool,” in *Women, Art, and Technology*, J. Malloy, Ed., Cambridge, Mass; USA: MIT Press, 2003, pp. 348–361. Accessed: Jul. 18, 2023. [Online]. Available: https://pamelaz.com/tool.htm
[39] M. H. S. Klug, “The Speaker is a Performer: Technological Design and Agencies in Electronic Music Soundsystems,” Bachelor of Arts with Departmental Honors, Wesleyan University, Middletown, CT, 2019. doi: 10.14418/wes01.1.1520.
[40] K. Höök, *Designing with the Body: Somaesthetic Interaction Design*. USA: MIT Press, 2018. [Online]. Available: https://doi.org/10.7551/mitpress/11481.001.0001
[41] G. Sullivan, *Art Practice as Research: Inquiry in Visual Arts*, 2nd ed. USA: SAGE Publications, Inc., 2010. Accessed: Aug. 26, 2019. [Online]. Available: https://au.sagepub.com/en-gb/oce/art-practice-as-research/book233902
[42] K. L. O. Burton, A. M. Morrow, B. V. Beswick, and G. P. Khut, “The Feasibility of Using the BrightHearts Biofeedback-Assisted Relaxation Application for the Management of Pediatric Procedural Pain: A Pilot Study,” *Pain Practice*, vol. 18, no. 8, pp. 979–987, 2018, doi: 10.1111/papr.12696.
[43] M. Fisken, *Flow Movement*. 2015. Accessed: Jan. 31, 2023. [Movement style]. Available: https://flowmovement.net

Snapping Matters: Context-Aware Onset Refinement for Automatic Music Transcription

Abhirup Saha
International Audio Laboratories Erlangen
abhirup.saha@audiolabs-erlangen.de

Hans-Ulrich Berendes
International Audio Laboratories Erlangen
hans-ulrich.berendes@audiolabs-erlangen.de

Meinard Müller
International Audio Laboratories Erlangen
meinard.mueller@audiolabs-erlangen.de

Ben Maman
International Audio Laboratories Erlangen
ben.maman@audiolabs-erlangen.de

ABSTRACT

Precise note-level annotations are critical for training automatic music transcription (AMT) systems, in particular note-onset labels, which form a core component of many recent AMT systems. However, high-quality annotations for real-world recordings are scarce. Sequence-level score–audio alignment methods such as dynamic time warping provide only coarse correspondence, making a local refinement step necessary. This refinement step, known as snapping, adjusts aligned score onsets using peaks in a neural onset posteriorgram and often determines whether weakly aligned score–audio pairs become usable training data at all. Despite its practical importance, snapping is typically treated as a simple post-processing heuristic and implemented with greedy local decisions. We present a systematic analysis of snapping strategies for training instrument-agnostic transcribers, demonstrating that snapping is essential for learning from weakly aligned data. Building on this, we formulate snapping as a per-pitch assignment problem and solve it via bipartite graph matching, yielding context-aware onset decisions under overlapping refinement windows and uncertain initial alignments. Extensive cross-dataset experiments across piano, chamber, and orchestral recordings show improved onset alignment and transcription accuracy over greedy snapping, with gains increasing for wider snapping windows and coarser initial alignments. Qualitative examples are provided on our project page: `https://abhirupsaha8.github.io`

1. INTRODUCTION

Automatic music transcription (AMT), the task of converting audio recordings into symbolic note representations, remains a central problem in music information retrieval (MIR). While piano transcription has progressed rapidly, reliable onset and pitch estimation for non-piano instruments and multi-instrument mixtures is still difficult due to timbral variability, polyphony, and the limited availability of precisely annotated training data. In particular, modern data-driven AMT models rely on supervision with accurate note onsets, yet onset-level ground truth is hard to obtain for real recordings outside controlled acquisition settings.

Copyright: ©2026 Abhirup Saha et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License 3.0 Unported, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

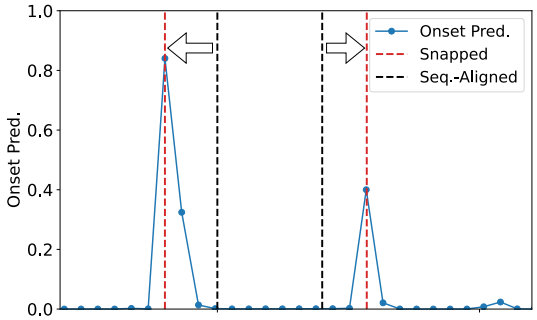


Figure 1. Snapping for a single pitch. The black dashed line ("Seq.-Aligned") shows onset timings from a sequence-level alignment (e.g., DTW), while the red dashed line ("Snapped") indicates the adjusted onsets after snapping. The blue line denotes activations in an onset posteriorgram.

A common workaround is to derive training labels by aligning a musical score (or a MIDI file) to the audio. Sequence-level alignment methods such as dynamic time warping (DTW) can provide robust coarse correspondences, for example based on chroma- or onset-related features [1, 2, 3]. However, even strong sequence-level alignments typically do not yield onset-accurate labels. Expressive timing deviations and local asynchronies (e.g., arpeggiation) can cause score events that are simultaneous in the score to be realized at different times in the audio, or vice versa. Consequently, directly transferring score onsets through a warping path often yields labels that are insufficiently precise for effective AMT training.

To bridge this gap, many recent pipelines adopt a two-stage strategy [7, 4, 14]: first compute a global, sequence-level score–audio alignment, then refine individual note onsets using neural onset activations, a step commonly referred to as snapping. In snapping, each score onset timing is refined within a local temporal window to a nearby peak in a learned onset posteriorgram, as depicted in Figure 1. Despite its practical success and growing adoption, snapping is typically implemented using local greedy decisions or simplified peak-picking heuristics. This is fragile when refinement windows overlap, because locally optimal choices can conflict, leading to duplicate assignments, missed peaks, or reduced global consistency. The problem becomes particularly pronounced when the initial alignment

is coarse and wider refinement windows are required.

This paper revisits *snapping* as an essential component of AMT label generation and asks how onset refinement should be formulated and solved to remain reliable under realistic alignment uncertainty. Our key idea is to treat snapping not as independent peak picking, but as a structured decision problem. For each pitch, we assign note events to candidate audio frames within admissible time windows, maximizing onset-posterior evidence while enforcing one-to-one consistency. This yields a principled and scalable refinement step that remains robust in the presence of overlapping windows and coarse initial alignments. The main contributions of this paper are as follows:

- We clearly distinguish *sequence-level alignment*, which defines a warping relation between timelines, from *note-onset-level alignment*, which maps discrete events to precise times, and we position snapping as the refinement step that connects the two.
- We formulate snapping as a per-pitch assignment problem based on bipartite graph matching, explicitly handling overlapping windows and global consistency beyond greedy heuristics.
- We provide a systematic cross-dataset evaluation analyzing how solver choice, window size, and initial alignment quality affect both alignment accuracy and downstream transcription performance, including robustness under very coarse initial alignments.
- We show that graph-based snapping yields consistent improvements in transcription accuracy across diverse datasets and becomes increasingly beneficial as refinement windows grow, which is precisely the regime required when labels are only weakly aligned.

This establishes snapping as a principled and tunable refinement step that bridges robust sequence alignment and onset-accurate supervision, enabling more reliable training for instrument-agnostic AMT under realistic conditions.

The remainder of this paper is organized as follows. Section 2 reviews related work on AMT and audio–score alignment. Section 3 introduces our method, including formal definitions of sequence- and onset-level alignment and a mathematical formulation of graph-based snapping. Section 4 defines the transcription and alignment tasks considered in this work. Section 5 presents experiments and evaluation procedures. Section 6 summarizes the findings and outlines directions for future work.

2. RELATED WORK

For an overview of automatic music transcription, see [8]. Early work focused on piano transcription, initially using non-negative matrix factorization (NMF) [9], before giving way to data-driven approaches. Key advances such as Onsets and Frames [10] and the MAESTRO dataset [11] enabled high-quality polyphonic piano transcription [12] based on note-onset detection.

For multi-instrument datasets [13], score–audio alignment via DTW [2] is common but prone to errors. Recent work [4] shows that onset features from pre-trained neural transcription models enable more precise score–audio alignment, improving sequence-level robustness [3] and allowing accurate note-onset annotations via snapping. A similar strategy was later used to construct guitar datasets [5, 14]. More recent work further simplifies alignment using histogram-based top-K peak picking [6].

While initially applied to piano and other instruments with sharp onsets such as guitar, onset-based transcription has been extended in recent work to strings, winds, and multi-instrument settings, both instrument-sensitive [16, 17, 4] and instrument-agnostic [18, 6].

3. METHOD

In this section, we describe our proposed method in detail. In Section 3.1, we formally define the notions of sequence-level and note-onset-level alignment, highlighting the differences between them. Then, in Section 3.2, we formulate how snapping—the refinement of a sequence-level alignment into an onset-level alignment—can be performed optimally using bipartite graph matching based on a learned note-onset posteriorgram.

3.1 Sequence-level vs. Note-Onset-level Alignment

Given an audio recording of a musical piece and a corresponding digital representation of the score, such as a piano roll or a MIDI file, we define two discrete timelines: the score timeline

$$\mathcal{T}_{\mathrm{s}} = [1 : T_{\mathrm{s}}] \coloneqq \{1, 2, \ldots, T_{\mathrm{s}}\} \tag{1}$$

of length $T_{\mathrm{s}} \in \mathbb{N}$, and the audio (physical) timeline

$$\mathcal{T}_{\mathrm{a}} = [1 : T_{\mathrm{a}}] \coloneqq \{1, 2, \ldots, T_{\mathrm{a}}\} \tag{2}$$

of length $T_{\mathrm{a}} \in \mathbb{N}$. We further define a set of possible pitches $\mathcal{P}$ of size P, and a set of possible instruments $\mathcal{I}$ of size I:

$$\mathcal{P} = \{p_1, \ldots, p_P\}, \quad \mathcal{I} = \{i_1, \ldots i_I\}. \tag{3}$$

The musical score can then be represented as a set of notes

$$\mathcal{N} \subset \mathcal{P} \times \mathcal{T}_{\mathrm{s}} \times \mathcal{T}_{\mathrm{s}} \times \mathcal{I} \tag{4}$$

of size N. Each note $n \in \mathcal{N}$ is represented as a tuple

$$n = (p,\, t^{\mathrm{on}},\, t^{\mathrm{off}},\, i) \tag{5}$$

where $p \in \mathcal{P}$ denotes the pitch, $t^{\mathrm{on}} \in \mathcal{T}_{\mathrm{s}}$ and $t^{\mathrm{off}} \in \mathcal{T}_{\mathrm{s}}$ are the note onset and offset positions on the score timeline, and $i \in \mathcal{I}$ is the instrument class.

A *sequence-level alignment* is defined as a function

$$\mathcal{W} : \mathcal{T}_{\mathrm{s}} \to \mathcal{T}_{\mathrm{a}}, \tag{6}$$

which maps each score-frame index to a corresponding audio-frame index. Such alignments are typically estimated from frame-level similarity using DTW or similar methods.

A *note-onset-level alignment* is a function

$$\mathcal{M} : \mathcal{N} \to \mathcal{T}_{\mathrm{a}}, \tag{7}$$

mapping each note in the musical score $\mathcal{N}$ to its corresponding onset time on the audio timeline $\mathcal{T}_{\mathrm{a}}$. In this work, we

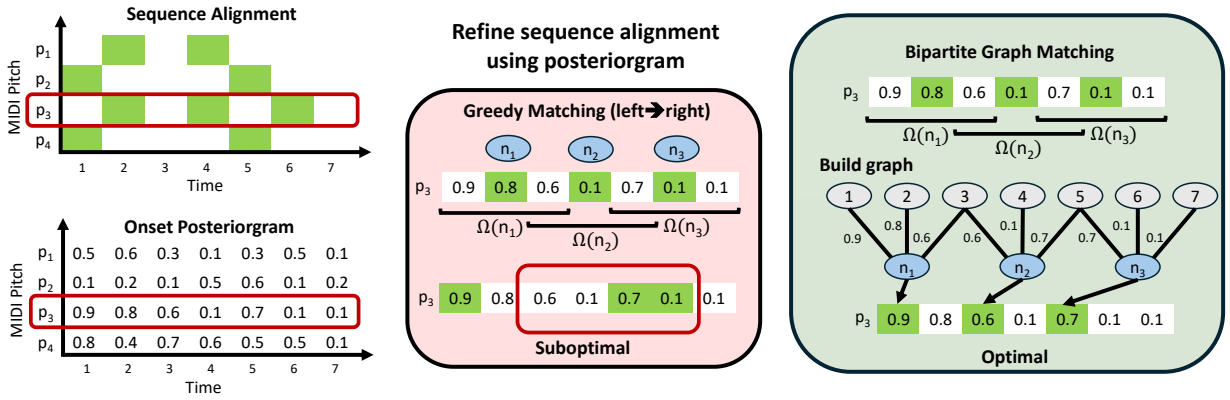


Figure 2. Snapping using greedy matching, compared to snapping based on bipartite graph matching.

focus on onset detection and consider only the onset component of each note, although a more general note-level alignment could also include note offsets.

A sequence-level alignment $\mathcal{W}$ induces an onset-level alignment $\mathcal{M}^{\text{seq}} : \mathcal{N} \to \mathcal{T}_{\text{a}}$ defined by:

$$\mathcal{M}^{\text{seq}}(n) = \mathcal{M}^{\text{seq}}((p, t^{\text{on}}, t^{\text{off}}, i)) = \mathcal{W}(t^{\text{on}}). \quad (8)$$

When aligning a musical score to an audio recording of a performance of the same score, a good sequence-level alignment does not necessarily guarantee a good onset-level alignment. For example, if two note onsets occur within the same frame on the score timeline but in different frames on the audio timeline (as in arpeggios or similar expressive timing variations), then any onset-level alignment induced directly from the sequence-level alignment will necessarily assign an incorrect onset time to at least one of the notes.

Denoting the true (latent) onset-level alignment by

$$\mathcal{M}^* : \mathcal{N} \to \mathcal{T}_{\text{a}}, \quad (9)$$

and the onset-level alignment induced by the sequence-level alignment by $\mathcal{M}^{\text{seq}}$, we assume in the following the existence of a uniform error bound B such that

$$|\mathcal{M}^{\text{seq}}(n) - \mathcal{M}^*(n)| \leq B, \qquad \text{for all } n \in \mathcal{N}. \quad (10)$$

Improved sequence-level alignment results in a smaller B.

We refer to *snapping* as the process of refining a sequence-level alignment into an onset-level alignment, i.e., estimating $\mathcal{M}^*$ from $\mathcal{M}^{\text{seq}}$, by incorporating additional information from a note-onset posteriorgram.

3.2 Snapping as Bipartite Graph Matching

We formulate *snapping* as a *bipartite graph-matching* problem, applied independently to each pitch, as illustrated in Figure 2 (right). For each pitch $p \in \mathcal{P}$, let $\mathcal{N}_p \subseteq \mathcal{N}$ denote the subset of notes whose pitch is p. We define

$$\mathcal{M}_p^* = \mathcal{M}^*|_{\mathcal{N}_p} : \mathcal{N}_p \to \mathcal{T}_{\text{a}}. \quad (11)$$

We estimate $\mathcal{M}_p^*$ with an onset-level alignment

$$\mathcal{M}_p : \mathcal{N}_p \to \mathcal{T}_{\text{a}}, \quad (12)$$

and subsequently unify these pitch-wise mappings to obtain a single onset-level alignment over the full note set,

$$\mathcal{M} : \mathcal{N} \to \mathcal{T}_{\text{a}}, \quad (13)$$

that estimates $\mathcal{M}^*$.

We assume that for each $p \in \mathcal{P}$ both the audio timeline and the score timeline are much longer than the number of occurrences $\mathcal{N}_p$, i.e., $T_{\text{a}}, T_{\text{s}} \gg |\mathcal{N}_p|$.

Although $\mathcal{M}_p^*$ is unknown, we assume access to a coarse onset-level alignment $\mathcal{M}^{\text{seq}} : \mathcal{N} \to \mathcal{T}_{\text{a}}$ induced by a sequence-level alignment (Figure 2, top left), which is locally accurate within a bounded temporal deviation B, independent of p, as in Equation 10.

We further assume the existence of a posteriorgram

$$f_\theta^p : \mathcal{T}_{\text{a}} \to [0, 1], \quad (14)$$

where $f_\theta^p(t)$ denotes the likelihood that an onset of pitch p occurs in the audio at time t. In practice, f_θ^p can be obtained from a neural-network-based automatic transcriber pre-trained on a related yet distinct domain.

Our goal is to estimate $\mathcal{M}_p^*$ using the prior information provided by f_θ^p and $\mathcal{M}^{\text{seq}}$; that is, we aim to refine the coarse estimation $\mathcal{M}_p^{\text{seq}} = \mathcal{M}^{\text{seq}}|_{\mathcal{N}_p}$ within the admissible windows defined by the temporal error bound B, guided by the posteriorgram f_θ^p. For each note occurrence $n \in \mathcal{N}_p$, we define the set of admissible candidate matches as

$$\Omega(n) = \{\, t \in \mathcal{T}_{\text{a}} \;\mid\; |t - \mathcal{M}^{\text{seq}}(n)| \leq B \,\}. \quad (15)$$

We refer to $\Omega(n)$ as the *snapping window*, and to B as the *snapping window size*. Since windows $\Omega(n)$ may overlap for different n, greedy per-window candidate selection (e.g., from left to right) may yield a suboptimal global matching (Figure 2, middle). An optimal matching can instead be obtained using classical bipartite graph matching algorithms [19, 20] [1] (Figure 2, right). We define a weighted bipartite graph

$$G = (V_1, V_2, E), \quad (16)$$

with vertex sets $V_1 = \mathcal{N}_p$ and $V_2 = \mathcal{T}_{\text{a}}$, and edge set

$$E = \bigcup_{n \in \mathcal{N}_p} \{\, (n, t) \mid t \in \Omega(n) \,\}. \quad (17)$$

Each edge (n, t) carries a weight $w_{n,t} = f_\theta^p(t)$, which quantifies the likelihood of matching the onset of the note event n to audio frame t. For all $t \notin \Omega(n)$, we set $w_{n,t} = 0$.

[1] https://docs.scipy.org/doc/scipy/reference/generated/scipy.sparse.csgraph.min_weight_full_bipartite_matching.html

We seek the matching

$$\widehat{\mathcal{M}_p} = \arg\max_{\mathcal{M}_p} \sum_{n \in \mathcal{N}_p} w_{n,\mathcal{M}_p(n)} \tag{18}$$

subject to the following constraints:

$$\mathcal{M}_p(n) \in \Omega(n), \quad \text{for all } n \in \mathcal{N}_p, \tag{19}$$

$$n \neq n' \implies \mathcal{M}_p(n) \neq \mathcal{M}_p(n'), \quad \text{for all } n, n' \in \mathcal{N}_p. \tag{20}$$

We solve for $\widehat{\mathcal{M}_p}$ using the algorithms of [19, 20].

As onset-level alignments obtained via snapping are used as labels for training a transcription model, the snapping window size B directly controls the strength of supervision. It can be adjusted based on the expected accuracy of the initial sequence-level alignment: smaller windows provide stronger supervision but require more precise sequence-level alignment, while larger windows tolerate greater alignment errors at the cost of weaker supervision.

4. INSTRUMENT-AGNOSTIC TRANSCRIPTION

Instrument-agnostic transcription estimates note activity without distinguishing between instruments. In this section we formalize this task, and outline the simplifying assumptions used in this work.

Starting from a note-event list (Equations 4, 5), we derive a note-onset piano roll $M^{\text{on}} \in \{0,1\}^{T_\text{s} \times P}$, where T_s is the number of timesteps and P the number of pitch bins:

$$M^{\text{on}}_{t,p} = \begin{cases} 1, & \exists\, t^{\text{off}} \in \mathcal{T}_\text{s},\, i \in \mathcal{I} : (p, t, t^{\text{off}}, i) \in \mathcal{N}, \\ 0, & \text{otherwise.} \end{cases} \tag{21}$$

Thus, $M^{\text{on}}_{t,p} = 1$ if and only if there exists a note of pitch p that starts at time t. In this work we focus on onset detection, though a note-activity piano roll considering note duration could be defined similarly.

Because instrument labels are discarded, notes from different instruments with the same pitch and onset time are merged into a single active entry in M^{on}. This is an inherent property of the instrument-agnostic formulation.

Our objective is to train a transcriber to directly predict from audio the piano roll corresponding to the underlying (latent) note-event list of the performance.

Training data are obtained by aligning score-derived piano rolls with real performances. In multi-instrument recordings, inconsistencies may arise when notes of the same pitch simultaneous in the score occur at different audio frames, or when notes from different score timesteps coincide in audio. Focusing on chamber music and piano, we assume such cases are rare and do not substantially affect training. They may be more prominent in orchestral settings; however, investigating this phenomenon is left for future work.

5. EXPERIMENTS

In this section, we present our experiments. We begin by introducing the datasets used for training and evaluation (Section 5.1). We then describe our approach for evaluating transcription and alignment accuracy (Section 5.2), followed by cross-dataset transcription evaluation (Section 5.3) and alignment accuracy evaluation (Section 5.4).

5.1 Datasets

Below, we describe the datasets used in this work. Importantly, MusicNet [13] is the *only* dataset used for training; all other datasets—including MAESTRO [11] which provides its own training split—are used solely for evaluation.

MusicNet [13] is a 34-hour dataset of 11 instruments, including multi-instrument chamber music ensembles of 1–8 players. Its main advantage is acoustic diversity and exclusive use of professional recordings. Its main limitation is low alignment accuracy of score annotations, making them insufficient for training transcription systems. We correct annotation timing errors using snapping, which we show to be highly effective.

MAESTRO [11] is a 140-hour dataset of solo piano performances recorded on a Disklavier. It provides highly accurate onset and offset annotations, but is limited in acoustic diversity. We use only the 20-hour test set for evaluation, disregarding the training set.

Saarland Music Data (SMD) [21] is a 5-hour dataset of solo piano performances recorded on a Disklavier, similar to MAESTRO but offering different acoustic conditions.

URMP [22] is a multi-instrument, multi-track chamber music dataset totaling 80 minutes. Each musical piece comprises up to five instrumental tracks. Its isolated monophonic recordings enable relatively accurate onset annotations (although not as precise as those of a Disklavier). Nevertheless, the dataset is small and acoustically uniform.

ChoraleBricks [23] is a multi-track wind music dataset consisting of four-part chorales, created using methods similar to those employed for URMP. The total duration of all pieces is less than 7 minutes. However, each of the four parts in every piece is recorded with multiple instruments, which gives the possibility of creating many combinations of instruments for each chorale. For our work, we use all possible four-part combinations, resulting in over 52 hours.

PHENICX [24] is an orchestral multi-track dataset created similarly to URMP and ChoraleBricks, but featuring full orchestral performances and a total duration of just over 10 minutes. Each piece comprises between 10 and 40 instrumental tracks. Like URMP, onset annotations are relatively accurate, but the dataset lacks acoustic diversity due to its uniform recording environment.

The Beethoven Symphony Excerpts Dataset (BSED) is an internal orchestral evaluation dataset with a total duration of 37 minutes, created by aligning musical scores with corresponding audio recordings. Its primary advantages are its acoustic diversity and the professional quality of the recorded performances. As separated tracks are not available, annotations were produced using audio–score alignment techniques [1, 2].

5.2 Evaluating Alignment Accuracy

There are two general approaches to evaluating alignment accuracy. The first is direct, comparing the estimated alignment to a ground-truth reference. The second is indirect, measuring how alignment affects transcription performance when aligned scores serve as training labels. The direct approach is explicit but less practical, as reference alignments at millisecond precision are rare. The indirect approach, though less explicit, is more scalable and provides meaningful evaluation via its effect on transcription metrics.

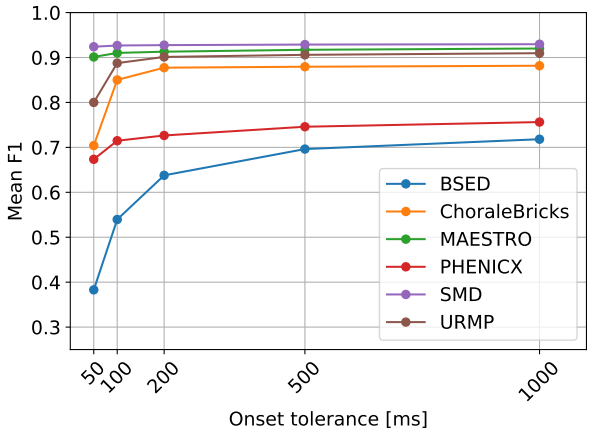


Figure 3. Note-level F1 score as a function of onset tolerance threshold, using the `DTW-BiP` (0.64s) model.

Accordingly, we adopt the indirect approach as our primary evaluation, measuring alignment quality through its effect on transcription performance. To complement this, we also perform a direct evaluation in a controlled setting. Our experiments correspond to the two approaches:

(i) **Cross-dataset transcription (Section 5.3):** We compare how different alignment policies used in training data annotation affect transcription performance.

(ii) **Direct alignment evaluation (Section 5.4):** Using the MAESTRO dataset which has high-accuracy reference alignments, we test alignment quality by attempting to recover the ground truth from intentionally perturbed labels.

5.2.1 Metrics

Our evaluation metrics consist of note-level precision, recall, and F1 score, where a predicted note is counted as correct if its onset is within a temporal threshold of the reference onset timing. In the following section (5.2.2) we elaborate on the exact threshold used.

5.2.2 Handling Evaluation Set Misalignment

It is important to recognize that some datasets may contain timing inaccuracies in their annotations, particularly those not recorded with specialized hardware such as a Disklavier. To prevent distorted evaluation, we adjust the onset tolerance according to the expected annotation quality.

For the piano datasets MAESTRO and SMD, the Disklavier provides temporal precision on the order of 3 ms. Consequently, for these datasets we adopt the standard 50 ms onset-tolerance threshold.

In the strings and winds datasets—URMP, ChoraleBricks, PHENICX, and BSED [2] —note-onset and offset annotations may be misaligned. In principle, these misalignments could be corrected using snapping; however, because snapping is precisely the method we aim to evaluate, doing so would bias the results. To accommodate small alignment errors, we instead apply a 100 ms onset tolerance threshold for URMP, ChoraleBricks, and PHENICX.

[2] In addition to sequence-level alignments, BSED also provides accurate note-level alignments obtained using snapping. Since this evaluation investigates snapping itself, we intentionally do not use them. Furthermore, for similar reasons, we use sequence-level alignments derived from signal-processing features rather than neural transcription features.

As shown in Figure 3, this results in a substantial increase in F1 for URMP, ChoraleBricks, PHENICX, whereas the impact on MAESTRO and SMD—whose annotations are highly accurate—is minimal. This pattern suggests that lower performance under tight thresholds often reflects annotation errors rather than transcription errors. We therefore consider 100 ms a reasonable approximation of typical human annotation error for these datasets.

BSED labels are obtained via DTW-based sequence alignment [2] on full polyphonic mixes, which can introduce larger timing errors. To account for sequence-level alignment inaccuracies, which may reach 0.5–1 seconds as shown in the following sections, we use a tolerance threshold of 500 ms. Figure 3 supports this choice: whereas in the multitrack datasets most of the F1 increase occurs when raising the tolerance from 50 ms to 100 ms, the F1 scores for BSED increase substantially up to a 500 ms tolerance.

5.3 Cross-dataset Transcription: MusicNet

In this section we focus exclusively on cross-dataset evaluation to better reflect real-world conditions: All compared models are trained on MusicNet, and tested on MAESTRO, SMD, URMP, ChoraleBricks, PHENICX, and BSED. All models are initialized from the same instrument-agnostic synthetic pre-training, released by [6].

We report results in order of increasing ensemble complexity. We begin with piano transcription (Section 5.3.2), proceed to instrument-agnostic transcription for small string and wind ensembles (Section 5.3.3), and finally address instrument-agnostic orchestral transcription (Section 5.3.4).

5.3.1 Compared Models

We compare models trained in an EM manner [4] under different alignment strategies varying in snapping algorithm, window size, and initial sequence alignment. We consider two snapping policies: Our proposed bipartite graph matching (`BiP`) and greedy snapping (`Gre`), commonly used in prior work. Snapping windows range from 0.1 s to 60 s.

Two types of initial sequence-level alignments are considered: Dynamic Time Warping based on chroma and onset features [1] (`DTW`) and linear stretching of the score to the audio timeline (`LS`). For `DTW`, snapping windows range from 0.1 s to 2 s. Since `LS` can introduce substantial alignment errors, it requires larger snapping windows to compensate; accordingly, we use windows of 2 s to 60 s.

For example, `DTW-BiP` (0.64s) denotes bipartite graph-based snapping with 0.64-second windows applied after DTW, whereas `LS-Gre` (60s) denotes greedy snapping with 60-second windows applied after linear stretching.

We also compare our method to a recent top-K peak-picking histogram-based approach [6] (`Hist`).

Finally, `Synth` refers to the instrument-agnostic, synthetically pre-trained model released by [6], from which all compared models are initialized.

5.3.2 Piano Transcription

Table 1 presents piano transcription results, training on MusicNet and evaluating on MAESTRO and SMD.

As shown, `DTW` alone produces poor labels, substantially reducing F1 compared to the `Synth` baseline (e.g., MAESTRO: 84.7% $\rightarrow$ 62.1%). Applying snapping after `DTW`

Transcriber	MAESTRO			SMD		
	P	R	F1	P	R	F1
Synth	88.4	81.6	84.7	93.0	85.6	88.9
DTW	96.6	48.6	62.1	96.1	56.0	69.0
Snapping: Previous Work [4, 6]						
DTW-Gre (2s)	94.0	84.8	89.0	97.5	87.4	92.1
DTW-Gre (0.64s)	93.8	85.2	89.2	97.5	87.2	91.9
DTW-Gre (0.1s)	95.8	84.5	89.6	98.1	85.9	91.4
LS-Gre (60s)	92.8	79.0	85.2	95.1	84.5	89.3
Hist	94.6	86.0	89.9	97.6	88.1	92.5
Snapping: Ours						
DTW-BiP (2s)	94.4	**86.6**	**90.2**	97.9	**88.2**	**92.7**
DTW-BiP (0.64s)	95.0	85.9	90.1	**98.2**	87.5	92.4
DTW-BiP (0.1s)	**96.3**	84.6	89.9	98.1	86.9	92.0
LS-BiP (60s)	93.1	84.2	88.3	96.9	87.6	91.9
LS-BiP (20s)	94.5	83.1	88.2	97.8	86.6	91.7
LS-BiP (10s)	94.4	80.4	86.6	97.7	85.7	91.1
LS-BiP (2s)	94.5	52.4	65.1	96.1	64.0	74.4

Table 1. Piano transcription results. Models are trained on MusicNet and evaluated on MAESTRO and SMD.

Transcriber	URMP			ChoraleBricks		
	P	R	F1	P	R	F1
Synth	77.6	78.0	77.5	87.5	72.7	79.1
DTW	97.1	62.9	75.5	97.3	53.9	68.6
Snapping: Previous Work [4, 6]						
DTW-Gre (2s)	91.2	84.5	87.5	93.3	77.0	83.9
DTW-Gre (0.64s)	93.0	85.5	89.0	94.2	80.1	86.2
DTW-Gre (0.1s)	**94.2**	83.7	88.4	**95.2**	77.3	84.9
LS-Gre (60s)	83.3	78.4	80.3	81.1	78.4	78.9
Hist	88.7	85.7	87.1	90.1	79.0	83.7
Snapping: Ours						
DTW-BiP (2s)	90.8	**86.8**	88.6	91.7	**82.4**	86.5
DTW-BiP (0.64s)	92.8	86.3	**89.3**	93.4	82.2	**87.0**
DTW-BiP (0.1s)	93.4	85.2	88.9	94.9	77.5	84.8
LS-BiP (60s)	87.3	81.4	84.0	89.1	73.0	79.2
LS-BiP (20s)	85.8	83.5	84.4	89.1	75.6	80.9
LS-BiP (10s)	90.1	81.1	85.2	92.6	70.1	78.7
LS-BiP (2s)	88.8	72.9	79.2	91.9	63.6	74.6

Table 2. Transcription results on small multi-instrument ensembles. Models are trained on MusicNet and evaluated on URMP and ChoraleBricks.

yields large gains (∼23–28%), increasing MAESTRO F1 to 89–90%. Relative to Synth, snapping provides a ∼3–5% improvement (84.7% → 90.2% with DTW-BiP, 0.64 s), demonstrating effective domain adaptation with weakly aligned data. These results show that snapping is essential for training with weakly aligned labels and that it makes such training beneficial.

After showing snapping to be essential, we compare different snapping methods. Our graph-based approach yields marginal yet consistent improvements of approximately 0.5–1.2% across models and datasets over greedy snapping used in prior work [4]. For example, on MAESTRO, using a 2 s window, F1 increases from 89.0 (DTW-Gre 2s) to 90.2 (DTW-BiP 2s). We further observe that the recent top-K peak picking histogram-based method (Hist) slightly outperforms greedy snapping. For example, on SMD, the F1 score increases from 91.9 (DTW-Gre 0.64s) to 92.5 (Hist). We hypothesize that this is because the top-K method does not suffer from issues caused by overlapping snapping windows. However, graph-based snapping (e.g., DTW-BiP 2s) still slightly outperforms the histogram top-K method. Although the difference is small for piano transcription, we will see that the gap becomes larger for strings and winds, reaching up to 5% (Sections 5.3.3, 5.3.4).

Next, we analyze the effect of snapping window size. A 0.1 s window already substantially outperforms the Synth baseline and DTW, and larger windows yield further gains, mainly through higher recall. On SMD, increasing the window from 0.1 s to 2 s raises recall and F1 from $86.9/92.0$ to $88.2/92.7$ (DTW-BiP 2s).

Finally, we examine extremely large windows. With linear stretching, small windows fail (F1 on MAESTRO degrades from 84.7 to 65.1 with LS-BiP 2 s). Compensating with 60-second windows, greedy snapping (LS-Gre 60 s) yields minimal gains if any at all compared to Synth (F1 $84.7 \rightarrow 85.2$, recall $81.6 \rightarrow 79.0$), whereas graph-based (LS-BiP 60 s) substantially improves both recall and F1 (F1 $84.7 \rightarrow 88.3$, recall $81.6 \rightarrow 84.2$). This shows that as the window size grows, greedy snapping degrades due to increased overlaps, while graph-based snapping remains robust.

5.3.3 Strings & Winds Transcription—Small Ensemble

Table 2 presents results for *instrument-agnostic* (multi-instrument) transcription on small musical ensembles. The compared models were trained on MusicNet and evaluated on URMP (chamber music with strings and winds) and ChoraleBricks (small wind ensembles).

Similar trends are observed to those seen in piano transcription: DTW labels degrade performance relative to the Synth baseline. For example, F1 on ChoraleBricks drops from 79.1 to 68.6. Applying snapping after DTW substantially improves F1. For example, DTW-BiP with a 0.64 s window increases F1 relative to Synth from 77.5 to 89.3 on URMP and from 79.1 to 87.0 on ChoraleBricks.

Graph-based snapping outperforms both greedy snapping and the histogram-based method. With 2 s windows, graph-based snapping (DTW-BiP 2s) improves F1 over greedy snapping (DTW-Gre 2s) from 83.9 to 86.5 on ChoraleBricks and from 87.5 to 88.6 on URMP. Compared to Hist, F1 increases from 83.7 to 86.5 on ChoraleBricks and from 87.1 to 88.6 on URMP. Larger windows amplify the benefit of graph-based snapping; on URMP, LS-BiP 60s improves F1 relative to LS-Gre 60s from 80.3 to 84.4.

Overall, results on URMP slightly exceed those on ChoraleBricks, possibly due to the higher proportion of string instruments in MusicNet and the higher annotation precision of URMP; increasing the onset tolerance from 100 ms to 200 ms raises F1 by ∼3% on ChoraleBricks, but only by ∼1% on URMP (Figure 3).

Compared to piano results (Table 1), performance on URMP and ChoraleBricks falls within a similar range, indicating that onset detection can be effective for string and wind instruments despite their less pronounced onsets.

Transcriber	PHENICX			BSED		
	P	R	F1	P	R	F1
Synth	83.8	54.8	66.0	83.2	39.5	51.3
DTW	93.7	41.9	57.2	89.3	15.4	24.3
Snapping: Previous Work [4, 6]						
DTW-Gre (2s)	85.8	60.0	70.4	81.6	59.9	67.2
DTW-Gre (0.64s)	85.5	60.3	70.4	78.9	62.4	67.6
DTW-Gre (0.1s)	87.1	60.6	71.2	83.9	53.4	62.3
LS-Gre (60s)	78.7	62.2	68.8	72.3	59.1	62.4
Hist	86.3	60.7	71.2	74.8	58.4	64.1
Snapping: Ours						
DTW-BiP (2s)	86.3	61.5	71.5	79.7	62.2	67.8
DTW-BiP (0.64s)	84.4	**62.3**	71.5	81.4	**63.8**	**69.6**
DTW-BiP (0.1s)	**87.6**	60.9	**71.7**	**84.4**	53.9	63.4
LS-BiP (60s)	85.1	61.4	71.0	72.6	62.2	64.7
LS-BiP (20s)	84.9	60.7	70.5	73.3	62.0	64.9
LS-BiP (10s)	85.4	60.3	70.3	73.3	61.3	64.5
LS-BiP (2s)	85.6	36.4	50.8	74.1	47.4	55.0

Table 3. Orchestral transcription results. Models are trained on MusicNet and evaluated on BSED and PHENICX.

5.3.4 Orchestral Transcription

Table 3 shows orchestral transcription results, traning on MusicNet and evaluating on PHENICX and BSED. As with piano and small ensembles, DTW alone is insufficient, while snapping (DTW-BiP, DTW-Gre) improves alignment, with graph-based snapping consistently outperforming greedy snapping. For example, DTW-BiP 0.64s raises F1 on PHENICX compared to DTW-Gre 0.64s from 70.4 to 71.5 and on BSED from 67.6 to 69.6. Improvement on BSED compared to Hist is substantial: From 64.1 to 69.6.

Overall metrics are lower than for piano or small-ensemble settings. For example, DTW-BiP (0.64 s) achieves an F1 score of 89.3 on URMP but only 71.7 on PHENICX. We attribute this gap to increased orchestral complexity: Performances in PHENICX involve 10–40 instruments, while URMP ensembles include no more than five.

5.4 Alignment Evaluation—MAESTRO

We evaluate alignment accuracy on the MAESTRO dataset, for which precise reference annotations are available. We randomly and uniformly perturb the MAESTRO onset labels within a window $[-w, w]$, where $w \in \{1, 5, 15, 60\}$ frames, corresponding to $0.032\,\text{s}$, $0.16\,\text{s}$, $0.48\,\text{s}$, and $1.92\,\text{s}$. The perturbed onsets are then recovered using snapping with the same window size. Alignment quality is measured by the F1 score with respect to the ground truth.

We compare three posteriorgrams obtained from the following: (i) the synthetically pre-trained model (Synth); (ii) the DTW-BiP 0.64 s model fine-tuned from Synth on MusicNet with weakly-aligned labels using snapping (MN); and (iii) the ground-truth piano roll (GT) as a sanity check.

Results are shown in Table 4. The GT posteriorgram achieves an F1 score of 100% for all w, for both greedy and graph-based snapping, confirming that snapping operates correctly with ideal posteriorgrams.

Method	w [s]	Pert	Synth	MN	GT
Gre	0.032	100	95.9	96.4	100
	0.16	30.1	92.5	93.0	100
	0.48	16.7	88.6	89.9	100
	1.92	9.6	84.7	87.1	100
BiP	0.032	100	95.9	96.4	100
	0.16	30.1	93.6	94.3	100
	0.48	16.7	91.0	92.8	100
	1.92	9.6	87.8	91.1	100

Table 4. F1 score evaluation of snapping applied to the perturbed MAESTRO test set (Pert) across varying perturbation windows w, using different onset posteriorgrams (Synth, MN, GT).

With Synth, F1 decreases as w increases, but graph-based snapping (BiP) still remains robust, achieving F1 of 91.0 at $w = 0.48\,\text{s}$ and 87.8 at $w = 1.92\,\text{s}$, compared to F1 scores of 16.7 and 9.6 for the perturbed labels (Pert). The differences between graph-based and greedy snapping grow with w: 93.6–92.5 = 1.1 for $w = 0.16\,\text{s}$, 91.0–88.6 = 2.4 for $w = 0.48\,\text{s}$, and 87.8–84.7 = 3.1 for $w = 1.92\,\text{s}$. This can be expected: larger windows increase overlap, making greedy snapping less optimal.

The MusicNet-trained model (MN) shows similar trends. However, it has consistently improved performance across snapping methods and window sizes compared to Synth; for example, at $w = 0.48\,\text{s}$ with graph-based snapping (BiP), F1 increases from 91.0 (Synth) to 92.8 (MN). This demonstrates that training with snapping on weakly-aligned data not only improves transcription metrics (as seen in Section 5.3), but also enhances alignment accuracy when using transcription features for snapping.

Finally, comparing with piano transcription results in Table 1, we find that the difference between greedy and graph-based snapping is larger for alignment than for transcription (up to 4% versus about 1%). This may indicate that the training process is robust to some degree of label noise, provided that a large portion of onsets is accurately aligned.

6. CONCLUSION

In this work, we investigated *snapping*—the refinement of sequence-level alignments into precise onset-level alignments using neural onset posteriorgrams. Through cross-dataset evaluations and controlled experiments, we showed that snapping is highly effective for onset-level alignment, enabling training transcribers with weakly-aligned labels with further gains when following a graph-based approach. Although we focused on instrument-agnostic AMT, the methods used may be extended to instrument-sensitive transcription, and possibly other timing-critical MIR tasks such as drum transcription or multi-pitch estimation.

Acknowledgments

This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Grant No. 500643750 (MU 2686/15-1). The International Audio Laboratories Erlangen are a joint institution of the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) and Fraunhofer Institute for Integrated Circuits IIS.

7. REFERENCES

[1] S. Ewert, M. Müller, and P. Grosche, "High Resolution Audio Synchronization Using Chroma Onset Features," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2009, pp. 1869–1872.

[2] M. Müller, Y. Özer, M. Krause, T. Prätzlich, and J. Driedger, "Sync Toolbox: A Python Package for Efficient, Robust, and Accurate Music Synchronization," *Journal of Open Source Software (JOSS)*, vol. 6, no. 64, pp. 3434:1–4, 2021.

[3] J. Zeitler, B. Maman, and M. Müller, "Robust and Accurate Audio Synchronization Using Raw Features from Transcription Models," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2024.

[4] B. Maman and A. H. Bermano, "Unaligned Supervision for Automatic Music Transcription in The Wild," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2022, pp. 14 918–14 934.

[5] X. Riley, D. Edwards, and S. Dixon, "High Resolution Guitar Transcription Via Domain Adaptation," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2024, pp. 1051–1055.

[6] J. Yaffe, B. Maman, M. Müller, and A. Bermano, "Count The Notes: Histogram-Based Supervision for Automatic Music Transcription," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2025.

[7] N. Hu and R. B. Dannenberg, "Bootstrap learning for accurate onset detection," *Machine Learning*, vol. 65, no. 2-3, pp. 457–471, 2006.

[8] E. Benetos, S. Dixon, Z. Duan, and S. Ewert, "Automatic Music Transcription: An Overview," *IEEE Signal Processing Magazine*, vol. 36, no. 1, pp. 20–30, 2019.

[9] P. Smaragdis and J. C. Brown, "Non-Negative Matrix Factorization for Polyphonic Music Transcription," in *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2003, pp. 177–180.

[10] C. Hawthorne, E. Elsen, J. Song, A. Roberts, I. Simon, C. Raffel, J. H. Engel, S. Oore, and D. Eck, "Onsets and Frames: Dual-Objective Piano Transcription," in *Proceedings of the International Society for Music Information Retrieval Conference, (ISMIR)*, 2018, pp. 50–57.

[11] C. Hawthorne, A. Stasyuk, A. Roberts, I. Simon, C. A. Huang, S. Dieleman, E. Elsen, J. H. Engel, and D. Eck, "Enabling Factorized Piano Music Modeling and Generation with the MAESTRO Dataset," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019.

[12] Q. Kong, B. Li, X. Song, Y. Wan, and Y. Wang, "High-Resolution Piano Transcription With Pedals by Regressing Onset and Offset Times," *IEEE/ACM Transactions of Audio, Speech, and Language Processing*, vol. 29, pp. 3707–3717, 2021.

[13] J. Thickstun, Z. Harchaoui, and S. M. Kakade, "Learning Features of Music from Scratch," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

[14] X. Riley, Z. Guo, and S. Edwards, Drew abd Dixon, "GAPS: A Large and Diverse Classical Guitar Dataset and Benchmark Transcription Model," *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2024.

[15] N. C. Tamer, Y. Özer, M. Müller, and X. Serra, "High-Resolution Violin Transcription using Weak Labels," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2023, pp. 223–230.

[16] Y. Wu, B. Chen, and L. Su, "Multi-Instrument Automatic Music Transcription with Self-Attention-Based Instance Segmentation," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 28, pp. 2796–2809, 2020.

[17] J. Gardner, I. Simon, E. Manilow, C. Hawthorne, and J. H. Engel, "MT3: Multi-Task Multitrack Music Transcription," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022.

[18] Y. Wu, W. Wei, D. Li, M. Li, Y. Yu, Y. Gao, and W. Li, "Harmonic Frequency-Separable Transformer for Instrument-Agnostic Music Transcription," in *2024 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2024, pp. 1–6.

[19] R. M. Karp, "An algorithm to solve the m× n assignment problem in expected time O (mn log n)," *Networks*, vol. 10, no. 2, pp. 143–152, 1980.

[20] R. Jonker and A. Volgenant, "A shortest augmenting path algorithm for dense and sparse linear assignment problems," *Computing*, vol. 38, no. 4, 1987.

[21] M. Müller, V. Konz, W. Bogler, and V. Arifi-Müller, "Saarland Music Data (SMD)," in *Demos and Late Breaking News of the International Society for Music Information Retrieval Conference (ISMIR)*, 2011.

[22] B. Li, X. Liu, K. Dinesh, Z. Duan, and G. Sharma, "Creating a Multitrack Classical Music Performance Dataset for Multimodal Music Analysis: Challenges, Insights, and Applications," *IEEE Transactions on Multimedia*, vol. 21, no. 2, pp. 522–535, 2019.

[23] S. Balke, A. Berndt, and M. Müller, "ChoraleBricks: A Modular Multitrack Dataset for Wind Music Research," *Transaction of the International Society for Music Information Retrieval (TISMIR)*, vol. 8, no. 1, pp. 39–54, 2025.

[24] C. C. S. Liem, E. Gómez, and M. Schedl, "PHENICX: Innovating the Classical Music Experience," in *Proceedings of the IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, 2015, pp. 1–4.

Spores: A Physarum-Inspired Instrument for Agent-Based Ecological Interaction

Kyle Braden Smith
Georgia Institute of Technology
ksmith683@gatech.edu

Alexandria Smith
Georgia Institute of Technology
alexandria.smith@gatech.edu

ABSTRACT

The ecosystem is the interface. Spores (2025) is a touchscreen instrument where sound is activated based on simulated slime mold colonies finding food. The performer becomes part of the ecosystem and a caretaker alongside the slime mold. They read health, growth rate, stress, and territorial spread through observable organism behavior, the way one reads an animal's body language. The performer distributes resources rather than issuing commands, acting as caretaker to a system that resists mastery.

In this paper, I discuss the theoretical underpinnings engaging with ecological systems, biomimetic systems, multi-agent systems in composition/instrument design, technical implementation, and composing for and performing with living environments where the performer becomes the care taker of an environment instead of the "commander" of the environment. I discuss incorporating biophilic design practices into working with agent-based models (ABM) and artificial intelligence in music, modes of interaction, and my biologically inspired process of collaborating with artificial intelligence. A seven-minute event-based improvisation demonstrates this approach across six sections exploring hunger, competition, and abundance.

1. INTRODUCTION

Spores (2025) is a touchscreen instrument and event-based score where sound depends on simulated slime mold colonies finding food. The simulation presents the following ecological metrics: health, growth rate, stress, territorial spread—as observable organism behavior. The performer reads colony states intuitively, the way one reads an animal's body language, through attention to animate form. Wilson's biophilia hypothesis suggests humans attend innately to life and life-like processes [1]; Spores exploits this tendency by representing synthesis parameters as organism survival. The constraint is visible before it is audible: no food, no sound.

Biomimetic systems have been used in sonification, but typically as passive data sources, or algorithms that model natural phenomena. Spores treats organism behavior as the interface design material. The simulation is not a data stream to be sonified; it is the control surface itself. This approach enables performers to learn slime mold dynamics through sound. Technique emerges from learning the conditions that sustain colonies, how competition fragments growth, and when resources become scarce. The instrument does not just represent or mimic Physarum, it makes its survival logic audible and interactive.

Spores challenges one-to-one mapping in instrument design by using ecological dynamics as the control metaphor. The performer is a part of a larger ecological system where they act as a caretaker and/or observer of environmental forces rather than trying to control the behavior of the instrument. The goal of the performance is to be a supporting figure in a larger ecosystem over controlling the desired sonic output.

This approach challenges F. Richard Moore's notion of 'control intimacy,' the match between an instrument's sonic variety and a performer's psychophysiological capabilities [2]. Wessel and Wright (2002) further argued that control intimacy depends on "compelling control metaphors" that ground abstract parameters in sensorimotor experience [3]. The most ubiquitous metaphors are geometric: XY pads for navigating timbral space, sliders and knobs for adjusting parameters. These models position control intimacy between performer and sound. Spores, however, blurs that relationship, placing an ecological system between the performer and the audio output.

This framing also engages questions familiar from work on agency in live electronics [4] and interactive systems [5]: how performers develop expertise with systems that resist mastery, and how audiences attribute outcomes across human and nonhuman actors.

In this paper, I present Spores: the current system architecture, a seven-minute event-based improvisation, and the performance practice it requires. Section 2 situates this work within instrument design, Physarum sonification, and multi-agent composition. Section 3 describes the piece. Section 4 details the system. Sections 5–6 address performance practice and open questions.

2. RELATED WORK

2.1 Indeterminacy in instrument design

Unpredictable behavior has served as a compositional resource in electronic instrument design since the mid-20th century. Tudor's Rainforest IV (1973) presented an electroacoustic environment where suspended sculptural objects vibrated in response to audio signals and dancer interaction, shaping sound through a complex network of material resonance [6]. Buchla's Source of Uncertainty module introduced voltage-controlled randomness as a new musical design primitive built directly into hardware [7]. Alternatively, Spores extends this lineage by exposing inde-

Copyright: ©2026 Kyle Braden Smith et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License 3.0 Unported, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

terminacy through simulated ecological dynamics: unpredictability emerges from simple, biomimetic agent-based intelligence rather than system complexity, or injected randomness.

2.2 Physarum as sonic material

Physarum polycephalum is a plasmodial slime mold—a single multinucleate cell that forms efficient transport networks while foraging. In many ways researchers argue it exhibits fundamental computational behavior. Tero et al. (2010) demonstrated that in certain controlled environments Physarum's intelligence results in networks that even rival the complexity, efficiency and tolerance of human-designed infrastructure [8].

Because of this, Physarum polycephalum has been previously identified and explored as a novel source of data for sonification [9]. Miranda, Adamatzky, and Jones attached sensors to living slime mold, converting bioelectric signals into audio [10]. Instead of using the slime mold as a passive source of data, Spores instead simulates Physarum dynamics using Jones's particle model [9]. In Jones's model, agents sample pheromone concentration at three forward sensors and rotate toward the strongest signal, depositing trail as they move to form emergent network structures. This enables real-time performer interaction and reproducible parameter tuning which is simply impossible with living specimens.

2.3 Multi-agent systems in composition

There are many examples of contemporary compositional systems that have distributed agency beyond the solo performer. Two works that have greatly influenced this project are Arne Eigenfeldt's *Coming Together: Shoals* and Özcan and Çamcı's *Proprius*. Eigenfeldt's frames real-time composition as interaction between musical agents sharing compositional control [11]. In *Proprius*, Özcan and Çamcı extended agency to listeners, whose spatial position in a virtual environment influenced the sonification of predator-prey population dynamics [12].

Both of these previous works prioritize system agency and map agent-based system dynamics to sonic output. In Spores, distinct colonies exhibit self preservation, competing for territory and nutrients with other colonies while the performer distributes resources rather than issuing commands. Combining real-time ecological simulation with per-colony MIDI Polyphonic Expression (MPE) control, the system foregrounds a performer-as-caretaker interaction which results in a constraint-based structure where silence is the default state.

3. SPORES

Spores is an event-based improvisation environment for a solo performer, where the performer takes care of the slime mold ecosystem for a fixed amount of time (for this performance, seven minutes). The performer is given a set of care instructions for the slime mold colony and performs them in real time. The premise: without food, the slime mold colonies cannot persist. The sonic output is a result of the health and/or vitality of the colony, thus if they are not fed or supported, it results in no sound.

The performer seeds, tends, and disturbs the virtual 2D environment resulting in six sections of emergent behavior through exploring care and understanding of the ecological system's constraints. These three actions, seeding, tending, and disturbing, frame the performer's relationship to the ecosystem as one of creation, sustenance, and disruption, with each section isolating a different ecological condition (see Figure 1).

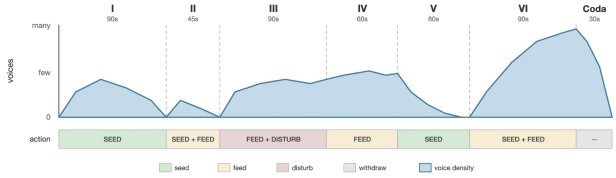


Figure 1. Compositional timeline for *Spores*. Sections I–VI and Coda with approximate durations. The shaded region indicates relative voice density; the bottom row shows primary performer actions per section.

I. Seeds Without a World. (∼90s) Colonies are seeded without any available food. Health depletes over time, voices thin, fade, and silence follows.

II. Futility. (∼45s) A single food node is placed beyond reach of the seeded colonies. Colonies sense the gradient, move toward it, and fail to arrive before death. Brief tones followed by silence.

III. Framework. (∼90s) A lattice of food points is placed by the performer. Sustained drone with timbral shifts as health fluctuates across colonies. Disturbance from the performer fragments some colonies, while others continue to persist.

IV. Two Sides / Contest. (∼60s) Food is placed on opposing ends of the virtual environment. Colonies grow along paths toward the center. Interaction between individual colonies triggers spike stress metrics, increasing detuning and spread. Territorial boundaries between colonies form.

V. False Blooms. (∼60s) Small colonies are seeded far from food and previously established healthy colonies. Food nodes, now monopolized by the larger colonies, are consumed, new colonies die out, silence follows.

VI. Density Change / Last Flower. (∼90s) The user places a dense distribution of nutrients in the bass register and slowly moves up the scale. Due to dense distribution of food, multiple colonies sustain. While the layers accumulate, timbre shifts per colony, and the system crescendoes as the dense distribution of resources is consumed.

Coda. (∼30s) The performer withdraws, resources deplete, and the voices fade into stillness and silence.

4. SYSTEM

The system runs multiple Python-based agent-based models on a 960×540 pixel grid displayed on a 7-inch capacitive Thin-Film Transistor touchscreen. Twelve global ecological metrics are transmitted via OSC; in the current configuration nine are mapped to specific synthesis parameters. Each colony transmits via MPE on its own MIDI channel (1–15), with derived metrics such as health, confidence, and stress controlling timbre.

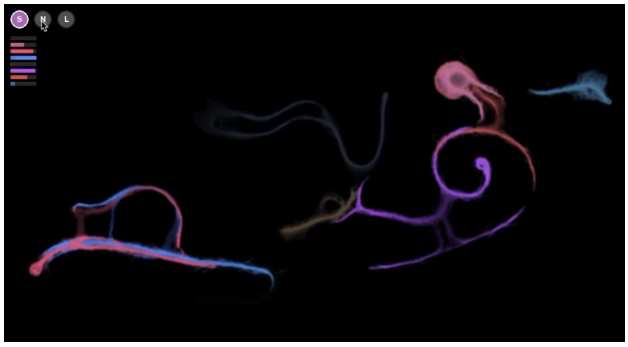


Figure 2. The *Spores* touchscreen interface displaying active slime mold colonies on a 960×540 pixel grid. Touch input interacts directly with the simulation surface via seed, nutrient, and disturb gestures.

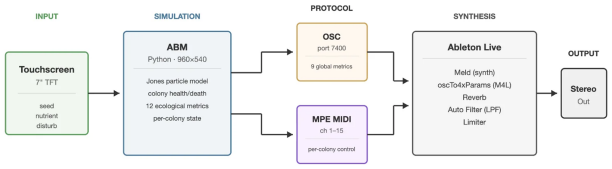


Figure 3. System architecture. Touch input controls a Python-based agent simulation. Global metrics route via OSC; per-colony control uses MPE MIDI on channels 1–15. Ableton Live handles synthesis and signal processing.

4.1 Modes of Interaction

Three input modes map touch position to simulation grid:

Seed: Tap to spawn a colony. Horizontal position determines base pitch within a Dorian scale rooted at MIDI 48 (C2).

Nutrient: Tap for discrete food nodes; drag to paint continuous nutrient corridors.

Disturb: Drag to clear locally, killing agents and erasing trail.

4.2 Agent-Based Model

The model implements Jones's (2010) particle approximation for Physarum polycephalum [9]. Each agent samples trail concentration at three forward sensors (left, right, center) separated by a configurable angle, then rotates toward the strongest signal. Agents deposit pheromone as they traverse the space acting as an attractor to other agents in the colony. Agents attract toward their own colony's trail (weight 1.8) and repel from other colonies' trails (weight −3.0). Territory forms through attraction and avoidance without explicit collision rules. Each colony's trail field does not persist forever and undergoes Gaussian diffusion and exponential decay each frame, being entirely eliminated after death. The boundaries of the display are reflective, restricting the colonies to the virtual petri-dish.

4.3 Health and Death

Colony health increases with food consumption and decreases via metabolism and foraging cost. Colonies below health threshold 0.02 enter a dying state, fade over approximately 3 seconds (180 frames at 60fps), and then disappear.

4.4 Ecological Metrics

The system computes twelve continuous ecological metrics normalized to [0,1]. Nine are currently mapped to synthesis parameters. Table 1 shows active mappings in the current configuration. Figure 4 shows how one such metric, stress, is derived from simulation state.

Metric	Target
intent_x	Pan
confidence	Drive
effort	LPF cutoff
breath_rate	LFO rate
breath_depth	LFO depth
feeding_contact	Osc A/B mix
stress	Detune / Spread
frontier	Reverb wet
density	Reverb decay / diffuse

Table 1. Nine of twelve ecological metrics mapped via OSC. Per-colony control uses separate MIDI channels.

```
def _compute_stress(self, disturbance,
                    agents, T):
    external = float(disturbance)
    fragmentation = self.\
        _compute_fragmentation_proxy(T)
    stress = 0.6*external + 0.4*fragmentation
    return float(np.clip(stress, 0.0, 1.0))
```

Figure 4. Derivation of stress metric. External disturbance (60%) and trail fragmentation (40%) are weighted and combined, then clipped to [0,1].

4.5 Per-Colony MPE Control

Each colony transmits on its own MIDI channel (1–15, round-robin):

Note On/Off: Triggered when colony is seeded and ends when colony dies or food depletes.

Pitch: Horizontal seed position → Dorian scale, root MIDI 48. The grid spans ∼3 octaves.

Pitch Bend: Population growth rate. Expanding colonies bend up (±2 semitones max), contracting bend down.

Channel Pressure: Colony health, smoothed.

CC74: 0.5 × frontier complexity + 0.5 × spread.

4.6 Signal Chain

Sound generation uses Meld, Ableton Live 12's bi-timbral macro-oscillator synthesizer, with both engines set to Basic Shapes (Engine B tuned one octave below).

Signal path: Meld → oscTo4xParams (Max for Live OSC receiver) → Sine LFO → Reverb → Auto Filter (LPF) → Limiter. Figure 5 shows the OSC routing configuration.

A video demonstration is available at: `https://vimeo.com/user252651913/spores-2025`

5. PERFORMANCE PRACTICE

Initial designs focused on using the slime mold to exclusively shape and modulate timbre while experienced electronic musicians input MIDI via a traditional keyboard controller. However, through informal practice and user testing,

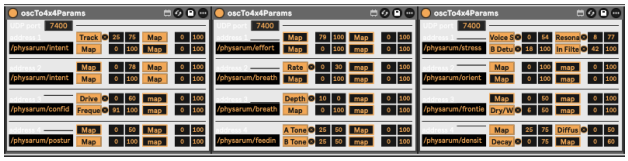


Figure 5. OSC routing in Ableton Live using oscTo4x4Params, a Max for Live device by nnirror. Each ecological metric maps to synthesis parameters via user-defined OSC addresses.

audience members could not distinguish performer-initiated pitch from system-generated variation, often attributing note events to the organism's position or shape.

The solution was to tie note events to visible colony position, triggering note on when a colony is seeded and note-off when a colony eventually dies out. This enabled feeding the system to produce sustained tones that brighten and shift as health increases. Disturbances, alternatively, are mapped to parameters that result in more dissonance, like oscillators detune and spread. The two events are distinct and differ in both pitch content and timbre. This design prioritizes audio-visual legibility specifically because the performer-as-caretaker metaphor depends on the performer and the audience being able to read the ecological source of a sound event—a transparency that would be counterproductive in acousmatic or club contexts but is central to this instrument's performative premise.

Early iterations approximated biologically realistic timing, which proved too slow for musical interaction. Parameters were tuned for responsiveness over realism. Painting gestures were added alongside single taps for nutrient input, allowing performers to draw continuous paths and exert greater influence on colony trajectory.

Skill with the system develops through spatial planning and temporal anticipation. Experienced performers learn to read trail patterns, predict colony movement, and time nutrient placement to sustain phrases. Unlike instruments where mastery means consistent reproduction, expertise here means reading system state and responding to emergent behavior. This mode of engagement aligns with situated listening frameworks that treat data not as control signals but as feedback requiring interpretation and response [13].

6. DISCUSSION

Spores reframes the performer-instrument relationship as tending rather than commanding. The display allows audiences to observe the user's interaction and more easily map the sonic output. Multimodal interaction clarifies the ecological metaphor. The compositional structure emerges where nourishment sustains and silence follows decay.

This framing raises questions familiar from work on agency in live electronics [4] and interactive systems [5]: How do performers develop expertise with systems that resist mastery? How do audiences attribute outcomes to performer skill, shared agency, and autonomous behavior? Empirical investigation with performers and audiences would help address these questions but lies beyond the present scope of our research.

Compositional challenges include calibrating the balance between silence and sonic density, determining section durations that allow ecological dynamics to emerge, and selecting mappings that remain expressive across varied performance conditions. Limitations include the fixed mapping from metrics to synthesis parameters in the current configuration, as different performers might prefer different relationships in the future. The seven-minute duration time constraint influenced its current compositional form; longer pieces would require more intuitive ecologically grounded mechanisms for recovery or reset.

7. FUTURE WORK

There are three avenues for future work. The first is extending the ecological interface to include multi-performer interaction, predator-prey dynamics, spatial audio, and more. The second is evaluating a locally trained neural cellular automata (NCA) core as replacement for the current agent based model. The NCA implementation will share the same interface and analysis layer, enabling direct comparison. The third is studying how performer-system relationships develop over extended engagement—comparing short-form performances like the current seven-minute structure with long-form sessions to understand how expertise and listening practices evolve with sustained interaction.

8. ETHICAL STATEMENT

No human subjects beyond the authors participated in evaluation. Material costs remain accessible for reproducibility.

Acknowledgments

This work was developed at the Creative Music Technology Lab, Georgia Institute of Technology.

9. REFERENCES

[1] E. O. Wilson, *Biophilia*. Cambridge, MA: Harvard University Press, 1984.

[2] F. R. Moore, "The Dysfunctions of MIDI," *Computer Music Journal*, vol. 12, no. 1, pp. 19–28, 1988.

[3] D. Wessel and M. Wright, "Problems and Prospects for Intimate Musical Control of Computers," *Computer Music Journal*, vol. 26, no. 3, pp. 11–22, 2002.

[4] J. Drummond, "Understanding Interactive Systems," *Organised Sound*, vol. 14, no. 2, pp. 124–133, 2009.

[5] O. Bown, A. Eldridge, and J. McCormack, "Understanding Interaction in Contemporary Digital Music: From Instruments to Behavioural Objects," *Organised Sound*, vol. 14, no. 2, pp. 188–196, 2009.

[6] J. Bohn, "David Tudor: Rainforest (Versions I and IV)," *Computer Music Journal*, vol. 23, no. 4, pp. 104–105, 1999.

[7] D. Buchla, "Source of Uncertainty Model 265 Schematic," Buchla & Associates, 1970.

[8] A. Tero, S. Takagi, T. Saigusa, K. Ito, D. P. Bebber, M. D. Fricker, K. Yumiki, R. Kobayashi, and T. Nakagaki, "Rules for Biologically Inspired Adaptive Network Design," *Science*, vol. 327, no. 5964, pp. 439–442, 2010.

[9] J. Jones, “Characteristics of Pattern Formation and Evolution in Approximations of Physarum Transport Networks,” *Artificial Life*, vol. 16, no. 2, pp. 127–153, 2010.

[10] E. R. Miranda, A. Adamatzky, and J. Jones, “Unconventional Computing for Music: Sound Synthesis with Slime Mould,” in *Proceedings of the International Computer Music Conference*, 2011, pp. 88–91.

[11] A. Eigenfeldt, “Real-time Composition as Performance Ecosystem,” *Organised Sound*, vol. 16, no. 2, pp. 145–153, 2011.

[12] Z. Özcan and A. Çamcı, “Compositional Considerations in the Design of an Interactive Agent-based Musical Ecosystem,” in *Proceedings of the International Computer Music Conference*, 2019.

[13] A. Smith, “Modes of Listening: Practice-Led Perspectives on Performing Biofeedback Music,” in *Proceedings of the International Computer Music Conference*, Boston, 2025, pp. 467–474.

oljud—ʇɥɹɥ(n): A Sonic Manifesto of Resistance to Generative AI in Music

Bob L. T. Sturm and Elin Kanhov
Speech, Music and Hearing (Tal, Musik, och Hörsel)
Royal Institute of Technology (Kungliga Tekniska Högskolan)
Stockholm, Sweden
{bobs, ekanhov}@kth.se

ABSTRACT

Some discourses about music and AI are frustratingly shallow and insular, raising outdated musical tropes and ignoring modern developments, flattening the rich and varied functions of music in life, and overlooking serious ethical issues with the technology (creating it, maintaining it, using it, and imposing it). We respond to this shallowness via artistic activism and agonistic artistic research, resulting in the site-specific work "oljud—ʇɥɹɥ(n)". Our action was directed at an "AI music composition" competition in 2025 organised as part of a very expensive engineering workshop focused on music generation research, but our motivations are more broad. This paper records the context, composition and realization of our "sonic manifesto of resistance", which was ultimately disqualified from the competition.

1. INTRODUCTION

The July 29 2025 call for submissions to the IEEE Big Data Workshop on AI Music Generation (AIMG), [1] begins:

> Music can touch the hearts of any audience without them possessing any knowledge of its context. The power of music is transcendental, and it stems from the timbre of the instrument(s), the fundamental rhythmic structure and melody, the dynamics, instrumentation, and many more, all of which cooperate in some form of harmony to create the final product. However, music is from not only the harmonious sounds produced from instruments, but also nature itself. The elegant songs sung by songbirds, the ringing water droplets, the enchanting waterfalls, the rustling leaves blown by the wind, the crashing waves at a beach, and many more are only a few examples of nature's music.

In addition to the invitation to submit research papers, the call describes a music competition but provides very few details and constraints:

> An AI-based music composition competition is incorporated into this workshop to stimulate both scientific and artistic creativity as well as provide such a unique platform. All researchers, musicians, lyricists, and hobbyists are welcome to submit their research works, compositions, and musical genres of all kinds are welcomed as well. The compositions may be comprised of original or human-enhanced AI-generated scores/lyrics.

No information is given about how music submissions are to be judged, but it does say the submission of compositions should include "1- or 2-page abstract, 1 music sheet, and 1 mp3 audio" submitted by Oct. 13 2025.

This call's allusion to music being a "universal language", along with the description of music as transcendental, and its vision of "harmonious cooperation" as some kind of natural ideal, presents a blatant contradiction of idealising nature while promoting a technology currently causing significant pollution [1–3]. This made us realize how pervasive some musical tropes still are, and how tightly interconnected they can be with thoughtless developments and uses of technology. Music history—particularly the 20th century—is filled with compositional attempts to challenge precisely these narrow ideals of music. For example, to what extent can one's "heart" be "touched" by Milton Babbitt's serial piano work "Post-Partitions" (1966), [2] Pauline Oliveros' electroacoustic noise composition "A Little Noise In The System" (1966), [3] Robert Ashley's "The Wolfman" (1964), [4] Elaine Radigue's "Transamorem Transmortem" (1973), [5] Maryanne Amacher's "Chorale 1" (1995), [6] Florian Hecker's "Resynthese FAVN" (2017), [7] Peter Ablinger's "4 WEISS" (2018/19) for large string orchestra and white

[1] https://groups.google.com/g/musicalmetacreation/c/gzwvLYNRUR8?pli=1, last accessed Dec 12 2025.

Copyright: ©2026 Bob L. T. Sturm and Elin Kanhov et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License 3.0 Unported, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

[2] https://newworldrecords.bandcamp.com/track/post-partitions, last accessed Dec. 12 2025

[3] https://imprec.bandcamp.com/track/a-little-noise-in-the-system-university-of-california-san-diego-electronic-music-studio-1967-1970, last accessed Dec. 12 2025

[4] https://pogus.bandcamp.com/track/robert-ashley-the-wolfman, last accessed Dec. 12 2025

[5] https://imprec.bandcamp.com/album/transamorem-transmortem, last accessed Dec. 12 2025

[6] https://youtu.be/sYWRS6xEdX8, last accessed Dec 12 2025

[7] https://blankformseditions.bandcamp.com/album/resynthese-favn-2, last accessed Dec. 12 2025

noise,[8] or any performance of Merzbow?[9] What "transcendental power" arises from these musical works purposely devoid of melody and functional harmony, with fundamental structures that can only be grasped by the most specialized composers [4]?

The appeal to the "harmonious" music in nature is similarly problematic. There are no harmonious sounds of nature. Not even the overtone series is transferrable to a logic of harmony, despite theoretical and ideological efforts of music theorists such as Schenker [5]. Moreover, the tropes of music in nature, nature as "musical," and music's "naturality", are tired. Appeals to nature have been present in music composition for centuries, if not longer [6]. Yet, the ways nature has been conceptualized in music and in other art forms throughout history are tightly interconnected with historical, epistemological, political, religious, philosophical, and aesthetic shifts [7]. That is to say, nature does not "sound" the same to the Romantic composers of the nineteenth century like Johannes Brahms [8], as it does to the spectral composers of the 1970s such as Gérard Grisey and Tristan Murail [9].

In his paper "Big Data and Musicology: New Methods, New Questions" [10] delivered to the American Musicological Society National Meeting, Wallmark bluntly tells musicologists that some among them must undertake research in music and big data:

> Do we really want a bunch of engineers and computer scientists setting the public agenda for music scholarship? I think you'll agree that we should not cede control of digital musicology to those who have the technical background but not the humanistic, cultural criticism skills to ask the right questions of their data. Musicologists need to be doing this stuff too, either individually or in collaboration with the computer nerds.

Ten years later, the 2025 AIMG call for work is a case in point, reflecting no discourse on what music is and can be, how it works and why it exists, where its value comes from, and the radical transformations music and musical thought have undergone during the 20th century. Music here seems to merely represent a data source to AI engineers, and an attractive playground to persuade peer reviewers lacking the "humanistic, cultural criticism skills" to substantively assess and comment on the contributions claimed in the engineering work they are reviewing.

And so, as two interdisciplinary practice-based researchers working in domains of AI and music,[10] we set out to respond to the 2025 AIMG call with artistic activism—or perhaps it is "making mischief" [12] via "cultural red teaming" [13]—by conceiving of, realising, and submitting a work that satisfies the stated requirements of the call for works, but breaks the assumptions and expectations of the organizers. It would be a "sonic manifesto of resistance" delivered to the doorstep of the workshop organisers,[11] participating but refusing to participate. Together with the discourse in this paper, our work becomes an output of agonistic artistic research, seeking to create knowledge by intentionally causing friction between fields without progressing into antagonism or worse.

The next section presents our broader motivations. Section 3 then discusses our resulting composition "oljud—ɹnɹbɹuıɟ(n)", our realisation of it for $n = 2$, our submission to the competition, and its results. We reflect on the experience in Sec. 4 and then conclude.

2. GENERATIVE AI AND POLLUTION

Generative AI[12] brings with it innovation, but also *pollution*. From data centers requiring massive amounts of power from coal and oil, thus polluting the air we breath [1–3], to human labor required to read and filter content generated by chatbots trained on text from the internet including its most vile parts, thus polluting the minds of people who take untreated psychological trauma back home [14], to the mined raw materials used to build and scale computing infrastructure, thus polluting many ecosystems [1], to the flood of AI-generated slop [15–19] from 'sloptrepreneurs' [20] in music streaming services [11, 21, 22], social media [17, 23, 24], academic conferences [25], and e-commerce [26], thus polluting our socio-cultural, political, intellectual discourses—is all this pollution worth the "synthetic text extruding machines" [27] trained not to (re)-produce facts, but instead confidently produce likely token sequences [28] that aim to maximize engagement, and appearing to be leading people into conspiratorial fever dreams [29], believing that they are solving fundamental mysteries of the universe [30], adopting dangerous health habits [31] and suicide [32], creating more work for others [33, 34], cheating themselves of an education [35], developing romantic relationships with chatbots [36, 37], reanimating and interviewing deceased victims of violence [38, 39]? The world seems to have left one pandemic only to enter another—a "mass delusion event" [40] that is polluting our personal and professional spheres, increasing the price of energy for us all [41], fomenting a widespread dependency on sycophantic bullshitting chatbots [27, 28, 42] while ultimately weakening the civic institutions that constitute democratic societies [43].

In restaurants, malls, gyms, taxis, waiting rooms, and the like, we are now being subjected to sound output by synthetic music audio extruding machines trained on data scraped from digital service providers without permission from the rights holders or the authors thereof [44, 45]. AI music generation companies claim to "democratize music", e.g., [19, 46], but what they seem to really mean is "maximize the number of paying subscribers", or, "Think of Spotify, but without having to pay royalties on billions of streams." It is now known that since January 2024, the commercial AI music generation company Suno has spent $32,000,000 on computing costs yet only $2,000 on data costs [19]. In November 2025, Suno settled with and was acquired by Warner Music Group, whereupon Suno CEO

[8] https://www.youtube.com/watch?v=4HQnKTQ5wXM, last accessed Dec 31 2025.

[9] https://merzbow.bandcamp.com, last accessed Dec. 12 2025

[10] Specifically, computer music, digital signal processing, machine learning, music informatics, musicology, music composition, and most recently "AI music studies" [11].

[11] Via the IEEE Big Data online submission portal.

[12] Machine learning systems that can synthesize, e.g., text, computer programs, images, speech, music and videos.

said in response: [13] "Our partnership with Warner Music unlocks a bigger, richer Suno experience for music lovers, and accelerates our mission to change the place of music in the world by making it more valuable to billions of people." Whether "it" refers to music itself or to Suno is not clear. Nonetheless, what we are not hearing behind all the AI music slop and its techno-optimist framing—e.g., music is a "problem" that technology will "solve"—is the deafening roar of the extruding machines with their thieved cogs [47], rendering inaudible the natural resources and human labor behind it.

Now, clearly, many are finding generative AI to be useful and to have effects that are positive, or at least not bad. OpenAI claims their "broadly democratized technology [ChatGPT] creates economic value through both increased productivity at work and personal benefit." [14] AI chatbots are solving long standing mathematics problems [48], efficiently and effectively programming complex software [49], and apparently doing "social science better than most professors," [15] in spite of a warning signed by 416 qualitative researchers against the use of generative AI for qualitative research [50]. These tools hold great benefit for non-English speakers in translating and rewording their work in non-native languages, and so they do contribute to leveling the "playing field" in knowledge work [51]. For some of the students we supervise at our university, we see them using generative AI to develop new skills and explore activities and domains that they previously found inaccessible, such as writing computer programs or music. We are aware of the fine line we must tread in being critical about the use of generative AI while not being classist or doing boundary work [51, 52]. Our artistic activism thus argues for reflexivity, transparency and accountability in any discourse about generative AI (including our own), a recognition of the historical pursuit of "machine intelligence" as one of deskilling and the consolidation of power in "factory masters" [53], and to counter predominant narratives of technological determinacy, inevitability, and the rest of the "TESCREAL bundle" [54].

3. COMPOSITION AND REALISATION

Our composition, "oljud—ɥiɥɹɥ(n)", is for any number $n > 0$ of people (need not be musicians) physically playing any number of air-blown instruments—the critical feature is that a human must play the instruments using air taken into their lungs—in a room housing a number of GPU clusters engaged with machine learning activities, e.g., training and inference. The score [16] and a realisation for $n = 2$ [17] are available online.



Figure 1. Instruments used for a realisation in a GPU cluster room with a sound level meter displaying 86 dB(A).

The score details eight stages: Initialization, Training, Augmentation, Halt, Debugging, Retraining, Testing, and End. The minimum duration of some of these stages is inversely proportional to n, mimicking the computational acceleration achieved by parallelism, e.g., stochastic gradient descent. Any realisation of the piece is to last at least $7 + 75/n$ minutes. It involves no audience other than those involved in the performance. We arrived to this score iteratively, starting with a team Slack thread on which Kanhov posted on Sep. 24 2025 a link to the 2025 AIMG call saying, "Anyone keen to explore the transcendence of music, the harmonious nature of sound, enchanting waterfalls and elegant songbird songs—all through the magic of AI? Then this is the workshop for you". Another team member [18] responded with a link to Merzbow; and Sturm posted, "What if we did a mash up? Field recording in the GPU farm with a bunch of us forming a "screaming choir" screaming out a manifesto? And it needs to be one hour long." Sturm then composed a vague score (e.g., "1. sit and listen for five minutes," and "2. play long tones").

We performed and recorded "oljud—ɥiɥɹɥ(2)" on Oct. 8 2025 in a GPU cluster room at our university containing over 100 GPUs (various models) being used by many researchers across several departments to train and run a multitude of AI models. Sturm asked for permission to access the room, in which his research team has their own cluster of 8 RTX 3090 GPUs—thanks to the MUSAiC project. [19] At the decided date and time, we met an IT manager outside, and Sturm explained that "we need to collect acoustic data for signal processing denoising laboratories"—the reason why the plastic bag we were carrying had many different whistles poking out. [20]

We had no experience of the environment before entering for our performance. The temperature was much more comfortable than we anticipated, and the noise level was quite intense (even with ear plugs). Figure 1 shows a sound

[13] https://www.wmg.com/news/warner-music-group-and-suno-forge-groundbreaking-partnership, last accessed Dec 12 2025

[14] "How people are using ChatGPT" Sep. 15 2025 https://openai.com/index/how-people-are-using-chatgpt

[15] https://alexanderkustov.substack.com/p/academics-need-to-wake-up-on-ai and https://statsandsociety.substack.com/p/you-should-absolutely-be-freaking, last accessed Mar. 26 2026.

[16] http://kth.diva-portal.org/smash/record.jsf?pid=diva2%3A2048181, last accessed Mar. 26 2026.

[17] https://bobltsturm.bandcamp.com/album/oljud-n-q-n-2, last accessed Mar. 26 2026.

[18] Dr. Ken Déguernel

[19] https://musaiclab.wordpress.com/, last accessed Mar. 26 2026.

[20] We have since tried to make a realization with more people in a room of GPU clusters at another nearby university, and were even invited to do so by a friend who works there as a research professor. But the research professor gave too many details to the IT staff that maintain the computational infrastructure and he was thus denied permission to host us.

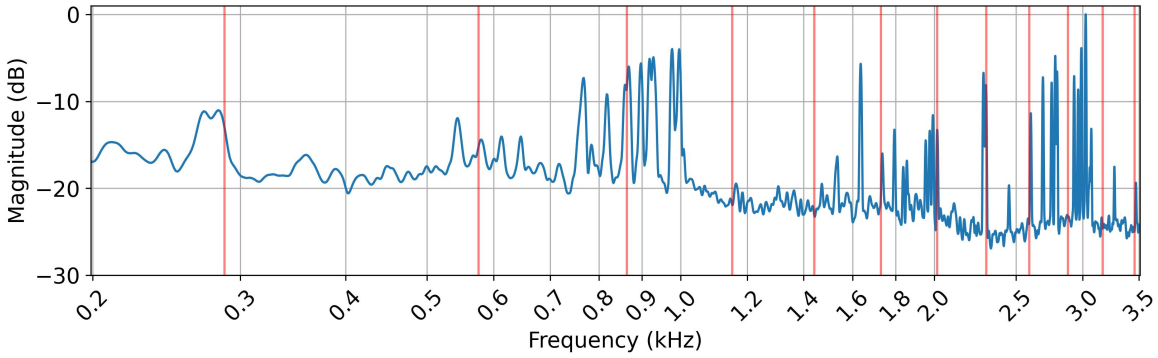


Figure 2. Magnitude spectrum of the acoustic environment of the GPU cluster room overlaid with the frequencies of harmonics of the pitch 'D4' given a tuning frequency of 432 Hz.

level meter measuring 85 dB(A) in the room, as well as the instruments we used in the performance. We placed two Zoom H4 Handy audio recorders about 6 meters apart at two elevations (about 1 meter and 2 meters from the floor), and started a video recorder before the performance (which stopped after 15 minutes because Sturm had not emptied the memory card). Figure 2 shows the magnitude spectrum of the acoustic environment, with reference to the first 12 harmonics of the pitch 'D4' given a tuning frequency of 432 Hz (i.e., the fundamental is 288.3 Hz). [21] We see spectral peaks at: 282.5, 767.5, 817.5, 867.5, 897.5, 928.75, 977.5, 996.25, 1636.25, 2288.75 and 2603.75 Hz. There are several clusters at higher frequencies. The overall sound of the machinery in the room is clearly not tonal.

Alone in the server room, we sat on two chairs about two meters apart and started a digital timer. Between us, we played nine different instruments (see Fig. 1): one ocarina in D, one recorder in E, one C5 whistle, one D3 whistle, three D5 whistles, one G5 whistle, and a train conductor's whistle. Sturm gave visual cues as each stage began. Figure 3 shows our placement near the beginning of stage 2. At a few times we stood up and moved around the space playing before sitting back down in swapped positions. We made no contact with any computational equipment, which was separated from us by floor-to-ceiling wire fencing. We finished about an hour after entering the space, and left it just as we had found it. Sturm mixed the two stereo recordings together in a DAW to create the final mix, using no effects except for a bit of equalization and multiband compression.

Annotated sonograms attached to our score show demarcations of the eight stages. One can see many drones in the environment, some of which have constant frequency and amplitude, e.g., at about 1.63 kHz, others with a constant frequency but undergoing amplitude modulation, e.g., at about 2.96 kHz with a period of 45 seconds. Also apparent is a subtle "step tone" component having a frequency stepping up and down in a non-periodic way in the range of $500 - 750$ Hz (we do not know the origin of that component). The coincidence of the fundamental frequency of the first long tone played initiating stage 2 (D5) with that of the "step tone" was entirely coincidental. As we walked back to our offices after the performance we agreed that the noisy environment was more unique and less harsh than we expected, and certainly much more sonically interesting than waterfalls.



Figure 3. A video still from the performance showing Sturm (left) producing the first long tone of the second stage while Kanhov (right) listens and reflects.

Like the GPU clusters, the performers move in parallel between various states of labor, from easy to difficult, responding to each other (what they can hear and see), and to their dynamic acoustic environment, by physically pressing through musical instruments the same air used to remove heat from the computational machinery. Nuance is impossible—skill is optional—endurance is required; yet much of the human labor in performing the work is invisible, mostly, masked by the noise of the machines. Our work's title joins together the Swedish and French words for "noise". [22]

The AIMG website in late Sep. 2025 stated the submission date was Oct. 20. [23] We submitted our recording and anonymised documentation (including score, link) on Oct. 14 and received an automated reply saying the submission was received. We heard nothing further, but when we checked the workshop website on Dec. 5 2025—three days before it was to begin— we noticed that music selections had been announced, the call text had been changed (removing the last two sentences of the text in the Introduction above), and stricter requirements for the music compe-

[21] This choice alludes to a frequency some people believe is transcendental—but that's a different artistic research project.

[22] At the time of composition, one author was considering a career move to Université de Montréal, where after a period of four years they would be required to speak and teach in French... so here was a start.

[23] `http://www.intellisky.org/workshops/AIMG2025/workshop_AIMG2025.html`, last accessed Dec. 30 2025.

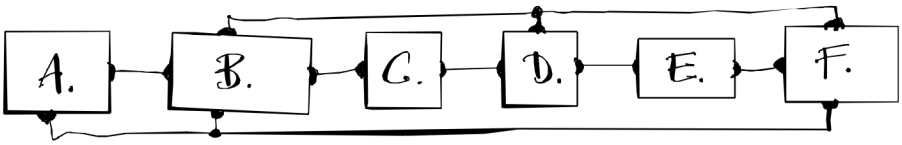


Figure 4. A workflow diagram of our artistic activism, moving from being motivated to protest (A), to arriving at a vision of activism (B), to composing and planning the act (C), to performing and recording the act (D), to possibly revising the composition (E), to reflecting on the action and possibly being moved to act again (F).

tition had been added: "Please note that composition submissions not created using AI or algorithmic methods, or including audio exceeding 10 minutes in duration, will not be eligible for participation in the AI Music Composition Competition." We sent a friendly email to the organisers on Dec. 12 2025 asking about the change in requirements, but there has been no response.

Figure 4 shows a diagram of our rather informal workflow: A) we read something that motivates us to protest; B) after a process of fuming and stewing with others we arrive at a vision of an act of artistic activism; C) we compose our action and plan its execution; D) we perform and record it, fuming and reflecting at the same time; E) we revise the composition; and then F) we reflect through writing and reading more deeply—which invariably leads to more fuming and motivation for activism, and so on. Perhaps this can be seen as a kind of "critical technical process" [55], and given its emphasis on recording our acts in the moment, a kind of "technical practice research" [56].

4. REFLECTIONS

So far, our artistic activism has resulted only in slight edits to the call text of the 2025 AIMG workshop. We are not disappointed that our work was not selected, but we are frustrated that the competition conditions were changed after we submitted and there was no attempt to notify us. Had we known about these changes we would have made our realisation be 10 minutes duration and explicitly employed some "algorithmic method"—for instance, we could apply a high-Q notch filter having a cutoff frequency set with a random walk. However, since the mandatory cost of registering to attend the *online* workshop to present any work was over 1,000 USD, we would have withdrawn our work had it been selected anyway. Such a high cost of participation shows who is welcome—and it is not anyone lacking financial resources, e.g., many humanities scholars, musicians and artists.

Some may ask, "Why not just email the workshop committee to lodge your protest? Why go to all this trouble to make your point?" Our practice-based research offered us other ways to think about protest, to think through the medium of music and sound, to approach it in non-discursive ways, to bear witness to our current times and predicaments, to challenge us with the opportunity to experiment with ways of resisting through the medium, and to toy around with the "play in the system" [57] given by the lack of details of the AIMG music competition. Our motivation is more broad than this, however (see Sec. 2), and our itch to engage in artistic activism had been growing for a while from a constant barrage of AI hype and critical work, e.g., [13,47,53,54,57–60]. Our composition and its realisation "flips the script" in AI music. "oljud—$\text{ɹnɹb}(n)$" situates itself not at a point of decoded generative AI output—the mobile phone, the tablet, the laptop computer, accessing the online AI music generation platforms streaming music audio inferred at some data center somewhere in the world (the locations of these data centers is notoriously difficult to find)—but at a point of origin, inside a physical meatspace working to regulate the heat and noise emitted by our modern day inference engines. "oljud—$\text{ɹnɹb}(n)$" brings the roar to the fore, placing us at a slop mine nexus literally vibrating with an intensity so high that permanent hearing loss would result without proper ear protection. While at the other end of the "series of tubes" [24] flowing from this room some user might be experiencing the "democratizing" power of generative AI, at our end the environment seems inhospitable for making music—and so we as performers try to inhabit the space, like music extremophiles, appearing in what few gaps of the audible spectrum we can occupy for as long as our breath holds.

Our work forms an arc of irony to that of the Italian futurists over a century before, who cheered technological advancement and supremacy. In addition to praising war and deriding women, the 1909 "Declaration of Futurism" by Marinetti [61] commands people

> [to sing] the nocturnal vibration of arsenals and docks beneath their glaring electric moons; greedy stations devouring smoking serpents; factories hanging from the clouds by the threads of their smoke; bridges like giants gymnasts stepping over sunny rivers sparking like diabolical cutlery; adventurous steamers scenting the horizon; large breasted locomotives bridled with long tubes, and the slippery flight of aeroplanes whose propeller has flag-like flutterings and applauses of enthusiastic crowds.

Russolo begins his 1913 manifesto "L'Arte de rumori" [62] by posing that noise started with machines: "In antiquity, life was nothing but silence. Noise was really not born before the 19th century, with the advent of machinery" [63, pg. 4]. He ends by predicting: "In a few years, the engines of our industrial cities will be skillfully tuned so that every factory is turned into an intoxicating orchestra of noises" [63, pg. 12]. Our composition "oljud—$\text{ɹnɹb}(n)$", substituting Russolo's intonorumori with a skillfully tuned computing factory of GPUs being commanded by people distributed over campus (and elsewhere), is not intended to be a celebration of technology. The irony comes when

[24] `https://en.wikipedia.org/wiki/Series_of_tubes`, last accessed Dec. 15 2025.

loved ones of the authors heard a playback of our realisation and remarked: "That sound brings back good memories of when I was experimenting with crypto"; and, "That reminds me of how I like to stand amid the freezers in the supermarket and listen to them drone". In the end we also have to agree that the noise of these machines was beautiful, and the serendipitous consonance of the instruments we chose to bring with that noise feels miraculous. So much for our intent to communicate the deafening pollution arising from slop factories, and our feelings of suffocation from the hype.

We would be remiss to not situate our work among others that involve noise, performance in challenging spaces, machines appropriated to make music, and artistic activism. In his 2007 book "Noise/Music" [64], experimental musician Paul Hegarty draws a solid line between technology and all developments of noise music and noise in music: from the Futurists, musique concrète and Fluxus, to rock, punk, metal, industrial, hip-hop, Japanese noise music, and glitch, among others. Our composition "oljud—ɟiɹɹɥ(n)" is a kind of noise music, but also a kind of field recording. While Ablinger's lifetime of work with noise comes from deep deliberation [65], "oljud—ɟiɹɹɥ(n)" is reactive and improvisational in both its composition and performance.[25] Of course music from machines and sounds produced by machines deserves mention of George Antheil, Edgar Varèse, Johanna Beyer, Pierre Shaeffer, Test Dept, and so on. When it comes to music in difficult environments, Stockhausen's "Helikopter-Streichquartett" comes to mind, involving distributing and coordinating a string quartet among four flying helicopters. Matthew Burtner's climate activism through his ecoacoustic music, e.g., his piece "Icefield",[26] is also especially relevant. This list is far from complete and the authors welcome suggestions.

5. CONCLUSION

Pratella's 1910 "Manifesto of Futurist Musicians" [66] encourages "the young, the bold and the reckless"

> To abstain from participating in any competition with the customary closed envelopes and related admission charges, denouncing all mystifications publicly, and unmasking the incompetence of juries, which are generally composed of fools and impotents.

Our artistic activism ignores this advice; and, furthermore, we would like to explicitly state that we do not believe the organisers of AIMG 2025 are "fools" or "impotents" (that goes beyond the agonism we aim to practice). Rather, we believe deeper reflection is needed on the part of those in privileged positions about what exactly they are attempting to do with AI and music and why through organising such activities—that is the ethics of the situation. An example is provided by Sturm's "Ai Music Generation Challenges" of 2020–2022 [67–69], the three objectives of which were:

> 1) to promote meaningful approaches to evaluating music Ai; 2) to see how music Ai research can benefit from considering traditional music, and how traditional music might benefit from music Ai research [70]; and 3) to facilitate discussions about the ethics of music Ai research applied to traditional music practices [71].

The design of each challenge was made in collaboration with music practitioners, and was transparent in its detailed descriptions of its aim, the task, the requirements for participation, and how submissions are to be evaluated. These challenges (intentionally not called "contests" or "competitions") did not seek to treat music as a sandbox for AI development, but rather to motivate the researcher think about "success" in narrow but well-defined and accessible contexts—not only in terms of the performance of AI models, but also the methodologies of the AI researcher, i.e., doing machine learning that matters for music [70].

In his 1977 book *Bruits* about music and economics, Attali argues that [72, p. 30]: "[Music's] primary function is not to be sought in aesthetics, which is a modern invention, but in the effectiveness of its participation in social regulation". As an "organization of noise" (p. 4), music is "a tool for the creation and consolidation of a community" (p. 6), and furthermore, the technologies for its distribution "are contributing to the establishment of a system of eavesdropping and social surveillance" (p. 8). Noise, Attali argues, is a source of power, and "the noises of a society are in advance of its images and material conflicts" (p. 11). Later in his text, Attali discusses the history of noise control, which, aligned with Russolo's claim of noise being inexistent before the invention of machines, began with the automobile (p. 123). The acoustic isolation of GPU clusters from "polite society" is also a form of noise control, and the development of AI is broadly but implicitly motivated by those in power to consolidate even more power [53]. "oljud—ɟiɹɹɥ(n)" unleashes this controlled noise, to render it as visible as the pollution emitted by the 35 methane gas turbines powering Musk's purposely racist chatbots [2]. Nature today is situated at the intersection of the sixth mass extinction and the fourth technological revolution, mediated by data and processed by machines [73]. Who knows, the machines involved in our realisation of "oljud—ɟiɹɹɥ(n)" might at the time of recording actually have been processing environmental data or calculating predictions of melting glaciers, ironically generating even more "enchanting waterfalls".

Acknowledgments

We would like to acknowledge the MUSAiC team (ERC-2019-COG No. 864189), the project Musical Entanglements with AI funded by the Swedish Research Council (No. 2024-01012), and the IT managers at KTH permitting us to take measurements in the GPU cluster room. We would also like to thank the first few listeners of our work, and the unexpectedly positive feedback we received from them. And thank you to Professor Jeff Morris for responding enthusiastically to our action.

6. REFERENCES

[1] K. Crawford, *Atlas of AI*. Yale University Press, 2021.

[25] It was some time after its composition that we became aware of Ablinger.

[26] `https://www.ravellorecords.com/catalog/rr8066/`, last accessed Dec. 31 2025.

[2] D. Kerr, “Elon Musk’s xAI powering its facility in Memphis with ‘illegal’ generators,” *The Guardian*, April 10 2025.

[3] T. Perkins, “Advocates raise alarm over Pfas pollution from datacenters amid AI boom,” *The Guardian*, Oct. 4 2025.

[4] M. Babbitt, “The composer as specialist,” *High Fidelity*, vol. 8, no. 2, 1958.

[5] H. Schenker, *Free Composition (Die Freie Satz) (trans. E. Oster)*. Longman, 1979.

[6] S. Clark and A. Rehding, Eds., *Music Theory and Natural Order from the Renaissance to the Early Twentieth Century*. Cambridge University Press, 2001.

[7] E. Kanhov, “Encounters between music and nature: A productive and transversal approach to contemporary music analysis,” Ph.D. dissertation, Stockholm U., 2023.

[8] L. B. Meyer, *Style and Music: Theory, History, and Ideology*. The University of Pennsylvania Press, 1989.

[9] J. O’Callaghan, “Spectral music and the appeal to nature,” *Twentieth-Century Music*, vol. 15, pp. 57–73, 2018.

[10] Z. Wallmark, “Big data and musicology: New methods, new questions,” American Musicological Society National Meeting, Tech. Rep., 2013.

[11] B. L. T. Sturm, K. Deguernel, R. S. Huang, A.-K. Kaila, P. Jääskeläinen, E. Kanhov, L. C. Vila, D. C. Dalmazzo, L. Casini, O. Bown, N. Collins, E. A. Drott, J. Sterne, A. Holzapfel, and O. Ben-Tal, “AI Music Studies: Preparing for the Flood,” in *Proc. AI Music Creativity*, 2024.

[12] A. Burchill, “Exploring agonism with mischief: participatory performance in the public realm,” Ph.D. dissertation, Loughborough University, 2018.

[13] E. Salvaggio, C. Sinders, and S. M. Swanson, “Cultural Red Teaming: ARRG! and Creative Misuse of AI Systems,” *Critical AI*, vol. 3, no. 1, 2025.

[14] B. N. D. Ranta, “The unknown women of content moderation,” in *Data Workers’ Inquiry*, M. Miceli, A. Dinika, K. Kauffman, C. S. Wagner, and L. Sachenbacher, Eds., 2024.

[15] M. Kirschenbaum, “Prepare for the textpocalypse,” *The Atlantic*, March 2023.

[16] D. Ø. Madsen and R. W. Puyt, “The 7Vs of AI Slop: A Typology of Generative Waste,” *SSRN preprints*, 2025.

[17] P. Bharadia, “Cat soap operas and babies trapped in space: the ‘AI slop’ taking over YouTube,” *The Guardian*, Aug. 11 2025.

[18] J. Rothman, “A.I. is coming for culture,” *New Yorker*, Aug. 2025.

[19] K. Robinson, “Suno creates an entire Spotify catalog’s worth of music every two weeks, says investor pitch deck for $250m fundraise,” *Billboard*, Nov. 2025.

[20] M. Read, “Drowning in slop,” *New York Mag.*, Sep. 2024.

[21] D. Milmo, “Spotify removes 75m spam tracks in past year as AI increases ability to make fake music,” *The Guardian*, Sep. 25 2025.

[22] L. C. Vila, B. L. T. Sturm, L. Casini, and D. Dalmazzo, “The AI Music Arms Race: On the Detection of AI-Generated Music,” *Trans. Int. Soc. Music Info. Retrieval*, vol. 8, no. 1, pp. 179–194, 2025.

[23] K. Knibbs, “Generative AI Podcasts Are Here. Prepare to Be Bored,” in *Wired*, May 24 2023.

[24] J. Koebler, “Where Facebook’s AI Slop Comes From,” *404media*, Aug 6 2024.

[25] K. Kousha and M. Thelwall, “How much are LLMs changing the language of academic papers after ChatGPT? A multi-database and full text analysis,” *arXiv:2509.09596*, 2025.

[26] K. Knibbs, “Scammy AI-Generated Book Rewrites Are Flooding Amazon,” *Wired*, Jan 10 2024.

[27] E. M. Bender and A. Hanna, *The AI Con: How to Fight Big Tech’s Hype and Create the Future We Want*. HarperCollins, 2025.

[28] M. T. Hicks, J. Humphries, and J. Slater, “ChatGPT is bullshit,” *Ethics and Info. Tech.*, vol. 26, no. 38, 2024.

[29] K. Hill, “They Asked an A.I. Chatbot Questions. The Answers Sent Them Spiraling.” *The Guardian*, June 13 2025.

[30] E. Siegel, “Why ’vibe physics’ is the ultimate example of AI slop,” *Big Think*, July 30 2025.

[31] A. Eichenberger, S. Thielke, and A. V. Buskirk, “A case of bromism influenced by use of artificial intelligence,” *Annals of Internal Medicine: Clinical Cases*, vol. 4, no. 8, 2025.

[32] L. Shroff, “ChatGPT Gave Instructions for Murder, Self-Mutilation, and Devil Worship,” *The Atlantic*, July 24 2025.

[33] K. Niederhoffer, G. R. Kellerman, A. Lee, A. Liebscher, K. Rapuano, and J. T. Hancock, “AI-Generated “Workslop” Is Destroying Productivity,” *Harvard Business Review*, 2025.

[34] A. Yang, “Humans are being hired to make AI slop look less sloppy,” *NBC News*, Aug. 31 2025.

[35] J. D. Walshe, “Everyone is cheating their way through college,” *New York Mag.*, July 5 2025.

[36] S. Cole, “ChatGPT’s Hail Mary: Chatbots You Can Fuck,” *404media*, Oct. 14 2025.

[37] S. Heritage, "'I felt pure, unconditional love': The people who marry their AI chatbots," *The Guardian*, July 12 2025.

[38] C. Neff, "AI of dead Arizona road rage victim addresses killer in court," *The Guardian*, May 7 2025.

[39] ——, "Jim Acosta interviews 'made-up' AI avatar of Parkland victim Joaquin Oliver," *The Guardian*, August 5 2025.

[40] C. Warzel, "AI is a mass-delusion event," *The Atlantic*, 2025.

[41] M. S. Smith, "The Hidden Behemoth Behind Every AI Answer," *IEEE Spectrum*, Oct 1 2025.

[42] T. R. Hannigan, I. P. McCarthy, and A. Spicer, "Beware of botshit: How to manage the epistemic risks of generative chatbots," *Business Horizons*, vol. 67, no. 5, pp. 471–486, 2024.

[43] W. Hartzog and J. Silbey, "How AI Destroys Institutions," *UC Law J.*, 2026.

[44] US District Court for the District of Massachusetts, "UMG Recordings, Inc. et al. v. Suno, Inc. et al 1:24-cv-11611," 2024.

[45] ——, "UMG Recordings, Inc. et al. v. Uncharted Labs, Inc. et al 1:24-cv-04777," 2024.

[46] V. Velardo, "The Sound of AI (Conversations): Valerio Velardo interviews Alex Mitchell," April 2022. [Online]. Available: https://youtu.be/iyTJF7b6BwE

[47] T. S. Goetze, "AI Art is Theft: Labour, Extraction, and Exploitation," in *Proc. FAccT*, 2024.

[48] D. Knuth, "Claude's cycles," Stanford University, Stanford Computer Science Department, Tech. Rep., 2026.

[49] J. Jiang, F. Wang, J. Shen, S. Kim, and S. Kim, "A survey on large language models for code generation," *ACM Trans. Softw. Eng. Methodol.*, vol. 35, no. 2, Jan. 2026.

[50] T. Jowsey, V. Braun, V. Clarke, D. Lupton, M. Fine, {414 Signatories}, and S. Silverio, "We reject the use of generative artificial intelligence for reflexive qualitative research," *QUALITATIVE INQUIRY*, 2025.

[51] A. Sarkar, "AI Could Have Written This: Birth of a Classist Slur in Knowledge Work," in *Proc. CHI EA*, 2025.

[52] L. Giray, "AI Shaming: The Silent Stigma among Academic Writers and Researchers," *Annals of Biomedical Engineering*, 2024.

[53] M. Pasquinelli, *The Eye of the Master: A social history of artificial intelligence*. Verso, 2023.

[54] T. Gebru and E. P. Torres, "The TESCREAL bundle: Eugenics and the promise of utopia through artificial general intelligence," *First Monday*, vol. 29, no. 4, 2024.

[55] P. E. Agre, *Social Science, Technical Systems, and Cooperative Work*. Psychology Press, 1998, ch. Toward a Critical Technical Practice: Lessons Learned in Trying to Reform AI.

[56] T. Pelinski, A. McPherson, and R. Fiebrink, "Ways of knowing, ways of writing: technical practice research in new musical instrument design," *J. New Music Research*, vol. 53, no. 1–2, pp. 79–92, 2024.

[57] A. W. Fisher, *The Play in the System: The Art of Parasitical Resistance*. Duke University Press, 2020.

[58] G. Vigliensoni, P. Perry, and R. Fiebrink, "A Small-Data Mindset for Generative AI Creative Work," in *Proc. Gen AI and HCI workshop at CHI*, 2022.

[59] D. McQuillan, *Resisting AI: An Anti-fascist approach to artificial intelligence*. Bristol University Press, 2022.

[60] J. Herington, R. Borasi, B. J. Guerroro, D. E. Miller, B. Koerner, Y. J. Jan, Z. Borys, and R. Roberts, "Musicians' ethical concerns about AI: an interview study," *AI & Society*, 2025.

[61] F. T. Marinetti, "Declaration of futurism," *Poesia*, vol. 5, 1909.

[62] L. Russolo, *L'Arte dei rumori*. Corso Venzia, 1916.

[63] ——, *The Art of Noise (futurist manifesto, 1913)*. ubuclassics, 2004.

[64] P. Hegarty, *Noise/Music*. Bloomsbury Publishing, 2007.

[65] G. D. Barrett, "Between noise and language: The sound installations and music of Peter Ablinger," *Mosaic*, vol. 42, no. 4, pp. 147–164, 2009.

[66] U. Apollonio, Ed., *Documents of 20th Century Art: Futurist Manifesto*. Viking Press, 1973.

[67] B. L. T. Sturm and H. Maruri-Aguilar, "The Ai Music Generation Challenge 2020: Double Jigs in the Style of O'Neill's "1001"," *J. Creative Music Systems*, 2021.

[68] B. L. T. Sturm, "The Ai Music Generation Challenge 2021: Summary and Results," in *Proc. AIMC*, 2022.

[69] ——, "The Ai Music Generation Challenge 2022," in *Proc. AI Music Creativity*, 2023.

[70] B. L. Sturm, O. Ben-Tal, U. Monaghan, N. Collins, D. Herremans, E. Chew, G. Hadjeres, E. Deruty, and F. Pachet, "Machine learning research that matters for music creation: A case study," *J. New Music Research*, vol. 48, no. 1, pp. 36–55, 2018.

[71] A. Holzapfel, B. L. Sturm, and M. Coeckelbergh, "Ethical dimensions of music information retrieval technology," *Trans. Int. Soc. Music Information Retrieval*, vol. 1, no. 1, pp. 44–55, 2018.

[72] J. Attali, *Noise: The political economy of music (trans. B. Massumi)*. University of Minnesota Press, 1985.

[73] D. J. Haraway, *Staying with the Trouble: Making Kin in the Chthulucene*. Duke University Press, 2016.

Linear A: A Composer's Integrated Workspace

Christopher Trapani

Louisiana State University
ctrapani@lsu.edu

ABSTRACT

This paper explores the tools used in the composition of Linear A *for Bohlen-Pierce clarinet and electronics (2021). The piece explores the many ways a single microtonal line can be manipulated: echoed, transposed, splintered, zig-zagged, chopped into bubbling snippets, and finally, realigned into a closing chorale. Microtonal features in the* bach *library, critical in both the planning and performance stages, are exploited to pivot between fingerings and the precise non-tempered Bohlen-Pierce tuning. A notation interface uses a system of markers to visualize canonic replies in the electronics. This information is in turn used to generate a master file for real time score following with* Antescofo~. *The result is a dynamic, flexible, self-contained environment that accommodates the peculiarities of the project.*

1. INTRODUCTION

Linear A, a work for Bohlen-Pierce clarinet plus electronics, was composed in 2021. Envisioning a piece for solo instrument in dialogue with canonic replies recorded in real time, I know that pre compositional planning would be crucial. My goal was to create an environment where the unique tuning of the Bohlen-Pierce scale [1] could be both visualized and heard, and where the timing and transposition of canonic voices could also be represented. The flexibility of the notation package *bach* would be exploited to move between unmeasured and quantized staff notation, as well as between sounding and fingered pitches for the clarinet. The ultimate achievement would be a further integration with realtime electronics: score and canonic information would be reinterpreted as commands for *Antescofo~,* preparing a text file that could pilot the score follower for a live performance.

2. FIRST STEPS

Work began by defining the pitches available on the instrument: those of a Bohlen-Pierce scale tuned to the concert low D of the clarinet, using the fingerings given by Nora-Louise Müller as a guide [2] (Müller, 40). The bach library's command "pitcheditrange" constrains to a specific collection of pitches.

Copyright: © 2025 Christopher Trapani. This is an open-access article dis- tributed under the terms of the Creative Commons Attribution License 3.0 Unported, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Switching between two sets of constrained pitches (by means of a live.text control) allows one to toggle dynamically between the Bohlen-Pierce collection ("CONCERT") and fingered pitches as preferred by clarinetists ("TRANSPOSED").

Continuing with the first steps of computer-assisted composition, I then devised a few ways of exploring the intervallic content of the Bohlen-Pierce scale. In one bach.roll, I highlight scale degrees that form a 3:5:7 relationship (the fundamental chord in the Bohlen-Pierce tuning, which is based on odd partial numbers). In a second bach.roll, I use the bach.circle object to create modes which are subsets of the tuning's 13 notes to a tritave, the Bohlen-Pierce equivalent of the octave, where all pitches repeat at the interval of a perfect twelfth (see Figure 1).

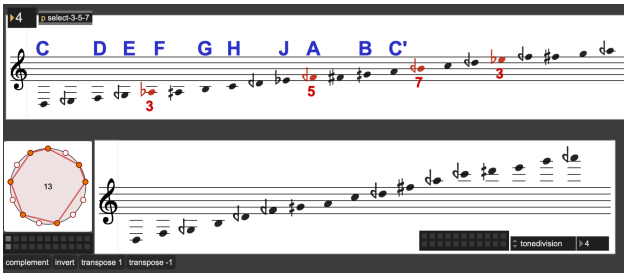


Figure 1. **Pitch Material**: Two bach.roll objects containing Bohlen-Pierce scales. The top highlights 3:5:7 relationships in the scale, the bottom displays certain modal subsets.

3. CANONS

The canonic sketchbook starts with an unmeasured representation of a clarinet phrase inside of a bach.roll. Markers are inserted to denote the start and end points of canonic replies. Each marker also contains information about the entry delay, relative speed, and transposition of each voice. With each canon in a separate voice, a second bach.roll combines the original voice with its canonic answers, with all staves in proportional notation. This visualization allows the composer to envision the potential result and to shape the network of canonic replies, adjusting speed and entry points as needed. Furthermore, the possibility of working in unmetered time space, before quantization, is a notable advantage (see Figure 2).

In a real-time performance, these markers are repurposed as the start and end points for buffer playback.

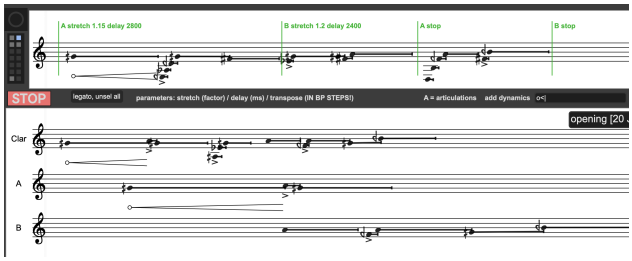


Figure 2. Canonic Sketchbook: Using markers insides of an unmeasured bach.roll, canonic replies to a given phrase can be envisioned in multiple voices

4. QUANTIZATION

Once the timing of canons has been worked out, the next step is a quantization of the unmetered bach.roll. A meter and tempo are given, yielding a quantized version of a line previously in spatial notation. Small adjustments in alignment can also be made to ensure a smooth quantization, and the values of minimal units can also be assigned. At this stage dynamic markings and articulations can also be added by altering the slot content of selected notes; a textedit box is introduced as a shortcut means of interacting with the bach.score interface (see Figure 3). The transposed version quantized phrase can the be exported to XML and opened inside any notation software, completing the transformation from conception to finalization of a particular phrase.

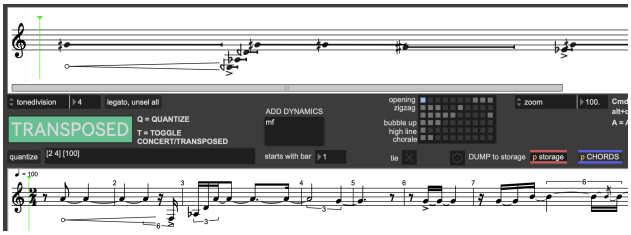


Figure 3. Quantization: An umetered phrase is quantized into an XML-ready bach.score, complete with dynamic markings.

5. ZIG-ZAG CANONS

A second type of canon used in the piece is a zig-zag canon, where a given portion of a line is repeated, first backwards—so that the audio splits off in two directions at the canon's starting point—then in its original direction, a process visualized in Figure 4. The chosen segment can be repeated multiple times with a decrescendo (made visible by choosing the greyscale alpha to represent velocity), reversing direction with each repetition.

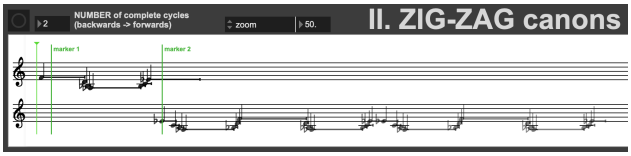


Figure 4. Zig-Zag Canon: Markers denote a portion of an unmetered phrase, which is repeated in alternating directions.

6. SEGMENTATION

In the third section of the piece, previously heard phrases on the clarinet are segmented into smaller chunks and played back through a concatenate synthesis engine using CataRT in MuBu. A patch, included here as Figure 5, was constructed to visualize and to compare the efficacy of three different segmentation types in MuBu: an FFT analysis using a loudness threshold, a YIN algorithm to segment according to fundamental frequency estimations, and a MEL analysis that detects fluctuations in the mel frequency parameter of a spectral flux analysis. For the purposes of *Linear A*, it was determined that with a clear signal a YIN analysis was sufficient for robust and reliable segmentation.

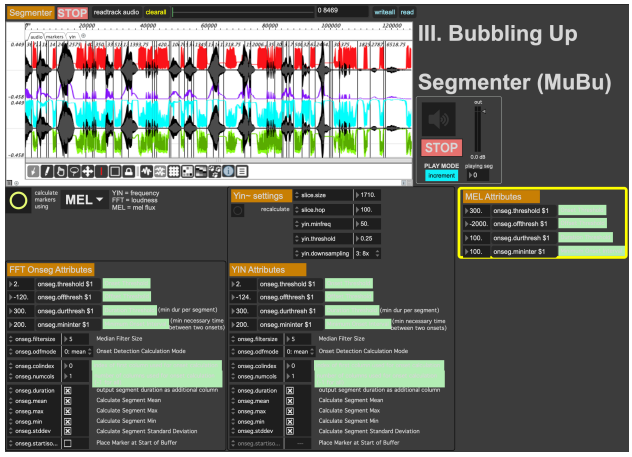


Figure 5. Segmenter: A patch constructed to visualize and compare segmentation algorithms in MuBu

7. SYNCHRONIZER

A patch known as the Synchronizer, developed with Greg Beller at IRCAM in 2014, takes multiple phrases with pre-analyzed attack points and represents them symbolically on multiple staves, again using the *bach* library for a microtonal proportional representation, seen here in Figure 6. These phrases can be re-aligned into a chorale by taking a list of onsets (either the onsets of a given voice or an independently generated list) and synchronizing all attacks to that onset collection. Additionally, each voice can be retuned freely, or snapped to a collection of given pitches.

Beneath the surface interface, each voice is reinterpreted by a SuperVP module. Onset and pitch parameters in each bach.roll are recalculated as transposition and time stretch values for the phase vocoder. A realtime rendering can be heard using SuperVP objects within Max, and an offline export option, communicating directly with the SuperVP engine, is also available.

The realtime version of *Linear A* aims to implement this process in real time, using *Antescofo~* as an intermediate step in determining attack points and alignment.

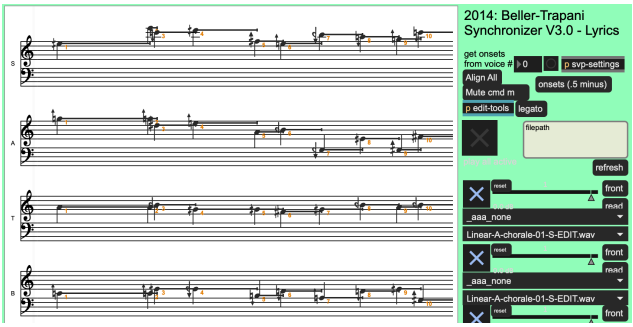


Figure 6. Synchronizer: Bridging the gap between symbolic and audio representations, the Synchronizer provides an interface that allows manipulations of independent voices that will be realized using SuperVP.

8. SCORE FOLLOWING

Once the complete score (see figure 7) has been compiled, a final step involves the transport of the score to a text file format, readable by the score follower *Antescofo~*. To accomplish this, I created a patch which reads the note values (given in midi cents, a language common to *Antescofo~* and *bach*) and rests. Embedded markers in the bach.score contain the necessary information for creating the canonic replies by formatting messages for speed, transposition, and entry delay for each canon, then sending this information to a sub patcher that pilots a *super-vp.play~* module for vocoded playback.

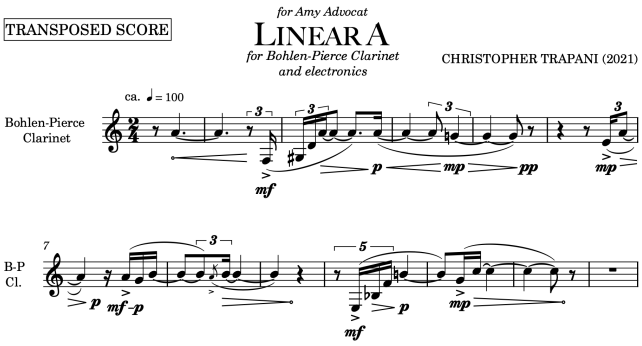


Figure 7. Score excerpt: the opening bars of *Linear A* in a finalized, formatted, transposed score.

Here is the point where real-time processes and computer-assisted composition achieve their peak synthesis. The bach.score generates a file which is ready-formatted for the score follower. The bach.score provides an entirely legible interface, though for the sake of convenience in formatting, the score is exported and typeset in Dorico. The *Antescofo~* text file score also provides a complete and self-contained representation of the entire piece, ready to pilot the accompanying patch.

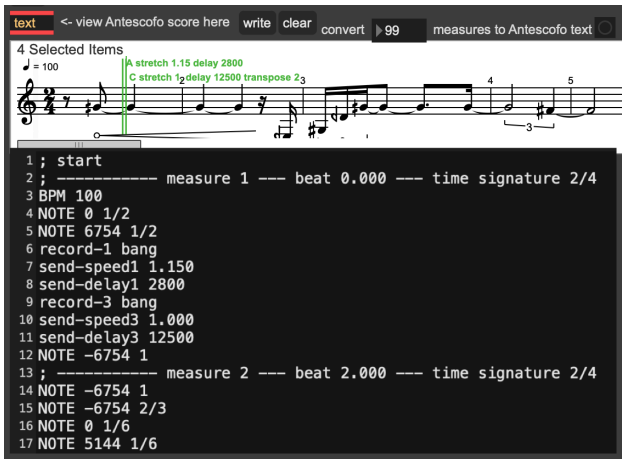


Figure 8. Antescofo~ file: the opening of *Linear A* as a bach.score, followed by a text file formatted specifically for *Antescofo~*.

9. CONCLUSIONS

The integrated workspace I devised for *Linear A* is a first step towards the fantasy of a certain breed of composer, one acquainted with computer-assisted composition: a streamlined and self-contained set of patches where the distance between conception, experimentation, implementation, and finalization is minimized. In this flexible space, timing and tuning are malleable, and a score materializes as a natural result of interactive processes.

There is ample work to be done in further refining the various interfaces introduced here, and there are many others that can be conceived of and introduced into the workflow, in this niche discipline whose horizons will undoubtedly expand in the coming years.

Acknowledgments

The *bach* library, created and maintained by Andrea Agostini and Daniele Ghisi, is an invaluable resource. I would like to acknowledge the contributions of Greg Beller and Diemo Schwarz. Finally, I would like to point interested listeners toward the excellent performance of *Linear A* recorded by Amy Advocat on my CD *Horizontal Drift* [3]. The score is available from Composers Edition [4].

10. REFERENCES

1. The Bohlen-Pierce Site (https://www.huygens-fokker.org/bpsite/), accessed December 19, 2025.

2. N. Müller, "The Bohlen-Pierce Clarinet: Theoretical Aspects and Contemporary Applications. Doctoral dissertation, Hamburg, 2019.

3. C. Trapani. *Horizontal Drift.* CD release: New Focus Recordings, 2022.

4. C. Trapani. *Linear A*. Composers Edition, 2021.

Recursive Radiance: Multimedia Interpretations of Traditional Chinese Aesthetics

Juan Carlos Vasquez
Duke Kunshan University
jv211@duke.edu

Zhonghao Chen
Xi'an Jiaotong Liverpool University
Zhonghao.Chen@xjtlu.edu.cn

ABSTRACT

This paper presents Recursive Radiance, a multidisciplinary artwork integrating traditional Chinese practices with contemporary technologies through parallel sonic and visual implementations. The project pairs a four-channel acousmatic composition with an installation of graphic scores inspired by the Jianzi notation system.

The sonic component transforms improvisations based on the traditional guqin piece "Cai Zhen You" (Wandering in True Essence) through fab synthesis, spatial diffusion, and electronic processing. The composition uses chaotic attractors for spatial movement, creating an immersive soundscape that embodies Daoist principles of fluidity and transformation. The visual component features a series of digital scrolls, functioning as both notation and a parallel artistic dimension. These graphic scores emerge from a hybrid methodology combining traditional ink boxes with tensioned strings, cyanotype printing, 3D environmental scans, and AI-assisted imagery created through LoRA models trained on interpretative readings of the Guqin's notation system. Recursive Radiance functions as both a hybrid physical-digital installation and a framework for cultural preservation.

This paper documents the completed musical composition, graphic scores, and conceptual approach to public engagement through immersive multimedia. Our research demonstrates how computational tools and engineering techniques can support artistic expression while preserving cultural heritage, offering new pathways for audience interaction with traditional art forms through contemporary multimedia experiences.

1. INTRODUCTION

Recursive Radiance represents a novel approach to integrating traditional Chinese artistic practices with contemporary digital technologies. This multidisciplinary project encompasses both an electroacoustic composition and a system of graphic scores, each drawing from historical Chinese traditions while engaging with cutting-edge technological processes. The work explores the creative potential that emerges at the intersection of cultural heritage and technological innovation.

Copyright: ©2026 Juan Carlos Vasquez et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License 3.0 Unported, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

The sonic component of Recursive Radiance originates from a performance and subsequent improvisations based on the traditional guqin composition *Cai Zhen You*, or "Wandering in True Essence", a piece included in the Xilutang Qintong, a compendium compiled in the mid-16th century during the Ming period [1]. The piece is guided by the Daoist concept of "divine radiance" as fluid and genuine, proposed by Cheng Xuaying (a prominent Daoist scholar) in his commentary to Zhuangzi, the foundational text of Daoism [2]. Recursive Radiance reimagines these materials through fab synthesis, a technique that emphasizes the tactile transformation of sound sources into new auditory dimensions [3]. This synthesis creates an intricate interplay between organic acoustic textures and acousmatic abstraction.

It is important to note that the guqin, historically revered as an instrument of spiritual cultivation, has found a renewed voice in contemporary electroacoustic music. Chinese composers have increasingly explored its integration into electronic soundscapes, reflecting broader trends in modern composition. Zhou Qian [4] and others have highlighted how traditional instruments, such as the guqin, are used as vehicles for expressing abstract philosophical ideas and bridging cultural dichotomies. Works like Zhou's *Qu Shui Ming Yun*, which juxtaposes guqin tones with electronic sounds, underscore the instrument's capacity to evoke a dialogue between ancient and modern sound worlds [4, 5].

On the other hand, our visual component for this practice-based research project is manifested as a series of graphic scores inspired by the traditional Jianzi notation system. A tradition evolving since the Tang Dynasty, the Jianzi notation system involves both the notation of pitches and playing techniques combined with Chinese characters [6]. It is also deeply linked to its cultural context: each piece usually contains an article that introduces the story and artistic facets of the piece [7].

Our visual scores emerge through a dynamic interplay between physical and conceptual realities; incorporating traditional ink-based workflows, cyanotype printing, 3D environmental scanning, and AI-generated imagery. They draw inspiration from the wandering practices of Daoist thought and the historical lineage of Shanshui painting, with visual elements that function both as notation for performance and as standalone artistic works: a parallel collaboration of image and sound. Our visual scores are deeply influenced by the Western musical practice of 'graphic scores', in which the 'score' is not necessarily meant to be a mapping of actions or even include any semantic content [8]. As such, the construction of sonic meaning relies heavily

on the performer/interpreter [9]. Similarly, the causality (first the score as instructions to recreate the music piece) is also questioned, allowing for the order of creation of one and the other to be subverted [10]. Recursive Radiance creates a parallel between sonic and visual expressions of the same philosophical principles. Integrating these dual aspects—sonic and visual, traditional and contemporary, physical and digital—makes Recursive Radiance to function as a hybrid physical-digital installation that invites audiences to engage with Chinese cultural heritage through immersive multimedia experiences.

In summary, the technical contribution of Recursive Radiance lies in its novel integration of multiple computational approaches within a conceptual framework derived from traditional Chinese aesthetics. Specifically, we demonstrate: (1) a methodological framework for translating historical notation systems into contemporary computational workflows; (2) the application of chaotic attractors as aesthetic tools for spatial sound diffusion guided by Daoist principles; (3) a customized approach to LoRA model training incorporating field-gathered data and conceptual parameters derived from traditional aesthetics; and (4) a recursive physical-digital-physical workflow that preserves material essence through technological mediation. This suite of techniques creates new possibilities for cultural heritage preservation through technology while avoiding reductive digitization of traditional practices.

2. CULTURAL BACKGROUND, AKA. THE "LOW-TECH" SPHERE

2.1 Philosophical Foundations

As mentioned before, the guqin piece *Cai Zhen You* exemplifies the profound influence of Daoist philosophy on Chinese music and culture. Rooted in the teachings of Zhuangzi, particularly his notions of *wuwei* (non-action) and *ziran* (naturalness), the piece invites listeners to experience a state of harmony and transcendence. Zhuangzi's concepts emphasize the importance of spontaneity and freedom from societal constraints, ideals that are embedded in guqin music. This connection positions the guqin as not merely an instrument but a spiritual tool that facilitates a return to simplicity and alignment with the Dao [11, 12].

These same Daoist principles fundamentally inform traditional Chinese visual arts, particularly Shanshui (mountain and water) painting and calligraphy. The concept of *Tian Ren He Yi*—the intrinsic unity between humanity and nature—serves as a philosophical foundation for both sonic and visual expressions in Chinese artistic traditions [13]. The legend of the Guqin master *Bo Ya*, who attained enlightenment through musical engagement with nature after creative stagnation, illustrates this principle and provides inspiration for the methodological approach of Recursive Radiance [14].

2.2 Notation Systems and Interpretive Freedom

The cultural significance of guqin music lies in its philosophical depth and unique aesthetics, represented by its notation system. The *jianzi* notation system, used to compose and interpret guqin pieces like *Cai Zhen You*, encourages interpretive freedom, mirroring Daoist ideals of spontaneity and individuality [15, 16]. Many compositions, including this one, draw directly from Daoist texts such as Zhuangzi's "The Course of Heaven," where music becomes a metaphor for cosmic harmony and human authenticity [17, 18]. These philosophical foundations render guqin music a powerful medium for exploring self-realization and transcendence [19]. `Figure 1` illustrates the score we followed for the source material to be recorded and later transformed into the audio component of Recursive Radiance.

Traditional Chinese ink-based workflows, including the mentioned Shanshui painting and its contemporary extensions, share this emphasis on interpretive freedom. The Jianzi notation system functions as both a generative and reconstructive archive, providing guidance while allowing for personal expression [6]. In Recursive Radiance, this system serves as a conceptual bridge between sonic and visual components, allowing for the translation of musical ideas into spatial compositions and vice versa.

2.3 Contemporary Relevance

Guqin's role as a "gentleman's instrument" further emphasizes its cultural and spiritual significance. Historically, it was played by scholars and sages who sought intellectual and emotional refinement through music. Compositions like *Cai Zhen You* embody this heritage, blending music with the Daoist pursuit of self-cultivation. The transcription work of guqin masters, such as Wang Duo, ensures the preservation of these philosophical expressions. This continuity highlights the guqin's timeless appeal as a medium for meditative reflection and cultural expression [20, 21].

Similarly, traditional Chinese visual arts have maintained their cultural significance while evolving through engagement with contemporary practices. The integration of oil-based media, such as gel and traditional ink boxes (carpentry tools historically used for precision marking) represents an evolution of visual expression that maintains connections to historical lineage. Engaging with these traditional tools and techniques through contemporary technological mediation has the aim of positioning Recursive Radiance within a continuum of cultural innovation rather than as a rupture from tradition.

Moreover, the guqin is a bridge between ancient and modern interpretations of Chinese philosophy and culture. Contemporary scholars continue to explore its therapeutic and meditative qualities, linking the guqin's aesthetics to well-being. As a result, its integration of sound and philosophy resonates deeply in both traditional and modern contexts, underscoring its role as an enduring symbol of Daoist thought and natural harmony [4].

3. SONIC IMPLEMENTATION

3.1 Recording Session

The recording process for *Recursive Radiance* took place at the Immersive Sound Studio at Xi'an Jiaotong Liverpool University, a facility equipped to handle a variety of recording needs, including those of large ensembles. The studio's live room offered an acoustically controlled environment ideal for capturing both the nuanced tonal qualities of the guqin and the ambient sonic details often deemed

"unwanted," such as friction sounds and position changes. This approach aimed to emphasize the raw materiality of the instrument, aligning with the Daoist principle of embracing natural imperfection (*ziran*). Two AKG 414 microphones were positioned in an A/B configuration at an elevated angle to ensure a balanced stereo image and high-fidelity capture of the instrument's spatial characteristics.

We followed the indications illustrated in the contemporary transcription of *Cai Zhen You* previously mentioned, providing information about string usage, pitch, and position. While the transcription includes suggested tempo and phrasing, it omits precise rhythmic notations, which we stylistically interpreted following the hallmark of traditional guqin performance practices. We were inspired by literature on guqin aesthetics, which values fluidity and individual expression over rigid formal structures [11, 12]. The transcription is divided into five sections, marked by double bars, each featuring distinct textures such as plucked strings and harmonics.

The recording session began with Zhonghao Chen performing a version of the original piece, as shown in Figure 2. This was followed by guided improvisations inspired by the structural and textural characteristics of *Cai Zhen You*. Such improvisational practices are consistent with the guqin tradition, which often involves reinterpreting canonical works to explore new aesthetic possibilities [15]. These improvisations served as the foundation for subsequent sonic transformations.

In the latter part of the session, various objects, including 3D-printed irregular balls, chains, and reeds collected from Tai Lake in Suzhou, were used for Fab synthesis exploration. These objects introduced new timbral possibilities by interacting with the guqin strings, creating dense and intricate textures. This exploration of unconventional techniques mirrors broader trends in electroacoustic music, where physical materials are used to extend the sonic palette of traditional instruments [3, 5].

Although the primary recording session lasted a full day, the preparatory phase—including planning and experimentation—spanned a month. Approximately two hours of continuous music were recorded, yielding a rich repository of raw material for the compositional process. This extensive recording phase not only provided diverse sonic textures but also captured the dynamic interplay between traditional and experimental practices, embodying the spirit of Recursive Radiance.

3.2 Composition

The compositional structure of Recursive Radiance evolves from an unmodified performance of Cai Zhen You into a complex, spatially immersive soundscape. The piece begins with the original recording in its pure acoustic form, serving as an anchor for the unfolding transformations. Gradually, the timbral qualities of the recording are extended and interpolated using sampling techniques, where the input signal is captured at regular intervals and processed to create smooth interpolations between sampled timbres. This process preserves the organic essence of the guqin while expanding its expressive capabilities through digital manipulation.

As the composition progresses, natural harmonic arpeggios introduce the external activators used in the Fab synthesis process. The piece transitions organically into a motor rhythm created by using the rigid segments of dried reed stems as percussive activators. This rhythmic layer develops dynamically, culminating in a texturally dense climax achieved through the use of multi-band distortion, modeled guitar amplifiers, and granular synthesis spatially diffused across the four-speaker setup. The introduction of these processed layers is managed via automated return tracks, ensuring a gradual and seamless buildup of intensity.

Following the climax, the composition moves into a section defined by harmonically dissonant chords. These chords are spatially diffused in a circular pattern, mimicking the brushstrokes of abstract painting, and are processed using tremolo effects modeled from guitar amplifiers. This section highlights the interplay between physical activation and digital processing, emphasizing the transformation of acoustic gestures into abstract sonic landscapes.

In the final section, the piece returns to the original Cai Zhen You recording. However, it is now accompanied by subtle textural remnants of the digital processes employed throughout the piece, creating a sense of closure while retaining the transformative spirit of the composition. The piece concludes at exactly 9 minutes, a duration imbued with symbolic meaning in Chinese culture, where the number nine (*jiu*) represents longevity, completeness, and transcendence. This choice reflects the Daoist themes of continuity and balance that underlie the work

3.3 Spatial Trajectories

The spatial trajectories for Recursive Radiance were crafted using Strange Mod, a Max for Live device developed by Dillon Bastan [22]. Strange Mod employs chaotic attractors, which are mathematical systems that generate nonlinear and non-repeating yet structured paths, to produce three-dimensional coordinates that can be mapped to various parameters, including spatial movement. This device provided an intuitive yet complex means of creating dynamic sound trajectories across the four-speaker setup, enhancing the immersive qualities of the acousmatic soundscape.

The use of Strange Mod aligns with the Daoist conceptual framework underlying the piece. Chaotic attractors inherently embody unpredictability within structure, a principle resonant with the Daoist notion of "wandering", particularly its interpretation as an spontaneous and unforced movement that unfolds within a natural, guiding order. By utilizing chaotic attractors, the spatial diffusion of sound in Recursive Radiance mirrors the philosophical themes of transformation and organic flow central to its design.

There are precedents for employing chaotic attractors in electroacoustic music. Miroslav Spasov's work, for example, demonstrates how strange attractors can be used to control sound processing in live electroacoustic compositions, offering nonlinear dynamical behavior that enhances interactivity and sonic variability [23]. In Recursive Radiance, the application of chaotic attractors provides endless variability in spatial movement, allowing the sound materials to "wander" unpredictably yet cohesively within the sonic space. This approach not only enriches the immersive experience but also extends the expressive capacity of

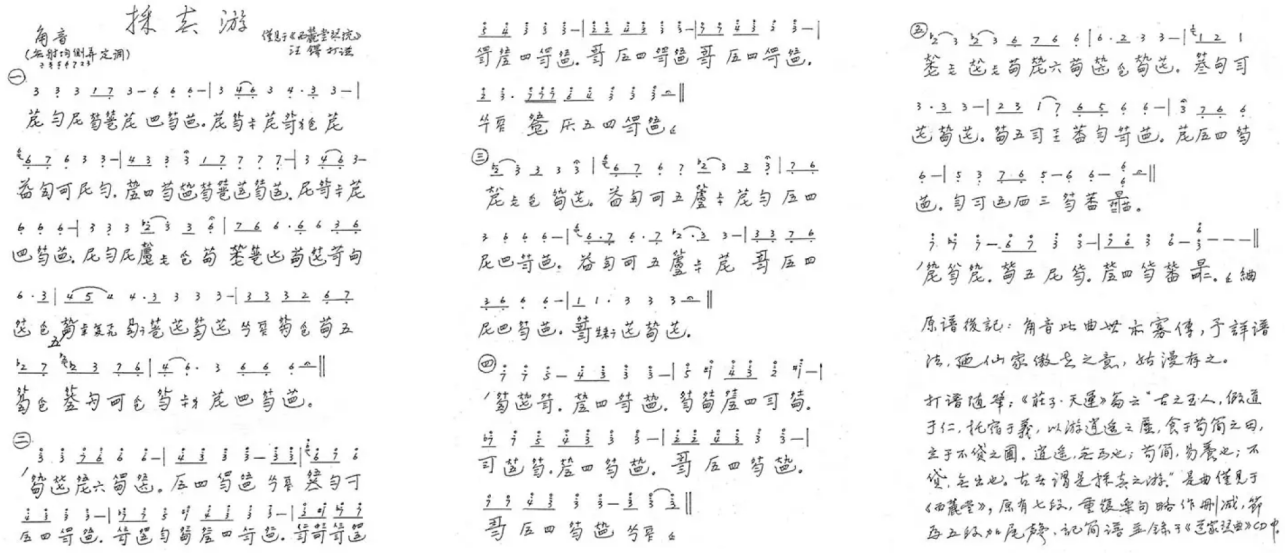


Figure 1. Contemporary transcription of Cai Zhen You. A poem is the main indicator for progression, accompanied by information about pitch, string, and finger position.



Figure 2. Recording of the Guqin in its traditional performance configuration. The location of the microphones are also in display



Figure 3. Implementation of Fab Synthesis

the acousmatic medium, reflecting a broader trend in contemporary electroacoustic practices.

4. VISUAL IMPLEMENTATION

4.1 Studio Exploration: Traversing Ink and Materiality

The visual component of Recursive Radiance was developed through a process that parallels the sonic exploration, blending traditional techniques with contemporary technological approaches. Initial work began with traditional ink media experimentation on Xuan paper–a fine-textured paper used in ancient China for painting–within the studio setting, establishing a foundation in historical Chinese artistic practices that would later be extended through digital means.

4.1.1 Ink Experimentation

Initial explorations translated Jianzi notation into intuitive mark-making on Xuan paper using both water- and oil-based media. This drawing process yielded atmospheric visuals reminiscent of Shanshui traditions while functioning simultaneously as process documentation and potential notation for future musical interpretations.

This approach reflects contemporary practices in visual music notation, where traditional scoring systems are reimagined through experimental techniques that emphasize process and materiality. The resulting ink drawings function as both documentation of creative process and as potential scores for future interpretations of the music, creating a recursive relationship between sonic and visual expression

4.1.2 Ink Box as Embodied Notation

A significant innovation in our methodology was the employment of the ink box, a traditional woodcraft tool utilized by Chinese carpenters for centuries. We repurposed this tool as a mechanism for creating visual scores that directly embody sonic principles. The ink box contains a

tensioned, ink-soaked string that, when stretched above the surface of paper and then released—much like plucking a musical instrument—generates precise, resonant line imprints. This process transformed the act of notation into a performance itself, as the physical gesture of "playing" the ink box directly produced mark-making that visually represented sonic vibration. These marks were strategically recorded on specific sections of the paper, constituting an embodied notation system that echoes the open-ended structure of Recursive Radiance. The resulting linear compositions serve as visual analogues to the acoustic tensions inherent in Cai Zhen You, creating a direct material relationship between sound and visual representation. This methodology stands in contrast to conventional Western notation systems by emphasizing the gestural and performative aspects of sound-making rather than symbolic representation.

4.2 Field Research and Environmental Integration

While studio exploration provided one trajectory for visual development, parallel field research established a complementary approach grounded in direct environmental engagement, reflecting the Daoist principles that underpin Cai Zhen You.

4.2.1 Site Selection and Conceptual Significance

Tai Lake in Suzhou, China was selected as the primary site for field research, offering multiple layers of relevance to the project. Historically, the region is recognized as the birthplace of the Wu Men school of Guqin and home to significant historical and contemporary masters.

The research approach at Tai Lake involved both direct bodily immersion and technologically mediated experiences, creating a methodology that integrated physical presence with digital augmentation. This interweaving of immediate experience and technological mediation enabled the generation of content that captures the fluid impressions of space, time, and gesture that characterize the guqin tradition.

4.2.2 Cyanotype and Photographic Documentation

On-site cyanotype printing emerged as a central methodology for directly engaging with the environment. This light-sensitive technique allowed us to capture the shifting light dynamics and environmental textures of Tai Lake without digital intermediaries. Organic elements such as leaves and insects left tangible imprints on treated surfaces, rendering the lake an active participant in the creative process rather than merely a subject of representation. This method foregrounded co-creative engagement with nature, embodying the Daoist principle of Tian Ren He Yi through direct material collaboration with the environment.

Alongside cyanotype processes, high-resolution photography and 3D scanning documented the environment in ways that extended beyond human perception. Photographic documentation preserved the interplay of light, shadow, and texture across the landscape, while 3D scanning translated the physical topology of Tai Lake's shoreline, vegetation, and organic formations into manipulable digital models. This comprehensive approach to environmental documentation created a rich archive of visual material that would later inform the development of the graphic scores.

4.3 Technological Mediation and Digital Transformation

The transition from field research to completed visual scores involved significant technological mediation, establishing a recursive dialogue between traditional practices and contemporary digital methods.

4.3.1 AI-Assisted Visual Development

Once a comprehensive digital archive of field documentation was established, we employed ComfyUI to generate images through a flux-based workflow that complemented the wandering aesthetic of the project. Specific LoRA (Low-Rank Adaptation) models were trained on categorized image datasets collected during field research, allowing for the creation of AI-generated visuals informed by interpretative readings of Cai Zhen You.

This approach to AI implementation stands apart from conventional generative methods by maintaining a direct material connection to both the physical environment and the musical source material. The training datasets were specifically organized to reflect the structural characteristics and philosophical underpinnings of Cai Zhen You, ensuring that the resulting images would not function as arbitrary visual content but as meaningful extensions of the composition's aesthetic principles.

4.3.2 Physical-Digital Recursive Translation

The process of creating the final visual scores involved a continuous recursive dialogue between digital and physical domains. Photographic images and 3D scans were transferred onto prepared surfaces such as fabric and paper, where they underwent further manipulation that enhanced their texture and depth. This approach aligns with the transmutative practices of contemporary artists who interrogate the interstitial space between digital and physical realms, creating work that exists simultaneously in both domains. Figure 4 illustrates six final graphic scores resulting from the process described so far.

The physical manipulation of digitally transferred images created new visual information that was subsequently re-digitized, establishing a feedback loop between material and virtual processes. This recursive methodology mirrors the philosophical structure of Cai Zhen You itself, which emphasizes continuous transformation and the fluidity of boundaries between seemingly distinct categories.

4.4 Spatial Configuration and Installation Design

The final implementation of the visual component manifests as an installation that extends beyond conventional gallery presentation formats, creating a spatial environment that invites audience engagement and embodies the wandering practices central to Daoist thought.

4.4.1 Scroll Composition and Fragmentation

The final installation is comprised of nine large hanging scrolls structured through Jianzi notation and incorporating three visualization layers: calligraphic gestures, ink-line drawings as embodied movement, and embedded Jianzi notation. A constellation of smaller suspended fragments

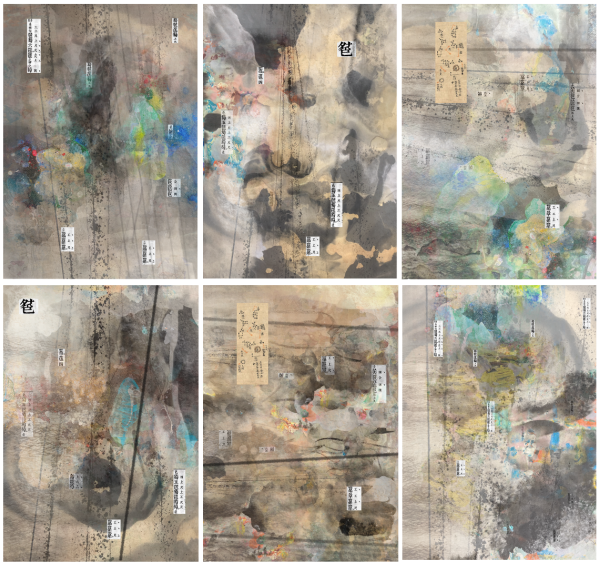


Figure 4. Final graphic scores for Recursive Radiance

creates spatial dialogue that requires audience movement for complete apprehension, referencing contemporary installation practices like Sarah Sze's spatiotemporal explorations [24].

Complementing these primary scrolls, a constellation of smaller fragments suspended in spatial dialogue create a dynamic visual environment that cannot be apprehended from a single viewpoint. These fragments reference Jianzi notation while evoking the liminal space between nature and gesture–This fragmentation methodology reconfigures spatial compositions through image dissection and reconstruction, producing a dynamic oscillation between presence and dissolution.

4.4.2 Space-Time Notation and Audience Agency

The resulting installation represents a space-time notation system that disrupts linear perception, generating a fluctuating experience of space, time, and gesture. Through the integration of traditional methodologies, technological mediation, and embodied practice, the installation invites the audience into an interactive field where notation, materiality, and perception converge in continuous transformation.

The visual implementation thus functions not merely as documentation or representation of the sonic component, but as a parallel creative trajectory that embodies the same philosophical principles through different material means. By integrating traditional Chinese practices with contemporary technologies and methodologies, the visual scores establish new relationships between historical lineages and current artistic practices, offering audiences multiple points of entry into the conceptual world of Recursive Radiance. Figure 5 showcases the AI-assisted sketch for deployment, and Figure 6 illustrates the full workflow of Recursive Radiance

5. CONCLUSIONS

Recursive Radiance demonstrates how traditional aesthetics can inform contemporary multimedia practice through



Figure 5. Concept of the final installation, integrating both the audio and visual components of Recursive Radiance

parallel sonic and visual implementations. The integration of the guqin's historical lineage with modern technological approaches creates a recursive dialogue between past and present, establishing new pathways for cultural preservation that avoid mere documentation in favor of active reinterpretation. This approach acknowledges the philosophical foundations of Chinese artistic traditions while engaging with current technological possibilities.

The sonic component transforms the traditional guqin piece Cai Zhen You through fab synthesis and spatial diffusion, creating an immersive sonic environment that embodies Daoist principles of fluidity and transformation. The use of chaotic attractors for spatial movement mirrors the philosophical concept of "wandering" central to the source material, allowing sound to move unpredictably yet cohesively within the sonic space. This methodology respects the interpretive freedom inherent in the guqin tradition while extending its expressive capabilities through contemporary means.

Similarly, the visual component establishes multiple layers of engagement with traditional practices through its innovative methodology. The repurposing of the ink box as a performative notation tool transforms traditional craftsmanship into a contemporary artistic practice, while the integration of field research at Tai Lake grounds the project in environmental engagement reflective of Daoist principles. The resulting graphic scores function both as notation and as standalone artistic works, creating a parallel visual expression of the same philosophical principles that underpin the sonic component.

Recursive Radiance ultimately points toward new possibilities for multimedia art that honors cultural heritage while embracing technological innovation. Through its multidisciplinary approach, the project suggests how computational tools and engineering techniques can support artistic expression that remains faithful to traditional philosophical principles while creating experiences that resonate with contemporary audiences. This framework for cultural engagement through immersive multimedia offers valuable insights for artists and researchers exploring the intersection of traditional practices and emerging technologies.

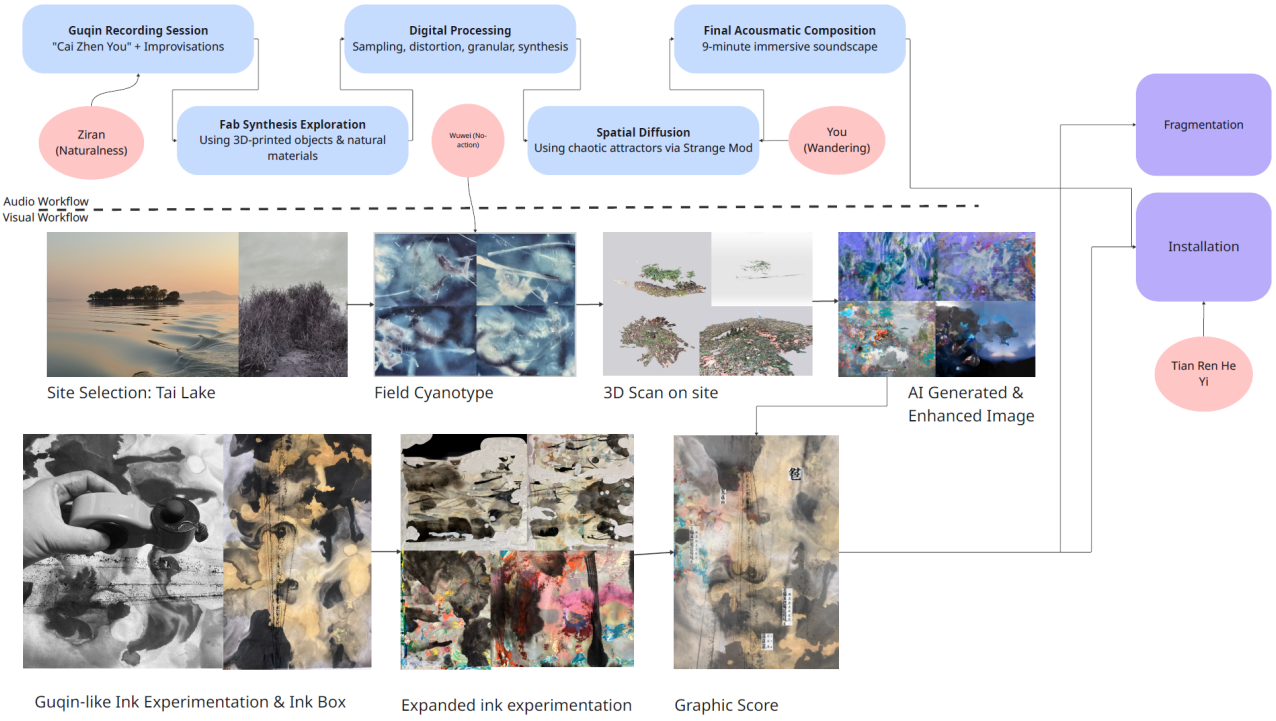


Figure 6. Recursive Radiance: Workflow and Conceptual Framework, including the correlation with the philosophical concepts involved.

6. REFERENCES

[1] U. Theobald. (2025) Xilutang qintong. [Online]. Available: https://m.guoqinwang.com/qu/2155.html

[2] M. Harack, "Cheng Xuanying's Conception of the Sage in the Zhuangzi," Ph.D. dissertation, 2007.

[3] P. Kokoras, "Fab Synthesis: Performing sound, from Musique Concrète to Mechatronics," *Musica/Tecnologia Music/Technology*, p. 103, 2022.

[4] M. Battier, "Crossing Lines: Stylistic gradients in Chinese electroacoustic music," *Organised Sound*, vol. 27, no. 3, pp. 284–293, 2022.

[5] Z. Qian, "Compositional Techniques and Sound Spatialisation in Shanghai-Based Electronic Music since 2000," *Organised Sound*, vol. 27, no. 3, pp. 294–306, 2022.

[6] B. Cao, *Tradition and Change in the Performance of Chinese Music*. Harwood Academic Publishers, 1998, vol. 2.

[7] Y. U. College for Overseas Education, "Chinese Culture Guqin," Mar. 2022. [Online]. Available: https://coe.yzu.edu.cn/en/info/1033/1330.htm

[8] J. Ganter and R. MacDonald, "Graphic scores: A collaboration between print and music," in *Impact 9 International Printmaking Conference*, 2015.

[9] I. O. Hayes, *Scoring the Unknown: Rethinking Fixity and Openness in Western Art Music Notation*. University of California, Irvine, 2023.

[10] A. Kim, "On Musical Notations," 2021.

[11] M.-Y. Lee, "Concept of Nature in the Musical Aesthetics of the Chinese Guqin," *Journal of Comparative Literature and Aesthetics*, vol. 46, no. 1, pp. 161–171, 2023.

[12] R. Mazur, "Music as a Tool for Self-Realization in Chinese Culture: Based on the Practice of Playing the Guqin," 2015.

[13] D. Brubaker, "Chen, Wangheng, Chinese Environmental Aesthetics, translated by Feng Su, edited by Gerald Cipriani," *Dao: a Journal of Comparative Philosophy*, vol. 16, no. 2, p. 281, 2017.

[14] A. Katz-Goehr, "Why did Yu Boya break the qin?" *Journal of Sino-Western Communications*, vol. 7, no. 1, p. 78, 2015.

[15] M. Chinn, "Representing the Great in Music," *Philosophy East and West*, vol. 71, no. 1, pp. 173–192, 2021.

[16] Y. Chen and P. Boonsrianun, "Transmission of Guqin Knowledge and Literacy by Changchao Lu." *International Journal of Education and Literacy Studies*, vol. 12, no. 1, pp. 170–176, 2024.

[17] Y. Fan, "Philosophical speculation in traditional Chinese music," *Trans/Form/Ação*, vol. 47, no. 4, p. e0240069, 2024.

[18] D. E. Cooper, "Music, Nature and Trasncendence," *Rivista di estetica*, no. 80, pp. 48–64, 2022.

[19] H. Deng, *Making the Intangible Tangible: Rediscovering Music and Wellbeing through the Guqin Culture of Modern China*. The Florida State University, 2020.

[20] H. R. A. Gaywood, “Guqin and Guzheng: the historical and contemporary development of two Chinese musical instruments,” Ph.D. dissertation, Durham University, 1996.

[21] M. Zhang, *The Qin and Literati Culture in Song China*. University of California, Los Angeles, 2019.

[22] D. Bastan. (2025) Strange Mod. [Online]. Available: https://dillonbastan.com/store/maxforlive/index.php?product=strange-mod

[23] M. Spasov, “Using strange attractors to control sound processing in live electroacoustic composition,” *Computer Music Journal*, vol. 39, no. 3, pp. 25–45, 2015.

[24] F. V. Josenhans, “Sarah Sze,” *Yale University Art Gallery Bulletin*, pp. 16–23, 2017.

Gestalt: A Symbiotic Framework for Real-Time Collaboration between Performers and Mass Audiences

Sitong Wu
Communication University of China
sitongwu@mails.cuc.edu.cn

ABSTRACT

This paper presents Gestalt, a real-time co-creative audiovisual performance system for professional performers and large-scale audiences. To address participation barriers, interaction latency, and unequal creative agency in Networked Music Performance (NMP), Gestalt adopts a browser-based heterogeneous architecture: performers retain structural control via MediaPipe-based motion capture, while 50–200 audience members participate through a mobile web multi-touch interface.

Centered on a mechanism termed "Translation," the system performs a dual reconstruction. On the audio side, an activity-weighted aggregation algorithm transforms large volumes of discrete gestures into coherent musical textures. On the visual side, audience touch inputs are streamed in real time to a physics-driven WebGL particle stage, translating collective crowd activity into ordered audiovisual forms.

Technically, the web frontend connects to a Max/MSP audio engine via OSC (Open Sound Control), and to the visual stage via WebSocket. Benchmark tests and pilot workshops examine how the architecture can preserve performer-led form while enabling audience aesthetic agency. Gestalt is released as an open-source platform for future interactive media creation.

1. INTRODUCTION

1.1 From Passive Listening to the "Web Agora": A Historical Trajectory of Audience Participation

Breaking the "fourth wall" between performers and audiences has long been a central concern in the evolution of computer music and interactive arts. Traditional concert models often position audiences as passive recipients, whereas Networked Music Performance (NMP) has sought to enable a decentralized "Web Agora"[1]. This idea can be traced back to early network computer music practices such as The Hub in the 1980s, which used computer networks to support decentralized collaboration among musicians [2]. With the proliferation of mobile communication technologies, attempts to extend such agency to large-scale audiences became increasingly feasible.

Copyright: © 2025 Sitong Wu. This is an open-access article distributed under the terms of the Creative Commons Attribution License 3.0 Unported, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Golan Levin's Dialtones: A Telesymphony (2001) is a landmark example in this trajectory. By triggering ringtones through pre-dialing, the work orchestrated a "symphony"of 200 mobile phones [3]. However, the audience primarily functioned as "sound emitters" rather than performers. Later, Echobo by Lee and Freeman (2013) leveraged smartphones and a native application to allow audiences to control pitch and rhythm, shifting participation from passive sound emission toward more active musical control [4]. Nevertheless, native app-based approaches can introduce substantial participation barriers in real-world deployments: requiring hundreds of attendees to download and install specific software reduces the fluidity of impromptu participation, and cross-platform maintenance increases deployment cost and complexity [5].

More recently, the Soundworks framework by Robaszkiewicz and Schnell has demonstrated the maturity of web-based participatory systems [6]. Using HTML5 and the Web Audio API, audiences can join performances by scanning a QR code, enabling a "zero-installation, instant-access" workflow. Despite these advances, an open challenge remains: how to support low-latency interaction for hundreds of concurrent audience participants while maintaining audiovisual coherence and artistic expressivity.

1.2 Core Contributions of the Gestalt System

To address these challenges, Gestalt contributes system designs and methods across three aspects:

Architecture: We introduce a bilayer heterogeneous collaboration architecture inspired by edge computing. By offloading parts of interaction processing to mobile clients and decoupling audio and visual data streams, the architecture supports low-latency operation under high concurrency with a unified deployment workflow across local and remote setups.

Interaction: We propose an activity-weighted aggregation mechanism that combines continuous parameter control with discrete event triggering. A multi-factor activity model is integrated with a touch-to-trigger event channel to improve participatory fairness while addressing the tension between continuous modulation and instantaneous triggering in collective interaction.

Tooling: We present a WYSIWYG (What You See Is What You Get) integrated visual monitoring and mapping toolchain. A real-time feedback stage visualizes audience touch inputs as dynamic particles, coupled with a configuration-driven, declarative mapping engine. This approach elevates the audience from "invisible sound emitters" to "visible performers," while significantly reducing

the technical barrier for artists designing and iterating complex audiovisual mappings.

2. SYSTEM ARCHITECTURE AND INTERFACE INTRODUCTION

2.1 Overview

As shown in Figure 1, Gestalt follows a four-layer architecture comprising a user interaction layer, a Node.js middleware server, an audio synthesis layer (Max/MSP), and a visual feedback layer.

The user interaction layer includes a performer client (port 3000) and an audience client (port 3002). The performer client uses MediaPipe for real-time pose and hand tracking and maps gestural features to control parameters via a client-side mapping engine (OSCExporter). The audience client supports 50-200 concurrent participants via a mobile web interface; each device runs TouchGestureDetector and TouchOSCMapper locally to map touch input to control parameters, offloading computation from the server (see Sec. 3.2).

The middleware server routes WebSocket messages and converts them to OSC. Performer data are forwarded directly to minimize latency, whereas audience data are processed by a weighted-sampling aggregation algorithm with explicit handling of one-shot trigger events. The audio engine receives OSC via separate Max/MSP udpreceive ports (7400 for performer, 7402 for audience), preserving performer-level structural control while enabling collective audience modulation (see Sec. 3.1). In parallel, the server periodically broadcasts an `audience_details` message via WebSocket containing per-user metrics (position, direction, velocity, intensity); the monitoring panel renders one particle per audience member, synchronized 1:1 with touch gestures, making participation visually explicit.

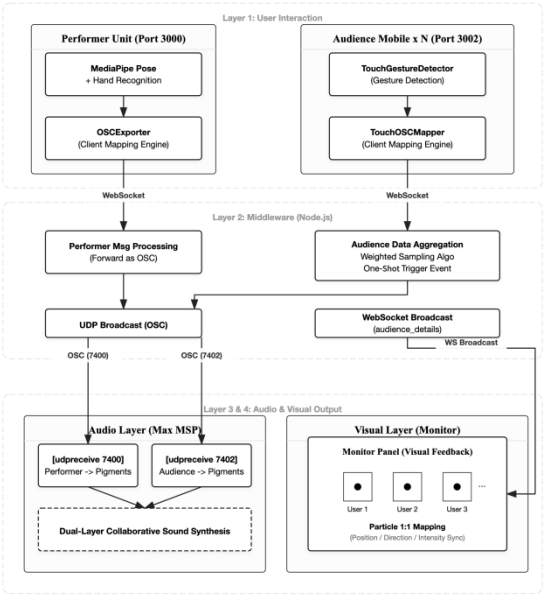


Figure 1. System architecture diagram.

2.2 Performer Motion Capture System

2.2.1 Interface Design and Functional Integration

The performer interface integrates real-time feedback and system control in a single workspace. The central viewport overlays 33 body landmarks and skeletal connections on the live video stream using an HTML5 Canvas, supporting intuitive performer control and monitoring of a "digital twin." Peripheral modules provide real-time network and system status (audience count, OSC connectivity, FPS), normalized parameter displays (0.00–1.00), and entry points to the mapping editor. The layout is optimized to prioritize information access and situational awareness during performance (Figure 2).

The mapping editor is implemented as a modal dialog, enabling performers to adjust gesture-to-audio parameter mappings in real time during rehearsals (see Sec. 2.4).

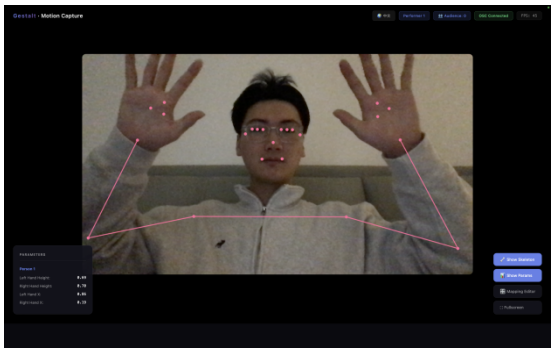


Figure 2. Performer interaction interface.

2.2.2 Pose Estimation Optimizations for Musical Interaction

To satisfy a sub-50 ms camera-to-OSC latency target in a web-based performance setting, we configure MediaPipe with the following optimizations:

Model Selection and Hardware Acceleration: We use the Pose Landmarker Lite model with GPU delegation enabled (WebGL backend). Compared to the Full model, this configuration trades approximately 15% keypoint localization accuracy for a ~5-fold improvement in inference speed. This speed-first choice matches musical control needs, where temporal responsiveness is typically more critical than millimeter-level spatial precision.

Macro-Micro Dual-Scale Control: Pose estimation and hand tracking run in parallel. Body pose features drive macro-level, gradual parameters (e.g., overall gain, filter cutoff), while hand features modulate micro-level timbral parameters (e.g., reverb depth, panning modulation rate). This design allows broad gestures to shape overall musical direction while finer hand movements sculpt subtle timbral variation.

Signal Post-Processing: To reduce visual-tracking jitter, we apply exponential moving average (EMA) smoothing and confidence gating (threshold = 0.5). This combination balances noise suppression with transient responsiveness and mitigates control artifacts caused by landmark drift.

With these settings, the system sustains 40 – 60 FPS on Apple M1 devices and achieves camera-to-OSC latency below 50 ms.

2.3 Audience Interaction and Visual Feedback

2.3.1 Mobile Touch Interaction Panel

The audience interface is implemented in WebGL (Three.js) and runs in mobile web browsers, providing lightweight 3D feedback (Figure 3). Following a "What You See Is What You Touch" principle, touch, swipe, and release actions generate real-time trail particles and burst effects, making the action-feedback relationship perceptually explicit.

The UI also communicates participation state and control intent: a status bar indicates connection state and the current number of online participants, an intensity meter (0-100%, cyan-purple) reflects interaction strength and supports an intuitive "volume contribution" metaphor, and a gesture display reports the detected gesture type (e.g., swipe up/down) to clarify which parameter mapping is active.

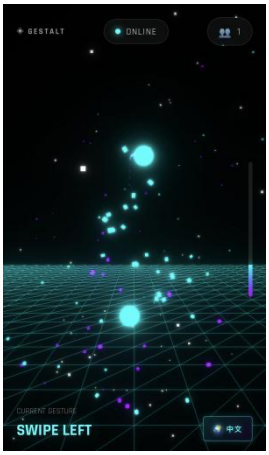


Figure 3. Audience mobile touch interaction panel.

2.3.2 Audience Monitoring System

The audience monitoring interface (Figure 4) is a full-screen venue projection that visualizes audience participation as a particle-based representation, making collective interaction legible. The interface is rendered with Canvas 2D.

A starfield and perspective grid provide a consistent background with parallax depth cues. Each connected audience member is mapped to a single particle driven by per-user data received via WebSocket (position, gesture type, intensity, speed, direction), supporting audiovisual coherence. Particle positions follow mobile touch coordinates; gesture categories are encoded by color (Up: cyan; Down: purple; Left: orange; Right: green; Stationary: blue). Interaction intensity and velocity modulate visual emphasis (e.g., halos/trails and burst-like sparks). When active particles are in close proximity, semi-transparent links are drawn to form an emergent network structure that reflects implicit coordination.

The interface also serves operational needs by presenting network topology, concurrent connection counts, and QR-code access guidance, and by providing access to the mapping editor for in-performance adjustments—supporting a closed-loop workflow from state monitoring to dynamic intervention.

Reflecting on the evolution of NMP, the design of the audience monitoring system provides three layers of innovative value. For participating audience members, seeing the real-time evolution of their own particles establishes a clear "action → sound" causal link. For the performer, the spatial distribution and color patterns of the particles indicate the direction of collective affect, which can in turn inform performance adjustments. For on-site spectators, the evolving particle swarm itself functions as a generative stage visual artwork. Taken together, this integrated visual feedback forms a core component of the system's co-creative experience.

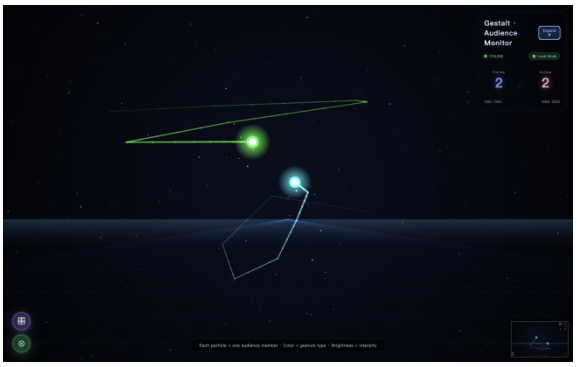


Figure 4. Audience monitoring system.

2.4 The Mapping Editor

2.4.1 Default Interaction Mapping Scheme

The system provides a pre-configured default mapping scheme for the performer and audience subsystems (See Table 1 and Table 2). Verified through iterative testing, the scheme establishes a control logic for the performer wherein "limb amplitude governs macro-level parameters, while hand details modulate micro-level timbre." For the audience, the system implements a parallel control hierarchy combining discrete note triggering (via the on-screen keyboard) with continuous timbre modulation (via swipe gestures). Artists may use this configuration directly or adapt it as a foundation for custom mappings.

Input	OSC Address	Effect
Left Hand Height	/pigments/VC	Engine1Vol
Right Hand Height	/pigments/VM	Engine2 Vol
Arm Span	/pigments/CUTOFF1, /pigments/CUTOFF2	Filter Control
Left Palm Openness	/pigments/reverb	Reverb Wet/Dry
Right Palm Openness	/pigments/rate	Panner Rate
Motion Speed	/pigments/midi1, /pigments/midi2	MIDI Input

Table 1. Performer default mapping (MediaPipe to Pigments synthesizers). EMA $\alpha = 0.3$

Input	OSC Address	Effect

Screen Keyboard	/pigments/midi	MIDI Note (C2–B6)
Swipe Up	/pigments/V1, /pigments/V2	Engine Vol
Two-Finger Distance	/pigments/reverb	Reverb
Gesture Energy	/pigments/rate	Panner Rate

Table 2. Audience default mapping (touch gestures to Pigments synthesizers).EMA $\alpha = 0.3$ for all continuous mappings; not applied to the screen keyboard.

2.4.2 GUI Mapping Editor

To reduce the technical barrier for artists, the system provides a graphical user interface (GUI) mapping editor that enables non-programmer users to customize interaction mappings without modifying source code.

The editor follows a declarative configuration approach: artists specify only "what input maps to what output," without implementing any processing logic. Each mapping entry contains six fields:

- *Gesture/Action:* name of the input parameter (e.g., swipe up).
- *Enable:* whether the mapping is active.
- *OSC Address:* output destination; multiple comma-separated addresses are supported for simultaneous output.
- *Input Range:* expected range of the raw input (e.g., [0, 1]).
- *Output Range:* target range of the mapped OSC value (e.g., [0, 127]).
- *Smoothing:* whether exponential moving average (EMA) smoothing is enabled, and its coefficient.

The GUI is organized around these fields. A generic mapping engine reads the configuration and automatically performs range scaling, EMA smoothing, and OSC message formatting. The editor also supports configuration management via three buttons at the top: Export (save the current configuration as a JSON file), Import (load a configuration from a JSON file), and Reset (restore the default configuration).

In addition to the default mapping parameters, the system exposes optional parameters that can be enabled as needed. Performer-side options (3 parameters) include left-hand horizontal position, right-hand horizontal position, and body tilt. These parameters can be used to control spatial effects (e.g., panning and stereo width) but are disabled by default to avoid excessive parameterization. Audience-side options (7 parameters) include swipe down, swipe left, swipe right, touch X coordinate, touch Y coordinate, swipe velocity, and finger count. Together, these cover the full set of four-directional swipe gestures and raw touch features. Artists may enable them to suit specific compositional goals (e.g., mapping directions to different timbral controls, or using finger count to control chord density).

Configuration changes are automatically persisted in browser localStorage and remain unchanged after page refresh. Mapping updates take effect immediately and synchronize across devices. On the performer client, configuration edits trigger a custom event, prompting the OSC export module to apply the new mapping without restarting the page. On the audience side, configuration edits are broadcast via WebSocket to all connected devices, ensuring a consistent mapping scheme—when the artist updates the configuration at the console, all audience phones update immediately.

2.4.3 OSC Reception Configuration (Max/MSP)

In the declarative mapping workflow, OSC addresses defined on the web frontend must match the Max/MSP routing exactly. In our Max/MSP + Pigments setup, performer and audience streams are separated by udpreceive ports 7400 and 7402. Incoming messages are routed by OSC address (e.g., /pigments/VC, /pigments/CUTOFF1), which must match the GUI mapping configuration (including case and slashes). Routed values are then bound to Pigments parameters via [parameter_ID $1] (e.g., 200 $1, 547 $1), using IDs from the Pigments parameter index. This "web address declaration → Max/MSP routing → synthesizer binding" chain allows mappings to be updated by changing OSC addresses in the GUI and updating the corresponding Max route entries, without rewriting processing logic.

3. UNDERLYING TECHNIQUES AND IMPLEMENTATION

3.1 Extensible Bilayer Collaborative Performance Architecture: Multi-Voice Orchestration with an Independent Audience Part

In conventional NMP systems, audience data are typically merged into the performer's control stream as modulation parameters. In practice, the audience primarily "assists" the performer's sound production rather than performing an independent musical part. The bilayer architecture proposed here fundamentally changes this paradigm. This section describes the logical allocation of musical agency between the performer and the audience.

By assigning independent OSC ports and separate synthesizer instances, each udpreceive object can be connected directly to its corresponding synthesizer, enabling plug-and-play modular configuration. The performer and the audience therefore control distinct voices, supporting polyphonic, contrapuntal textures. Unlike the traditional "performer-led, audience-assisted" model, this design grants the audience voice-level control, enabling independent adjustment of pitch, timbre, filtering, and other synthesis parameters, and establishing a peer-to-peer musical dialogue with the performer. At the orchestration level, this supports a more democratized form of collaboration: the audience shifts from passive "parameter influencers" to active "voice performers." Meanwhile, the performer may still guide the audience as needed in live contexts, enabling coordinated co-performance.

The architecture is extensible: port 7401 is reserved for a potential second performer, allowing seamless expan-

sion to a three-layer configuration (dual performers + audience).

3.2 Client-Side (Edge) Computation and Audiovisual Data-Stream Decoupling

The data-processing architecture of this interactive system must satisfy two seemingly conflicting requirements. Audio control requires aggregating input from hundreds of audience members into manageable control signals. In contrast, visual feedback requires preserving per-user data to achieve 1:1 particle mapping. This section describes how the system addresses both requirements through client-side (edge) computation and audiovisual data-stream decoupling.

3.2.1 Client-Side (Edge) Computation: Pushing Mapping to the Edge and Distributing Load

In a traditional centralized architecture, raw touch data are uploaded to the server for unified processing. When the audience size exceeds approximately 100 users, the server must execute thousands of mapping operations per second. More critically, parameter mapping involves compute-intensive operations such as range scaling, smoothing filters, and multi-address output. If handled entirely on the server, these operations become a scalability bottleneck.

The proposed system adopts a push-to-edge strategy by offloading parameter mapping to each audience member's mobile device. As described in Section 2.1, each device runs two modules independently—TouchGestureDetector (gesture recognition) and TouchOSCMapper (parameter mapping)—performing the complete transformation from raw touch events to OSC control values locally. The server receives only lightweight, pre-mapped data, thereby distributing computational load across a large number of edge devices.

3.2.2 Audiovisual Data-Stream Decoupling: A Bilayer Heterogeneous Design

To address the challenge of maintaining coherent audiovisual feedback under large-scale audience participation, this paper proposes a bilayer heterogeneous data-stream design.

Audio Control Stream: This stream transmits client-side-mapped OSC parameter values. Upon receipt, the server performs activity-weighted aggregation (see Sec. 3.3), fusing inputs from hundreds of participants into a single control signal, which is then broadcast via UDP to Max/MSP to drive the collective voice.

Visual Feedback Stream: This stream transmits raw touch features (position, direction, velocity, intensity). The server does not aggregate these data; instead, it periodically broadcasts an `audience_details` message via WebSocket, preserving each audience member's individual information. The monitoring panel then renders an independent particle for each user, achieving 1:1 visual mapping.

The rationale behind this heterogeneous stream separation is that audio control inherently requires many-to-one aggregation, whereas visual feedback requires one-to-one mapping—two fundamentally conflicting requirements. By processing both streams in parallel, the system maintains a smooth audiovisual experience even with 200 concurrent participants: the monitoring panel renders at 60 fps, the mobile interface renders at 30 fps, the audio aggregation period is 50 ms, and typical touch-to-OSC end-to-end latency is approximately 100 ms (estimated from touch acquisition, client-side processing, network transmission, server aggregation, and OSC emission). This design fundamentally resolves the dilemma faced by conventional systems, which often yield either "visually rich but sonically chaotic" or "sonically precise but visually monotonous"outcomes.

3.3 Multi-Factor Activity-Weighted Aggregation Algorithm: Balancing Fair Participation and Artistic Expression

In large-scale audience-participatory performances, a core technical challenge is how to effectively fuse asynchronous inputs from tens to hundreds of audience members into a single audio control stream. Simple averaging can cause the "silent majority" to dilute the contribution of actively engaged participants, whereas random sampling may omit critical artistic expressions. To address this, the system implements a multi-factor activity-weighted aggregation algorithm that balances participatory fairness and artistic expression by granting greater influence to more actively interacting users.

3.3.1 Activity Scoring Model

The system computes an activity score S for each online audience member in real time by combining four independent factors:

$$S = w_1 \cdot F_t + w_2 \cdot F_i + w_3 \cdot F_v + w_4 \cdot F_s \qquad (1)$$

The factors are defined as follows:

- F_t(Time Freshness): measures data recency.
- F_i(Intensity): reflects touch pressure and contact area.
- F_v(Velocity): measures gesture dynamics; raw velocity is scaled by 2× and then clamped to 1.
- F_s (Activity State): set to 1 for dynamic gestures and 0.3 for stationary contact. This factor ensures that stationary participants retain a sense of participation (30% weight) rather than being excluded entirely.

The default weight configuration is: w_1= 0.4, w_2= 0.3, w_3= 0.2, and w_4= 0.1. This configuration prioritizes data recency and interaction intensity, consistent with the real-time responsiveness required for musical control. The activity score S is used to determine who enters the sampling stage for aggregation.

3.3.2 Sampling Strategy and Performance Optimization

When the audience size exceeds 80, the system ranks all audience members by activity score in descending order and samples only the top 80 most active participants for aggregation, ensuring that aggregation latency remains in the millisecond range.

3.3.3 Weighted-Average Aggregation

For participants in the sampled set, the system uses interaction intensity I_{i_i} as the aggregation weight and performs weighted aggregation per OSC address:

$$V = \Sigma(v_{i_i} \cdot I_{i_i}) / \Sigma(I_{i_i}) \quad (2)$$

where v_{i_i} is the mapped parameter value of participant i and I_{i_i} is the corresponding interaction intensity. When I_{i_i} is below 0.1, that participant is excluded from aggregation to avoid noise interference. This algorithm balances "participatory democracy" with "expressive priority": each audience member's touch input is included in the computation, but more active participants exert greater influence on the musical output. A fixed 50 ms aggregation period ensures fluid real-time interaction.

3.4 One-Shot Discrete Event Handling: Integrating Continuous Control with Instantaneous Triggering

In collective interactive music systems, continuous parameter control (e.g., filter sweeps, volume fades) and discrete event triggering (e.g., MIDI notes, percussion hits) are fundamentally different, and the two are rarely integrated. The former is well suited to weighted-average aggregation—multiple users' swipe intensities can be fused into an overall trend. The latter, however, cannot be averaged: the mean of pitch 60 (C4) and pitch 64 (E4) is 62 (D4), which is musically meaningless and can lead to chaotic performance output. To address this, the system implements a dedicated discrete-event processing channel, detailed as follows.

3.4.1 On-Screen Virtual Keyboard Mapping

The system maps the entire touch screen of each mobile device to a virtual keyboard spanning five octaves (Figure 5). Touch positions are converted from 2D screen coordinates to MIDI notes:

• *X-axis (horizontal)*: determines the semitone position within an octave. The axis is divided into 12 regions corresponding to C, C#, D, D#, E, F, F#, G, G#, A, A#, and B.

• *Y-axis (vertical)*: determines the octave register. The bottom of the screen corresponds to the low register (C2), and the top corresponds to the high register (B6).

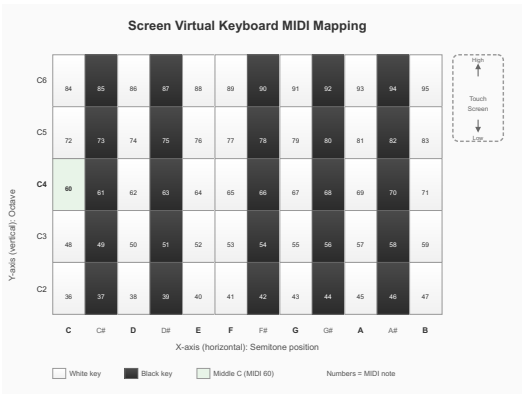


Figure 5. Screen virtual keyboard MIDI mapping.

3.4.2 One-Shot Trigger Handling (Client Timing and Server Immediate Response)

Unlike continuous parameters that are transmitted every frame, the on-screen keyboard emits a note only once at touch start; touch movement or release does not re-trigger, avoiding unintended note streams during sliding. To support this one-shot behavior end-to-end, the client marks discrete-event messages with the flag `oneShot: true`. When the server detects this flag, it applies three rules: (i) fixed weighting (1.0) independent of interaction intensity so that even light taps trigger notes; (ii) no weighted averaging—use the latest-arriving trigger value to avoid pitch ambiguity under simultaneous triggers; and (iii) immediate clearing—after sending the OSC message, remove the corresponding address from the audience data state to prevent the same note from being re-triggered in the next aggregation cycle (50 ms later). This mechanism cleanly separates instantaneous triggering from continuous control within the same aggregation-based architecture.

This design allows the same system architecture to seamlessly support two fundamentally different types of musical control. Audience members can first tap the screen to trigger discrete MIDI notes and thus trigger sound, and then use sustained sliding gestures to control continuous timbral parameters (e.g., two-finger distance controlling reverb dry/wet ratio), enabling rich and dynamic sonic expression. This addresses the technical challenge that collective interactive systems typically struggle to support individual instantaneous control, expanding the dimensions of musical expression available to audience participation.

3.5 Zero-Friction Multi-Scenario Deployment Architecture: Seamless Transition from Local Performances to Global Interconnectivity

Networked music performance systems face a practical challenge: network conditions vary substantially across rehearsals, local performances, and international events. A rehearsal room may have no public-Internet access; small venues may depend on local WiFi; and international conferences (e.g., ICMC) involve participants distributed worldwide, requiring public-Internet access that may need to traverse firewalls. Conventional systems often require scenario-specific reconfiguration or even code changes, which imposes a significant technical burden on artists. To address this, the system provides a unified deployment framework that enables seamless switching among six combined deployment modes ranging from local single-machine deployment to public-Internet tunneling via an interactive startup script, achieving zero-friction adaptation across scenarios.

In single-machine mode, MediaPipe motion capture, the audience touch subsystem, and Max/MSP run on a single computer. In dual-machine mode, motion capture and audio synthesis run on the performer machine, while the audience monitoring system runs on a separate visual machine. The two machines are connected to the same LAN and use LAN UDP broadcast for cross-machine OSC communication. This separation prevents high-load

tasks from competing for CPU/GPU resources and reduces the performance requirements of any single machine.

The startup script configures the system via environment variables (PERFORMER_ID, OSC_PORT, USE_BROADCAST, etc.), allowing the same codebase to run under different topologies. Crucially, a public Cloudflare Tunnel enables global access, eliminating the need for server-client co-location. In public-Internet modes, a script automatically synchronizes the tunnel URL to generate the necessary audience QR code.

4. SYSTEM EVALUATION

This chapter provides a quantitative assessment of the system's performance boundaries based on an automated stress-testing script and preliminary user testing. Note that the system has not yet been validated in large-scale public performances; the following analysis is based on stress-test data collected under laboratory conditions.

4.1 Concurrent Capacity Stress Test

To quantify practical system capacity, we developed an automated stress-testing script to simulate concurrent audience connections and data transmission at different scales. Tests were conducted in the local single-machine deployment mode. The script emulated real client behavior by establishing WebSocket connections, sending registration messages, and continuously transmitting touch-gesture data and OSC-mapping messages using the same message formats as the live system.

4.1.1 Test Configuration

Test Hardware: Intel-based MacBook Pro.

Simulated clients were configured as 20% high-intensity (30 ms), 50% medium-intensity (50 ms), and 30% low-intensity (100 ms) send intervals. Concurrency was increased in stages from 10 to 200 participants (10→30→50→80→100→150→200).

4.1.2 Local Mode Test Results (Table 3)

Deployment Mode: local single-machine mode.

Users	Success (%)	Throughput (msg/s)	Status
10	100%	349	Stable
30	100%	1,218	Stable
50	100%	1,823	Stable
80	100%	3,625	Stable
100	100%	3,645	Stable
150	100%	5,887	Stable
200	100%	7,540	Stable (staggered)

Table 3. Local Mode Test Results

Analysis: In local mode, the system remained stable with 200 concurrent audience members and a peak throughput of 7,540 msg/s. The activity-based sampling strategy at the 80-user threshold operated as intended, maintaining stability under high concurrency. The 200-user stage used staggered connections (batches of 50), matching typical performance scenarios where audiences join progressively rather than simultaneously.

4.1.3 Public-Internet Mode Test Results (Table 4)

Deployment Mode: Cloudflare tunnel public-Internet mode.

Users	Success （%）	Throughput (msg/s)	Status
10	100%	411	Stable
30	100%	1,271	Stable
50	100%	1,745	Stable
80	100%	3,009	Stable
100	70%	2,626	Partial failures
150	87%	5,093	Unstable

Table 4. Public-Internet Mode Test Results

Analysis: In public-Internet mode, the system remained stable with 80 concurrent audience members and a throughput of 3,009 msg/s. Above 100 users, partial connection failures emerged due to tunnel-service connection limits and network latency. The stable concurrency ceiling is approximately 53% of local mode, consistent with the additional forwarding and TLS overhead of tunnel-based Internet communication.

4.2 Latency Analysis

System end-to-end latency was estimated from code analysis and empirical measurement. Touch acquisition contributes <1 ms and client-side processing adds 1–3 ms in both local and public-Internet modes. Network transmission is <5 ms over WiFi (local) versus 50-200 ms over a public-Internet tunnel. Server-side aggregation is fixed at 50 ms, and OSC transmission over UDP local loopback adds <1 ms. Total latency is therefore ~60 ms in local mode and 100‑260 ms in public-Internet mode.

While a ~60 ms local latency is high for tight rhythmic synchronization, it is well-tolerated for gradual timbral modulation, ensuring users maintain a clear sense of agency. Although Public-Internet latency is higher, the system remains effective for collective participation where absolute rhythmic precision is less critical. Future work will focus on minimizing latency to expand support for more rhythmically demanding interactions.

4.3 Small-Scale User Test

To verify basic system usability, we conducted a small-scale test (Figure 6) with 5–10 volunteers under a local WiFi network and Cloudflare tunnel.(single-machine deployment mode)

Observations:

• All participants successfully joined by scanning a QR code, validating the feasibility of the zero-installation design.

• Perceived latency between touch interaction and audio feedback was acceptable; no noticeable lag was reported.

• The monitoring panel correctly displayed a particle for each participant, with particle motion synchronized to mobile gestures.

• The system ran stably throughout the test with no crashes or connection drops.

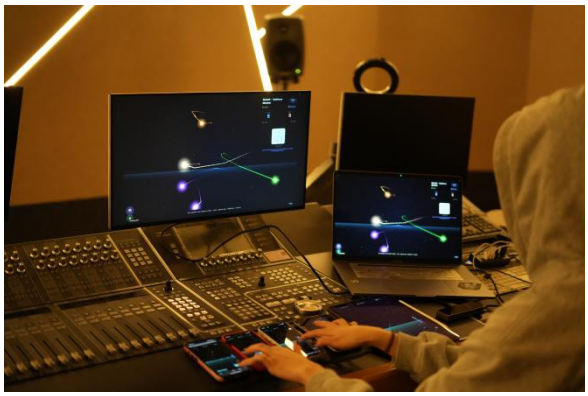


Figure 6. Small-scale demonstration.

4.4 Limitations and Future Work

Limitations: Results are based on simulated clients and small-scale tests; the system has not yet been validated in large-scale public performances, where network complexity and device heterogeneity may affect outcomes. In single-machine mode, stable concurrency is capped at 200 users locally and 80 over the public Internet; larger deployments require dual-machine operation. Local mode depends on venue WiFi quality, while public-Internet mode is constrained by tunnel-service limits and network fluctuations.

Future work: (1) large-scale in-situ evaluation with performance and user-experience data; (2) server-cluster architectures for higher concurrency; (3) improved public-Internet stability via WebRTC or other P2P approaches to reduce reliance on centralized tunnels; (4) expanded interaction modalities (e.g., accelerometer and gyroscope input).

5. CONCLUSION

This paper presents Gestalt, a real-time co-creative audiovisual performance system that addresses the tension between performer-level structural control and large-scale audience agency in Networked Music Performance. Gestalt adopts a bilayer heterogeneous architecture grounded in mobile edge computing: computation is offloaded to audience devices and audiovisual data streams are decoupled. In local-network mode, laboratory stress tests indicate stable operation with 200 concurrent participants and an estimated end-to-end latency of approximately 60 ms, demonstrating the practical potential of web technologies in professional music performance.

Beyond engineering considerations, Gestalt's core contribution is its "Translation" mechanism: an activity-weighted aggregation algorithm transforms tens to hundreds of discrete, asynchronous micro-gestures into coherent musical textures, while physics-based particle visualization renders per-user audience data as visible, dynamic particles. This design reframes the audience from passive recipients to co-creators with aesthetic agency.

Current validation is limited to laboratory stress tests and small-scale workshops; large-scale public-performance evaluation remains future work. Gestalt is released as open source (Link to https://github.com/Purest-11/Gestalt) to support academic exchange and reuse.

6. REFERENCES

[1] A. Stolfi, M. Barthet, F. Goródscy, A. Deusany, and F. Iazzetta, "Open Band: Audience Creative Participation Using Web Audio Synthesis," *Proceedings of the 12th International Audio Mostly Conference*, London, U.K., 2017, pp. 1‑8.

[2] S. Gresham-Lancaster, "The Aesthetics and History of the Hub: The Effects of Changing Technology on Network Computer Music," *Leonardo Music J.*, vol. 8, pp. 39‑44, 1998.

[3] G. Levin, "Dialtones (A Telesymphony)," Performance, Ars Electronica Festival, Linz, Austria, 2001.

[4] S. W. Lee and J. Freeman, "Echobo: A Mobile Music Instrument Designed for Audience to Play," *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*, Daejeon, Korea, 2013, pp. 450‑455.

[5] O. Hödl, C. Bartmann, F. Kayali, C. Löw, and P. Purgathofer, "Large-Scale Audience Participation in Live Music Using Smartphones," *J. New Music Research*, vol. 49, no. 2, pp. 192‑207, 2020.

[6] S. Robaszkiewicz and N. Schnell, "Soundworks−A Playground for Artists and Developers to Create Collaborative Mobile Web Performances," *Proceedings of the 1st Web Audio Conference (WAC)*, Paris, France, 2015.

OrbitScore: A Domain-Specific Language for Polymetric Live Coding Based on Multilayered Temporal Structures

Hiroshi Yamato
Keio University / Signal compose Inc.
yamato@signalcompose.com

ABSTRACT

This paper presents OrbitScore, a domain-specific language (DSL) for live coding polymetric rhythm patterns based on the theory of Multilayered Temporal Structures (MLTS). While existing live coding languages such as TidalCycles and Sonic Pi provide rich pattern manipulation capabilities including polyrhythmic support, OrbitScore offers an intuitive syntax where the `beat(n by m)` *notation directly maps to the theoretical 4:(n/4) framework, enabling each sequence to maintain its own meter and allowing performers to create intricate polyrhythmic textures in real-time. The system integrates with SuperCollider for low-latency audio synthesis and provides a declarative, method-chaining syntax designed for live performance. We describe the theoretical foundation, DSL design, implementation architecture, and demonstrate the system's capabilities through a live coding performance. Our contribution lies in bridging the gap between the theoretical framework of Multilayered Temporal Structures and practical live coding tools, making polymetric expressions accessible to performers.*

1. INTRODUCTION

The diversification of polyrhythms in popular music since the 2000s has created new challenges for music production tools. While Digital Audio Workstations (DAWs) and MIDI-based systems assume a single, unified tempo, contemporary music increasingly incorporates complex polymetric patterns where multiple rhythmic layers operate at different tempos simultaneously.

Existing live coding languages such as TidalCycles [1] and Sonic Pi [2] have made significant contributions to algorithmic music performance, including support for polyrhythmic patterns. This paper presents OrbitScore, a domain-specific language that provides an intuitive syntax for expressing the theory of Multilayered Temporal Structures (MLTS), where the `beat(n by m)` notation directly maps to the theoretical 4:(n/4) framework.

Our contributions are threefold:

1. A DSL implementation that realizes the theoretical framework of Multilayered Temporal Structures, making complex polymetric patterns expressible in live performance.
2. Integration with SuperCollider for low-latency real-time audio synthesis.
3. A demonstration of polymetric live coding through public performance.

Rather than claiming to "revolutionize" the listening experience, we aim to make polymetric performance *possible* through accessible tools, bridging the gap between theoretical concepts and practical music-making.

Copyright: ©2026 Hiroshi Yamato. This is an open-access article distributed under the terms of the Creative Commons Attribution License 3.0 Unported, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

2. RELATED WORK

2.1 African Polyrhythm and Its Transmission

The foundation of polyrhythmic practice in contemporary music can be traced to West African musical traditions. As Pressing notes, the 12-beat cycle serves as "the prototype of polyrhythm in African music," with the fundamental cross-rhythm existing between 4-beat and 3-beat pulses [3]. This "vertical hemiola" represents the most primitive form of African polyrhythm, creating tension through the simultaneous articulation of duple and triple subdivisions.

Toussaint's geometric analysis of rhythm demonstrates that African timeline patterns exhibit optimal mathematical properties for creating rhythmic complexity [4]. These patterns, particularly the *tresillo* (3+3+2), have been extensively adopted in Latin music, rhythm and blues, jazz, rock and roll, and contemporary popular music, establishing a documented historical continuity from African rhythmic practice [5].

Indian classical music similarly employs *tala*—cyclic rhythmic patterns where multiple temporal divisions coexist, creating polymetric structures through overlapping patterns [6]. The *clave* pattern in Afro-Cuban music further demonstrates this transmission, where African standard patterns were reinterpreted within a 4/4 framework while retaining their polyrhythmic tension [7].

Critically, we distinguish between cross-rhythm and true polyrhythm. Cross-rhythm involves multiple time signatures dividing one cycle by integer ratios, while polyrhythm encompasses non-integer ratio divisions characterized by "sway or accent"—the micro-timing variations that create groove [8].

2.2 American Minimalism and Phase Techniques

Steve Reich's phase-shifting techniques, while often cited as polyrhythmic, derive primarily from canon form traditions rather than direct African influence. As Reich himself stated regarding his use of 12/8 time and repetitive patterns: "I didn't write it because I went to Ghana...I went

to Africa to confirm" [9]. His methodology emphasizes temporal "shifts" rather than the African concepts of "sway" or "accent."

This distinction is significant for our work: while Reich's techniques produce rhythmic complexity through gradual phase displacement, they operate within a unified temporal framework. In contrast, the Multilayered Temporal Structures we address involve genuinely independent tempo streams that periodically align—a fundamentally different approach to polyrhythmic composition.

2.3 Polyrhythm in Contemporary Popular Music

Since the 2000s, popular music has increasingly incorporated complex polyrhythmic elements that challenge conventional production tools. Albums such as Robert Glasper's *Black Radio* (2012) demonstrate how rhythmic complexity derived from African traditions continues to influence contemporary jazz and hip-hop. Artists working at the intersection of these genres regularly employ patterns that standard DAW quantization cannot adequately represent.

Contemporary composers such as Naruyoshi Kikuchi have explicitly employed polymetric structures, creating works where "4/4 and 5/4 coexist—a rhythm that sounds square from one perspective...but sounds 'accented' from the other." These practical applications demonstrate the need for tools that can express independent meters within a unified temporal framework.

2.4 Live Coding Languages

TidalCycles, developed by McLean, provides a Haskell-based DSL with sophisticated pattern manipulation through its mini-notation system [1]. The language's approach to pattern transformation has influenced subsequent live coding language design [10]. While TidalCycles supports polymetric expressions through pattern manipulation, expressing multi-layered temporal structures with independent synchronization points requires complex compositional strategies.

Sonic Pi, designed by Aaron et al., emphasizes temporal semantics and accessibility [2]. Its programming model focuses on time-based execution with educational applications [11]. However, the system does not directly address independent meter assignment per sequence—each sequence shares a global temporal reference.

Renney and Gaster's work on digital representation of rhythm explores DSL approaches to polyrhythmic and polymetric expression [12], identifying the gap between theoretical rhythmic concepts and practical computational tools.

2.5 Polytempo Research

Kocher's Polytempo Composer provides computational tools for synchronizing different tempo progressions [13]. This work addresses the mathematical foundations of tempo relationships, computing synchronization points where divergent tempo streams reconverge. However, the focus remains on composition and notation rather than live performance and real-time interaction.

2.6 Research Gap

While these systems provide powerful tools for rhythmic expression, none directly addresses the complete set of requirements for expressing Multilayered Temporal Structures in live performance:

- Independent time signatures per sequence (not merely pattern manipulation)
- Automatic synchronization at calculated measure boundaries
- Real-time modification during performance
- Low-latency audio output suitable for live contexts

OrbitScore addresses this gap by providing a DSL specifically designed for polymetric live coding with integrated audio synthesis.

3. BACKGROUND: MULTILAYERED TEMPORAL STRUCTURES

3.1 Motivation and Definition

Contemporary popular music production relies heavily on conventional time signatures (4/4, 3/4) inherited from classical notation systems. Despite innovations in modern genres attempting to depart from standard rhythmic forms, these remain constrained by the assumption of a unified temporal grid. The historical collision of African and European musical traditions generated creative tensions that produced jazz and blues, demonstrating how rhythmic diversity drives musical innovation.

We propose **Music with Multilayered Temporal Structures** as a framework characterized by two essential elements:

1. Multiple melodic, rhythmic, or instrumental lines operating at **different tempos** simultaneously.
2. These lines **periodically aligning** on the first beat of calculated measures.

This distinguishes our approach from random multi-BPM layering: the periodic convergence creates intentional structural relationships between tempo streams. Each layer maintains its own rhythmic identity while participating in a larger polymetric architecture.

3.2 The n-beat in 4/4 Notation System

Rather than using traditional tuplet notation, we employ a notation system that explicitly represents the relationship between layers. The notation $4 : (n/4)$ describes a four-beat measure where the beat unit is derived from n beats mapped onto a 4-beat reference:

- $4 : (4/4)$: Standard four beats per measure (reference tempo)
- $4 : (5/4)$: Five beats compressed into the duration of four reference beats
- $4 : (3/4)$: Three beats expanded to span four reference beats

- $4 : (7/4)$: Seven beats compressed into four reference beats

When $4 : (4/4)$ and $4 : (5/4)$ run simultaneously, they share a common reference but progress at different rates. The $4 : (5/4)$ layer completes its cycle in 80% of the time required by $4 : (4/4)$, creating a rhythmic tension that resolves at synchronization points.

3.3 Mathematical Foundation

3.3.1 Synchronization Point Calculation

The beat where all layers realign can be calculated using the Least Common Multiple (LCM) of the n-beat values. For layers with beat counts $n_1, n_2, \ldots, n_k$:

$$\text{sync\_beat} = \text{LCM}(n_1, n_2, \ldots, n_k) + 1 \tag{1}$$

The number of measures (bars) until synchronization for layer i:

$$bars_i = \frac{\text{LCM}(n_1, n_2, \ldots, n_k)}{n_i} \tag{2}$$

The length of each beat in layer i, relative to the LCM period:

$$beat\_length_i = \frac{\text{LCM}}{n_i \times 4} \tag{3}$$

3.3.2 BPM Conversion Across Layers

When a base tempo (BaseBPM) is established for the reference layer, the effective tempo for layer i with n_i beats is:

$$BPM_i = \frac{BaseBPM}{n_i/4} = \frac{4 \times BaseBPM}{n_i} \tag{4}$$

For example, at BaseBPM = 120:

- $4 : (4/4)$: $\frac{4 \times 120}{4} = 120$ BPM
- $4 : (5/4)$: $\frac{4 \times 120}{5} = 96$ BPM
- $4 : (3/4)$: $\frac{4 \times 120}{3} = 160$ BPM

Figure 1 illustrates how these different beat counts map onto the same reference duration.

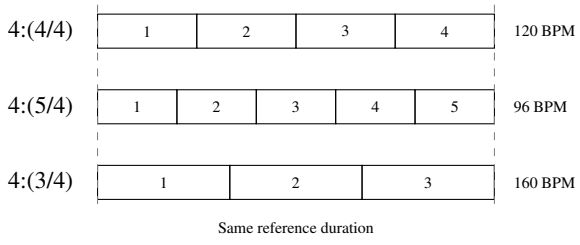


Figure 1. Comparison of n-beat notation showing how different beat counts map onto the same reference duration. At BaseBPM = 120, the effective tempos are 120, 96, and 160 BPM respectively.

3.4 Worked Examples

3.4.1 Two-Layer Example: 4-beat and 5-beat

Consider a composition combining $4 : (4/4)$ and $4 : (5/4)$:

- LCM(4, 5) = 20
- The 4-beat layer plays 5 measures (20 ÷ 4 = 5)
- The 5-beat layer plays 4 measures (20 ÷ 4 = 4)
- Synchronization occurs on beat 21

This creates a 20-beat cycle where the two layers gradually diverge and reconverge, generating "polyrhythmic fluctuation" through their interaction. Figure 2 illustrates this synchronization pattern.

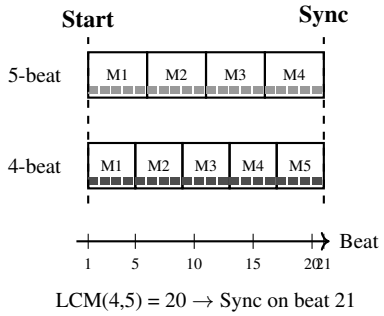


Figure 2. LCM-based synchronization of 4-beat and 5-beat layers. The 4-beat layer completes 5 measures while the 5-beat layer completes 4 measures, both converging on beat 21.

3.4.2 Three-Layer Example: 3-beat, 4-beat, and 5-beat

A more complex composition combining $4 : (3/4)$, $4 : (4/4)$, and $4 : (5/4)$:

- LCM(3, 4, 5) = 60
- The 3-beat layer plays 20 measures
- The 4-beat layer plays 15 measures
- The 5-beat layer plays 12 measures
- Synchronization occurs on beat 61

At BaseBPM = 120, the effective tempos are 160, 120, and 96 BPM respectively. The points where any two layers coincide create characteristic rhythmic motifs within the 60-beat cycle.

3.5 Variable Beat Notation vs. Fixed Beat Notation

Two notational approaches can represent Multilayered Temporal Structures:

Variable Beat Notation: Each layer maintains independent bar lines, showing the n-beat patterns as they naturally occur without forced alignment. This preserves the conceptual independence of each tempo stream.

Fixed Beat Notation: All layers are transcribed to a conventional aligned score format based on a primary meter. This facilitates reading for ensemble performance but obscures the underlying polymetric structure.

OrbitScore adopts the Variable Beat approach internally, with each sequence maintaining its own temporal reference while the scheduler manages synchronization automatically.

4. ORBITSCORE DESIGN

4.1 Design Principles

OrbitScore is designed around four core principles derived from the requirements of polymetric live coding:

1. **Independent Timing**: Each sequence maintains its own meter, implementing the Variable Beat Notation approach at the computational level.
2. **Declarative Syntax**: Method chaining enables readable, performable code that can be modified during performance without cognitive overhead.
3. **Live Coding Ergonomics**: Immediate execution with minimal ceremony—no compilation step, instant feedback.
4. **Theoretical Fidelity**: Direct mapping from the $4 : (n/4)$ notation to the `beat(n by 4)` syntax preserves the conceptual model.

4.2 Language Architecture

OrbitScore is implemented as a TypeScript-embedded DSL. The language provides three primary constructs:

- **GLOBAL**: The root timing context establishing base tempo
- **Sequence**: Individual pattern containers with independent meters
- **Transport**: Performance control commands for real-time manipulation

4.3 Basic Syntax

The language provides a global context and sequence objects:

```
var global = init GLOBAL
global.tempo(140)
global.beat(4 by 4)

var kick = init global.seq
kick.beat(4 by 4).length(1)
kick.sound("kick.wav")

var bass = init global.seq
bass.beat(5 by 4).length(1)
bass.sound("bass.wav")
```

The `bass` sequence operates at $4 : (5/4)$, creating a polymetric relationship with the `kick` at $4 : (4/4)$. Both sequences reference the same global transport, ensuring synchronization at LCM boundaries while maintaining independent beat progressions.

4.4 Polymeter Implementation

The `beat(n by m)` syntax directly implements the $4 : (n/4)$ notation:

```
seq1.beat(4 by 4)  // Standard: 4:(4/4)
seq2.beat(5 by 4)  // Polymeter:
    4:(5/4)
seq3.beat(3 by 4)  // Third layer:
    4:(3/4)
seq4.beat(7 by 4)  // Complex: 4:(7/4)
```

The scheduler manages each sequence independently at its own loop duration. Since all sequences share the same global tempo reference, their cycles naturally reconverge at the LCM of their beat values, creating the periodic alignment characteristic of Multilayered Temporal Structures.

4.5 Hierarchical Play Structure

The `play()` method supports nested tuple structures for expressing complex rhythmic patterns within each beat:

```
seq.play(1, (2, 3), 4)
// Position 1: slice 1 plays
// Position 2: (2, 3) subdivides into
    two events
// Position 3: slice 4 plays
```

This creates a hierarchical time division where nested tuples recursively subdivide their allocated time slots. The system generates `TimedEvent` objects with:

- `sliceNumber`: Position in the sequence
- `startTime`: Calculated onset time
- `duration`: Event length
- `depth`: Nesting level for dynamic processing

This approach enables tresillo patterns, tuplet figures, and complex subdivisions within the polymetric framework.

4.6 Transport Commands

Performance control is provided through intuitive transport commands:

```
RUN(kick, snare)    // One-shot playback
LOOP(bass)          // Continuous loop
MUTE(hihat)         // Mute hihat (
    preserves timing)
MUTE()              // Unmute all
global.start()      // Start scheduler
global.stop()       // Stop all
    sequences
```

The `MUTE` command preserves timing context, allowing performers to remove and reintroduce elements without disrupting synchronization.

4.7 Live Performance Workflow

A typical performance session follows this pattern:

```
// Setup phase
var global = init GLOBAL
global.tempo(120)

// Build layers incrementally
var kick = init global.seq
kick.beat(4 by 4).length(1)
kick.sound("kick.wav")
kick.play(1, 0, 1, 0)
LOOP(kick)

// Add polymetric layer
var perc = init global.seq
perc.beat(5 by 4).length(1)
perc.sound("clave.wav")
perc.play(1, 0, 1, 1, 0)
```

```
LOOP(perc)

// Modify during performance
perc.beat(7 by 4)  // Shift to 7-beat
```

The performer incrementally constructs the polymetric texture, modifying parameters in real-time as the music evolves.

5. IMPLEMENTATION

5.1 System Architecture

OrbitScore is implemented as a TypeScript-based system comprising five interconnected components:

1. **Parser**: Tokenizes and parses OrbitScore DSL syntax into an abstract syntax tree (AST), handling the `beat(n by m)` notation and nested play structures.
2. **Interpreter**: Evaluates the AST, maintaining sequence state and triggering timing calculations when parameters change.
3. **Timing Engine**: Calculates event timing for each sequence based on beat parameters, generating `TimedEvent` objects with precise onset times.
4. **Audio Engine**: Manages SuperCollider communication via OSC, handling buffer allocation, playback triggers, and transport control.
5. **VS Code Extension**: Provides IDE integration including syntax highlighting, Cmd+Enter execution, and real-time feedback.

The architecture follows a reactive pattern: parameter changes propagate through the timing engine, which recalculates schedules and updates the audio engine without interrupting playback. Figure 3 illustrates this component structure.

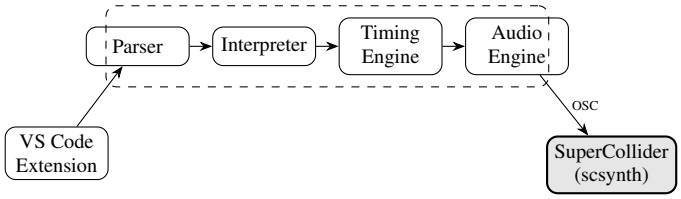


Figure 3. OrbitScore system architecture. DSL components process code and schedule events, communicating with SuperCollider via OSC.

5.2 SuperCollider Integration

Audio synthesis is delegated to SuperCollider's `scsynth` server via OSC (Open Sound Control). The implementation uses the `supercolliderjs` library for TypeScript/Node.js integration.

5.2.1 OSC Communication Layer

The system establishes a bidirectional OSC connection:

- **Outbound**: Synth triggers, buffer commands, transport messages
- **Inbound**: Server status, buffer completion notifications

Messages are scheduled with timestamps for sample-accurate playback, leveraging SuperCollider's internal scheduling precision.

5.2.2 Audio Capabilities

Key audio processing features include:

- Audio file playback (WAV, AIFF, MP3, MP4)
- Dynamic buffer management for sample loading and unloading
- Real-time parameter modulation (pitch, amplitude, pan)
- 48kHz/24-bit audio output with low latency

The latency characteristics depend primarily on SuperCollider's audio hardware configuration. With a 256-sample buffer at 48kHz, the system achieves approximately 5ms audio latency, suitable for live performance contexts.

5.3 Timing Engine Details

The timing engine implements the mathematical framework described in Section 3:

5.3.1 Synchronization Through Independent Loops

Rather than explicitly computing LCM synchronization points, OrbitScore employs a simpler approach: each sequence loops independently at its own bar duration. When sequences with beat values n_1 and n_2 are active, they naturally reconverge at the LCM of their periods. For example, with `beat(4 by 4)` (2000ms at 120 BPM) and `beat(5 by 4)` (2500ms), both sequences align every 10000ms (LCM of 2000 and 2500).

This design relies on the computational precision of the host machine's clock. Within a single process, timing drift is negligible. However, irrational or non-terminating decimal ratios may introduce cumulative drift over extended performances—a limitation that also reflects the inherent difficulty of precisely reproducing Multilayered Temporal Structures even with computational tools.

5.3.2 Event Scheduling

For each sequence, the engine generates a schedule of `TimedEvent` objects for one bar cycle. Events include:

- Absolute start time (milliseconds from loop start)
- Duration (calculated from beat subdivision)
- Slice number and depth (for hierarchical `play()` patterns)

The scheduler pre-sends OSC messages to SuperCollider to ensure timing accuracy despite network latency.

5.4 Development Process

The implementation followed test-driven development practices:

- **Unit Tests**: Event timing calculation, parse tree generation
- **Integration Tests**: OSC communication, buffer management, transport coordination
- **Manual Testing**: Audio output verification requires human evaluation, as automated testing of timing accuracy in audio playback remains challenging

Development utilized LLM assistance (Claude Sonnet 3.5 via Cursor initially, CLI-based Claude Code with Opus 4.5 currently) for code generation, debugging, and documentation. The DSL-first approach proved particularly effective: by designing a language specification readable by both humans and machines, we established a shared understanding that enabled rapid iteration. The initial performance-ready version was completed in approximately three weeks. The LLM proved especially effective for generating edge-case tests and identifying synchronization bugs in the scheduler.

6. EVALUATION

6.1 Performance Demonstration

Scratch&Build is a meetup event centered on scratch audio-visual sessions through live coding, where performers build programs from zero within limited time slots. OrbitScore was demonstrated at the first Tokyo edition of this event at TOKYO NODE LAB on October 6, 2025, which also featured the international live coding community "Fake[dac]" from London/Zurich as special guests.

6.1.1 Performance Setup

The performance used the following configuration:

- MacBook Pro (M1) running OrbitScore in VS Code
- SuperCollider audio server with 256-sample buffer (approximately 5ms latency)
- External audio interface at 48kHz/24-bit
- Screen projection showing live code to the audience

6.1.2 Demonstrated Capabilities

The 50-minute performance demonstrated the core polymetric capabilities:

- **Independent Meters**: Simultaneous sequences at $4 : (4/4)$, $4 : (5/4)$, and $4 : (7/4)$ running with automatic synchronization
- **Real-time Modification**: Beat values changed during playback, with the system recalculating synchronization points seamlessly
- **Hierarchical Patterns**: Nested `play()` structures creating tresillo and other complex subdivisions
- **Dynamic Layering**: Incremental addition and muting of layers to build and deconstruct the polymetric texture

The audience could observe the code changes and their immediate sonic results, providing transparency into the polymetric relationships being created.

A video recording of the performance is available online [1] .

6.2 Addressing Production Environment Gaps

Prior research on Multilayered Temporal Structures identified a critical gap: "inadequate production environments" for composing and performing such music. Existing DAWs could not handle multi-BPM or multilayered time signature configurations, forcing composers to use workarounds such as MIDI stretch functions to "force consistency."

OrbitScore directly addresses this gap:

- The `beat(n by m)` syntax makes polymetric relationships explicit and executable
- Each sequence maintains independent timing while sharing a global tempo reference
- Real-time modification enables iterative exploration of polymetric textures

This represents a concrete response to the prior research's call for "the development of production environments" to advance rhythmic cognition and structural understanding of multilayered music.

6.3 Development Metrics

The GitHub repository [2] provides quantitative data on the development process:

- **Commit History**: Iterative development over 6 months with regular refinements
- **Test Coverage**: Core timing engine functions covered by unit tests
- **Code Size**: Approximately 7,000 lines of TypeScript (excluding tests and configuration)

The LLM-assisted development approach enabled rapid iteration, particularly for the timing engine where edge cases required careful handling. Document-driven development, particularly DSL-centered specification, proved essential for effective LLM collaboration. By maintaining a single source of truth in the language specification, we minimized discrepancies between the LLM and the author's understanding. This shared foundation enabled coherent playtesting where both parties operated from aligned expectations. In this sense, OrbitScore represents a co-creation with an LLM agent—an achievement that would not have been possible for the author alone within such a short timeframe.

[1] Performance video: `https://youtu.be/-U44Hd5I7zk?t=3011`

[2] `https://github.com/signalcompose/orbitscore`

6.4 Comparison with Existing Languages

To illustrate OrbitScore's contribution, we compare the code required to express a 4-beat and 5-beat polymetric pattern:

OrbitScore (direct polymeter syntax):

```
kick.beat(4 by 4).play(1,0,1,0)
perc.beat(5 by 4).play(1,0,1,1,0)
```

TidalCycles (polymeter via pattern length):

```
d1 $ sound "{kick kick kick kick, perc
    perc perc perc perc}"
```

TidalCycles supports polymeter through the $\{\}$ syntax, where patterns of different lengths share the same pulse—both patterns operate at the same tempo, with the longer pattern wrapping across cycle boundaries. In contrast, OrbitScore's `beat(n by m)` syntax implements true Multilayered Temporal Structures: each layer operates at a distinct effective tempo (e.g., $4 : (5/4)$ runs at 80% of the base tempo), with independent bar structures reconverging at LCM boundaries. This reflects the MLTS theoretical framework where layers maintain genuinely independent temporal progressions rather than fitting different pattern counts into a shared timeframe.

This syntactic directness reduces cognitive load during live performance, where performers must rapidly express musical ideas. The OrbitScore pattern directly maps to the theoretical $4 : (4/4)$ and $4 : (5/4)$ notation without intermediate calculation.

6.5 Limitations

The current implementation has several limitations that scope future work:

- **Platform**: Currently macOS only due to SuperCollider integration dependencies
- **Learning Curve**: The DSL requires familiarity with both programming concepts and polymetric theory; educational materials are not yet developed
- **No "Listening"**: The system cannot respond to external audio input or other performers' timing
- **Limited Synthesis**: Audio output relies on sample playback; no built-in synthesis capabilities beyond SuperCollider's features

The "listening" limitation is particularly significant for ensemble performance. Human musicians naturally adjust their timing based on what they hear from others—micro-timing variations that create groove. OrbitScore currently operates as a strict scheduler, unable to adapt its timing to external inputs. This precludes genuine ensemble performance with human musicians who expect temporal dialogue.

7. DISCUSSION AND CONCLUSION

7.1 Summary of Contributions

This paper presented OrbitScore, a domain-specific language that makes polymetric live coding possible by implementing the theory of Multilayered Temporal Structures. Our contributions address the research gap identified in Section 2:

1. **Theoretical Implementation**: We provide the first practical implementation that directly maps the $4 : (n/4)$ notation to executable code, bridging the gap between theoretical concepts and live performance practice.
2. **Independent Meter Management**: The `beat(n by m)` syntax enables each sequence to maintain its own time signature, with independent loops naturally reconverging at LCM boundaries.
3. **Audio Integration**: SuperCollider integration provides low-latency audio synthesis suitable for live performance contexts.
4. **Practical Validation**: Public performance demonstrated that the system enables real-time creation and modification of polymetric textures.

7.2 Relationship to Prior Work

OrbitScore differs from existing live coding languages in its explicit focus on Multilayered Temporal Structures. While TidalCycles provides powerful pattern manipulation through transformation operators and polymeter support via its $\{\}$ syntax [1], OrbitScore's `beat(n by m)` notation directly expresses the MLTS theoretical framework. Sonic Pi's temporal semantics [2] operate within a unified time framework. OrbitScore's contribution is not to replace these systems but to provide specialized support for a specific class of rhythmic structures that are increasingly relevant in contemporary music.

The connection to African polyrhythmic practice, traced through the *tresillo* and its descendants [3, 4], grounds OrbitScore in a rich tradition of rhythmic complexity. By making these patterns expressible in live coding, we extend this tradition into computational performance practice.

7.3 On AI Collaboration

The development of OrbitScore utilized LLM assistance for code generation and debugging. LLMs can parse and generate OrbitScore patterns, suggesting possibilities for AI-assisted composition. However, a fundamental challenge remains: **listening**—the ability to adapt timing based on what other performers play. This capability, essential for ensemble performance, remains an open research problem for AI-human musical collaboration.

7.4 Future Directions

Several directions for future work emerge from this research:

Responsive Timing: Developing mechanisms for the system to adapt its timing based on external input, enabling ensemble performance with human musicians. This might involve audio analysis to detect tempo fluctuations and adjust scheduler timing accordingly.

Cross-platform Support: Extending beyond macOS to Linux and Windows to enable broader adoption.

Educational Materials: Creating tutorials and documentation that explain both the DSL syntax and the underlying Multilayered Temporal Structures theory.

Network Performance: Enabling distributed polymetric performance over networks, where multiple performers contribute sequences from different locations.

External MIDI Integration: Supporting MIDI output via virtual MIDI buses (e.g., IAC Driver on macOS) would enable OrbitScore to control external synthesizers and DAWs, expanding sound design possibilities beyond sample playback.

7.5 Conclusion

OrbitScore demonstrates that complex polymetric patterns, grounded in the theory of Multilayered Temporal Structures, can be made accessible through domain-specific language design. By providing direct mapping from theoretical notation to executable code, we lower the barrier to exploring rhythmic complexity in live performance.

The system addresses a genuine need in contemporary music production, where polyrhythmic elements derived from African traditions increasingly appear in popular genres [3, 4]. While current tools assume unified temporal grids, OrbitScore embraces independent meter relationships as a first-class concept.

Looking forward, the challenge of "listening"—enabling computational systems to participate in the temporal dialogue that defines ensemble music-making—remains the frontier for AI collaboration in music. OrbitScore provides a foundation for exploring this space, making explicit the structural relationships that must eventually become responsive.

Acknowledgments

This paper was written with assistance from Claude (Anthropic) for drafting and revision. All content was reviewed and verified by the author.

8. REFERENCES

[1] A. McLean, "Algorithmic Pattern," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2020, pp. 265–270.

[2] S. Aaron, D. Orchard, and A. F. Blackwell, "Temporal Semantics for a Live Coding Language," in *Proceedings of the 2nd ACM SIGPLAN International Workshop on Functional Art, Music, Modeling & Design*, 2014, pp. 37–47.

[3] J. Pressing, "Black Atlantic rhythm: Its computational and transcultural foundations," *Music Perception*, vol. 19, no. 3, pp. 285–310, 2002.

[4] G. T. Toussaint, *The Geometry of Musical Rhythm: What Makes a "Good" Rhythm Good?* CRC Press, 2013.

[5] K. Agawu, *Representing African Music: Postcolonial Notes, Queries, Positions*. Routledge, 2003.

[6] C. Sachs, *Rhythm and Tempo: A Study in Music History*. New York: W.W. Norton & Company, 1953.

[7] K. Tsukada, *Afrika ongaku no shōtai [The True Nature of African Music]*. Tokyo: Ongaku no Tomo Sha, 2016, in Japanese.

[8] J. London, "Three things linguists need to know about rhythm and time in music," *Empirical Musicology Review*, vol. 7, no. 1–2, pp. 5–11, 2012.

[9] D. Nicholls, *The Cambridge History of American Music*. Cambridge University Press, 1998.

[10] A. McLean and G. Wiggins, "Texture: Visual notation for live coding of pattern," in *Proceedings of the International Computer Music Conference*, 2011, pp. 621–628.

[11] A. F. Blackwell and S. Aaron, "Craft practices of live coding language design," in *Proceedings of the International Conference on Live Coding*, 2015.

[12] S. Renney and B. R. Gaster, "Digital Expression and Representation of Rhythm," *Proceedings of the International Conference on Live Coding*, 2019.

[13] P. Kocher, "Polytempo Composer: A Tool for the Computation of Synchronisable Tempo Progressions," 2016.

CNN-BILSTM HYBRID MODEL WITH PHYSICAL CONSTRAINTS FOR AUTOMATIC PIANO FINGERING GENERATION

Tianze Zhang
Guangdong Country Garden School
zhangtianze@gcgsedu.com

Shingyui He
Guangdong Country Garden School
heshingyui@gcgsedu.com

Lei Xuan
Guangdong Country Garden School
xuanlei@bgyschools.cn

ABSTRACT

Piano fingering is a pivotal technique that piano learners must master. To address the difficulties in the application and arrangement of fingering during practice and early stages of learning, this paper proposes an artificial intelligence hybrid model based on Bidirectional Long Short-Term Memory (BiLSTM) network, Convolutional Neural Network(CNN) and Attention, with the aim of automatically generating piano fingerings. The model extracts physical features, including spatial, temporal, hand motion, and fingering information, and integrates biomechanics constraints during neural network training for the first time. Based on the aforementioned algorithms, the model achieved a good result. This innovative methodology enhances predictive performance by accurately capturing the complex physical interactions inherent in piano fingering. This paper also compares the model with fingerings generated by other algorithms to verify the reasonableness and effectiveness of the hybrid model in piano fingering prediction. In conclusion, the model can efficiently and conveniently provide fingering support for piano learners, and has strong application prospects and practical value.

1. INTRODUCTION

Piano fingering, the strategic assignment of fingers to musical notes, is a critical yet understudied aspect of musical performance [1]. While composers occasionally annotate fingerings to guide interpretation, the majority of sheet music lacks such instructions, leaving performers to devise their own solutions. This process demands expertise, physical adaptability, and iterative rehearsal, as pianists must balance ergonomic efficiency, playability, and artistic expression [2]. The combinatorial complexity of fingering choices—growing exponentially with note sequence length—renders manual planning time-consuming and error-prone, particularly for novices. Although any finger can theoretically strike any key, practical constraints such as hand span, note transitions, and biomechanical limits drastically narrow viable options. Automating this decision-making process thus holds significant potential to reduce cognitive load, accelerate learning, and enhance performance fluidity.

Motivated by the challenges of manual fingering design, this paper explores data-driven solutions through iterative experimentation. The research began with an analysis of two datasets: the Piano Fingering (PIG) dataset [3], which provides comprehensive annotations, and the Thumbset dataset [4], whose partial annotations introduced noise and inefficacy. Initial attempts using raw data with convolutional neural networks (CNNs) and bidirectional long short-term memory networks (BiLSTMs)[5, 6] yielded suboptimal results, revealing the need for structured feature engineering. We subsequently developed three data-processing pipelines: (1) *raw data*, (2) *base features* (pitch, duration, hand position, etc.) augmented by Word2Vec embeddings[7], and (3) *base features* integrated with *physical constraints* (spatial-temporal relationships, hand kinematics, and historical fingering patterns). These pipelines were tested across five models: CNN[8], BiLSTM[9], a hybrid BiLSTM+CNN[8], Ar-GNN (a graph-based approach)[4], and an Ar-BiLSTM[4]. Through rigorous comparison, the hybrid BiLSTM+CNN model paired with the third data pipeline—emphasizing physical constraints—emerged as the most effective, achieving superior accuracy and playability by explicitly encoding biomechanical principles.

Our key contributions include:

- **Physical Constraint Integration**: Systematic encoding of biomechanical principles (spatial reach, finger kinematics, temporal coordination) into both architectural design and data preprocessing, a dimension neglected in prior work.
- **Hybrid CNN-BiLSTM Architecture**: A novel fusion of CNN-based local spatial pattern extraction and BiLSTM-driven sequential dependency modeling, enabling joint learning of fingering dynamics.
- **Two-Stage Planning with Attention**: A prediction framework combining initial neural predictions (augmented by attention mechanisms) with forward planning to optimize global fingering coherence, addressing the myopia of purely statistical approaches.

2. RELATED WORKS

Automatic Piano Fingering (APF), a task aimed at assigning finger numbers to notes in piano music, has become a field of study within the music information retrieval (MIR). Existing approaches to APF generally fall into two categories: rule-based methods and data-driven methods.

Copyright: ©2026 Tianze Zhang et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License 3.0 Unported, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

2.1 Rule-based Methods

Initially, APF relied heavily on rule-based techniques that use manually designed rules and heuristics. Parncutt et al.[10, 11] first modeled piano fingering as a dynamic programming problem, incorporating manually defined transition rules to minimize fingering complexity. Later developments [12, 13], such as Balliauw et al.'s work[14], extended these approaches into combinatorial optimization frameworks, utilizing search algorithms like variable neighborhood search to optimize fingering decisions according to ergonomic principles and musical context. Despite their interpretability, rule-based methods require substantial manual configuration and fail to adapt effectively to diverse musical styles and individual player preferences.

2.2 Data-driven Methods

To address limitations of rule-based approaches, recent research has shifted toward data-driven methods, utilizing annotated datasets to capture implicit fingering patterns. Nakamura et al.[3] introduced the widely adopted Piano Fingering (PIG) dataset, a benchmark that provides detailed annotations by professional pianists across a variety of Western classical compositions. Utilizing this dataset, Nakamura proposed a hidden Markov model (HMM) to model fingering transitions, achieving a robust baseline performance superior to earlier rule-based methods. However, simpler neural network architectures explored by Nakamura performed worse, indicating the need for more advanced architectures and richer data representations.

Subsequent studies explored various neural network architectures to address this gap. Ramoneda et al.[4] significantly extended this research by introducing the ThumbSet dataset, a larger, albeit noisier, collection of partially annotated piano scores aimed at improving deep neural network generalization. They proposed autoregressive neural network models including Long Short-Term Memory networks (LSTM) and graph neural networks (GNN), outperforming prior HMM and LSTM baselines. Their approach employed autoregressive decoding and beam search methods, enhancing sequential coherence of generated fingering patterns and reducing unrealistic fingering transitions.

Similarly, Li et al.[15] introduced a feature fusion methodology leveraging both pitch and tempo information, recognizing that existing methods neglect important factors influencing fingering decisions. They utilized Word2Vec continuous bag-of-words (CBOW) models to fuse musical context into feature vectors, enhancing the model's capacity to capture comprehensive musical attributes. This study further employed bidirectional LSTM combined with conditional random fields (BiLSTM-CRF), achieving marked improvements over conventional HMMs and basic LSTM models in terms of fingering accuracy and playability. Furthermore, Zhao et al. [16] proposed a pitch-difference sequence and fingering matching model by augmented the dataset with transfer rules of fingerings. However, some constraints like the cooperation of hands are not considered in the model design, potentially leading to insufficient fingering transfer knowledge.

Most recently, Gao et al.[17] and Iman et al. [18] approached APF through a model-based reinforcement learning framework, explicitly modeling the piano fingering annotation task as a combinatorial optimization problem constrained by human anatomical and biomechanical considerations. Their method leveraged reinforcement learning with a sophisticated environment model and a well-defined reward function based on minimal hand motion. This approach effectively eliminated physically impossible fingerings and substantially reduced ergonomic inefficiencies. Furthermore, their model introduced an innovative music segmentation technique, enabling parallel processing and thus improving computational efficiency and scalability. However, it's difficult to determine the reward function for the reinforcement learning and requires multiple adjustments. Furthermore, the algorithm considers multiple objectives for the fingering optimization, but it lacks of clear priorities, leading to conflict in some rules.

3. AUTOMATIC PIANO FINGERING SYSTEM

Our research builds upon and addresses these limitations by systematically integrating comprehensive physical constraints, novel hybrid deep learning architectures (CNN-BiLSTM), and advanced two-stage planning mechanisms. We emphasize the explicit encoding of biomechanical principles into data preprocessing and model architectures, leveraging physical constraints to guide model predictions. By employing iterative experimentation and rigorous evaluation methodologies, our approach seeks to bridge the gap between biomechanical realism and data-driven learning, achieving state-of-the-art accuracy, enhanced playability, and improved generalizability.

3.1 Data Representation

Piano fingering annotations may come in two different formats, depending on the annotation process. The first format is a set of finger numbers manually annotated on sheet music to suggest a fingering (or fingerings) for a composition. The second format is a list of finger numbers employed by a musician during a performance, which are directly derived from the performance. The latter form of fingering data is found in the PIG and ThumbSet datasets.

Given a sequence of T notes in a piece, the fingering is encoded by assigning a finger label y_t to the t-th note, where values from 1 to 5 denote the right hand, values from -1 to -5 denote the left hand, and a value of 0 indicates the absence of a finger annotation. Additionally, the pitch of each note is represented by x_t, with t serving as the note index. This digital representation follows the methodology detailed in [4].

3.2 Basic Score Features Extraction

The fundamental features for our automatic fingering model are extracted from piano performance recordings in text format. Each note entry N_i contains the following attributes: $(t_{on}, t_{off}, p, v_{on}, v_{off}, c, f)$, where t_{on} represents onset time, t_{off} represents offset time, p denotes spelled pitch, v_{on} and v_{off} are onset and offset velocities respectively, c indicates the channel (0 for right hand, 1 for left hand), and f is the finger number.

From these raw attributes, we derive several essential musical features:

1. **Normalized Spelled Pitch**: To standardize enharmonic spellings (e.g., C♯ and D♭), we implement enharmonic normalization:

$$p_{norm} = \mathcal{N}(p)$$

where $\mathcal{N}$ is our normalization function that maps each pitch to a preferred spelling variant.

2. **Duration**: Calculated as:

$$d_i = t_{off} - t_{on}$$

3. **Black Key Identification**: A binary feature indicating whether a note is played on a black key:

$$b_i = \begin{cases} 1, & \text{if } m_i \mod 12 \in \{1, 3, 6, 8, 10\} \\ 0, & \text{otherwise} \end{cases}$$

4. **Chord Detection**: Notes are marked as part of a chord when their time intervals overlap within the same hand:

$$c_i = \begin{cases} 1, & \text{if } \exists j \neq i : h_i = h_j \wedge t_{on,j} < t_{off,i} \\ 0, & \text{otherwise} \end{cases}$$

where h_i represents the hand assignment of note i.

These features are numerically encoded using Labe Encoders to convert categorical variables into numerical indices suitable for neural network processing. The encoding process maps each unique value of a feature to an integer:

$$p_{enc} = \mathcal{E}_p(p_{norm}), d_{enc} = \mathcal{E}_d(d), h_{enc} = \mathcal{E}_h(h)$$

where $\mathcal{E}_p$, $\mathcal{E}_d$, and $\mathcal{E}_h$ are the encoder functions for pitch, duration, and hand respectively.

3.3 Quantification of Biomechanical Fingering Constraints

The biomechanical constraints of piano fingering are quantified through a set of physical features that model the anatomical limitations of the human hand. These constraints are critical for generating ergonomically viable fingering solutions.

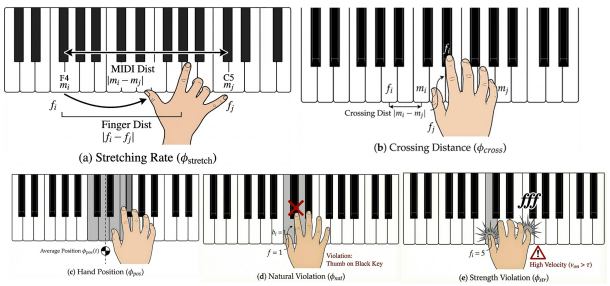


Figure 1. Physical Features

The physical constraint features are categorized into several distinct groups:

1. **Stretching Rate** ($\phi_{stretch}$): Quantifies the extension required between consecutive fingers:

$$\phi_{stretch}(i, j) = \frac{|m_i - m_j|}{|f_i - f_j|}$$

where m_i and m_j are MIDI numbers, and f_i and f_j are finger numbers.

2. **Crossing Distance** (ϕ_{cross}): Measures the physical difficulty of finger crossings:

$$\phi_{cross}(i, j) = \begin{cases} |m_i - m_j|, \\ \text{if } (f_i < f_j \wedge m_i > m_j) \vee (f_i > f_j \wedge m_i < m_j) \\ 0, \text{otherwise} \end{cases}$$

3. **Hand Position** (ϕ_{pos}): Tracks the current position and movement of the hand:

$$\phi_{pos}(t) = \frac{1}{|N_t|} \sum_{i \in N_t} m_i$$

where N_t is the set of notes currently being played at time t.

4. **Natural Violation** (ϕ_{nat}): Identifies fingerings that contradict natural hand positioning:

$$\phi_{nat}(i) = \begin{cases} 1, & \text{if } (f_i = 1 \wedge b_i = 1) \vee \text{other violations} \\ 0, & \text{otherwise} \end{cases}$$

5. **Strength Violation** (ϕ_{str}): Penalizes assignments requiring excessive force from weaker fingers:

$$\phi_{str}(i) = \begin{cases} v_{on,i} \cdot \omega_f, & \text{if } f_i \in \{4, 5\} \wedge v_{on,i} > \tau \\ 0, & \text{otherwise} \end{cases}$$

where ω_f is a finger-specific weight and τ is a velocity threshold.

These biomechanical constraints are incorporated into a specialized loss function during model training:

$$\mathcal{L}_{phys}(y, \hat{y}, \Phi) = \mathcal{L}_{CE}(y, \hat{y}) \cdot E[\omega(\Phi)]$$

where $\mathcal{L}_{CE}$ is the cross-entropy loss, Φ is the set of physical features, and $\omega(\Phi)$ is a weighting function that increases penalties for physically implausible fingerings.

3.4 Bi-LSTM-CNN Hybrid Network

Our model employs a hybrid architecture combining Convolutional Neural Networks (CNN) and Bidirectional Long Short-term Memory (BiLSTM) networks to capture both local pitch patterns and sequential dependencies in fingering decisions. The Physical-enhanced Fingering Model can be formalized as:

$$\hat{y} = \mathcal{F}(X, \Phi; \theta)$$

where X represents input features, Φ denotes physical constraint features, and θ encompasses all model parameters.

The architecture consists of several key components:

1. **CNN Feature Extraction**: The convolutional block extracts local pitch patterns and musical structures:

$$H_{conv} = \sigma(BN(Conv(X_{reshaped})))$$

where σ denotes the activation functionx (LeakyReLU), BN is batch normalization, and $Conv$ represents the convolutional operation.

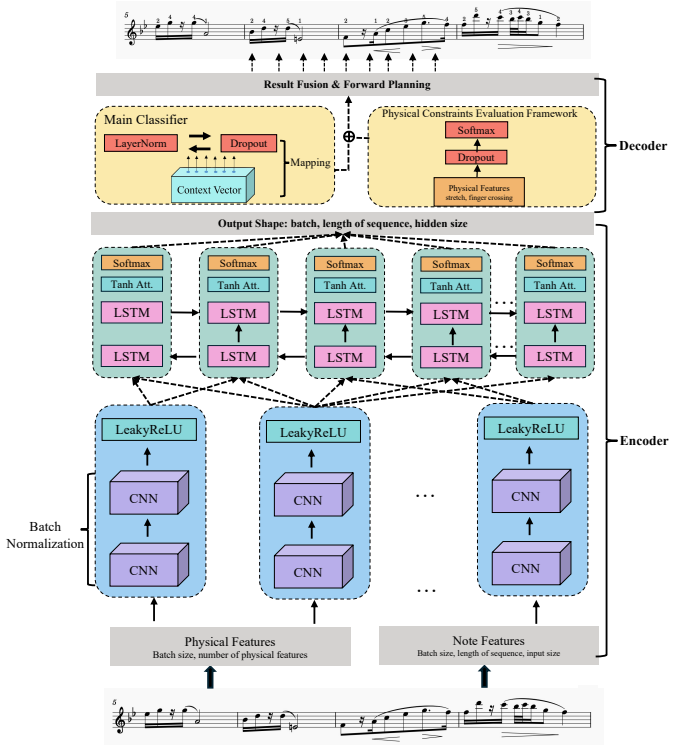


Figure 2. Hybrid Model Flow

2. **BiLSTM Sequential Modeling**: The bidirectional LSTM layer captures long-range dependencies:

$$H_{lstm} = BiLSTM(H_{conv})$$

3. **Dual Classification Approach**: The model employs two classifiers - a main classifier and a physical constraint classifier:

$$\hat{y}_{main} = W_{out} \cdot c + b_{out}$$

$$\hat{y}_{phys} = \text{PhysNet}(\Phi)$$

where $W_{out} \in R^{h \times n_c}$ is the output weight matrix with h being the hidden dimension and n_c the number of classes, $b_{out} \in R^{n_c}$ is the bias vector, and PhysNet is the physical constraint network defined as:

$$\text{PhysNet}(\Phi) = \text{softmax}(W_{\Phi 2} \cdot \text{ReLU}(W_{\Phi 1} \cdot \Phi + b_{\Phi 1}) + b_{\Phi 2})$$

where $W_{\Phi 1}$, $W_{\Phi 2}$, $b_{\Phi 1}$, and $b_{\Phi 2}$ are learnable parameters.

4. **Combined Prediction**: The final prediction integrates both classifiers:

$$\hat{y}_{final} = (1 - \lambda) \cdot \text{softmax}(\hat{y}_{main}) + \lambda \cdot \hat{y}_{phys}$$

where λ is the physical constraint weighting parameter.

This architecture effectively balances musical pattern recognition with physical plausibility constraints, leading to more realistic fingering predictions.

3.5 Attention Mechanism and Forward Planning

The attention mechanism in our model enhances performance by dynamically focusing on the most relevant parts of the input sequence when making fingering decisions. The attention weights α are computed as:

$$e_t = \tanh(W_a H_{lstm,t} + b_a)$$

$$\alpha_t = \frac{\exp(v_a^T e_t)}{\sum_j \exp(v_a^T e_j)}$$

where W_a, b_a, and v_a are learnable parameters. These weights are then used to compute a context vector c that emphasizes important musical patterns.

The forward planning algorithm further enhances decision quality by evaluating potential fingering choices based on their physical plausibility. For each input sequence X, the algorithm:

1. Generates initial predictions using the main network:

$$P(f|X) = \text{softmax}(W_{out} \cdot c + b_{out})$$

2. Selects the top-k most probable fingerings:

$$f_{top-k} = \arg\max_k P(f|X)$$

3. Evaluates each candidate fingering using physical constraints:

$$s(f) = P(f|X) \cdot (1 + PhysNet(\Phi_f))$$

where Φ_f represents the physical features for fingering f.

4. Selects the optimal fingering:

$$f_{optimal} = \arg\max_{f \in f_{top-k}} s(f)$$

This forward planning approach mimics human decision-making by considering both immediate musical context and physical comfort of potential fingerings. The planning depth can be adjusted by parameter d_{max} (maximum recursion depth in the search), with deeper planning yielding more accurate but computationally intensive results.

The forward planning capability distinguishes our model from conventional sequence-to-sequence approaches, allowing it to generate fingerings that not only respect musical patterns but also adhere to the biomechanical constraints of piano playing.

4. EXPERIMENT AND EVALUATION

4.1 Experimental Setup

All experiments were conducted in a controlled environment to ensure reproducibility and fair comparison. Table 1 details the hardware and software configuration used in our experiments.

Component	Specification
CPU	Intel i9-13900X
GPU	NVIDIA RTX 4090 (24 GB)
Framework	PyTorch 2.4

Table 1. Hardware and Software Configuration

The proposed physical-enhanced CNN-BiLSTM model architecture was configured with hyperparameters optimized for piano fingering prediction as presented in Table 2.

All models were implemented using the PyTorch framework and trained with the Adam optimizer. We employed

Parameter	Value
Hidden size	192
CNN layers	2
BiLSTM layers	2
BiLSTM direction	Bidirectional
Dropout rate	0.2
Physical constraint features	12
Physical weight (λ)	0.3
Batch size	32
Learning rate	0.005
Weight decay	1e-5
Optimizer	Adam
Scheduler factor (γ)	0.5
Planning depth (d_{max})	3
Top-k candidates	3

Table 2. Model Configuration Parameters

a learning rate scheduler that reduces the rate by a factor of 0.5 when validation loss plateaus. Cross-validation was performed with each model trained for 40 epochs to ensure convergence while using early stopping with a patience of 10 epochs to prevent overfitting.

4.2 Dataset Statistics

Our experiments were conducted on two primary datasets: the Piano Interpretation Generation (PIG) dataset and the ThumbSet dataset. The PIG dataset consists of 309 pieces, while the ThumbSet contains a much larger collection of 178,737 pieces.

Table 3 presents the distribution of notes by hand across both datasets.

Hand	PIG Dataset		ThumbSet	
	Count	%	Count	%
Right hand	54,239	54.2	4,813,743	61.0
Left hand	45,801	45.8	3,083,487	39.0
Total	100,040	100.0	7,897,230	100.0

Table 3. Note Distribution by Hand

The annotation percentage differs significantly between datasets, with the PIG dataset being fully annotated (100%), while ThumbSet contains partial annotations with only 51.7% of notes having fingering information.

For finger distribution analysis, Table 4 presents the percentage distribution of fingers in both datasets.

Finger	PIG Dataset		ThumbSet	
	Right	Left	Right	Left
1 (thumb)	26.5	32.0	13.3	14.8
2 (index)	23.9	21.1	11.5	11.5
3 (middle)	19.8	14.4	11.9	9.4
4 (ring)	14.4	10.3	8.0	5.7
5 (pinky)	15.3	22.3	7.3	9.9
Unannotated	0.0	0.0	48.0	48.8

Table 4. Finger Distribution (%)

The data shows significant differences in finger distribution patterns between the two datasets, with the thumb and index fingers more frequently used in the PIG dataset compared to ThumbSet. These imbalances were addressed during training through appropriate weighting strategies.

For our experimental evaluation, we used a standard 70%/15%/15% split for training, validation, and testing, ensuring that pieces remained intact rather than splitting across measures to maintain musical coherence.

4.3 Evaluation Metrics

To comprehensively evaluate piano fingering models, we adopt the metrics proposed by Nakamura et al. [3] and extend them to capture different aspects of fingering quality. Given that piano pieces often have multiple valid fingering annotations in datasets, we utilize the following set of metrics:

- M_{gen} (General Matching Rate): The average matching rate between predicted fingerings and all ground truth labels across the test set. For a test set with N pieces, M_{gen} is calculated as:

$$M_{gen} = \frac{1}{N} \sum_{i,j} a_{i,j} \tag{1}$$

  where $a_{i,j}$ represents the matching rate between the model's prediction and the j-th fingering label sequence of the i-th piece. For specific i and j, $a_{i,j}$ is computed as:

$$a_{i,j} = \frac{1}{n} \sum_{t=1}^{n} y(t) \text{ XNOR } \bar{y}(t) \tag{2}$$

  where n is the sequence length, $\bar{y}$ is the model-generated fingering, y is the ground truth label, and XNOR represents the logical equivalence operation.

- M_{high} (Highest Matching Rate): For multi-labeled piano scores, this metric uses the highest matching rate among all labels for each piece:

$$M_{high} = \frac{1}{N} \sum_{i} \max_{j} a_{i,j} \tag{3}$$

- M_{soft} (Soft Matching Accuracy): This metric acknowledges that pianists often have multiple valid fingering options for the same passage. It allows for acceptable alternative fingerings when evaluating predictions.

- M_{cp} (Change Position Rate): This metric accounts for the sequential characteristics of fingering by measuring hand position changes. Two types of position changes are considered: finger-crossing (using the thumb to cross fingers and change hand position) and shift movement (changing hand position without crossing fingers). M_{cp} is calculated by normalizing the number of position changes in the predicted fingering with the average number of position changes in expert annotations.

The M_{cp} metric is particularly valuable as it does not require direct comparison with human annotations for evaluation. It captures the biomechanical efficiency of the predicted fingerings, as pianists typically aim to minimize hand position changes to reduce physical effort while maintaining musical expression [11, 19].

It is important to note that piano fingering is not unique, and any piece may have multiple valid fingering solutions. Therefore, these metrics collectively provide a more nuanced evaluation than traditional accuracy measures, better aligning with the practical considerations of piano performance.

4.4 Experimental Results

4.4.1 Ablation Study

To analyze the contribution of distinct components in our proposed methodology, we conducted an ablation study by systematically removing key modules from the complete architecture. Table 5 summarizes the results on the PIG test set.

Model Configuration	$M_{gen}(\uparrow)$	$M_{high}(\uparrow)$	$M_{soft}(\uparrow)$	$M_{cp}(\uparrow)$
Complete model	82.6	92.1	97.6	0.996
No physical constraints	71.2	74.4	95.8	0.984
No attention mechanism	81.5	89.8	97.4	0.998
No forward planning	80.9	89.2	97.5	0.994
Basic model	72.4	76.2	95.5	0.991

Table 5. Ablation Study Results

The Complete model achieves the highest overall performance ($M_{gen} \approx 82.6\%$, $M_{high} \approx 92.1\%$). Removing physical constraints results in the most significant performance degradation (M_{gen} drops by 11.4 points), confirming that biomechanical priors are crucial for generating realistic fingerings. The attention mechanism and forward planning (beam search) contribute marginally to the general matching rate but are essential for reaching the highest possible consistency (M_{high} and M_{cp}).

Additionally, we conducted feature ablation experiments to assess the impact of different physical constraint categories. Table 6 details these findings.

Features	$M_{gen}(\uparrow)$	$M_{high}(\uparrow)$	$M_{soft}(\uparrow)$	$M_{cp}(\uparrow)$
All physical features	82.6	92.1	97.6	0.996
Only basic	72.3	75.8	96.0	0.982
Basic + spatial	81.6	90.6	97.3	0.996
Basic + temporal	82.0	90.9	97.4	0.996
Basic + hand	81.3	90.3	97.2	0.997
Basic + fingering	81.4	90.1	97.3	0.994

Table 6. Feature Ablation Results

The results indicate that no single physical feature dominates; rather, they work synergistically. While utilizing all physical features set yields the best results ($M_{gen} \approx 82.6\%$), individual additions like spatial (stretch/crossing) and temporal (chord context) features already provide a substantial boost over the basic feature set (M_{gen} increases from $\sim 72\%$ to $\sim 81 - 82\%$). This suggests that both geometric hand constraints and rhythmic context are vital for accurate prediction.

4.4.2 Comparison with Other Algorithms

We benchmarked our Hybrid CNN-BiLSTM approach against standard baselines (CNN, BiLSTM) and recent autoregressive models (ArLSTM, ArGNN) [4], using Base, Word2Vec [7], and Physical features. Table 4.4.2 presents the comparative results on both the PIG dataset and ThumbSet dataset.

The performance on ThumbSet is consistently lower across all models compared to the PIG dataset. This discrepancy is likely attributable to the large volume of noisy labels in ThumbSet; achieving higher accuracy may require more precise data cleaning and extended training schedules.

The experimental outcomes highlight several key findings:

1. Physical Constraints are Decisive: Across all architectures, incorporating physical features consistently outperforms Base and Word2Vec inputs. The jump in M_{gen} (from $\sim 70\%$ range to $> 80\%$ range) validates our hypothesis that explicit biomechanical knowledge is more effective than implicit embedding learning for this task.

2. Hybrid Architecture Competitiveness: Our proposed Hybrid CNN-BiLSTM with physical constraints achieves state-of-the-art performance ($M_{gen} \approx 82.6\%$, $M_{high} \approx 92.1\%$), effectively matching or slightly surpassing the strong BiLSTM baseline. It notably outperforms the CNN-only model and Word2Vec-based variants.

3. Word2Vec Limitations: Contrary to some prior works, Word2Vec features did not yield superior performance compared to Base features in our experiments. This implies that for fingering, precise pitch/time intervals are more discriminative than semantic pitch embeddings.

In summary, the combination of a Hybrid CNN-BiLSTM architecture with a rich set of physical constraint features and beam-search planning offers a robust and effective solution for automatic piano fingering, balancing local pattern recognition with global biomechanical consistency.

4.5 Case Study

To demonstrate the practical effectiveness and interpretability of our Hybrid CNN-BiLSTM model, we present detailed qualitative analyses of two contrasting musical textures: a monophonic Baroque piece requiring high finger mobility, and a complex polyphonic Impressionist work requiring voice segregation.

4.5.1 Monophonic Analysis

For the monophonic analysis, we selected the opening measures of J.S. Bach's *Two-part Invention in C Major* (BWV 772). This piece is characterized by scalar runs and broken arpeggios that demand high-velocity precision and efficient hand positioning.

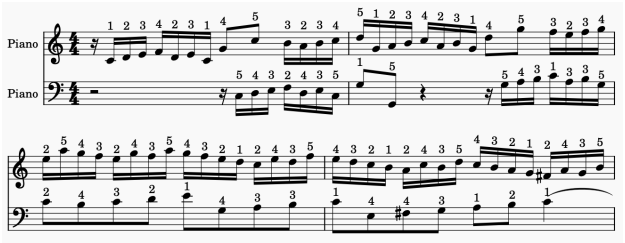


Figure 3. Generated fingering for J.S. Bach, Invention No. 1 in C Major.

Model	Feature Type	PIG Dataset				ThumbSet			
		$M_{gen}(\uparrow)$	$M_{high}(\uparrow)$	$M_{soft}(\uparrow)$	$M_{cp}(\uparrow)$	$M_{gen}(\uparrow)$	$M_{high}(\uparrow)$	$M_{soft}(\uparrow)$	$M_{cp}(\uparrow)$
CNN	Base	72.1	75.7	96.1	0.991	67.8	70.5	93.2	0.985
BiLSTM	Base	73.1	77.2	96.0	0.988	69.4	72.8	93.8	0.982
Hybrid CNN-BiLSTM	Base	70.3	73.2	95.7	0.984	66.5	69.1	92.5	0.979
CNN	Word2Vec	68.6	71.5	94.8	0.977	64.2	66.9	91.0	0.968
BiLSTM	Word2Vec	68.9	71.6	94.4	0.974	65.1	67.4	91.5	0.965
Hybrid CNN-BiLSTM	Word2Vec	70.5	73.5	95.9	0.979	66.8	69.7	92.1	0.971
CNN	Physical	81.7	90.7	97.3	0.996	76.5	84.2	95.4	0.989
BiLSTM	Physical	82.4	91.8	97.5	0.994	77.8	86.5	95.8	0.987
ArGNN	Physical	80.9	89.4	97.5	0.991	75.2	83.1	95.1	0.985
ArLSTM	Physical	80.6	90.4	97.0	0.994	74.9	84.8	94.7	0.988
Hybrid CNN-BiLSTM	**Physical**	**82.6**	**92.1**	**97.6**	**0.996**	**78.1**	**87.3**	**96.0**	**0.990**

Table 7. Comparison with Other Algorithms (PIG Dataset and ThumbSet)

As illustrated in Figure 3, the model generates a fingering strategy that prioritizes horizontal smoothness. Notably, in measures 3 and 4, which feature continuous descending sixteenth-note runs, the model predicts a sequence that facilitates rapid finger mobility without "locking" the hand or creating awkward crossings. This confirms the model's ability to plan ahead for high-density note sequences, likely a result of the BiLSTM's bidirectional context and the forward planning mechanism.

However, a limitation is observed in the left hand (measures 2, 3, and 4). The model occasionally assigns the same finger to two consecutive distinct pitches (specifically on the transition between A and B). This suggests that while the global optimization is effective, the local penalty for consecutive same may need fine-tuning to better penalize non-legato slides in non-chromatic passages.

4.5.2 Polyphonic Analysis

To evaluate the model's capability in handling multiple simultaneous voices, we analyzed a passage from Ravel's *Pavane pour une infante défunte*. This excerpt features a distinct melody in the upper register of the right hand, an oscillating inner accompaniment, and wide harmonic support in the left hand.

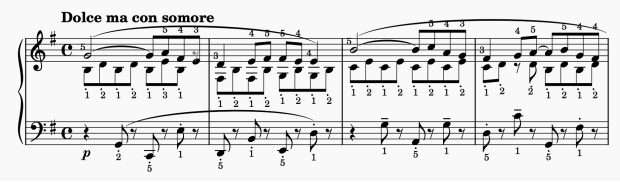


Figure 4. Generated fingering for Ravel, *Pavane pour une infante défunte*.

The generated result (Figure 4) demonstrates a sophisticated understanding of voice leading. In the right hand, the model effectively segregates the hand into two functional zones: the upper melody is assigned predominantly to fingers 3, 4, and 5, while the inner accompaniment utilizes fingers 1, 2, and 3. This stratification prevents "collision" between the melodic lines. Furthermore, for the repetitive inner figures (such as the B-D and C-E oscillations), the model adopts a very human-like 1-2-1-2 alternating pattern. This reflects the model's learning of biomechanical comfort, as alternating strong fingers (thumb and index) is significantly more ergonomic than using a single finger or weaker finger combinations for repetitive dyads.

In the left hand, the texture consists of wide arpeggiated intervals, often exceeding a sixth. Because the model incorporates physical constraints regarding hand span ($\phi_{stretch}$), it correctly identifies the optimal strategy of using the 5th and 1st fingers (5-1-5-1) for these wide leaps (e.g., C-C-D-B-E...), maximizing reach and stability.

Despite these successes, the polyphonic complexity introduces a trade-off. In ensuring the smooth continuity of individual voices, the model occasionally produces vertical intervals that may be physically demanding for smaller hands. For instance, certain dyads (simultaneous notes like D-B or E-C) are assigned a 2-5 fingering to maintain the voice separation. This indicates that future iterations of the physical constraint module might benefit from a dynamic weighting system that balances horizontal voice continuity against vertical hand span comfort, particularly for chord-like structures embedded within polyphony.

5. CONCLUSION

In this paper, we addressed the challenge of automatic piano fingering by developing a hybrid CNN-BiLSTM model with integrated physical constraints. Our approach bridges the gap between purely data-driven methods and biomechanical principles essential for playable fingering solutions.

The experimental results demonstrate conclusively that our hybrid architecture outperforms existing methods across all evaluation metrics. By combining CNN-based local pattern extraction with BiLSTM-driven sequential dependency modeling, our model captures both the spatial and temporal aspects of piano fingering. The integration of physical constraints proved particularly impactful, with ablation studies showing significant performance degradation when these constraints were removed.

Our two-stage planning approach with attention mechanisms effectively addresses the limitations of purely statistical methods by allowing the model to optimize global fingering coherence. The attention component helps the model focus on relevant parts of the musical context, while forward planning ensures biomechanical feasibility across longer sequences.

The comprehensive evaluation framework we employed—M_{gen}, M_{high}, M_{soft}, and M_{cp} metrics—provides a nuanced assessment of fingering quality beyond simple accuracy measures. This evaluation approach better aligns with practical piano performance considerations, where multi-

ple valid fingering solutions often exist.

Despite the advances made, several limitations remain. The current model does not fully capture individualized playing styles based on hand size or technique preferences. Additionally, while our approach works well on both fully annotated (PIG) and partially annotated (ThumbSet) datasets, the available training data represents only a fraction of the vast piano repertoire.

Future work could explore player-specific customization to account for individual anatomical differences and skill levels. Users may submit a small set of scores annotated with their own historical fingerings; an LLM-based API can then distill the user's stylistic regularities and preferences, which are injected as priors into the physical constrainer. Integration with automatic music transcription systems could extend the practical applications of our approach. Additionally, exploring transformer-based architectures might further improve the model's ability to capture long-range dependencies in complex musical passages.

By successfully integrating deep learning techniques with domain-specific biomechanical knowledge, our research contributes to the development of more effective piano performance assistive technologies, with potential applications in music education, composition, and digital score interfaces.

6. REFERENCES

[1] A. Telles, "Piano fingering strategies as expressive and analytical tools for the performer," *Contemporary Piano Music: Performance and Creativity*, vol. 151, 2021.

[2] E. Clarke, R. Parncutt, M. Raekallio, and J. Sloboda, "Talking fingers: An interview study of pianists' views on fingering," *Musicae Scientiae*, vol. 1, no. 1, pp. 87–107, 1997.

[3] E. Nakamura, Y. Saito, and K. Yoshii, "Statistical Learning and Estimation of Piano Fingering," 2020. [Online]. Available: https://arxiv.org/abs/1904.10237

[4] P. Ramoneda, D. Jeong, E. Nakamura, X. Serra, and M. Miron, "Automatic piano fingering from partially annotated scores using autoregressive neural networks," in *Proceedings of the 30th ACM International Conference on Multimedia*, 2022, pp. 6502–6510.

[5] S. Saha, F. Bovolo, and L. Bruzzone, "Change detection in image time-series using unsupervised LSTM," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2020.

[6] L. Shan, Y. Liu, M. Tang, M. Yang, and X. Bai, "CNN-BiLSTM hybrid neural networks with attention mechanism for well log prediction," *Journal of Petroleum Science and Engineering*, vol. 205, p. 108838, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S092041052100499X

[7] Y.-C. Chen, S.-F. Huang, H.-y. Lee, Y.-H. Wang, and C.-H. Shen, "Audio Word2vec: Sequence-to-Sequence Autoencoding for Unsupervised Learning of Audio Segmentation and Representation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 9, pp. 1481–1493, 2019.

[8] T. Su, "Analysis of Piano Playing Techniques Research Using Long Short-Term Memory Networks (LSTM)," in *Proceedings of the 2024 International Conference on Machine Intelligence and Digital Applications*, ser. MIDA '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 511–516. [Online]. Available: https://doi.org/10.1145/3662739.3672177

[9] H. Guan, Z. Yan, and T. Hsu, "Automatic Piano Fingering Estimation Using Recurrent Neural Networks," 2021.

[10] M. Gellrich and R. Parncutt, "Piano Technique and Fingering in the Eighteenth and Nineteenth Centuries: Bringing a Forgotten Method Back to Life," *British Journal of Music Education*, vol. 15, no. 1, p. 5–23, 1998.

[11] R. Parncutt, J. A. Sloboda, E. F. Clarke, M. Raekallio, and P. Desain, "An Ergonomic Model of Keyboard Fingering for Melodic Fragments," *Music Perception*, vol. 14, no. 4, pp. 341–382, 07 1997. [Online]. Available: https://doi.org/10.2307/40285730

[12] J. P. Jacobs, "Refinements to the ergonomic model for keyboard fingering of Parncutt, Sloboda, Clarke, Raekallio, and Desain," *Music Perception*, vol. 18, no. 4, pp. 505–511, 2001.

[13] E. Nellåker and X. Lu, "Optimal piano fingering for simple melodies," 2014.

[14] M. Balliauw, D. Herremans, D. Palhazi Cuervo, and K. Sörensen, "A variable neighborhood search algorithm to generate piano fingerings for polyphonic sheet music," *International Transactions in Operational Research*, vol. 24, no. 3, pp. 509–535, 2017. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/itor.12211

[15] L. Qiang, W. Zhengbiao, and G. Xin, "Piano fingering generation with deep musical score feature fusion," *Journal of Intelligent Systems*, vol. 18, no. 6, pp. 1287–1294, 2023.

[16] H. Zhao, X. Guan, and Q. Li, "Estimation of Playable Piano Fingering by Pitch-difference Fingering Matching Model," *arXiv preprint arXiv:2108.09058*, 2021.

[17] W. Gao, S. Zhang, N. Zhang, X. Xiong, Z. Shi, and K. Sun, "Generating Fingerings for Piano Music with Model-Based Reinforcement Learning," *Applied Sciences*, vol. 13, no. 20, 2023. [Online]. Available: https://www.mdpi.com/2076-3417/13/20/11321

[18] A. P. Iman and C. W. Ahn, "A Model-Free Deep Reinforcement Learning Approach to Piano Fingering Generation," in *2024 IEEE Conference on Artificial Intelligence (CAI)*. IEEE, 2024, pp. 31–37.

[19] C.-Z. A. Huang, A. Vaswani, J. Uszkoreit, N. Shazeer, I. Simon, C. Hawthorne, A. M. Dai, M. D. Hoffman, M. Dinculescu, and D. Eck, "Music Transformer," 2018. [Online]. Available: https://arxiv.org/abs/1809.04281

Hexagram-Based Semantic Composition: Discretizing Embedding Spaces into Symbolic Compositional States for Improvised Performance

Zhang Yuan
Central Conservatory of Music
zhang_yuan@ccom.edu.cn

Zhang Xinran
Central Conservatory of Music
zxinran@ccom.edu.cn

ABSTRACT

Recent AI-assisted composition systems often emphasize sound generation and continuous parameter control, treating semantic embeddings as latent spaces for producing musical material. This paper proposes an alternative approach in which semantic computation supports compositional decision-making rather than sound generation. Building upon prior work on semantic-driven digital scores, we introduce a method that extracts semantic embeddings from a composer-defined corpus and intentionally reduces and discretizes the resulting continuous space into hexagram indices. These indices are interpreted as symbolic compositional states, not as predictors of musical content. Each state configures structural conditions—such as the balance between fixed and open score layers, performer agency, and degrees of stability and change—within which musical behavior may emerge. Hexagrams neither encode musical materials nor prescribe performer actions; instead, they function as compact interfaces for selecting and constraining compositional worlds. An artistic example demonstrates how this hexagram-based discretization supports interpretable, repeatable, and open-ended improvisational performance within a digital score framework. By combining semantic embedding reduction with a highly compressed symbolic system, this work reframes AI-assisted composition as a decision-oriented practice that assists composers in shaping musical worlds rather than automating sound production.

1. INTRODUCTION

In the past decade, AI has entered compositional workflows not only as a generator of sounds and symbols, but as a framework for representing, selecting, and constraining musical possibilities. Within AI-assisted composition, semantic embeddings derived from textual or multimodal corpora are increasingly employed to inform musical processes.[1] However, in many existing systems, these embeddings function primarily as latent control spaces that directly drive sound generation or continuous parameter modulation.[2]-[4] While effective for producing musical material, such approaches often collapse semantic representation into sonic output, limiting interpretability at the compositional level and reducing opportunities for performer agency, structural reflection, and repeatable decision-making.

This paper proposes an alternative framework in which semantic computation informs composition through symbolic state configuration rather than sound generation. Building upon prior work on semantic-driven digital scores [5, 6], we introduce a hexagram-based semantic composition method that intentionally reduces and discretizes continuous embedding spaces into a small set of enumerable indices. In this paper, "hexagram" refers to the six-line figure of the I Ching, used here not for divination but as a discrete symbolic model for compositional state selection. Rather than imposing an I Ching worldview on performers, the hexagram is adopted as a culturally situated and structurally explicit six-line binary framework for organising layered musical behaviour. The resulting states configure score-level conditions—such as degrees of openness, distributions of performer agency, and tendencies toward stability or change—within which musical behaviour may emerge through performance. Within a digital score framework, these states operate as semantic decision interfaces that support open-ended improvisation while maintaining structural coherence, thereby reframing AI-assisted composition as a process of symbolic state selection rather than automated sound production.

2. FRAMEWORK OVERVIEW: FROM SEMANTIC SPACE TO HEXAGRAM STATES

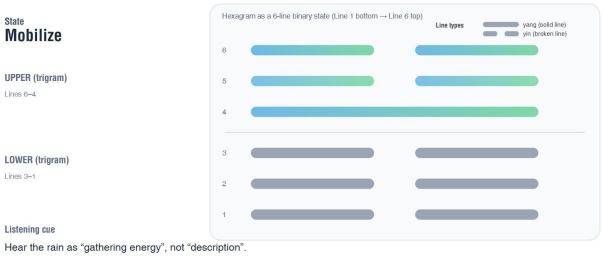


Figure 1. Hexagram as a six-line binary compositional state. In this paper, the term refers to the six-line I Ching figure, read from bottom (Line 1) to top (Line 6). Yang lines are solid and yin lines are broken; the upper and lower trigrams together form a compact symbolic state.

Copyright: © 2026 Zhang Yuan et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License 3.0 Unported, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

The hexagram configures score-level conditions rather than encoding notes, sounds, or fixed performer actions.

2.1 Hexagrams as Symbolic State Structures

As shown in Figure 1, a hexagram is a six-line binary figure from the I Ching (Yijing), a classical Chinese symbolic system [7], composed of yin (broken) and yang (solid) lines and yielding 64 discrete configurations. In the present framework, the hexagram is used as a structural state model rather than as a divinatory or textual interpretive system. Each hexagram serves as a compact, enumerable, and interpretable state label for the digital score. It does not encode pitches, rhythms, or instrumental techniques directly. Instead, it configures score-level conditions such as the balance between fixed and open layers, the distribution of performer agency, and tendencies toward stability, activation, or dissipation over time. Performers then realise these conditions through instrument-specific gestures and sonic materials during performance. In this sense, the hexagram operates as an intermediary between continuous semantic representations and discrete compositional organisation.

2.2 System Pipeline Overview



Figure 2. Overview of the semantic-to-symbolic pipeline: Text corpus → Embedding extraction → Dimensionality reduction → Quantization (1 – 8, 1 – 8) → Hexagram state → Score conditions (layers/agency/stability-change). The discretized pair indexes a symbolic compositional state that configures score-level conditions rather than generating musical materials or prescribing performer actions.

To enable robust and interpretable state selection, the system then performs reduction and discretization. The vector is projected into a low-dimensional semantic plane via a dimensionality reduction method (t-SNE in the current implementation). The resulting two-dimensional coordinates are then normalized and uniformly quantized into integer values from 1 to 8, producing a pair of discrete indices. This step is central to the method: rather than preserving maximal semantic detail, the discretization intentionally prioritizes stable, enumerable outcomes over continuous modulation.

The discretized index pair is then mapped onto a hexagram state. This mapping does not treat the indices as musical information; instead, it uses them as symbolic tokens for selecting a compositional state. Once selected, the hexagram state configures the conditions of the digital score, including the distribution of fixed versus open layers, the allocation of agency between performers, and the structural tendency toward stability, activation, or dissipation across the six-line form. The musical result is thus shaped by a clear separation of roles: continuous semantic embeddings support computational selection, while discrete symbolic states support compositional structuring and performer interpretation.

3. COMPUTATIONAL SEMANTIC REPRESENTATION METHOD

Building upon the conceptual framework outlined in Section 2, this section details the specific computational procedures used to implement the proposed semantic-to-symbolic pipeline.

3.1 Semantic Feature Extraction from Textual Corpus

Semantic features are extracted using the LaBSE (Language-agnostic BERT Sentence Embedding) model [8]. LaBSE is a pretrained multilingual sentence encoder designed to project semantically related sentences into a shared embedding space. In this work, LaBSE is employed exclusively as a semantic representation model, not as a classifier, predictor, or generative component. Each sentence is processed independently and encoded into a fixed-length embedding vector that captures high-level semantic characteristics of the text. These embeddings are treated strictly as semantic descriptors rather than latent variables for sound synthesis or musical material generation. The model is neither fine-tuned nor evaluated against benchmark datasets; its role is to provide a consistent, pretrained semantic encoding of linguistic inputs. By relying on a pretrained model, the system avoids corpus-specific overfitting and ensures stability and repeatability across runs.

At this stage, no musical, poetic, or symbolic interpretation is imposed. The textual inputs are not analyzed in terms of literary meaning, cultural reference, or affective content, but are instead treated as neutral linguistic artifacts whose semantic properties can be computationally represented. This abstraction allows the embedding space to function as a neutral semantic substrate upon which subsequent computational decisions can be constructed.

3.2 Dimension Reduction and Quantization Strategy

Although LaBSE embeddings reside in a high-dimensional continuous semantic space, operating directly in such a space is impractical for compositional decision-making. High-dimensional representations are difficult to interpret, unstable for symbolic mapping, and poorly suited for reproducible structural control. For this reason, the system deliberately reduces the semantic representation to a low-dimensional form and subsequently quantizes it.

First, the extracted sentence embedding vectors are projected into a two-dimensional semantic space using the t-SNE algorithm [9]. This reduction is not intended to preserve maximal variance or fine-grained semantic relationships, but rather to establish a coarse and interpretable semantic plane suitable for robust state selection. The resulting two-dimensional values are normalized to ensure consistent numerical ranges across different textual inputs. Second, the normalized values are discretized through uniform quantization into integer

values ranging from 1 to 8. This discretization is a deliberate methodological choice rather than a loss of precision. By transforming continuous semantic variation into discrete indices, the system prioritizes repeatability, interpretability, and symbolic compatibility over smooth parameter modulation. Unlike approaches that employ continuous semantic control for expressive interpolation, the present method emphasizes stable, enumerable states.

The output of this stage is a pair of discrete integers for each input sentence. These values no longer function as coordinates within a continuous semantic space, but as computational tokens prepared for symbolic indexing. Their purpose is not to encode musical information, but to serve as inputs for a subsequent mapping process in which discrete values configure compositional conditions. The discretized integer pairs are subsequently mapped onto symbolic state indices. Importantly, the symbolic system itself remains independent of the semantic computation described above: the numerical indices do not encode musical material, nor do they predict or generate sound.

3.3 Encoding Divinatory Randomness within the Computational Pipeline

An additional conceptual alignment emerges from the role of randomness in both the computational and symbolic domains. The dimensional reduction stage employs t-SNE, whose stochastic initialization and neighborhood optimization introduce controlled non-determinism into the projection process. While this randomness is typically regarded as a technical property of the algorithm, it plays a structurally analogous role to randomness in traditional hexagram divination, where contingent procedures (e.g., yarrow stalks or coin tossing) are essential for initiating the divinatory state. In the present system, this stochastic component is not treated as noise to be eliminated, but as an integral part of the semantic-to-symbolic transition. Through dimensional reduction, discretization, and symbolic indexing, the logic of divinatory randomness is effectively encoded into the data pipeline itself. The result is a deliberately constructed, artificial divination process in which randomness, semantic structure, and symbolic outcome are internally unified. This allows the generation of hexagram states to remain grounded in computational logic while preserving the functional role of chance that is central to traditional Bagua-based systems, thereby achieving coherence between algorithmic procedure and symbolic intent.

4. ARTISTIC EXAMPLE: “LISTENING TO RAIN” AS AN IMPROVISED DIGITAL SCORE

This section demonstrates how the proposed semantic-to-symbolic pipeline is realized in a performer-facing digital score. Using Listening to Rain as a case study, we show how a discretized index pair selects a hexagram state and how that state configures layered score conditions for improvised performance.

4.1 Example Input and Hexagram Result

To demonstrate the semantic-to-symbolic method in performance, we present an artistic example titled Listening to Rain. The input corpus consists of five Chinese lines selected by the composer as semantic seeds:

(1) 听雨 *(Listening to Rain)*

(2) 天街小雨润如酥 *(Light rain on the avenue, moist as fine silk)*

(3) 空山新雨后 *(After the fresh rain in the empty mountains)*

(4) 急雨翻银河 *(Sudden rain overturns the Milky Way)*

(5) 风回雨定芭蕉湿，一滴时时入昼禅 *(Wind returns, rain settles; banana leaves are wet—drops fall, again and again, into a midday meditation)*

Following the pipeline described in Sections 2–3, the corpus is embedded, aggregated into a corpus-level representation, reduced to a low-dimensional plane, and quantized into a discrete integer pair. In this instance, the discretized output is (8, 4), indexing the hexagram Lei-Di Yu (雷地豫). The hexagram is not used as a semantic narrative to be decoded into musical material; instead, it functions as a symbolic compositional state that configures the conditions under which musical behavior may emerge.

A brief performer-facing annotation accompanies the state selection as an orientation cue rather than an instruction set: _“The sound of rain gradually gathers; the earth responds. Do not rush to move—movement arises from within.”_ This text frames an improvisational attitude (accumulation, resonance, internally motivated activation) while leaving concrete musical decisions to performers within the state’s constraints and affordances.

4.2 Score Realization: Six-Layer Animated HTML5 Canvas Interface

The selected hexagram state is realized as a single-screen digital score interface implemented in HTML5 Canvas [10] and designed for 16:9 stage projection. The interface presents six horizontal layers corresponding to the six lines (yao) of the hexagram from bottom (Line 1) to top (Line 6). Each layer displays the corresponding yin/yang line glyph (broken for yin, unbroken for yang), providing a persistent symbolic reference to the underlying hexagram state.

However, these six layers are not treated as independent functional modules or prescriptive performance instructions. Rather than assigning specific musical parameters, instrumental techniques, or fixed structural roles to each line, the six-layer interface functions as a unified symbolic field through which performers collectively enter and negotiate a shared musical state.

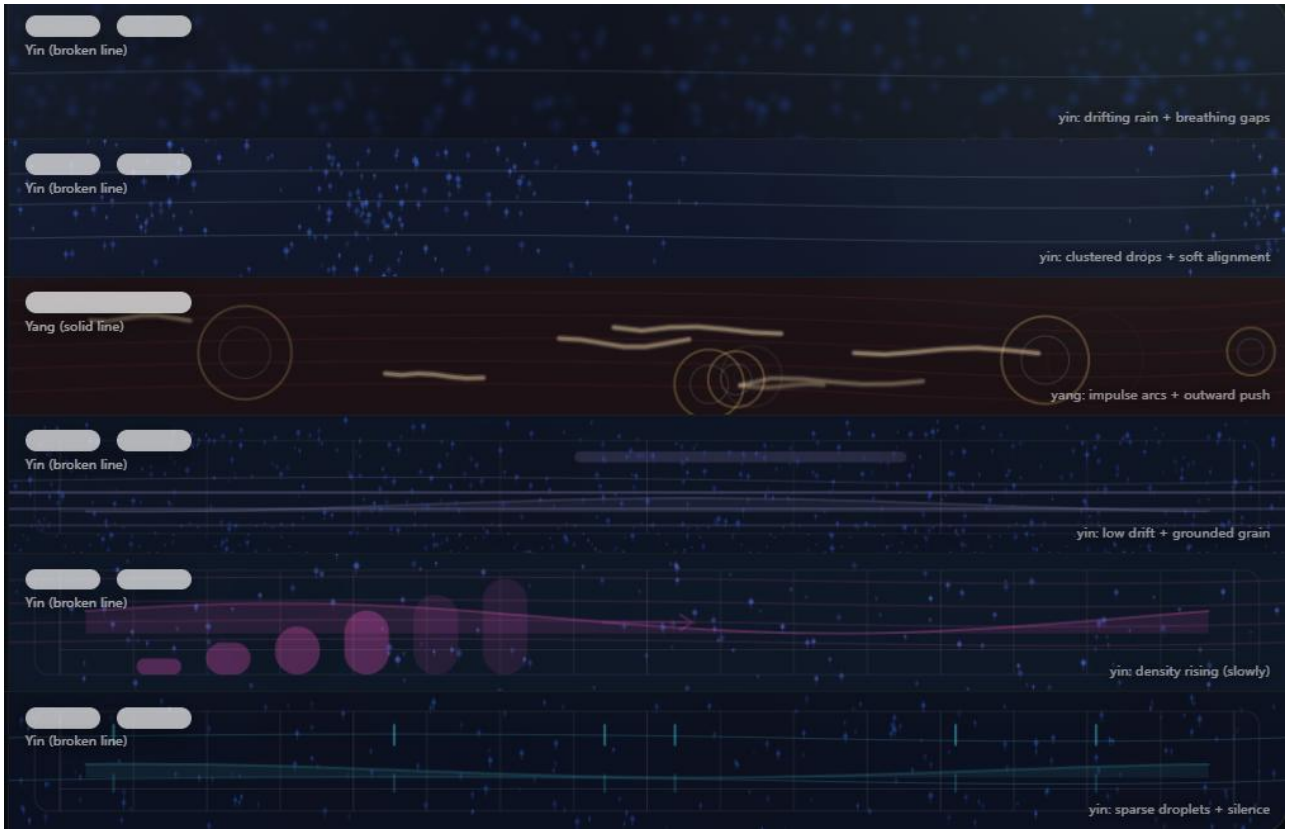


Figure 3. Screenshot of the 16:9 HTML5 Canvas digital score for Listening to Rain (Lei-Di Yu, (8,4)). The interface presents six horizontal layers corresponding to the six lines of the hexagram (Line 1 bottom → Line 6 top), with yin/yang glyphs providing a persistent symbolic reference to the selected state. Rather than prescribing fixed instrumental actions or sectional boundaries, the animated score functions as a shared orientation space through which performers collectively enter, interpret, and shape the musical state.

In performance, the composer provides performers with three interconnected materials: the original poetic text corpus, the corresponding hexagram result, and a brief performer-facing annotation (gua-yao text) that frames the energetic tendency of the state. The animated visual score does not specify exact entrances, durations, or transitions. Instead, it serves as a continuously present orientation space that supports listening, mutual awareness, and emergent structural shaping.

The six lines are therefore understood holistically rather than analytically. Performers do not "play a specific line" in a literal sense; instead, they respond to the overall tension, density, and directional tendency implied by the full hexagram configuration. Musical form is not pre-segmented by the composer, but grows naturally through ensemble interaction as performers inhabit the state together.

This proved particularly significant in rehearsal. Rather than relying on explicit sectional boundaries, the score intentionally leaves formal articulation open, allowing performers to generate structure organically through listening and interaction. In this way, each state developed its own beginning, accumulation, transformation, and closure from within the performance itself.

4.3 Performer Roles and Agency Distribution

The example is designed for flexible performer configurations, with agency distributed across layers to foreground performer decision-making while maintaining structural coherence. Rather than assigning fixed instrumental parts, the score defines layer-based roles that may be realized by different instruments, voices, electronic performers, or media performers according to the performance context.

The lower layers tend to support grounding, continuity, and gradual accumulation, while the upper layers tend to emphasize emergence, transformation, and event-based articulation. These assignments are presented as tendencies rather than fixed parts; in the score interface, they may be indicated as "suggested lead" cues to support rapid performer orientation without closing interpretive space.

Crucially, roles arise from state affordances — the structural conditions configured by the hexagram—rather than from predetermined musical material or instrumentation. Performers choose concrete sounds, techniques, gestures, or media actions locally, while the hexagram-based state constrains global organization through layer design, agency distribution, and stability–change tendencies. For documentation or rehearsal repeatability, the computational pipeline can be executed with recorded settings (e.g., random seeds for the reduction stage), allowing the same corpus to reproduce the same discrete indices and, therefore, the same symbolic state under controlled conditions.

5. DISCUSSION AND CONCLUSION

5.1 What This Adds: Decision-Oriented Semantic Composition

This work proposes a compositional framework in which semantic computation, symbolic reduction, and performance design converge through the use of hexagrams as minimal decision systems. In contrast to prevailing AI-assisted composition approaches that emphasize continuous parameter control or automated sound generation, the proposed method deliberately discretizes semantic representations into symbolic compositional states. Semantic embeddings are not treated as latent spaces for producing musical material, but as inputs for selecting and configuring higher-level structural conditions.

Crucially, the hexagram does not function as a cultural reference, narrative symbol, or divinatory explanation. Instead, it operates as a compact and enumerable interface for configuring compositional states, shaping distributions of performer agency, temporal orientation, and degrees of stability and change. By separating state-level decision-making from material-level execution, the framework preserves interpretability and repeatability while avoiding prescriptive control over sound.

The artistic example Listening to Rain demonstrates how such symbolic states can be realized as a performer-facing digital score. Instrumental or media roles and interactions emerge from state affordances rather than from fixed musical material: performers inhabit complementary functions of continuity, accumulation, transformation, and event-based articulation, negotiated locally within shared structural constraints.

More broadly, the approach illustrates how a historically concise symbolic system can be reactivated as a contemporary compositional tool that mediates between continuous computation and embodied musical action. By employing an extremely compact symbolic structure to address the long-standing tension between control and freedom in improvised and computer-mediated music, this work suggests a pathway for integrating culturally specific symbolic logics into computational music practice without relying on stylistic imitation.

5.2 Limitations and Future Work

The choice of dimensionality reduction and quantization strategy influences the stability of state selection and warrants further investigation. In particular, stochastic components in the reduction process (e.g., initialization and neighborhood optimization) may affect the reproducibility of discrete indices, suggesting the need for controlled parameterization or alternative reduction methods in future implementations. Future work may extend the framework to additional input modalities and include performer-centered studies to evaluate how symbolic state-based scores are interpreted and negotiated in practice.

Acknowledgments

This work was supported by the project "Digital Scores – Investigating the Technological Transformation of the Music Score" (DigiScore, Grant Agreement No. 101002086).

6. REFERENCES

[1] J.-P. Briot, G. Hadjeres, and F.-D. Pachet, *Deep Learning Techniques for Music Generation*, Springer, 2020.

[2] T. Borsos et al., "AudioLM: A Language Modeling Approach to Audio Generation," Proceedings of the 39th International Conference on Machine Learning (ICML), 2022.

[3] J. Copet et al., "Simple and Controllable Music Generation," Proceedings of the 37th Conference on Neural Information Processing Systems (NeurIPS), 2023.

[4] A. van den Oord et al., "WaveNet: A Generative Model for Raw Audio," Proceedings of the 9th ISCA Speech Synthesis Workshop, 2016.

[5] C. Vear, The Digital Score: Musicianship, Creativity and Innovation. Routledge, 2019.

[6] Y. Zhang and X. Zhang, "Chinese Characters as Digital Scores: A Prototype for Explainable, Semantics-Driven Composition", Proceedings of the Tenth International Conference on Technologies for Music Notation and Representation (TENOR 2025), Beijing, China, Oct. 2025.

[7] T. K. Hon, "Chinese Philosophy of Change (Yijing),"Stanford Encyclopedia of Philosophy, 2019.

[8] Feng, F., Yang, Y., Cer, D., Arivazhagan, N., and Wang, W. Language-agnostic BERT sentence embedding. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL), System Demonstrations.

[9] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," Journal of Machine Learning Research, vol. 9, pp. 2579 – 2605, 2008.

[10] L. Vickery, C. Hope, and S. James, "Digital adaptions of the scores for Cage Variations I, II and III," ICMC 2012 Proceedings.

SCMoo: A Live Codeable VR Environment

Charles C Hutchins
University of Kent
charles@celesteh.com

Shelly Knotts
Northumbria University
shelly.knotts@northumbria.ac.uk

ABSTRACT

After the loss of Mozilla Hubs and the end of most Metaverse hype, we present a retro, text- and sound- based VR platform for live coding interactive music in SuperCollider, which is accessible, enjoyable and lower carbon than polygon-based systems. In the 1990s text-based MUDs (Multi-User Dungeons) and MOOs (MUDs Object Oriented) were inhabited by hundreds of users [1]. The communities in these spaces could design any avatars they wanted, which could perform any actions they could describe (limited only by imagination and language) as the medium itself was text. MOOs provided all users with the possibility to add objects, rooms, actions, behaviours and other features to the environment [2] through object-oriented programming. The collaboratively built VR environment was live coded by the users who built features through iterative design within the shared platform. This demo presents SCMoo, which is a reimplementation of a LambdaMOO-like system, written in the musical programming language SuperCollider. SCMoo is a multi-user platform for sound making and role-play.

1. INTRODUCTION

For several years, the live code community has been interested in telematic music and platforms. The Reduce Art Flights [3] campaign and especially "We Won't Fly for Art" pledge of 2009 [4] [5] highlighted the carbon costs of large international live coding events. Using online venues instead also allows for greater participation from people who are unable to travel for reasons of cost, caring responsibilities, disability or border restrictions.

Eulerroom, a website and social media presence for hosting and promoting live streams from the live coding community was founded in 2015 [6] to consolidate the increasing number of live coding streaming events. In early 2019, Alex McLean proposed that the ICLC should become several regional conferences connected telematically. [7]

Browser-based environments such as Estuary [8] and Flok [9], enable collaborative multi-lingual live coding and have lead to projects such as international live coding ensemble SuperContinent[10] – and more recently Pastagang an 'radically open jamming group' of live coders who meet online to jam and perform together [11]. Typically in such environments live coders use chat boxes or code comments to communicate with each other and the audience about the code, errors, or to add social context to the collaboration and performance.

During the Covid-19 lockdowns, graphical VR environments such as Mozilla Hubs were adopted by communities, such as the live coding community, as gathering spaces for networked music performances, collaborative exchange, and social interaction. Mozilla Hubs is no longer funded by the Mozilla Foundation [12] as tech attention drifts towards AI. Adopters of the virtual platform are looking forward for the next thing to replace it. We propose instead looking backwards to text based VR which creates a detailed and robust improvisatory environment, which is also more accessible than Metaverse spaces.

When Neal Stephenson published the science fiction novel Snow Crash in 1993 [13], there were thriving VR communities using text-based MUDs (Multi-User Dungeons) and MOOs (MUDs Object Oriented) [1]. The users of these spaces could create any avatar they wanted, which could perform any action they could describe, limited only by their imagination and language, as the medium was text. MUDs were typically highly hierarchical with access to abilities, game areas and other affordances determined through player experience. MOOs attempted to resist hierarchical structures by providing all users with the possibility to add objects, rooms, actions, behaviours and other features to the environment [2] through object-oriented programming. These affordances facilitated the collaborative building of what MOO's inventor called "text-based virtual realities" [14], live coded by the users, who could dynamically build specific objects and features through iterative design within the shared platform. Although the metaverse has fallen out of fashion, MOOs provide a liberatory, FLOSS approach to building shared virtual space. As low-resource projects, they are a degrowth telepresence, accessible to users of older hardware, people with slow connections, and self-hosters.

2. BACKGROUND

Previous work on music performance in virtual spaces includes the Avatar Orchestra Metaverse based in Second Life, members of which included Pauline Oliveros [15, 16]. There was a flurry of interest more recently, motivated by Covid-19 lockdowns and the tech industry's pivot towards VR - particularly at Facebook [17]. The Live Coding community's interest in telematic concert venues predates the Metaverse hype, as they moved to Mozilla Hubs (a FLOSS system) early in the pandemic [18].

The terrain of performative text description has been previously explored by Craig Latta's "dynamic interactive fiction system", Quoth [19], which is an "executable natu-

Copyright: ©2026 Charles C Hutchins et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License 3.0 Unported, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

ral language" [20] which has been likened to "a 1980s adventure-game text engine" [21]. Latta describes his motivation as accessibility, arguing that Natural Language Processing is more comprehensible for non-programmers in the audience than coding languages would be [19].

Similar ideas have been proposed in the distant past, on the Live Code email list, including IRC integration [22] and a series of live codeable rooms. [23] To our knowledge, neither proposal has ever been implemented.

The SCMoo project extends experiments with LambdaMOO and a SuperCollider bridge, undertaken by Hutchins in 2003, in which users could attach a ringtone-like pitch sequence to themselves and to objects in the room. The custom pitch sequence would play when a state change was observed in the environment.

3. IMPLEMENTATION

The demo presented here – SCMoo – is a prototype reimplementation of a LambdaMOO-like system, written primarily in the musical programming language SuperCollider. SCMoo enables multiple participants and performers to interact through text chat and role-play with the possibility to engage with and create sound making objects in the text-based virtual environment.

When inhabitants interact in the system they receive text and sound responses according to their interactions, and the interactions of their co-inhabitants, within the environment.

An environment has been built by the authors for the purposes of the demo, but - as well as interacting in the current environment - demo attendees can also create new objects, behaviours and sounds to the environment on-the-fly using the SCMoo interface and simple SuperCollider functions.

SCMoo runs in a peer-to-peer like model, managed via a Node server, which copies OSC-like messages to all connected users. Each user holds a local copy of the SCMoo data (which is locally downloaded and stored in a JSON file) to render sound and text on the user's machine. This local sound rendition significantly reduces network overhead when compared with streaming audio.

This use of Node makes Moo network traffic look like normal web traffic. No special ports are opened and tunnelling is not required. This increases user accessibility over using UDP servers like OSCGroups.

The current version of SCMoo and associated SuperCollider libraries are available at `https://codeberg.org/celesteh/SCMoo` released under the GPL license.

4. INTERACTION

The SCMoo JSON file, which users download or share ahead of a performance, contains every MooObject, including all attached data and the code for functions and sound. User interaction can also change or set data during a performance, which is also shared in real-time with others.

When users connect to SCMoo, they are placed in a room. They are provided with a description of the room, which lists notable features and may include hints on what they might wish to do whilst there. A list of objects in the room

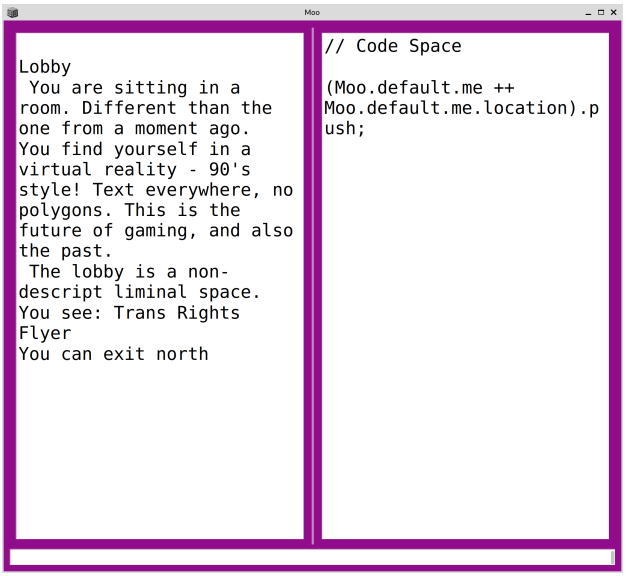


Figure 1. The SCMoo Interface has output on the upper left, a code input area on the upper right and a space for chat messages and MOO interactions on the bottom

is also provided and finally, a list of exits, should they wish to navigate to a different room.

For example, in the current version of SCMoo, one of the rooms has a "trap door" object, which could have the states "open" or "closed". Users interact with MOO objects by typing imperative English sentences in the interface. To open the trap door, a user would type "open trap door". When they enter this command, the text output tells them and others in the room that the "trap door" has been opened. It also changes the "trap door"'s state from "closed" to "open". Users can climb through the trap door when it is open, but not when it is closed.

The "trap door", like all other objects in the Moo, can also have associated audio code. When a user opens the door, other players hear a creaking sound which accompanies that action. This feature can be used for adding sound effects, music, or ambient sound to the environment.

Objects can be created and modified in real-time by all users using the SCMoo interface (which includes an input text area) and SuperCollider functions. New objects and behaviours are propagated to all current users via OSC and saved locally by all connected users.

For example, a user in the bar room might decide to add a piano. They could also add some actions, known as "verbs", which all users could then invoke on the object. Actions attached to verbs can include adding sounds. Through this method users are able to collaboratively build a sound environment in real time.

Users can find out what verbs are already attached to objects by typing "@examine [object name]" in the SCMoo interface.

The ability to dynamically modify the environment makes SCMoo well suited to live coding performances and - with a particular genre slant in the sound code - algoraves. Other potential deployment scenarios for this format include virtual installations and an embedded possibility for user play.

5. WHIMSY

The ability to create new objects along with relatively low barriers to modifying the functionality sets MOOs apart from other networked games. For many users, creating text environments is more accessible than working in 3D graphics, where, by contrast, "user creations often [seem] amateurish" [1]. As the text is a key part of the environment, this requires making a shared reading experience entertaining. "A characteristic of [MOOs] is the common use of puns. Their designers feel obligated to continue the tradition by interjecting a bit of whimsy" [24]. A key hook for user engagement in VR environments is fun [25].

However, 'low barrier' is not 'no barrier'. For SCMoo's remote audiences, there is also a learning curve. "There is nothing trivial about the mental and physical efforts required to begin to orient oneself in MOOs." [26] (Which is not to say that graphical VR is without user challenges [27]).

In the current version of the SCMoo, the authors implemented aspects of whimsy. The user enters into a room modelled on a bar which is grounded in reality. However as users move into further rooms and towards the outer edges of the map the rooms, objects and behaviours become increasingly surreal, playing with the potential of text to generate complex and absurd experiences with no additional time or resource overheads.

6. FUTURE WORK

SuperCollider is not sandboxed, and currently verbs are written in sclang, the SuperCollider language, which is a potential security risk. There is some effort to prevent users from calling system commands or otherwise causing trouble, but a better solution would be to implement the LambdaMOO language and limit use of sclang to highly privileged users, known as "wizards". This will also require the Node server to provide authentication services via a login system.

This centralisation also offers an ability to centrally manage a database of objects. This has the advantage of making users' changes less volatile. This could also reduce network traffic as users would only download objects in their current room or objects that they wish to interact with.

At the time of writing, remote participation has not been an issue as all performances have been in a set, physical location, with the performers' views projected to an audience in the room, however these performances have relied on the remote Node server to relay messages.

While text is generally easier to create than 3D graphics, this system does introduce or exacerbate accessibility issues around language. The prototype assumes levels of English proficiency. This does mirror the construction of many live coding languages – SuperCollider is based in English. However, live coding is a global movement and therefore has linguistic diversity, so it would be desireable to implement other languages in future versions. MUDs and MOOs do exist in Spanish [28], Italian [29], and other languages.

SCMoo has scope for internationalisation, where objects could include descriptions and text output localised to the user's language. Or a database and parser could be built wholly in other languages. As a proof of concept, the demo includes a draft of an Esperanto parser which uses noun endings rather than word order to identify direct and indirect objects in user's command sentences.

7. CONCLUSIONS

The Silicon Valley dream has pivoted to a "long term" approach, in which investors can ignore their participation in the carbon economy to focus instead on humans a thousand or a million years from now. Tech oligarchs are spending their money and research accordingly, so that AI and virtual meeting spaces are stepping stones to running trillions of "people" in an eternally happy simulation while the real earth dies [30, 31].

Instead, in SCMoo, the text-based environment creates a telepresence of the imagination, while the music grounds it in a real lived experience. We present this as an alternative vision, merging embodied and virtual to create the sustainable virtual fantasy spaces imagined by permacomputing enthusiasts [32].

8. REFERENCES

[1] A. A. Reed, "1990: LambdaMOO," May 2021. [Online]. Available: https://if50.substack.com/p/1990-lambdamoo

[2] R. Bartle, *Designing virtual worlds*, ser. New riders games series. New Riders, 2004, tex.lccn: 2003106801. [Online]. Available: https://mud.co.uk/richard/DesigningVirtualWorlds.pdf

[3] M. Andrews, "RAF / Reduce Art Flights," https://reduceartflights.lttds.org/, 2025.

[4] M. Garret and R. Catlow, "'I Will Not Fly for Art' - PledgeBank," Jun. 2009. [Online]. Available: https://web.archive.org/web/20090621075411/http://www.pledgebank.com/wewontflyforart

[5] R. Catlow, "We Won't Fly for Art: Media Art Ecologies," *Culture Machine*, vol. 13, 2012.

[6] A. McLean, "[Livecode] EulerRoom From," May 2015.

[7] ——, "[Livecode] Future Distributed ICLC?" Apr. 2019.

[8] D. Ogborn, J. Beverley, L. N. del Angel, E. Tsabary, A. McLean, and E. Betancur, "Estuary: Browser-based collaborative projectional live coding of musical patterns," in *International Conference on Live Coding (ICLC)*, vol. 2017, 2017.

[9] munshkr, "flok — codeberg.org," https://codeberg.org/munshkr/flok, [Accessed 18-12-2025].

[10] S. Knotts, C. Betancur, A. Khoparzi, M. Laubscher, M. Marie, D. Ogborn, C. Oka, and E. Tsabary, "Negotiating Shared Live Coding Practices in SuperContinent, an Online Laptop Ensemble," *eContact!*, vol. 21, no. 1, 2022.

[11] Pastagang, "Pastagang: Jamming together far apart," in *Proceedings of the first Alpaca conference on Algorithmic Patterns in the Creative Arts*, Sheffield, 2025. [Online]. Available: https://2025.algorithmicpattern.org/proceedings/T7YPKG/paper.html

[12] Mozilla Foundation, "End of support for Mozilla Hubs," Nov. 2024. [Online]. Available: https://support.mozilla.org/en-US/kb/end-support-mozilla-hubs

[13] N. Stephenson, *Snow Crash*. London: Roc, 1993.

[14] P. Curtis, "Mudding: Social phenomena in text-based virtual realities," in *Culture of the Internet*. Psychology Press, 2014, pp. 121–142.

[15] G. F. Martín, "Social and psychological impact of musical collective creative processes in virtual environments; The Avatar Orchestra Metaverse in Second Life," *Music Technol.*, vol. 75, pp. 75–87, 2018.

[16] X. Alarcón and R. Herrema, "Pauline Oliveros: A shared resonance," *Organised Sound*, vol. 22, no. 1, pp. 7–10, Apr. 2017, publisher: Cambridge University Press (CUP).

[17] L. Turchet, "Musical Metaverse: vision, opportunities, and challenges," *Personal and Ubiquitous Computing*, vol. 27, no. 5, pp. 1811–1827, Oct. 2023, publisher: Springer Science and Business Media LLC. [Online]. Available: https://link.springer.com/10.1007/s00779-023-01708-1

[18] Cárdenas, Alexandra, "DEAR LIVE CODING WORLD COMMUNITY," Apr. 2020, accessed: 2025-07-17.

[19] C. Latta, "a dynamic interactive fiction system," Jan. 2011. [Online]. Available: https://netjam.org/quoth/

[20] C. Nilson, "Dvd program notes," *Computer Music Journal*, vol. 35, no. 4, pp. 119–137, 2011.

[21] N. Collins, "Live coding of consequence," *Leonardo*, vol. 44, no. 3, pp. 207–211, 2011, iSBN: 0024-094X Publisher: MIT Press 55 Hayward St., Cambridge, MA 02142-1315, USA journals-info@ mit. edu.

[22] Algomantra, "[Livecode] IRC+livecode?" Sep. 2008.

[23] M. Corso, "[Livecode] an Idea for a Livecode Environment," Apr. 2008.

[24] J. Falsetti, "What the Heck is a MOO? And What's the Story with All Those Cows?." in *Proceedings of the Annual Meeting of the Teachers of English to Speakers of Other Languages*. Long Beach, CA, USA: ERIC, 1995, publisher: ERIC.

[25] J. M. Carroll and J. C. Thomas, "Fun," *ACM SIGCHI Bulletin*, vol. 19, no. 3, pp. 21–24, 1988, iSBN: 0736-6906 Publisher: ACM New York, NY, USA.

[26] K. Parrish, "The Ergodics of Reading MOO. A Non-Trivial Pursuit," *Cybertext Yearbook*, 2002.

[27] M. McGill, D. Boland, R. Murray-Smith, and S. Brewster, "A dose of reality: Overcoming usability challenges in vr head-mounted displays," in *Proceedings of the 33rd annual ACM conference on human factors in computing systems*, 2015, pp. 2143–2152.

[28] Simauria Networks, "Simauria :: Información General :: ¿Qué Es Un MUD?" http://www.simauria.org/general_mud.php.

[29] Rocca d'Argento MUD, "RdAMUD - Client," http://rdamud.afraid.org/, 2017.

[30] A. Crary, "The toxic ideology of longtermism," *Radical Philosophy*, vol. 214, pp. 49–57, 2023.

[31] É. P. Torres, "Why longtermism is the world's most dangerous secular credo | Aeon Essays," Oct. 2021. [Online]. Available: https://aeon.co/essays/why-longtermism-is-the-worlds-most-dangerous-secular-credo

[32] A. Mansoux, B. Howell, D. Barok, and V.-M. Heikkilä, "Permacomputing aesthetics: potential and limits of constraints in computational art, design and culture," *Ninth Computing within Limits*, 2023.

Incisions: Tangible Latent Space Exploration with Three Sound Balls

Matthias Jung
Institute for Music, Science and Engineering
King Mongkut's Institute of Technology Lat Krabang
Bangkok (Thailand)
matthias.ju@kmitl.ac.th

ABSTRACT

This system proposes an interactive, tactile approach to collaboratively exploring machine learning models in real-time. The current work-in-progress design connects three handheld spherical devices (sound balls) to three machine learning models. Equipped with pressure sensors and gyroscopes, the sound balls send readings from an ESP32 as OSC over Wi-Fi to a Max/MSP patch that hosts the model playback. The patch employs a combination of open-source and self-trained models, which are mixed into a master output audible through headphones by the three sound ball players, who collaboratively explore the models within a latent-dimension setup.

Keywords latent space exploration, RAVE, sound balls, tangible multi-user instruments, co-creation, accessible musical instruments

1. INTRODUCTION

Spherical handheld interfaces for musical expression (referred to sound balls here below) have been used as both standalone musical instruments, as well as interfaces for participatory works without integrated playback. They have been proven to be accessible and playful devices, and are well suited for interactive or collaborative creative scenarios. With recent micro-controllers such as the ESP32 it has become easy to send sensor readings over WiFi and fit everything into small, handheld balls with little weight. At the same time, there have been advances made in real-time use of machine learning models, and those models have been integrated in commonly used audio production and performance software during recent years. Here, RAVE (real-time variational auto-encoder) with its nn~ Max/MSP-object is a prominent example, allowing to explore timbre spaces of audio datasets via a set of latent parameters. The suggested system makes use of both developments and suggests to combine the affordances of tactile interfaces with RAVE. In addition, it uses three balls simultaneously to control the latent parameter space among three RAVE models on different levels.

Copyright: ©2026 Matthias Jung et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License 3.0 Unported, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

2. BACKGROUND AND RELATED WORK

2.1 Spherical Handheld Interfaces for Musical Expression

Early examples for sound ball designs include Weinberg et al.'s "Embroidered Musical Ball" [1, 283], and Jensenius et al.'s "Music Balls" [2], both designed to include musical novices as users of those accessible small musical instruments. Many of those instruments explored different flex and force-sensing resistors or conductive thread as sensors, feeding the sensor readings in desktop computers with a cable-bound interface, then mapped to different sounds via MIDI.

The Bluetooth Radio Ball Interface (BRBI) introduced a wireless handheld foam ball integrating a 3-axis accelerometer, transmitting sensor data via Bluetooth as OSC messages for sound generation [3]. While not strictly spherical, Bowen's Soundstone similarly employs a Bluetooth-enabled accelerometer-based interface for gestural input to Max/MSP, extending interaction through integrated visual and haptic feedback using a tri-color LED and a vibrator [4]. Extending the scale and bodily engagement of such devices, the Balance Ball enables full-body interaction through rolling, lifting, and riding gestures. It incorporates multiple sensors, including accelerometer, gyroscope, photodiode, and LEDs, and maps motion data to velocity, pitch, and tempo MIDI messages for musical control [5].

Collaborative ball-based instruments emphasize shared control and accessibility. The StringBall is designed for two performers connected via opposing strings passing through the ball, allowing lateral motion, lifting, and shaking gestures to produce chaotic sound textures [6]. Its simplicity targets inclusive use by non-musicians and children. In contrast, the commercial Oddball [1] supports complex rhythmic interaction through learned bouncing gestures mapped via a smartphone application. While offering advanced musical capabilities, its cost limits scalability for educational contexts and participatory performance.

2.2 Latent Space Navigation in RAVE

Real-time audio synthesis based on prior training of machine learning models with audio datasets have been made accessible in the last years with Somax [7], DDSP [8], or RAVE [9], to name some of the widely used examples. The model used for both training and real-time synthesis in Max/MSP in this work is the latter: Realtime Audio Vari-

[1] Oddball https://oddballism.com/en-eu

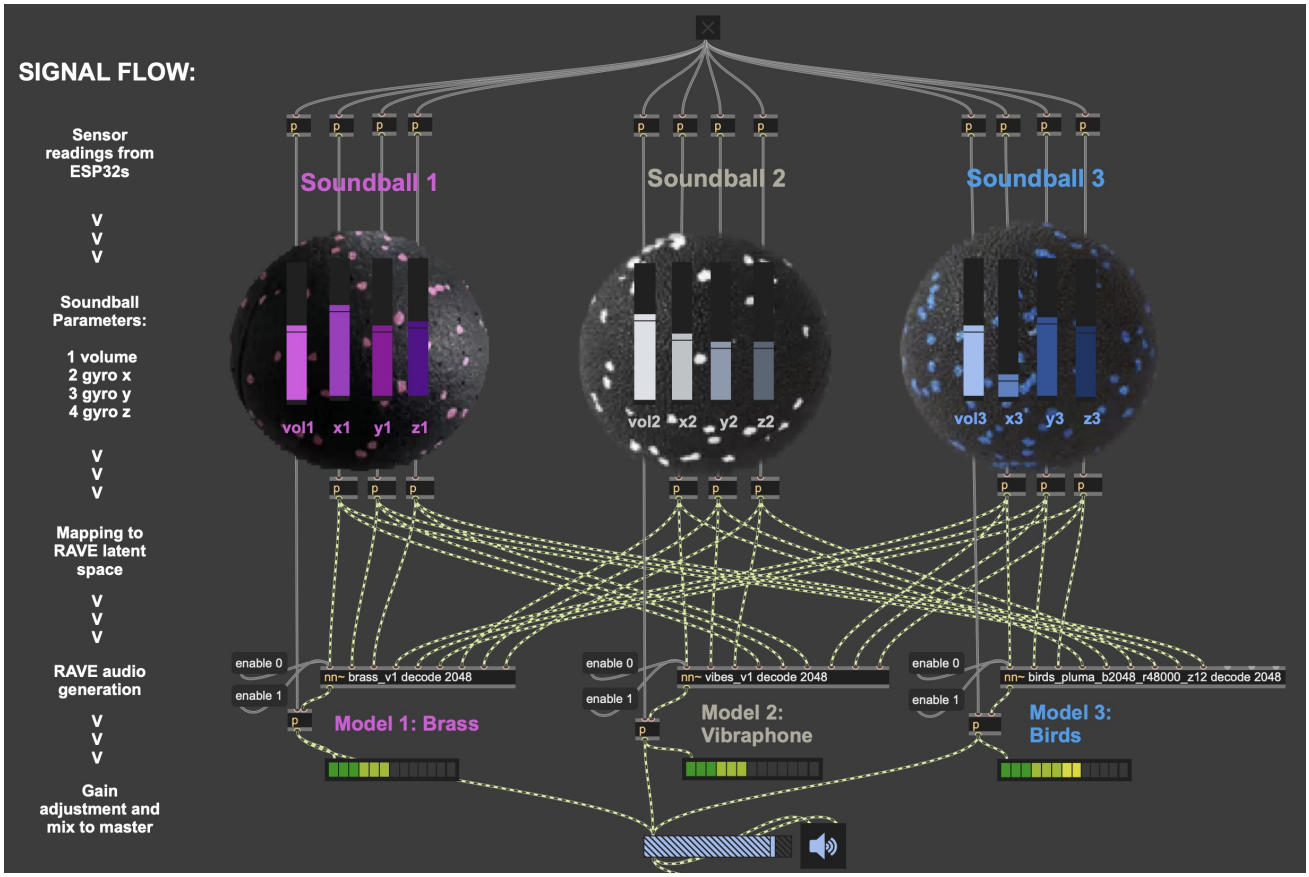


Figure 1. Max/MSP patch for Incisions.

ational autoEncoder (RAVE) [2] . Those tools make it possible to explore creative novelty by interacting with machine learning models trained in advance, then used for musical performance in real-time [10]. RAVE's decode function in the nn∼ object for Max/MSP also allows synthesis without audio input, i.e. the descriptor-based generation via latent space parameters only [11].

3. SYSTEM DESIGN AND ARCHITECTURE

The title of this paper points to the idea of "cutting through the timbre space," very much like an incision. The system uses the feature of the nn∼ object by mapping the sensor readings of the three sound balls to three nn∼ objects using the decode function for descriptor-based generation based on those variables. The signal flow is shown in figure 1 and can be described by the following three stages:

1. Sending and receiving sensor readings: The four sensor readings are being sent from each of the three sound balls' ESP32s to Max/MSP over Wi-Fi [3] .

2. Mapping sensor readings to control parameters in Max/MSP: The pressure sensor reading is mapped to the volume control of each of the three RAVE models. The x,y,z positional readings of the sound balls are mapped to the first three parameters of each or the adjunct model of each sound ball. The gyroscope readings of each sound ball are also mapped to the inlets 4-6, and 7-9, of the other two models respectively. The readings therefore are not only used for one model, but also translated to both other model's latent variables. The fact that the first variables have more influence on the generated audio output of each model takes care of the fact that the main agency of each player is still remains with "their" sound ball.

3. Mixing and playback: All model output is mixed together into a master output based on the pressure readings and played back via an external interface with 3 headphones so that all players hear the same master audio output.

The appended video shows a simulation of the changing values of the sound ball based on random up/down movements, as would be the case when three human players are using the system. However, it has to be noted that humans will move in a very different fashion and especially the volumes are supposed to be controlled in close interaction, i.e. people might listen to each others models, try to hold still for a while, or take turns in their explorations of the timbre spaces. At the time of writing, we also explore the introduction of an additional layer of musical parameters (such as pitch, delay, drive, reverb, etc.) mapped to the latent space parameters, which is showing some promising results for future designs.

[2] https://github.com/acids-ircam/RAVE

[3] This has been the setup for previous participatory work with the sound balls and works seamlessly for three sound balls.

3.1 Sound Ball Interface

As described in section 2, the sound ball follows a tradition of wireless, handheld sensor balls. The author has experimented with different designs during the last year and now built the sound balls used for this work from massage balls, that are made of denser foam material. This makes it easy to control the pressure sensor more accurately in contrast to balls with softer material. The limitation to just one pressure sensor was found to work best when controlling gyroscope parameters, and the mapping of the pressure sensor to the individual volume is easy to understand for novices to regulate "their" contribution to the master mix.

Figure 2 shows a photo of the sound ball to be used for the interactive demo, while figure 3 shows a design sketch including its components. Those are as follows:

1. A massage ball made of hard foam, carved out from the inside to about 1cm and re-assembled after components were built in.

2. An on/off switch and a 3.7V 2000mAh lithium battery pack to power the ball for around 2-3 hours (usually enough for a concert lengths or exhibition period).

3. A loading socket (USB-C) and TP4056 loading circuit.

4. An MCU with Wi-Fi (ESP32) to act as a Websocket client for the Max/MSP-bound server.

5. A pressure sensor (FSR-402) and a gyroscope (MPU-6050).



Figure 2. Current prototype of the Sound Ball.

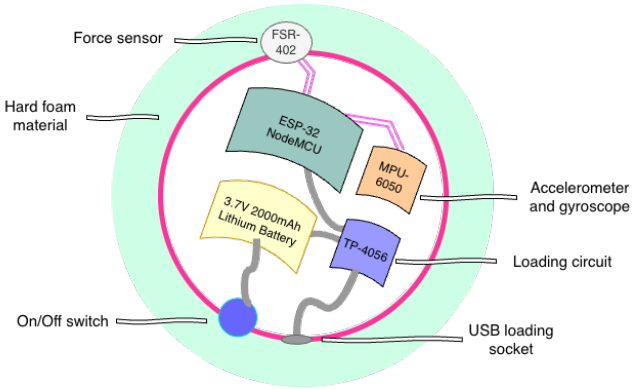


Figure 3. Design of the current prototype of the Sound Ball with components.

3.2 Datasets and Training

For testing the system design, the following three RAVE models were used inside a Max/MSP patch with the nn~ object using its decode function.

- **Model 1** For the first model, the Philharmonia brass recordings were used [4], specifically the instrument categories trumpet, trombone, and tuba. This lead up to a total length of the audio dataset of 15min and 21 sec. The RAVE v2 model was trained at 44.1k sample rate, a compression ratio of 512, 57.6 M total parameters and 9 latent dimensions. This number was chosen due to the fact that each of the gyroscopes from the sound balls are sending 3 values, i.e. 9 values to use for the latent parameters in total. Training was concluded after 32000 steps and the model size is 230.347 MB.

- **Model 2** The second model used for the experiment was generated studio recordings of a Musser vibraphone, recorded and played by the author recently. The total length of that dataset was only 3min and 7sec long and used similar parameters as model 1. However, training was concluded already after 16000 steps given the small dataset.

- **Model 3** As a third model, a RAVE model trained on bird recordings curated by Giacomo Lepri, from the RAVE collection by the Intelligent Instruments Lab [12] was used [5].

4. CONCLUSIONS AND FURTHER WORK

A tangible interface for exploring RAVE models collaboratively in real-time has been suggested for investigation and a prototype has been built based on the previously proposed sound ball instrument. The pressure sensor is used to control the individual volume of each sound ball, while the x,y,z readings of the gyroscope sensor are being used to control the latent space parameters of three RAVE models. Results are preliminary and more models will be generated for further testing the system design. Further work should

[4] The licence for this dataset is CC 3.0 and the data was acquired from https://philharmonia.co.uk/resources/sound-samples.

[5] CC-BY-NC-4.0 licensed model "birds_pluma_b2048_r48000_z12.ts" from https://huggingface.co/Intelligent-Instruments-Lab/rave-models.

also include different setups for controlling the other players' latent space parameters, after the demo has been tested with both musical amateurs and professionals, as well as discussed with experts.

Acknowledgments

This work was kindly supported by the King Mongkut's Institute of Technology Ladkrabang Research Fund (Research Start-up Grant for New Faculty Members, grant number KREF186907). The author thanks both the project mentor Associate Prof. Dr. Sorasak Danworaphong of the Institute of Music, Science and Engineering, King Mongkut's Institute of Technology Ladkrabang, as well as the co-principle project investigator Prof. Dr. Stefan Weinzierl of the Audio Communication Group, Technische Universität Berlin.

5. REFERENCES

[1] G. Weinberg, M. Orth, and P. Russo, "The Embroidered Musical Ball: A Squeezable Instrument for Expressive Performance," *CHI '00 Extended Abstracts on Human Factors in Computing Systems*, pp. 283–284, 2000. [Online]. Available: https://doi.org/10.1145/633292.633457

[2] A. R. Jensenius and A. Voldsund, "The Music Ball Project: Concept, Design, Development, Performance," *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2012. [Online]. Available: http://www.nime.org/proceedings/2012/nime2012_161.pdf

[3] W. S. Yeo, "The Bluetooth Radio Ball Interface (BRBI): A wireless Interface for Music/Sound Control and Motion Sonification," *Proceedings of the International Computer Music Conference*, 2006.

[4] A. Bowen, "Soundstone: A 3-D Wireless Music Controller," *Proceedings of the 2005 Conference on New Interfaces for Musical Expression*, pp. 268–269, 2005.

[5] T. Yamaguchi, T. Kobayashi, and S. Hashimoto, "Balance Ball Interface for Performing Arts," *Human Interface and the Management of Information. Interacting with Information: Symposium on Human Interface 2011, Held as Part of HCI International 2011, Orlando, FL, USA, July 9-14, 2011, Proceedings, Part II*, pp. 440–445, 2011.

[6] W. S. Yeo, J. W. Yoon, and H. Y. Cho, "The Stringball: A Ball Interface With Strings for Collaborative Performance," in *Proceedings of the International Computer Music Conference*, 2007.

[7] J. Borg, "Somax 2: A Real-time Framework for Human-Machine Improvisation," *STMS IRCAM & Aalborg University, Internal Student Report*, 2019.

[8] J. Engel, L. Hantrakul, C. Gu, and A. Roberts, "DDSP: Differentiable digital signal processing," *Proceedings of the Eighth International Conference on Learning Representations*, 2020. [Online]. Available: https://arxiv.org/pdf/2001.04643

[9] A. Caillon and P. Esling, "RAVE: A Variational Autoencoder for Fast and High-Quality Neural Audio Synthesis," in *Proceedings of the 23rd International Conference on Digital Audio Effects (DAFx)*, 2021. [Online]. Available: https://arxiv.org/abs/2111.05011

[10] A. Lunt and S. Trump, "Latent Space Explorer," *Proceedings of the International Conference on AI Music Creativity (AIMC) 2023*, August 29 2023, https://aimc2023.pubpub.org/pub/zgc5j7ha.

[11] A. Caillon, A. Bitton, B. Gatinet, and P. Esling, "Timbre latent space: exploration and creative aspects," 2020. [Online]. Available: https://arxiv.org/abs/2008.01370

[12] Intelligent Instruments Lab, "rave-models (Revision ad15daf)," 2023. [Online]. Available: https://huggingface.co/Intelligent-Instruments-Lab/rave-models

Controlling Musical Parameters in Neurorehabilitation with a Haptic Finger Tracker

Juliana Lüer
Institute of Mechatronics in Mechanics, Hamburg University of Technology and ligeti zentrum, Hamburg, Germany
juliana.lueer@tuhh.de

Christoph Salje
Music Therapy Institute and Teaching Outpatient Clinic, University Hospital Hamburg-Eppendorf and ligeti zentrum, Hamburg, Germany
c.salje@uke.de

Prof. Dr.-Ing. Thorsten A. Kern
Institute of Mechatronics in Mechanics, Hamburg University of Technology and ligeti zentrum, Hamburg, Germany
t.a.kern@tuhh.de

ABSTRACT

Patients in neurorehabilitation often face not only severe motor impairments, but also associated psychological problems. Music therapy can make a valuable supplement to purely verbal psychotherapy, but its use is limited as patients often cannot play conventional musical instruments due to motor skill limitations. This can hinder the psychological recovery, where musical expression is essential. To address this, the Haptic Finger Tracker *was developed, emerging from a project at the* ligeti zentrum*, Hamburg, a collaborative initiative where researchers and artists work on interdisciplinary projects. This paper describes a prototype that translates minimal finger movements into congruent sound and haptic sensations. Technically, the device uses flex sensors and an inertial measurement unit (IMU) to capture a range of small-scale finger movements. Using the Open Sound Control (OSC) protocol, these captured gestures are then translated to control musical elements such as pitch, volume and arpeggios. Simultaneously, a vibrotactile actuator provides haptic feedback aimed at enhancing the user's sense of engagement and embodiment. The resulting portable prototype provides patients with a creative outlet, fostering self-efficacy. This work establishes a technical foundation for future neurorehabilitative tools that utilize multisensory feedback to improve patient outcomes.*

1. INTRODUCTION

This project aims to overcome the challenges in experiencing self-efficacy among individuals with severe physical and neurological impairments through the application of a novel therapeutic tool. The theoretical foundation for this work is provided from several distinct fields:

Neuroplasticity

The human brain consists of several hundred million nerve cells that form functional networks. These networks are not static, they undergo modifications in response to injury or learning processes, a phenomenon known as neuroplasticity. This continuous reorganization of the nervous system involves the formation of new synaptic connections. Modern neurological rehabilitation uses this inherent ability for therapeutic treatments. Even in cases of significant damage to the peripheral or central nervous system, intensive and targeted neurorehabilitation can increase cortical representation and improve function. This approach is typically more effective than compensatory plasticity of the uninjured hemisphere. [1, 2]

For instance, studies involving intensive, structured training of primate fingers have demonstrated that the corresponding central representations can be selectively enlarged. This activity-dependent neuroplasticity is associated with improvements in the execution of motor functions and sensory processing. [3, 4]

Psychological Distress

Following an acquired brain injury, individuals may develop comorbid psychiatric disorders. Among the most common are depressive disorders, anxiety disorders, stress and adjustment disorders, and addictions [5, 6, 7]. The risk of developing a mental health condition is highest during the first year after post-injury but remains elevated throughout the individual's life [8]. Gauggel shows that in the field of neuropsychological therapy , inventions and treatment goals are predominantly focused on function-related and behavioral approaches [9]. According to Klein and Guthke, brain damage can result in impaired awareness of the disorder, a deficit that severely restricts the ability for self-exploration, which is crucial for conversational psychotherapy. Consequently, it is often necessary to adapt therapeutic methods to the specific needs and characteristics of patients with acquired brain injury [10].

Music Therapy

Music plays an important role in modern healthcare. Music therapy is a psychotherapeutic intervention aimed at activating a patient's psychological and social resources to support their recovery process. This therapeutic approach involves active musical engagement through various activities, all conducted within the framework of a therapeutic relationship. Improvisation on musical instruments serves as a nonverbal medium for self-expression. It enables feelings and emotional issues that arise during therapeutic conversations to be experienced directly and processed more effectively [11]. However, for patients with significant mo-

Copyright: ©2026 Juliana Lüer et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License 3.0 Unported, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

tor impairments, the use of music therapy in early rehabilitation has been severely limited [12].

Haptic Feedback

Haptic feedback, essential for playing analog instruments, is key for human-machine interaction, enhancing immersion and engagement [13, 14]. It serves three primary functions: increasing realism in virtual environments [15], transmitting information [16] and evoking emotions [17]. These functions improve user engagement, user experience and the sense of embodiment.

In neurorehabilitation, haptic feedback provides tactile cues (e.g. vibrations, force feedback) to guide movement and link a therapeutic task to the user's sensory experience [18, 19]. Integrating haptic feedback with auditory information, particularly music, can create a rich, multisensory experience that may benefit therapeutic outcomes [20]. Vibrotactile feedback synchronized with music can heighten a user's sense of immersion and social connection [17], crucial for engagement in long-term rehabilitation [1, 21].

Patients, who are often unable to use traditional musical instruments or interfaces, could benefit immensely from the motivational advantages of music therapy. But in the current landscape of therapeutic tools is a lack of adaptable, customizable and accessible haptic systems designed for the specific needs of patients with severe motor impairments. This paper aims to address this gap by introducing a novel system that integrates high-resolution finger tracking with customizable auditory and vibrotactile feedback, designed specifically for a therapeutic context. The aspect of inducing targeted neurophysiological change forms an important foundation for the rehabilitative and therapeutic processes that the newly developed tool is intended to facilitate.

2. RELATED SOLUTIONS

Commercially available haptic devices have limited usability for our specific therapeutic application.

The design of novel interfaces for musical expression has a long history, with pioneering work by researchers like Machover [22] and Sonami [23] establishing the mapping of human gestures to musical control. While these systems demonstrated the potential of gestural control, they were often designed for large, expressive movements of performers.

Most critical, available systems today are typically designed for e.g. gaming to track larger, rapid movements of healthy individuals and lack the sensitivity required to accurately capture the **small-scale, slow movements** characteristic of patients with neurological impairments. Maggioni et al. stated systems like *MediaPipe* or *Leap Motion* as systems with approximately 2 to 3 mm spatial error [24]. The most accurate motion tracking systems are marker-based optoelectronic devices, which have a long setup and calibration time, high costs, and are bulky to transport between different hospital rooms [24]. Additionally, haptic feedback in existing models is often absent or improperly positioned, such as on the fingernail, which is not an optimal location for simulating the tactile sensation of playing an instrument and could be uncomfortable [25]. Another limitation is that many of these systems are designed as complete gloves, which restricts their use to hands of a specific size and shape; anatomical variations such as deformations or swelling can render the glove unusable [26]. Finally, for clinical use, these devices must be easily disinfected, and the common alternative of wearing a secondary glove often compromises usability and tactile sensitivity [27, 28, 29].

3. OUR HAPTIC FINGER TRACKER

Developed with a music therapist, the system is designed for single-finger operation to support patients with severe movement restrictions. A primary technical requirement was the ability to detect movements with a resolution below 1 mm. The integration of haptic feedback at the fingertip was an essential feature, considered fundamental for enhancing patient engagement and providing a tactile connection to the process of virtual tone generation. The key design criteria also included a mobile and lightweight construction (< 50 g), along with ergonomic features to ensure that the device did not impose additional restrictions on patient movement.

The system's interface utilizes the Open Sound Control (OSC) protocol, which facilitates straightforward a future integration with existing musical production and control tools like PlugData. Finger flexion and extension control the volume of a generated tone, while abduction and adduction in the vertical plane control the pitch. Horizontal abduction and adduction respectively trigger ascending and descending arpeggios.

Other requirements were considered for high accuracy, fast responsiveness, user comfort and wearability, robustness to noise, ease of integration, low cost, and low power consumption. These criteria were incorporated into a weighted decision matrix to evaluate sensor technologies for joint angle measurement (volume/pitch) and linear displacement (arpeggio). Following a comparison between flex sensors, strain gauges, inertial measurement units (IMU) and Leap Motion technology for joint angle measurement, the flex sensor was selected. For linear displacement sensing, the IMU was chosen.

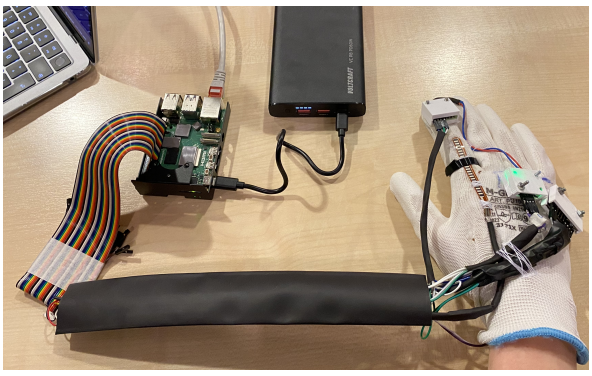


Figure 1. The Haptic Finger Tracker Device, showing the sensor glove, microcontroller, and connection to a computer.

For haptic feedback, a linear resonant actuator (LRA) was selected over an eccentric rotating mass (ERM) due to its superior control over distinct vibration patterns and the faster response time.

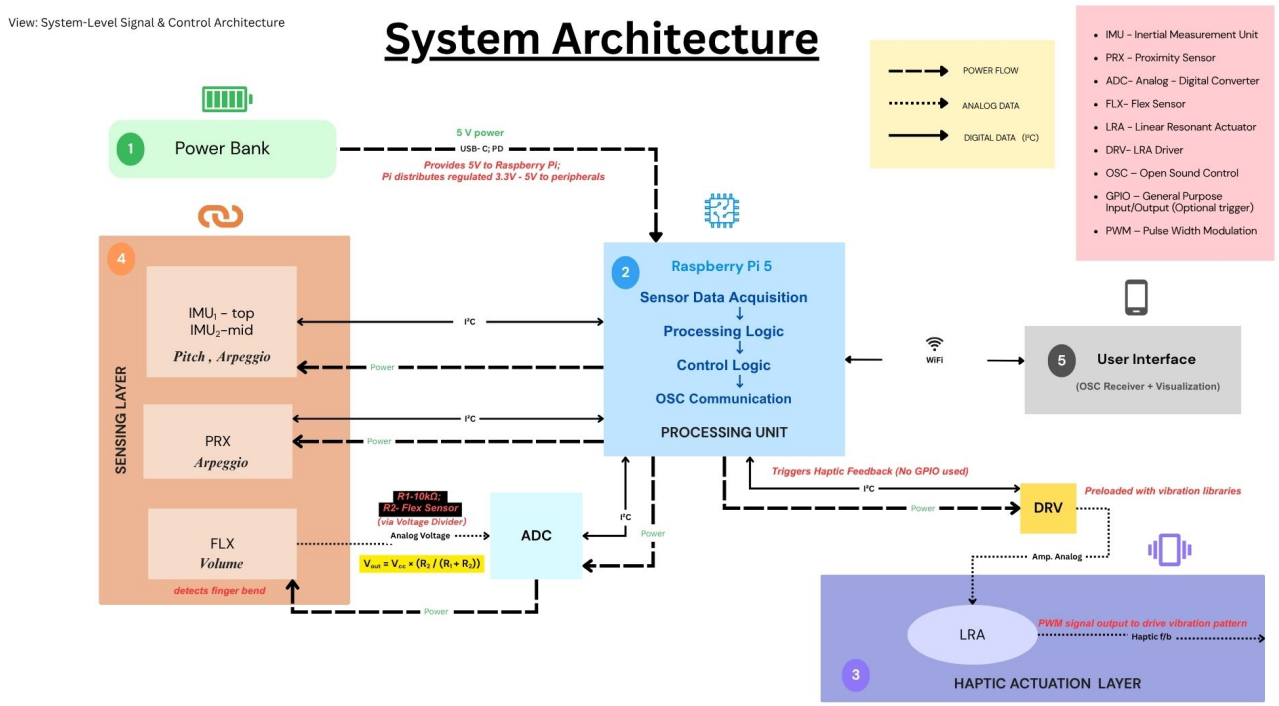


Figure 2. The system architecture of the Haptic Finger Tracker Device.

The system's architecture (Fig. 2) uses a Raspberry Pi 5 to manage data acquisition, processing and communication. It provides significant computational power for real-time processing while creating a path toward a fully embedded, standalone device in future iterations.

4. RESULTS AND DISCUSSION

The primary goal was to develop a system enabling patients to interact with a virtual instrument a music therapy context. The system was specially designed for use in the neurological rehabilitation, targeting patients with a severely restricted range of motion to facilitate their engagement in musical expression.

The current prototype underwent a preliminary qualitative evaluation of the collaborating music therapist who has experience in neurorehabilitation. The evaluation consisted of a session where the therapist used the device while employing verbal feedback. The primary goal was not to measure clinical outcomes, but to asses the usability of the core audio-haptic feedback loop and confirmed the prototype as a mobile, lightweight, and ergonomic solution. The assessment also identified opportunities for further improvement. Its design, which includes a disinfectable glove, promotes clinical hygiene and enables an inclusive and emotionally engaging experience for patients. Volume modulation was implemented using a flex sensor positioned on the ventral position of the index finger, while pitch control was achieved via a second flex sensor located on the dorsal side. Control over arpeggios was achieved using an inertial measurement unit (IMU). The haptic feedback was provided by a linear resonant actuator (LRA).

The system enables simultaneous control of auditory (OSC) output and haptic feedback via single-finger movement. This provides an intuitive and highly accessible interface for patients with significant motor impairments.

This work presents a haptic finger tracker that successfully enables musical expression from minimal gestures, providing a tool for self-efficacy in neurorehabilitation. While initial feedback confirms the prototype's ergonomic design, its current high latency and timing inconsistencies are key limitations. Future work will prioritize optimizing system response time, developing a more robust sensor fixture, and creating an mobile application for therapists to customize the audio-haptic feedback. These steps will move the prototype towards a therapeutic tool validated through user studies.

Acknowledgments

This research was funded under the Program "Innovative Hochschule" (innovative university) by the Federal Ministry of Research, Technology and Space (BMFTR) of Germany (Grant No. 13IHS232C) and the City of Hamburg

5. REFERENCES

[1] W. Müllbacher, "Neuroplastizität," in *Klinische Neuropsychologie*, J. Lehrner, G. Pusswald, E. Fertl, W. Strubreither, and I. Kryspin-Exner, Eds. Vienna: Springer Vienna, 2011, pp. 611–625. [Online]. Available: http://link.springer.com/10.1007/978-3-7091-0064-6_44

[2] L. Jäncke, *Lehrbuch Kognitive Neurowissenschaften*, 1st ed., ser. Lehrbuch Psychologie. Bern: Huber, 2013.

[3] M. M. Merzenich, R. J. Nelson, M. P. Stryker, M. S. Cynader, A. Schoppmann, and J. M. Zook,

"Somatosensory cortical map changes following digit amputation in adult monkeys," *Journal of Comparative Neurology*, vol. 224, no. 4, pp. 591–605, Apr. 1984. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1002/cne.902240408

[4] R. Nudo, G. Milliken, W. Jenkins, and M. Merzenich, "Use-dependent alterations of movement representations in primary motor cortex of adult squirrel monkeys," *The Journal of Neuroscience*, vol. 16, no. 2, pp. 785–807, jan 1996. [Online]. Available: https://www.jneurosci.org/lookup/doi/10.1523/JNEUROSCI.16-02-00785.1996

[5] T. A. Ashman, L. A. Spielman, M. R. Hibbard, J. M. Silver, T. Chandna, and W. A. Gordon, "Psychiatric challenges in the first 6 years after traumatic brain injury: cross-sequential analyses of axis I disorders," *Archives of Physical Medicine and Rehabilitation*, vol. 85, pp. 36–42, Apr. 2004. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0003999304000814

[6] L. De Wit, K. Putman, I. Baert, N. B. Lincoln, F. Angst, H. Beyens, K. Bogaerts, N. Brinkmann, L. Connell, E. Dejaeger, W. De Weerdt, W. Jenni, C. Kaske, A. Komárek, E. Lesaffre, M. Leys, F. Louckx, B. Schuback, W. Schupp, B. Smith, and H. Feys, "Anxiety and depression in the first six months after stroke. A longitudinal multicentre study," *Disability and Rehabilitation*, vol. 30, no. 24, pp. 1858–1866, Jan. 2008. [Online]. Available: http://www.tandfonline.com/doi/full/10.1080/09638280701708736

[7] R. Whelan-Goodinson, J. Ponsford, and M. Schönberger, "Validity of the Hospital Anxiety and Depression Scale to assess depression and anxiety following traumatic brain injury as compared with the Structured Clinical Interview for DSM-IV," *Journal of Affective Disorders*, vol. 114, no. 1-3, pp. 94–102, Apr. 2009. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0165032708002619

[8] S. Koponen, T. Taiminen, R. Portin, L. Himanen, H. Isoniemi, H. Heinonen, S. Hinkka, and O. Tenovuo, "Axis I and II Psychiatric Disorders After Traumatic Brain Injury: A 30-Year Follow-Up Study," *American Journal of Psychiatry*, vol. 159, no. 8, pp. 1315–1321, Aug. 2002. [Online]. Available: https://psychiatryonline.org/doi/10.1176/appi.ajp.159.8.1315

[9] S. Gauggel, "Grundlagen und Empirie der Neuropsychologischen Therapie: Neuropsychotherapie oder Hirnjogging?" *Zeitschrift für Neuropsychologie*, vol. 14, no. 4, pp. 217–246, Dec. 2003. [Online]. Available: https://econtent.hogrefe.com/doi/10.1024/1016-264X.14.4.217

[10] "Klein&Guthke," in *Update Neurorehabilitation 2022: Tagungsband zur Summer School Neurorehabilitation*, T. Platz, Ed. Bad Honnef: Hippocampus Verlag, 2022, vol. Klein&Guthke, pp. 227–244, meeting Name: Summer School Neurorehabilitation.

[11] D. Oberegelsbacher and T. Timmermann, "Definition," in *Lehrbuch Musiktherapie*, 3rd ed., ser. UTB Psychologie, Medizin, H.-H. Decker-Voigt, Ed. München: Ernst Reinhardt Verlag, 2020, no. 3068, pp. 18–20.

[12] Baumann, M., "Neurorehabilitation," in *Lexikon Musiktherapie*, H.-H. Decker-Voigt and E. Weymann, Eds. Hogrefe, 2021, pp. 451–454. [Online]. Available: https://elibrary.hogrefe.com/book/10.1026/02836-000

[13] T. A. Kern and C. Hatzfeld, "Motivation and Application of Haptic Systems," in *Engineering Haptic Devices*, T. A. Kern, C. Hatzfeld, and A. Abbasimoshaei, Eds. Cham: Springer International Publishing, 2023, pp. 3–33, series Title: Springer Series on Touch and Haptic Systems.

[14] M. Ziat, T. Rolison, A. Shirtz, D. Wilbern, and C. A. Balcer, "Enhancing virtual immersion through tactile feedback," in *Proceedings of the adjunct publication of the 27th annual ACM symposium on User interface software and technology - UIST'14 Adjunct*. Honolulu, Hawaii, USA: ACM Press, 2014, pp. 65–66. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2658779.2659116

[15] H. Culbertson and K. J. Kuchenbecker, "Importance of Matching Physical Friction, Hardness, and Texture in Creating Realistic Haptic Virtual Surfaces," *IEEE Transactions on Haptics*, vol. 10, no. 1, pp. 63–74, Jan. 2017. [Online]. Available: http://ieeexplore.ieee.org/document/7539397/

[16] T.-S. Dalsgaard and O. Schneider, "A Unified Model for Haptic Experience," *ACM Transactions on Computer-Human Interaction*, p. 3711842, Jan. 2025. [Online]. Available: https://dl.acm.org/doi/10.1145/3711842

[17] R. K. Venkatesan, D. Banakou, M. Slater, and M. M., "Haptic feedback in a virtual crowd scenario improves the emotional response," *Frontiers in Virtual Reality*, vol. 4, p. 1242587, Nov. 2023. [Online]. Available: https://www.frontiersin.org/articles/10.3389/frvir.2023.1242587/full

[18] L. Piggott, S. Wagner, and M. Ziat, "Haptic Neurorehabilitation and Virtual Reality for Upper Limb Paralysis: A Review," *Critical Reviews in Biomedical Engineering*, vol. 44, no. 1-02, pp. 1–32, 2016. [Online]. Available: http://www.dl.begellhouse.com/journals/4b27cbfc562e21b8,73da10e30f69cc45,6cfef8db3bc2f715.html

[19] S. Demain, C. D. Metcalf, G. V. Merrett, D. Zheng, and S. Cunningham, "A narrative review on haptic devices: relating the physiology and psychophysical properties of the hand to devices for rehabilitation in central nervous system disorders," *Disability and Rehabilitation: Assistive Technology*, vol. 8, no. 3, pp. 181–189, May 2013. [Online]. Available: http://www.tandfonline.com/doi/full/10.3109/17483107.2012.697532

[20] J. Zheng, Q. Ma, W. He, Y. Huang, P. Shi, S. Li, and H. Yu, "Cognitive and motor cortex activation during robot-assisted multi-sensory interactive motor rehabilitation training: An fNIRS based pilot study," *Frontiers in Human Neuroscience*, vol. 17, p. 1089276, Feb. 2023. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fnhum.2023.1089276/full

[21] M. Olsen, C. Nielsen, C. Emme, N. Godtfredsen, and H. Hansen, "Why We Keep Going: A Qualitative Longitudinal Study of the Motivation and Engagement Among Patients with COPD During a Long-Term Danish Tele-Rehabilitation Program," *Patient Preference and Adherence*, vol. Volume 19, pp. 3573–3587, nov 2025.

[22] T. Machover, "Interactive instruments for experts and amateurs," *The Journal of the Acoustical Society of America*, vol. 95, no. 5_Supplement, pp. 2888–2888, May 1994. [Online]. Available: https://pubs.aip.org/jasa/article/95/5_Supplement/2888/675772/Interactive-instruments-for-experts-and-amateurs

[23] L. Sonami, "NIME2014 Artistic Keynote Laetitia Sonami," Goldsmiths, University of London, Jul. 2014. [Online]. Available: https://www.youtube.com/watch?v=nEkufXDb-Ro

[24] V. Maggioni, C. Azevedo-Coste, S. Durand, and F. Bailly, "Optimisation and Comparison of Markerless and Marker-Based Motion Capture Methods for Hand and Finger Movement Analysis," *Sensors*, vol. 25, no. 4, p. 1079, Feb. 2025. [Online]. Available: https://www.mdpi.com/1424-8220/25/4/1079

[25] Advanced Realtime Tracking GmbH & Co. KG, "Datasheet Fingertracking2 (Tactile)," Jun. 2023. [Online]. Available: https://ar-tracking.com/system/files/2023-06/Technical_Datasheet_ART_Fingertracking2_tactile.pdf

[26] Quantum AI, "Datasheet Manus Prime 3 Haptic XR," 2023. [Online]. Available: https://imsys-vr.de/wp-content/uploads/Datasheet-prime-3-Haptic-XR-1.pdf

[27] Universitätsklinikum Hamburg-Eppendorf, "Zentrale Hygieneordnung, Teil I, Ziele, Organisation, Zuständigkeiten, Informationspflichten," Feb. 2023.

[28] Bundesministerium der Justiz und für Verbraucherschutz, "Gesetz zur Verhütung und Bekämpfung von Infektionskrankheiten beim Menschen - Infektionsschutzgesetz," Dec. 2023. [Online]. Available: https://www.gesetze-im-internet.de/ifsg/

[29] "Anforderungen an die Hygiene bei der Reinigung und Desinfektion von Flächen: Empfehlung der Kommission für Krankenhaushygiene und Infektionsprävention (KRINKO) beim Robert Koch-Institut," *Bundesgesundheitsblatt - Gesundheitsforschung - Gesundheitsschutz*, vol. 65, no. 10, pp. 1074–1115, Oct. 2022. [Online]. Available: https://link.springer.com/10.1007/s00103-022-03576-1

Reconfiguring Sonic Memory in Live Coding Performance

Riccardo Mazza
APM - Saluzzo
riccardo@riccardomazza.com

ABSTRACT

This paper presents a live coding performance system that reconfigures autobiographical sound materials through real-time interaction with a machine learning process. Rather than treating sonic memories as fixed archival objects, the system approaches memory as a dynamic and unstable process, continuously reshaped during performance. Recorded sound fragments are analyzed using FluCoMa descriptors and organized within a navigable two-dimensional space. A lightweight autoencoder is employed not as a high-fidelity generative model, but as a constrained transformation device that introduces controlled deviations, thereby altering the relationship to the source recordings. The resulting sounds are not reproductions of the originals, but transformed traces that require reinterpretation in real time. Within this framework, performance becomes a negotiation between intention, algorithmic transformation, and emergent sonic behavior. The performer does not retrieve memories, but actively reshapes them, generating new memory traces through interaction. The system adopts a human-in-the-loop approach, in which the model acts as a mediating structure rather than an autonomous agent. The contribution of this work lies not in technical novelty, but in proposing a practice-based perspective on how machine learning can function as a performative medium for memory transformation in live coding contexts.

1. INTRODUCTION

The notion of memory as a non-linear, stratified field draws from Henri Bergson's concept of *durée*—lived time as qualitative experience rather than measurable sequence [1]. In *Matière et mémoire*, Bergson describes memory not as a chronological archive but as coexisting planes of the past that are selectively actualized in the present [2]. This vision resonates with Deleuze and Guattari's rhizomatic model—a non-hierarchical, centerless structure of multiple connections and heterogeneous elements [3].

These philosophical frameworks inspire our performative and computational approach to sonic memory. Rather than organizing autobiographical sound fragments chronologically, our system creates a rhizomatic "memory map" by arranging recordings according to spectral similarities within a traversable two-dimensional latent space, setting aside temporal or narrative logic. In parallel, a lightweight autoencoder generates wavetables derived from the analyzed material. Rather than reconstructing or directly transforming the original audio, the autoencoder produces intermediate representations that function as new sonic material or control structures within the performance. These generated wavetables do not preserve the identity of the original recordings, but rearticulate their spectral characteristics in a reduced and unstable form, creating an indirect relationship between source material and resulting sound.

This approach reflects a broader movement in experimental music communities where machine learning tools are integrated into established creative coding environments. Alongside IRCAM's research on variational audio synthesis [4] and neural synthesis models such as RAVE [5], similar tendencies can be observed in tools like FluCoMa [6] for SuperCollider and Max, MuBu [7] for gesture-sound mapping and corpus-based synthesis, and community-developed instruments exploring data-driven performance. These developments share a common philosophy: AI as a mediating structure embedded within real-time, improvisational workflows, rather than as an autonomous generative agent [8].

Within this process, the performer functions not as an archivist retrieving fixed objects, but as an active agent whose decisions continuously redirect the trajectory of sonic transformation. Rather than reproducing or stabilizing past material, the interaction generates new memory traces that exist only within the temporal unfolding of the performance itself. Central to this design is a human-in-the-loop approach, in which machine learning operates as a mediating structure that destabilizes and continuously reconfigures the relationship between past sonic traces and present musical action.

2. SYSTEM ARCHITECTURE AND WORKFLOW

The system prioritizes performative immediacy over analytical precision. It is implemented using the FluCoMa (Fluid Corpus Manipulation) toolkit [6][9], which provides real-time machine learning and audio analysis objects designed specifically for creative applications. Rather than constituting a fixed processing pipeline, the system operates as a set of interacting layers that support the continuous reconfiguration of sonic material during performance.

The architecture is structured around three interconnected components: a descriptor-based mapping of recorded material, a machine learning layer for the generation of derived representations, and a synthesis stage that translates these representations into performable sound. All components are optimized for real-time interaction, allowing the performer to engage with the system as the performance unfolds.

Copyright: ©2026 First Author et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License 3.0 Unported, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

2.1 Real-time Analysis Pipeline

The workflow consists of three parallel processes that execute upon loading a new audio memory:

1. **Segmentation and 2D Mapping**
    - `FluidBufNoveltySlice` segments the audio using selectable algorithms (Spectrum, MFCC, Chroma, Pitch, or Loudness), allowing performative exploration of different segmentation strategies
    - `FluidBufSpectralShape` and `FluidBufLoudness` extract two-dimensional features from each slice
    - `FluidKDTree` organizes slices in a navigable 2D space, accessible via mouse, MIDI controllers, or live coding patterns
    - A `SynthDef` enables real-time manipulation (rate, filtering, envelopes) of triggered slices
2. **Autoencoder Layer**
    - MFCC analysis (13 coefficients) generates the spectral dataset
    - `FluidMLPRegressor` implements an encoder/decoder architecture (13→8→3→8→13)
    - Range normalization calculates min/max values for consistent mapping
    - Training proceeds asynchronously, allowing the performer to navigate the 2D space in the interim
3. **Derived Sonic Material**
    - Upon point selection, the decoder generates 13 MFCC values in real time
    - Spectral centroid extraction derives a fundamental frequency
    - Wavetable synthesis produces 512–2048 sample buffers via additive processing
    - The resulting wavetable is used as a dynamic sonic resource or control structure within the performance

This parallel architecture ensures no interruption in performative flow—the performer can begin exploring immediately while the system progressively enriches the sonic possibilities as the autoencoder completes its training. Crucially, the autoencoder layer is not conceived as a fixed generative model, but as an open transformation framework that can mediate, reshape, or extend the sonic material in different ways depending on the performative context.

3. PERFORMANCE INTERFACE AND CONTROL

The system implements a dual-layer performance interface that combines gestural control with algorithmic manipulation, accompanied by real-time visual feedback of the navigated memory space.

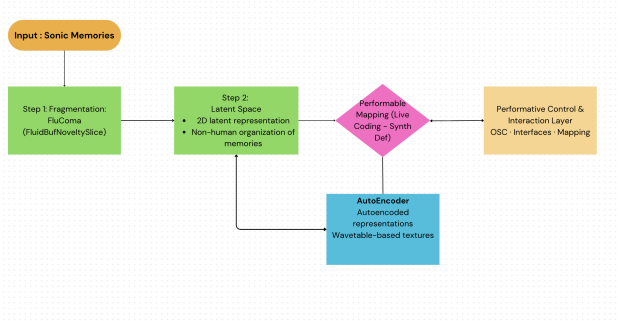


Figure 1. System workflow diagram showing three parallel processes: (1) Real-time segmentation and 2D spatial mapping for immediate performative interaction, (2) Asynchronous autoencoder training on MFCC features, and (3) On-demand wavetable synthesis from decoder output. The architecture enables continuous performance while progressively enriching sonic possibilities.

3.1 Dual Performance Layers

Live Coding Layer: SuperCollider patterns and routines enable algorithmic exploration of the navigable sonic space, allowing the performer to define temporal sequences, probabilistic selections, and parametric variations. This layer provides precise control over timing and structure while maintaining improvisational flexibility. The system employs global variables (prefixed with ˜) rather than JITLib proxies, prioritizing robustness and reliability for state sharing. Routines construct real-time patterns for navigating the 2D space:

```
(
x.stop;
x = Routine {
    inf.do { arg i;
        ~point.(~last);
        ~rate = rrand(0.25, 2.5);
        ~slice.wait;
    }
}.play;
)
~sliceThres = 0.35;
~minSlice = 20;       // in hops
~sliceAlgo = 2;  // 0.Spectrum, 2. MFCC etc.
~centroidRate = 0.25;
//Wavetable Synth Controls
~a_Ratio = 0.25/2.0; // sub-octave
~b_Ratio = 1.0;      // fundamental
~c_Ratio = 1.5;      // fifth
```

The code example above illustrates one possible approach among many—during performance, such routines are written, modified, and combined in real time, contributing to the continuous reconfiguration of the sonic material. This live coding practice enables rapid exploration of different sonic behaviors: adjusting segmentation parameters (`~sliceThres`, `~sliceAlgo`), transforming synthesis ratios, or constructing complex temporal patterns. Each performance generates unique code structures as the performer responds to the emerging sonic landscape, tracing evolving trajectories through continuous code intervention.

Gestural Control Layer: TouchOSC interfaces or MIDI controllers map continuous data to system parameters, en-

abling immediate physical engagement with the material. The 2D latent space visualization (`FluidPlotter`) responds to MIDI CC messages (X/Y coordinates), allowing tactile navigation through the memory space while observing the visual representation of selected points. This layer complements the live coding process by introducing continuous, embodied interaction, enabling the performer to negotiate between discrete algorithmic structures and fluid sonic transformation.

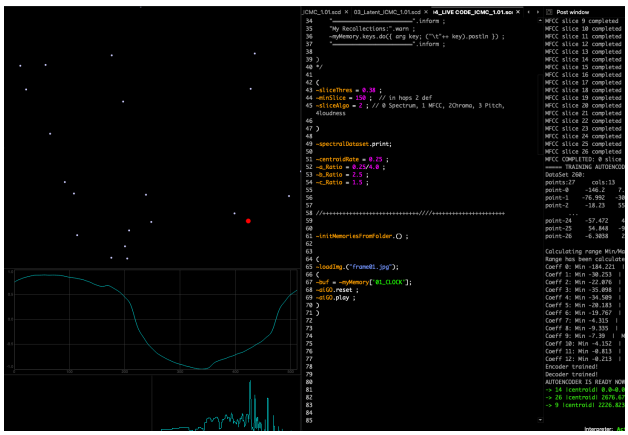


Figure 2. Performance interface showing the navigable 2D latent space (left), real-time wavetable visualization (right), and FFT analysis of the processed output (top). The visual feedback enables immediate correlation between spatial navigation and spectral transformation.

3.2 Real-time Synthesis Control

The autoencoder layer provides a set of derived representations from each selected point in the latent space, which can be used in different ways within the synthesis process. In the current implementation, this process generates a wavetable that the performer distributes across multiple synthesis layers through SuperCollider `SynthDef` instances. More generally, however, the autoencoder is not limited to wavetable generation, but can act as a transformation layer capable of reshaping, modulating, or extending the sonic output during performance.

This creates a recursive system where synthesis parameters are informed by the analysis itself—fundamental frequencies derived from spectral centroids, amplitude envelopes shaped by loudness detection, and timbral variations drawn from the MFCC analysis that feeds the autoencoder. Rather than establishing a fixed signal chain, the system sustains a continuous interplay between analysis, transformation, and synthesis.

The current implementation illustrates this malleability through three harmonic voices whose frequency ratios are treated as real-time variables (shown here in their initial configuration):

- **Fundamental** (f): Base frequency derived from spectral centroid
- **Secondary Layer** (ratio variable, default $f/2$): Lower harmonic for depth
- **Tertiary Layer** (ratio variable, default $f \times 1.5$): Upper harmonic for brightness

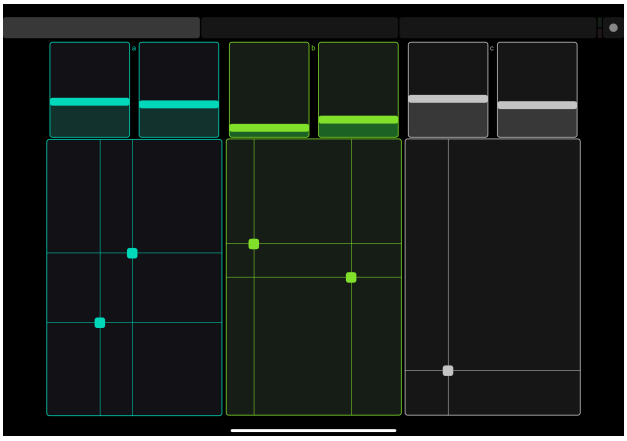


Figure 3. TouchOSC interface for real-time wavetable performance. Three faders control harmonic layers, while additional controllers modulate pitch response, amplitude envelopes, and filter parameters. The interface enables gestural control complementary to live coding interventions.

Each instantiated voice receives individual amplitude control via the TouchOSC interface. Notably, parameters such as filtering, pitch deviation, and temporal manipulation (playback rate) are exposed as continuous variables rather than fixed constants. While currently mapped to specific ranges (e.g., rate $0.25\times$–$2\times$), these boundaries are defined in software and can be dynamically rescaled or modulated by other synthesis parameters.

The performer can redefine these relationships during performance, treating the autoencoder's output not as a stable sound object but as a flexible and evolving resource. In this sense, the synthesis stage does not simply render the result of a transformation, but becomes part of a broader process in which analysis, machine learning, and performative action continuously inform one another. This dual control paradigm allows the performer to establish slow gestural arcs via controllers while executing rapid structural variations through code, creating a dialogue between embodied and algorithmic performance strategies.

4. CONCLUSIONS AND FUTURE WORK

This project is not presented as a standalone software toolkit, but as an exploratory *workflow*—a practice-based approach that uses existing frameworks to investigate AI as an intermediary layer for memory transformation. In this context, the distortion inherent in neural compression becomes a creative feature rather than a limitation, introducing instability and deviation as compositional resources.

Early performances suggest that the system's constraints (reduced dimensionality, the indirect relationship between source and output, and the interplay between gestural and coded control) expand rather than limit creative possibilities.

Future developments will explore the integration of additional performative layers—visual, electroacoustic, and extended sonic processing—whose parameters are derived from the latent space or the autoencoder output, expanding the framework beyond audio-only contexts while preserving its core logic of continuous, analysis-driven reconfiguration.

Ultimately, the goal is not to stabilize or finalize the system, but to continue exploring how computational processes

can support new forms of musical thinking, in which memory is understood not as a fixed archive but as a transforming field shaped by performance, perception, and technological mediation.

4.1 Video Demonstration

A short video demonstration illustrating the performative workflow is available at: `https://vimeo.com/1176246964/1a7dca7d3a?fl=pl&fe=cm`

5. REFERENCES

[1] H. Bergson, *Essai sur les données immédiates de la conscience*. Paris: Félix Alcan, 1889.

[2] ——, *Matière et mémoire. Essai sur la relation du corps à l'esprit*. Paris: Félix Alcan, 1896.

[3] G. Deleuze and F. Guattari, *Mille plateaux: Capitalisme et schizophrénie 2*. Paris: Les Éditions de Minuit, 1980.

[4] P. Esling, A. Chemla-Romeu-Santos, and A. Bitton, "Generative Timbre Spaces with Variational Audio Synthesis," in *Proceedings of the 21st International Conference on Digital Audio Effects (DAFx-18)*, 2018, pp. 411–418.

[5] A. Caillon and P. Esling, "RAVE: A variational autoencoder for fast and high-quality neural audio synthesis," *arXiv preprint arXiv:2111.05011*, 2021.

[6] P. A. Tremblay, G. Roma, and O. Green, "Enabling Programmatic Data Mining as Musicking: The Fluid Corpus Manipulation Project," in *Computer Music Journal*, vol. 45, no. 2. MIT Press, 2021, pp. 9–23.

[7] F. Bevilacqua, N. Schnell, D. Schwarz *et al.*, "MuBu: A Max Toolbox for Multimodal Analysis of Sound and Motion, Sound Synthesis and Interactive Machine Learning," https://ircam-ismm.github.io/max-msp/mubu.html, 2010, iRCAM – Sound Music Movement Interaction Group.

[8] A. Chemla-Romeu-Santos, S. Ntalampiras, and P. Esling, "Latent Diffusion Models for Generative Music: A Survey," *arXiv preprint arXiv:2308.09878*, 2023.

[9] O. Green, P. A. Tremblay, and G. Roma, "FluCoMa: A Framework for Music Audio Analysis in Creative Coding," in *Proceedings of the International Computer Music Conference*. ICMA, 2022, pp. 265–270.

Robotics for Digital Artists: OSC-ROS Integration

Kieran McAuliffe
Hochschule für Musik und Theater Hamburg
brian-kieran.mcauliffe@hfmt-hamburg.de

Ornella Tortorici
Technische Universität Hamburg
ornella.tortorici@tuhh.de

Ali Elnwegy
Technische Universität Hamburg
ali.el-nwegy@tuhh.de

ABSTRACT

The Robot Operating System (ROS) has become a de facto standard for robot software development, offering powerful tools for real-time communication, control, and simulation. However, its complexity presents significant barriers for creative practitioners. In contrast, the accessible Open Sound Control (OSC) is widely adopted in the creative coding community and supported by numerous artistic software environments. This paper showcases the work in progress OSC-ROS tool designed to lower the entry barrier for artists working with robotic systems by enabling bidirectional communication. The tool can be used in both directions: either receving OSC messages from the user and converting them into joint trajectories sent over ROS to a robot controller or receiving information from a robot via ROS messages and converting them to OSC messages. The tool also provides corresponding logging information. Two interactive application setups built with OSC-ROS are displayed here: a custom-built painting robot responding to sounds and a sonified industrial robot arm. These applications highlight how robotic systems can function both as expressive actuators and as performative interfaces.

1. INTRODUCTION

1.1 Robotics and Performing Arts

Automated musical systems have a long history in performance practice, dating back to the player pianos of the nineteenth century [1]. Contemporary research has continued this tradition, with robotic instrumentalists capable of participating in interactive contexts. Recent work has explored robotic musicians as interactive sound installations [2], as collaborators in human–robot musical improvisation [3], and as coordinated members of large robotic musical ensembles [4]. Beyond instrumental performance, robotic systems have also been integrated into broader performing arts practices, including systems that generate paintings in response to live musical input [5] and robots that participate in choreographed dance performances alongside human performers [6]. These works illustrate how robotics is increasingly used beyond automating musical execution, in multi-modal performances and human–machine collaborations. However, these collaborations bring together two distinct technical worlds, each with its own standard protocol: the creative arts community, which widely employs Open Sound Control (OSC), and the field of robotics, which has converged on the Robot Operating System (ROS).

Copyright: ©2026 Kieran McAuliffe et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License 3.0 Unported, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

1.2 ROS

ROS is a powerful set of libraries, tools and protocols for robot software development [7]. It has been accepted as the standard by both academia and industry, and is supported by most commerical products.

One service ROS provides is low-latency, real time communication between nodes using either UDP or TCP connection. This can be used for a variety of purposes, including controlling and logging movements. Another valuable service is a generalized system of control. A URDF file (Unified Robot Description Format) contains all information required to create a digital model of the robot, including joints, links, collisions and appearance. ROS was designed for robotics experts, and as such is difficult to use for multimedia artists. Specifically, it only runs on a specific Ubuntu distribution, must be programmed in Python or C++, and requires users to understand its complex communication scheme.

1.3 OSC

OSC [8] is a protocol for sending XML-like formatted strings via UDP, which has been widely adopted by the creative coding community for its simplicity and ease of use. OSC was originally intended for sound purposes, but has been adopted by the larger creative coding community for its simplicity and ease of use. It is supported by many artistic softwares such as DAWs, MaxMSP, TouchDesigner and more. Libraries are available for development in all popular programming languages.

1.4 Existing OSC ROS package

There exists a rososc package [9]. However, it is a lower level bridge designed for experts familiar with ROS and robotics. Additionally, it works for ROS1 but ROS2 is not supported, rendering it obsolete.

2. OSC-ROS TOOL

The OSC-ROS tool itself is a translation layer between OSC messages and ROS messages in the context of robotics (Figure 1). This constraint was chosen due to the main usage of ROS in the field of robotics. The tool generates a set of OSC message converters based on the structure of the robot, e.g. a six degrees of freedom robot will have six joints which can be controlled and their states reported to OSC.



Figure 1: Diagram showing the internals of the tool and how it translates packages from ROS to OSC and Vice Versa.

Address	Data Format	Description
`/joint_positions`	`[q1, ..., qN]` or `[q1, ..., qN, duration]`	Move all joints
`/joint_position/{joint}`	`q1` or `[q1, duration]`	Set one joint only
`/tcp_coordinates`	`[x, y, z, roll, pitch, yaw]` or with duration	Move to Cartesian pose
`/joint_trajectory`	`[[q1,...], [q2,...], ...]`	Follow a joint-space trajectory
`/cartesian_trajectory`	`[[x1,...], [x2,...], ...]`	Follow a Cartesian-space trajectory
`/speed_scaling`	`float` (e.g., 0.5 -> 50% of max speed)	Adjust execution speed

Figure 2: Available OSC commands for OSC-ROS.

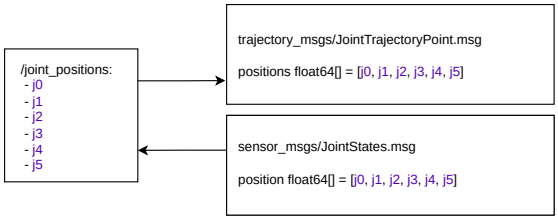


Figure 3: Diagram showing how messages are translated from ROS to OSC under the context of robotics

The primary challenge in translating messages between OSC and ROS stems from the inherent flexibility of each protocol. Both platforms are designed to be versatile, allowing developers to define arbitrary message structures. This versatility precludes a universal standard for message formats, making a general-purpose, one-to-one translation between any OSC message and any ROS message intractable. Our approach resolves this by constraining the scope of the translation. By establishing a predefined set of supported message formats, the problem is simplified to a well-defined mapping between two known and bounded structures.

For robotic control, our tool implements a specific message mapping. For a 6-DOF robot, as an example, a list of six floating-point numbers sent to a designated OSC address is translated into a ROS message containing a six-variable array, published to a corresponding topic, and vice versa, as illustrated in Figure 3. This structure is designed to align with standard ROS trajectory control, which typically requires both a target point and the duration to reach it. To simplify this for the end-user, our tool automatically calculates the required duration based on a maximum achievable joint velocity. This velocity can be dynamically adjusted by the user via the *"/speed_scaling"* OSC address (see Figure 2), abstracting away the need to provide a duration with each command. However, the option to manually specify a duration is retained for advanced use cases.

Providing Joint control using this interface is sufficient but does not fully capture the capabilities of the robot. Cartesian control is the most used robot control method for everyday tasks. The OSC-ROS interface also provides Cartesian control capabilities by means of Inverse Kinematics. The interface uses the Rigid Body Tree of the provided robot to calculate the inverse kinematics for cartesian coordinates. When an OSC message with cartesian coordinates is received, the inverse kinematics problem for the cartesian coordinates are calculated and sent to the ROS side as joint coordinates, eliminating the need of a cartesian controller to be provided by the robot driver. This is sometimes needed as some robot drivers do not provide any cartesian controller interface to their robots.

To configure the interface for a specific robot, the tool must know its physical structure, particularly its number of joints and kinematic chain. This information is typically standardized in an URDF file provided by the manufacturer (see Figure 4 for an example of URDF file). This file contains the links of a robot, how they are connected, their inertia, their dimensions, how to define their collision perimeters, and sometimes a link to their 3D CAD files. By parsing this file, our tool automatically extracts the robot's rigid body tree, determines the number of joints, and configures the inverse kinematics (IK) solver. This automated configuration not only facilitates the message translation between OSC and ROS but also adds high-level functionality, such as Cartesian control, which may not be natively available on the robot's controller.

3. CASE STUDIES

To evaluate the practical application and creative potential of our tool, we developed two case studies through artist-

```
// demo_bot.urdf
<?xml version="1.0"?>
<robot name="demo_bot">
<link name="base_link">
 <visual>
  <geometry>
   <sphere radius="0.2"/>
  </geometry>
 </visual>
 <collision>
  <geometry>
   <sphere radius="0.2"/>
  </geometry>
 </collision>
 <inertial>
  <mass value="1.0"/>
  <inertia ixx="0.1" ixy="0" ixz="0" iyy="0.1" iyz="0" izz="0.1"/>
 </inertial>
</link>
<joint name="hinge" type="revolute">
 <parent link="base_link"/>
 <child link="tool_link"/>
 <limit effort="1.0" lower="-1.5" upper="1.5" velocity="1.0"/>
</joint>
<link name="tool_link"/>
</robot>
```

Figure 4: Example of a URDF file.

engineer collaborations. These applications highlight distinct communication schemes, demonstrating how the tool can be used to control an expressive painting robot and to sonify the movements of an industrial robot arm.

3.1 Case study: OSC to ROS

In this communication scheme, an end user application generates OSC messages which the OSC-ROS tool uses to control a robot.

3.1.1 Painting Robot

The painting robot was developed at the Ligeti Center as a tool to explore the multimodality of arts with the help of robotics. Its initial version was capable of automatically translating musical parameters into paintings through an integrated algorithm, not easily accessible. The version presented in this paper can be controlled by artists, opening up a range of applications. The robot, as seen in Figure 5, is made up of 11 different motors that independently control both the brush displacement and the paint distribution on the canvas. It is composed of an arm with a shoulder and an elbow joint, which control the brush in a horizontal plane. The heights of the brush is also controlled in order to adjust its pressure. The orientation of the brush is controlled to follow the arm movement but can also be adjusted by the user via OSC. The color comes from a storage of acrylic paint in separated syringes for the 3 primary colors as long as black white and water. The paint is pushed out by stepper motors and travels in tubes along the arm to reach the paint dispenser, which mixes the paints together and drops it on the canvas before the brush goes over it. Finally, there is an area on the side where the brush can be cleaned in water in a rotating cup.

3.1.2 Workflow

The initial implementation of the painting robot features a PlugData interface for real-time control (see Figure 6).

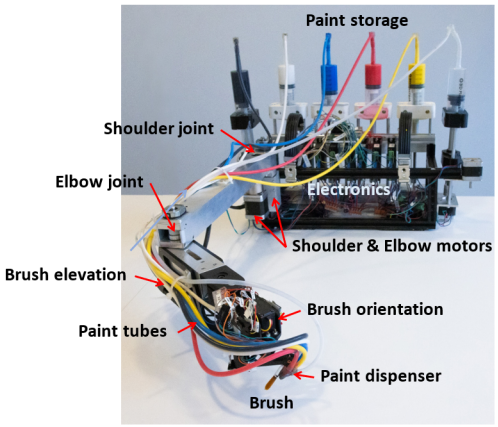


Figure 5: Overview of the painting robot that was interfaced for a control via OSC to ROS.

This allows a user to directly manipulate the Cartesian position of the robot's end-effector. The OSC-ROS tool receives these OSC commands, calculates the corresponding joint positions via inverse kinematics, and transmits them to the robot as ROS messages. Future development will introduce a more granular level of control, enabling the direct manipulation of individual joints and the separate control of paint dispensing for each color. This enhanced interface will also facilitate the mapping of external OSC data sources, such as real-time sound analysis, to these control inputs, providing artists with an intuitive pathway to create complex sound-to-painting interactions.

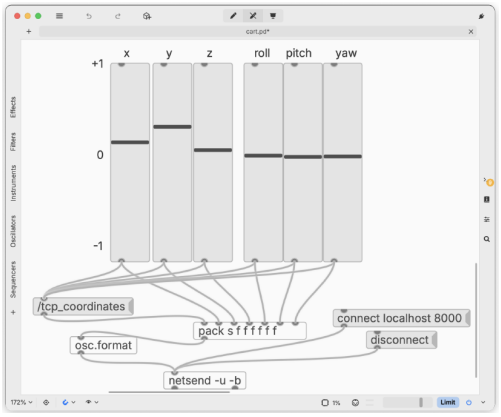


Figure 6: Example use in PlugData. Sliders are inputs for the OSC message. Each slider controls one Cartesian position and orientation. Sliders can be replaced with other output devices already implemented in PlugData.

3.2 Case study: ROS to OSC

The second use case demonstrates the reverse dataflow, where the robot functions as an input device for a user application. This is facilitated by the "Freedrive" mode available on some collaborative robots, which compensates for gravity while allowing a user to manipulate the arm by hand. In this configuration, the OSC-ROS tool subscribes to the robot's joint state topics in ROS, acquiring real-time position, velocity, and rotation data. It then broadcasts this data as OSC messages, allowing a variety of creative po-

tentials, such as a performer using a robot as a musical interface.



Figure 7: Universal Robotics UR10e

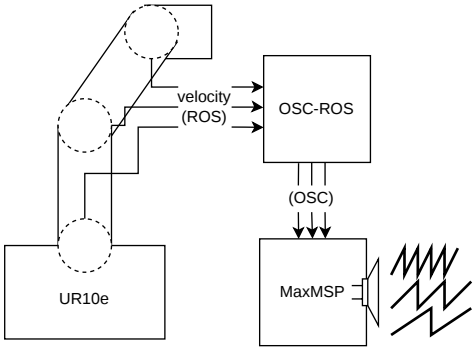


Figure 8: Backwards communication scheme with UR10e

3.2.1 UR10e

The Universal Robotics UR10e, shown in Figure 7, is a medium sized industrial arm with 6 revolute joints that can reach up to 1.5 meters. It can be put in the aforementioned Freedrive mode via the press of a button, or the connection of a circuit.

3.2.2 Workflow

We run the UR10e ROS2 driver and the OSC-ROS tool in a Docker Container on a dedicated machine. On the same local network, another machine runs MaxMSP which receives OSC messages from the ROS2 machine to sonify the movement of the robot. Each joint is treated like a bowed string and assigned a saw wave which responds to the joint velocity. As a performer rotates a joint by hand, this increasing velocity causes a saw wave to increase in amplitude while a low pass filter is released. Pulling the joint especially hard causes the pitch of the saw wave to bend upwards. Moving multiple joints of the arm simultaneously, such as when manipulating the end point, thus can create chords. Figure 8 shows this workflow.

4. CONCLUSION

In this paper, we presented the OSC-ROS tool, a bidirectional communication bridge designed to lower the technical barrier for artists and creative practitioners working with robotics. By establishing a constrained but flexible messaging schema, the tool facilitates the control of robotic systems from common artistic software environments and enables the sonification of robotic motion. The two case studies — a painting robot and a sonified industrial arm — demonstrate its practical application in creating expressive, interactive art. The open-source tool, along with all supporting documentation, is available at [10].

4.1 Limitations

Testing revealed two primary limitations. Firstly, sending a continuous stream of Cartesian position commands can result in discontinuous or ”jerky” motion, as the robot’s controller constantly cancels and recalculates its path. Secondly, users observed that the end-effector does not always travel in a straight line between points. This is an inherent characteristic of inverse kinematics solutions for articulated robots, where joint-space interpolation results in curved paths in Cartesian space. Both issues highlight the need for user education on robotic motion principles and could be partially mitigated by improved trajectory planning within the tool itself.

4.2 Future Work

Future development will focus on expanding the tool’s expressive capabilities. We plan to implement direct velocity and acceleration control over the joints, which will allow for more nuanced and dynamic motion profiles beyond simple position commands. Another priority is to add a built-in mechanism for recording, editing, and replaying trajectories. This feature would enable artists to choreograph complex sequences offline and integrate them into live performances without relying on external sequencers. These enhancements will further bridge the gap between artistic intent and robotic execution.

Acknowledgments

The project OSC-ROS was funded as part of the ligeti center under the program ”Innovative Hochschule” (innovative university) by the German Federal Ministry of Research, Technology and Space (BMFTR) and the City of Hamburg (Grant No. 13IHS232A).

5. REFERENCES

[1] A. Kapur, “A History of Robotic Musical Instruments,” in *Proceedings of the International Computer Music Conference (ICMC)*, 2005. [Online]. Available: https://www.mistic.ece.uvic.ca/publications/2005_icmc_robot.pdf

[2] K. Wu *et al.*, “Robotic Musicians as Interactive Sound Installations,” *arXiv preprint*, 2024. [Online]. Available: https://arxiv.org/abs/2407.16956

[3] G. Weinberg and S. Driscoll, “A Robotic Percussionist for Human-Robot Musical Collaboration,” in *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*, 2009. [Online]. Available: https://www.nime.org/proceedings/2009/nime2009_070.pdf

[4] G. Weinberg and G. Hoffman, “Robotic Musicianship in Large Musical Ensembles,” in *IEEE International Conference on Robotics and Automation Workshops*, 2008. [Online]. Available: https://ieeexplore.ieee.org/document/4686665

[5] R. Savery, A. Savery, and J. Baird, “Robotic Arm Generative Painting Through Real-time Analysis of Music Performance,” in *Proceedings of the ACM/IEEE International Conference on Human-Agent Interaction (HAI)*, 2022. [Online]. Available: https://tesseract.art/hai-2022

[6] E. Ackerman, “Building Human-Robot Relationships Through Music and Dance,” IEEE Spectrum, 2021. [Online]. Available: https://spectrum.ieee.org/robot-dance-music

[7] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, “Robot Operating System 2: Design, Architecture, and Uses in the Wild,” *Science Robotics*, vol. 7, no. 66, 2022.

[8] M. Wright and A. Freed, “Open Sound Control: A New Protocol for Communicating with Sound Synthesizers,” in *Proceedings of the International Computer Music Conference (ICMC)*, Thessaloniki, Greece, 1997, pp. 101–104.

[9] “ROSOSC: ROS OSC Bridge and Utilities,” https://wiki.ros.org/rososc, Open Source Robotics Community, 2013, rOS wiki page for the rososc stack, a collection of ROS nodes and utilities for interacting with the Open Sound Control (OSC) protocol. Accessed: 2025-12-17.

[10] K. McAuliffe, A. Elnwegy, and A. Schaefer. Open access repository of the OSC-ROS python library. [Online]. Available: https://collaborating.tuhh.de/m-4/imek-public/python-libs/ros2osc2ros_python-library

B3-H4RSH: A Noise-based Multiplayer Game for Mobile Music-Making

Luca Morino
Conservatorio "F. Bonporti" of Trento
luca.morino@conservatorio.tn.it

Nicola Conci
University of Trento
nicola.conci@unitn.it

Fabio Cifariello Ciardi
Conservatorio "F. Bonporti" of Trento
fabio.cifariellociardi@conservatorio.tn.it

ABSTRACT

Over the past two decades, artists and composers have increasingly explored mobile phones – ubiquitous and accessible devices – as instruments for musical performance and, in particular, as interfaces for audience participation and collaborative music-making. This paper presents B3-H4RSH, an interactive mobile music system. Implemented as a web application for smartphone browsers on a co-located network, the system interconnects participants' devices, employing competitive multiplayer mechanics to structure interdependencies among players and shape the music-making act within a noise-music paradigm. By influencing and responding to one another's actions, participants collectively diffuse sound throughout the space from their smartphones while competing to achieve the "harshest" sonic outcome – and win.

1. INTRODUCTION

In recent times, the proliferation of web technologies and the evolution of modern smartphones have positioned the mobile phone as a ubiquitous and portable computing platform for artistic and research practices. The use of spectators' personal devices in performances has motivated a variety of projects, from audience-participation practices to collaborative music-making. Within this context, mobile musical systems span from instrument-oriented to control-interface applications that offer participants varying degrees of agency, from influencing aspects of a performance to effectively co-producing (or even constituting) the performance itself. Such systems are often designed with the explicit goal of maximising audience engagement. In this regard, games – understood as rule-based systems with variable, quantifiable, and valued outcomes – offer a productive framework for structuring participative form.

To explore these possibilities, we introduce *B3-H4RSH*, an interactive mobile music system. As part of a broader research project, *B3-H4RSH* is conceived as a probe to investigate players' interpersonal dynamics that emerge in a sound-mediated, ludic, and interactive context. By adopting competitive multiplayer gameplay to shape musical evolution, the project builds on the idea of inviting participants to *play* with noise, in both the musical and ludic senses of the term.

Copyright: ©2026 Luca Morino et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License 3.0 Unported, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Noise music has been described as an oppositional practice, a critical lens on musical institutions, legacy, and common practices [1]. Moreover, from Futurist invocations of battle to contemporary engagements with sonic warfare and militarised listening technologies, noise aesthetics are historically entangled with conflict and violence, an ambivalent pleasure through which aggressive sound may repel and attract simultaneously [2].

2. BACKGROUND

2.1 Participatory Music and Mobiles

Mobile phones have been widely used as networked interfaces in participatory performance contexts. An early example is *Dialtones (a Telesymphony)* by Golan Levin, commissioned by Ars Electronica in 2001 [3], which created a spatially distributed musical experience using ringtones through the audience's phones. From the mid-2000s onward, mobile music-making expanded by exploring the potential of mobile devices for distributed computation [4, 5] and for supporting participation – for instance, by granting audiences control over performers [6] or explicitly engaging them in interactive play [7].

Recently, Matuszewski *et al.* [8] have reframed co-located, web-based practices as *Situated Networked Music Systems* (SNMS). From a compositional perspective, Xambó and Roma [9] propose thirteen compositional dimensions to support the analysis and design of audience participation performances through mobile. Hödl and Turchet [10] further contribute a TMAP Framework for designing and evaluating *Technology-Mediated Audience Participation* performances. Beyond these frameworks, mobile-based projects have been used to examine the social dimensions of interconnected systems [11, 12], collaborative and playful interactions [13, 14], and competitive, game-oriented audience participation [15].

2.2 Games in Music-Making

Historically, composers have explored the integration of game elements into musical works. Iannis Xenakis, for example, incorporated game-like rules in *Duel* (1959), a piece for two orchestras in which the conductors "compete" by selecting alternative score sequences in response to the other ensemble's actions [16]. John Zorn's *Cobra* (1984), for instance, uses cards and hand signals to cue performers' actions – such as forming and dissolving subgroups or exchanging roles and leadership [17]. Other works have explored *sonification* strategies that translate gameplay, particularly in pre-digital games such as *Chess* or *Go*, into musical form [18, 19].

Recently, composers embedded game mechanics and gamification [1] strategies into a single-player, video-game driven electroacoustic improvisation [20], interactive screen-scores [21], or collaborative networked music performances [22]. Studley's recent work [23] explores music creativity in a single-player competitive gameplay, arguing that the two are not mutually exclusive and proposing the development of competitive games for creating music. Although game-based interactive systems have recently attracted increasing research interest, the use of explicit, noise-oriented competitive multiplayer game mechanics to structure interaction in mobile music systems remains underexplored.

3. B3-H4RSH

In this section, we present *B3-H4RSH*, a prototype interactive mobile music system. Its audio engine, based on a digital no-input feedback loop and overdrive distortion, locally generates "harsh noise" on each device. Participants play through a minimal interface while competing within a multiplayer setting. The primary game objective is to achieve the "harshest" (i.e., most distorted and abrasive) sound first. The system interconnects players' devices, synchronises their noise-generating instruments, and coordinates the underlying game logic. Further details can be found at the following open repository: `https://github.com/lucamorino/B3_H4RSH`

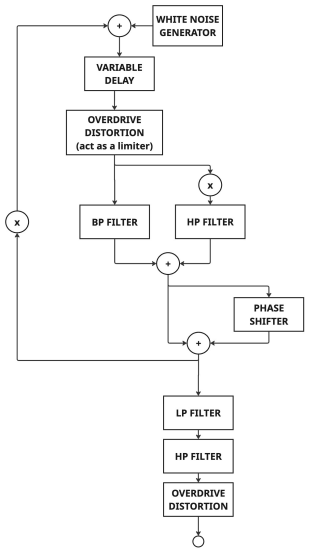


Figure 1. Digital no-input feedback loop schema

3.1 Digital No-Input Feedback Loop

Developed in RNBO [2], the audio engine implements a digital no-input feedback loop inspired by the no-input mixing desk practice [24]. The feedback loop is excited by a low-amplitude white noise signal and is rapidly driven and maintained in self-oscillation via high feedback gain (see Figure 1). A hyperbolic tangent function provides overdrive and soft limiting, preventing numerical blow-up (*NaN*). The signal is further shaped by a variable delay and two filtering stages: first, a *band-pass* filter sets the principal resonance (together with the delay time), while a parallel *high-pass* filter boosts high-frequency energy; second, a parallel, variable *all-pass* filter acts as a phase shifter, producing consequent timbral motion. At the output, an additional overdrive stage is applied, followed by high-pass and low-pass filters to constrain the spectrum to a bandwidth suited to smartphone loudspeakers.

3.2 The Player Interface

In the present preliminary release, the player interface is intentionally minimal. It comprises a real-time waveform display, a life counter [3], a pad, and a slider (see Figure 2). The waveform display and life counter provide players with immediate feedback on sound flow and game status, whereas the pad and slider control sound generation. The pad also functions as a gate – tap-and-hold to trigger and sustain the sound. Each new trigger initialises a random configuration of the synthesiser's parameters. The pad and slider values are not directly mapped to synthesis parameters. Instead, they define three-dimensional coordinates of a point within a virtual cube: the pad controls the horizontal plane (x–y axes), the slider sets the vertical coordinate (z axis). The distance between the player's position and a hidden target point – the goal – drives parameter evolution: as the distance decreases, distortion and modulation increase, producing a more aggressive sound and making the player aware of their proximity to the goal. When the player reaches the target point, the instrument is configured to produce the harshest possible sonic output.

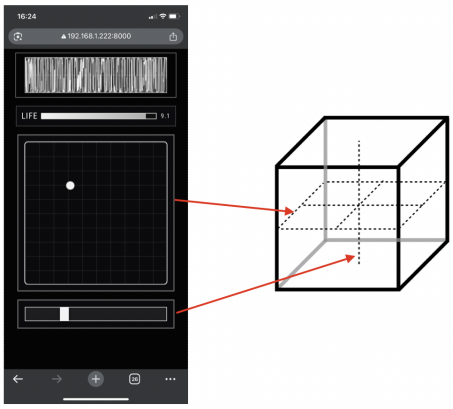


Figure 2. Left: Web application interface (Google Chrome on an iPhone 11). Right: cubic virtual interaction space. Arrows indicate the dimensions controlled by the XY pad and the slider.

3.3 *Harshness* Value

Harshness is a sonic quality associated with the perception of non-linearities in the audio signal [25]. In noise music, "harsh noise" denotes a specific subgenre, and harshness is often framed as an abrasive force arising from high sound-pressure levels and spectral saturation [26]. Although harsh-

[1] In the field of Human-Computer Interaction, "gamification" is defined as the use of game design elements in non-game contexts and can be seen as the extension of games beyond entertainment, such as serious games, pervasive games, alternate reality games, or playful design.

[2] https://rnbo.cycling74.com/

[3] see Section 3.4 for life point mechanism details.

ness may also correlate with the psychoacoustic dimension of *roughness*, in this work we adopt an *ad hoc* parametrisation that targets distortion-driven abrasiveness and is more closely related to *sharpness*, the concentration of energy in the upper portion of the spectrum.

Harshness parameter is estimated locally on each client device. A filter bank divides the audio signal into five bands (0-400 Hz, 400-800 Hz, 800-2000 Hz, 2000-4000 Hz, and over 4000 Hz). An envelope follower is applied to each band to estimate its energy. Based on empirical tuning of the noise instrument, the bands are then combined using unequal weights that emphasise higher-frequency content, yielding a single scalar *harshness* value. When this value exceeds a predefined threshold, a "harsh sound" event is detected, and an "attack mechanism" is triggered.

3.4 Game Design

Game mechanics constitute the core of the project, defining the objectives and dynamics of play and structuring the interplay among participants. Conceived primarily as a competitive player-versus-players scenario, each participant begins with a fixed number of life points. When a player's life points are depleted, their instrument becomes deactivated. The objective is to remain the last player able to sound. The gameplay is designed to underpin competition among participants by defining a valued sonic outcome: players seek to produce the "harshest" sound to affect other participants' life points. When a player's instrument exceeds the harshness threshold, an "attack mechanism" is triggered on the other players' devices, reducing their life points by 0.5 per second while simultaneously disrupting their sound generation. The attacking player sustains the harsh sound for as long as possible; however, after two seconds, the system automatically relocates the hidden target point to new random coordinates in the virtual space (as seen in 3.2). This alters the synthesis parameters and typically reduces the harshness measure below the threshold. Players under attack must endure the disruption and associated life-point loss unless they can exceed the harshness threshold and become themself "attacking" players.

3.5 System Architecture and Web Application

A client–server architecture is implemented in Node.js and deployed over a Wireless Local Area Network (WLAN). Both the server and the client application are developed in JavaScript, using the *soundworks API* [4] , a web framework that supports synchronisation and shared client-side state. The server runs as a Node.js process started by Max via the *node.script* object. Thus, a dedicated Max patch implements the primary logic for data control and automations. To bridge Max with the *soundworks* environment, the Max external object *soundworks.shared-state* [5] is used, which enables Max to communicate via WebSocket with the shared classes and schemas managed by *soundworks* across the server and clients (see Figure 3).

Although this architecture may appear even over-elaborate relative to the project's immediate aims, it offers several advantages for future development. Running the server within the Max environment facilitates integration into more complex audiovisual and interactive performance contexts, in which participants could control not only sound but also lighting and video processes implemented in Max. In addition, the system can be readily incorporated into Ableton Live via Max for Live, enabling its use within established digital audio workstation workflows.

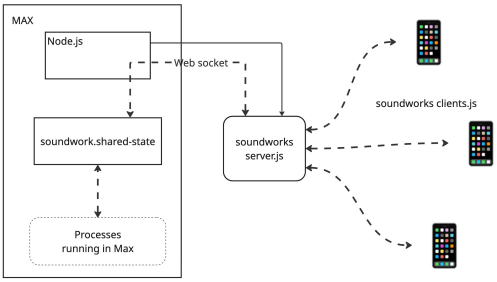


Figure 3. The system and the web architecture

4. FURTHER RESEARCH

B3-H4RSH is part of a broader research project aimed at examining sound-based interpersonal interactions that are musically meaningful within the context of mobile-mediated, game-driven participatory music. By framing gameplay as a layer for collaborative music-making, the research explores how game rules and goals can underlie collective agency over sonic outcomes, fostering playful and engaging experiences.

Building on the implementation of a mobile music system driven by multiplayer mechanics, the research will further investigate how different types of game mechanics can shape musical interplay and what interpersonal dynamics are elicited, such as group formation, coordination, and leadership, across competitive and collaborative gameplay. The expected outcomes include empirically grounded guidelines and design models for sound-based interaction and game mechanics, with potential applications to music performance and education, multiplayer gaming, and mobile entertainment.

5. CONCLUSIONS

B3-H4RSH is a co-located, browser-based mobile music system that combines collective music-making with competitive gameplay within a noise-music paradigm. In doing so, it establishes a form of sonic interdependence among players, driven by game mechanics, suggesting that game rules cannot only engage participants in collective music-making but also function as compositional devices that shape sonic material, participant agency, and social dynamics in sound-mediated participatory settings.

Acknowledgments

We acknowledge financial support by the Italian Ministry of University and Research (MUR) - PNRR Mission 4 Component 1, Investment 4.1 – M.D. n. 629/2024, within the National Recovery and Resilience Plan (NRRP) – CUP H62B24003590007.

[4] https://soundworks.dev/

[5] Available at https://github.com/collective-soundworks/soundworks-max, last access 31/12/2025.

6. REFERENCES

[1] P. Hegarty, "Noise Not Music," in *Noise as a Constructive Element in Music*. Routledge, 2022, pp. 72–85.

[2] P. J. Woods, "The abject pleasures of militarised noise," *Culture, Theory and Critique*, vol. 64, no. 1-2, pp. 196–211, 2023.

[3] G. Levin, S. Gibbons, G. Shakar, Y. Sohrawardy, J. Gruber, J. Lehner, G. Schmidl, and E. Semlak, "Dialtones (A Telesymphony): Final Report," PDF report, 2002. [Online]. Available: https://www.flong.com/storage/pdf/reports/dialtones_report.pdf

[4] A. Tanaka, "Mobile music making," in *Proceedings of the 2004 conference on New interfaces for musical expression*, 2004, pp. 154–156.

[5] K. Tahiroglu, N. N. Correia, and M. Espada, "PESI Extended System: In Space, On Body, with 3 Musicians." in *NIME*, 2013, pp. 35–40.

[6] Y. Wu, L. Zhang, N. Bryan-Kinns, and M. Barthet, "Open symphony: Creative participation for audiences of live music performances," *IEEE MultiMedia*, vol. 24, no. 1, pp. 48–62, 2017.

[7] S. W. Lee and J. Freeman, "echobo: Audience participation using the mobile music instrument," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2013, pp. 450–455.

[8] B. Matuszewski, N. Schnell, and F. Bevilacqua, "Interaction Topologies in Mobile-Based Situated Networked Music Systems," *Wireless Communications and Mobile Computing*, vol. 2019, no. 1, p. 9142490, 2019.

[9] A. Xambó and G. Roma, "Performing audiences: Composition strategies for network music using mobile phones," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2020, pp. 55–60.

[10] O. Hödl and L. Turchet, "The TMAP Framework: Describing the Design Space of Technology-Mediated Audience Participation in Live Music," *IEEE Access*, vol. 13, pp. 38 593–38 608, 2025.

[11] S. W. Lee, A. Willette, D. Koutra, and W. S. Lasecki, "The effect of social interaction on facilitating audience participation in a live music performance," in *Proceedings of the 2019 Conference on Creativity and Cognition*, 2019, pp. 108–120.

[12] L. Dahl, J. Herrera, and C. Wilkerson, "Tweetdreams: Making music with the audience and the world using real-time Twitter data," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2011, pp. 272–275.

[13] M. K. Koszolko and T. Studley, "From Site-Specific Sampling to Gamification: An exploration of performative engagement with the environment," *Organised Sound*, vol. 28, no. 3, pp. 338–351, 2023.

[14] T. Carson, "Mesh Garden: A creative-based musical game for participatory musical performance," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2019, pp. 339–342.

[15] T. Studley, "SwimTunes: A gamified music performance system for co-creating with a novice audience," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2025, pp. 415–421.

[16] M. Malt and B. Sluchin, "15. Game Theory and Formal Structure in Iannis Xenakis's Duel (1959)," *Meta-Xenakis: New Perspectives on Iannis Xenakis's Life, Work, and Legacies*, 2024.

[17] J. Brackett, "Some Notes on John Zorn's Cobra," *American Music*, vol. 28, no. 1, pp. 44–75, 2010.

[18] D. Parson, "Chess-based composition and improvisation for non-musicians," in *Proceedings of New Interfaces for Musical Expression (. . . , 2009.*

[19] S. Thon, "Go To Music-Move 37," in *International Computer Music Conference, ICMC 2024*, 2024, pp. 26–29.

[20] C. Michalakos, "Designing musical games for electroacoustic improvisation," *Organised Sound*, vol. 26, no. 1, pp. 78–88, 2021.

[21] T. Fukuda, "Super colliders: A gamified screen-score," in *Proceedings of the International Conference on Technologies for Music Notation and Representation*, 2020, pp. 158–168.

[22] A. Çamcı, C. Çakmak, and A. G. Forbes, "Applying game mechanics to networked music HCI systems," in *New directions in music and human-computer interaction*. Springer, 2019, pp. 223–241.

[23] T. J. Studley, "Exploring Real-Time Music Composition through Competitive Gameplay Interactions," Ph.D. dissertation, The University of Newcastle, Australia, 2020.

[24] T. Mudd, "Playing with feedback: Unpredictability, immediacy, and entangled agency in the no-input mixing desk," in *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, Hamburg, 2023, pp. 1–11.

[25] A. Anikin, K. Pisanski, M. Massenet, and D. Reby, "Harsh is large: nonlinear vocal phenomena lower voice pitch and exaggerate body size," *Proceedings of the Royal Society B*, vol. 288, no. 1954, p. 20210872, 2021.

[26] S. Graham, *Becoming Noise Music*. Bloomsbury Publishing, 2023.

Motion Capture: Mocopi and Sound Interaction in Max

Takayuki Rai
trai@me.com

Haruka Hirayama
Hokkaido Information University
hirayama@do-johodai.ac.jp

ABSTRACT

This paper presents a Local Space–based motion analysis method using Sony's Mocopi motion capture system[1] in the Max environment for real-time audio interaction. In previous work, joint position data were converted into World Space coordinates and applied to interactive audio control. However, identical body movements produced different coordinate values depending on the performer's facing direction, weakening the intuitive correspondence between bodily sensation and control data.
To address this issue, this study introduces a Local Space representation defined by the performer's body orientation. Joint positions are transformed from World Space into a forward-facing Local Space coordinate frame, enabling consistent detection of body movements regardless of orientation on stage.
Based on this framework, several Max external objects were developed to compute movement distance, direction, and angular features in Local Space. Live demonstrations confirm that movement-based audio control becomes more stable and intuitive, supporting interactive performance.

1. INTRODUCTION

In the author's previous paper presented at ICMC 2025, a relatively affordable and easy-to-use real-time motion capture system, Mocopi, released by Sony in 2024, was introduced. To realize sound interaction within the Max environment, several Max external objects running on macOS were developed. Joint data transmitted from Mocopi were converted into three-dimensional coordinates in World Space and visualized as a skeleton model using DIPS[2]. In addition, application examples were presented in which sound generated in Max patches was controlled based on inter-joint distances, polygonal areas formed by the body, and joint displacement.
However, these approaches were based on joint coordinates mapped into World Space, which revealed a critical limitation: the meaning of the XYZ coordinates—particularly in the XZ plane—changes significantly depending on the orientation and rotation of the body. As a result, even identical bodily movements yield different numerical values when the performer's facing direction changes, leading to a mismatch between the performer's embodied perception and the joint position data. This mismatch undermines bodily agency in interactions based on human motion.
In response to this issue, this paper proposes, as the second phase of the research, a method for computing joint positions based on Local Space, a body-centered coordinate system. By estimating body orientation and enabling the description of joint movements independent of facing direction, the proposed approach aims to establish a more direct correspondence between the performer's embodied perception and joint position data. This study presents a fundamental framework for restoring bodily agency in sound and visual interaction using motion capture data.

1.1 Data Flow from Mocopi to Max

This section briefly reviews the flow of motion data from Mocopi to Max, as described in the previous paper. Mocopi is a motion capture system composed of six sensors worn on the body. Data transmitted from these sensors are received by the Mocopi application developed by Sony, which runs on a smartphone. Based on these data, the application uses AI-based processing to convert the information into 27 joint positions in real time, which can be transmitted to a PC or Mac. The data can also be recorded locally on the smartphone as BVH files (Figure 1).

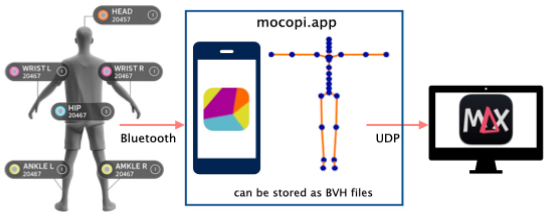


Figure 1. Mocopi data flow[3]

To receive the UDP data transmitted from the Mocopi application in the Max environment, the author developed a Max external object named mcpReceiver.mxo, which marked the starting point of this research. The received joint data are processed using OpenGL objects within DIPS and converted into 27 joint coordinates in World Space, which are then rendered as a skeleton model composed of 3D objects and lines within a DIPSWindow. Through this process, a basic framework for applying the Mocopi motion capture system in the Max environment was established.
Building on this framework, several Max external objects were developed to compute inter-joint distances, joint angles, and polygonal areas based on World Space joint data.

[1] https://www.sony.co.jp/en/Products/mocopi-dev/en/

[2] DIPS (Digital Image Processing) is a plugin software package for Max.

[3] The human body image shown on the left and Mocopi icon are based on sample visual materials provided by Sony for the mocopi system.

Using these objects, application examples were presented in which bodily movement was used to control sound processing in Max patches, such as filter control based on the distance between the wrists, amplitude control based on the area of a polygon formed by the body, and trigger generation for sound events in response to ankle movement.

2. LOCAL SPACE DEFINITION

In the first phase of this research, joint position data obtained from Mocopi were treated as coordinates in World Space and used to control sound and visual interactions. However, this approach revealed a fundamental problem: as the orientation of the body changes, the meaning of the XYZ joint coordinates also changes, resulting in different numerical values for identical bodily movements.

As a consequence, the correspondence between embodied sensation and data becomes unstable, making it difficult to achieve interaction based on bodily agency in performances and improvisational contexts. To address this issue, this study introduces a Local Space representation—a body-centered coordinate system—rather than relying on a fixed global reference frame.

2.1 Body Orientation Estimation

In this study, body orientation is estimated using the joints corresponding to the roots of the left and right legs, specifically Bone ID 19 (left leg) and Bone ID 23 (right leg) in the Mocopi data stream. These joints are adopted as the basis for constructing the Local Space, as the regions around the legs and waist tend to maintain relatively stable configurations during performance and bodily expression, making them suitable for estimating the overall facing direction of the body.

Specifically, the vector obtained from the positions of the left and right leg joints is defined as the lateral axis of the body. A direction orthogonal to this vector on the horizontal plane is then computed to determine the forward direction of the body. For the vertical axis, the global Y-axis provided by Mocopi, which is aligned with the direction of gravity, is adopted as the up direction.

Using these three axes—forward, lateral, and vertical—a body-oriented coordinate frame is constructed that translates and rotates together with the performer.

2.2 Joint Position Computation in Local Space

Once the body-oriented coordinate frame has been defined, joint position data are transformed from World Space into Local Space. The origin of the Local Space is defined by the joint corresponding to Bone ID 0, which represents the waist and serves as the center of the body.

First, a vector from the reference joint to the target joint is computed in World Space. This vector is then projected onto the three body-oriented axes defined in the previous section, yielding the joint coordinates in Local Space.

Through this process, movements such as extending an arm forward are represented by consistent numerical values in Local Space, regardless of the performer's facing direction. As a result, the relationship between bodily movement and numerical data becomes stable, enabling intuitive interaction grounded in embodied sensation.

2.3 Comparison of World and Local Space

To clarify the difference between World Space–based joint representations and the proposed Local Space representation, a simple bodily movement is considered as an example. When a performer extends the right arm forward in front of the body, the displacement is recorded along different axes in World Space depending on the performer's orientation. In contrast, because Local Space employs a coordinate frame defined relative to the body's orientation, this movement is consistently detected as a displacement in the forward direction, regardless of the direction the performer is facing.

Thus, Local Space representations provide a stable correspondence between bodily movement and numerical data, enabling interaction design that aligns more closely with embodied sensation. On the other hand, World Space representations are effective when dealing with relationships to specific positions or regions on stage—for example, determining whether the performer has touched a particular point in the performance space. Therefore, World Space and Local Space should not be regarded as opposing concepts, but rather as complementary representations that can be selected or combined according to the intended purpose.

3. IMPLEMENTATION IN MAX

3.1 Incorporating Body Orientation in Max

First, a Max external object named mcp2DHeading.mxo was developed. This object receives the positions of any two joints in World Space as input, performs directional calculations based on the method described in Section 2, and outputs the forward direction formed by the two input points as an angular value. In the basic configuration, Bone ID 19 (left hip joint) and Bone ID 23 (right hip joint) are used as inputs, and the forward direction of the waist is output as an angle in World Space.

At the same time, in order to visualize the body orientation obtained by mcp2DHeading within a Max patch, another object named headingDial.mxo was developed. This headingDial is a circular object that displays the forward direction in World Space as 0 degrees pointing downward, and represents the body orientation as an arrow rotating counterclockwise over a full 360-degree range. Using this object, it becomes possible to intuitively grasp the orientation of the body, or the perpendicular direction formed by any two points projected onto the XZ plane, directly within a Max patch (Figure 2).

By supplying joint position data such as Bone ID 14 (left shoulder) and Bone ID 18 (right shoulder) to mcp2DHeading, it is also possible to detect the orientation of the upper body alone. The target of direction calculation is not limited to the hip joints or shoulders; directions formed by the two feet, the two wrists, or any arbitrary pair of joints on the XZ plane can likewise be detected and compared. To simplify implementation for specific purposes, several Abstracts encapsulating mcp2DHeading—namely headingWaist, headingShoulder, headingHands, and headingLegs—have been prepared. In addition, an object called headingDial2.mxo was developed, which displays two input direction angles as arrows within a circular interface

and outputs the angular difference between them. This approach makes it possible, for example, to quantitatively measure the amount of torsion between the upper and lower body. The angular information related to body orientation obtained by these methods can also be used as control parameters for various sound processing operations generated within the Max environment.

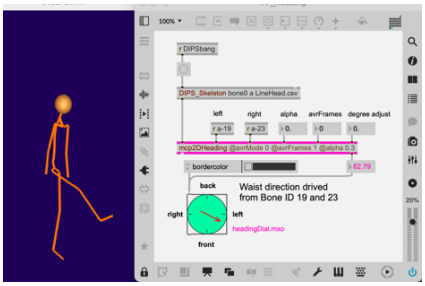


Figure 2. mcp2DHeading and headingDial

Furthermore, an object for detecting and displaying angular changes along the Y axis—particularly the relative angle of the head with respect to the body—is currently under development.

3.2 Converting World Space Data to Local Space

Next, a Max external object named mcpWStoLS.mxo was developed. This object takes as input the positional data of 27 joints that have been converted into World Space values within a Max patch, transforms these values into Local Space based on the body orientation, and outputs the corresponding 27 joint positions in Local Space.

At the same time, the body orientation itself is also output as an angular value in World Space, in the same manner as mcp2DHeading. To facilitate easier handling of the input and output of mcpWStoLS, the object is wrapped by an Abstract Object named m.LSout.

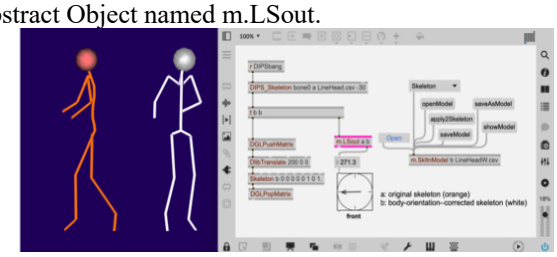


Figure 3. m.LSout

In Figure 3, a skeleton model based on World Space values (orange) is displayed on the left side of the DIPSWindow, while a skeleton model converted into Local Space (white) is displayed on the right. While the World Space skeleton model rotates in accordance with changes in the performer's body orientation, the Local Space skeleton model maintains its forward direction fixed along the positive Z-axis. In other words, regardless of the direction the performer is facing in World Space, the skeleton model converted into Local Space is always rendered in a forward-facing posture.

4. LOCAL SPACE APPLICATIONS

4.1 mcpMovedDistance.mxo

mcpMovedDistance.mxo extends the body motion detection object mcpMovement.mxo developed in Phase 1 and is designed to allow switching between World Space and Local Space for evaluating motion direction. The object computes and outputs the displacement between any two frames as the overall magnitude (XYZ) as well as axis- and plane-based components, enabling detection of specific components such as motion along the Z-axis or within the XZ plane (with optional averaging).

To demonstrate the effectiveness of the Local Space representation, a simple sound control example was implemented in which a forward kicking motion of the right foot was assumed. A Max patch was constructed to generate a trigger at the onset of the foot movement and play a sound when the measured displacement of the right foot between specified frames exceeded a predefined threshold. While World Space–based processing detects forward motion along different axis components depending on the performer's facing direction, Local Space–based processing enables consistent trigger generation for the same movement regardless of orientation.

As a result, the action of "kicking the foot forward" can be stably detected independent of body orientation and used as a trigger to control the same sound event. It should be noted that mcpMovedDistance.mxo internally switches between World Space and Local Space for evaluating motion direction and does not require mcpWStoLS, which converts joint positions from World Space to Local Space.

4.2 mcpMovedAngle.mxo

Similarly, detection of movement angles in Local Space has also been explored. Using movement angle as a parameter makes it possible to evaluate motion independently of body size or limb length. This property is particularly useful for applying the same interaction design across multiple performers with different physical conditions and is expected to be beneficial for future developments.

5. OBJECT IMPROVEMENTS

This section reports improvements to several general-purpose patches and objects included in the project and briefly discusses future directions.

5.1 DIPS_Skeleton.maxpat

This research began with an investigation into methods for rendering skeleton models in the Max environment based on joint data obtained from Mocopi. After presenting the basic approach in the previous paper, a new Abstract patch named DIPS_Skeleton was developed to simplify the rendering process.

This patch allows parameters such as the bone data stream name received via UDP, the skeleton name, and the skeleton model definition file (CSV) to be specified as arguments. In addition, the visibility of the DIPSWindow and the rendering and editing of the skeleton model can be controlled via inlets. As a result, the number of skeleton-rendering objects required in the main patch is reduced, making it easier to organize video and audio processing components.

Furthermore, mcpSkltnModel.mxo, which saves skeleton models as CSV files, was functionally updated and wrapped as m.SkeletonModel. Improvements and extensions of Mocopi-related objects, including those around

mcpReceiver, will be continued in future work.

5.2 JIT_Skeleton.maxpat

Development is also underway for a skeleton-rendering patch in the Jitter environment (Figure 4). In this implementation, maintaining compatibility with CSV files defined for DIPS is adopted as a core design principle. At present, bone data are converted into World Space using DIPS objects; however, future versions are planned to eliminate this dependency.
In addition, approaches for rendering a human body model as an avatar, rather than as a skeleton model, within the Jitter environment are currently under investigation (Figure 5).

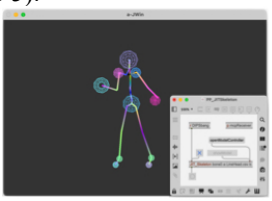


Figure 4. Jitter visualization

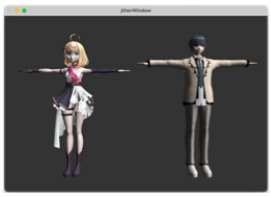


Figure 5. Avatars[4]

5.3 Other Auxiliary Objects

Objects developed in Phase 1, such as mcpTriangle.mxo and mcpPentagon.mxo, were originally designed to output polygonal areas and perimeters. In future work, these objects will be extended to additionally output angle information at each vertex. Because angle-based features are less affected by differences in body size or physique, they are expected to serve as effective parameters for pose detection and for the integrated analysis of multiple movements and joint actions.
In addition to bvhsender.app provided by Sony, this research has also developed objects such as mcpBVHplayer.mxo for directly playing BVH files within the Max environment. Similar to the mcpBuffer family of objects presented in the previous paper, these objects enable frame-based playback as well as control of playback speed and direction, thereby contributing to more efficient patch construction and verification.

6. DEMONSTRATION

The first phase of this research was presented at ICMC 2025. In addition, a poster presentation was given in September 2025 at the 144th SIGMUS (Music Information Science) meeting of the Information Processing Society of Japan. The second phase of the research presented in this paper will also be demonstrated as the first on-site, hands-on presentation outside Japan, in a Poster + Demo format.

[4] The skeletal mesh used for visualization on the left was based on RAYNOS.dae, a sample 3D model publicly distributed by Sony as part of the mocopi system.

7. CONCLUSIONS

7.1 Advantages

As shown in the previous presentation, the primary advantage of applying body-movement–based creation using Mocopi in the Max environment is that, unlike camera-based systems, it enables stable detection of bodily motion regardless of lighting conditions on stage, including movements occurring in occluded regions of three-dimensional space, in addition to information from the frontal direction. Furthermore, with the Local Space–based body movement detection method proposed in this study, this advantage is further extended, making it possible to establish a natural and intuitive correspondence between bodily movement and sound, regardless of the direction the performer is facing on stage.

7.2 Toward a Unified World/Local Space Design

In this Phase 2, development has focused primarily on objects designed around the assumption of Local Space. However, if joint data in World Space can be consistently transformed into a forward-facing Local Space representation using mcpWStoLS.mxo, it becomes unnecessary to strictly distinguish between objects designed for World Space and those designed for Local Space, as processing can be performed based on the transformed data.
In this case, the body-movement analysis tools developed in Phase 1 can also be reused as Local Space–based processes. At present, the benefits of continuing to develop separate object sets for World Space and Local Space are limited, aside from differences in computational load. Therefore, future work will explore a unified object design that does not strictly separate World Space and Local Space.

7.3 Future Plans

As a future direction, we plan to investigate methods for extracting multiple motion elements from the diverse information derived from body movement and handling them in an integrated manner. Through this approach, we aim to realize more advanced interactions based on bodily performance and to open up new expressive possibilities in improvisational performance and artistic creation.

8. REFERENCES

[1] T.Rai, “Developing Max Objects for Mocopi: New Motion Capture System,” Proceedings of ICMC 2025, Boston, USA, 2025, pp. 590–593.
[2] Cycling’74, “Max SDK 8.2.0 Documentation,” Cycling’74. 2021.

Architectures of Alteration: Designing and Integrating Hybrid Kinetic Robotic Systems and Light Choreography in Eternal Dawn

Mohammad Sadeghi[1,2] Kyrillos Adeeb[1,2] Inderawes Khalil[1,2] Diego Muhr[3]

Alexander Schubert[3] Thorsten A. Kern[1,2]

[1]Institute for Mechatronics in Mechanics, Hamburg University of Technology (TUHH), Eissendorferstr. 38, 6 21073 Hamburg, Germany.

[2]Haptic Lab, Ligeti center - Laboratories for Innovation and General-audience Edification through the Transfer of Ideas (Ligeti), Veritaskai 1, 21079, Hamburg, Germany.

[3]Faculty of Music, Department of Multimedia Composition, Hochschule für Musik und Theater Hamburg (HfMT), 20148 Hamburg, Germany.

ABSTRACT

Contemporary performance increasingly relies on kinetic, robotic, and responsive environments that demand tightly integrated engineering systems capable of acting as expressive agents. Developing such hybrid systems contributes to new modes of staging, embodiment, and dramaturgy, offering artists tools for creating dynamic environments that extend beyond the limitations of human gesture alone. This paper presents the design and integration of two hybrid kinetic systems developed for the performance Eternal Dawn*: a ceiling-mounted robotic arm and a motor-matrix architecture controlling suspended rectangular light frames. The robotic arm operates as a supervisory and interactive entity, shifting from analytical scanning to aggressive pendulum-like motion to intimate duet-like encounters. The motor-matrix system dynamically reconfigures the spatial geometry of the laboratory, synchronizing kinetic light choreography with sound and movement to construct adaptive architectural states. Synchronization with musical structures is achieved using Open Sound Control (OSC) messages ensuring accurate temporal coordination. The motors are controlled via a programmable logic controller (PLC) and a dedicated human–machine interface (HMI) managing motion parameters, sequencing, and safety functions. The proposed systems proved effective as expressive kinetic agents, demonstrating a versatile platform for integrating robotic motion and dynamic light architectures into similarly experimental performance setting.*

Keywords: *Installative Music Theatre, Hybrid Kinetic Systems, Robotic, Light Choreography, Open Sound Control, Programmable Logic Controller.*

1. INTRODUCTION

In recent years, artists in the field of performance, installation art, and music theatre have embraced machines not merely as stagecraft, but as collaborators in meaning-making. Technologies such as robotics, motion control, and responsive lighting are increasingly woven into the core of dramaturgy, serving as interactive partners that shape spatial, visual, and choreographic outcomes in real time [1]. From algorithmic choreography to robotic scenography, these practices redefine presence and participation on stage, giving rise to hybrid forms of expression. Since the early 2000s, artists and researchers have explored the limits of human embodiment through technological augmentation. Surgical performance and cybernetic prosthetics have interrogated bodily norms and proposed the obsolescence of the biological body [2-3]. More recently, permanent implants enabling expanded perception, such as hearing color or sensing seismic activity, have highlighted posthuman modes of embodiment [4]. These experiments laid the groundwork for integrating technology as an active dramaturgical force. Performances that enabled audience-controlled bodily manipulation or involuntary robotic choreography further pushed the integration of machine agency into live staging [5]. Other works have featured robotic scenography capable of replacing human actors while conveying narrative and emotional content [6]. Theoretical frameworks describing these developments define such performance as the co-constitution of body and technology [7]. These hybrid systems generate shared embodiment and feedback, allowing agency, gesture, and meaning to emerge through entangled human–machine interaction [3-4]. In this context, Eternal Dawn extends the mentioned ideas through an installative music-theatre performance that stages bodily transformation in a fictional laboratory. Over the span of an artificial dusk-to-dawn cycle, performers interact with a robotic arm and suspended light frames, systems designed to behave expressively. Their motion is synchronized with music and dramaturgical cues, supporting an evolving narrative of augmentation, supervision, and distributed control.

Robotic systems and automation in performance offer both creative potential and technical challenges. While they provide precise motion and symbolic presence, they often require complex interfacing and careful safety management. This paper presents the technical design and implementation of Eternal Dawn's kinetic systems. A ceiling-mounted industrial robotic arm and a motor-matrix light architecture were programmed using OSC-over-MIDI with Max/MSP for musical synchronization. Motion control was handled via a programmable logic controller (PLC) and a human–machine interface (HMI), enabling real-time adjustments and layered behavior modes. The proposed design and fabrication approach offers a practical methodology for embedding adaptive robotic scenography into other artistic productions, where

machines operate not just as kinetic objects but as expressive agents integrated into the performance's dramaturgical structure.

2. DRAMATURGICAL CONCEPT

Eternal Dawn unfolds as a performative laboratory, which explores states of bodily transition, mediated through engineered systems and artificial rituals. Set within a clinically designed circular stage environment, the work portrays a speculative space in which seven performers undergo incremental physical and cognitive alterations. The narrative arc is structured between a preparatory prelude and a reinitialization coda, implying the continuity of an experiment whose outcomes are neither stable nor conclusive. Rather than presenting transformation as linear or heroic, the performance reveals a looped ecosystem of technical trials, configurations of bodies, tools, and interfaces tested across a cycle that mimics natural twilight and reemergence. Central to the conceptual design is the notion of evolution as supervised process. Augmentations are not autonomous enhancements, but interventions that emerge from a system larger than the individual. Throughout the piece, performers acquire and interact with prosthetic structures, wearable sensors, chemical processes, and connective data cabling. These interfaces operate in a shifting field of intention: some offer agency or resonance; others restrict, manipulate, or exceed performer control. This ambiguity destabilizes the boundaries between user and device, evoking a vision of embodiment as modular and unstable, a system of parts in active negotiation rather than fixed unity. While the work gestures toward transhumanist motifs, enhancement, interfacing, sensory expansion, it simultaneously challenges them by highlighting the erosion of singular identity. In one scene, human cognition is symbolically transferred to a quadruped robot through a mechanized "birth," while in another, the ensemble becomes physically linked via data umbilicals, dissolving autonomy into collective circuitry. These moments reflect a posthuman aesthetic in which control is diffused, and performance emerges through interrelation rather than mastery. The visual and spatial architecture mirrors these ideas. A ceiling-suspended robotic arm takes on various affective and functional roles observing, manipulating, threatening, and embracing performers in different phases. Surrounding it, a network of suspended light frames defines the perimeter of the stage. Their motion is governed by a custom-built automation system referred to as the Motor-Matrix, which enables coordinated transformations of the environment. These frames subtly reshape the spatial topology and performers' relation to the architecture. Far from being passive set pieces, they act as kinetic agents that generate tension, rhythm, and shifts in orientation. Together with the robotic arm, the Motor-Matrix contributes to a responsive dramaturgy in which mechanical elements shape interaction, proximity, and atmosphere. In this setting, transformation is not linear or goal-driven, but recursive and environmental, an open system of affective, mechanical, and symbolic change. Figure 1a provides a visual overview of the stage layout, showing the placement and interaction of the key elements discussed.

3. SYSTEM DESIGN AND METHODS

3.1 Robotic Arm Motion

The design of the automation system, including the interfaces to the robotic components, is presented in the following section. The first section focusing on the robotic arm and the second section discussing the motor matrix.

3.1.1 Mechanical configuration

The robot was planned to be mounted at the center of the ceiling to enable interactive motion with the performer during the show. In this context, a high degree of freedom (DOF) motion was essential to help the artists simulate more complex and flexible gestures through the robot. For this purpose, the industrial version of the Universal Robot (UR10e) was selected. To extend the robot's reach and enhance its expressive capabilities, a custom 2-DOF arm extension was designed and fabricated. One DOF enabled a bending motion, while the other provided rotational movement at the endpoint, where a light tube was attached. A key engineering challenge was the limited payload of the UR10e under dynamic motion, which imposed constraints on motor selection due to weight restrictions. Simultaneously, the motors needed to generate sufficient torque during various motion phases. After initial calculations and benchmarking, the DYNAMIXEL XM540-W270-R servo motor was selected for its lightweight (165 g) design and holding torque of up to 12 Nm. The mechanism was evaluated through dynamic motion simulations before fabrication, as described in the following section. The final mounted robot with its custom arm extension is illustrated in Figure 1b. The inset of Figure 1b highlights progressive iterations of the mechanism, which was fabricated using PETG-CF filament on a Bambu Lab X1 3D printer.

3.1.2 Kinematic Simulation

SolidWorks Motion Analysis provides a physics-based multibody dynamics environment that enables dynamic evaluation of mechanisms directly from their CAD assemblies. The solver can incorporate real-world effects such as gravity, external forces, friction, dampers, and contact interactions. Using the mass, center-of-mass position, and inertia tensors automatically extracted from the CAD geometry, the software constructs a constrained dynamic system that satisfies Newton–Euler/D'Alembert principles. Although SolidWorks does not explicitly formulate the robot dynamics in closed form, it numerically integrates an equivalent set of multibody equations of motion. In other words, Motion Analysis provides an effective framework for evaluating dynamic behavior, exporting joint loads, and validating actuator requirements for structural design.

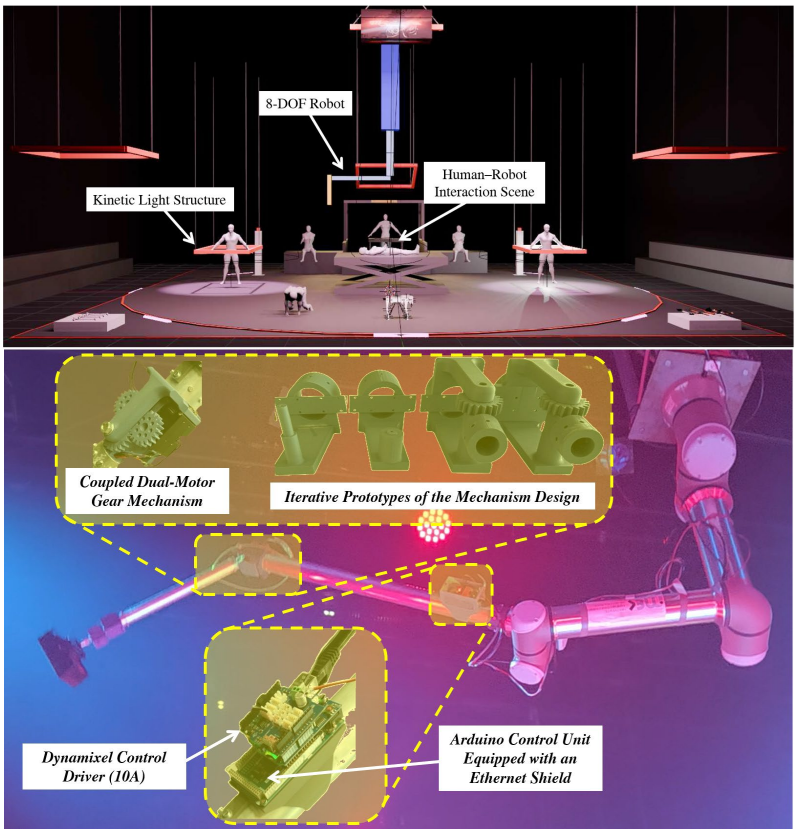


Figure 1. (a) Stage configuration schematic, designed in capture software;
(b) UR10e robot with custom-designed arm extension, control electronics, and prototype mechanism.

In this project, with the 3D CAD model of the robot, including the proposed arm extension, the kinematic properties of each joint were defined according to the manufacturer's specifications for the Universal Robots UR10e. The resulting assembly was driven using user-specified motion inputs representing the desired trajectories, enabling the simulation to compute the reaction forces and torques required at each motorized joint and thereby perform a numerical inverse-dynamics analysis. Using this approach, torque–time profiles for all joints were obtained along the prescribed trajectories, inherently capturing gravitational loading and inertia-coupling effects. Figure 2 depicts the simulation results.

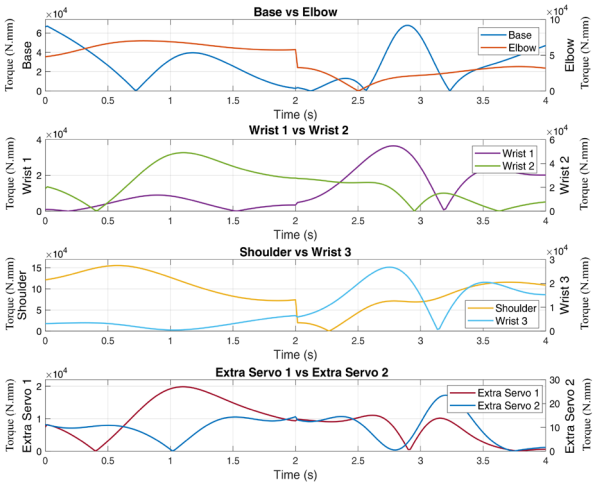


Figure 2. Simulated torque profiles for robotic joints during critical motion cycles.

As can be seen, the maximum torque values for the motors of the Universal Robot were within an acceptable range, indicating that the robot can handle the motion. The results reveal that for the extension arm mechanism, under the most critical condition, approximately 19 Nm of torque is needed to handle a payload of approximately 2 kg. The weight estimate is based on the combined mass of the 3D-printed mechanism, three servo motors, and the light tube attached to the tip of the extended arm. Simulations suggested that two motors should be coupled to meet the required torque output in the bending mechanism.

3.1.3 Control Implementation

The control hardware (inset of figure 1b) consists of an Arduino microcontroller equipped with a Dynamixel Shield for TTL half-duplex communication and an Ethernet Shield that exposes a TCP command port used by the Universal Robot controller. The Dynamixel X-series actuator is operated in Extended Position Mode, with motion parameters configured at runtime through PROFILE_VELOCITY and PROFILE_ACCELERATION registers. All actuator interaction uses Protocol-2 bulk instructions. In this regard, the SyncWrite is used to issue atomic Goal Position updates, while SyncRead retrieves Present Position, Present Velocity, and Present Current at high rate for feedback and logging. A parallel SyncRead polls the Hardware Error Status register as part of a continuous watchdog loop. When a non-zero fault code is detected, our specifically designed firmware for this stage

performance triggers a self-healing sequence that performs torque-off, reboot, and automatic re-initialization of operating mode, current limit, and motion profiles, ensuring uninterrupted operation during long experimental runs. This architecture provides deterministic command timing, high-frequency telemetry, and robust autonomous fault recovery within the UR–Arduino–Dynamixel control chain.

3.1.4 HMI Logic Architecture and Communication

As the total length of the robot, including the arm extension, was approximately 3 meters, and it needed to be mounted on a ceiling with a height of around 5 meters, a custom human–machine interface (HMI) was essential for safe and efficient training. Manual positioning was impractical due to limited physical access, so the HMI enabled remote control of the robot in 3D space, allowing operators to move it to desired positions during the choreography process. To streamline training and rehearsal, the interface also included features for editing motion sequences, adjusting pause durations, and fine-tuning the timing of each programmed scene. These tools significantly accelerated the development of complex, expressive movements. Figure 3 shows the flowchart of the motion recording workflow used to program the multi-joint robotic system. The process begins with the operator activating the system and entering the motion recording mode. Robot joints can be adjusted using increment buttons or by manual repositioning of the robot to the desired position. Once the desired configuration is reached, the operator can record either permanent waypoints or temporary test positions. Temporary waypoints allow the operator to preview the motion with a specific duration, enabling real-time assessment and duration tuning before finalizing the point. Once confirmed, the waypoint is saved, and the user may choose to delete or re-record recent entries as needed. The system supports direct navigation to any saved point by name, streamlining testing and sequencing. After a sequence of waypoints is completed, the trajectory can be played back or saved for future use. A postprocessing script is optionally applied to smooth the trajectory for stage-ready execution. Additionally, previously saved sequences can be loaded for quick deployment. This interactive and modular structure facilitates precise motion programming while preserving creative flexibility during iterative development. It is particularly well-suited for robotics integrated into performative contexts such as theatre or installation art.

3.2 Motor Matrix: Light Choreography

3.2.1 Mechanical configuration

The light choreography system was shaped by two dramaturgically distinct but technically related elements: one focusing on illumination and the other on motion. Element one consisted of standalone vertical Titan Tubes (1.05 m, Astera, Germany) that served as direct light sources and were each suspended and actuated by an individual motor unit. Element two comprised larger rectangular aluminum frames, each constructed from eight Titan Tubes, with their four corners connected to dedicated motorized units mounted to the ceiling.

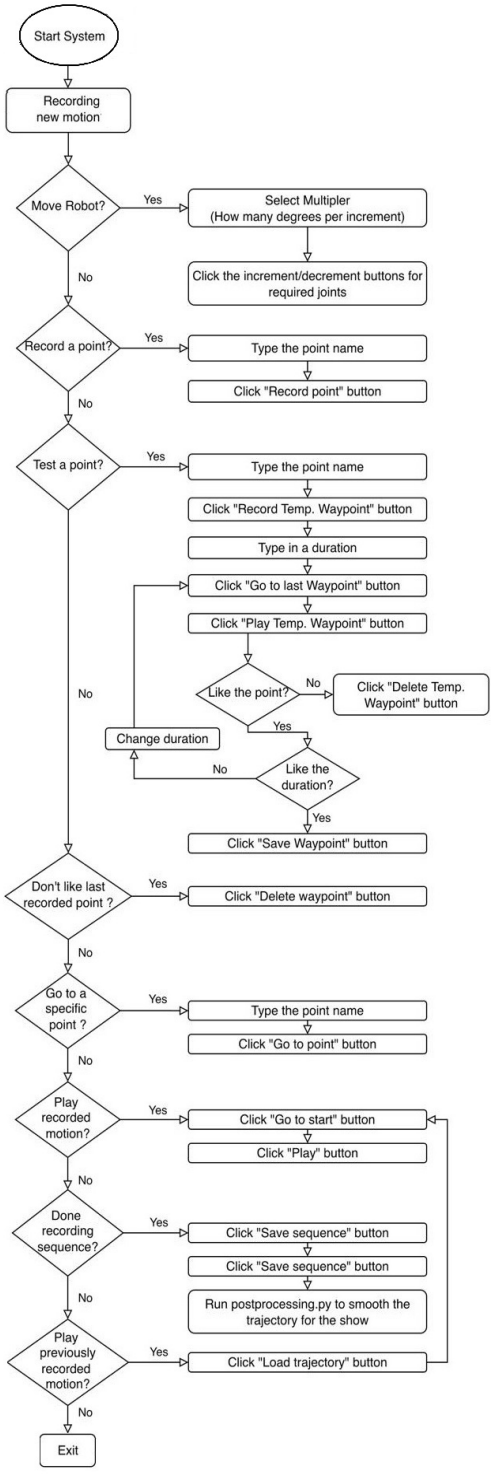


Figure 3. Workflow diagram illustrating the procedure for recording, testing, editing, and saving motion and sequences within the motion-authoring interface.

While differing in scale and spatial behavior, both elements used similar motorized winch systems for vertical and articulated movement. Together, they formed a unified kinetic-light environment capable of dynamic spatial reconfiguration and synchronized light choreography.

The frame was capable of multiple motions, including vertical translation (Z-axis), roll, pitch (half-tilt), and circular trajectories synchronized with musical structures. The Titan Tubes supported pixel-level control, tunable white light, and adjustable brightness via wireless DMX. Each frame weighed approximately 20 kg. To support expressive motion, the target speed range for the system was set between 10 mm/s and 500 mm/s, with a maximum vertical travel of 8 meters, determined by the height of the theater's ceiling system. A motion response time of up to 500 ms was deemed visually imperceptible and acceptable for choreographic purposes. From an engineering perspective, the mechanical architecture was defined by this motion envelope and weight requirement. Each motorized unit employed a winch system using steel cable (2 mm diameter, 3 kN tensile strength) and a spool mechanism (SD300k, max load: 14 kg) for wrapping and unwrapping the cable. A mechanical guide was incorporated to prevent rope entanglement during operation. To ensure a safety factor of 2.0, each motor was designed to support at least 10 kg of load. Motor selection was based on required torque, power output, and shaft compatibility. Accurate motor power calculation is essential to ensure that the coil current remains within safe operating limits, preventing thermal overload during prolonged use. Reducing spool diameter was prioritized to minimize dynamic torque and improve mechanical efficiency. A stepper motor was selected to achieve motion precision within millimeter tolerances. Figure 4 illustrates both the schematic and the fabricated version of the Motor-Matrix unit.

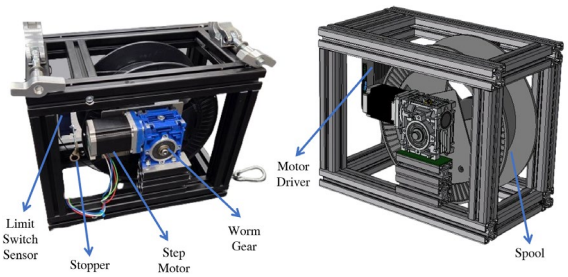


Figure 4. Schematic (right) and fabricated (left) Motor-Matrix unit showing key components.

The Sanyo Denki SM2564 (1.8° step angle, 2.5 Nm holding torque, 2 A/phase, 24 V DC) offered an appropriate balance of resolution and cost-effectiveness. To enhance torque and provide passive braking, a worm gear (NMRV30, Xingda Machinery, China) with a 1:15 ratio, 20 Nm permissible torque, and 76% efficiency (η) was coupled to each motor. The self-locking property of the gear added a safety feature in case of power loss. To define the upper stop position, a mechanical limit switch (XCKN2145G11, Schneider Electric, Indonesia) was installed. This normally open switch was selected based on its rated actuation speed and minimum operational torque. A 480 W power supply (HLG480H-24A, Mean Well, Taiwan) provided 24 V DC with constant current regulation to power the stepper motors, drivers, and control unit. Four motors were grouped to simplify system integration. Equations 1–4 describe the relationships governing mechanical power, linear velocity, electrical load, and response timing, serving as foundational models for similar Motor-Matrix motion systems.

$$P_{mech} = \tau \times \omega = s_f \times r_{spool} \times \left(f_{fric} + \frac{mv^2}{4d_{travel}} + mg \right) \quad (1)$$

$$P_{elect} = I_{phase} \times v_{supply} \quad (2)$$

$$v_{linear} = \omega \times r = \frac{maximum\ motor\ RPM}{Gear\ ratio} \times \frac{2\pi}{60} r_{spool} \quad (3)$$

$$t_{switch} \leq \frac{d_{overtravel}}{v} - t_{system} \quad (4)$$

where P_{mech} is the mechanical power, τ is the spool torque, ω is the angular velocity, s_f is the applied safety factor, m is the suspended mass, P_{elect} is the electrical input power, I_{phase} is the motor phase current, v_{supply} is the driver supply voltage, v_{linear} is the linear motion velocity after gearing, t_{switch} is the required limit-switch response time, and _ is the system latency.

3.2.2 Control Architecture and Communication protocol

From an engineering perspective, system reliability was a critical design consideration, as any technical failure could significantly disrupt the artistic continuity of a live performance. To ensure robust and deterministic control, a programmable logic controller (PLC) module (Siemens S7-1200, model 6ES7214, Germany) was selected to generate high-frequency pulse signals for the stepper motor drivers. Each motor was driven by a two-phase DM542T driver (China) configured with over-current protection and set to 1600 pulses per revolution, based on the desired motion resolution and speed range. To simplify wiring and communication architecture, each PLC unit controlled four motors using pulse and direction signals, with a shared enable line. The complete system of four PLCs, controlling 16 actuators, was interconnected via the Profinet protocol, enabling real-time monitoring and coordination of actuator positions. Profinet high-speed protocol, deterministic data exchange ensured reliable synchronization across distributed motor units, which was essential for maintaining precise motion cues during live performance. The selected PLC model provided 10 digital outputs (two per motor for pulse and direction signals, plus one shared enable) and 14 digital inputs, four of which were used for limit switch integration. The system supported 4 dedicated switching frequency of up to 100 kHz, allowing precise control of motion speed and timing. A human–machine interface (HMI) was integrated to enable real-time system monitoring, manual override functionality, and fault diagnostics during both setup and live performance. Communication between the HMI and the PLCs was established via the Profinet protocol, ensuring fast and secure data exchange. For this purpose, the KTP400 Basic Panel (Siemens, Germany) was selected. The design objectives and functional layout of the

HMI are described in the following section. Figure 5, shows the logic architecture of the system.

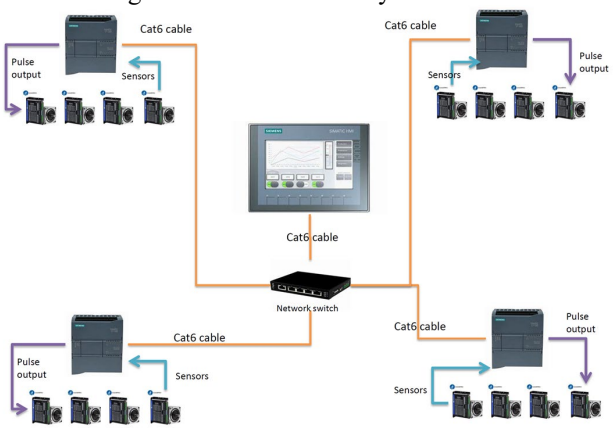


Figure 5. Communication architecture of the Motor-Matrix system.

3.2.3 Interface Design and Program Logic

The design of the HMI prioritized clarity, usability, and responsive control across multiple motorized units during both performance and setup. A centralized welcome screen displays the status of all four PLC units, each indicated by blinking icons to confirm real-time connectivity. From this overview, the operator can navigate directly to the dedicated interface for each PLC-controlled group, either a frame or a set of standalone motors. Each control page provides access to motion modes including translation, roll, pitch, and circular movement, with user-defined parameters such as velocity, range, and cycle period. Position feedback is displayed numerically to support precision control and diagnostics, while Open Sound Control (OSC) activation and reset buttons allow for synchronized motion during live performance. A separate OSC monitoring page provides real-time signal visualization across all PLCs, streamlining troubleshooting during rehearsals or shows. Safety is ensured through color-coded emergency stop buttons accessible on all screens. Figure 6 depicts the flowchart outlining the core logic of the PLC control program used for Motor-Matrix initialization, motion selection, and real-time execution. Upon cycle start, the PLC reads all relevant inputs, such as limit switch status and axis states, and updates system flags on the HMI. A homing procedure is triggered either manually or through Flash Recovery mode, ensuring that each axis reaches its zero-reference point. Once all axes are homed and idle, the system enters a ready state and awaits user input to select the motion control mode. The operator may choose between manual jog/relative control or automatic periodic motion, with each motion path governed by safety constraints including speed, range, and duration. Alternatively, the system may enter an OSC mode, where commands are received via UDP packets. In this mode, the system checks that no axes are in motion before decoding motion types and updating input values based on force limits and dynamic constraints. The motion commands are continuously monitored at 300 ms cycles to check for stop conditions.

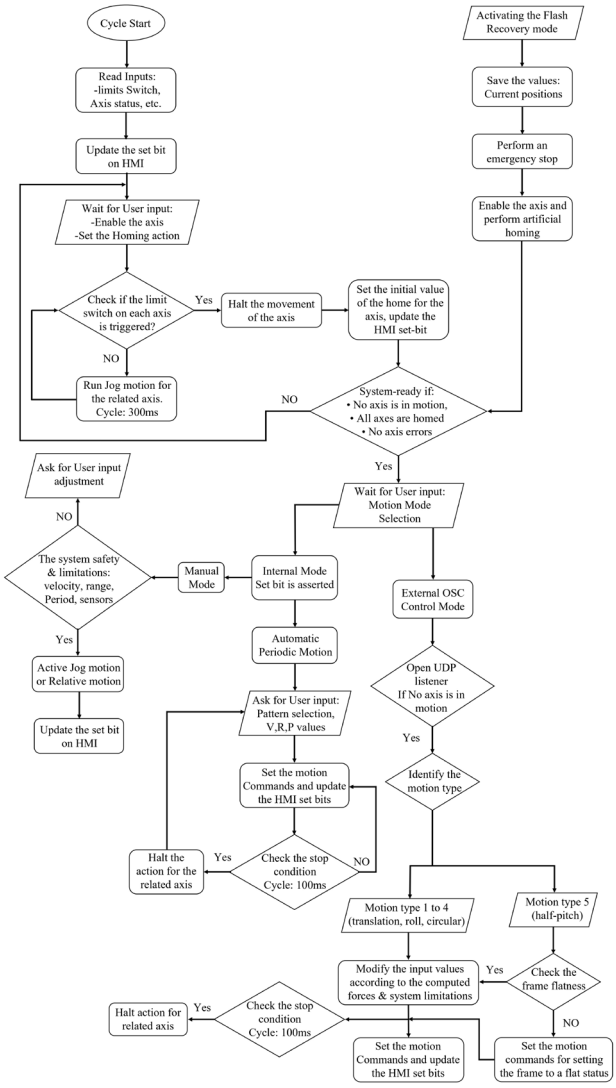


Figure 6. Flowchart of the PLC control logic for initialization, motion selection, and runtime execution.

4. WORKFLOW AND INTERFACES WITH THE STAGE PLATFORM

4.1 Robotic Actuation Interface and Dynamics

4.1.1 HMI-Assisted Choreography Programming

The development of a custom human–machine interface (HMI) was essential in bridging the gap between technical complexity and artistic intent during the robot choreography process. The HMI allowed artists to program sequences spatially, testing movement, timing, and coordination with music elements without requiring direct manual manipulation. Designed with a choreography-driven workflow in mind, the interface featured joint-level control via increment/decrement buttons, a selectable multiplier for precise movement, and tools for recording, testing, and refining waypoint sequences. The layout supported intuitive navigation, enabling users to preview transitions, edit durations, and iterate on timing to match the expressive needs of the performance. Two dedicated pages structured the workflow: one for trajecto-

ry recording and motion refinement, and another for execution and OSC-based control during live shows. The modular interface supported seamless toggling between manual and synchronized modes, promoting tight integration with musical and spatial design. This engineering solution empowered choreographers to shape the robot's gestures as expressive components of the performance, emphasizing the role of real-time interfacing in supporting embodied, musical storytelling. Figure 7 shows the designed HMI interface.



Figure 7 Designed HMI recording interface for joint-level control, waypoint programming, and motion sequence management.

4.2 Motor-Matrix Dynamic and Interface

The dynamic behavior of the four-motor rope matrix requires continuous verification that commanded motions remain within the allowable load range for each actuator. For motions with non-uniform load distribution such as roll, circular trajectories, or partial pitch, the controller computes the expected rope forces and automatically restricts user inputs to safe operating zones. In practice, the circular and roll motions were limited to ±300 mm of lateral displacement to prevent excessive torque on individual motors, while the half-pitch motion was constrained to 600 mm, approximately forty-five degrees, with a reduced speed limit of 50 mm per second to satisfy the artistic requirement for slow and expressive movement. For generating circular motion, the frame is driven by a single master angle that advances in fixed steps based on the allowed velocity, acceleration, amplitude, and cycle time. At each update, the controller computes phase-shifted sinusoidal offsets, $L1 = a \cdot \sin(\theta)$, $L2 = a \cdot \cos(\theta)$, $L3 = -a \cdot \sin(\theta)$, and $L4 = -a \cdot \cos(\theta)$, so that opposite motors move in mirrored patterns and the frame traces a stable and repeatable circular path. All user inputs are clamped to hardware-safe limits, and a rising-edge start trigger resets the master phase and cycle counters to ensure reproducible motion. The final version of the designed HMI, developed according to the engineering requirements outlined in Section 3.2.3, is shown in Figure 8. The layout clearly separates high-level frame selection from detailed pattern-control pages, offering an intuitive workflow for both technical operation and live artistic performance. During performance, the monitoring page enables OSC status indication, allowing the operator to supervise all four PLCs; when an OSC message is received, the corresponding "PLC-OSC" indicator flashes to confirm command reception and motor activation. For safety, the same page includes an emergency button and a recovery-flash function that allows the operator to restore the system within ten seconds without requiring a homing procedure. For detailed inspection of actuator states, the operator navigates to the second-level control pages by selecting the corresponding PLC on the monitoring screen, where real-time feedback, current motor positions, and active modes are clearly displayed.

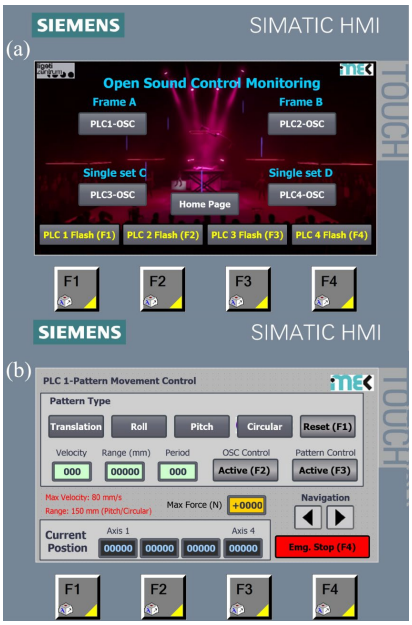


Figure 8. (a) Main HMI page for selecting frame control and monitoring functions. (b) Motion-control interface showing pattern selection, and position feedback.

4.3 Integration in a Live Performance Environment

In the futuristic music-theater project, multiple technological elements, including music, lighting, fog systems, motion lighting, and robotic movements, must operate in a synchronized manner to fulfill the artistic concept in real time. To achieve this, both the Motor-Matrix and Robot systems were controlled via OSC, with master control coming from Max/MSP and integrated into the grandMA3 console (MA Lighting GmbH, Germany). The OSC–MIDI communication pipeline ensures accurate timestamping and temporal alignment between all components. Algorithm 1 shows the simplest version of the OSC logic. The proposed algorithm converts high-level motion commands into structured OSC messages and dispatches them to the winch actuators with controlled timing. Each incoming command is first mapped to the appropriate actuator group and control type based on the motor identifier and requested action. The resulting OSC message is appended to a queue, and for A-series motors a reset message is added to ensure consistent behavior. A timed dispatcher then processes the queue sequentially, transmitting each message with a fixed inter-message delay to prevent communication overload and guarantee reliable actuator response. This combined mechanism provides a simple but robust interface between high-level control instructions and low-level actuator execution.

Algorithm 1: OSC-Based Command Mapping

```
\begin{algorithmic}
\State \textbf{Input:} High-level command $c = (cmd, motor, period, speed, target)$
\State \textbf{Output:} OSC messages sent to actuator groups
\State Determine actuator group $g$ from $motor$
\State Map $cmd$ to control type $t$:
\State \hspace{1.2em} $t = i$ for $motor = A_i$ (translation),
\State \hspace{1.2em} $t = 1$ for $motor \in \{B,C\}$,
\State \hspace{1.2em} $t \in \{2,3,4,5\}$ for roll, pitch, circular, or half-effect
\State Enqueue OSC message $(g, t, period, speed, target)$
\If{$motor = A_i$ and $cmd =$ translate}
    \State Enqueue reset message $(g, 9, 0, 0, 0)$
\EndIf
\Statex
\State Define spacing interval $\Delta t$ (e.g., 150 ms)
\While{queue $Q$ is not empty}
    \State $m \gets$ dequeue($Q$)
    \State Send $m$ over OSC to the winch controller
    \State Wait $\Delta t$
\EndWhile
\end \end{algorithm}
```

Figure 9 shows the live performance of Eternal Dawn (Le Maillon Theater, Stromsburg, France). While the system successfully enabled precise synchronization between robotic movements, lighting structures, and musical elements, certain limitations, such as actuator payload constraints and occasional communication latency, posed challenges during highly dynamic scenes. As with all live performances, slight deviations by performers in timing or spatial positioning are inevitable. For intense interaction scenes involving the robot, internal safety features, including collision detection and a backstage emergency stop mechanism, were implemented to ensure safety. Future iterations may benefit from adaptive motion planning algorithms supported by live image processing for improved safety, as well as deeper integration with real-time audio analysis to enhance the responsiveness and creative flexibility of technologically augmented performance environments

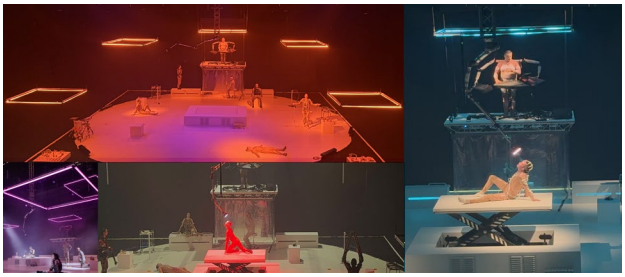


Figure 9. Real performance featuring robotic arm, Motor-Matrix light frames, and performers.

5. CONCLUSIONS

This work illustrates the practical and artistic potential of combining robotic actuation with dynamic light choreography in a live performance setting. Through an integrated design process bridging dramaturgy, mechanical engineering, and real-time control, both the ceiling-mounted robotic arm and the Motor-Matrix light frames proved capable of delivering precise, repeatable, and emotionally resonant motion behaviors. The combination of OSC-based synchronization, PLC-driven actuation, and custom HMIs enabled performers and technicians to shape movement with a high degree of temporal accuracy while maintaining robust safety and reliability during live operation. Dynamic simulations and force-constrained control strategies ensured that mechanical components operated within safe limits, even in demanding choreographic configurations. The resulting systems supported complex interactions between performers, machines, and musical structures, enriching the dramaturgical narrative through responsive spatial and kinetic transformations. While the implementation revealed practical challenges, including payload constraints and occasional communication latency, the modular architecture provides a strong foundation for future refinement. Enhancements such as improved sensing, and deeper audio–motion integration may further expand the expressive potential of hybrid robotic scenography in performance contexts.

Acknowledgments

The project was carried out in the context of interdisciplinary collaboration between TUHH and HFMT University. The technical aspect of the project was funded through the “Innovative Hochschule” (Innovative University) program by the Federal Ministry of Education and Research (BMBF) of Germany (Grant No. 13IHS232C) and the City of Hamburg.

6. REFERENCES

[1] C. Capeto, “Dramaturgy of Robots and Machines,” in Proceedings of EVA London 2020: Electronic Visualisation and the Arts, A. F. Sayers, Ed. BCS, The Chartered Institute for IT, 2020, pp. 167–175.

[2] A.-I. Someșan, “Transhumanism in Performance Art,” Journal of Intercultural Management and Ethics, vol. 5, no. 4, pp. 51–57, 2022.

[3] M. Donnarumma, “Beyond the Cyborg: Performance, Attunement and Autonomous Computation,” Int. Journal of Performance Arts and Digital Media, vol. 13, no. 2, pp. 105–119, 2017.

[4] M. Shildrick, “Visceral Prostheses and Posthuman Embodiment,” in Bodies of Technology, N. Smelik and L. Koolen, Eds. Amsterdam University Press, 2020, pp. 27–42.

[5] Y. Sone and R. Savery, Eds., Cultural Technologies: Robots and Artificial Intelligence in the Performing Arts. Routledge, 2025.

[6] M. Buhaiciuc, “When Human Emotion Meets Technology: A Study of Machover’s Death and the Powers, The Robot Opera,” Bulletin of the Transilvania University of Brașov, Series VIII: Performing Arts, vol. 6(55), no. 2, pp. 43–48, 2013.

[7] J. Parker-Starbuck, Cyborg Theatre: Corporeal Technological Intersections in Multimedia Performance. Palgrave Macmillan, 2011.

The Magic Alchemical Drum Set: a transducer-driven light-up drum set using timbres and scales derived from sonified chemical element spectra

Walker Smith
Stanford University CCRMA
walksmit@ccrma.stanford.edu

ABSTRACT

The Magic Alchemical Drum Set is an interactive audiovisual instrument that integrates three lines of preliminary research: (1) the construction of element-specific timbres using sonified spectral data and perceptually motivated transformations, (2) the design of unequal-tempered microtonal scales derived from elemental spectra and implemented on a LumaTone keyboard, and (3) a transducer-driven drum set that physically couples these sounds to acoustic percussion instruments and synchronized lighting.

Together, these components form a system that transforms static spectroscopic data into a playable, performative instrument emphasizing tactile interaction and audiovisual correspondence. The paper provides a brief overview of related work, outlines the design considerations underlying the scales and timbres, and documents the construction and use of the drum set in both compositional and interactive installation contexts, including feedback from participants. A detailed demo video is provided along with all necessary code. Conclusions and future work in the scale and timbre design, as well as interactive audiovisual instrument design, are presented.

1. INTRODUCTION AND BACKGROUND

This work extends prior work in the Interactive Musical Periodic Table (iMPT)[1][2], which sonifies the spectra of elements obtained from high-resolution spectral data sets[3] [4][5]. The visible emission spectra act as unique "fingerprints" for each element, and are sonified by additive synthesis; the frequencies of visible light are divided by 10^{12} to move them into the audible range, and the intensity of each line is mapped to the amplitude of the corresponding sine wave, creating a unique sinusoidal cluster for each element. While prior work has explored musically interesting and perceptually relevant transformations of these spectra (see Section 2.1), the current interaction paradigm in the iMPT is limited. Users can select an element to see its spectrum and hear a sonification of it, and they can adjust parameters to change the timbre. However, this exists entirely on a computer screen and operates more as an auditory display of scientific phenomena than a playable musical instrument.

Sonifications of molecular and quantum simulations have been developed with more interesting interaction paradigms, including playable "molecular musical instruments" created from interactive sonifications and visualizations of molecular activity in virtual reality[6], and interactive sonifications of quantum phenomena in which playing notes on a MIDI keyboard controls sonification and visualization parameters[7].

In contrast to dynamic datasets like molecular simulations or time-evolving quantum systems, elemental spectra are static. The goal of this work is to transform these static spectra into sounds with more temporal dynamism, to create a more interactive and embodied playing experience. This work presents additional timbral developments leading to percussive timbres derived from element spectra, design of microtonal scales derived from element spectra and programming on a LumaTone keyboard, and construction of a transducer-driven drum set to further bring these sounds to life by using them to excite real-world objects.

Copyright: ©2026 First Author et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License 3.0 Unported, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

2. TIMBRES FROM CHEMICAL ELEMENT SPECTRA SONIFICATIONS

2.1 Prior Work: Overtone Stretching

Prior work[2] introduced *overtone stretching* as a method for expanding sonifications of visible element emission spectra into multi-octave timbres. Because the visible spectrum spans approximately one octave, direct sonification by frequency transposition (dividing optical frequencies by 10^{12}) produces collections of sinusoids confined to a single octave. Overtone stretching redistributes these pitch classes across multiple octaves by selectively moving low-amplitude tones into higher octaves.

Each spectral component with frequency f_i and normalized amplitude A_i is assigned a positive integer octave displacement H determined by a user-defined octave-cut parameter h (Eqn. 1). The new overtone-stretched frequency is calculated in Eqn. 2. The resulting waveform is synthesized additively using the stretched frequencies f_k and the original amplitudes $A_i = A_k$ in Equation 3.

$$H = \lfloor \log_h A_i \rfloor \quad (1) \qquad f_k = f_i \times 2^H \quad (2)$$

$$S = \sum_{k=0}^{N-1} A_k \cos(2\pi f_k \frac{r}{r_o} t) \quad (3)$$

2.2 Polyphonic Element Synthesizers from Overtone-Stretched Spectra

In Equation 3, all frequencies f_k are subsequently scaled to a reference pitch using the ratio r/r_0 of the current fre-

quency r to a reference frequency r_0, allowing the timbre to be pitch-shifted corresponding to the frequency of a user-input MIDI note r. This is implemented in a polyphonic element synthesizer Max patch (see Section 7). These frequencies r and r_0 could come from 12TET, but they can also be from tuning systems derived from the element spectra themselves as discussed in section 3.

The user-defined octave cut factor, h, is mapped to the Mod Wheel of the LumaTone, allowing easy timbral modification (see Demo Video, section 6). While overtone stretching enables significant timbral variation, the resulting sounds still remain temporally static due to constant-amplitude sinusoids. The following section extends this approach by introducing amplitude-dependent temporal decay, producing more instrument-like and percussive behavior.

2.3 Modal Synthesis with Resonant Filter Banks

A temporal envelope is achieved using damped sinusoids rather than simple additive synthesis. Specifically, a resonant filter bank (the Max/MSP *resonators* external[8]) creates decaying sinusoids when prompted by an impulse. The exponential decay rate for each frequency f_k is derived from its amplitude A_k as shown in Eqn. 4, such that louder tones (corresponding in turn to brighter spectral lines) produce longer decays, and quieter tones (dimmer lines) produce more rapid decays. The overall decay rate for all partials can be adjusted with a user-defined decay-rate multiplier, τ_0, which is mapped to the Pitch Wheel (leftmost) of the LumaTone.

From an instrument design perspective, this synthesis method produces much more interesting temporal envelopes. When the octave stretch and temporal decay are combined, the resultant sound consists of loud, slowly-decaying low frequencies and quieter, more quickly decaying higher frequencies. This mirrors the character of real instruments and physical models such as the Karplus-Strong plucked string algorithm[9]. From a sonification perspective, mapping line intensity to decay rate also connects conceptually to spectroscopy, as the intensity of the spectral lines is related to its emission lifetime[10].

$$S(t) = \sum_{k=0}^{N-1} A_k e^{-t/(\tau_0 A_k)} \cos(2\pi f_k \frac{r}{r_o} t) \qquad (4)$$

The rich inharmonic spectra of many elements yield very percussive timbres with this synthesis technique. Furthermore, different fundamental frequencies ($\frac{r}{r_0}$) sound like different percussion instruments. Playing oxygen's timbre with hydrogen's scale on the LumaTone, the lower pitches resemble kick drums, mid-ranges sound like toms, and higher pitches sound like metal percussion–an entire percussion library just from one sonified element! (See Demo Video, section 6). The percussive quality inspired the use of literal percussion and transducers in Section 4.

3. MICROTONAL SCALES FROM ELEMENT SPECTRA

In addition to element-derived timbres, unequal-tempered microtonal scales were also derived from select spectra and programmed on a LumaTone keyboard. The LumaTone is the ideal instrument for this, as one can easily program custom microtonal scales and customize the color of individual keys.

Hydrogen and helium were chosen as case studies due to their relative simplicity. Hydrogen's spectrum contains seven prominent spectral lines, three of which lie within only a few cents of one another. Including all such closely spaced tones would result in perceptually redundant pitches, so tones separated by only a few cents were excluded, yielding a compact and playable scale (Fig. 1).

Helium presents a denser case, with 31 spectral lines within the visible range. Informal listening tests indicated that removing tones below 10% of the maximum amplitude did not substantially alter the perceived tonal character of the spectrum, as they were largely masked by stronger partials within the same critical bands [11]. In contrast, removing tones above this amplitude threshold noticeably altered the tonal character, suggesting 10% as a perceptually meaningful cutoff. Applying this criterion yielded a reduced 10-note pitch collection, from which closely spaced tones were again removed, resulting in a seven-note scale (Fig. 1). These design choices prioritize perceptual distinctness and playability; alternative scales could be derived by adopting different thresholds or retaining more closely spaced tones.

Two keyboard layouts were implemented on the LumaTone for both scales. In the *guitar layout*, pitches ascend diagonally across the keyboard like frets on a guitar string, while the *piano layout* alternates pitches in a manner analogous to a piano's white and black keys (Fig. 1).

LumaTone layout files, along with corresponding scala and tuning files for use in Max and any DAW, are provided (Section 7).

4. TRANSDUCER-DRIVEN LIGHT-UP DRUM SET

The concept of *element percussion* was extended to acoustic instruments using transducers attached directly to a drum set. Dayton Audio exciters were mounted on selected drums using Gaffer's tape, allowing synthesized element timbres to physically excite the toms, snare, and cymbal.

Each synthesized timbre was routed to a dedicated audio channel and amplifier driving a corresponding exciter. Different regions of the LumaTone keyboard control different drums, spanning from low-frequency instruments (kick and floor tom) to higher-frequency instruments (snare and cymbal). The kick drum was simulated by a small speaker due to the lack of a sufficiently powerful exciter. [1]

The polyphonic element synthesizers were implemented as standalone Max patches, and also as Max for Live devices within Ableton Live, the latter enabling the integration of pre-composed material with live performance on the drum set.

Together, the transducer-driven drums transform the synthesized element timbres into physically resonant, percussive instruments, reinforcing the tactile and embodied qualities of the system.

[1] A detailed tech rider documenting construction of the drum set is available: https://drive.google.com/file/d/1oSrhwUZD-2Xr-gYX_hwcNupHdmn6opCS/view?usp=sharing

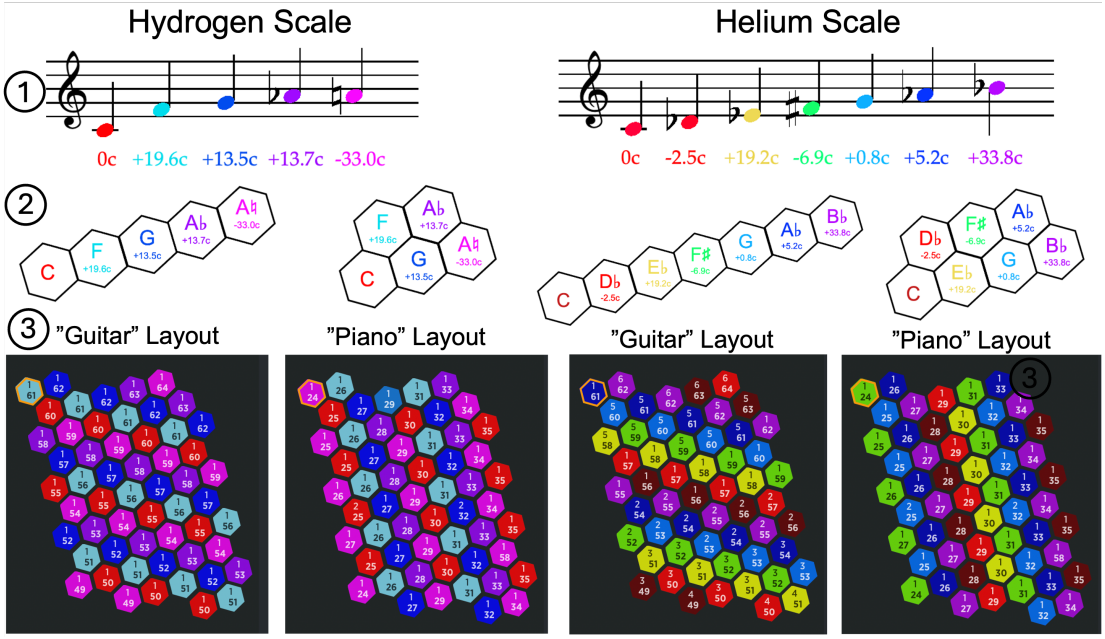


Figure 1. (1) Scales created from the visible spectra of hydrogen and helium, with cent deviations from the notated pitch indicated below each note. (2) Diagrams showing the "guitar" and "piano" layouts. (3) The corresponding guitar and piano layouts in the LumaTone Editor app.

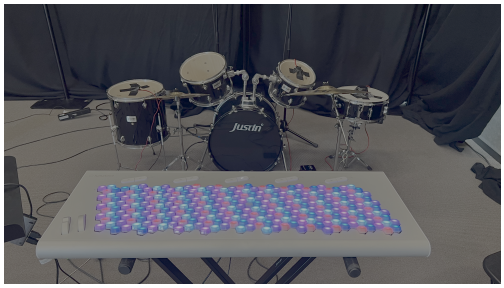


Figure 2. The Magic Alchemical Drum Set (background) with LumaTone keyboard programmed to the scale of hydrogen (foreground). Exciters are placed on select drums and lights underneath the drums.

4.1 Lights and Audiovisual Interaction

LED lights were placed under the drums to extend the color–pitch associations already present in the LumaTone interface. Pressing a blue key in the "High Tom" section of the keyboard not only triggers the corresponding pitch on the tom, but also lights up that drum head in vibrant blue. This supports intuitive exploration of the relationship between color and pitch, while also functioning as a playful and engaging audiovisual instrument.

Lighting is controlled wirelessly using a DMXKing USB–DMX interface and a Di-Fi transmitter. Different regions of the LumaTone trigger lights on different drums, with routing configurable to accommodate variations in drum materials and visibility. For instance, one setup used a black kick drum through which the light is not visible. Instead, the sound of the kick drum triggers light output on the next-lowest-pitched drum (the low tom). When a white or translucent kick drum head is used, the light can be routed directly to the kick drum itself. This flexible routing allows the audiovisual mappings to be adapted to different physical setups while preserving a consistent interaction logic.

5. COMPOSITION AND INSTALLATION

A composition was created in which the Magic Drum Set functions as an autonomous performer (see Section 6.1). While the scale is derived from hydrogen and the timbres from oxygen, the rhythms and formal structure of the piece were composed intuitively with no direct relationship to the underlying spectral data. This demonstrates that the system operates not only as a sonification tool, but as a general-purpose musical instrument. The composition also plays with expectations of audiovisual synchronization, introducing counterpoint between light and sound through extended passages where some drums are illuminated rhythmically while not creating any sound.

The system was also presented as an interactive installation, allowing participants to select elements and explore different timbral palettes. Informal feedback suggests that the layout of the LumaTone and interaction paradigm are intuitive, while the specific mappings (i.e. which sections control which drums, the fact that different lights also create different pitches) reveal themselves gradually, encouraging exploration and sustained engagement.

Many participants reported that the simple delight in audiovisual correspondence (i.e., pressing a red button and seeing a red light) along with the timbral variety (resulting from changing elements, or simply exploring the immense sonic possibilities of a single element), made it a tool they felt they could play with for hours. Others reported that the interaction made it feel as though they were actually playing the drums.

Originally conceived as an educational sonification of chemical spectra, the project has evolved through multiple layers of translation into a form of audiovisual instrument design. While these transformations obscure the source data, the connection remains generative rather than didactic—producing an experience that is inviting without requiring prior scientific knowledge. In this sense, the

project may be moving away from musical chemistry and toward music alchemy. Nevertheless, this interactive exploration may be able to serve an educational purpose, and future work aims to explore how these interactive explorations can support intuitive understanding of timbral differences between element spectra sonifications.

6. DEMO VIDEOS

The included demo video (https://www.youtube.com/watch?v=rk9A0fAYQNg) presents Element Scales in section 1; section 2 presents the polyphonic synthesizer with element timbres from in sections 2.2 and 2.3, and section 3 illustrates the basic functionality of the Magic Alchemical Drum Set, along with clips of people interacting with the installation.

6.1 Additional Demo Videos

A more extensive demo video showing people interacting with the drum set installation is included in the separate submission for the Installation Track (Submission Number 695), and also available here: https://www.youtube.com/watch?v=kkC_YeyNW8o. The video of the composition created using the drum set as an autonomous instrument is included here: https://www.youtube.com/watch?v=ZQIbQqP2I0c.

7. CODE

All code (including Max patches, Max4Live instruments, tuning files, and LumaTone Editor files) is available on Github: https://github.com/anonymousrhinocerous800-eng/ICMC-2026-Innovation-Showcase---Code.git.

8. CONCLUSION AND FUTURE WORK

The Magic Alchemical Drum set presents a culmination of preliminary findings in three major areas: scale design from spectra of chemical elements, timbre design from elements, and exciter-driven drum sets. This work represents an exploration at the intersection of sonification, scale and timbre design, and audiovisual instrument design, with all areas ripe for further research. Regarding element timbres, the use of more complex attack envelopes–even perhaps unique attack envelopes for each partial–could add even more dynamism to the sounds. The envelopes could be derived from an additional aspect of the spectral data. As more scales are designed, it would also be interesting to implement research on complementary tuning systems and timbres[12]. The drum set can continue to be a valuable tool for composition and interactive installations, and further feedback from participants will help to refine the interaction design.

Acknowledgments

Special thanks to Prof. Dr. Georg Hajdu for his mentorship throughout this project. This work was supported by the Graduate Research & Internship Program in Germany (GRIP) grant from Stanford University, and was completed in part at Stanford University CCRMA and in part at the Ligeti Zentrum in Hamburg, Germany.

9. REFERENCES

[1] W. Smith, "Interactive Musical Periodic Table: Sonification of Visible Element Emission Spectra," in *Proceedings of the 21st Sound and Music Computing Conference (SMC)*, G. Campos, I. Salselas, J. Vieira, M. Conceição, N. Fonseca, R. Penha, Ângela da Ponte, and D. Andrikopoulos, Eds. Porto, Portugal: SMC Network, July 2024. [Online]. Available: https://smcnetwork.org/smc2024/papers/SMC2024_paper_id192.pdf

[2] W. Smith and M. Bosi, "Sonification Strategies for Mapping Visible Element Emission Spectra to Perceptually Relevant Sounds While Retaining Tonal Information," in *Proceedings of the 20th International Audio Mostly Conference*, ser. AM '25. New York, NY, USA: Association for Computing Machinery, 2025, p. 223–235. [Online]. Available: https://doi.org/10.1145/3771594.3771617

[3] NIST, "Atomic Spectra Database," https://www.nist.gov/pml/atomic-spectra-database, accessed May 2025.

[4] F. M. P. III, *MIT Wavelength Tables, Second Edition*. Cambridge, MA: MIT Press, November 1982.

[5] J. G., "Visible Spectra of the Elements," https://www.atomic-spectra.net/, n.d., accessed: 2025-02-24.

[6] T. J. Mitchell, A. J. Jones, M. B. O'Connor, M. D. Wonnacott, D. R. Glowacki, and J. Hyde, "Towards molecular musical instruments: interactive sonifications of 17-alanine, graphene and carbon nanotubes," in *Proceedings of the 15th International Audio Mostly Conference*, ser. AM '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 214–221. [Online]. Available: https://doi.org/10.1145/3411109.3411143

[7] J. L. Müller, A. Freye, and T. Hermann, "Interactive Sonification of 2D Quantum Systems," in *Proceedings of the 30th International Conference on Auditory Display (ICAD 2025)*, Coimbra, 2025.

[8] T. Jehan, A. Freed, and R. Dudas, "Musical Applications of New Filter Extensions to Max/MSP," in *Proceedings of the International Computer Music Conference (ICMC)*, Beijing, China, 1999, pp. 504–507.

[9] K. Karplus and A. Strong, "Digital Synthesis of Plucked-String and Drum Timbres," *Computer Music Journal*, vol. 7, no. 2, pp. 43–55, 1983. [Online]. Available: http://www.jstor.org/stable/3680062

[10] National Institute of Standards and Technology, "Atomic Spectra Database Lines Help: Relative Intensities," https://physics.nist.gov/PhysRefData/ASD/Html/lineshelp.html, 2025, accessed January 2, 2026.

[11] E. Zwicker and H. Fastl, *Psychoacoustics: Facts and Models*, 1st ed. Berlin Heidelberg: Springer-Verlag, 1990.

[12] W. A. Sethares, *Tuning, Timbre, Spectrum, Scale*, 2nd ed. Springer, 2005.