Non-invasive Gesture Sensing, Physical Modeling, Machine Learning and Acoustic Actuation for Pitched Percussion

by

Shawn Trail
Bachelor Arts, Bellarmine University, 2002
Master of Music., Purchase College Conservatory of Music, 2008

A Dissertation Submitted in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

In Interdisciplinary Studies: Computer Science and Music

Non-invasive Gesture Sensing, Physical Modeling,
Machine Learning and Acoustic Actuation for Pitched Percussion

by

Shawn Trail
Bachelor Arts, Bellarmine University, 2002
Master of Music., Purchase College Conservatory of Music, 2008

Supervisory Committee

Dr. Peter F. Driessen, Co-Supervisor
(Electrical Engineering)

Dr. W. Andrew Schloss, Co-Supervisor
(Music)

Dr. George Tzanetakis, Co-Supervisor
(Computer Science)

Supervisory Committee

Dr. Peter F. Driessen, Co-Supervisor
(Electrical Engineering)

Dr. W. Andrew Schloss, Co-Supervisor
(Music)

Dr. George Tzanetakis, Co-Supervisor
(Computer Science)

ABSTRACT

This thesis explores the design and development of digitally extended, electro-acoustic (EA) pitched percussion instruments, and their use in novel, multi-media performance contexts. The proposed techniques address the lack of expressivity in existing EA pitched percussion systems. The research is interdisciplinary in nature, combining Computer Science and Music to form a type of musical human-computer interaction (HCI) in which novel playing techniques are integrated in performances. Supporting areas include Electrical Engineering- design of custom hardware circuits/DSP; and Mechanical Engineering- design/fabrication of new instruments. The contributions can be grouped into three major themes: 1) non-invasive gesture recognition using sensors and machine learning, 2) acoustically-excited physical models, 3) timbre-recognition software used to trigger idiomatic acoustic actuation. In addition to pitched percussion, which is the main focus of the thesis, application of these ideas to other music contexts is also discussed.

Contents

List of Tables

List of Figures

ACKNOWLEDGEMENTS

This work is the culmination of my life experiences and a compilation of what inspires me in the world merged into a singular project. Where possible I have tried to give direct attribution, but due to the vast network of amazing artists, engineers, musicians, designers, bands, teachers, philosophers, writers and generally topically interesting people that have been my muse or lent me support over the years not all due credit is listed. However, this is my heartfelt thank you to the anonymous group of individuals I have crossed paths with leading to this point of accomplishment in my life. Here's to paying it forward, the way you did for me. More specifically, this work would not have been possible without the direct input and influence of specific people in my life. While this list is not comprehensive it reflects a certain group of people that this project owes a debt to.

First and foremost I thank my collaborators: Michael Dean, Dan Godlovitch, Stephen Harrison, Leonardo Jenkins, Thor Kell, Duncan MacConnell, Steven Ness, Wendy Nie, Gabrielle Odowichuk, Jeff Snyder, and Tiago Fernandes Tavares. This work is the result of true interdisciplinary collaboration. In nearly all facets, this work would not have been possible without this multidisciplinary team of engineers and artists. I firmly believe in the adage that "...the whole is greater than the sum of its parts...", in this case I would not have been able to realize any one aspect of this work on my own at the rate and in the way we did. I learned so much working with this team and will always look on my time at MISTIC as a special period where nearly anything we could dream up were possible.

Secondly I thank my mentors that gave me opportunities or shared their knowledge with me: Niels Boettcher, Richard Burchard, Mantis Cheng, Sofia Dahl, Smilen Dimitrov, Peter Denenberg, Suzanne Farrin, Andy Farnell, Steven Gelineck, Lorel Lashley, Payton MacDonald, Andrew McPherson, Valerie Naranjo, Ben Neill, Dan Overholt, Wyatt Page, Kevin Patton, Stefania Serafin, Eric Singer, and Du Yun.

I would be remiss if I did not thank specific family or friends that aided in invisible ways to this project, such as lending me a place to sleep, giving me rides to the airport, loaning me money, playing me a new record for some much needed sonic respite, or just literally offering a shoulder to lean on in hard times: John Cason, Robert DiNinno, Barbara Duncan, Maggie Gardiner, Jason Greer, Lee Gutterman, Matt Langley, Calum and Fergus MacConnell, Julian Marrs, Jon Miles, Phil Moffa, Greg Morris, Kaitlin Paul, Jeremy Sherrer, Barbara Sizemore and Tyler Trotter.

To my teachers in Ghana where I learned what it means to be an innovator while being accountable to your community: Francis Kofi Akotua, Christopher Ametefe, William Anku, John Collins, SK Kakraba, Kakraba Lobi, and Aaron Bebe Sukura.

Thank you to individuals/organizations Don Buchla, Perry Cook, Adrian Freed, Randy Jones, Ajay Kapur, Max Matthews, Robert Moog, Miller Puckette, Nandini Ranganathan, Trimpin, Adam Tindale, Michel Waisvisz, David Wessel, Ableton, Cycling 74, ICMA, NIME, SMC, STEIM, EMMI, Pretty Good Not Bad, Ministry of Casual Living, Good Party, Gary Cassettes and Auralgami Sounds.

I must thank several musical influences that directly inform this project: Freak Heat Waves for their inexhaustible dedication to their craft and artistic vision, Fela Kuti for his commitment to social rights activism, Jaga Jazzist for their seamless combination of jazz and electronics, Bjork for her insatiable sense of sonic exploration, Konono No.1 for the vast unknown of African electro-acoustic music they represent, Sun Ra for the vision that comes only through sound, Pat Metheny for his professionalism and ability to accomplish the impossible, and lastly DUB- arguably the first bona fide form of cultural electronic music. This work sought to directly marry the techniques used to produce DUB at the mixing-desk with acoustic instruments into one intuitive practice. "DUB is DUB" Mad Professor (MOOGFEST 16).

Supervisory Committee My supervisory committee is an amazing group of pioneering multidisciplinary musicians, engineers, and computer scientists. This journey has taught me an infinite amount about music, computing and research and will be the foundation I go forward on. Infinite gratitude for the support and guidance, especially through the difficult moments and for inviting me in first place to pursue this work. It has been a true labor of love, something I feel is essential and without your mentoring this project could not have been what it is: something I am incredibly proud of and perfectly true to and beyond the vision. Thank you for supporting this research. I brought an idea to you. You saw its potential and how it fit with your existing research, helped me procure funding and it was made far richer from each of your unique and individual perspectives.

Funding Sources Thank you to: Ord and Linda Anderson Interdisciplinary Graduate Fellowship, Social Sciences and Humanities Research Council of Canada, Natural Sciences and Engineering Research Council of Canada, and Fulbright.

DEDICATION

This dissertation is dedicated in two parts- first to my mother and father, Carol and Bennie Trail. My parents have supported me through thick-and-thin my entire life and have shown me, through action, what perseverance and hard work can accomplish. Their optimism and love of life are unparalleled and their mutual work ethics and dedication to excellence are an inspiration. I grew up watching my mother sing in church with joy and my father and I took combo-organ lessons together starting when I was 5. Both experiences are key to who I have become as a musician. We always had a computer in the house so I got started early on my Commodore 64. My father was a teacher and school principle so I was always in a learning environment feeding my curiosity. I have fond memories of going to work with my father and playing Oregon Trail in his school library all day while he worked not knowing that those early computer sounds were forming the initial framework for the soundtrack to my life. My parents supported every musical interest I ever had, buying me instruments and gear, paying for lessons, hosting the jam sessions, listening to my crazy ideas, giving me a little spending money to go record shopping, taking me to concerts, letting me play shows on school nights, coming to my recitals, and giving me the encouragement when the going was rough. My mother and father are from Appalachia and had to work hard to have the lifestyle they wanted. My mother didn't get to go to college, even though she was awarded a full-academic scholarship- she had to work from an early age to help with family responsibilities. When I got my Fulbright, she had been laid off of her job of 30+ years and my father went to work on the night shift at the local grocery store to make ends meet showing me the true meaning of necessity, hard work and responsibility. Completing this Phd was the single most difficult task of my life and is only a small reflection of what my parents have accomplished. They are my true heroes.

The second half of this work is dedicated to my daughters Alma and Isla. Nearly everything I do is for you. I hope this work serves as inspiration for you to seek collaboration through diversity, help your fellow, believe in yourself, follow your dreams, set your goals high, remain impervious to discouragement, never settle, be passionate about your work, never stop learning, and to persevere through the difficult times.

FORWARD

In Ghana the xylophone player will often design and build their own instruments specific to their own needs. The instrument is also central to their society. The computer is central to our society and also now our most ubiquitous folk music instrument. Now more than ever we have the ability, using modern physical computing techniques, to realize our own electronic music instruments. This body of work is meant to serve as a framework for artists and educators that want to design their own solutions working with computers in context specific, musically appropriate ways not reflected in the consumer market. We need to be the designers of our instruments like in Ghana- or technology will consume us- products will no longer reflect the needs of the artists and entire disciplines will be threatened. Music, and more generally sound, are the primary mediums that I experience the world through, as it is for many sound artists. Sound has been an emancipatory medium for me, giving my life personal meaning and trajectory. Access to information and resources are the biggest hurdles to pursuing music technology vocationally. Also, the skills learned through music technology education can be a vehicle for self-empowerment leading to pathways in engineering and the sciences. Where possible I have sought to develop low-cost solutions to the technological hurdles presented here-in in hopes that the work can reach communities that previously would have had no means in which to practice electronic music. This work has also been, and will continue to be adapted and applied pedagogically in efforts to formalize a framework for Digital Music Instrument design as a standard to general music education and computer literacy across economic, linguistic, demographic, cultural, and/or gender barriers.

"...social experiences and productions of sound and audition...may inform emancipatory practices...sound works to unsettle and exceed arenas of visibility by relating us to the unseen...particular tonalities and musics, silences, and noises may transgress certain partitions or borders, expanding the agentive possibilities of the uncounted and the underheard...to embolden the voices of the few...within a multiplicity of territories and languages."- Sonic Agency: Sound and Emergent Forms of Resistance. Brandon LaBelle.

Part I

INTRODUCTION

Chapter 1

Overview

> ”...noise is triumphant and reigns sovereign over the sensibility of men. This evolution of music is comparable to the multiplication of machines, which everywhere collaborate with man...creating such a variety and contention of noises that pure sound in its slightness and monotony no longer provokes emotion. We must...conquer the infinite variety of noise sounds... combining in our thoughts the noises of trams, of automobile engines, of carriages and brawling crowds...the practical difficulties involved in the construction of these instruments are not serious...thus every instrument will have to offer...extended range...” The Art of Noises: Futurist Manifesto, Luigi Russolo. ca 1913

1.1 Outline

The focus of this research is to develop new interactive digital systems for acoustic pitched percussion based multi-media performance. The goal is to present a flexible set of ideas and proof-of-concept prototypes to pitched percussionists, composers, and sonic interaction designers rather than providing a specific integrated implementation or detailed technical guidelines. A fundamental goal has been to develop intuitive interfaces that augment traditional instrumental techniques without interrupting traditional performance modes. This general approach is referred to as non-invasive gesture sensing. Harnessing the broad variety of gestures available to the expert, the interfaces afford the performer the ability to network acoustic instruments with computers. Many new interfaces target users with limited instrument playing abilities while others target experts more.

This work offers insight into the theory and practice of Digital Music Instruments (DMI) while delivering novel forms of HCI, MIR/machine learning, physical modeling synthesis, gesture recognition, and acoustic actuation appropriate for pitched percussion instruments.

This dissertation presents studies in three areas: (i) electro-acoustic pitched percussion instrumental performance contexts; (ii) digital augmentation and interfacing the computer; (iii) acoustic actuation (musical robotics) via the idiomatic pitched percussion interfaces. It is organized in three parts:

- Part I ”INTRODUCTION” presents a short background and description of the topic (pitched percussion context and scope), the objectives and aims (contemporary considerations and problems for the instrument), an overview of some of the related work that has been done by others (history of electro-acoustic pitched percussion), and the introduction to the twelve related peer-reviewed publications written while conducting the research described in this dissertation.

- Part II ”IDIOMATIC INTERFACES, MACHINE LEARNING AND ACTUATION” focuses on the thematic contents of the associated publications providing an overview of aspects of design and deployment, experimental results, system descriptions and contributions to the field: alternative interfaces, custom DSP, idiomatic actuation and applications of the proposed techniques applied to contexts beyond pitched percussion.

- Part III ”CONCLUSION” discusses final thoughts and takes a look at possible future directions for pitched-percussion hyper-instruments.

1.2 Background

This research presents the design and development of digital interactivity for the acoustic pitched percussion instruments. The main idea is to acquire and analyze performance gestures used to play traditional and conventional concert instruments. The extracted information is used for real-time control to provide extended digital capabilities to these instruments. This is in order to effectively bridge musical human gestures and complex computer interaction methods in a musically appropriate manner for the particular instrument group under consideration (Musical-HCI). In many cases, my research studying traditional instruments/techniques in Ghana has informed my approach to sound design. Extended techniques and physical modifications of specific traditional instruments result in modulation of the overall sound [27]. For instance, how one holds a Likembe will dictate the amount of buzz produced by the tines rings. I model this using position sensors and digital filters and/or control data applied to the direct audio signal or a sequencer of the instrument in question. As playing position changes, so do corresponding parameters of the digital processing on the signal. Where possible I have tried to directly couple traditional acoustical sound design principles with the digital counterpart prototyped. This is an intentional grounding of novel technologies designed for sonic interaction. Here the goal is linking traditional analog sonic interaction design techniques, which are embedded with cultural meaning, to novel digital interactive sound design methods. I have chosen the conventional concert marimba as the primary model for this work. The vibraphone, Gyil and Likembe are also augmented and explored in this thesis.

There are important challenges when incorporating pitched percussion in contemporary settings. The contemporary marimbist is largely limited to the orchestra or as a soloist in the classical concert hall. Some times when appearing in chamber groups or electro-acoustic music compositions in alternative venues amplification using diaphragm microphones is utilized but amplification tends to be rare.

Early proponents of the electrified marimba include Mickey Heart (Grateful Dead) and Ruth Underwood (Frank Zappa). The vibraphone, primarily appearing in rock and jazz, is more commonly electrified but this is still rare. The lemellophone family of traditional instruments is some times amplified with pickups, but established traditions are few within this scope. As a whole, electrified pitched percussion is a largely undocumented field.

There are some limited commercial xylophone style MIDI controllers, but no lamellophone based instruments, outside of prototypes and an obscure patent [72]. This thesis seeks to address these concerns by introducing a collection of tools, implementations, new instruments, and custom software and hardware innovations that augment and expand the existing electronic pitched percussion performance canon.

The ideas and techniques presented in this thesis can form the foundation of an integrated framework and some steps have been taken in this direction. Potential users should approach this set of innovations modularly and apply aspects described to their own practice as they see fit. The novel techniques developed can be utilized pedagogically and are meant to serve as a resource for composers seeking new sound palettes and modes of interaction between performer and computer. Equally, this research is meant for designers and engineers to use as a point of reference for the musical sensitivities informing the fabrication of the prototypes developed herein. Largely speaking, commercial music technology solutions are mass marketed to a broad user. My contributions are to be viewed as an open system for the rapid-prototyping of non-invasive gesture sensing, signal processing, and acoustic actuation specific to expert-level EA pitched percussion performance. Some of these ideas have also been applied to other instrumental contexts and described in more detail later in the thesis.

My overarching artistic goals for this research are to compose and perform new musical works for a Modern Orchestra of both conventional instruments and Hyperinstruments (acoustic instruments enhanced with sensors, gesture sensors, and robotic actuators), while formalizing their sonic possibilities and performance contexts. MISTIC's previous research on new DMI's concentrated on innovations in virtual computer-sound controllers like the Radiodrum, the Soundplane, and enhanced electro-acoustic percussion instruments like the EDrumset. In these scenarios sound generation was either oriented towards electronics (synthesized instruments played through speakers) or acoustics (various drums struck by a robotic mechanism). This body of work expands the ensemble to now include pitched percussion, focusing on the Western marimba, vibraphone, Gyil and lamellophones. This work is inspired by: DUB (King Tubby, Scientist), Hip Hop (J Dilla, Prince Paul) and Jazz (Miles Davis, Herbie Hancock); 20th Century Minimalism (Steve Reich, Manuel Gottshing); West and South African xylophone and drumming traditions; the Indonesian Gamelan; and various lamellophone cultures from West, Central, and Southern Africa- including Mbira, Kalimba, and Likembe. Additionally, experimental electronic and rock music pioneers such as Kraftwerk and Tortoise directly inform this work. Equally, this research is indebted to the international league of computer music researchers, engineers, and artists that view the future through a steady, curious and optimistic scope. No tradition was ever static or will ever be.

A further area of development is in robotic musical instruments controlled by either hyperinstruments or other gesture sensors [31]. At MISTIC, we refer to the evolution of new musical instruments in terms of both the sound generator and the controller. Traditionally, these separate functions are aspects of one physical system; for example, the violin makes sound via vibrations of the violin body, transmitted via the bridge from the strings, which have been excited by the bow or the finger. The artistry of the violin consists of controlling all aspects of the strings vibrations [151]. The piano is a more complex machine in which the player does not directly touch the sound generating aspect (a hammer hitting the string), but still the piano has a unified construction in which the controller is the keyboard, and is directly linked to the sound generation. For hyperinstruments, we decouple these two aspects, allowing for the controller to have an effect that is either tightly linked to the sound produced (as any conventional acoustic instrument has to be) or can be mapped to arbitrary sounds. There is a long history of mechanically controlled musical instruments that produce acoustic sound. More recently a variety of such instruments that can be digitally controlled have been built. The term robotic music instruments is frequently used to describe them- here we refer to them as actuated acoustic instruments. At their most basic form they are simply devices that respond to control messages the composer/performer generates. Robotic musicianship refers to a more sophisticated instance where the control of the robotic instrument is informed by programming it to autonomously "listen" and respond to the musical context. For example a robotic percussion instrument might use real-time beat tracking to accompany a live musician making expressive tempo changes or a robotic piano might use chord detection to determine what key to play in. An example context for this work is Jazz Guitarist, Pat Methenys (17-time Grammy winner) Orchestrion Project[86]. Metheny performs solo accompanied by an entire stage full of robotic instruments. In the fall of 2010 I was the Robotics and Control Interface Technician on the World-tour. The Orchestrion has provided an invaluable experience observing what does and doesn't work in a rigorous performance context using experimental interfaces and DMI's. As it takes many years to realize and master any instrument, an extended instrument combines musical mastery with a technical requirement that nearly doubles the proficiency overhead of the artist. As such, towards developing an open framework for pitched percussionists, my goal has been to design and develop the technology utilized in my own music as opposed to using commercial solutions.

1.3 Motivation for this work

The conventional modern marimba is marginalized in contemporary music due to limited portability, amplification limitations, its antiquated image and limited performance contexts. The acoustic xylophone does not lend itself to ready amplification in a setting where electric instruments are present. Relying on a diaphragm microphone as the sole method for electrifying the xylophone fails due to inherent and unavoidable debilitating feedback issues. This severely limits the instruments range of modern performance contexts. This research seeks to effectively solve this. Beyond simply amplifying the sound, the contributions of this work allow for extensive novel signal processing and human-computer interaction possibilities using various custom designed gesture sensing platforms and signal analysis using music information retrieval (MIR) .

As a particular example consider an enhanced African xylophone described later in the thesis. The developed instrument has been equipped with a custom direct analog audio pickup system used concurrently with a custom gesture sensor system and software that analyzes, processes and converts the audio signal and specified gestures into control data that can then be re-routed and mapped to MIDI, OSC, or potentially other types of networking protocol- vastly expanding the instruments sonic potential and digital interactivity. The custom gesture-sensing interfaces allow for interaction with computer parameters via inherent performance techniques of the instrument, preventing performance interruption typically occurring from engaging with a knob/fader in a conventional electro-acoustic setting. In doing so the instrument is modular and compatible with virtually all MIDI hardware/software platforms, so that any user could develop their own custom hardware/software interface with relative ease and vast potential. Additionally, the system allows for use in rigorous stage environments where substantial sound reinforcement is prevalent.

Since amplifying and processing the acoustic audio is presumed, the cumbersome resonators underneath the bars are no longer needed. This innovation greatly reduces the frames mass, improving the portability of an otherwise very large, bulky, nearly immobile instrument. The minimized frame is light and strong- resilient enough to withstand the rigor of constant use, yet light enough to manage as a single compact item in transport. The legs are a keyboard stand and the performer can decide to sit or stand: making it easier to engage pedal boards;incorporate gesture sensing with the legs; novel interfaces; or even play percussion with foot pedals. The frame fits in a custom case without requiring the bars to be removed- the pickup system remains mounted and protected in transit. This makes the instrument more portable and manageable for a performance.

Cultural Urgency

The Xylophone is one of our oldest instruments, perhaps the first instance that melody (the voice) and struck rhythm (percussion) were combined to communicate. In cultures such as the Lobi (Northwestern Ghana), the xylophone plays an integral role in the retention and transmission of complex information systems that took centuries to develop. These are pre-digital coding systems intrinsic to the xylophone itself [145]. Modern advancements in consumer based digital technologies are allowing us unprecedented control over sound in ways that make traditional methods seem primitive. However, what the traditional model provides in terms of performance is a context for proven successful socially relevant uses of music. This information is essential to retaining, and evolving essential methods of music production that were developed through explicit cultural impetus over many epochs.

The marimba, presented in this work, will serve as a nexus for the xylophone traditions at large. These xylophone traditions are endangered in nearly all indigenous contexts, largely due (aside from war, genocide, deracination, or natural disaster) to the youth- responsible for the tradition- being drawn to modern production trends popularized by media from the West. In societies where the indigenous xylophone exists, musical innovators typically have little access to experimental electronics, or even knowledge of the history of their use artistically. Lack of resources due to the prohibitive cost factors associated with purchasing equipment, the difficult nature of importing gear, and limited accessibility to relevant recordings or literature perpetuate the disassociation even more. Eradicating the limitations imposed by geography and class can potentially sustain the evolution of specific endangered musical systems by making possible previously inconceivable collaborations between isolated, indigenous cultures. From polyphony, to the piano, to recorded music- technology has been at the forefront of musical innovation and is the reason traditions have not remained static. In 2018 the computer stands as the primary folk instrument of humans in general, perhaps even superseding the guitar, drums, and voice forever [84]. Musicians in this new field need to have a mastery of their tools more than ever as the relevance of fading traditions wane.

In this thesis, I have strove to make the computing platform and sensor interfaces open-source and extremely low-cost where possible. With the advancement of LiPo and solar batteries the system can run in remote rural areas without need of electricity. This project subsequently seeks to bridge the gap caused by lack of exposure and economic limitations artists face when seeking to develop their own digitally extended platform. By creating and laying forth an open-framework and direct entry point of reference to the general set of centralized computer augmented pitched-percussion innovations outlined in this dissertation- future artists, researchers, and educators can use and expand upon this work towards the development of a sustainable online community using the cloud, IOT, and social media with a low barrier for entry.

1.4 Novel Contributions

Pitched Percussion Hyperinstrument System

The overall contribution of this thesis can be described as a pitched idiophone hyper-instrument platform. It is comprised of:

1. idiomatic gesture sensing interfaces;
2. gesture and signal processing software;
3. a physical model;
4. a custom computing platform;

Each of this component contributions is further described below:

Gesture Sensing Interfaces

1. foot switches,
2. computer vision based 3D skeletal tracking system;
3. 9DOF sensor position tracking system;
4. membrane sensor linear position tracking array;
5. capacitive touch and proximity sensing array;
6. pressure sensor array;
7. optical position tracking sensor array.

Controller information

- assignable momentary and toggle footswitch data;
- multinode skeletal tracking including;
- mallet differentiation in 3 assignable axis;
- assignable pitch, yaw, roll and acceleration;
- assignable linear position tracking;
- direct and discrete touch;
- vertical proximity and touch pressure.

-gestures derived from the interface are reassignable.
-assignments are converted to various mapping logic.
-mapping logic is routed to gesture and signal processing software.

Gesture and Signal Processing Software

1. gesture recognition;
2. gesture prediction;
3. mapping logic automation;
4. pattern differentiation;
5. dynamics calibration;
6. proprioception.

-gesture recognition component analyzes 3D gestures;
-prediction component anticipates control data generated from gestures;
-mapping logic is applied to machine control.

Physical Model

1. a physical model that replaces material resonators;
2. resonators of xylophone;
3. resonators of a traditional Ghanaian xylophone;
4. piezo based surrogate sensor system used for dynamic control of model.

-model reacts to user input.
-model employs piezo system.
-piezo system is coupled with microphone input.
-piezo and mic comprises surrogate system.
-algorithm analyzes audio input data
-algorithm analysis is correlated to training library.
-library is converted to control data for model.

Acoustic Acuation

1. tine excitation;
2. bi-modal bar actuation;
3. multi-modal string platform.

-tine: electro-magnetic resonance
-2 bar modes: piezo driven transducer; solonoid strike.
-3 string modes: self listening/tuning; picking; electro-magnetic resonance.

Computing Platform

1. a single board computer;
2. freely assignable GPIO;
3. Linux open architecture;
4. low latency audio IO.

The thesis is organized following this hierarchy of contributions and describes in more detail each one of them and what role the author had in its development.

Chapter 2

Related Work

2.1 New Instruments for Musical Expression (NIME)

Although instruments of the modern symphony orchestra have reached maturity, musical instruments continue to evolve. A significant area of development is in electroacoustic instruments, combining natural acoustics with electronic sound and/or electronic control means, also called hyperinstruments [5]. The evolution of new musical instruments can be described in terms of both the sound generator and the controller. Traditionally, these separate functions are aspects of one physical system; for example, the violin makes sound via vibrations of the violin body, transmitted via the bridge from the strings, which have been excited by the bow or the finger. The artistry of the violin consists of controlling all aspects of the strings vibrations [136]. The piano is a more complex machine in which the player does not directly touch the sound-generating aspect (a hammer hitting the string), but still the piano has a unified construction in which the controller is the keyboard, and is directly linked to the sound-generation. For hyperinstruments these two aspects are decoupled, allowing for the controller to have an effect that is either tightly linked to the sound produced (as in any conventional acoustic instrument) or can be mapped to arbitrary sounds. Modern advancements in consumer based digital technologies are allowing for unprecedented control over sound in ways that make traditional methods seem primitive. What the traditional model provides in terms of performance is a context for proven-to-be successful socially binding uses of music. This information is critical to retaining and evolving essential methods of music production, dissemination and performance.[28] John Collins describes "The Musical Ghost in the Machine" where he discusses the problems of losing the flow of creative energy that takes place between musicians in a live context. Specifically in this example the machine lacks the ability to express the context of meaning within the music specific to tradition. He explains this aspect is lost due to lack of local rhythmic content and no relation between song- melody and tonal movement of the language. Computers present the risk of eschewing tradition at large. This work seeks to raise awareness of the relevance of tradition when employing novel uses of technology.

Hyperinstruments are augmented acoustic instruments with added electronic sensors used as gesture acquisition devices. Designed for interfacing a computer in a more intuitively musical nature than conventional means, they leverage the performer's instrumental expertise [128]. The acoustic pitched percussion family is problematic when included in electro-acoustic contexts because it is difficult to acquire a direct signal for sound reinforcement, signal processing or generating control data. While typically used for audio reinforcement, microphones are prone to feedback in most EA situations for this particular application as they don't fully capture the signal and pick up unwanted environmental sonic artifacts. These are poor conditions in which to generate control data from. It is also desirable to avoid directly adding electronics and sensors to the instrument that impede traditional technique or corrupt the sound. A common approach to providing digital control without modifying the instrument is the use of an external interface such as a fader/knob style MIDI controller or a set of foot pedals. However, this is problematic, as the performer has to either stop playing in order to interface the computer or use one hand. Foot pedals are better, yet still can be distracting since a mallet player is often moving about on their feet.

To address these concerns a set of tools that take advantage of dynamic gestural parameters intrinsic to pitched percussion performance have been developed [34]. These tools are designed for capturing these gestures in musically meaningful, non-invasive ways in order to control the computer in performance idiomatically to the conventional pitched percussion family. The tools are based on a combination of non-invasive sensing, high-level gesture control detection, and MIR extraction methods. The goal is to develop novel tools any performer can easily use without substantial technical installation requirements or musical constraints. The system also addresses cost issues taking advantage of low-cost and open source resources when possible. In general, the marimba is used as a canonical example for this work, however the toolset is also applied to the vibraphone, the Gyil (an African xylophone from Ghana), and the Likembe- a central African lamellophone.

2.2 Electronic Pitched Percussion

A pitched percussion instrument is a percussion instrument used to produce musical notes of one or more pitches, as opposed to an unpitched percussion instrument which is used to produce sounds of indefinite pitch. The term pitched percussion is now preferred to the traditional term tuned percussion. In this thesis, the xylophone and lamellophone groups serve as the primary instrument targets for augmentation.

Xylophone

Previous areas of research that directly inspire this body of work include Randy Jones' Soundplane (a force-sensitive surface for intimate control of electronic music that transmits x, y and pressure data using audio signals generated by piezo input captured at a high audio sampling rate) [108], Adam Tindales acoustically excited physical models using his E-Drumset (a piezo based drum surface that uses audio signals, created by actual drumming, to drive physical models) [107], Ajay Kapurs North Indian hyperinstrument and robotic instrument system [106] and the Radiodrum [12].

These examples represent a culmination of platforms this work merges, applied specifically to pitched percussion[1]. Past work regarding electronic pitched percussion includes: Simmons Silicone Mallet [104], Mallet Kat[2], Xylosynth[3], Marimba Lumina[4], The Deagan Electravibe and The Ludwig Electro-Vibe electric vibraphone systems[17]; Deagan Amplivibe and Mussar Marimbas Inc. Ampli-pickup[62]; and Ayottes (Fig. 2.1), K and Ks[5], which formerly featured MIDI (Fig. 2.2), , Vanderplaas's[6] (currently features MIDI), and Mallettechs [7] piezo based systems [131].

Conventional percussion based trigger interfaces typically use piezo/FSR technology (in some cases acquiring the direct audio signal which is scaled for data and amplified for processing), but usually only detect onset and velocity scaled to the relatively limited MIDI protocol, compared to Open Sound Control [150]. In this case most of the dynamic information contained in the performance gestures is lost. Seeking to address these concerns this thesis explores analyzing/utilizing the full audio waveform of

[1] http://nuke.ninoderose.it/Vibrafono/vibrafonoelettrico/tabid/95/Default.aspx
[2] https://www.alternatemode.com/malletkat/
[3] http://www.wernick.net/
[4] http://www.absolutedeviation
[5] http://kksound.com/instruments/vibraphone.php
[6] http://www.vanderplastal.com/
[7] https://www.mostlymarimba.com/accessories/v3-pickup-system.html

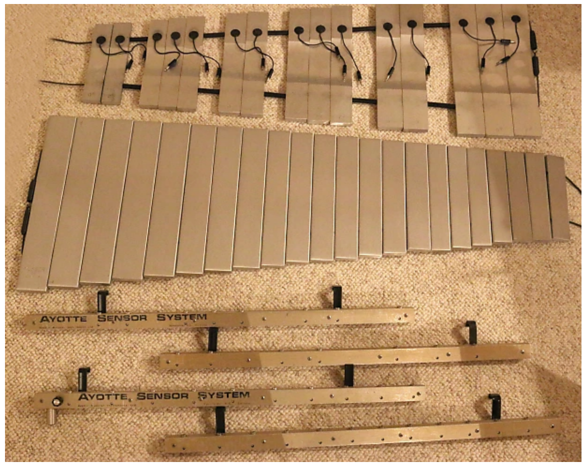


Figure 2.1: Ayotte sensor system (ca. 2000)

the acoustic instrument [4] coupled with extended 3D control mapping options [36] to also take advantage of the entire range of motion involved with performance in order to modulate the source audio and control other instruments (synths/robotics/FX) remotely without interrupting conventional performance techniques.

Commercial mallet-percussion based MIDI control solutions are limited. The Simmons Silicon Mallet and MalletKat both incorporate rubber pads with internal FSR's laid out in a chromatic keyboard fashion and vary in the octave range afforded. The Simmons model offered very little reconfigurability as it was basically an autonomous instrument and had a 3 octave mallet keyboard controller played with mallets or sticks connected to an analog sound module similar to the SDS8[8], but laid out in a xylophone format. The MalletKat has some internal digital sound sampling synthesis and offers the same typical internal MIDI configurability as high-end piano style keyboard controllers yet catered to the mallet instrumentalist. It can connect to any MIDI compatible sound module or otherwise with appropriate MIDI conversion interfaces (MIDI/CV/OSC,etc.). K and K Sound used to produce a piece of hardware that extracts control data from a vibraphone using an array of piezo sensors (fig. 2.2) and is the first known commercial device of this type.

The Xylosynth, by Wernick, is a digital instrument that aims at providing the same natural haptic response as a wooden xylophone. It generates MIDI notes yet produces no acoustic sound. Strikes are converted to MIDI note data, and therefore it is purely a controller requiring an additional sound module/instrument to generate sound. The

[8]http://www.simmonsmuseum.com

Figure 2.2: Original KandK MIDI-Vibe

Marimba Lumina (ML) is another mallet style controller. ML adds a few new features to the usual capabilities of a mallet instrument controller. Expanding the potential for expressive control, the instrument responds to several new performance variables, including position along the length of the bars, mallet presence, dampening, and note density. Additionally, it allows one to relate different controls to be triggered when playing different bars. User definable ”Zones” also allow portions of the instrument to respond to gesture in different ways. ML can also identify which of four color-coded mallets has struck- mallet differentiation. This allows one to program different sounds for different instrumental responses for each mallet. This affords the ability to implement musical structures in which one mallet selects a course of action while others modify or implement it, or to simplify voice leading and give each mallet independent expressive control over the same sounds. The ML is a direct influence that has influenced this research, as it is now extremely rare and expensive. The work presented in this thesis offers a way to emulate behaviors exclusive to the ML while adapting them to acoustic instruments.

While these mallet based DMI's and controllers are all relatively robust, they are still unable to offer the same haptic and spatial feedback as an acoustic instrument. Typically limited to MIDI formats, they require the instrumentalist to modify playing technique in order to achieve successful results. They are also costly and relatively rare.

Lamellophone

Lamellophones have a series of tines fixed at one end and free on the other. Musical tones are produced by depressing and releasing the free end with the thumb, allowing the tine to freely vibrate [68]. They are ideal instruments for augmentation being hand-held, compact and ergonomic. The instruments design facilitates uninterrupted traditional playing while allowing the performer to engage various specific sensors with broad or minute arm/hand gestures or explicitly with the free fingers. The unused surface area of the body is ideal for intuitive sensor placement, while direct audio acquisition is simple and robust via a piezo transducer. Lamellophones are typically acoustic folk instruments, however, augmentations, such as these, equip the instrument for contemporary settings where audio processing/sound reinforcement are presumed while offering experimental, novel interactive sound design/multi-media performance practice capabilities.

Previous work in lamellophone hyperinstruments can be reviewed in [132]. Electrified lamellophones are common [142], however lamellophone hyperinstruments are rare. This work is inspired by Konono No.1 from the Democratic Republic of the Congo. Konono No.1, a band founded by Mawangu Mingiedi in 1966, combining voice, homemade percussion, hand-carved wooden microphones, and 3 custom built electric Likembe. Their music was originally adapted Zongo ritual music played on horns crafted from elephant tusks. Based centrally in the dense urban environment of Kinshasa, their acoustic music became inaudible because of increased noise pollution. Compelled to build amplification systems for their instruments using discarded, obsolete electronics, their motivation was partially of practical concern [46]. To sustain their musical heritage, amplification was essential. As a result, an unprecedented neo-traditional musical movement was born, spawning a range of new instruments and sound design practices. The advent of electronics in their music propelled Konono No.1 onto the world stage, while leaving an indelible mark on their own cultures contemporary music scene [20]. In 2007, they toured as the supporting backing band for Bjork (14 time Grammy Award nominee from Iceland), while collaborating with critically acclaimed producer Timbaland. Konono No.1s album Live At Couleur Cafe was nominated for a Grammy in 2008. They won a Grammy in 2011, as guests on pianist Herbie Hancocks album The Imagine Project. My system utilizes relatively easily sourced, low-cost components drawing from Konono No. 1s [46] spirit of using salvaged electronic components to develop new electro-acoustic instruments, analog signal processing and amplification systems for their neo-traditional lamellophone based music.

Instrument manufacturer Hohner experimented with electric lamellophone designs in the past mid-century. The Guitaret[9] has internal tines that are excited when the instruments unique keys are depressed (Figure 2.3) producing a lamellophone style tone [152]. The Pianet[10] is another of their lamellophone inspired instruments from the same period [144]. An electric piano with tines that are, by default, slightly depressed in its nonplayed state- a piano-style key press releases the tine causing the tine to vibrate (Figure 2.4) producing the respective tone- an inverse lamellophone effect. Both have passive electromagnetic single coil pickups per tine summing to an outputting monophonic signal intended for direct audio amplification via standard 1/4 jack. Although unique, rare, and antique, they are excellent examples of re-purposing ancient musical instrument engineering techniques to achieve novel results in contemporary music. This type of re-purposing is the focus of OSPs engineering approach for this work; much the way Konono No.1s work re-purposes technology in the form of recycling discarded analog electronics to arrive at new instruments, sounds, and context based on traditional means.

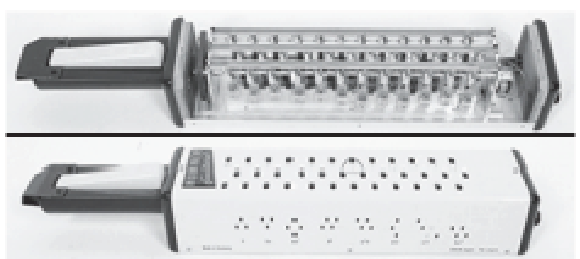


Figure 2.3: Guitaret

Lamellophone style hyperinstruments can be seen in the work of: Adrian Freeds Kalimba Controller [48]; Fernando Rocha and Joseph Mallochs Hyper-Kalimba [111], Jose A. Olivares Kalimbatronique [97]; Ian Hattwicks Mbira controller [55]; and Daniel Schlessingers Kalichord [117]. Kalimbo by Rob Blazy[11] is an example of hybrib EA/digital digital design using OSP and arriving at a new instrument. A patent from 2004 [72] describes an augmented Mbira for interactive multi-media systems. An electromagnetically actuated, electrified lamellophone can be seen in Octants work

[9]https://en.wikipedia.org/wiki/Guitaret
[10]https://en.wikipedia.org/wiki/Pianet
[11]https://vimeo.com/179818357

[12]. There are even virtual software versions: plugin instruments and multi-touch screen instruments coupled with a lamellophone style GUI graphic for interfacing the synthesis models.

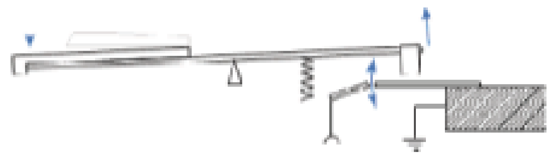


Figure 2.4: Pianet tine

2.3 Non-Invasive Sensing

The focus of this thesis is the development of a system for the design of digitally extended musical instruments (DMI) and their utilization in multi-media performance contexts- specifically, traditional and conventional acoustic pitched percussion instruments. In this section, previous work in non-invasing sensing which is a core idea in this work is presented. An important influence has been the 3D control paradigm previously developed on the Radiodrum. The Radiodrum is an electromagnetic capacitive 3D input device that uses audio carrier waves transmitted by the tip of the drum stick to track the sticks tips movement in 3D space through capacitive sensing using an antenna. Invented by Dr. Max Matthews [81], it has been pioneered and mastered by Dr. Andrew Schloss [118]). Using the Kinect (a motion sensing input device for the Xbox 360 video game console) aspects of the 3D sensing features of the Radiodrum are emulated and compared in this thesis. The Kinect has native features that, when adapted for use within vibraphone performance contexts, offers a dynamic and flexible intuitive interface prototyping platform. This is an example of effectively embedding new technologies onto traditional instruments in a non-invasive capacity.

This work has been influenced by several new music instruments beyond the pitched percussion family. The Theremin is unique and relevant in that it is played without physical contact controlled by 3D hand gestures and therefore can be considered non-invasive. The modern Moog Ethervox, while functionally still a Theremin, can also be used as a MIDI controller and allows the artist to control any MIDI enabled FX/synthesizer/robotic instrument with it.

[12]http://matthewsteinke.com/octant

With the development of new gaming interfaces based on motion controls, it has become easier to generate low-cost and robust musical interfaces controlled by bodily movement. That allows one to use natural motion, which, as observed by [34], is an inherent part of the performance of a pitched-percussion player. Motion-based game controllers were used as musical tools in [56], taking as basis the Wiimote device. In [87], the idea of using the Kinect as a controller for musical expression is discussed. Here, the Kinect is a low cost way to obtain 3D position data in order to enhance the soundscape manipulation capabilities of a performer. The ability to network existing musical instruments with computer systems in this way creates large potential.

Hyperinstruments are acoustic instruments that have been augmented with sensing hardware to capture performance information[80]. The most common use of hyperinstruments has been in the context of live electro-acoustic music performance where they combine the wide variety of control possibilities of custom controllers with the expressive richness of acoustic instruments. Another interesting application of hyperinstruments is in the context of performance analysis. The most common example is the use of (typically expensive) acoustic pianos fitted with a robotic actuation system on the keys that can capture the exact details of the player actions and replicate them (Yamaha Disklavier). That system allows the exact nuances of a particular piano player to be captured, and when played back on the same acoustic piano it will sound identical to the original performance. The captured information can be used to analyze specific characteristics of the music performance such as how timing of different sections varies among different performers. John Collins discusses the potential for analysis in African music where the rhythmic systems and interaction between performers are so nuanced that "...purely for the analytical reasons of studying the minute details of complex rhythms (such as local cross-rhythmic ones) the computer is very useful - as standard notation is often not precise enough to deal with the tiny spaces between the quavers"[28]. *Mistic*, the lab in which this research took place, has been an early proponent of Computational Ethnomusicology in this regard[103].

The majority of hyperinstruments that have been proposed in existing literature have either been modifications of existing Western musical instruments (such as guitars, keyboards, piano and strings) or totally novel. The focus of this work is to offer context for instruments beyond the conventional marimba and vibraphone, as this work includes extending the Gyil (Ch3.2a/b), a traditional wooden West African xylophone, by digital sensing capabilities. The conventional Gyil has 14 or 15 wooden bars tuned to a pentatonic scale mounted on a wooden frame. Hollowed-out gourds are hung from the frame blow the wooden bars acting as resonators. The Gyil's sound is characterized by a buzzing resonance, due to the preparation of the gourds- holes with a type of wax paper covering (typically spider silk egg casings). The cover acts as vibrating membranes, but have irregular forms due to their material properties and the whole shape.

A similar emphasis on western music occurs in musicology and MIR research. As mentioned above, this has prompted research in computational ethnomusicology, which is defined as the use of computers to assist ethnomusicological research [50], [123]. Outside of exclusively Western intstruments, contemporary research is being done regarding the use of hyperinstruments in other music cultures. For example, it has been explored in the context of North Indian music [3] and Indonesian Gamelan where digital sensors have been used in the development of Gamelan Electrica [70], a new electronic set of instruments based on Balinese performance practice. One interesting motivation behind the design of Gamelan Electrica is a reduction in physical size and weight, simplifying transportation- this is carried over into the exclusivity of the manufacturing of traditional instruments as well [15]. This concern also motivated investigating replacing the gourd resonators by simulating them digitally using physical modeling. The hybrid use of microphones to capture sound excitation and simulation to model the needed resonances has been proposed in context of percussion instruments and termed acoustically excited physical models [7], [4]. Past physical modeling of pitched percussion instruments focused on the vibraphone and marimba [143]. This method can also be used to aid in the preservation of the instrument, as the tradition wanes and becomes endangered[43].

Most music in the world is orally transmitted and traditionally was analyzed based on manual transcriptions either using common music notation or invented conventions specific to the culture studied posthumously. In the same way that direct sensing technology opened up new possibilities in the study of piano performance, direct sensing can provide invaluable information towards the understanding of traditional music. Music is defined as much by the process of creation as by recorded artifacts [23]. Capturing information about musicians actions can aid in understanding the process of music creation. This is particularly important in orally- transmitted music cultures that are slowly hybridizing or disappearing due to the onslaught of global pop music culture. The goal is that by interfacing the Gyil with the computer we will be able to understand more about how it is played, enable the creation of idiomatic electro-acoustic compositions and performances that integrate it, and better understand the physics of its sound production.

In addition, the work intends to spark collaborations with masters of the tradition where electro-acoustic mediums are largely unexplored due to the typically limited access to the associated technologies[13]. Indirect acquisition refers to the process of extracting performance information by processing audio of the performance captured by a microphone rather than using direct sensors. It has been motivated by some of the disadvantages of hyperinstruments such as the need for modifications to the original instrument and the difficulty of replication [134],[127]. In general it requires sophisticated audio signal processing and sometimes machine learning techniques to extract the required information. An interesting connection between direct and indirect acquisition is the concept of a surrogate sensor. The idea is to utilize direct sensors to train and evaluate an algorithm that takes as input the audio signal from a microphone and outputs the same control information as thedirect sensor [11]. When the trained surrogate sensor(s) exhibits satisfactory performance it can replace the direct sensor(s).

2.4 Physical Modeling Synthesis

In this thesis, a physical model for the Gyil is described. The Gyil is an African mallet instrument with a unique system of resonator gourds mounted below wooden bars. The Gyil gourds are unique in that they have holes drilled in them covered over with membranes. These membranes react to sound pressure in a highly non-linear fashion and produce a buzzing sound when the bars are played with force. Physical modeling efforts have typically focused on modeling plucked and bowed stringed instruments, struck bars, and membranes usually in the context of western classical music instruments. The Gyil, and the nonlinear processes that are the cause of its characteristic sound, have not been studied before in the context of physical modeling synthesis. One objective behind creating the model is to provide performers and composers an audio effect that can be applied to the signal of any mallet percussion instrument. This enables the use of techniques associated with the Gyil without having to include the actual gourds and membranes. The gourds are cumbersome, fragile, difficult to construct, and cannot be added to conventional pitched percussion

[13]https://www.raspberrypi.org/blog/guest-post-a-pi-lab-in-rural-ghana/

instruments. Electro-acoustic mallet percussionists have little choice but to apply filters and audio effects that are designed for other instruments such as guitar pedals. Contrastingly, this model and its associated audio processing effect are historically relevant to pitched percussion instruments/sound design paradigms.

The Gyil is an ancestor of the marimba, and originates in western Africa within the region defined by the political borders of northwest Ghana, northeast Ivory Coast, and southwest Burkina Faso. It has strong roots in the region near Wa, Ghana, on the Black Volta river region where it forms a key part of the Lobi and Dagara cultures, occupying a central role, with Gyil performances accompanying major ceremonies and festivals [89]. There is a special variety of Gyil for solo performances at funerals and it is believed that the sound emitted from the instrument escorts the deceased soul into the next world.

The Gyil is a pentatonic instrument with between 11 and 18 keys played with rubber tipped wooden mallets that are held between the index and middle fingers and about the size of a nickel in diameter. The keys are carved from a local hardwood called Legaa, and are mounted to a wooden frame with leather strips [90]. A dried calabash gourd is hung below each bar. These gourds have irregularly spaced holes drilled in them, which are covered over with spider silk egg casings. These egg casings form membranes that produce a buzzing sound when the bars are played with enough force. This buzzing sound distinguishes the Gyil and it is this aspect of the instrument that the focus of the modeling efforts are on. (Figure 2.5) shows photos of a Gyil and the associated gourds.

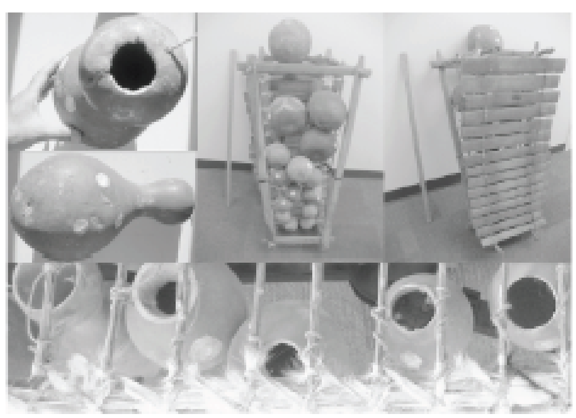


Figure 2.5: Gyil: bass gourd (14" deep, 10"dia.-body, and 4"dia.-mouth)

The sound-design of the buzzing is motivated by the Lobi belief that the vibrations it produces balance the water in the human body and have physiological healing qualities. There is no traditional written notation for the Gyil, although efforts of transcription have been published by Ghanaians [135]. Traditionally the repertoire and technique is passed on orally. Players typically perform in groups solo or in duo, accompanied by a bell, a calabash resonator drum with animal skin stretched over an opening played by hands called the Kuar, and a double pitched log drum played by sticks called the Gangaa. The music combines pre-written melodies with improvisation comprised of bass ostinatos using the left hand and lead solos using the right hand, similar to jazz arrangements. The repertoire is made up of pieces that are time, season, and context appropriate- funeral songs for men vs. women, wedding songs, etc.

Physical modeling is an audio synthesis technique in which the sound of an instrument or circuit is emulated by numerically solving the differential equations that describe the physics of the system [75]. In contrast to other synthesis techniques the parameters of physical models have a direct correlation to the properties of the sound-generating object being modeled. For example, in a model of a vibrating string the length, linear density, and stiffness can be directly controlled with parameters [98]. Most research on physical models has focused on modern Western European instruments. Even for instruments that have been intensely studied, such as the violin, the resulting sound is still quite distinct from the original acoustic instrument [124] [38] [122]. At the same time physical models provide a realism and physicality to the generated sound that is impossible to achieve with techniques such as waveform sampling.

In early work, physical modeling was based on linear approximations of non-linear phenomena, such as the behavior of percussive surfaces [30] [76]. Non-linear models can provide a more realistic sonic experience [16], however are considerably more complex to understand [45]. This difficulty has motivated the development of different exploratory interfaces for the parameters of these models [52] as well as learning models that can automatically obtain interesting non-linear mappings for sound synthesis [40].

Another idea is to combine or manipulate the audio signal acquired from an acoustic instrument with digital processing as a form of hybrid synthesis [126], [7]. There are some existing efforts to obtain physical models for nonwestern-European instruments, like the ancient Greek auloi [64], as well as pitched percussive instruments [143], [57]. However, there are many instruments, which have not been physically modeled, including the Gyil. As described in section 2.3, each wooden bar of the Gyil is retrofitted with contact microphones providing clean audio signals for each bar that can be used for sound enhancement, synthesis and transcription [113]. An early version of the physical model of the gourds described previously in 2.3 was also used. The algorithm is improved, the design is explained in more detail (Ch3.2c), various configurations get explored, and feedback from a group of expert musicians having intimate knowledge of the instrument is reported.

2.5 Acoustic Actuation

Introduction

There is a long history of mechanical devices that generate acoustic sounds without direct human interaction starting from mechanical birds in antiquity to sophisticated player pianos in the early 19th century that could perform arbitrary scores written in piano roll notation. Using computers to control such devices has opened up new possibilities in terms of flexibility and control while retaining the richness of the acoustic sound associated with actual musical instruments. The terms music robots or music robotic instruments have been used to describe such devices [67]. These new robotic instruments have a legitimate place with potential to become part of an embedded conventional musical practice beyond research curiosity. While musical robotics might seem niche and esoteric at this point [22], historic innovations such as monophonic to polyphonic music, electrical amplification of the guitar, or computers in the recording studio all brought skepticism before eventually becoming mainstay practices. Although such music robots have been used in performance of both composed and improvised music as well as with or without human performers sharing the stage, they are essentially passive output devices that receive control messages and in response actuate sound producing mechanisms. Their control is typically handled by software written specifically for each piece by the composer/performer. However there are new companies emerging producing commercials solutions [http://polyend.com/].

The author has worked intimately with musical robotics, most notably: Trimpins: (CanonX+4:33=100) robotically actuated prepared piano instruments. This set of instruments employ electro-magnetic string actuation making the strings resonate, creating a droning sound effect. Solenoid beaters either hit the strings directly or various preparations attached to the strings creating percussive effects; Eric Singers

LEMURbots, his Guitarbot [119]; and Ragtime Wests pneumatic guitars [14]. The author also composed and performed with EMMIs[15] robotic string instrument- AMI (Automatic Monochord Instrument)[137]. In retrospect all of the instruments had quirks, as expected. The Guitarbot, while arguably the most expressive, is manually tuned and has no self-sensing technology. It typically goes out of tune during a concert, is cumbersome to transport and difficult to maintenance. Trimpins pianos were a site-specific installation and not practical instruments, being large and immobile. AMI serves as a model for modularity, mobility, as well as expressivity.

Design

In the past few years there has been a growing interest in music robotics. The proliferation of physical computing and the support of online user communities dedicated to specific platforms have made working with sound and computers, along with building custom hardware and software for such tasks more accessible than ever to the general public. Robotic instruments that generate sound acoustically using actuators have been increasingly developed and used in performances and compositions over the past 10 years [67].

This facet of the natural sound radiation patterns occurring from acoustic instruments coupled with the sophisticated control paradigms that computers allow make musical robotics a dynamic platform. When observing the electronic music performance convention of the performer using a computer managing many layers of software synths, samplers, audio loops, and effects processing occurring simultaneously one gets a better perspective on the motivational factors that musical robotics might offer. Simply put, the new instruments being introduced in the research and artistic community are opening up the possibility for a world of acoustic sounds that have yet to be heard, matched with a level of digital interactivity that has never before been possible. These new instruments present unprecedented potential for novel multi-media performance contexts with vast richness. Aside from modifying existing acoustic instruments, a designer can invent entirely new instruments considering the physics of available actuators so as to optimize their potential. Adapting acoustic instruments designed specifically for the gestures a human produces using conventional techniques is often difficult to emulate mechanically on any level of musical sophistication comparable to a humans nuances on the same instrument[110].

[14] http://www.ragtimewest.com/

[15] http://expressivemachines.com/dev/wordpress/

Proprioception

Proprioception, in this case, refers to the digital electro-mechanically actuated acoustic instrument to be autonomously able to sense, respond to, and create musical events specific/appropriate to the musical setting. An early example of an automated, programmable musical instrument ensemble was described by Ismail al-Jazari (ca. 1136-1206 A.D.) a Kurdish scholar, inventor, artist, mathematician that lived during the Islamic Golden Age (the Middle Ages in the west). Best known for writing the Book of Knowledge of Ingenious Mechanical Devices in 1206, his automata were described as fountains on a boat featuring four automatic musicians that floated on a lake to entertain= guests at royal drinking parties. It had a programmable drum machine with pegs (cams) that bumped into little levers that operated the percussion. The drummer could be made to play different rhythms and different drum patterns if the pegs were moved around, performing more than fifty facial and body actions during each musical selection. This was achieved through the innovative use of hydraulic switching[60].

A modern example of a robotic musical ensemble is guitarist Pat Methenys Orchestrion, which was specifically influenced by the Player Piano 2. Metheny cites his grandfathers player piano as being the catalyst to his interest in Orchestrions, which is a machine that plays music and is designed to sound like an orchestra or band [18]. As the Robotics and Interface Technician for that project I was working intimately with the instruments and designing interface solutions between the guitar and computer that was driving the robotics. A seminal text on the subject is Machine Musicianship [112], in which one of the sections describes a comprehensive system for the composition, creation and performance between humans and robots. Rowe describes improvisational and composition systems that combine features of music feature extraction, musical analysis and interactivity to generate engaging experiences for the audience.

Chapter 5 describes recurring problems I have experienced with robots in music performance that have led to the development of signal processing and machine learning techniques informed by music information retrieval ideas. Musicians, through training, acquire a body of musical concepts commonly known as musicianship. Machine musicianship refers to the technology of implementing musical process such as segmentation, pattern processing and interactive improvisation in computer programs. The majority of existing work in this area has focused on symbolic digital representations of music, typically MIDI. The growing research body of MIR, especially audio-based, can provide the necessary audio signal processing and machine learning techniques to develop machine musicianship involving audio signals.

In my work, the integration of machine musicianship and music robotics has been used to develop a robotic percussionist that can improvise with a human performer. these innovations are described in detail in Chapter 5. Work by Overholt [39] and Kapur [136] have greatly influenced the development of the interface solutions presented, building on previous work in the field of gesture sensing and machine learning for performance. Shimon, a robotic marimba [146] is an example of relevant previous work surrounding Machine Musicianship- where a computer system can analyze and perform musical events autonomously with human performers. Closely related to my work, Shimon's human-robot based Jazz improvisation system [58], uses a gesture based framework recognizing the musicianship involved not just the production of notes, but also of the intentional and consequential communication between musicians [59].

2.6 Summery

This is a very broad thesis describing a multi-modal approach to EA pitched percussion augmentation. The thesis covers a lot of different topics and for a first time reader it might be it hard to navigate. It is intended that this document provides a nexus for the pitched percussionist and composer seeking to delve into idiomatic hyperinstrument design and rapid prototyping. The work is meant to provide a foundation for the history of the context and offer insight into where possible directions might be found.

The framework presented proves to be an instrument with great flexibility due to the spectrum of solutions explored. The range of topics and techniques used to solve the goals of research might superficially seem unrelated. However, when leveraged, the extensive set tools provide vast possibilities for control of realtime multimedia, audiovisual processes for novel interaction to the performer.

Part II

IDIOMATIC INTERFACES, MACHINE LEARNING AND ACTUATION

Chapter 3

Musical-HCI

3.1 Gesture Sensing

This work merges my experience as a percussionist, electronic musician, and studio producer into a singular, real-time performance based practice combining what would have previously only been possible to realize sonically in a static, pre/post-production context. Using sensing techniques allowing for uninterrupted interfacing via traditional playing techniques, the primary contribution is less novel engineering than it is an explicit, approach to developing minimally invasive hyperinstrument augmentations for idiomatic Machine Learning interaction for the pitched percussion family. Aside from serving the authors artistic motivations, the work is intended to provide an easily reproducible, low-cost context for other artists to abstract inexpensively towards developing their own previously inconceivable hyperinstruments with relative ease.

3.1.1 Xylophone Based

In order to take into account the specifics of musical interaction, one needs to consider the various existing contexts- sometimes called metaphors, for musical control [147] where gestural control can be applied to computer music. In order to devise strategies concerning the design of new hyperinstruments for gestural control of sound production/modulation, it is essential to analyze the characteristics of actions produced by expert instrumentalists during performance [35]. The importance of the study of gestures in new hyperinstrument design can be justified by the need to better understand physical actions and reactions that take place during expert performance.

The xylophone is marginalized in contemporary music for several reasons: the antiquated, bulky frame makes it hard to move, and its sound is generally difficult to amplify without expensive special equipment; however, it makes for good coupling of technology because of the very constant visceral movement required to produce sound. This is good for two reasons: the sound produced is well suited for conversion into control data and the motion involved in the process is ideal for computer vision recognition. Thus, along with furthering specific 3D control practices, I subsequently seek to advance the role of the acoustic xylophone tradition to incorporate advanced HCI capabilities via this method. Below a set of techniques for extending the digital control possibilities of pitched percussion instruments is described.

Computer Vision

A) Magic Eyes: An initial design goal for a new interface using the Kinect was achieving a set of multi-axes, mappable 3D gesture controllers. This was done in collaboration with graduate research assistant, electrical engineer Gabrielle Odowichuck. My role was as design engineer, where I contributed towards high-level theoretical design concerns, sensor implementation and mapping strategies. The first challenge of my new musical interface was communicating with the Kinect controller. OpenNI API was chosen to communicate with the Kinect because it offered access to the Kinects embedded user tracking functionality. From there, a C++ application was developed that takes the 3D position data provided by the Kinect and sends the position data as a stream of Open Sound Control (OSC) messages.The Kinect and it's associated software libraries are used to perform human skeleton tracking. This tracking produces estimated positions of the performer's mallet tips, which can then be mapped to musical parameters.Once the position data is in the OSC format, many other music-related programs can receive and manipulate the data.

B) Fantom Faders: Each Cartesian axis of motion was mapped to filter parameters that modify live and sampled audio from the vibraphone, creating control parameters accessed through the mallets in the space over the keys oriented as a virtual MIDI mixer. This tracking is used to augment each bar of the vibraphone with the functionality of a fader. Using this technique on a 37 bar vibraphone, it is possible to provide the functionality of 37 virtual faders that may be used as mappable controllers with unlimited preset variations.

This augmentation, illustrated in (Figure 3.1), provides an intuitive platform that allows the performer to control a large number of sliders without interrupting performance. In this case, mallet tips are tracked based on their color. In order to detect the positions of the mallet tips, the color image from the video camera is transformed into Hue, Saturation, and intensity Values (HSV). A threshold is applied to each of these signals to filter out unwanted colors. The resulting signals are combined, and a contour detection algorithm is executed. This process yields bounding rectangles that identify the mallet tips. The centroid of the bounding rectangle is assumed to be the position of the mallets in the virtual representation.

Figure 3.1: Virtual vibraphone faders

Determining the position of the mallet tips in terms of the vibraphone bars requires the creation of a virtual representation of the instrument. This representation was created using explicit measurements of the vibraphone's bars. These measurements were used to create a model consisting of the set of bars, each with a corresponding placement, size, pitch, and an associated control value. The algorithm supplies the mallet positions relative to the camera, but position data in the virtual space is preferred. Effectively mapping the tracked mallets involved several transformations, and requires a calibration phase in which the position of the vibraphone with respect to the camera is also recorded. Once the position of the mallets in the same space as the vibraphone is obtained, the system yields information on what bar is currently covered by the mallet, and a fader value associated with that bar. A delay-based activation time was added to the algorithm, so that the performer must pause a few milliseconds on each bar before the sliders will latch start to change based on mallet movement within the fader zone.

The 640x480 resolution used in my prototype is sufficient to perform accurate detection of the mallet positions. The main resolution restriction is that the camera should point at the vibraphone from a distance that makes each bar to be present in a reasonable number of pixels. While this may suggest using a higher resolution, it is important to note that the algorithms must be executed in real-time, therefore less data is desirable. Future work could involve fusing the video camera data with IR sensors data, as well as using motion-tracking algorithms in combination with the current contour detection algorithms. This fusion will improve the robustness of the tracking system, and address the current sensitivity to changes in ambient lighting.

Experiment

In many ways, the Radiodrum and Kinect are similar controllers. Both return a set of 3D positions in real time. However, there are some major differences between these two pieces of technology. In this experiment, I present some early experiments that aim to demonstrate some of the major differences between these sensors.

(Figure 3.2) shows the basic layout of the hardware. The Kinect connects to the computer via USB, and the Radiodrum via firewire through an audio interface. I have also connected a microphone to the audio interface, which is used as a reference when comparing the reaction of the sensors - similar to [78].

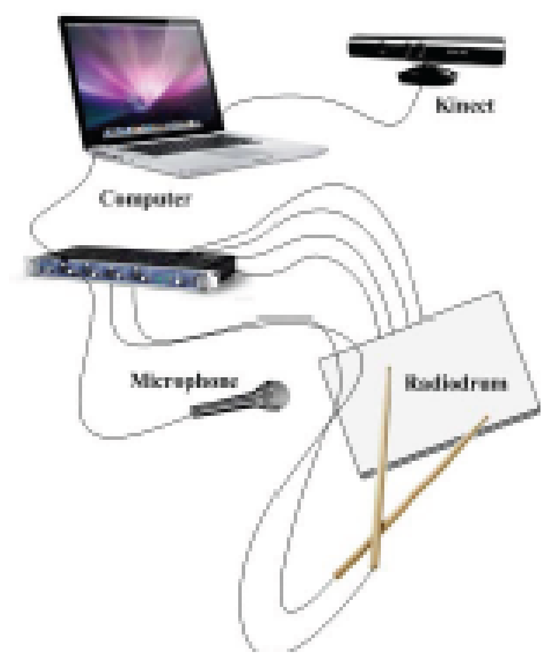


Figure 3.2: Hardware diagram

Custom software was developed to record and compare data received from both sensors. The program flow is shown in (Figure 3.3). A C++ program was written that takes the user tracking data made available by OpenNI and sends it as OSC data. A Max/MSP program then receives data from the audio interface and converts it into position data, upsamples the Kinect data, and saves all the data to a file. The transport of data between the two programs is notable as it may lead to increased temporal effects. Various movements were captured in an attempt to demonstrate some of the observed effects discovered when using these sensors in a musical context.

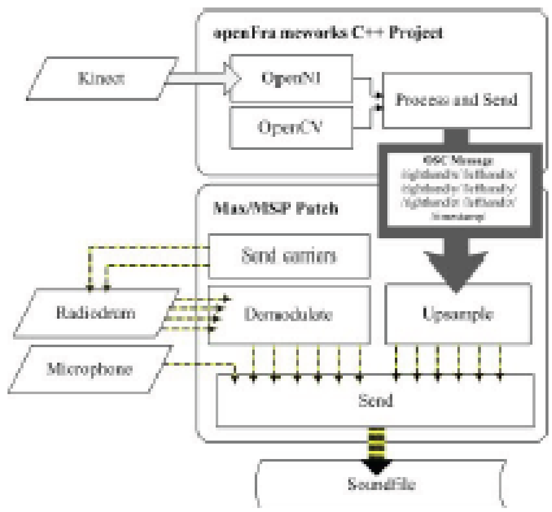


Figure 3.3: Software diagram

This work is not an attempt to show whether one device was superior to the other. Instead, I more interested in comparing these sensors so that data can be more intelligently fused for better control over the instrument. I experimented with practical applications using the Kinect in a musical scenario. Using a vibraphone as the sole sound source, I arranged a system for improvisation that would take advantage of the new extended capabilities.

The audio from the acoustic vibraphone was captured with an Akai 414 microphone and input into an RME Fireface 800 connected to a Macbook Pro hosting Ableton Live. The Vibraphone audio channel went to the Master bus with a slight bit of equalization and compression; it was also routed to Auxiliary sends 1 and 2 (fig. 3.4). On Aux Return 1 was a harmonic resonance filter plug-in. The X axis in the right hand controlled the tonic of the 7 voice harmonic scale pre-programmed into the filter. Moving the mallet to the left would lower the tonic, as would the inverse-moving the mallet to the right, would raise the pitch (see Fig. 3.5). The Y axis controlled the global frequency gain of the filter, allowing a performer to over-accentuate the high frequencies of the filtered audio by raising the mallet and boosting low fre-

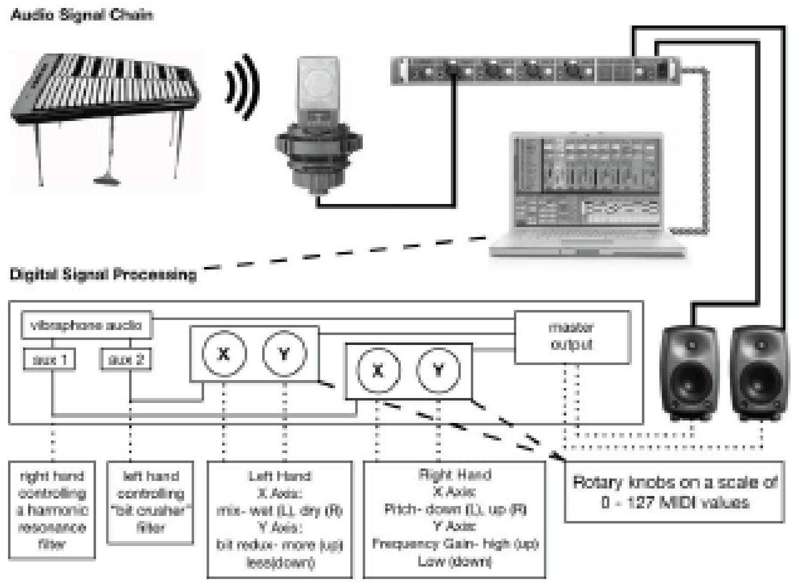


Figure 3.4: Audio aignal chain

quencies as the mallet is lowered. Aux Return 2 hosted a bit reduction plug-in that was used as a type of digital distortion, much the way a guitarist would use an analog fuzz pedal. Except, in this case, the Ghanaian Gyil was a direct influence in that in a traditional context the Gyils resonators (made of gourds) would have holes drilled into them with spider egg casing stretched over them resulting in an intentional distorting buzz of the acoustic sound. The X axis in the left hand controlled the ratio of the filter applied to the source signal. Moving the mallet to the left would reduce the effect resulting in a dry source signal, moving the mallet to the right, would increase the wetness of the filter effect. The Y axis controlled the rate of reduction. The bit rate decreases when the mallet is raised resulting in more distortion, lowering the mallet results in less distortion. These effects are calibrated so that the default playing position of the vibraphone results in no processing reacting only to extended gestures.

The specific mappings and filter parameters chosen were not arbitrary, but rather specific to the researchers artistic practice. Being both a sound designer and computer musician, the researcher is also a vibraphonist and chose intuitive mappings based on specific vibraphone techniques within a given original composition. The extended digital sound design parameters were based on the familiarity of both the music and instruments natural characteristics.

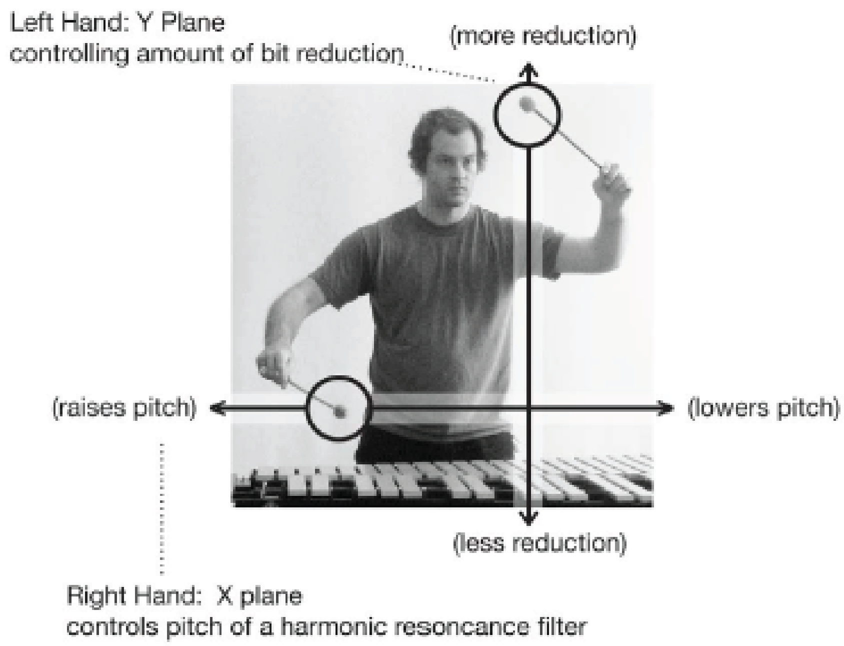


Figure 3.5: Music control design

Latency

Humans can control transient events with a relatively high speed. This is demonstrated in the percussive technique known as the flam, where trained musicians can play this gesture with a 1ms temporal precision [41]. It is also important to look at the accuracy of events. Delays experienced by the performer will change the perceived responsiveness of the musical instrument, a major consideration for musicians.

A basic difference between these two sensors is the vast difference in the frame rate of the captured data. The Radiodrum sends continuous signals to an audio interface, and the sampling rate of the data is determined by the audio interface. For this experiment, I used a frequency of 48000 Hz, but higher rates are possible. Conversely, the Kinect outputs position data at approximately 30Hz, the starkest difference in the capabilities of the sensors.

I begin by demonstrating this latency by holding the tip of the Radiodrum stick, and hitting the surface of a frame drum that has been placed on the surface of the Radiodrum. Now I have the output of three sensors to compare. The microphone has very little delay, and the auditory event of the frame drum being hit will occur before these events are seen by the gesture capturing devices. For my purposes, the audio response is considered a benchmark. The Radiodrum will capture the position of the

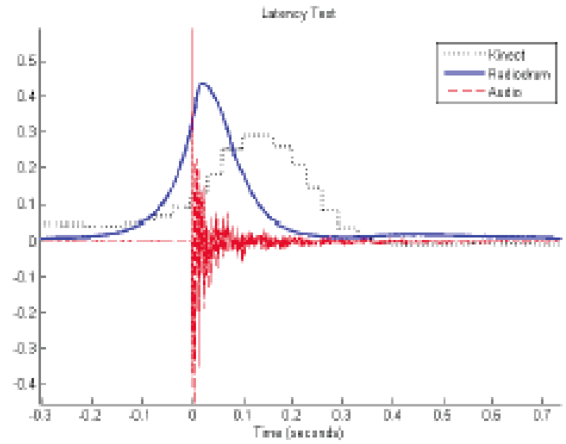


Figure 3.6: Latency: Radiodrum vs. Kinect

tips of each drumstick during this motion, and the Kinect will capture the position of the users hands. I performed simple piece-wise constant upsampling to the Kinect data, so that the low frame rate is evident.

As seen in (Figure 3.6), the Kinect displays a significant amount of latency. A small amount of this could be attributed to the data transport, but the slow frame rate makes it nearly impossible to detect sudden events like the whack of a mallet.

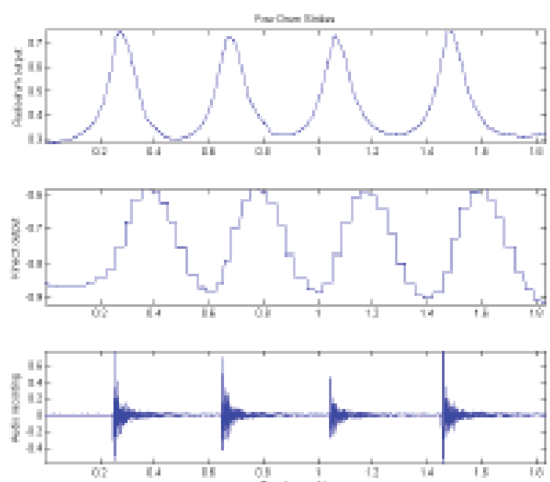


Figure 3.7: Captured motion of four drum strikes

Although small temporal changes will not be detected by the Kinect, capturing slower movements is still possible. The following plot (Fig.3.7) shows four discrete hits of the drum. Although the Kinect would not be able to use this information to produce a responsive sound immediately, it is still perform beat detection to determine the tempo a performer is playing at. However, it is already a sufficient alternative to the traditional fader style mapping parameter, as we have shown in our use here for this example.

Range of Sensing

The choice of mapping for gestures onto or into audio data has also been a source of significant attention. How much apparent change should a movement produce?

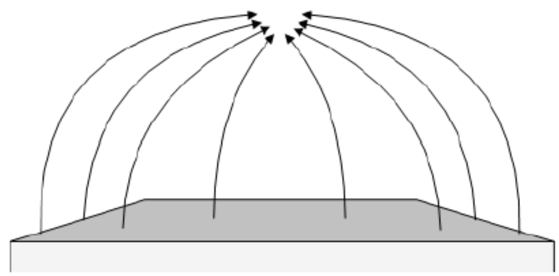


Figure 3.8: Radiodrum viewable area

First, I examine the range of motion for both sensors. The Radiodrum will only return consistent position data while the drum sticks are in an area close above the surface of the sensor(Fig. 3.8). It will also tend to bend the values towards the center as the sticks move farther above the surface.

Perspective viewing gives the Kinect a much larger viewable area . The Kinects depth sensors field of view is 57 degrees in the horizontal direction and 43 degrees in the vertical direction. This means that at closer ranges, the Kinect cannot detect objects far to the sides of the camera whereas when depth is increased, objects far from the center of view may be detected (Fig. 3.9).

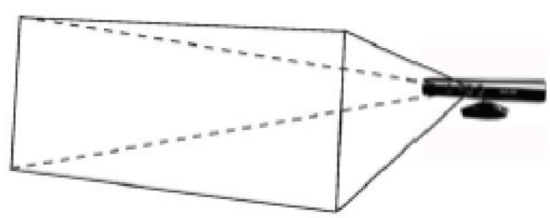


Figure 3.9: Kinect viewable area

To demonstrate the constriction on possible movements recorded by the Radiodrum, I recorded the captured output of a user moving their hand back and forth while holding the Radiodrum stick. The Kinect is able to capture a much larger range of gestures (Fig. 3.10).

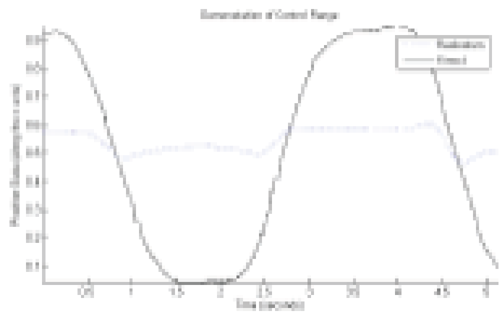


Figure 3.10: Horizontal range of Radiodrum vs. Kinect

One of my goals is to go beyond simple static one-to-one mappings and incorporate Machine Learning for gesture recognition. While one-to-one mappings are by far the most commonly used, other mapping strategies can be implemented. It has been shown that for the same gestural controller and synthesis algorithm, the choice of mapping strategy became the determinant factor concerning the expressivity of the instrument [22]. I have been actively developing a systemized mapping strategy for many years and will use this as an apparatus to inform and evaluate my advancements with the Kinect.

Another scenario that is in development but has yet to be implemented involves extending the predicted movements from the Kinect to include accurate tracking of the tips of a mallet. Verifying the accuracy and precision of mallet tip position data acquired from both the radio drum and the Kinect would also be vital when effectively modeling of an interactive virtual marimba. As previously established, the Radiodrum is accurate near the surface of the instrument, but position data veers towards the center of the rectangular surface as the tip is moved away. Preliminary measurements ranging from half a meter to a full meter away from the plane containing the Kinects cameras demonstrated that depth values increase linearly (R2 = 1) along a straight line perpendicular to said plane. However, the slope of the line was not 1. Rather, it varied from 1.01 to 1.05. In cases where only the relative change in position matters, such a value would not strongly affect results. When the absolute position is desired, as in our application, it means the deviation from the actual value is as much as 5 cm at a depth of 1 meter. Another potential area of exploration involves comparing the three-dimensional coordinate measurements from both the Radiodrum and the Kinect with a ground truth and attempting to compensate for the disparity.

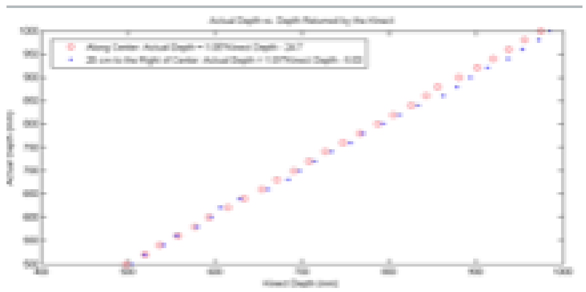


Figure 3.11: Kinect depth variance

This shows that there is significant latency and temporal jitter from the Kinect data relative to the signals from the Radiodrum making direct fusion of the data difficult except for slow movements (Fig. 3.11). One potential way to help resolve this issue is to extend the body model to include more of the physics of motion. The current body model is largely based on just the geometry of segments (a kinematic description) whereas a full biomechanical model would include inertial (kinetic) parameters of limb segments as well as local limb acceleration constraints. Once the biomechanical model is initialized with some slow moving data, it can be used to predict (feed-forward) the short term future motion and then the delayed motion data from the Kinect can be used to make internal corrections (feedback). Also, because internally the biomechanics body model will have estimates of limb segment accelerations, it would be relatively easy to incorporate data from 3D accelerometers placed on important limb segments (such as the wrist) to enhance motion tracking.

The Kinect sees the world from a single view and so although it produces depth information, this can only be for objects closest to it (in the foreground), all objects in the background along the same Z axis are occluded. This results in occlusion shadows where the Kinect sets these areas to zero depth as a marker. These regions appear as shadows due to the area sampling of the coded-light algorithm used to estimate scene depth. The use of a pair of Kinects at +/- 45 degrees to the performance capture area has the potential to make the motion capture much more robust to occlusions and should improve spatial accuracy. Practically the IR coded-light projectors on each Kinect would need to be alternated on and off so they dont interfere. If the timing is synchronized and interleaved, the effective frame rate could be potentially doubled for non-occluded areas.

General free motion (motion with minimal physical contact) is usually imprecise in absolute terms as it relies on our proprioception (our internal sense of relative position of neighboring parts of the body) combined with vision to provide feedback on the motion. The exception to this is highly trained free motion such as gymnastics or acrobatics. Physical contact with a target reduces the spatial degrees of freedom and provides hard boundaries or contact feedback points. In the absence of this hard feedback, gestures performed in free space are going to be difficult to automatically recognize and react too, unless it is highly structured and trained. This requirement could significantly distract from the expressive nature of a performance. Because the proposed future system will include a biomechanical body model of the motion, it should be possible to predict potential inertial trajectories (gestures) in real-time as the trajectory evolves. This would allow the system to continuously predict the likely candidate match to the gesture and output an appropriate response. The trajectory mapping will need to be robust and relatively invariant to absolute positions and rotations.

My Kinect instrument is compatible with virtually all MIDI hardware/software platforms so that any user can develop their own custom hardware/software interface with relative ease. Externals can be developed in Max/MSP for use in Max4Live and will enable anyone to plug in a Kinect and adapt this platform for 3D sound control to their own practice with minimal configuration. This has many implications, putting technology that previously was restricted to research purposes into the common musicians hands. My belief is that these new instruments have a legitimate place [136] with potential to become part of an embedded conventional musical practice, not just a research curiosity. While hyperinstruments might seem niche or esoteric at this point [61], historic innovations such as the leap from monophonic to polyphonic music, electrical amplification of the guitar, and computers in the recording studio all brought skepticism, eventually becoming mainstay practices.

Non-invasive Sensing

All systems described in this study were implemented with the aim of being used in a computer music environment such as Max/MSP[1] and Ableton Live[2]- what is felt to be the most ubiquitous, comprehensive, and intuitive audio computing platform

[1] https://cycling74.com/
[2] https://www.ableton.com/

for performance, to date. At the time of this work, the computer vision system used the openCV5 and openFrameworks6 libraries. It is compiled as a standalone program that yields Open Sound Control (OSC) messages. The gesture repetition algorithm described in Chapter 4 was implemented as a patch for Max/MSP. This allows the user to design the exact means by which each mode will be triggered the user may require a specific button to trigger each mode, or simply a button that cycles through all modes, or even some other interface that makes more sense in an specific context. The patch also offers, as feedback, the current value for the maximum autocorrelation coefficient and the index of that value when both remain more or less constant for some time, it means that the system has acquired a certain gesture. Last, the drum pattern detection algorithm also described in Chapter 4 was implemented using the Max4Live framework as well.

This work presents different ways of augmenting pitched percussion instruments such as the marimba, vibraphone and thumb piano with digital control capabilities. Using these approaches the instruments can be turned into hyperinstruments without requiring invasive additions to the instrument. In retrospect regarding the usability of the interfaces in performance, the virtual faders are responsive and accurate to the point that using the mallet tips on the surface of the bar proves a sufficient and intuitive fader style interface. The technique required is inspired by the acoustic pitch bend extended technique used by expert vibraphonists. The system could be improved by adding vertical detection of the mallet tip to engage/disengage rather than only a time based approach. This would solve the problem of accidental activation when playing on a single bar long enough to engage it's respective fader. The two methods of gesture control based on repetition and rhythmic patterns offer effective ways to transmit control data while holding the mallets. Issues with calibration and mappings could be assisted with a more comprehensive GUI. While the system can stand to be improved, it has proven to be effective in performance, reliably within its known limitations on stage. In future work, further improvements on the usability and effectiveness of the proposed algorithms will be implemented.

3.1.2 Lamellophone Based

A) WIIKEMBE:

Typically hyperinstruments require specialized engineering for the development of proprietary prototypes. Because the field of hyperinstruments is relatively young, standardized systems need to be designed from existing paradigms (OSP). Such paradigms are typically singular in nature, solely reflecting the idiosyncrasies of the artist and often difficult to reproduce. This work simplifies this process, offering a basic design prototyping platform.

The Wiikembes (Fig. 3.12)design facilitates uninterrupted traditional playing while the sensors can be engaged with broad or minute arm/hand gestures or explicitly with the free fingers. The unused surface area of the lamellophone body is ideal for intuitive sensor placement, while direct audio acquisition is simple and robust via piezo transducer. These augmentations equip the instrument for contemporary settings where audio processing and sound reinforcement are commonplace. Percussionists and EA musicians typically utilize their feet in various ways, so there is an included simple custom built Arduinio/Puredata footswitch described- its construction/implementation are trivial.

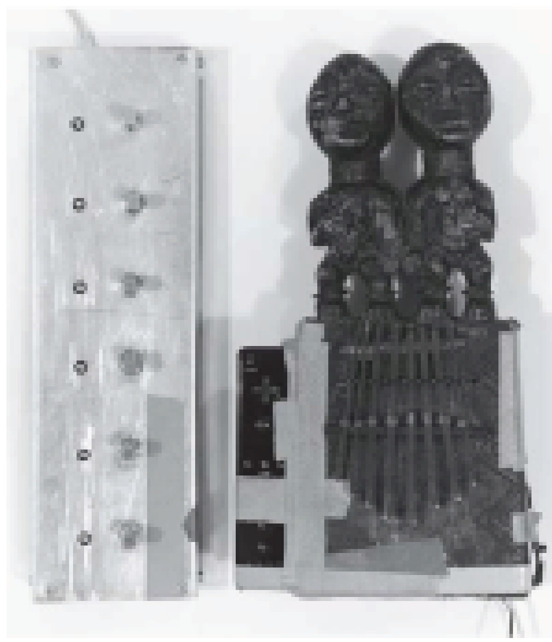


Figure 3.12: Wiikembe and custom footpedal

Like Konono No. 1, the Wiikembe uses easily sourced, low cost components. The design is meant to be easily replicable and modular so that any lamellophone performer could render their own version with little engineering background. The Wiikembe is an augmented Likembe: a Wiimote, 2 rotary potentiometers, 1 slider potentiometer, 1 membrane potentiometer [47] and a piezo transducer[3]. The sensors input into the analog GPIO of an Arduino[4] Nano then bussed to Pd[5] via serial USB and converted into MIDI, then mapped into Ableton Live for control of filter parameters, volume levels, and audio event sequencing. The Likembe audio is acquired via piezo transducer and input into the computer as a discrete mono signal. A flowchart of the configuration can be seen in (Fig. **??**).

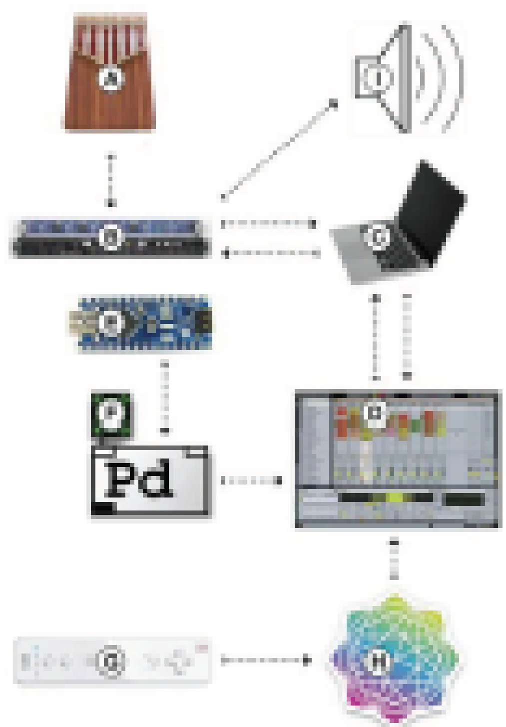


Figure 3.13: Wiikembe system overview

Wiikembe Components

1. Likembe

The Likembe used for this study is made from wood which is hand carved with an effigy at the top for ornamentation. The tines are mounted to the body in the conventional method with a bridge. Additionally, small metal rings have been added

[3]https://www.contactmicrophones.com/
[4]https://www.arduino.cc/
[5]https://puredata.info/

to the tines to create a desired sustaining buzz when the tines are plucked. This is typical of many traditional African instruments [14]. The body has been carved out on the inside creating a resonant chamber. The instrument is tuned to a single octave diatonic scale. Since the instrument is non-western, and therefore doesn't conform to the tempered chromatic scale, the suggestion of the octave and corresponding tones are only relative to each preceding note when ascending the scale regarding tuning. Meaning, the instrument is tuned by ear and not by mathematically uniformed spaced intervals based on Hz [96]. The exact frequencies of each tine on this specific Likembe can be reviewed in [133].

The hand carved, wood Likembe has tines mounted to the body over a bridge with small metal rings added creating a desired sympathetic, sustaining buzz when plucked (Fig. 3.14)- a sound design practice common in West Africa [33].

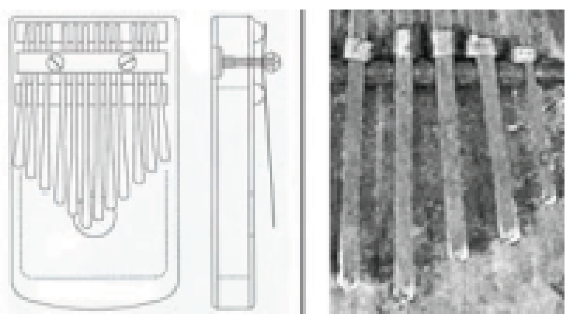


Figure 3.14: Likembe tines with rings

The body is carved out on the inside creating a resonant chamber. The instrument is tuned to a single diatonic scale by adjusting the length of the tine (longer = lower frequency/shorter = higher frequency- see Fig. **??**). Being non-western, it doesnt conform to the tempered chromatic scale. The suggestion of a tonic/intervals or octave is only relative to the notes preceding it and their context within the music- since the instrument is tuned by ear and not by mathematically and uniformly spaced intervals [96]. (Table 3.1) shows the approximate scale of the instrument.

2. Wiimote

The Wiimote offers versatility in a music controller [149]. Combining wireless 3D position tracking [74] with switches, the Wiimote extends the Likembe, affording the performer a broader set of musical gestures not possible with only potentiometers. The Wiimotes design situates transparently with the ergonomics of the Likembe. The

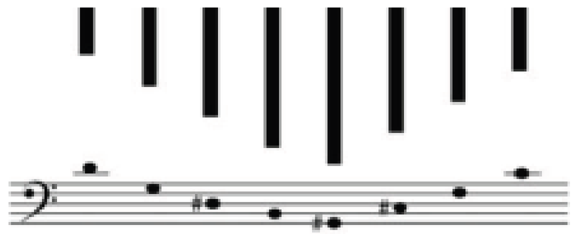


Figure 3.15: tine layout/approx. A=440

Tine	True Frequency (Hz)	Nearest Pitch Range (note/Hz)
1	213	G#3 = 207.65
2	247	B3 = 246.94
3	272	C#4 = 277.18
4	308	D#4 = 311.13
5	352	F4 = 349.23
6	383	G4 = 392
7	540	C#5 = 554.37
8	624	D#5 = 622.25

Table 3.1: Likembe tuning

top switches are easily accessible via the thumb, while the switch on the underside is accessible via the forefinger, both with minimal adjustment to conventional playing position. The position tracking allows the performer to interface DSP parameters without interrupting playing in the way having to turn a knob might. This presents an level of interaction with software FX, synthesizers, and more that would otherwise require interruption of playing the traditional instrument in order to engage the filter parameters.

3. Arduino and Sensors

An Arduino Nano is mounted to the back of the instrument (Fig. 3.17). Four dedicated sensors send CC data into Pd via Pduino [121]. Pd is used because it is free, widely supported, and relatively easy to prototype in.

This Likembe has a groove carved out of its back which allows the potentiometers (pots) to sit comfortably recessed in the instruments body. The pots are mounted to prototyping board alongside a low-profile slider. Situated on the instruments rear, the sensors are out of the way, yet easily accessible from a natural playing position. They fit comfortably within the space confines as dedicated controllers specific for filter

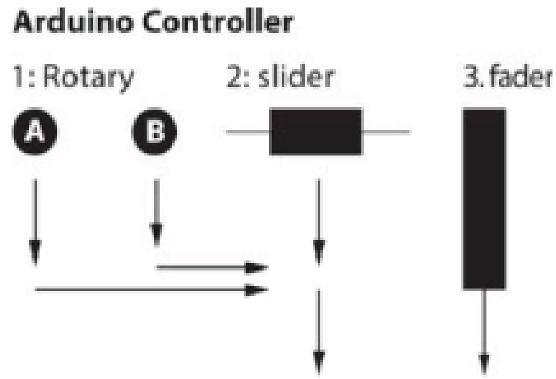


Figure 3.16: 1.A-rvrb, B-dly; 2. FX vol.; 3. Wiikembe vol.

parameters meant for this type of interaction [51]. Lastly, a membrane potentiometer is mounted on the right side of the instruments body. Nearly as thin as paper, it is minimally invasive to the Likembes traditional playing techniques, yet remains easily accessible via the performers index finger [51]. The sensor data from the Arduino is converted into MIDI in Pd and routed into Ableton Live.

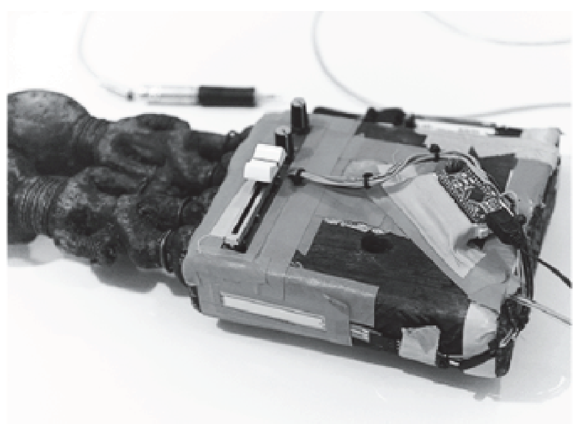


Figure 3.17: Arduino controller

4. Software

Osculator[6] was used to receive and convert the Wiimote data into OSC or MIDI messages to be mapped to the users software or hardware of choice. Osculator is flexible, inexpensive, robust and intuitive (no longer supported). It streamlines the

[6]https://osculator.net/

sensor data acquisition, scaling and mapping stage of working with the Wiimote. It is also easy to use and reliable. In this implementation all data is sent as MIDI data into Ableton Live and mapped to filter parameters and device controls (Fig. 3.12). Live is used as a sequencing, looping, effects processing, and audio mixing environment. Audio from the instrument is acquired from the contact mic via the audio interface. The signal has a dedicated channel in Live and passes through a frequency shifting delay plug-in (Fig. 3.18). In this case, the free Valhalla DSP Frequency Echo[7] and the Ableton Looper device respectively. The signal is output mono via Output Channel 2 on the audio interface to its own speaker. The Wiimotes gesture tracking is mapped to: (Table 3.2); and the Wiimotes switches control: (Table **??**).

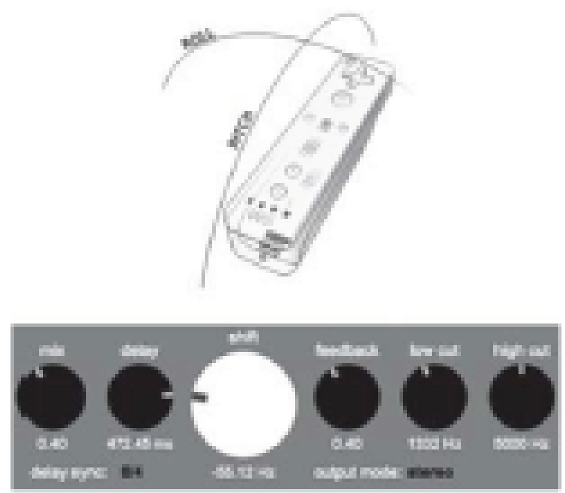


Figure 3.18: Wiimote axes/filter parameters

Axes	**Mapping**
Pitch+	freq. shift up
Pitch-	freq. shift down
Yaw+	delay rate8
Yaw-	feedback rate
Roll+	filter wet
Roll-	filter dry

Table 3.2: 3D gesture mappings

Mappings are such that he Wiikembes volume is controlled by the membrane

[7]https://valhalladsp.com/shop/delay/valhalla-freq-echo/

potentiometer. Two rotary knobs control auxiliary sends A and B to: Send A- Reverb; and send B- Dimple Delay (native Live devices). Both aux channels are bused to the master channel, whose volume is controlled via the slider potentiometer (Figure 3.16). The effects chain outputs a mono signal from Output Channel 1 on the interface to its own speaker. These explicit mixing controls are essential to computer music performance of any style, offering spatiality to an otherwise 2D acoustic experience.

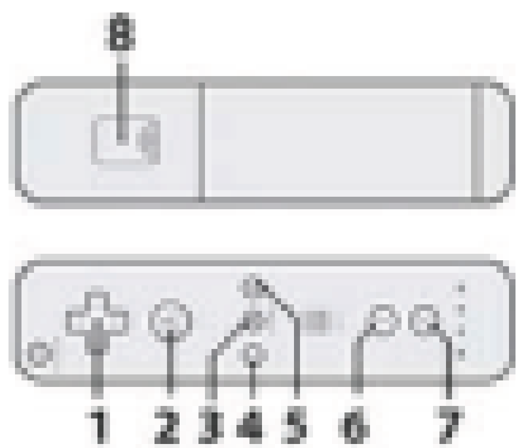


Figure 3.19: Wiimote switch labels

The Arduino interface is designed to afford the performer the ability to mix multiple audio signals from the instrument itself with minimal adjustment of technique or interruption of performance. These specific mappings are mostly arbitrary, except for reflecting the artists performance idiosyncrasies and personal mapping preferences. Any artist can map the controllers to whatever parameters they like. However, it is worthy to note that the looper device in Live is configured so that one button press activates the loop recording, loop completion, then playback[8]. A double press starts and stops play, respectively. The button chosen for this function is nearly at the location of the thumb in normal playing position and makes for a very intuitive interface for this application versus a conventional looping pedal paradigm or external controller (Fig. 3.19). The 3D tracking allows the player to interface parameters typically reserved for traditional knob turning which forces one to disengage from instrumental practice in order to manually interface a separate hardware device. Ben Niell shows an example of mounting knobs to a trumpet embedding the knob interface on the

[8]https://www.ableton.com/en/manual/welcome-to-live/

instrument itself, making for a more intuitive interface [91]. The Wiikembe goes beyond this by turning the whole instrument into several different knobs based on the axis of motion simply by mounting a 3D position-tracking sensor to the instruments body.

Switch	**Key Code/MIDI**	**Mapping**
1- up	key 5	loop double speed
1- down	key 2	loop half speed
1- left	key 3	loop undo
1- right	key 4	clear loop
2- A	key 6	view looper device
3- home	MIDI note F2 ch 7	tap tempo
4- minus	key 1	loop reverse
5- plus	MIDI note F2 ch 12	freq. filter on/off
6- 1	MIDI note F2 ch 5	track mute
7- 2	MIDI note F1 ch 5	track arm
8- B	spacebar	global transport

Table 3.3: Wiimote: switch mappings

B) EL-LAMELLOPHONE (El-La):

Components

The design is adaptable, easily replicable [69], and open-source so that any lamellophone performer could render their own version with little engineering background. El-La (short for Electric-Lamellophone) builds on the same system as the Wiikembe and expands those contributions (Figure 3.20).

The sensor interface is custom built incorporating a capacitive touch sensor controller, a three-axis gyroscope, a force sensitive resistor (FSR), 6 switches, 2 rotary Potentiometers, a membrane slider potentiometer, an x/y axis joystick, and LEDs for visual feedback. To facilitate easy sensor acquisition, all sensors interface with an Arduino Nano. The capacitive touch sensor controller interfaces with the Arduino using an I2C interface, while the IMU outputs all sensors processed by its onboard AT-mega328 via a serial stream (UART). All other sensors (buttons, pots, joystick, and FSR) interface by way of the Arduino Nanos GPIO. The Nano plugs directly into and is powered by the Beaglebone and communicates via USB. The whole system is reconfigurable, easily removable and portable (Figure 3.21).

Figure 3.20: El-lamellophone

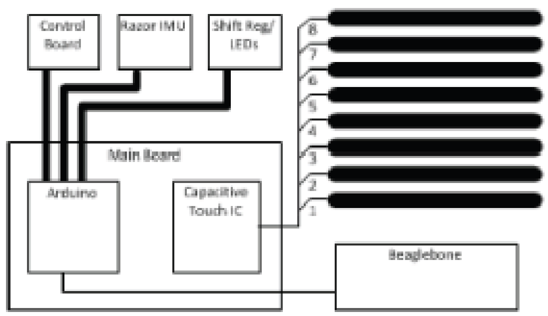


Figure 3.21: Modularity of components

1. **Interface**

Piezo

A contact microphone, mounted on the body of the El-Lamellophone, is connected straight to the 1/8" audio jack of the Beaglebone audiocape[9] to acquire the audio signal. The signal is then processed in Pure Data.

Arduino

The Arduino Nano is a small, complete, and breadboard-friendly board based on the ATmega328 (Arduino Nano 3.0) and works with a Mini-B USB cable 1 (Figure 5.21)

[9]https://www.element14.com/community/docs/DOC-67906/l/beaglebone-audio-cape

Sensors

The HCI components of the El-Lamellophone are strategically located so that they don't interfere with the performer's playability of the instrument [114].

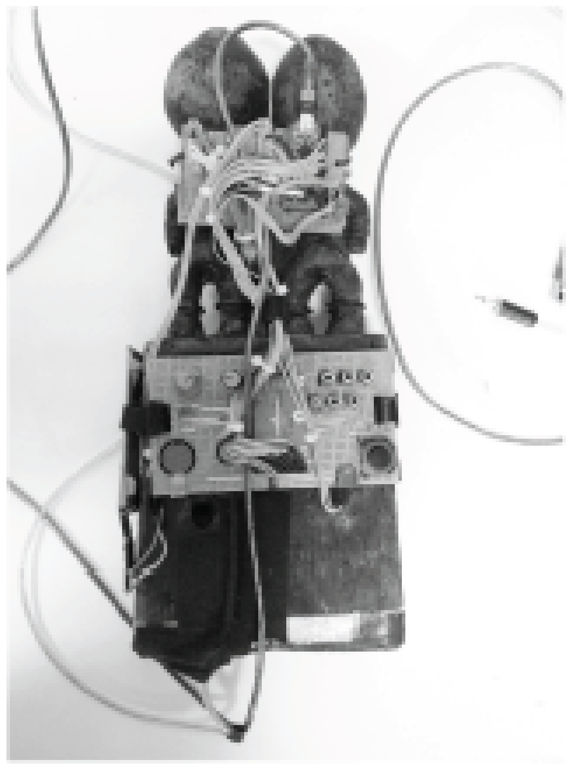


Figure 3.22: Mounted system

Switches, Rotary and Membrane Potentiometers: Six Pushbuttons and two (47k - the value chosen was arbitrary) rotary potentiometers are mounted on the upper back panel- accessed by the left and right index finger respectively. One Force Sensing Resistor (FSR) is located on the lower right front panel, below the tines, easily reachable by the right middle finger. The value of 2k was chosen for the resistor in series with the Force Sensitive Resistor (FSR) by connecting the FSR circuit to the gain CV of an analog synthesizer while viewing the output voltage on an oscilloscope. This value gave the best compromise between music expressiveness and full-scale reading on the oscilloscope. Also mounted on the back panel is one XY resistive analog joystick accessible by the left middle finger. On the side panel one SoftPot resistive membrane is mounted. The pushbuttons connect to the digital pins of the Arduino board, while the rest of the components connect to the analog pins(Figure 3.23).

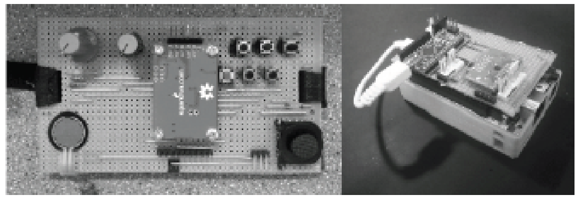


Figure 3.23: Sensor interface (l); Beaglebone (r)

Capacitive Touch: For this implementation 20 AWG wire is used soldered to the eight tines on the El-Lamellophone as electrodes. Each wire was then connected to the MPR121 Capacitive Touch Sensor Controller[10] from Freescale. This package is capable of controlling twelve electrodes and interfaces via I2C protocol. It also includes a three-level filtering process to eliminate any noise-induced detection. The maximum response time is 32usec. and its output byte contains touch/release status for all tines. An advantage of this package is that it allows simultaneous touch detection of the eight tines, which in this case is used for polyphony.
9DOF: For the gyroscope implementation an InvenSense ITG 3200 three-axis (X (roll), Y (pitch) and Z (yaw)) gyroscope with I2C interface is used. This small package (4 4 0.9 mm) digitally outputs rotational data from any of the three sensed axes at a rate of 20ms. In the current implementation the gyroscope is built into a 9DOF Razor Inertial Measurement Unit (IMU)[11] via I2C protocol. The IMU's firmware scales the 16-bit output range of the gyro- scope into a [180.00, 180.00] range for each axis and then transmits the rotational data to the host Arduino board via the serial ports Tx and Rx, at a baud rate of 56700 bps. (Figure 3.24) shows the direction of rotation of each of the sensing axes, in reference to the El-Lamellophone. In the future a cheaper platform for prototyping could be explored, this model was readily available at no cost so it was used. Lamellophone based gesture-sensing using an IMU builds on the Wiikembes contributions [133].

Visual Feedback

A 10-segment LED bar graph is used to provide state and sensor feedback to the EL-lamellophone user. To economize the wiring footprint and then number of Arduino pins needed, two 8-bit Shift Registers are used to control the LED bar graph.

[10]https://www.sparkfun.com/datasheets/Components/MPR121.pdf
[11]https://www.sparkfun.com/products/retired/10736

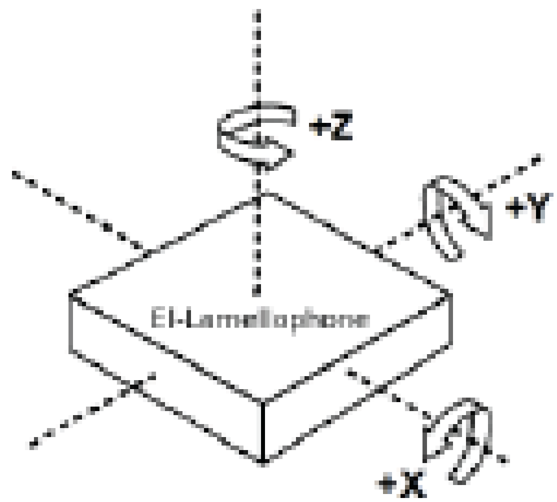


Figure 3.24: 9DOF axes of rotation

Firmware

The Arduino firmware is simple: each loop an output vector of floating point numbers is populated with the measurement from each sensor, and then sent to Pure Data using a serial connection. First, all analog sensors are read and stored in the first section of the vector. The Capacitive Touch Sensors states are stored in the next portion of the output vector. Finally the IMU's serial stream is parsed and stored at the end of the output vector. The output vector is now fully populated and ready to be transmitted to Puredata. From the Arduino's perspective, transmission of the output vector to Pure Data is accomplished by writing the entire vector to the serial port one value at a time, with a newline character signifying the end of the vector.

2. Host Computer and Software

Beaglebone

Significant previous work has been done by the author using this autonomous music-computing platform featuring the Beaglebone. Technical specs (including the performance of the DSP latency) can be seen in [79] and described in detail in Chapter6.

Puredata

The DSP environment is built in Puredata (Pd). Pd was chosen because it is free, widely supported, flexible, and robust. Serial data is received from the Arduino and then scaled and mapped to control specific effect parameters (delay length, feedback level, filter frequency, etc.). This simple prototyping system is robust but offers a lot of flexibility and potential. The sensor data can also provide an intuitive control interface for synthesis applications including controlling oscillator frequency, filter frequency/bandwidth, MIDI control/note attributes, and envelope values (attack, decay, sustain, release) (Figure 5.27). This application allows El-la to be used as a versatile synthesizer controller while the lamellophone can still be played as usual- all housed in one unit.

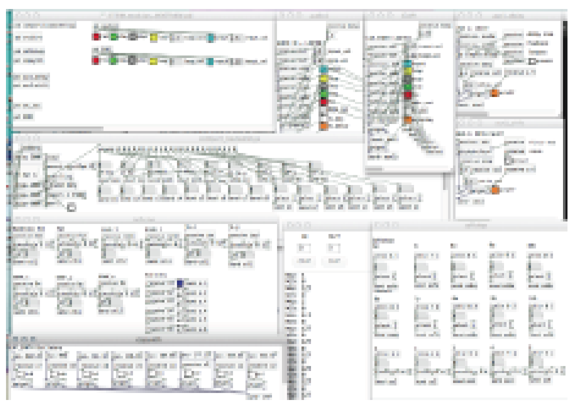


Figure 3.25: Puredata patch

Comport and Sensor Data Parsing: All of the sensor data is transmitted to Puredata by way of a serial connection with the Arduino. The comport object provides a serial port interface within Puredata, making all sensor measurements available. The incoming ASCII bytes are then converted to numerical values.
Audio Analysis for Control Data: Audio analysis of the input signal is accomplished to provide an amplitude envelope tracking system. RMS energy is thresholdeded to a usable range, and energy levels are displayed as they occur. This provides a simple means for visualizing the rise and fall of the total signal energy, and any further thresholding control can be added as needed. This data can be used to create dynamic, audio responsive control data for other devices.

3.2 Gyil Gourd Physical Modeling Synthesis

A physical model for the Gyil, an African pentatonic idiophone with wooden bars having a similar sonic characteristics to the western marimba is proposed. This project was done in collaboration with Mathmatics Post-Doctoral research fellow Dr. Dan Godlivitch. Here my role was as design engineer, where I contributed towards high-level theoretical design concerns, sensor implementation and mapping strategies. The primary focus is modeling the gourds that are suspended beneath each bar having a similar role to the tuned tubes below the bars in western mallet instruments. The prominent effect of these resonators is the added buzz that results when the bar is struck. This type of intentional sympathetic distortion is inherent to African instrument design as it helps unamplified instruments be heard above crowds of people dancing and singing. The Gyils distortion is created by drilling holes on the sides of each gourd and covering them with membranes traditionally made from the silk of spider egg casings stretched across the opening. By analyzing the sonic characteristics of this distortion it is found that the physical mechanisms that create it are highly nonlinear, as such it is desirable to model them computationally. In addition to the fully synthetic model I consider a hybrid version where the acoustic sound captured by contact microphones on the wooden bars is fed into a virtual model of the gourd resonators. This hybrid approach simplifies significantly the logistic of traveling with the instrument as the gourds are bulky, fragile and hard to pack. I propose several variants of the model, and discuss the feedback received from expert musicians.

3.2.1 Model Description

This work introduces a physical model for the Gyil, an African mallet instrument with a unique system of resonator gourds mounted below wooden bars. These gourds are unique in that they have holes drilled in them, which are covered over with membranes. These membranes react to sound pressure in a highly non-linear fashion and produce a buzzing sound when the bars are played with force.

Physical modeling efforts have typically focused on modeling stringed instruments, struck bars, and membrane models usually in the context of western classical music instruments. To the best of my knowledge, the Gyil, and the nonlinear processes which are the cause of its characteristic sound, have not been studied before in the context of physical modeling synthesis.

One objective behind creating the model is to provide performers and composers an audio effect that can be applied to the signal of any mallet percussion instrument. This enables the use of techniques associated with the Gyil without having to include the actual gourds and membranes. The gourds are cumbersome, fragile, difficult to construct, and can not be added to conventional pitched percussion instruments. Electro-acoustic mallet percussionists have little choice but to apply filters and audio effects that are designed for other instruments such as guitar pedals. In contrast, my model and its associated audio effects are idiomatic to pitched percussion instruments.

3.2.2 Gyil: Physical Measurements

Each individual Gyil is unique, as it is constructed using natural materials and is tuned by ear by its builder. We present physical measurements made using a specific Gyil with the goal of obtaining reasonable ranges for the parameters of the proposed model. The synthesis of the wooden bar sound can be performed using standard modal synthesis techniques [6] and therefore the focus of the measurements was to obtain information about the prepared gourds. The resonant characteristics of each gourd depend on its geometry, so we measured the width, the height, and the radius of the mouth of all the gourds for the specific instrument considered. The buzzy sound that is a characteristic of the Gyil is created by several membranes that are attached to the gourd so their number and radius were also measured. The measurements are shown in (Table 3.4).

Parameter	**Typical Values**
Gourd width	5cmm - 25cm
Gourd height	5cm - 40cm
Mouth radius	1cm - 5cm
Membrane radius	.5cm - 1cm
Number of membranes	2 - 5

Table 3.4: Dynamic gourd dimensions

The gourds are natural objects and therefore there is no consistency in their exact shape. As seen in (Figure 3.26), their shapes can be very different making it practically impossible to model them in detail. The same holds for the spider silk egg casings used in the membrane. For that reason, the sonic characteristics of the gourd were evaluated by comparing two different recordings: one of a note of the Gyil

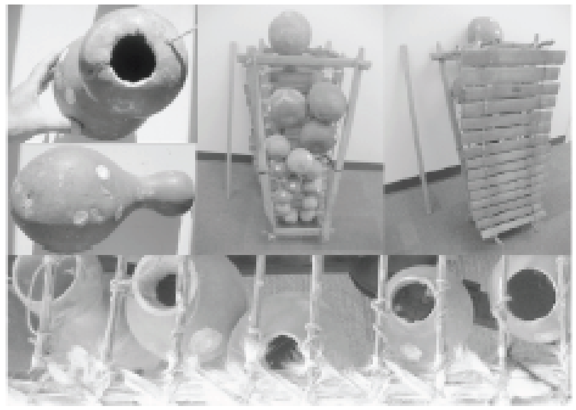


Figure 3.26: Gourds/frame construction

without the gourds (that is, a sound similar to the one of a xylophone) and the other one of the same note of the same Gyil, with the gourds reattached. The Discrete Fourier Transform of the first 300 ms after the attach was calculated and is shown in (Figure 3.27). The energy present below the fundamental is likely to be caused by intermodulation distortion.

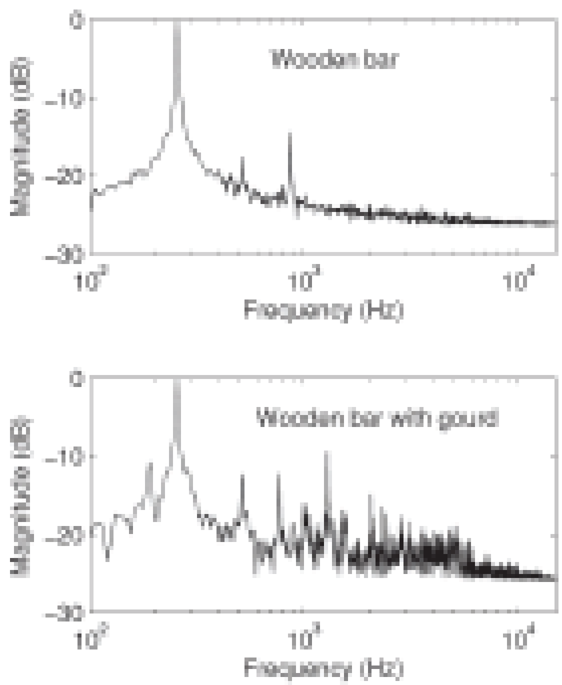


Figure 3.27: DFT: resonating wooden bar results

As can be seen, attaching the gourds to the Gyil introduces non-harmonic frequency components around 1.2 kHz and 1.8 kHz that indicate non-linear behavior. These frequency components cause the characteristic buzzy sound of the Gyil. In the next section a technique for obtaining this type of distortion is described.

The instruments used were built in 2002 for the author while on a six-month research visit to Ghana. They were crafted by the late master builder/performer Kakraba Lobi[12]. The two instruments were built as a pair and tuned to a western G major pentatonic scale. This is unorthodox, as the instruments are usually tuned by ear. The one Gyil was kept intact, while the gourds were removed from the other one for hybrid synthesis.

Model Description

A model of the Gyil gourds that produce the characteristic buzzing sound of the instrument is presented. The gourds are treated in isolation from the wooden bars. This is justified by the fact that the ratio of the mass of the bars to the sound pressure levels produced is large and the gourds are not precisely tuned to resonate at the frequency of the bars so it can be assumed there is no feedback of acoustic energy from the resonators to the bars. Consequently, the Gyil is treated as a source-filter system, and focus modeling efforts on the gourds, and in particular aim to simulate the non-linear response of the vibrating membranes as well as the filtering characteristics of the gourd.

In constructing a mathematical description of the Gyil, two options present themselves: a model can be built that is a direct representation of the understanding of the physics of the system, or we can abstract and idealize the processes at work, representing them by algorithms which perform a similar function to the various components of the instrument. The first option has the advantage of the ultimate degree of realism available, however it can lead to highly complex numerical models which are unusable in real time. In particular, even a simplified mathematical description of the physics of the membranes would require a high degree of complexity due to the irregular shapes of the membranes, the unique material properties of the egg casing, and the coupling between the egg casings and the resonating surface of the gourd, which is itself highly irregular. The second option is to represent the processes using signal processors which replicate the effect of the process. In addition to the inherent nonlinearity of the system, each Gyil is built by hand, using the available materials

[12] https://www.discogs.com/artist/1537170-Kakraba-Lobi

(e.g. wood, gourd shells, and spider silk), that the builder judges to be most suitable. In the course of construction, the instrument is tuned by ear by the craftsman. Under these circumstances, it makes little sense to attempt to create an exact model of a particular Gyil. Instead, I seek to emulate the sonic features identified as characteristic of the instrument. The proposed model is based on a general structure with adjustable parameters that are set by the user for fine tuning; I propose that this process is analogous to the decisions made by the craftsman when building the physical instrument.

The proposed system consists of a model for the gourd itself which is excited with either a signal generated by a modal model [6] of the wooden bar, or by audio acquired from the wooden bars using either a microphone or contact microphones [113]. The modal model used in this study was created in the Reaktor[13] programming language, and was tuned to the sound of the xylophone recordings by ear and by comparison with spectral plots of the recordings. The modal model used 6 partials above the fundamental, and was excited with a filtered impulse mixed with a enveloped burst of white noise, in order to simulate the sound of the mallet strike on the bars.

The gourd is modeled as a resonant filter [99] and the membranes are modeled as a non-linear function that yields the Gyils buzzing sound. The output of the membranes is heard both directly and after being filtered again by the gourd. This leads to the model structure seen in (Figure 3.28).

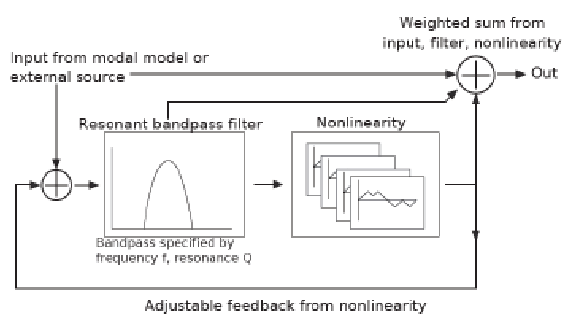


Figure 3.28: Signal flow for one gourd

The center frequency of the resonant filter model associated with a gourd is inversely proportional to the volume of the gourd. For that reason, larger gourds are chosen to be associated with the lower pitched notes. The filters Q-factor Q is inversely proportional to the size of the opening at the top. Due to the natural shape

[13]https://www.native-instruments.com/en/products/komplete/synths/reaktor-6/

of the gourds used, the resonators hung below the higher bars tend to have wider apertures, and the model can be tuned to reflect this.

Modeling the Membranes

The membranes present a non-linear behavior that is difficult to quantify, as there are no precise measurements for the characteristics of the specific spider silk used in the papering of the gourd holes. While some Gyil makers use varieties of paper as a replacement for the silk, the rheology of paper membranes is an unexplored field, and there are few physical results on which to base the investigation. It is reasonable to assume that the membranes respond linearly to low-intensity acoustic signals, but are only able to vibrate within a certain range of amplitudes, hence clipping the input signal in levels A+ and A- as in: Uneven tension on the membrane, coupled with the material properties can lead it to fold back on itself (crinkle) when forced above a certain amplitude A. I represent this behavior using a wave folder nonlinearity. Other options which have been considered and tested are a cubic nonlinearity, and a sine waveshaper. Preliminary testing showed that the cubic waveshaper did not generate sufficient harmonic content to begin to emulate the behavior of the membranes, and the sine shaper performed similarly to the wave folder, but with a higher computational cost. Due to the position of the membrane on the outside of the gourd, its motion may be impeded (Figure 3.29) when the pressure inside the gourd is drawing it inward. This behavior is included by making the wave folder asymmetric.

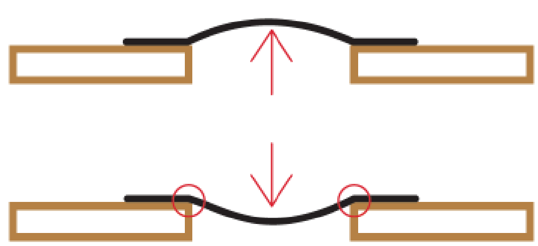


Figure 3.29: Membrane's asymmetric motion

The output signal from the wavefolder may be expressed in terms of the input signal as: where xout is the output, xin is the input, and A+ (A-) is the amplitude at which the wave is folded over for positive(negative) amplitudes. The wavefolder produces a signal with more high-frequency energy than a simple clipper due to a higher level of inter-modulation distortion. To implement asymmetric behavior in the model, a symmetric wavefolder with parameter $A = A+ = A$ and add an offset c to the input signal prior to applying the non-linear function is implemented. Adding an offset will change the energy of the harmonics added to the output. The two parameters, A and c , that define the behavior of the membrane model are respectively connected to the age of the membranes, as old membranes tend to require more energy to trigger the non-linear behavior, and to the asymmetry of the membranes construction, which is determined manually by the craftsman. Since there is a great deal of variability between individual membranes, both in material properties, size, and position, the level of asymmetry in the wavefolding is varied between them. In order to explore a range of possible parameterizations of the behavior of the membranes, three different algorithms are tested: a simple clipper, a simple wave folder, and a series association in which the output of a wave folder is clipped. These algorithms are compared in Section 3.2.3. As typically there are several membranes (between 3 and 5) attached to each gourd we run several membrane models (with N being the number of them) in parallel. To mimic the different sizes of the membranes, and variations in material, the non-linearities must have different parameter settings for A and c. Although these differences could be set manually for each membrane, that would imply adjusting $2N$ parameters. To simplify this tuning process, an adjustable scaling parameter $0 < s_w < 1$ is specified so that, for a model with N membranes, the level A related to the non-linearity in the nth membrane is given by $s_n^w A$, and the offset level c is proportionally adjusted so as to keep the asymmetry constant, thus reducing the number of adjustable parameters for the wave folders from $2N$ to 3.

Signal Flow

Using a resonant filter to model the gourd, and clipping and wavefolding to model the behavior of the membranes, the building blocks of the Gyil gourd model are established. To complete the gourd-membrane model, the output of the membrane models to the gourd are coupled by summing the membrane outputs and feeding them back into the resonant filter with a controllable gain g. The optimal signal flow for the whole instrument is next configured. In the physical instrument, sound energy created by striking a bar excites the gourds, with the strongest excitation in the gourds immediately proximal to the bar. The sound heard by listeners is a mixture of the direct sound from the bar, the sound from the mouths of the gourds, and the sound from the membranes (an example of the signal flow is shown in (Figure 3.30).

In order to explore the relative importance of these sources, a number of signal path structures can be implemented.

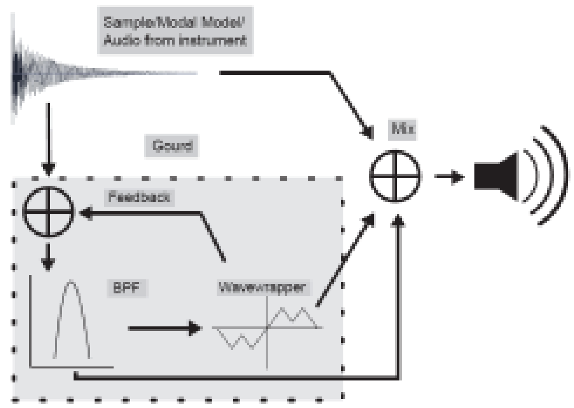


Figure 3.30: Signal flow for one-gourd/one-membrane model variant

The question of the importance of the excitation of more than one gourd by the striking of a bar can be addressed by comparing the performance of a model in which there is a one-to-one connection between the bars and the resonant filters to one in which the signal generated by the bar is fed into a number of parallel gourd models, with signal weightings given by an array of values B. In a system with p bars, and p gourds, the amplitude of the signal from bar i is bi (t), the level of the signal reaching gourd j is Bijbi(t), where 0 Bij 1, and we assume that the bar and gourd indices are such that Bii ¿ Bij , 8j 6= i , so that the gourd directly below the bar receives the strongest signal. The attenuation matrix as the product of a scalar blev and a matrix B that is normalized so that the largest element is equal to 1. May be written. By doing so, the global level of the signal that reaches the gourds with a single parameter can be adjust. When signal level from the bars to the gourds not immediately underneath them is negligible (i.e. , Bij = ij) the model structure can be simplified by reducing it to a bar-gourd signal flow. For any situation where the off-diagonal elements of B are non-zero, the signal must be fed from the bar into a number of parallel gourd algorithms, and sum the result. Regardless of the configuration of the bar-gourd system, the signal from the bar, and the signals from both the gourd and the membranes must be summed. To obtain these signals, two output taps in the gourd algorithm are taken- one immediately after the resonant filter, and one immediately after the non-linearity. By adjusting the relative levels of these taps the output of the gourd model can be fine-tuned to best emulate the behavior of the physical Gyil. In the following section a user survey investigating the algorithms is presented, as previously discussed. The performance of the three different forms of the non-linearity have been evaluated by a trained Gyil player, as is the coupling of the bars to the gourds.

Parameter	**Range**	**Meaning (note/Hz)**
f_c	20 to 10^4	the volume of the gourd
Q	0 to 1	size of mouth of the gourd
A	0 to 1	age of the membranes
c	0 to 1	asymmetry of the membranes
N	> 0	number of membranes
S_w	0 to 1	similarity of the membranes
g	0 to 1	membrane feedback gain
k	0 to 1	membrane direct gain

Table 3.5: Model parameters/Gyil physical properties

In addition, the performance of a physical model which uses a modal algorithm to simulate the bars is compared to the data produced by a hybrid synthesis method in which the acoustic signal of the bars captured by contact microphones is used as input to the gourd and membrane model. Although the model proposed in this section relies on a number of parameters that must be manually set, all of them are meaningful and correlated to actual characteristics of the Gyil. The correspondence between the model parameters and physical properties is shown in (Table 3.5).

3.2.3 Experimental Results

The proposed model was evaluated in two different ways. The first was through a comparative analysis, in which the output of the implemented models was compared to the audio signal recorded using a real Gyil. In the second, the model was applied as a digital effect and fed with the audio signal acquired from the wooden bars without attached gourds using contact sensors as well as the output of the modal synthesis model. By means of a survey I solicited feedback from expert musicians about the different configurations of the model and report on the findings.

Comparative analysis

The experiments in this section were performed by the author who is a percussionist with experience playing the Gyil. The parameters of the model were changed until the maximum auditory similarity, in the perception of the user, was obtained. The timbre of the model is greatly improved when a wave folder is used for the nonlinearity. The sound is further improved by adding clipping in series with the wave folder. This process corresponds to the expected behavior of the membrane. The final timbre lacks some of the brightness of the sound of the real instrument, but does sound like a Gyil. The magnitude spectra of the signals yielded by each model, considering the first 0.3s after the onset, were calculated are shown in Figure 3.30.

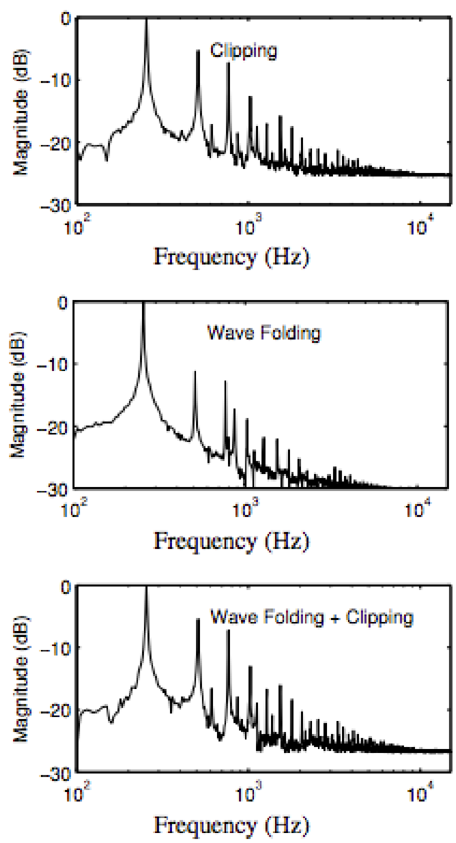


Figure 3.31: DFT non-linearities

Comparing these results to the spectrum of the real Gyil in (Figure 3.31), it can be seen that the digital models produce more energy below 1 kHz. Conversely, the physical Gyil has more high-frequency energy that is only present in the models that include clipping. The real sound has significant energy between 1.2kHz and 1.8kHz, which was not produced by any of the models. These harmonics may play a role in defining the timbre of the real instrument.

The energy of the harmonic content in the model follows the general rule 1/f, whereas that rule is clearly not applicable for the spectra of the sound of the real Gyil. The 1/f behavior is common to most non-linear mathematical functions, which means the behavior of the membrane can not be modeled as a simple non-linear function. Despite the limitations of the model, the obtained timbre is considerably similar to the one of the real instrument. Interestingly, although waveshaping synthesis is known to shorten the decay envelope of sounds [63], signals processed by my model had greater sustain than their acoustic counterparts. The attenuation of the decays of the Gyil sounds is likely due to coupling between the bars and gourds, and remains a matter for future investigation.

Qualitative Analysis

A qualitative assessment was also conducted to deal with aspects hard to identify with the spectral analysis method used. This assessment was based on short recordings of a physical Gyil as well as synthesized clips. A phrase was played on a xylophone retrofitted with contact microphones [113] for multi-channel recording. Using a simple thresholding process, symbolic data was extracted from this recording and used to render an audio file with the modal model that simulates vibrating wooden bars [6].

In total, four sound configurations were considered: a real Gyil recording, a multi-channel hybrid synthesis model in which each bar of the prepared xylophone was used to drive a different instance of the gourd model, a single channel hybrid synthesis model in which the signal of the prepared xylophone was mixed to mono and then driven into a single instance of the model, and a single-channel fully synthesized version in which the audio signal obtained from modal synthesis was used to drive the model. The audio files obtained were analyzed by experienced Gyil players, who kindly provided feedback regarding the usability of the sounds in real performances and the similarity of the modeled distortion to the buzzy sound of the real instrument. My goal was not a quantitative user study where findings are aggregated but rather to solicit feedback about my choices from musicians and scholars with significant experience with the Gyil. The names and affiliations of the experts are provided in (Table 3.6).

Name	**Info**
Greg Harris	Professonal vibist, world music specialist, music producer, Gyil resea
Birdooh SK Kakraba	Professional Gyil performer, composer, instructor- Hewale Sounds En
Dr. Payton MacDonald	Chair of percussion studies at William Patterson University, master r
Valerie Naranjo	Professional percussionist, clinician, Gyil expert, West African mallet
Dr. Karolyn Stonefelt	Chair of percussion studies, State Univeristy of New York Fredonia, C
Aaron Bebe Sukura	Professional Gyil performer, composer, instructor at the ICAMD, Un
Dorawana Edmund Tijan	Master Gyil performer, composer, teacher and builder from Saru, Gh
Dr. Michael B. Vercell	Director of the World Music Performance Center, West Virginia Univ
Bernard Woma	Master Gyil Performer, teacher, and founder of the Dagara Music Ce

Table 3.6: Gyil expert feedback

While the evaluators could identify the model output as having some characteristics of the Gyil sound, many of them commented that the synthesized samples do not fully capture the richness and life of the sound of a Gyil. Despite this shortcoming, the samples obtained by hybrid synthesis were generally considered to be usable for performance. Of all of the model variants, the one that was preferred by most of the specialists was the gourd model with one channel, which is surprising given that it should be equivalent as having a physical instrument with one single gourd. The synthesis of the sound using the modal model was considered the least realistic and, in general, not sounding like a Gyil.

The detailed comments from the evaluators suggested that the buzz sound of synthesized samples was too dark, mellow and predictable, whereas the buzz characteristic from a real Gyil is more inconsistent, in the sense that it does not present the same response at every stroke of the mallet. It was observed that the samples recorded from a real Gyil lacked some desirable sonic characteristics, like a deeper bass resonance and a correct timbre of the buzz. This is partly due to the particular physical instrument that the digital model was based on that has aged membranes.

Chapter 4

Sensing

4.1 Machine Awareness

In this section, different ways for pitched percussion instruments to sense the gestures of a performer and the associated musical environment are presented. They have been informed by the performance practice of the author and used in a variety of live music performances. More specifically two abstractions are presented: "Ghost Hands", a system for detecting, capturing and replaying repeated performer gestures, and "Mstr Drmmr++", a system for recognizing patterns of percussive events from a user defined set of patterns as a way of triggering computer-controlled musical events. Even though these abstractions are described in the context of electro-acoustic pitched percussion instruments, they can also be applied in other musical performance contexts.

4.1.1 Gesture Prediction

"Ghost Hands"

A machine learning tool intended to recognize repetition within the gesture control language and then automate/loop patterns it recognizes as being constant within a detection threshold [131] was developed in collaboration with software engineer Dr. Tiago Tavares. My role was as design engineer, where I contributed towards high-level theoretical design concerns, sensor implementation and mapping strategies. This appropriates and abstracts the audio looping paradigm, essentially serving as a real-time event parameter automator. While it can be used as an audio looper or event sequencer, it has primarily been explored as an effects parameter looper. The way an audio engineer/producer might automate effects processing parameters in post-production, Ghost Hands is designed for the stage. When a particular gesture is repeated enough times based on the device threshold settings, the gesture is learned and then repeated. The control data is then mapped to the parameters of choice, in this case the various delay settings on the Valhalla device. When the filter parameter loop has been established, the performer can continually re-contextualize how the filter modulates the incoming audio by constantly shifting the relationship between musical phrases and the continually changing filter parameters. In this way, the filters can be explored in a more dynamic capacity rather than being set to sheerly static settings. Because the hands are preoccupied during a lamellophone performance, Ghost Hands, coupled with the gesture tracking, affords the performer freedom of expressivity in an unprecedented, non-invasive, intuitive manner without interrupting the traditional playing techniques of the instrument. Many parameters can be mapped and automated simultaneously. There is no limit to how many parameters are engaged at a given instance. This opens up an entirely new world interaction that would have previously required several other individuals to be at a mixing desk turning knobs during performance in order to approximate the same results.

Forecasting of Position Data

A method capable of forecasting the continuation of a given data array $x[n]$, where $x[n]$ is the value of the n-th reading of a certain sensor is described. The forecasting algorithm does not require any previous training on previous templates, which means that the musician has freedom to improvise and generate creative soundscapes and define the appropriate gestures while performing. The key idea is that the gesture is identified by being repeated without requiring a preset looping duration. When forecasting, the system provides data, which aims to be equivalent to the sensor data that would be yielded if the musician continued the previous gesture. This allows the performer to continue playing while still getting the soundscape variations derived from sensor data. The forecasting method is based on evaluating what time lag, in samples, is most likely to be the fundamental period of the received data. This means that although the motion may be freely composed and improvised only repetitive motions can be predicted by the system. The method begins by estimating the autocorrelation of the last N input samples, which is defined as (Equation 4.1):

$$r_x[k] = \sum_{n=0}^{\frac{N}{2}-1} x[n]x[n-k] \tag{4.1}$$

An autocorrelation signal presents peaks at positions k that correspond to the time lags to which *x[n]* is mostly self-similar. However, there are some other peaks, especially at the zero time lag $k = 0$ that must be filtered out. In order to do that, a signal $\acute{r}_x[k]$ is calculated by upsampling $r_x[k]$ by a factor of two and subtracting it from the original signal. The value $j = arg$ max $\acute{r}_x[k]$ is obtained by a simple linear search. The forecasting, then, proceeds by yielding an estimate $\hat{x}[N+1] = x[N-j+1]$. The quality of the estimate may be evaluated by the ratio $r_x[j]/r_x[0]$, which will present values closer to 1 when the signal is significantly periodic. The forecasting system works in three different modes: learning, predicting and bypass. The learning mode should be triggered when a new gesture is to be acquired. While in this mode, the internal buffer is updated and the forecasting algorithm is executed at each new sample. Since learning mode assumes that the gesture is not yet learned, the system yields bypassed data instead of predicted data.

When the prediction mode is triggered, the system starts yielding forecasts, adding them to the internal buffer as if they were received as inputs. While in this mode, the system does not recalculate the forecasting algorithm, sticking with a single value for j during the whole process. The operation of bypass mode is to simply bypass input data to the output, while ignoring any operations regarding processing. Hence, if the prediction mode is triggered while the system is in bypass mode, the system will recall the last learned pattern. When operating in the bypass mode, the system preserves the last learned gesture; hence it may be used again by triggering the prediction mode.

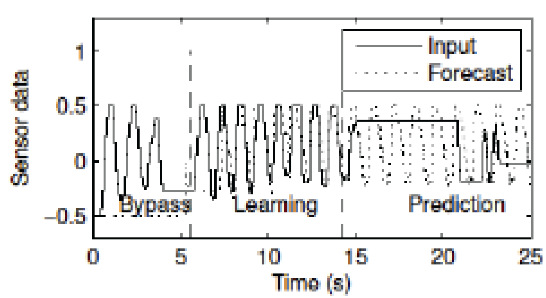


Figure 4.1: Forecasting system operation modes

The internal operation of the system is visualized in (Figure 4.1), which shows the input (bypassed) data from a one-dimensional sensor and the predicted data. The figure was generated while acquiring manually-driven data from a one-dimensional sensor. In the first five seconds, the system is in bypass mode, and the prediction system hasn't received any samples yet. When the learning mode is triggered, the predicted data quickly synchronizes with the repetitive sensor data received as input. When the prediction mode is triggered, the forecasting system continues the learned gesture and ignores the sensor data, creating a mode for potential man-machine improvisation.

4.1.2 Drum Pattern Identification

Mstr Drmmr++

Another approach to gesture control is to recognize percussive patterns from a user defined set of patterns in order to initiate computer-controlled musical events. A software device was developed in collaboration with undergradute computer science research assistant Mike Dean. My role was as design engineer, where I contributed towards high-level theoretical design concerns, sensor implementation and mapping strategies. The software is named Mstr Drmmr based after the tradition of the master drummer in West African drumming- this is another application of OSP. The machine learning software has been designed to listen and react much in the same way an ensemble will follow a Master Drummer in a traditional context.

> Chernoff states: "A master drummer's varied improvisations will isolate or draw attention to parts of the ensemble more than they seek to emphasize their own rhythmic lines, and a musician must always play with a mind to communicative effectiveness."[**?**] Master drummers will know all the parts of the arrangement, the respective dances, and give the calls and breaks in music direction. The software is meant to emulate this role, offering the performer the ability to train the algorithm with a musical phrase "break" that can be mapped to various control parameters. When the software "hears" the phrase, it reacts accordingly.

The proposed system aims at providing the performer the possibility of influencing the computer ensemble using musical cues with-out having to alter the playing technique, which represents a significant improvement over conventional external tactile controls. My approach is based on calculating the similarity between an input drum pattern, played at a certain point during the execution of a piece, and a previously recorded pattern. The similarity is a continuous value that is greater when the pattern played is closer to the recorded pattern. If the similarity value exceeds a certain user-defined threshold, the system triggers an associated action.

Two equal length sequences of numerical data can be compared in terms of a single value between one and zero. A value of one indicates the two sequences are identical, whereas zero indicates a drastic lack of similarity between the two data sets. This similarity computation is known as correlation. (Equation 4.2) computes a value that describes similarity between two sequences.

$$Similarity = \frac{\Sigma_{n=0}^{N-1} x[n]y[n]}{\Sigma_{n=0}^{N-1} x[n]^2 \times \Sigma_{n=0}^{N-1} y[n]^2} \tag{4.2}$$

To account for slight variation, sequences can be convolved with a Gaussian function with standard deviation and mean, as shown in (Equation 4.3), before being compared. This process results in a smearing of the sequence's values to adjacent locations, as seen in (Figure **??**).

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{(x-\mu)^2}{2\sigma^2}} \tag{4.3}$$

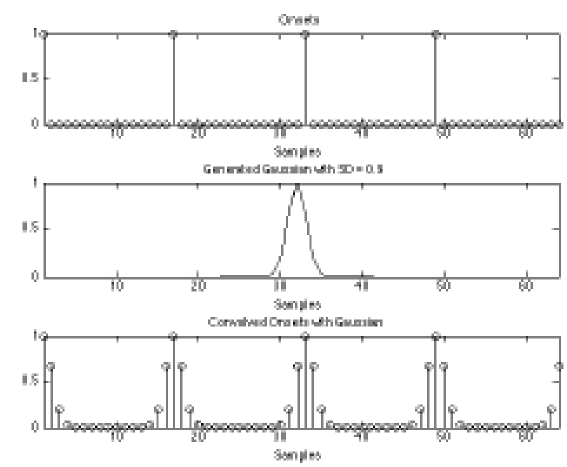


Figure 4.2: Convolution with Gaussian function.

To assemble such sequences in real time, the proposed system assumes that the player will perform to a specific tempo. For each time interval in the measure, if an onset is detected, it is notated in the sequence using a value of one. Otherwise a value of zero is used indicating a rest. The onset algorithm used calculates the power of the input signal and triggers onsets when a threshold has been surpassed [77].

From the methods described, similarity between musical patterns can be calculated. To do so, the system must have two modes of operation: recording a template pattern and comparing incoming patterns to the template pattern. When comparing incoming patterns the similar value of the input pattern to a set of stored templates is calculated and if it exceeds a user defined threshold an event is triggered.

In order to gain perspective on the effectiveness of the system in a real performance situation, the pattern recognition system was tested with a human percussionist performing on a traditional West African pitched percussion instrument known as the Gyil. The instrument provided audio input to the software through the use of contact microphones that were mounted to each bar of the instrument. The pattern recognition system was used to recognize patterns played on a single bar. Of the two types of tests done, the first provided similarity values when a pattern was played during a performance. In this first test, the sampled bar was only played when the user wished to trigger the software using a pattern. Otherwise, the bar was avoided as the performer played the instrument. In the second test, the entire instrument was utilized by the performer; the sampled bar was used for both triggering patterns and for playing.

The first test used five unique, one measure long rhythms that were tested ten times each at two different tempos. The two tempos used to test varied by one-hundred beats per minute. For all one-hundred measurements in the first test, an average similarity of 0.731 was calculated. The measured similarity values were affected most drastically by two factors: the human inaccuracies when recording the template pattern and the human inaccuracies when repeating the pattern in order to trigger some action. In the second test the system incorrectly triggered an action during a sixteen bar solo. During each solo performance, the user played rhythms on the sampled instrument bar every second measure. The number of times an action was not triggered when the player intended to do so was also documented. Of the ten tests for these sixteen bar solos, the system incorrectly identified a player's pattern as being the template pattern only twice, or 0.025 of the measures in which the bar was played. Of these two incorrectly identified patterns, there was stark similarity to the template pattern, and it was unsurprising that the system gave a false positive result. When the player intended on triggering an action by performing the template pattern, these tests gave no negative results.

4.2 Surrogate Sensor Framework

The work described in this section has been motivated by three applications: computer analysis of Gyil performance, live improvised electro-acoustic music incorporating the Gyil, and hybrid sampling and physical modeling. In all three of these cases, detailed information about what is played on the Gyil needs to be digitally captured in real-time. I describe a direct sensing apparatus that can be used to achieve this. It is based on contact microphones and is informed by the specific characteristics of the Gyil. An alternative approach based on indirect acquisition is to apply polyphonic transcription on the signal acquired by a microphone without requiring the instrument to be modified. The direct sensing apparatus developed can be used to acquire ground truth for evaluating different approaches to polyphonic transcription and help create a surrogate sensor. Some initial results comparing different strategies to polyphonic transcription are presented.

The custom Gyil surrogate sensing research describes related work and provides context for the custom sensing apparatus coupled with a physical model of the gourd resonators to create an acoustically excited hybrid model retaining the wooden bars while eschewing the gourds and reproducing their sonic contribution via MIR/DSP. Experiments in using sound source separation algorithms tailored specifically to the Gyil performing real-time causal transcription potentially bypass the need for direct sensing. The direct sensing apparatus is used to train this indirect surrogate" sensor by automatically providing ground truth to evaluate how good the transcription is. An underlying motivator is to achieve better results in both sensor design and audio analysis by tailoring algorithms specifically for the Gyil.

The Gyil (pronounced Jee-Lee) is a central instrument native to the Lobi culture in the Black Volta River Basin on the boarders of Burkina Faso, Cote d'Ivoire, and Ghana [96]. It is a wooden xylophone whose actual number of bars vary between 11 and 22, but is typically 14. The bars are arranged in a single row, made of Legaa wood (the region's indigenous hardwood), and use gourds (relative in size to the bar they are under) as resonators. The instrument resembles the marimba sonically and is graduated in size left to right- bigger/lower pitch to smaller/higher pitch, respectively. Its tuning is technically pentatonic, and spans three to four octaves (specifically accommodating vocal range).

Because West Africans don't traditionally have a collectively established fixed frequency to designate as a systematic tuning convention, e.g. A=440Hz, instruments are tuned at the discretion of the instrument maker and each has its own unique tuning, even if meant to be a pair. In general, African equidistant tuning is based on the recognition of steps that resemble one another. When a key does not conform yet is tolerated, it is considered dissonant. This idea of dissonance in the Lobi context will actually change the scale system from tetratonic to pentatonic, and is perceived as something that should be avoided. This is an important consideration when it comes to physical modeling, because the convenience of uniform distribution is not an afforded luxury, therefore each instrument and each bar on that instrument must be considered individually, along with its corresponding gourd and the sympathetic resonances from each unique neighboring gourd and the subsequent individual character of each bar.

The direct sensing prototyped relies on a combination of standard digital pro-audio equipment: Firewire AD/DA multi-channel audio interface, personal computer with appropriate audio processing capabilities, necessary software, playback speakers and low cost (less than 1 cent when bought in bulk) piezo transducers being used as contact microphones (hot glued directly to the bars) to acquire direct digital audio (Figure 4.3).

Figure 4.3: Surrogate sensor system

The acquired signals have a very low signal-to-noise-ratio and capture well the acoustic sound. A similar approach does not work very well when used with metal vibraphone bars, as metal bars are much more sensitive to the damping effect of attaching a piezo, which attenuates the higher frequency partials, and shortens the decay time. It does work with other wood pitched percussion instruments, such as the chromatic, western concert xylophone and Marimba. However, digital audio input using the method described is impractical, and cost-prohibitive (unless custom multiplexing is used) requiring more than 37 channels for any instrument over the standard 3-octave range. At the time of this publication, this is still a cost prohibitive design goal.

The system presented here is designed so that each bar has a piezo-electrical sensor glued onto it on the player's side near the bridge, but not uniform as the underside of the Gyil bars are irregular. The most resonant location for the piezos were sought, offering the best replication of the bars sound. Each sensor is connected to a 1/4" jack input into a Metric Halo[1] 2882 audio interface. The 2882 interface has eight 1/4" inputs with digitally controlled gain. This system utilizes two *2882's* linked optically allowing for 16 direct, identical channels of audio input, ideal for our Gyil which has 15 bars. The cabling was fabricated using low cost materials and the analog circuit is completely passive. Ableton Live and Max/MSP were used to process the incoming 15 channels of audio. Each bar is assigned its own channel in Live and can be processed individually as needed using native devices, third party plug-ins, or custom Max4Live devices. Monitoring was done on a stereo pair of *Genelec*[2] *8030A* speakers via a *MIDAS*[3] *Verona 320* mixing console, but any monitoring configuration can be implemented.

4.2.1 Hybrid Acoustic/Physical Model

The Gyil is unique within the xylophone family primarily in the specific use of gourds that are hung on the body of the instrument, below the wooden bars as discussed in Chapter 3. They act as resonators amplifying the sound of the wooden bars. The uniquely characteristic buzzing sound of the Gyil is the result of the preparation of the gourds, which have holes drilled in them and papered over with spider silk egg casings. The egg casings act as vibrating membranes, but are irregular in their form, due to the shape of the holes, and their material properties.

A physical model for the Gyil gourd has been developed and is described further in this section. It has been motivated by the desire to create sound synthesis techniques based on physical modeling for the Gyil as well as a way to simplify the transportation of the instrument as the gourds are large in volume, awkward in shape, and fragile. The direct sensing apparatus described in the previous section can be easily packed as

[1]http://mhsecure.com
[2]https://www.genelec.com/
[3]https://www.musictri.be/brand/midas/home

it consists of wires, contact, microphones and an external multi-channel sound card. The wooden bars can be easily folded and are quite robust. In addition it lowers the cost of making an instrument. The developed physical model is designed to act as an electronic replacement for the gourd while preserving the unique timbre of the instrument.

By having separate audio input through the contact microphones for each bar, as described in the previous section, each bar can be fed into separate gourd models resulting in a more realistic model of the actual instrument. The proposed system consists of a model for the gourd itself, and a model for the membrane. The assumption that the feedback from the gourd to the wooden bar is negligible and therefore can be viewed as a source-filter system is made. In order to make measurements of the acoustic properties of a single gourd, the bars were removed from the Gyil frame. Investigations were carried by exciting the gourd with a sine and triangle waves of fixed frequency and varying amplitude which led to the wave-wrapping model of the membrane described below. It was found that there is a clear threshold over which there is significant high frequency distortion which we speculate is caused when the egg casing membranes start deforming instead of simply vibrating. The gourds are considered as simple resonant bodies, which can be modeled by resonant band pass filters [99]. The frequency of the filter is inversely proportional to the volume of the gourd, and the resonance or Q-factor is inversely proportional to the size of the opening at the top. For that reason, larger gourds are chosen to be associated with the lower pitched notes. The range of gourd sizes may either be selected by hand, or a scaling constant, 0 ¡ s_f ¡ 1, may be used, so that the center frequency of the bandpass filter representing the smallest gourd, f_0, is selected and the frequency of the n^th gourd is given by $s_f^{N-1} f_0$

The membranes are modeled using wave folders, which have the property to wrap the input signal around two predefined limits. Mathematically, for an input signal with amplitude xin, and specified reflection amplitude A, the the output signal y is given by (Equation 4.4):

$$x_out = \begin{pmatrix} A & -cA < x_in < A \\ x_in - A & x > A \\ cA - x_in & x_in < -cA \end{pmatrix} \tag{4.4}$$

where $0 \leq c \leq 1$ is a parameter which we may set to adjust the asymmetry of the folding, that is, the difference between the absolute values of the high-level and the low-level limits, as can be seen in (Figure 4.4). The choice of wave folding over a clipping function reflects the need for a greater degree of harmonic content to be generated than clipping produces, when processing a dynamic signal.

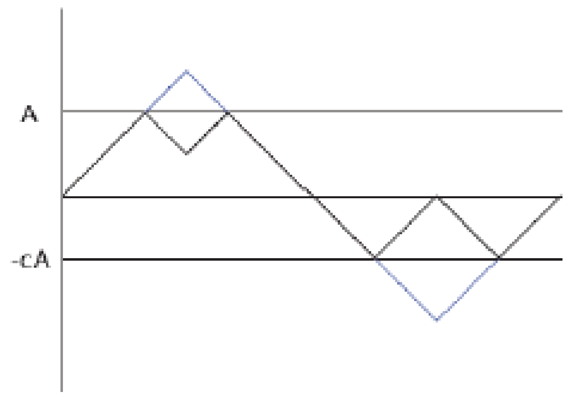


Figure 4.4: Simple wave folder w/ adjustable symmetry

The resonant filter yields signal to a number of membranes, which are wave folders as described by (Equation 4.4). To mimic the different sizes of the membranes, and the slight variations in material, the wave folders have different parameter settings for A and c (as defined in Equation 4.4). To simplify the implementation of the model, and to make it more transparent, an adjustable scaling parameter $0 < s_w < 1$ is specified so that, for a model with N membranes, the level at which signal is folded in the nth membrane is given by s $s_n^w A$, thus reducing the number of adjustable parameters for the wave folders from N to 2. Last, the output of the wave folders is fed back into the resonant filter, with a controllable gain. The whole system may be visualized in (Figure 4.5).

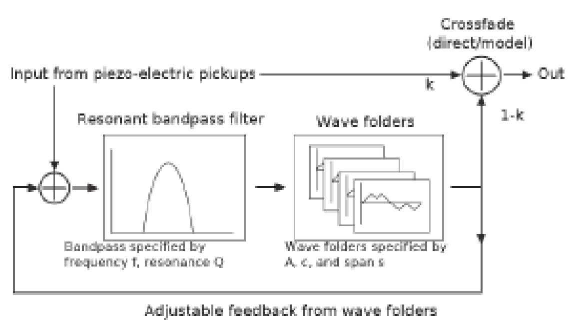


Figure 4.5: Diagram of signal flow for one gourd

The signal yielded by the physical model is the output of the resonant filter, which will be very close to the sound of the vibrating bar for low amplitudes and will have a considerable amount of distortion for high amplitudes. The gourd has the effect of amplifying the two harmonics present in the bar, and introducing a significant amount of spectral content between 1kHz and 7kHz, as can be seen by comparing the top and middle plots in (Figure 4.5). This gourd model, driven by the output of the piezo-electric direct sensor, succeeds in introducing high frequency content in a similar manner to the Gyil gourd, as seen in the lowest spectral plot in (Figure 4.6). It can be seen that the Gyil gourd produces a more broadband spectrum than the model, which is characterized by well-defined spectral peaks. The broad band spectrum introduced by the gourd suggests a greater degree of non-linearity in the physical system than in the first-order model.

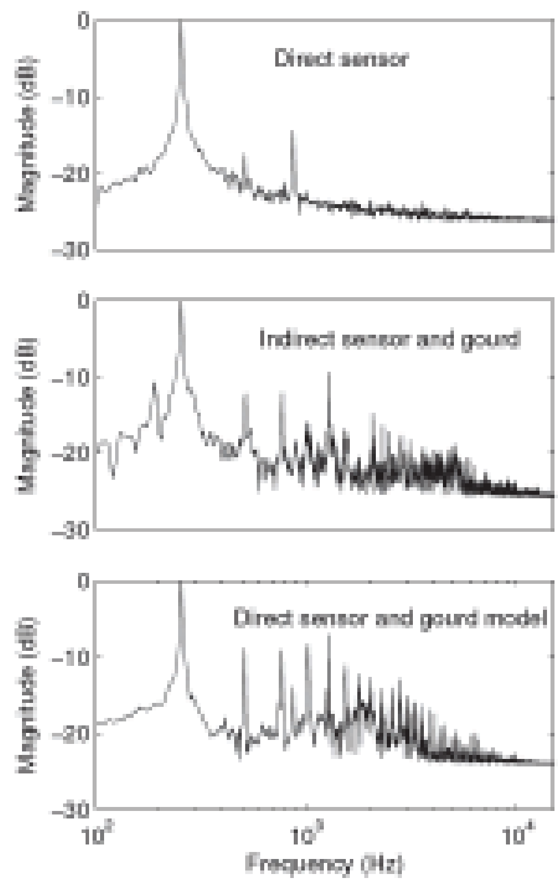


Figure 4.6: Various spectral plots for Gyil

The proposed model benefits from the direct sensors. In the physical instrument, larger gourds are related to the sound of lower pitched notes with higher intensity, and the same holds for smaller gourds and higher pitched notes. If the sound signal from the Gyil is simply obtained through a single microphone, this pitch-dependent coupling is not possible. By using a different channel for sensing each bar, it is possible to route signals through different instances of the model described above, hence obtaining a sound whose timbre is closer to that of the natural Gyil sound.

4.2.2 Sound Source Separation and Automatic-Transcription

The goal of automatic source separation is to separate the individual sound sources that comprise a mixture of sounds from a single channel recording of that mixture. Such systems can be used as front ends to automatic transcription which has the related but different goal of detecting when and for how long different sources are activated without requiring actual separation of the signals [141]. A large family of source separation/music transcription algorithms are based on various forms of factorization in which the time/frequency representation of the mixture is decomposed into a weighted linear combination from a dictionary consisting of basis functions that correspond to the time/frequency representations of the individual sound sources [120]. The majority of existing approaches are not causal (require all of the input signal) and are designed for more general types of sound sources. Therefore they are not appropriate for real-time processing. This work investigates the potential of such techniques in a real-time context and with a dictionary specifically tailored to the Gyil. One possible criterion for evaluating how good a certain approximation is using the least mean squares [53]. Since sound source activities are always positive, the non-negativity criterion may be incorporated in the calculation of the approximation, leading to the Non-Negative Least Squares (NNLSQ) algorithm [53]. The NNLSQ algorithm has been used in the context of transcription (that is, the detection of symbol note data) by Niedermayer [94], and, as it represents a form of dictionary-based factorization, it may also be used to perform some forms of sound source separation while forming the basis of this transcription system.

The application of digital signal processing techniques are used for the purpose of obtaining the same data yielded by the multi-channel direct sensors, but using as input only a single channel. This single channel can either be the summation of the sensor signals or the sound data acquired from a regular microphone that is not directly attached to the instrument. The motivation is to attempt to extract the same information without requiring any modifications to the actual instrument. For this purpose it was decided to investigate sound source separation algorithms. In order to obtain satisfactory performance the approach was tailored to the Gyil. In order to evaluate transcription algorithms it is necessary to have some form of ground truth of what the right answer should be. In most of the existing literature this is obtained through symbolic representations of the music that is then synthesized and passed through the source separation system. In the music traditions being evaluated here there is no concept of a score and therefore evaluation of a transcription system would require time-consuming manual creation of ground truth. By utilizing the direct sensing apparatus described above one can effectively collect unlimited amounts of ground truth and training data simply by playing the instrument. The techniques described below rely on a factorization algorithm called Non-Negative Least Squares (NNLSQ), as proposed in [54], which aims to obtain the best description (considering the least square error) of a certain signal using a non-negative combination of a set of pre-defined basis functions. The signal representation used for the purpose of this detection is the logarithm of the magnitude spectrum, which was obtained by dividing the input signal in frames of 12ms (with a 6ms overlap), multiplying each frame by a Hanning window, zero-padding it to twice its length and, for each frame x, calculating the spectrogram as $y = log_1 0(1 + ||DFT(x)||)$. The resulting spectra are trimmed in order to eliminate frequency components outside the spectra of the instrument's notes (and reduce the computational load). The basis vectors are obtained by averaging a few frames of the spectrogram of an execution of each isolated note. The NNLSQ algorithm is, then, executed over each frame of the analyzed piece's spectrogram, yielding the activation level of each note. This approach is similar to the one proposed by [94]. This information may be used either for sound source separation of for automatic transcription, as described below.

The dataset used in this experiment consists of an audio recording approximately 2 minutes long. It was simultaneously recorded using the direct sensors (as separate tracks) and a microphone placed in front of the instrument. After the recording, the data from the direct sensors was arterially mixed, simulating the process of analog mixing that is part of the first scenario described above. Hence, there were three different synchronized tracks: a multi-channel track where each bar of the instrument is explicitly recorded in one different channel, a single-channel track containing data obtained from the direct sensors and a single-channel track obtained from a microphone recording. All experiments in this section take in account using direct (contact) and indirect (microphone) sensors both to acquire the basis functions and to test the performance of the analyzed methods.

Sound Source Separation

In this section, an NNLSQ-based technique for sound source separation is evaluated. This technique relies on the assumption that polyphonic audio signals that result from the mixing of several different sources are, in terms of physical measures, the sum of the signals corresponding to each individual source. Also, the human perception derived from listening to that sound is essentially the superposition of the sensations triggered when listening to each individual source. Therefore, a reasonable mathematical model for the phenomenon of sound source identification is:

$$X = BA \tag{4.5}$$

In that model, B is a set of basis functions, that is, a set of vector representations of the sources to be identified, X is the representation of several measures of the phenomenon in the same domain as B, and A is a set of weight coefficients that represent how much each source defined in B is active in each measurement.

Evaluation of the technique is based on a resynthesis schema, as follows. Since the spectral representation related to each base vector is known, it is possible to re-synthesize the expected audio for each channel. That is done by summing the magnitude spectral representation of each basis, weighted by the activation level of that note, using the phase information from the input signal and then calculating an inverse DFT. In the experiments, the input signal was separated and then remixed. The final result was compared to the input using the evaluation method described by [24]. The Signal-to-Distortion ratio (SDR) is reported, as it is the only meaningful evaluation metric for one input, one output systems. The results can be viewed in (Table 4.1).

	Basis Source	
Input	**Microphone**	**Contact**
Microphone	0.0408	0.0049
Contact	0.0857	0.1558

Table 4.1: Sound-source separation: Signal-to-Distortion ratio (SDR)

As it may be seen, the use of sound-source separation techniques is not as effective as the use of individual sensor data regardless of the basis vectors used. This means that the use of multiple channels of direct sensors is interesting for the purpose of obtaining audio data. The next section describes an algorithm for obtaining control data, namely, the onsets and pitches that are played.

Automatic Transcription

Although obtaining audio data from the mono signal, as described above, may be difficult with today's techniques [141], it might only be desired to simply obtain the control data that would be provided by the direct sensors. Experiments were conducted aiming to determine the limits under which control data may be reliably obtained from these single-channel mixes. The ground truth transcription data was obtained by automatically detecting onsets using straightforward energy thresholding in the multi-channel track (offsets were ignored, as the Gyil does not have gestures related to onsets). This method of obtaining ground truth data is called surrogate sensing [5], and by using it a large amount of annotated data can be acquired in real-time.

The method used to obtain symbol data in this paper relies on obtaining basis vectors and then running the NNLSQ algorithm over the testing data. In order to obtain discrete control data, the activation levels are, yielded to a rule based decision algorithm that works as follows. First, all activation levels below a certain threshold are set to zero. After that, all values whose activation level difference are below another threshold b are set to zero. When a non- zero value for the activation value is found, an adaptive threshold value is set to that level multiplied by an overshoot *factor c*. The adaptive threshold decays linearly at a known rate d, and all values activation levels below it are ignored. Finally, the system deals with polyphony by assuming that a certain activation level only denotes an onset if it is greater than a ratio g of the sum of all activation values for that frame. After this process a list of events, described by onset and pitch, is yielded. The events in the ground truth list and in the obtained list are matched using the automatic algorithm described in [25]. An event is considered correct if its pitch is equal to the pitch of the matched event and its onset is within a 100ms range of the ground truth. The values reported are the Recall (R, number of correct events divided by the total number of events in the ground truth), Precision (P, number of correct evens divided by the total number of yielded events) and the F-Measure $(F = 2RP/(R + P))$. These are standard metrics used for evaluating transcription and originating from information retrieval. (Tables

4.2,4.3) show these coefficients for both analyzed pieces.

Input Source	**Recall**	**Recall**	**Precision**
Microphone	36.39	33.63	34.95
Contact	81.01	44.91	57.79

Table 4.2: Direct sensor: detection accuracy (%)

Input Source	**Recall**	**Recall**	**Precision**
Microphone	59.81	47.97	53.24
Contact	20.89	74.16	32.59

Table 4.3: Indirect sensor: detection accuracy (%)

As can be seen, results degrade when using different sensors to calculate the basis vectors than the sensors used to acquire the signal in which the detection will be performed on. That is because the microphone signal is considerably noisier - due to normal ambient noise when playing, as well as sounds from the Gyil frame - and the basis vectors obtained from microphone recordings take these model imperfections into account. It is also important to note that the precision obtained when analyzing direct sensors was always greater, which is easily explainable by the absence of ambient noise in the recording. The results, however, indicate that there is significant room for improvements, and for greater accuracy requirements it is necessary to use the multi-channel direct sensor data, as current real-time techniques for polyphonic transcription have limited reliability. The direct sensing has been essential in enabling us to evaluate these different approaches. It is also important to note that the results degrade drastically when using basis functions that are not obtained from Gyil recordings.

Chapter 5

Actuation

5.1 Auto-Calibration

In this section we describe a system designed to aid the computer in hearing itself so that it can monitor its own output for auto-calibration. This was project was developed in collaboration with software engineer Dr. Steven Ness, who was a Phd candidate at the time. Here my role was as design engineer, where I contributed towards high-level theoretical design concerns, mapping strategies and system implementation. The typical architecture of interactive music robots is that the control software receives symbolic messages based on what the other performers (robotic or human) are playing as well as messages from some kind of score for the piece. It then sends control messages to the robot in order to trigger the actuators generating the acoustic sound. In some cases the audio output of the other performers is automatically analyzed to generate control messages. For example audio beat tracking can be used to adapt to the tempo played. Self-listening is a critical part of musicianship as anyone who has struggled to play music on a stage without a proper monitor setup has experienced.

However this ability is conspicuously absent in existing musical robots. One could remove the acoustic drum actuated by a solenoid so that no sound would be produced and the robotic percussionist will continue blissfully playing along. This work has been motivated by practical problems experienced in a variety of performances involving percussive robotic instruments. (Figure 5.1) shows an experimental setup in which solenoid actuators are used to excite different types of frame drums. The ability of a robot to listen especially to its own acoustic audio output is critical in

Figure 5.1: Solenoid actuated frame drum array

addressing these problems, as such adapted relevant music information retrieval techniques for this purpose are described in Chapter 4. More specifically, self-listening can be used to automatically map controls to actuators as well as used to provide self-adapting velocity response curves. Pitch extraction and dynamic time warping can is used for high-level gesture analysis in both sensor and acoustic domains.

MOTIVATION

MISTIC has extensive experience designing music robotic instruments, implementing control and mapping strategies, and using them in live and interactive performances with human musicians, frequently in an improvisatory context. In addition the author has regularly performed with robotic instruments. One of the most important precursors to any musical performance is the sound check/rehearsal that takes place before a concert in a particular venue. During this time the musicians setup their instruments, adjust the sound levels of each instrument and negotiate information specific to the performance such as positioning, sequencing and cues. A similar activity takes place in performance involving robotic acoustic instruments in which the robots are set up, their acoustic output is calibrated and adjusted to the particular venue and mappings between controls and gestures are established. This process is frequently tedious and typically requires extensive manual intervention. This work attempts to utilize MIR to simplify and automate this process. This is in contrast to previous work in robotic musicianship that mostly deals with the actual performance. More specifically, three problems are focused on: automatic mapping, velocity calibration, and melodic and kinetic gesture recognition.

The experimental setup used consists of a modular robotic design in which multiple solenoid-based actuators can be attached to a variety of different drums. Audio signal processing and machine-learning techniques are employed to create robotic musical instruments that listen to themselves using a single centrally located microphone.

It is a time consuming and challenging process to setup robotic instruments in different venues. One issue is that of mapping, that is, which signal sent from the computer maps to which robotic instrument. As the number of drums grows, it becomes more challenging to manage the cables and connections between the controlling computer and the robotic instruments. The system proposed here performs timbre classification of the incoming audio, automatically mapping solenoids correctly in real-time to the note messages sent to the musically desired drum. For example, rather than sending an arbitrary control message to actuator 40 the control message is addressed to the bass drum and will be routed to the correct actuator by simply listening to what each actuator is playing in a sound-check stage. That way actuators can be moved or replaced easily even during the performance without changes in the control software. The same approach is also used to detect broken or malfunctioning actuators that do not produce sound.

When working with mechanical instruments, there is a great deal of non-linearity and physical complexity that makes the situation fundamentally different from working with electronic sound, which is entirely virtual (or at least not physical) until it comes out of the speakers. The moving parts of the actuators have momentum, and changes of direction are not instantaneous. Gravity may also play a part, and there is friction to be overcome. Frequently actuators are on separate power supplies which can result in inconsistencies in the voltage. The compositional process, rehearsal and performance of The Space Between Us by by David A. Jaffe[1], in which Andrew Schloss was soloist on robotic percussion, involved hand-calibrating every note of the robotic chimes, xylophone and glockenspiel. This required 18+23+35 separate hand calibrations and took valuable rehearsal time. The Orchestrion[2], in which I was Robotics Technician, had 100+ actuators on stage that needed calibrated every concert, beginning at 9am and taking the entire day before sound-check 2hours before concert. Many times during the concert instruments would become uncalibrated, which was difficult to fix while in performance.

[1] https://2104310a1da50059d9c5-d1823d6f516b5299e7df5375e9cf45d2.ssl.cf2.rackcdn.com/2014/03/The-Space-Between-Us.pdf

[2] http://www.nonesuch.com/albums/the-orchestrion-project

I describe a method for velocity calibration, that is what voltage should be sent to a solenoid to generate a desired volume and timbre from an instrument. Due to the mechanical properties of solenoids and drums, a small movement in the relative position of these two can lead to a large change in sound output. The most dramatic of these is when during performance a drum moves out of place enough that a voltage that at the start of the performance allowed the drum to be hit now fails to make the drum sound. Depending on the musical context, this can be disastrous in a performance context. Good velocity scaling is essential for a percussion instrument to give a natural graduated response to subtle changes in gesture, e.g. a slight increase in the strength (velocity) of a stroke should not result in a sudden increase in the loudness of sound.

Issues like velocity calibration or control mapping seem quite pedestrian, or even trivial until one has grappled with this problem with real instruments. The ability of a robotic instrument to perceive at some level its own functioning is important in making robust, adaptive systems that do not require regular human intervention to function properly. This ability of the actuator to perceive its own status is referred to here as proprioception

I also describe some experiments recognizing melodic and kinetic gestures at different tempi and with variations in how they are performed. This can be viewed as an exchange of cues established before the performance especially in an improvisatory context. This allows higher level gestures to be used as cues without requiring exact reproduction from the human performer interacting with the robotic instrument and enables a more fluid and flexible structuring of performances.

5.1.1 Drum Classification for Auto-Mapping

An experiment to investigate the performance of an audio feature extraction and machine learning system to classify drum sounds to perform automatic mapping was performed. The audio features used were the well known Mel-Frequency Cepstral Coefficients (MFCC) calculated with a window size of 22.3ms. These were then used as input to a Support Vector Machine (SVM) machine learning system. A dataset of audio with 4 different frame drums being struck by the robot with a time of 128ms between strikes Was collected, then all the MFCC of this audio calculated, and then the 8 highest MFCC0 (roughly corresponding to perceptual loudness) was found and these were marked as onsets in the audio. The MFCC feature vectors corresponding to these onsets were used to train the classifier.

Peak offset	**Percent correct**	**Peak offset**	**Percent correct**
0	1	2	3
66.38	91.95	91.67	91.95
4	5	6	7
90.52	86.49	86.49	77.59

Table 5.1: SVM classifier accuracy

A separate test data set was also collected. Percussive sounds can be challenging to classify, as there is not a lot of steady state spectral information. The results of this experiment gave a classification accuracy of 66.38%, as shown in the first line (Peak offset 0) in (Table **??**). The same experiment was performed again but using different offsets from the highest peak in window sizes of 22.3ms. When all frames were classified with the frame immediately after the highest peak, a classification accuracy of 91.95% was obtained. This result is interpreted to mean that the resonance after the transient is clearly distinguishable for different drums, whereas the transient at the onset is fairly similar for different drums. This performance quickly degrades when moving away from the onset.

These results are for individual 22.3ms frames so it is easy to get 100% correct identification by voting across the entire recording which can then be used for the automatic mapping. Each solenoid on the robotic instrument is actuated in turn, the audio is classified and then the appropriate mappings are set so that the control software can address the actual frame drums rather than the actuators.

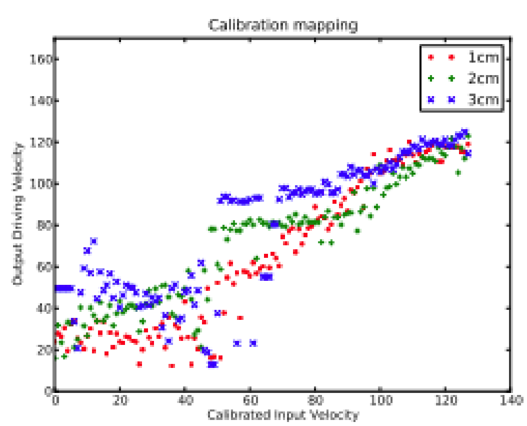


Figure 5.2: Calibrated input velocities mapped to output driving velocities

5.1.2 Timbre-Adaptive Velocity Calibration

The acoustic response of a drum both in terms of perceived loudness and timbral quality is non-linear with respect to linear increases in voltage as well as to the distance of the solenoid to the vibrating surface. In the past calibration was performed manually by listening to the output and adjusting the mapping of input velocities to voltage until smooth changes in loudness and timbre where heard. In this section I describe how to derive an automatic data-driven mapping that is specific to the particular drum.

The first objective is to achieve a linear increase in loudness with increasing MIDI velocity for a given fixed distance between beater and drumhead. However, in practice, the beater may be mounted on a stand and placed next to the drumhead mounted on a different stand. Thus the distance between beater and drumhead will vary depending on setup, and may even change during a performance. Thus a second objective is to achieve a similar loudness versus MIDI velocity (corresponding to voltage) curve over a range of distances between beater and drumhead.

To achieve these objectives audio was collected for all velocity values and three distance configuration (near 1cm, medium 2cm, far 3cm). The loudness and timbre variation possible is captured by computing MFCC for each strike. More specifically for each velocity value and a particular distance a vector of MFCC values is obtained. The frequency of beating was kept constant at 8 strikes per second for these measurements. The first MFCC coefficient (MFCC0) at the time of onset is used to approximate loudness. Plots of MFCC0 for the distance configurations are shown in (Figure 5.4.a).

In order to capture some of the timbral variation in addition to the loudness variation the MFCC vectors are projected to a single dimension (the first principal component) using Principal Component Analysis (PCA) [63]. As can be seen in (Figure 5.4.c) the PCA0 values follow closely the loudness curve. This is expected, as loudness is the primary characteristic that changes with increasing velocity. However, there is also some information about timbre as can be seen by the near plot that has higher variance in PCA0 than in MFCC0.

The goal is to obtain a mapping (from user input calibrated velocity to output driving velocity) such that linear changes in input (MIDI velocity) will yield approximately linear changes in the perceived loudness and timbre as expressed in PCA0. Data is utilized from all the three distance configurations for the PCA computation so that the timbrespace is shared. That way even though separate calibration mappings for each distance configuration are gotten, they have the property that the same calibrated input value will generate the same output in terms of loudness and timbre independently of distance.

In order to obtain this mapping the PCA0 values for each distance configuration are quantized into 128 bins that correspond to the calibrated input velocities. The generated mapping is the wrong way i.e. from output driving velocities to calibrated input velocities and is not an injection (one-to-one function) so it can not be directly inverted. To invert the mapping for each calibrated input velocity (or equivalently quantized PCA bin) the average of all the output driving velocities that map to it are taken as the output driving value. This calibration mapping is shown in Figure 5.2. (Figures 5.4.b and 5.4.d) show how changing the calibrated input velocity linearly results in a linearized progression through the timbrespace (PCA0) and loudness (MFCC0). These graphs directly show the results of this calibration but it is also possible to fit lines to them. In either case (direct calculated mapping or line fit) the calibrated output changes sound smoother than the original output.

5.1.3 Gesture recognition using Dynamic Time Warping

Collaborating musicians frequently utilize high-level cues to communicate with each other especially in improvisations. For example a jazz ensemble might agree to switch to a different section/rhythm when the saxophone player plays a particular melodic pattern during soloing. This type communication through high level cues is difficult to achieve when performing with robotic music instruments. In my performances I have utilized a variety of less flexible communication strategies including pre-programmed output (the simplest), direct mapping of sensors from a performer to robotic actions, and indirect mapping through automatic beat tracking. The final experiments described in this work show how high-level gesture recognition that is robust to changes in tempo and pitch contour can be correctly identified and used as a cue. This system is flexible and can accept input from a wide variety of input systems. Experimental results with the Radiodrum are shown as well as melodic patterns played on a vibraphone. There has been considerable work done in the area of using Dynamic Time Warping for gesture recognition, including work done by Akl and Valaee [8] and Liu et al. [61].

For the first experiment, the most recent iteration of the Radiodrum system was used, a new instrument designed by Bob Boie that dramatically outperforms the original Radiodrum in terms of both data rate and accuracy. A professional percussionist generated 8 different instances of 5 types of gestures, which were an open stroke roll, a sweep of the stick through the air, a pinching gesture similar to the pinch to zoom metaphor on touchscreens, a circle in the air and a buzz roll. Collected were (X, Y, Z) triplets of data from the sensor at a sample rate of 44100Hz and then downsampled this data to 120Hz to allow the comparing of gestures that were on average 1-2 seconds in length while remaining within the memory limits of our computer system. It was empirically determined that this rate captured most of the information relevant to gesture recognition. From this data, the similarity matrix of each gesture to each other gesture is computed. Dynamic Time Warping [115] is used to compute an alignment score for each pair of gestures that correspond to how similar they are. For each query gesture we return a ranked list based on the alignment score and calculate the average precision for each gesture. As can be seen in (Figure 5.3), gesture identification is quite reliable in both cases. The Mean Average Precisions (MAP) are 0.931 and 0.799.

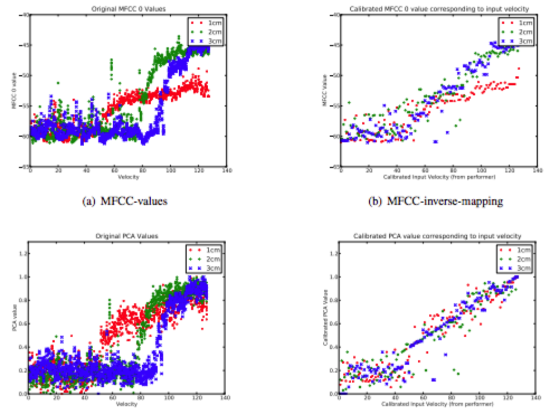
radiodrum			Vibraphone		
Gestures	AP	P@1	Gesture	AP	P@1
roll	0.866	1.0	pattern1	0.914	1.0
sweep	0.980	1.0	pattern2	0.812	0.9
pinch	0.837	1.0	pattern3	0.771	0.9
circle	1.000	1.0	pattern4	0.882	1.0
buzz	0.978	1.0	pattern5	0.616	0.9
MAP	0.931	1.0	MAP	0.799	0.94

Figure 5.3: Precision of Radiodrum and vibraphone gestures

5.2 Marimba platform

DSRmarimbA

Typically electronic music requires speakers which flatten all the layered voices that might be occurring in the performance into one focal point radiating outward in the direction the speaker is facing, usually situated in a stereo configuration. In an otherwise all acoustic setting these sounds would have a spatialized context radiating in a 360 degree pattern while modulating each others character as well. Musical robotics retain the acoustic, physical vibrations of the acoustics of the instrument



(a) MFCC-values (b) MFCC-inverse-mapping

(c) PCA-values (d) Calibrated PCA

Figure 5.4: Loudness and timbre based velocity calibration

occurring in a spatial context while allowing for (at times) unprecedented musical results that humans cannot execute physically as the instrumentalist.

The DSRmarimba (DSRm- name coming from the idea of adding variable Decay Sustain and Release amplitude envelopes to the bars Attack) has set out to develop, compile, and embed certain sensing and actuation techniques in order to minimize the often cumbersome footprint of actuated pitched-percussion instruments while preserving conventional playing techniques so a performer can play in tandem with the actuated capacity creating an all-in-one framework (Figure 5.5).

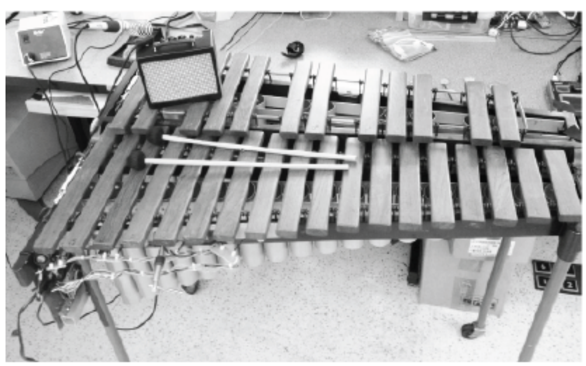


Figure 5.5: DSRmarimbA

The DSRm. is meant to be portable and compact, yet robust and flexible as a platform for testing various pitched percussion actuation and interfacing techniques. Design problems that DSRm. addresses for improved functionality and increased flexibility are as follows: low-cost, easy to implement, ready cased solenoids mounted underneath the bars so that the instrument could still be played conventionally and direct signal acquisition for filtering and amplification. Goals were that the DSRm be autonomous, eschewing the PC dependency. Pure Data (Pd) is used. Pd runs on the Beaglebone, being Linux compatible. This affords embedding the computing environment on the DSRm., while taking advantage of Pd s existing DSP libraries. This way, I can develop new custom interfaces for the DSRm that are compatible with the existing system.

5.2.1 Physical Design

1. Hardware Configuration
The DSRm s computer is the Beaglebone. PD is used to communicate between interfaces and the solenoids, as well as being the DSP environment. Both the DSRm and the Likembe have piezo pickups on them that plug directly into the Beaglebones audiocape, which provides stereo audio ADC/DAC via two 1/8th jacks. Their signals are routed and processed in PD then output directly to their own speakers if selected or into the xylophone bars, as described later. The Likembe has a custom Arduino Nano interface that plugs into the Beaglebones USB port. This sends the serial data from the sensors into Pure Data for mapping. The Beaglebone communicates to an Arduino UNO, which has a shield and breakout circuit that drives the solenoids via control messages from Pd (Figure 5.6), where a custom sequencer (Fig. 5.7) has been designed to accept messages from the bar hits via triggers detected by the piezos, or from the capacitive touch interface on the Likembe.

2. Mechanical Implementation
The current version of the DSRm is a rapid prototype. Solenoids were spaced appropriately and taped to a ruler (Figure 5.10) which is C clamped to the frame of the marimba. The solenoids patch into a protoboard that has the Arduino Uno and Beaglebone mounted to it (Fig. 5.8). This board setup is designed to be easily removable and reconfigurable. The board is velcroed to the frame of the marimba for easy removal.

Despite its primitive nature the prototype is quite robust and streamlined for the marimbas frame, while allowing for easy access to all the I/O of both the Beaglebone and Arduino for maintenance. The audio mixer is also built on a protoboard and velcroed to the frame making it easily removable, as well. The Piezo s are taped to the underside of the bars and patched into the mixer respectively.

3. Solenoid

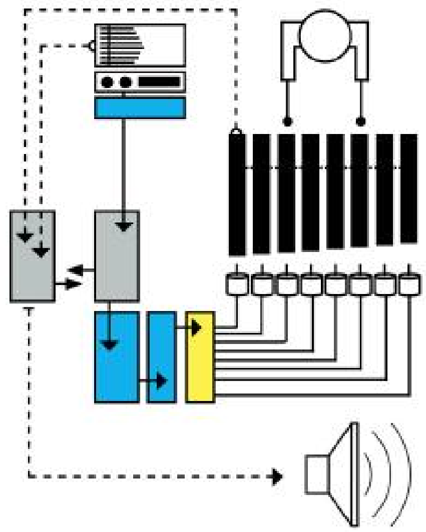


Figure 5.6: DSRm hardware configuration

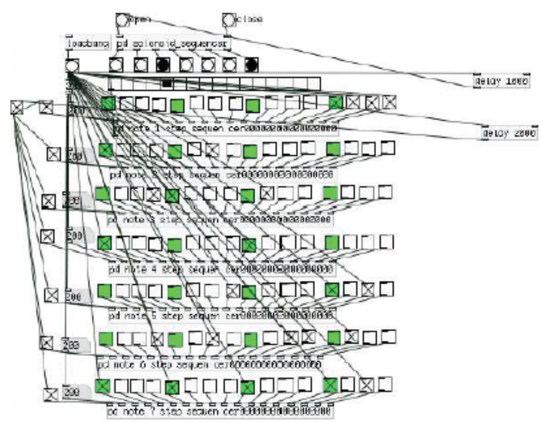


Figure 5.7: Solonoid sequencer

The main goal was to implement an actuation system that did not interfere much with the natural configuration and playing technique of the marimba. For this purpose a set of 5V solenoids were found. These solenoids have a 4.5mm throw, and where placed underneath the marimba bars that provided strong enough hits (65 gf @ 4mm throw) to the bars to produce audible signals and permit amplification with a good signal-to-noise ratio. The unit is the Sparkfun ROB-11015 and only cost $4.95US each (Figure **??**).

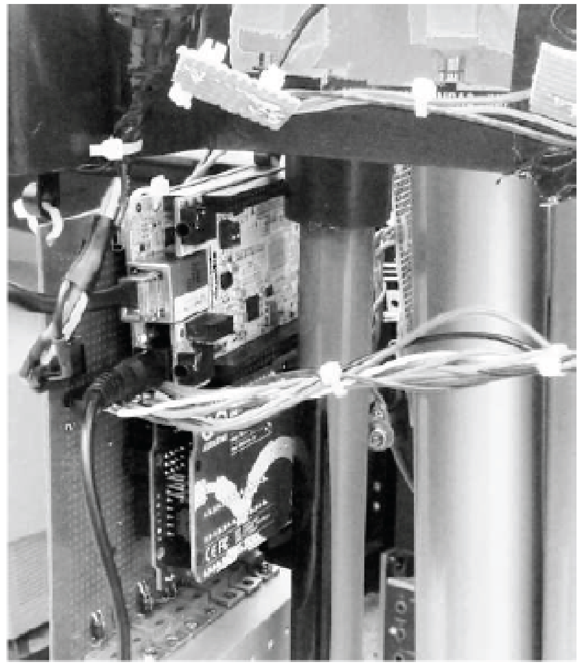


Figure 5.8: Beaglebone/Arduino/heatsink

Figure 5.9: Solenoid

5.2.2 Electrical Design

1. Solenoid Actuation

A circuit based on Darlington-transistors was designed to control the solenoids. An Arduino UNO connected to a sensor shield, drives the solenoids. The schematic shown in (Figure 5.11) represents the systems circuit. A TIP102 Darlington transistor was

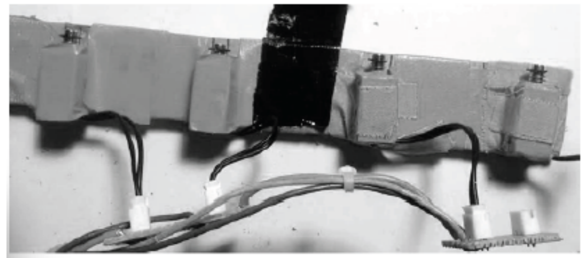


Figure 5.10: Ruler and solenoids

used, with the Base connected to a digital pin of the Arduino, the collector connected to one of the terminals of the solenoid, and the emitter connected to ground. The other terminal of the solenoid was connected to power. A 1N4004 Diode was connected in parallel to the solenoid, with its anode connected to the collector of the transistor, to dissipate any remaining energy when the solenoid is turned off. A 1K ohm resistor was connected between the Arduinos digital pin and the transistors base to limit the current, protecting the pin.

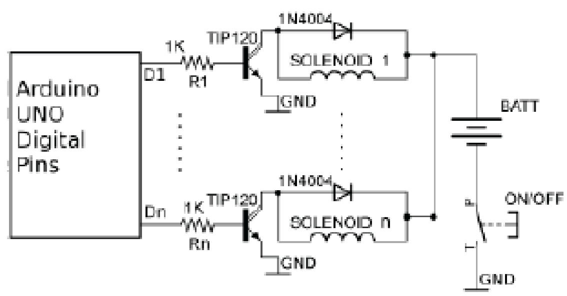


Figure 5.11: Schematic of solenoid actuation

Initially I tried to achieve a more streamlined and enclosed system, initial testing was done with the Arduinos 5V pin used to power the circuit for the solenoids. Unfortunately this resulted in some unreliable behavior, with some solenoids not being triggered at all. It was decided then to incorporate a dedicated battery pack to power the solenoids. A power switch was incorporated to disconnect the battery pack from the circuit when not in use.

The Arduinos firmware consists mainly of a routine that reads incoming data from the serial port that communicates when to trigger the corresponding solenoids. When the system is turned on, the microcontroller configures the digital pins connected to the solenoids to Output mode and triggers each solenoid once to make sure that the circuit is functioning correctly. The internal LED (digital pin 13) on the Arduino

```
int numSol = 7; // number of connected solenoids
int sol; //solonoid number
int led = 13;
// Initialization
void setup() {
  Serial.begin(57600);
  pinMode(led,OUTPUT);
  digitalWrite(led,LOW);
  // initialize the digital pins as an output.
  for(int s=2;s<2+numSol;s++){
    digitalWrite(led,HIGH);
    pinMode(s, OUTPUT); delay(200);
    digitalWrite(s,HIGH); delay(10);
    digitalWrite(s,LOW);  delay(1);
    digitalWrite(led,LOW); delay(200);
  } }
void loop() {
  while (Serial.available()>0){
    sol = Serial.read();
    // trigger the corresponding solenoid
    digitalWrite(sol,HIGH); delay(1);
    digitalWrite(sol,LOW); delay(1);
  } }
```

Figure 5.12: Arduino code

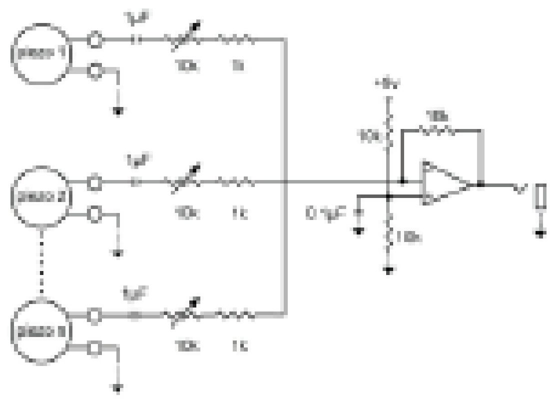


Figure 5.13: Piezo summing mixer

lights up when every pin is being set up to provide visual feedback (Figure 5.12).

2. Audio PIckup and Mixer

Piezos are used for direct audio acquisition and a summing mixer has been designed to accept all the signals of each bar, outputting a summed mono signal (Figure 5.13).

3. DSP

This project draws from previous Open Music Computing work expanding previous research [79], which continues to be extended as new musical applications arise. The system provides analog-to-digital conversion (ADC) by way of a TLV320AIC3106 codec provided by a Beaglebone expansion board, known as the Audio Cape. This expansion also provides digital-to-analog conversion (DAC) for output. Since this framework provides audio interfacing, the DSRms mixer is connected directly into the 1/8 audio input jack on the Beaglebone, which can receive both the Likembes and the DSRms signals independently using a standard dual-mono to stereo adaptor.

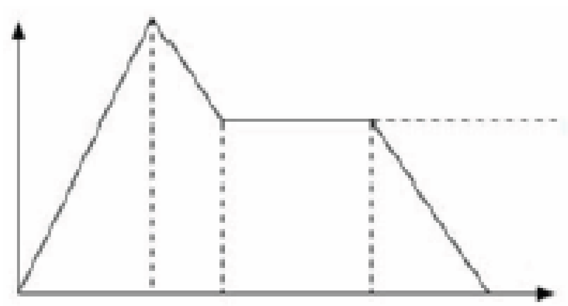


Figure 5.14: ADSR

The DSP system uses PD. PD facilitates the capture of audio, as well as the rapid prototyping of custom digital signal processing (DSP) techniques. Both the marimba and Likembe have piezo pickups that input directly into the audiocape. Within Pd their signals are routed independently into various filters for creative processing. For testing a multi-channel audio looper, a multi-tap delay with low-pass frequency filtering and a sinusoidal synthesizer instrument are implemented. The various parameters of each of the modules are controlled via the Likembe and marimba interfaces.

4. Bar Triggers

The piezo signal from each bar that has been summed for audio routing can be repurposed for control signal data providing the amplitude envelope (ADSR: 5.14) of each hit independently by being split and routed into the analog inputs of the of the Arduino or Beaglebone GPIO (Figure 5.15). This works the same way as conventional drum triggers. In Pd the data is scaled and used the same as any FSR would be for percussion triggering. This offers the possibility of looping musical phrases (MIDI) played by the human that can be continued to be played by the solenoids.

5.2.3 DSR - bars as speakers

DSR stands for decay, sustain, release. When you strike the marimba you naturally have an attack and and a relatively fast decay that cannot be expressively sustained.

With this system one can shape the envelope the same way one would with a synthesizer, only it would all be driven from an acoustic audio waveform triggered by the bar hit. A link to a video demo below shows where the bar hit triggers an oscillator to be played back through the bar. During the video, filters are applied to the signal from the digital oscillator in which case the parameters are modified via a MIDI interface. It can be imagined that the authors previous work with idiomatic and non-invasive gesture sensing for the marimba could be employed to have a fully dynamic system tailored to the DSRm [114]. The DSRm builds on the work of C. Britt [19].

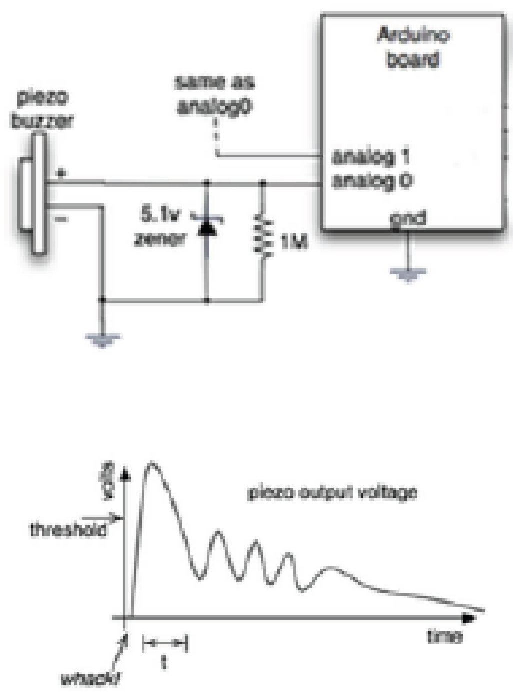


Figure 5.15: Piezo schematic and response

However, the DSRm has the potential for broader extended acoustic sound design capabilities, as well as the option to acquire a direct signal for filtering possibilities and/or amplification using the piezos as both sensors and actuators. The EmVibe only is able to produce long sustaining tones with minimal dynamic range. This is a result of driving the bars to reach their self-resonating frequency and offers little control of the dynamics and no possibility of direct signal acquisition of the bars acoustic sound. Also, in contrast to the EmVibe, the DSRm.s digitally extended sounds are not interfered by traditional striking of the bars by a human, which allows 3 dimensions of playability: 1. with the mallets; 2. with the solenoids; and 3. with the audio signals. Given that there are 30+ bars on the marimba, theoretically, one could perceive there to be that many speakers and/or solenoids allowing for a wide range of sonic possibilities. The EmVibes major drawback, besides the lack of dynamic control over the actuated sound, is that the extended sound is dampened and compromised when the bar is struck by the mallet or sustain pedal closing; this is not the case with the DSRm.

5.2.4 Idiomatic HCI

The majority of existing work in this area has focused on symbolic digital representations of music, typically MIDI. This system is embedded and as a result uses its own custom protocol. The typical architecture of interactive music robots is that the control software receives symbolic messages based on what the other performers (robotic or human) are playing as well as messages from some kind of score for the piece. It then sends control messages to the robot in order to trigger the actuators generating the acoustic sound. In this case the solenoids are interfaced two ways via software. One that is controlled from the tines of a Likembe (a type of lamellophone) via a custom designed interface and another that is controlled from triggers generated by the hits on the xylophone bars by the human performer using piezos.

A capacitive touch interface has been created to allow for communication directly with the DSRm via Pd on the Beaglebone via the tines. A press of the tine will result in the corresponding marimba bar to be struck by the solenoid without interrupting play on the Likembe- creating a symbiotic duo instrumental relationship. This instrument is called El-Lamellophone (El-la) [130] and is a system for lamellophone hyperinstruments. Ella, combined with the sensing paradigms developed specifically for mallet instruments in the authors previous work ([49], [113], [93]) offer a complete sensing and DSP paradigm for pitched percussion providing unprecedented and seamless integration with the computer. This level of custom multi-model interfacing techniques coupled with the various novel sound production methods create a complete performance system [73]. The DSRm eschews the laptop and need for speakers, while retaining the power of DSP filters and synthesis capabilities, which are designed to be interfaced by the traditional marimbist s conventional techniques. For a detailed look at the overview of how the interfacing integrates view (Figure 5.16).

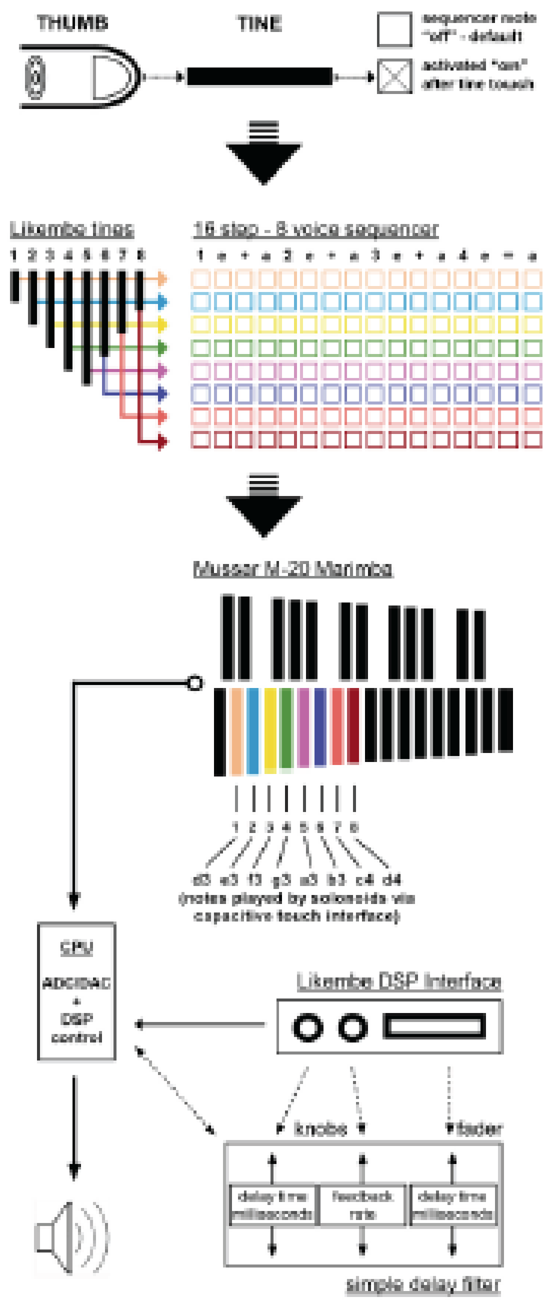


Figure 5.16: Ella system flow

5.3 Lamellophone Tine Excitation

A system for actuating the tines of the instrument electromagnetically has been developed, but not yet integrated into the existing instrument. This aspect of acoustic actuation was developed in collaboration with Jeff Snyder *(Director of Electronic*

Music, Princeton University). Experiments, tests, and implementation done on his behalf were contributions towards [131].

This aspect of acoustic actuation was developed in collaboration with Jeff Snyder (Director of Electronic Music, Princeton University). Experiments, tests, and implementation done on his behalf were contributions towards [131].. This work is primarily based on prior developments by Berdahl [42], McPherson [82], and Britt [110]. McPhersons Techniques and Circuits for Electromagnetic Actuation outlines the possible methods that can be employed for this purpose, this design derives from the force on a ferromagnetic object.

Britt [19] suggests that the complexity of the circuits used to drive both McPherson's and Berdahl's electromagnetically enhanced pianos/strings can be reduced when the system is driving an object without significant harmonic overtones. This is possible because a simpler amplifier that lacks a linear response can be used, a square wave input will produce a sine wave output on the physical vibrating body - the harmonic overtones of the input signal are ignored by the vibrating body. However, in [19], the large mass of the vibraphone bar necessitates a powerful amplifier[130].

In addition to sharing the same enharmonic overtone property as vibraphone bars, the lamellophone tines have added advantages from their low mass and natural ferromagnetic properties, which allow for even greater simplification of the amplifier. The current amplifier design uses a single-ended 12V supply driving a single MOSFET transistor with a freewheeling diode in parallel. This configuration is effective at driving the tines with a usable amplitude, is very low cost, and requires minimal heat-sinking. The most expensive element of the system is the electromagnets (ELMATU021020[3], 28AWG), which cost 20 a piece in the quantities purchased for the project. The single-ended nature of the amplifier system also eliminates the need for the 3-state waveform required by the vibraphone bar, so a much simpler PWM output directly from a microcontroller pin can serve as the driving frequency for EL-La. Polyphony handling for the current system is an adaptation of the system used by the EM-Vibe, described in [19]. Generation of the multiple PWM waveforms is handled by a series of inexpensive microcontrollers communicating over I2C, a significant advantage in cost over the 8-channel audio interfaces used in the previous systems mentioned.

[3] SolenoidCity.com

5.4 Auto-monochord

5.4.1 Introduction

EMMIs AMI [137] inspired the design goals of STARI (Self-tuning Auto-monochord Instrument)- being a singular, streamlined unit with a self-contained pick-up/amplification system (Figure 5.17). Its actuators feature a picking mechanism similar to the Guitarbot, electro-magnetic actuation and a solenoid beater hitting the strings similar to Trimpins pianos, a string dampening mechanism, and a fretting system that affords the production different frequencies.

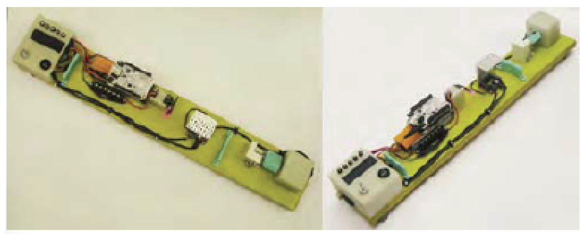


Figure 5.17: STARI

The motivation for this work is to explore the possibilities of musical-proprioception in autonomous, self-actuated musical instruments. Its intended uses are in computer music pedagogy: physical music computing- sensor technologies, micro-controllers; actuation- motors, solenoids, electro-magnets; embedded DSP- creative audio filter design, MIR, and synthesis; frequency analysis pertaining to music theory and audio perception; rapid prototyping- 3D printing, hardware design/fabrication; and composition and performance. This section details the instrument's ability to actuate itself for analysis and self-tune.

5.4.2 Design Considerations

The monochord (Figure 5.18) demonstrates the mathematics and resultant frequencies of musical pitch. The tonics (open string) next higher octave is half the strings length (2:1). A third of the strings length is three times (3:1) the strings root frequency. This relationship will continue exponentially.

Our design required the instrument to self-tune to a particular frequency using a motor to adjust the string tension up or down and one to actuate the string. STARI is meant to be portable and compact, yet robust and flexible as a platform for testing various string actuation methods and interfacing techniques.

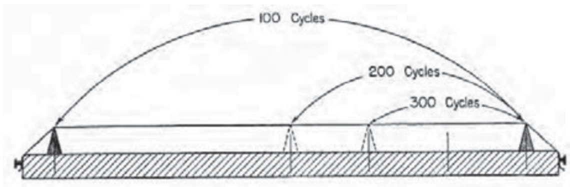


Figure 5.18: Monochord

Design problems that STARI addresses for improved functionality and increased flexibility are as follows: AMI received control input via a PC using Max/MSP. Goals were that STARI be autonomous, eschewing the PC dependency. Max/MSP requires a PC and is relatively expensive. Pd is used instead, being free and ubiquitous within the computer music community. Pd runs on the Beaglebone, being Linux compatible. This affords embedding the computing environment on the instrument, while taking advantage of the software s existing DSP libraries for MIR analysis. Like the Guitarbot, AMI is tuned manually. STARI design goals were to develop a self-aware instrument capable of sensing its own frequencies played in real-time in order to continually monitor and correct itself autonomously during use, should the instrument detune. This is in response to user experience with Guitarbot.

An important aspect to consider when designing the frequency estimation-to-string actuation process is a tuning threshold (Table 5.2) of acceptable estimation. A certain range of tolerance is necessary when determining if the string is in-tune or not. In a conventional, Western equal-tempered scale (A=440Hz), the octave is divided into 12 discrete semitones. Each semitone consists of 100 cents, allowing for extremely fine tuning. (Table 5.2) shows example for the relationship between target, frequency and its high F_high and low F_low threshold, respectively.

Tuning Threshold
target = 200Hz desired frequency
n = 30 cent threshold
high = F1 (2(n/1200)) = 203.4Hz
Flow = F1/(2(n/1200)) = 196.5Hz

Table 5.2: String tuning threshold formulas

5.4.3 System Description

STARI utilizes a collection of commercially available electrical components, guitar string and pickup, a micro-controller, microcomputer, and custom built 3D printed structural components (Figure 5.19). The instrument is implemented using a motor driven tuning mechanism. The stringss signal is acquired via a passive single coil electro-magnetic guitar pickup and routed to the Beaglebone for analysis and processing. The signal is output to a speaker for amplification. The string is actuated by standard guitar pick fitted to a servo motor and controlled natively from Pd or remotely via external interface. The string tunes itself using the embedded analysis software determining the frequency. Control signals are sent to the stepper motor to tighten/loosen the string, raising/lowering the pitch respectively, until reaching the target pitch. (Figure 5.19) shows a flowchart of STARIs operation.

2a Mechanical Design

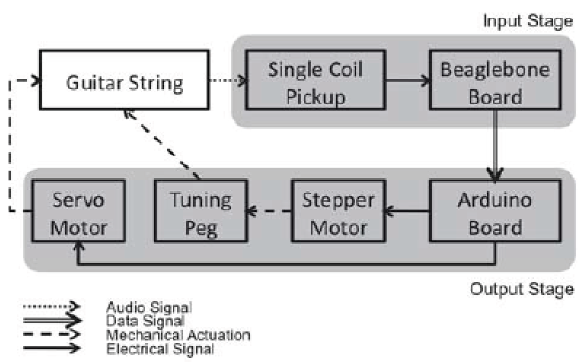


Figure 5.19: STARI: flowchart

A string tensioning system fixed to a board measuring 25 x 4 x .75 was designed. The bridges (Figure 5.20) were custom designed (using the free Autodesk Inventor Fusion program) and 3D printed and spaced 14 apart.

The string was fixed to the board through a hole- the strings stopper fixed to the body underside the same as an electric guitar. The other end was wound to a standard tuning peg mounted to a custom bracket designed to withstand the tension of a tuned string. Because this force was relatively severe, a substantial bracket was required to fully brace the tuning peg. The end of the peg meant for turning was fitted with a custom bracket fixed to the stepper motor axis. The stepper motor presented substantial body vibration requiring a secure enclosure to brace it to the

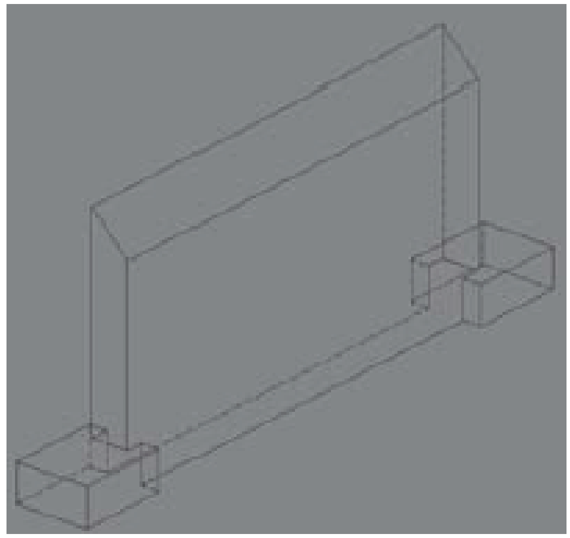


Figure 5.20: STARI: bridge

wood. Large, dense rubber feet to further reduce the vibrations in efforts to reduce the resulting noise produced when in use were selected. (Figure 5.21) shows the 3D printed mounts.

An H-bridge is mounted next in line, bolted to the wood foundation. The servo motor required a bracket for mounting. We found that it oscillated, interfering with the motors performance and ability to pluck the string. We solved this by bolting it to a 3D printed bracket (Figure 5.22). It was mounted just off center from the string with just enough reach for the pick to pluck the string sufficiently. A small slit was cut into the arm of the motor so a pick could be fit and glued on. This was a convenient solution and prevented having to alter the structure significantly. The wiring from the stepper motor was wrapped in heat shrink to keep the unit tidy and secure. The battery pack was zip-tied to the body of the wood through holes drilled into the body, the Beaglebone was mounted above (Figure 5.23). The guitar pickup is placed nearer the opposite bridge from the stepper motor to avoid magnetic noise interference and plugged into the Beaglebones audiocape[4] via 1/8 connector.

The Arduino and custom shield were housed in a canopy designed without need for permanent mounting so the user can easily maintenance the circuit (Figure 5.24). The backside has a small opening to access the USB port of the Arduino and the front is open so the cables can be easily managed. Five LEDs are mounted for real-time visual feedback of the systems state, as well as an LCD display for graphical feedback.

[4] $https://elinux.org/Beagleboard:BeagleBone_Capes$

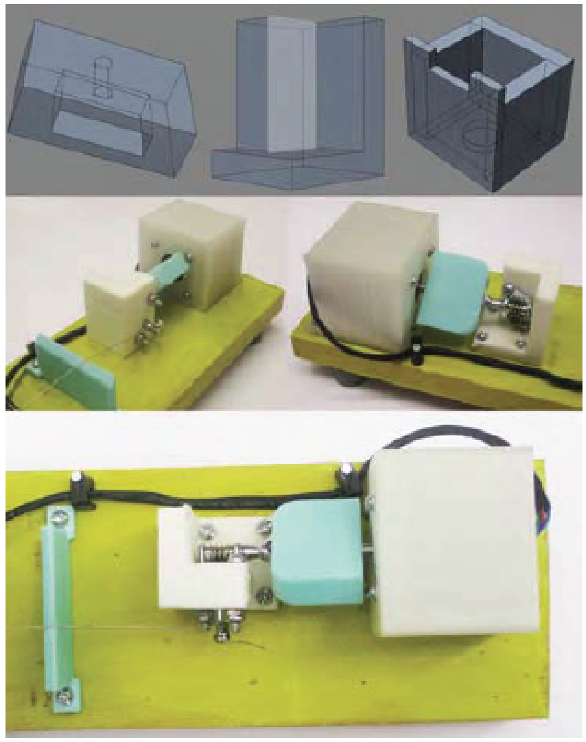


Figure 5.21: Motor enclosure, tuning mount, bridge

Switches provide a simple, proprietary interface for testing. The device can also be interfaced remotely via the Beaglebones USB port.

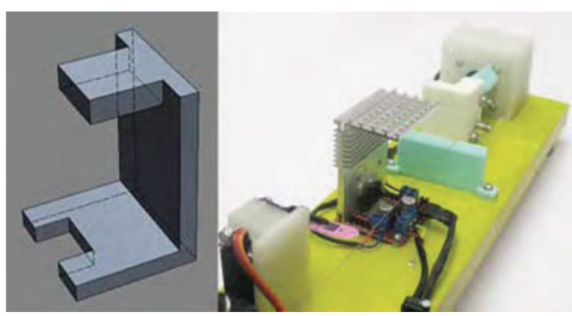


Figure 5.22: Servo/guitar pick and h-bridge

2b. Electrical and Software Design

The operation of the self-tuning string system is coordinated by a Single-Pole-Double Throw (SPDT) switch (S1 in Figure 5.25). The switchs center position is the default mode where power to the motors is cut; thereby eliminating the idle noise they produce when not activated. The switch selects between Reset (L position) and Tuning (R position) operation mode.

In Reset mode the system unwinds the string, preparing the system for tuning operation. Tuning mode is the systems main operation mode. In this mode the process of plucking and detecting the strings frequency and adjusting its tension takes place. To better understand the process involved, the system has been broken down into an Input

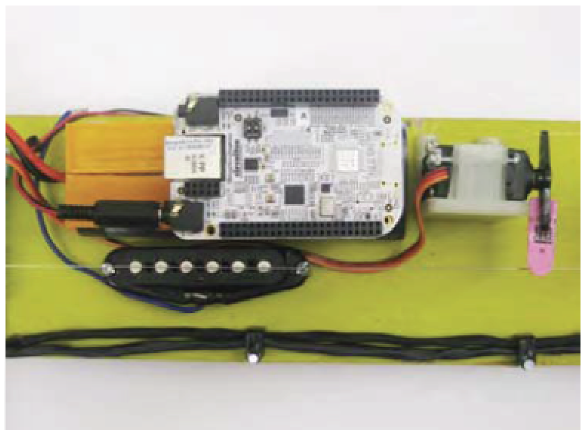


Figure 5.23: Beaglebone, battery, pickup, pick unit

Stage and an Output Stage. The Input stage deals with the capture of the signal from the string and subsequent processing to determine its frequency. The Output stage deals with the control and actuation of the motor to adjust the strings tuning. The output stage is done using the Arduino UNO board.

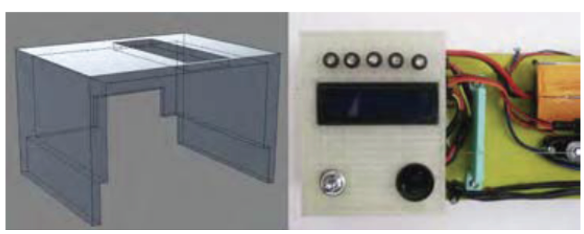


Figure 5.24: Canopy model/interface

(Figure 5.25) shows the schematic for the current build of the functioning system. (Table 5.3) shows the pin configuration for associated schematic and current system build.

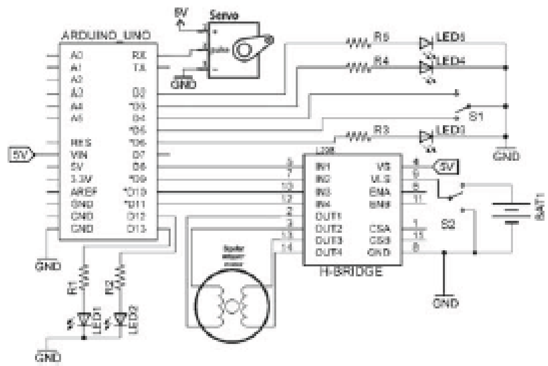


Figure 5.25: System schematic

Input Stage

In this stage the string is plucked and a transducer captures the signal. The transducer is connected to a processing unit, which performs operations to determine the frequency. Noise coming from the stepper and servo motors interfere with the strings audio signal. This is avoided by using a single coil guitar pick up, electromagnetically converting the strings vibration to an electric signal. This type of transducer is independent of acoustic signals and thus wont pickup errant environmental noise; however, it did present the magnetic noise from the motors. This did not interfere with the analysis software, but introduces noise in the audio output.

The frequency estimation expands previous research [79], which continues to be extended as new digital musical applications arise. The framework provides analog-to-digital conversion (ADC) by way of a TLV320AIC3106 codec provided by a Beaglebone expansion board, known as the Audio Cape. This expansion also provides digital-to analog conversion (DAC) for output. Since this framework provides audio interfacing, the transducer is connected directly into the 1/8 audio input jack in the Beaglebone. The flowchart in (Figure 5.26) shows the process and components for this approach.

The framework uses the robust, object oriented, audio programming language Puredata. Pd is open-source and free. Programming is done in a graphical environment called a patch by way of interconnecting processing objects, resembling the patching paradigm in analog audio signal chains (or even more simply- a flowchart). Being oriented for audio applications and given its relative ease of use (as compared to low-level C programming typical to embedded DSP), PD facilitates the capture of audio, as well as custom digital signal processing (DSP) techniques such as frequency

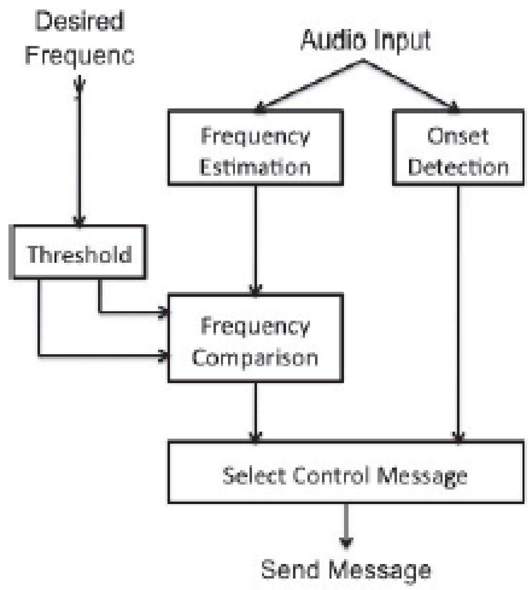


Figure 5.26: Beaglebone flowchart

estimation.

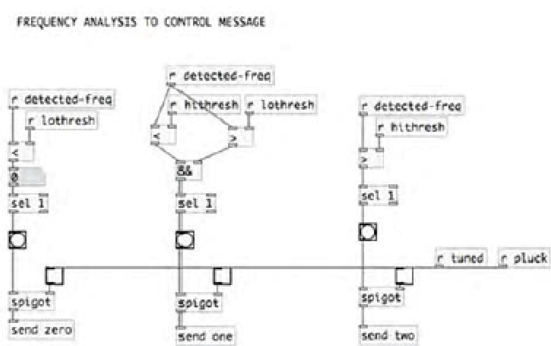


Figure 5.27: Control message management in Pd

The PD patch continuously analyzes the audio from the codec, and compares the estimated frequency to the target frequency. Simultaneously, a serial connection is created to communicate from Pd in the Output Stage to control motor actuation. The frequency difference is translated to a simple control message system with three possible states: tune up, tune down, dont tune. (Figure 5.27) shows how the control messages are generated based on frequency estimation in PD.

After the control message is sent from Pd to the Output Stage via the serial connection, the latter needs to send an acknowledgement message to allow Pd to send another control message. A frequency error tolerance is incorporated in the Pd patch as well, based on the tuning threshold. (Figure 5.28) shows the stage corresponding to the tuning threshold. The detection accuracy of Pd was consistently sensing within 3Hz of the target frequency based on the thresholding settings.

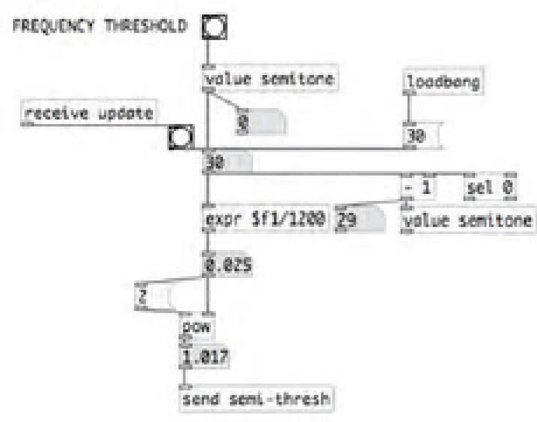


Figure 5.28: PureData patch for tuning thresholding

Fiddle, a standard object for frequency estimation in Pd [77], was used. Fiddle continuously estimates the pitch and amplitude, as well as provides sinusoidal peak analysis. The incoming audio from the plucked string is analyzed and its fundamental frequency is estimated. Once the patch detects that the string has been plucked, the difference between detected frequency and desired frequency is calculated, and translated into a control message. The way Pd detects that the string has been plucked is by amplitude onset analysis. Bonk (another object described in [77]) detects onsets and sends a message when the audio passes a user defined threshold. In this case, the message is set to engage the tuning apparatus. (Figure 5.29) depicts the onset and frequency analysis stage.

Output Stage

The output stage refers to the systems actuation of the string via servo motor and activation of the stepper motor to tune it to the target frequency. In both approaches this stage deals with getting the control signal after the estimation of the difference between the target frequency and the frequency of the string and turning the stepper motor to adjust the tension of the string. Also this stage includes the appropriate visual feedback associated with the tuning operation (Table 5.3).

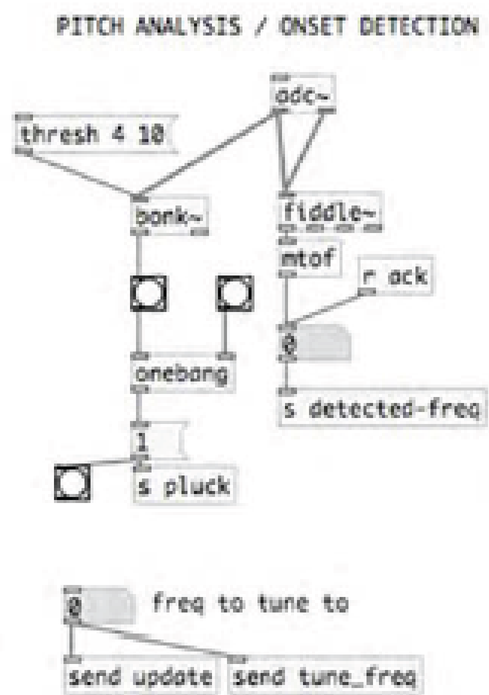


Figure 5.29: Frequency estimation/onset analysis

Stepper Motor

The stepper motor turns the tuning peg and maintains string tension. A Bipolar stepper motor (42BYGHW811) was chosen, which easily drives the unloaded tuning peg. An estimate of the maximum tension that the motor can withstand can be made by studying the physical properties of the vibrating string. It is known that the fundamental frequency of a tensioned string is equal to the velocity of the wave divided by twice the length of the string, with the velocity of the wave equal to the square root of the string tension divided by its mass per unit length. Since the parameter wanted is the tension T the (equation 5.1) is as follows[5]:

$$T = (4.45 \times ((2 \times L \times f)^2) \times UW)/386.4 \qquad (5.1)$$

where f = frequency in Hz; L = string length from bridge to nut; and UUW = Unit Weight or Mass per Unit Length (MPL) of the string. In this project, L = 35.5mm and f =261.6Hz (middle C). The guitar string that was available is the DAddario PL019, which has a UW = 0.01428g/cm. Solving the equation above, the

[5] $www.daddario.com/upload/tension_chart_13934.pdf$

Pin#	Description
0	Red High LED
2	Servo signal
3	Yellow High LED
4	Tune Mode (switch SW2)
5	Reset Mode (switch SW2)
6	Green High LED
8	Stepper Motor 'In 1'
9	Stepper Motor 'In 2'
10	Stepper Motor 'In 3'
11	Stepper Motor 'In 4'
12	Yellow Low LED
13	Red Low LED

Table 5.3: STARI: GPIO assignments

resulting tension needed for the string is 5023.75gf.

Motor torque was calculated using the following (Equation 5.2):

$$\tau = F \times r \tag{5.2}$$

The length of the lever arm r is 2 cm, represented by the motors coupling device to the tuning peg. The stepper motor has a rated torque of 4240gf.cm, which results in a = 8400gf. It was determined that this motor is more than capable of driving the tuning peg past the required string tension to reach the target frequency.

H-Bridge

To drive the motor, a ready-made dual motor driver based on the L298N H-bridge was chosen. The order in which the wires coming from the motor connect to each of the **Out** headers of the driver are as follows: (**Red = 1, Blue = 2, Black = 3, Green = 4**). Another feature is the user accessible 5V regulator, which in this case, has been enabled to power the Arduino from the battery coming into the driver rather than from the USB port, making the system autonomous, without need of an external power source. In the input stage, the stepper motor is set to move ±15 steps for every control message coming from the Beaglebone board. This parameter ensured a controlled variation of the strings tuning and a more accurate result.

Chapter 6

Contributions to other musical contexts

6.1 Extended Framework

In this section, additional contributions beyond the pitched percussion family of instruments are described. Themes and ideas described in this thesis are used to inform the design of a reconfigurable novel guitar controller, a minimally-invasive sensing platform for the trumpet, and a graphical user interface for geometric rhythm theory. These contributions were collaborations. The work on guitar augmentations was developed by Duncan MacConnell, an undergraduate research assistant in computer science at the time. The trumpet research was developed by Master of Engineering research assistant Leo Jenkins. In both projects my role was as design engineer, where I contributed towards high-level theoretical design concerns, sensor implementation and/or mapping strategies.

6.1.1 Guitar

RANGE- Reconfigurable Autonomous Novel Guitar Effects

The RANGE guitar is a minimally-invasive hyperinstrument incorporating electronic sensors and integrated DSP. It introduces an open framework for autonomous music computing eschewing the use of the laptop on stage. The framework uses an embedded Linux microcomputer to provide sensor acquisition, analog-to-digital conversion (ADC) for audio input, DSP, and digital-to-analog conversion (DAC) for audio output. The DSP environment is built in Pd. The sensors selected can be mounted in a variety of ways without compromising traditional playing technique. Integration with a conventional guitar leverages established techniques and preserves the natural gestures of each players idiosyncratic performing style. The result is an easy to replicate, reconfigurable, idiomatic sensing and signal processing system for the electric guitar requiring little modification of the original instrument

Electric guitar players have utilized audio effects since their inception. An extensive variety of DSP guitar effects are offered commercially, some of which even provide a code environment for user modification of DSP algorithms[1]; however, in most cases the functionality of these devices is specific and their programmability is limited. These commercial audio effects are typically implemented either as foot pedals or as separate hardware devices. An alternative is the use of a laptop and audio interface to replace the dedicated guitar effects. This approach is generic in the sense that any audio effect can be implemented as long as the computer is fast enough to calculate it in real-time. Using a laptop is also completely open, flexible, and programmable. However such a setup requires more cables, more power, and is cumbersome to transport and awkward on stage [36]. In both of these cases (dedicated hardware or laptop) the control of the effects is separated from the actual guitar playing as shown in (Figure 6.1)

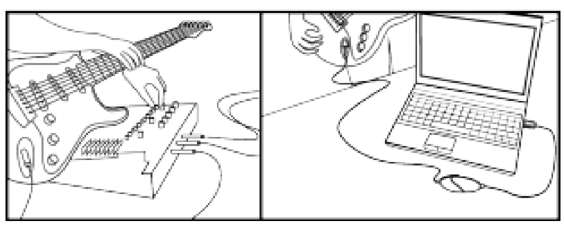


Figure 6.1: Guitar/FX interaction

The guitar has been a subject to augmentation, perhaps, since its advent. Historic examples include the Guitar Organs, the analog synth controller Stepp DGX MIDI guitar[2], and the Roland G303[3]. Systems for mobile devices acting as the DSP/sensor interface host have also become common [139].

Augmented guitars have also been explored. MIT's Chameleon Guitar has multiple soundboards, each equipped with piezo sensors and DSP filtering to simulate the

[1]https://line6.com/tcddk/
[2]http://www.muzines.co.uk/articles/stepp-dg1-electronic-guitar/253
[3]http://www.joness.com/gr300/G-303.htm

guitar tones offered from different wood [153]. Another example, the Moog Guitar, is an electric guitar with onboard sliders that control augmentation of the guitars traditional sound by sending electro-magnetic energy into strings. This allows for infinite note sustain, while similarly pulling energy from the strings creates short staccato sounds. Edgar Berdahl introduced a similar idea in his Feedback Guitar [42].

The DUL Radio [74] from the Center for Digital Urban Living at Aarhus University, Denmark is a wireless accelerometer sensor package designed for artists to use with Pd or Max/MSP. The group demonstrates the device by attaching an accelerometer to the headstock of the guitar for 3D gesture tracking. Unfortunately it was not possible to integrate it with the RANGE system because of the LINUX requirement, which wasnt supported by the DUL drivers at the time of this project.

MOTIVATION

In designing an augmented guitar instrument, consideration must be taken to ensure the extensions do not inhibit traditional guitar technique. Effort should be made to create intuitive control interfaces that take advantage of the guitar players natural performance technique. Traditional audio effect units and commercial DSP solutions tend to disregard this, forcing the musician to interact with musical parameters by way of non-musical gestures: turning a knob or adjusting a fader [51]. This conflicts with the guitarists normal gestural interaction, fails to convey any meaningful event information, and can even act as a distraction for the audience [88].

There has always been a union of guitar and effect despite a separation of guitar playing and effect control. To address this issue, minimally invasive sensors have been integratred on the body of the guitar to allow natural and intuitive DSP control. The RANGE system was designed for use in performance contexts to allow guitar players more expressivity in controlling DSP effects than conventional pedal controllers provide.

The proximity of the sensors to the guitarists natural hand position is important, as it allows the guitarist to combine DSP control with traditional guitar playing technique. Like the Moog Guitar, the sensors sit flat on the guitar body, eliminating any interference with a guitarists performance technique. Further, we have reduced the hardware dependencies, cabling, and power requirements to a minimal footprint. Design goals were motivated by the desire to shift away from the cumbersome and distracting laptop on stage in exchange for a smaller, open architecture. This framework is designed to take advantage of low-cost electronic components and free open-source software, facilitating reconfiguration and adaptation to the specific needs of different instruments and musicians.

SYSTEM DESCRIPTION

The RANGE system from previous work developing robust hyperinstrument prototypes, which provides audio input, output, and sensor acquisition. The system itself provides a completely open platform for designing and testing hyperinstruments. The hardware and software components that comprise this system are modular in nature and the configuration is designed so that any user can adapt this work for their own use. For this implementation, the sensor interface consists of three membrane potentiometer strips mounted on the body of the guitar which feed into the analog inputs of the embedded Linux computer. The Beaglebone is used for its flexibility and low cost. The guitars audio signal goes directly into the Beaglebone for analysis and processing using an Audio Cape for ADC/DAC. Pitch tracking is performed on the incoming audio signal using Fiddle [105] and used to generate control data. The control data and DSP is managed in Pd. The potentiometer outputs are mapped to continuous controller values that modify the parameters of effect parameters, oscillators, and filters. These potentiometers offer the guitarist a broad range of interface solutions and sound design possibilities in a small embedded format that has previously been only possible with a laptop. The system has relatively low cost (all prices in US dollars/at the time of this project): Beaglebone (89), audio cape (58), and membrane sensors (3 13 = 39) for a total of 186 USD. (Figure 6.2) shows a schematic diagram of the system.

1. Analog Sensor Input

Membrane potentiometers are a common sensor for capturing musical data, and are often incorporated into hyperinstrument design. Adrian Freed [47] provides a detailed look at force sensing resistors, membrane potentiometers, and other sensors. His Many and DuoTouch Augmented Guitar Prototype provides simple and elegant circuit solutions to achieve desired sensor behaviour for musical applications (Figure 6.3).

The RANGE guitar is equipped with three 50mm SoftPot membrane potentiometers. These sensors are arranged on the body of the guitar, near the volume and tone

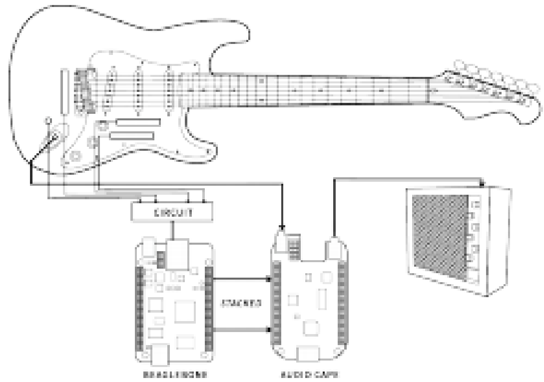


Figure 6.2: Schematic of RANGE

Figure 6.3: Adrian Freed's augmented guitar prototype

controls. This arrangement allows the guitarist to easily access the sensors, and the orientation affords comfortable interaction. The sensors are limited to the body of the guitar corresponding to the expressive hand of the guitar player. The expressive hand, responsible for the rhythm and dynamics of the guitar, is most suited for acute sensor control. In addition to the three touch sensors, toggle switches are also mounted to the guitar to provide a simple method for switching software state.

Traditional potentiometers use the position of a sliding wiper to determine resistance. Membrane potentiometers function similarly, providing a variable resistance level based on the position of the user's finger. The main difference is that membrane potentiometers only allow current to flow when the membrane is pressed, and so the value is lost when the user's finger is removed. The RANGE's sensor input behaviour must be consistent and stable in order to be used musically. Specifically, the instrument design requires that the analog input values remain when the membrane is not pressed.

In order to secure a stable and usable signal from the membrane potentiometers, pull-up resistors are used. This forces the potentiometer to open circuit when the finger is removed. A simple software solution is used within Pd for detecting when the membrane is forced open, and the previous buffered value is retained. (Figure 6.4) shows the corresponding circuit.

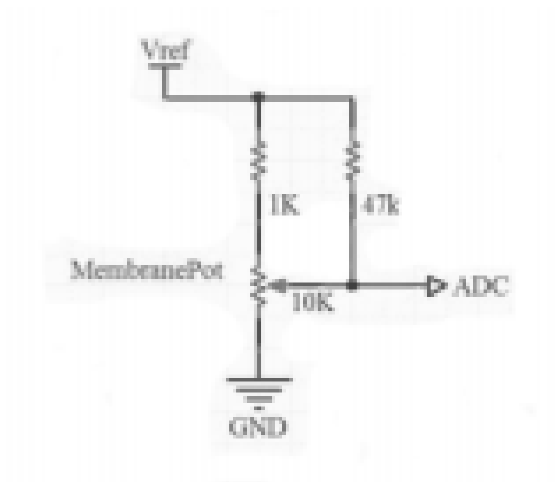


Figure 6.4: Membrane potentiometer circuit"

2. Hardware, Software, and Latency

The RANGE uses a Beaglebone microcomputer, which provideson-board GPIO and ADC pin access, as well as UART, I_2C and SPI. It features an *ARM 600 MHz Cortex-A8core* using the ARMv7-Architecture, as well as full USB and Ethernet support. The Beaglebone is becoming widely supported in the embedded computing community [13][83], and many expansion Capes[4] are being developed to provide an array of hardware interaction opportunities. The Beaglebone Audio Cape provides audio input and output by way of two 3.5 mm connectors, and supports sampling rates up to 96 kHz for capture and playback by way of the capes TLV320AIC3106 codec. The system described provides a complete DSP platform, allowing users to connect to the Beaglebone via ethernet for rapid interface prototyping.

[4]https://beagleboard.org/cape

For this iteration, Ubuntu 12.04 was used, which facilitates Pd installation and interfaces well with the audio codec provided by the Beaglebone Audio Cape. To provide access to the Beaglebonés GPIO and ADC pins, a Pd external has been developed. The ADC provides 12-bit values, which are accessed by the external by directly reading the corresponding files in the userspace. The external is designed to report analog and digital pin values each time the object receives a bang message. In this way, pin values can be obtained at any rate, and can be coupled with other musically timed events within the patch. The control data is not altered in any way by the external that retrieves it, as it is meant to make any sensors data available within Pd.

Control of digital audio effects has been a desired function of the RANGE from its inception. The stable sensor values allow for reliable control, while placing the sensors directly on the body allows the guitarist faster and more intuitive interaction. Analog values obtained by the Pd external can be scaled and mapped to any control. Therefore specific effect parameters (delay length, feedback level, filter frequency, etc.) are adjusted by the touch potentiometers. This simple prototyping system is robust but offers a lot of flexibility and potential. The potentiometers can also provide an intuitive control interface for synthesis applications. Some novel applications include controlling oscillator frequency, filter frequency/bandwidth, MIDI note attribute, and envelope values (attack, decay, sustain, release). This application allows the RANGE to be used as a versatile synthesizer controller while the guitar can still be played as usual. (Figure 6.5) shows an example mapping, with sensor input controlling a typical electric guitar effect chain on the left and common DSP applications on the right.

In contrast to pure controller approaches that utilize a laptop for DSP, RANGEś goal is to use the Beaglebone for both control and DSP. Many modern guitar effects are actually internally implemented using a dedicated embedded DSP chip even though to a guitar player they appear similar to traditional analog pedals. RANGE makes this DSP functionality accessible providing a wide range of possibilities for both digital audio effects and their control. In order to be a viable platform for this purpose it is critical that the overall system latency is appropriate for music applications. RANGE provides simple sensor and audio throughput, using Pd with the ALSA API.

For latency tests, it is important to perform measurements under different system (CPU/DSP) load applications [78]. For this system, all tests were performed in the "normal state" (audio throughput, no effects processing) as well as the "use state"

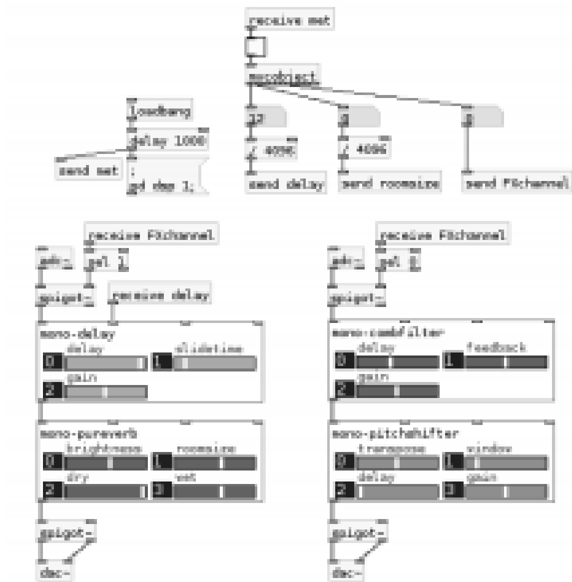


Figure 6.5: Pd guitar fx

(audio throughput, effects processing). In the normal state, audio latency for the system corresponds to the audio delay set by Puredata. With Puredata set to 10ms audio delay, there is a total system delay of 10ms. With effects engaged, the use state latency is measured at 12ms. This difference can be attributed to the system load increase from the signal processing. For pitch analysis, normal and use state behaved the same, measuring a total pitch analysis latency of 15ms. All results reflect usable latency levels, for audio applications, as they approach the general latency goal of 10ms ([78], [101]). (Figure 6.6) shows the audio and pitch-tracking latency.

6.1.2 Trumpet

EROSS- Easily Removable, wireless Optical Sensing System

This section introduces a minimally-invasive, wireless optical sensor system for use with any conventional piston valve acoustic trumpet. It is designed to be easy to install and remove by any trumpeter. The goal is to offer the extended control afforded by hyperinstruments without the hard to reverse or irreversible invasive modifications that are typically used for adding digital sensing capabilities. Utilizing optical sensors to track the continuous position displacement values of the three trumpet valves are transmitted wirelessly and can be used by an external controller.

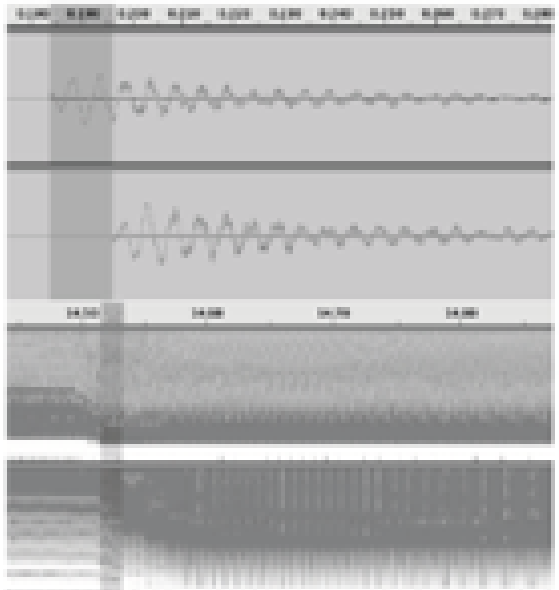


Figure 6.6: Audio latency measurements

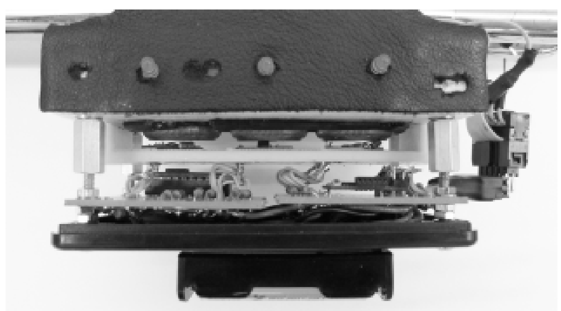


Figure 6.7: EROSS

The hardware has been designed to be reconfigurable by having the housing 3D printed so that the dimensions can be adjusted for any particular trumpet model. The result is a low cost, low power, easily replicable sensor solution that offers any trumpeter the ability to augment their own existing trumpet without compromising the instruments structure or playing technique. The extended digital control afforded by EROSS is interweaved with the natural playing gestures of an acoustic trumpet. This seamless integration is critical for enabling effective and musical HCI (figure 6.7).

BACKGROUND

As discussed, hyperinstruments are expanded acoustic musical instruments that use digital sensors to capture detailed aspects of the performer gestures as well as provide additional expressive capabilities through digital control. They enable the musician to access the diverse possibilities afforded by digital control while at the same time leveraging the skill developed through years of training by professional musicians. Since then, a variety of different hyperinstruments, designed for the main musical instrument families, have been proposed. Early representative examples include: the *Hypercello* (strings) [102], the *Metasaxophone* [21] and *Hyperflute* (winds)[44], the Morrison Digital Trumpet[5], and the Cook/Morrill *trumpet controller* (brass) [100].

Any acoustic instrument can serve as the basis for designing and building a hyperinstrument and the concept has also been applied to non-western instruments such as the sitar [2], and the Gyil African xylophone [113]. Despite the potential of hyperinstruments to enable extended performance techniques, their adoption has been slow and idiosyncratic. This is mostly due to the following factors:

1. they are expensive custom-made devices that are hard to replicate (typically there is only a single instance in existence);
2. they frequently require invasive modifications to the already expensive (both in cost and skill) original acoustic instrument;
3. their operation and maintenance requires significant technical expertise with electronics.

The focus of the work presented herein is to augment traditional trumpet playing with digital control. There is a long history of approaches that have been proposed to achieve this. The most generic approach is to use external controllers such as foot pedals, knobs, sliders, and keyboards to control digital processes as well as filters/effects on the audio signal produced by the horn. An iconic example of this approach was the use of electric guitar effects by Miles Davis to modify the sound of the trumpet during his electric phase in the 1970s[37]. As any controller can be used for this purpose, this approach provides a very rich set of control possibilities to the trumpet player. In addition, it requires no modification to the acoustic instrument other than attaching a microphone to the horn. However, interacting with the

[5] http://www.digitaltrumpet.com.au/

controllers is external to the trumpeter's technique. This creates additional cognitive load and is similar to trying to play two instruments at the same time.

An alternative approach is to directly introduce sensors for digital control on the body of the trumpet making it a hyperinstrument. A good overview of different approaches and sensors that can be used for augmenting the trumpet can be found in the Master thesis of Thibodeau [125]. The pioneering Cook and Morrill [**?**] [100] was built to create an interface for trumpeter Wynton Marsalis. Sensors on the valves, mouthpiece, and bell enabled fast and accurate pitch detection and provided extended computer control. Another well-known example is the Mutantrumpet, a hybrid electro-acoustic instrument designed and evolved over many years by composer and inventor Ben Neill [92]. The Mutantrumpet started as an acoustic instrument (three trumpets and a trombone combined into a single instrument). In the mid 1980s electronics were integrated to the instrument in collaboration with synthesizer inventor Robert Moog[148] and in the 1990s the instrument was made computer interactive. By attaching the sensors directly on the instrument the digital control is more easily accessible by the player and therefore is more naturally integrated with the traditional way of playing the instrument. However, each instrument is unique, idiosyncratic and custom-made which makes replication and therefore adoption difficult. The modifications to the instrument are extensive (sometimes even radically modifying the original instrument as in the case of the Mutantrumpet) and the sensing apparatus is hard to remove, if not required. Finally, the detailed design plans and component parts of most augmented trumpets (and hyperinstruments in general) are not publicly available.

Here is presented a low-cost, easily removable and minimally invasive optical sensing system (EROSS) that provides continuous control data from the position of the valves on the acoustic trumpet. The housing has been 3D printed making it fully customizable for any standard trumpet. The sensors and housing bracket mount under the valves with-out obstructing the conventional playing position and the two pieces are connected together via magnets without altering the horn's structure or requiring cumbersome fastening. The approach has been inspired by the Electrumpet [71], which is an enhancement of a normal trumpet with a variety of electronic sensors and buttons. The Electrumpet can be attached and detached to any Bb trumpet. It is common for trumpet players to attach various types of mutes to their horn. It is hoped that making the sensing apparatus no more difficult to attach and remove than a mute will facilitate adoption. Like the Electrumpet EROSS' design plans are available, the sensing apparatus is removable, and we utilize wireless communication. This approach is more naturally integrated as it focuses on providing continuous control data from the positions of the three valves used for playing. In contrast, the Electrumpet provides additional valve-like potentiometers and buttons that are not part of regular trumpet playing. The use of magnets and 3D printing for the housing is an additional difference and a contribution of EROSS. For sensing the continuous valve positions utilize optical sensing, unlike the variable resistor potentiometers used in the Electrumpet. This optical sensing technique has been proposed in the design of another augmented trumpet, which was done as a course project at Cornell University [32]. Unlike our approach it was used for discrete, rather than continuous control. Figure 1 shows EROSS mounted on a trumpet.

MOTIVATION AND DESIGN

A long term objective for this research is to design and establish a robust, low cost, reconfigurable sensing hardware and software framework that can be adapted to various musical instruments in order to give them hyper-instrument capabilities without requiring invasive and hard to reverse modifications to the acoustic instrument. There is a strong need for such a framework because of the high cost and hard to replicate nature of existing hyperinstrument designs. Advances in open source software, electrical sensors, micro-controller frameworks such as Arduino, and 3D printing open the possibility of creating reusable, adaptable designs that can easily be applied to existing instruments without requiring extensive technical expertise and leveraging traditional playing technique. It is hoped that as electronics on stage are becoming more ubiquitous in many musical genres, approaches like EROSS will reach larger communities of users. The proposed trumpet sensing system is a good case study of how such a framework can be used for instrument augmentation.

A. Design Considerations

The proposed system, compared to other augmented trumpets that have been described, is minimal. It only provides continuous control information from the position of the three piston valves that are used for playing. This design decision was influenced by the designed principles for new musical interfaces outlined by Cook [31] that emphasize simplicity (Programmability is a curse and Smart instruments are often not smart) and integration with existing playing techniques (Copying an instrument is dumb, leveraging expert technique is smart). It was also wanted for the system design to be minimally invasive and easily removable for the reasons outlined above. The first one deals with the fact that the trumpet should remain physically unmodified, even not removing the valve bottom caps. These caps have a hole at the center, which would work as an advantage. The location of the sensor would be such that the IR emitter would shoot a pulse through this hole and the reflection off the bottom surface of the piston would be detected by the photo-pin-diode. All trumpet pistons are hollow cylinders with cylindrical tunnels welded across and an irregular bottom surface with a hole in its center, which is concentric to the hole in the valve bottom cap. This configuration makes the optical path for the sensor multifaceted. When not pressed, the piston is at its maximum distance from the sensor and its bottom surface hole represent a small portion of the area covered by the sensors detection zone. When fully pressed it is at the minimum distance from the sensor and the hole represent the majority, if not all, of the area covered by the detection zone, making it difficult to sense the appropriate distance to the piston's surface. (Figure 6.8) depicts valve's configuration and its interaction with optical sensor.

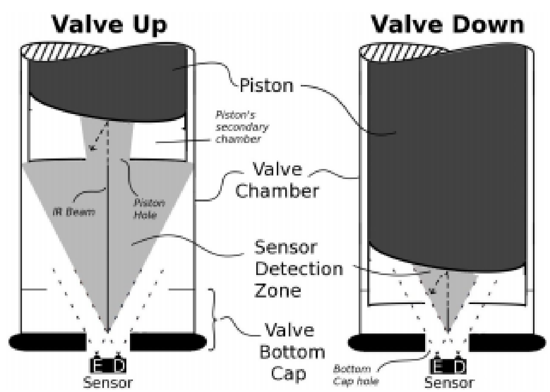


Figure 6.8: Sensor detection zone

Another challenge was the idea of having a system that could be easily attached and detached to the trumpet, but at the same time be sturdy enough to keep the sensors as steadily as possible beneath the valves bottom cap. Optical sensors tend to have high sensitivity, and even the slightest of movements and/or temperature changes can affect the systems accuracy. To compensate for this issue a calibration stage needed to be included when designing the software. Also the system would have to be mounted in a way such that the performer shouldnt compromise their natural grasping technique.

B. Sensor Placement

A critical aspect of the design is the placement of the optical sensors for determining the valve positions. This section describes the empirical investigation that was carried to determine this optimal placement according to a set of design constraints. Five candidate locations near the bottom cap were considered (indicated as $SL_1, ..., SL_5$ in Figure 6.9). The optical sensor consists of an emitter and a detector. For each candidate location, the measured optical sensor readings for eight valve positions (indicated as $VP_1, ..., VP_8$ in Figure 6.10) were collected. The full range of possible valve displacement was divided linearly into these eight valve positions where VP_1 is the fully released valve position and VP_8 is the fully pressed valve position.

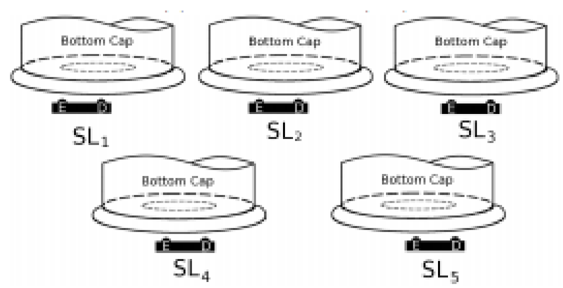


Figure 6.9: Sensor Locations (SL)

Figure 6.10: Valve Positions (VP)

The measured optical sensor readings are expected to be noisy. Thus to determine the best sensor placement, multiple measurements at each candidate location were needed to obtain the noise distribution and a clear set of criteria by which to score each location. The vector of measurements for each configuration are denoted as

$$x_l^p = x_l^p[1], ..., x_l^p[1000]$$

where $l = 1...5$ corresponds to sensor location and $p = 1...8$ corresponds to the valve positions. For each configuration of l,p we calculate the following statistics:

$$\mu_l^p = \frac{1}{N}\Sigma_{i=1}^N x_l^p[i]$$

$$\sigma_l^p = \sqrt{\frac{1}{N-1}}\sigma_{i=1}^N (s_l^p[i] - \mu_l^p)^2$$

$$\rho_l^p = |max(x_l^p) - min(x_l^p)|$$

where $N = 1000$, is the sample mean, is the sample standard deviation, and is the range. For each configuration the number of measurements are calculated C_l^p from x_l^p that fall within $\pm\sigma_p^l$ from the mean μ_l^p. Any configuration for which C_p^l was less than 70%.

Due to the complexity of the optical path, three criteria were used to score the possible locations. The first one is Linearity, because it is desired that the systems transfer function (valve position vs. output data) is as linear as possible. The second one is Dynamic Range, to ensure a better signal-to-noise ratio. The third one is Robustness, because preference is for a location that robustly handles noise, such as the ones induced by the sensor and the mechanics of the system, providing consistent output measurements. A score was calculated for each criterion and each sensor location l.

- **Linearity**

  The curve fitting cost L^1 from a least squares linear fit on the means $\mu_l^1, ...,^8$ of each VP instance corresponding to a particular location l (each curve is fitted to 8 points). Lower cost signifies a more linear response from the system.

- **Dynamic Range**

  The distance in measurement space between the minimum and maximum of two adjacent valve positions: $H_l^p = |min(x_l^p) - max(x_l^p)|$ where $p = 2, . . . , 8$. To obtain a single score for a particular location l each valve position was weighed with the following results: $wh^p = [1.05, 1.15, 1.25, 1.35, 1.45, 1.55, 1.65]$. The weights were determined empirically in order to emphasize the accuracy at the fully pressed position ($p = 8$). The final Dynamic Range score is $H_l = \Sigma_{p=2}^8 wh^p \times H_l^p$. A higher score implies better dynamic range.

- **Robustness**

  The ratio $R_l^p = \sigma_l^p / \rho_l^p$ between the standard deviation and range for each configuration l, p (the d index is dropped based on the first experiment). As defined, this criteria is likely to be sensitive to outliners, but in fact this might be advantageous when taking into account any possible noise, such as what is induced by the EROSS not being mechanically locked when mounted. To obtain a single robustness score for a particular location l each valve position is weights with the following results: $wr^p = [1, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7]$. The final robustness score is $R_l = \Sigma_{p=1}^{8} wr^p \times R_l^p$. The weights were determined empirically in order to emphasize the accuracy at the fully pressed position ($p = 8$). A lower score implies better robustness.

The scores for the three most suitable sensor locations are in (Table 6.1). These scores show that, while SL_4 is the more robust location, SL_1 has the least variation from a linear curve (Linearity criteria) and the wider dynamic range.

Sensor Location	**Linearity** (L_1)	**Dynamic Range** (H_l^8)	**Robustness** R_l^8
SL_1	752	270	8
SL_4	2511	316	6
SL_5	1714	402	20

Table 6.1: EROSS: sensor test results

The scores R_l, H_l, L_l were normalized and added with equal weight to yield a final score S_l. The sensor location $\hat{l}$ that yielded the highest score $\hat{l} = \arg max_l S_l$ was selected for the placing of the sensor and corresponds to the detector being positioned at the center, while the emitter sits at the edge, of the caps hole (SL_1).

SYSTEM DESCRIPTION

The use of optical sensors provided the least invasive option out of different valve sensing approaches considered, such as the ones described in [125], [1]. Since the system is intended to perform range control on the trumpet valves rather than just event detection, a tightly focused sensor implementation was needed in order to achieve accurate results. As in previous work, the systems sensor location is at the bottom of the valves, below the bottom caps for easy retractability of the system. The systems main components are the optical sensors, the microcontroller and the wireless transceiver.

Optical Sensors

A fully integrated proximity sensor was chosen so the constraints on spatial resolution and robustness could be achieved. This sensor is a small surface mount device measuring 4mm (L) x 4mm (W) x 0.75mm (H). It includes an IR emitter, photo-pin-diode and processing circuitry such that up to 16 bits of effective proximity resolution are possible. This sensor has an effective angle of half sensitivity of $\pm 55°$, meaning that if we assume a coneshaped detection zone with a $\pm 55°$ angle (Figure 6.8), then the intensity of radiation of the IR signal is at its maximum in the middle, and half the maximum on the edges. For proximity measurements, the emitter sends a train of pulses at a specific frequency and the detector, tuned to that same frequency, captures the reflected signal, which is subsequently processed and converted into the 16-bit output value, known as counts. The inherent noise from the circuitry is between ± 5 and ± 20 counts. It features an I^2C interface which is supported by the chosen microcontroller, and its performance features are configured by writing to registers in its internal processing unit.

Although it seems counterintuitive, the optimal location SL_1 allows for the IR signal within half of the detection zone to enter the valve chamber, while allowing full exposure of the detector to capture the reflected signal (see Figure 6.9). Also, in principle this location ideally eliminates reflections off of the bottom cap (Table **??**).

Development Board

A ready-made wireless development board (ti-eZ430-RF2500) was used. This package was chosen because of its immediate availability as well as ease of integration with the optical sensors via I^2C, plus the integrated wireless RF transceiver. Also, the fact that this boards size is 33mm (L) x 20mm (W) x 4mm (H) makes it more suitable for a minimally-invasive system. In addition, the microcontroller (MSP430F2274) in this board has some performance features that are relevant for this application. Perhaps the most important feature is the Low-Power mode (LPM), in which the microcontroller shuts down its CPU while in idle state (i.e. waiting for the user to press the activation button). Since the transmitters side (Tx) is powered by a battery and not a power supply from the wall or a computer, this feature is very advantageous, as described later. It is also worth noting that the RAM size for this particular microcontroller is 1KB, so the systems software design had to be constrained by memory availability. The Flash memory size is 32KB and can be used when RAM is not available, however its slower performance must be taken into account.

A. Hardware Design

The system consists of three optical sensors located underneath the bottom valve caps connected via I^2C to the wireless development board.

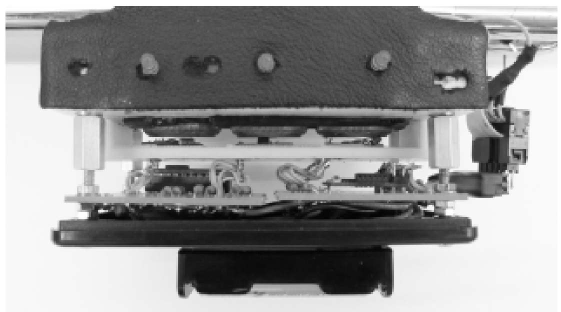


Figure 6.11: EROSS mounted under the valves

Unfortunately all three sensors come preprogrammed with a fixed slave address, which basically defeats the purpose of the I^2C protocol multi-drop capability. To overcome this issue an I^2C-based multiplexing device (ti-PCA9544A) was incorporated into the design. An actuation button (SENSE), strategically attached to a trumpet valve guard so it doesnt interfere with the natural grasp of the instrument, gives the user control over the system to enable/disable sensing operation. Three LEDs (one for each valve) convey basic status information for the sensors. The battery pack was attached at the bottom of the system, as well as an On/Off switch and a Calibration button (CAL). A removable mounting structure was 3D-printed. The structure consists of two C-shaped sliding brackets coupled around the valve chambers for mechanical support. One bracket holds the circuit and sensors attached to the bottom, and the valve guard with the actuation button and status LEDs attached to the side. The other bracket slides apart in order to detach the sensing apparatus. (Figure 6.11) shows a close up picture of the mounted system and (Figure 6.12) shows the 3D-printed mounting bracket with the attached sensors from a top view.

B. Software design

Due to the I^2C-bus topography the sensing needed to be done sequentially. For the purpose of this system, the process of reading each sensor once is referred to as a

Figure 6.12: Top view of system housing

sensing cycle. The transmitters side of the system is the main component and its operation is described in this section.

When powered-up, the microcontroller first makes sure the sensors are connected by testing the I^2C bus. Then it configures the sensors for proximity measurements by writing into the **IR LED current; Proximity Measurement Signal Frequency**; and **Proximity Modulator Timing Adjustment** registers. Once the sensors are initialized the system shuts down the CPU and waits for the user to press the CAL button. When the calibration (CAL) button is pressed the system goes into Calibration mode. In this mode the system performs ten sensing cycles and averages the measurements. This process is done twice, one for the valves up to determine the lower boundary or min, and one for the valves down to determine the upper boundary or max. In this stage the slope of the linear model is calculated as (Equation 6.1):

$$m = \frac{1024}{max - min} \tag{6.1}$$

After calibration the CPU is shut down while it waits for the user to press the SENSE button. Due to the complexity of the optical path, there was a concern that there may not be a monotonic relationship at the output (one value count for each position); and in that case it would be impossible to linearize the readings from the sensor, but in practice there was in fact a monotonic relationship. Linearization was done through the equation $Y = m(xmin)$, where x is the value from the sensor and Y is the linearized output value. m is the slope and min is the lower boundary, as calculated in the Calibration stage. In Sensing mode the system wirelessly transmits the linearized, averaged measurements from three sensing cycles. It stays in this mode until the system is powered-down again.

PERFORMANCE

Calibration is done every time the system is powered-up and it compensates for any gain and offset error that the system may introduce. Main performance parameters are Spatial/Temporal resolution, System Robustness and Power Consumption.

Spatial / Temporal resolution

The trumpet valve has a range of 17mm (0.669"), and the optimal sensor position produces a full scale dynamic range of about 1000 counts or 10 bits, so the spatial resolution of 16.5m/value (micrometers per value). In practice it has been found that there is a variation in full scale dynamic range across valves of 1-bit, possibly due to a variation in the reactive properties of the bottom of pistons. This small loss in resolution has minimal effect on practical performance. Latency is determined by the time it takes the system to read data from the sensors and end it over to the receiver. When using the microcontroller's Low-Power mode, the latency is 3:6ms. The threshold for noticeable latency is 20mc for the average human ear, but could go as low as 15mc.

	Idle	**Sensing**
Type	LPM — no LPM	LPM — no LPM
AAA	170 — 94	77 — 78
LiPo	206 — 114	94 — 94

Table 6.2: EROSS battery specifications

This difference of 11:4ms gives room to do more sensing cycles and average them, thus perhaps improving the robustness of the system. Operating the microcontroller in low-power mode has no significant effect on the latency.

System Robustness

Under repeated trials involving attaching and detaching the current system, the greatest non-linearity transfer error, or variance, was of 15, or about 150 counts in 1000 without averaging. Even though the robustness can be greatly improved by averaging the measurements and reducing the system's profile, it performs consistently after attaching/detaching. Time and usage would probably decrease the system's performance as the bottom of the pistons become less reflective, due to dust and spit. This issue is largely mitigated by the calibration stage.

Power Consumption

An estimation of the current drawn by the system would help determine the battery's life expectancy. For the first prototype two rechargeable AAA batteries with 700mAh of capacity were used; for the second implementation, a LiPo battery with a capacity of 850mAh was used. To estimate the battery's life cycle twovperformance scenarios were studied:

1. Idle: After Initialization and Sensor Calibration, the system stays on but never goes into Sensing Operation mode.

2. Sensing: After Initialization and Sensor Calibration, the system goes into Sensing Operation mode and stays there until the battery is depleted.

These two scenarios cover the range of use of the system from minimum performance to full Sensing Operation. Two estimations were done, one with use of the microcontroller's Low-Power mode (LPM) and the other one without. Based on the results of (Table 6.2), it is clear that the long battery life will ensure that changing or recharging batteries is infrequent.

6.2 Geometric Rhythm Theory

ABSTRACTThis paper provides a description of a graphical geometric rhythm sequencer: geoSEEq (Gq)- a software rhythm sequencer that visualizes the geometric patterns of specific rhythms in a pulse defined cycle. In this case, the rhythms are visualized in a way that illuminates shapes formed by drawing lines between positive events on the cycle by clicking on the dots representing the pulses. Western notation, box notation, and geo-notation are viewed simultaneously. Gq shows that shapes present in commonly found rhythms around the world tend to reveal universally shared geometric patterns otherwise imperceivable without this method. These patterns often also reveal that common rhythms share symmetry as a trait in the shapes they create. This symmetry suggests why certain rhythms have groove and others dont.

6.2.1 Geometric Sequencer

GeoSEEq (Fig. 6.13) is a music sequencer that allows the user to view and program geometric shapes within the space of a sphere along the line of a pulse division to control musical events. Features include editing/viewing the music notation of the respective rhythms being sequenced and editing/viewing the box notation (MIDI style) of the respective rhythms.

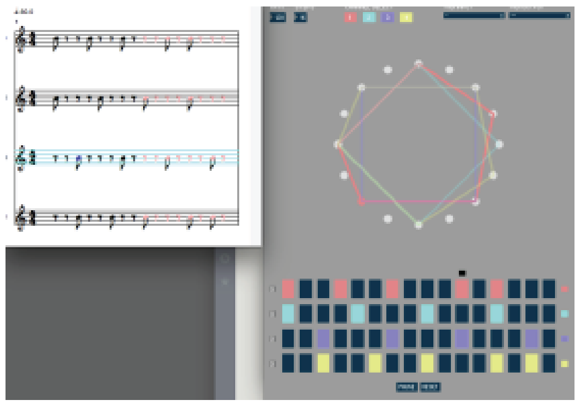


Figure 6.13: Geo_SEEq

Editing happens from all three viewing categories- geometric, notation, and box with all reflecting change in real-time. The amount of steps can also be edited in real-time. There are four potential voices that can occur concurrently- each can be selected, muted and reset individually. They are designated by color and viewable simultaneously. The program can receive MIDI in order to be controlled externally while still conveying the geometry of that particular rhythm. The sequencer can also send MIDI in order to control external hardware with the geometric patterns being visualized. Gq exposes that certain rhythms are often synonymous with one another, yet have different names, instruments, and purposes for their execution causing them to appear intrinsically unique from one-another through a cultural lens. These cultural differences can impose bias where the inherent symmetry would otherwise be imperceivable without this form of novel geometric graphical representation as notation. Gq reveals a visual profile that exposes rhythmic universality through inherent geometry not evident in traditional notation.

The Theme of Cycles

The theme of cycles, like in life, is ever-present in music. From the initial purity of a sine tone at a specific Hertz (Fig. 6.14), to harmonic cycles (Fig. 6.15), cycles in the form of ostinato (Fig. 6.16), to using cycles as a specific philosophical impetus for new compositional schemes [26], cycles are an ever-comprising element of music.

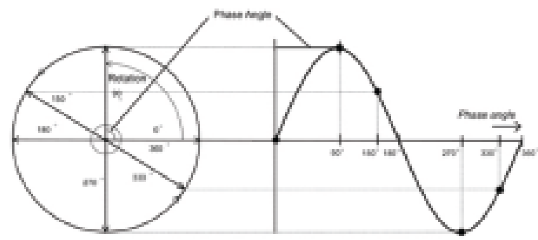


Figure 6.14: Sine wave cycle

Cycle can mean several things in music. Acoustically, it refers to one complete vibration (Fig. 6.14), the base unit of Hertz being one cycle per second [109]. Theoretically, an interval cycle is a collection of pitch classes created by a sequence of identical intervals. Individual pieces that aggregate into larger works are considered cycles, for example, the movements of a suite, symphony, sonata, or string quartet [138]. Harmonic cycles- repeated sequences of a harmonic progression- are at the root of many musical genres, such as the twelve-bar blues and most Pop songs. The Circle of Fifths shows how cycles define harmonic theory in Western music (Fig. 6.15). Rhythm cycles are universally pervasive- critical in classical Indian music (Talas), Indonesian Gamelan (Colotomy), the Ghanaian Gyil [145], and to the polyrhythmic West African drumming music comprised of many interlocking rhythmic loops comprising the whole, which is investigated in this section.

While the geometry of melodic content has been analyzed in depth [140], the work presented here focuses singularly on rhythm cycles (Fig. 6.17). West African music in general provides an excellent fabric for analysis when observing rhythm cycles because it is predominantly rich with polyrhythmic looping patterns.

Rhythm cycles, in traditional West African music, have been closely observed by Willi Anku. It is noteworthy to state the author took a course with Anku titled African Rhythm Theory at the University of Ghanas International Center for African Music and Dance in 2002.

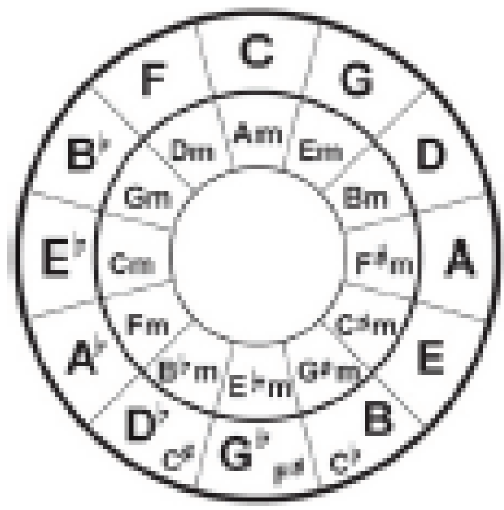


Figure 6.15: Circle of Fifths

Figure 6.16: simple bell pattern ostinato commonly found in African drumming

Anku states:

> Much of African music is circular. This circular concept of time ultimately defines a structural set. The set represents a structural module from which the entire performance is derived. The performance consists of a steady ostinato framework of multi-concentric rhythms on which various manipulations of the set are realized by a leader (e.g. lead drummer). It is in these complex structural manipulations (against a background of a steady ostinato referent) that Africa finds its finest rhythmic qualities [10].

Here, Ankus work reveals the nature behind interlocking rhythmic ostinati that might seem random when only perceived linearly (Fig. 6.18. Anku states that there is an intrinsic logic to the nature of the lead drummers solos where the various separate phrasings relationship to one-another would be imperceivable without the graphical representation. In this case, instead of viewing the lead drummers solo as expressive improvisations, a much deeper connection between the base ostinato (ensemble), seemingly extemporaneous embellishments (lead solo), individual parts to one-another, and overall form of the particular piece in general is revealed (Fig. 6.19).

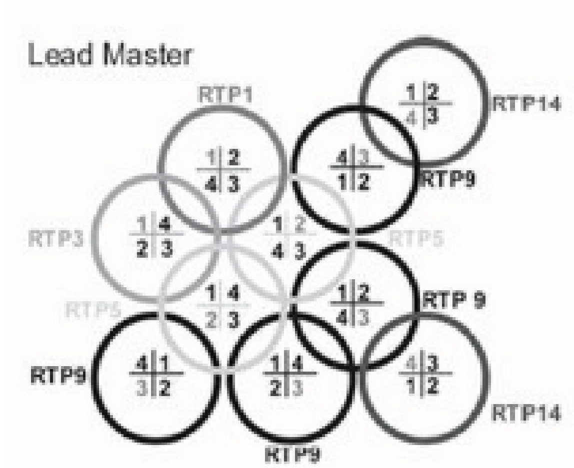


Figure 6.17: Ankus over-arching interlocked cyclic form

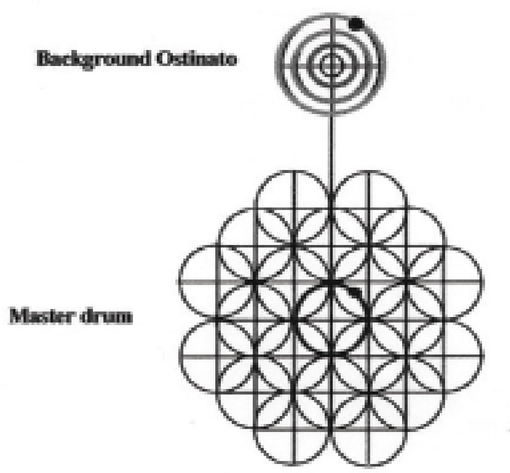


Figure 6.18: Lead part in relation to background ositinato

Each circle represents a particular single rhythmic phrase and intersects with the next one at various Relative Time Points (RTP). This illustrates how parts align with one-another on the timeline and where they share the downbeat with the background ostinato. This matrix of intricately intersecting concentric circles represents how the lead drummer navigates his solo and formulates the arrangement of lead phrases concurrently with the ensemble playing the primary loop in a musically/socially meaningful manner- as opposed to meaningless improvising/noodling.

Ankus theories are reflected in the work of Godfried Toussaint.

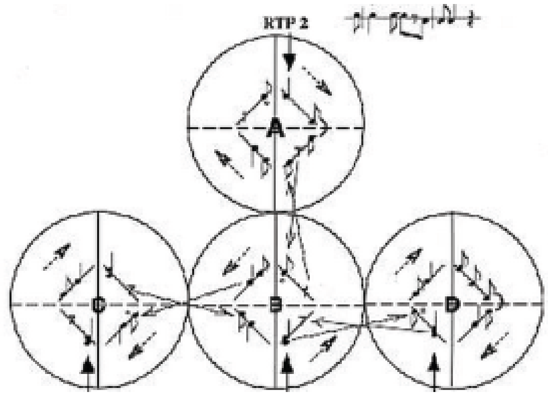


Figure 6.19: Getting from phrase to phrase

Toussaint states:

> "If we were to take a common rhythm native to Cuba- the Son Clave and view side-by-side with another commonly found rhythm from Ghana- Fume Fume, we see that by extrapolating the geometric line created by connecting positive events on the cycle we get the same shape, or rhythm. This similarity would otherwise be obscured against the background ostinato we observed in Ankus work, but using Toussaints method for geometric notation, we see that the phrases are actually nearly identical despite having a different total number of pulses, meter, relationship to the timeline (RTP) and different support patterns (Fig. 6.20). Despite the difference in cultural contexts and amount of pulses- both beats divide their own global cycle the same and are symmetrical in the same orientation."

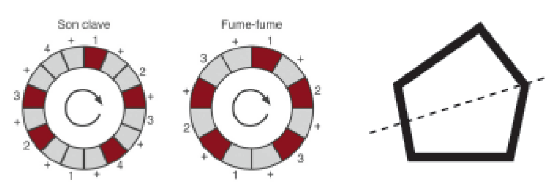


Figure 6.20: Son Clave (Cuba) compared to Fume-fume bell (Ghana).

This concept of Toussaints is used as the basis of design for the Gq prototype, real-time geometric sequencer. The main goals were to be able to program, edit, and visualize from all categories. (Fig. 6.21) provides a mock template for my design goals.

Design Goals:

1. Rotary Notation using Toussaints concepts;
2. Conventional Notation reflecting the geometry;
3. Box Notation (Standard MIDI format);
4. Global parameters- tempo, pulses, meter, and bars.

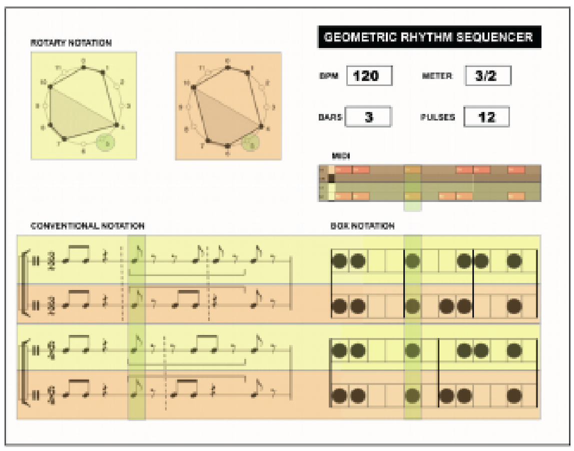


Figure 6.21: Prototype mock-up

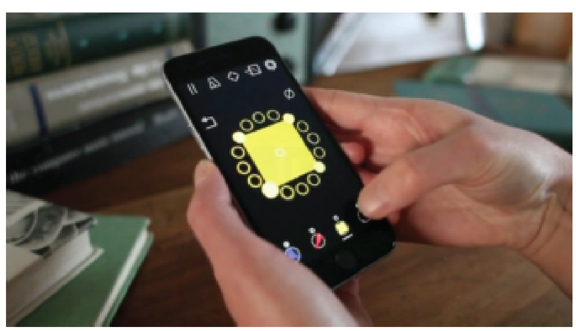


Figure 6.22: Rhythm Necklaces app (Meara OReilly and Sam Tarakajian)

Related Work

Toussaints concepts can be seen in several recent commercial Apps. Rhythmnecklaces[6] (Fig. 6.22) directly references Toussaint. Patterning[7], by Olympia Noise Company, reflects a very similar paradigm where, what would conventionally be the number of steps to program for a rhythm sequence are instead represented as equally divided zones on a circle (Fig. 6.23). Patterning has some really slick user interface features that look to be very dynamic and musical with many tweakable parameters that you would expect from a pro-studio grade piece of music sequencing software. A hardware version that is a distant kin to Tousaints theories is Jeff Mills modified Roland TR 909[8] - a classic drum machine where the artist wanted a tactile circular interface for real-time sequencing in performance. Buchla has released the 252e Buchla Polyphonic Rhythm Generator[9] (Fig. 6.25) and Zoom created and marketed the Arc[10] circular interactive groove box (Fig. 6.24).

Figure 6.23: Pattering: Olympia Noise Company

[6]http://rhythmnecklace.com/
[7]http://www.olympianoiseco.com/apps/patterning/
[8]http://daily.redbullmusicacademy.com/2015/04/how-jeff-mills-built-a-ufo-drum-machine
[9]https://buchla.com/product/252e-buchla-polyphonic-rythm-generator/
[10]https://www.zoom-na.com/products/production-recording/digital-instruments/zoom-arq-ar-96-aero-rhythmtrak

Figure 6.24: Zoom Arc

Figure 6.25: 252e: Buchla Polyphonic Rhythm Generator

6.2.2 System Description

1. Software Design

The software was prototyped in Max/MSP and Processing[11]. Max handles all control logic between the GUI (developed in Processing), audio events (managed in Processing), and notation visualization (built using MaxScore[12]). Max receives 4

[11] https://processing.org/

[12] http://www.computermusicnotation.com/

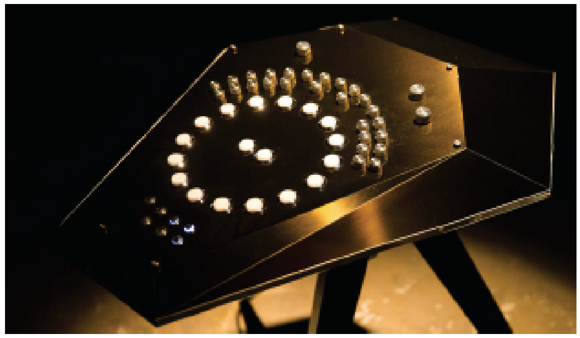


Figure 6.26: Modified TR-909

channel OSC data as live notation from processing for populating MaxScore and sending/receiving MIDI.

Processing

The Graphical User Interface was programmed in Processing because it is streamlined for OSC, MIDI, and Audio implementations, as well as visual processing. Percussion samples are called from Processing. The Processing patch receives and sends MIDI CCs for controlling an external sound module or for using an external sequencer. Each channel is represented in a unique color.

Max Notation

The sequencers musical notation is powered by and visualized using Max Score which requires a license. Since the time of this project, there is now a notation visualizer in Max 7 which might prove to be useful for future iterations of Gq. The patch works both ways- editing the Max score reflects in Gq and vice versa. This is useful for analyzing specific rhythms- either starting with the notation, as in the case of transcription or entering musical examples- or viewing the notation of a particular geometric shape. Box notation reflects all changes and can also originate a pattern (Fig. 6.2) with all changes reflecting in all categories (Box, Musical, Geometric).

Control Mappings

Control and Input/Output: *reset, toggles, channel clear, hot keys, midi clock*(Table 6.3)

ASCII	**MIDI**
SPACE = play/pause	MIDI Notes: Kick 36; Snare 38, Hi-Hat 42, Ride 51
X = reset	MIDI Output - [tx] to instrument/device
1-4 = mutes	MIDI Input - [rx] from external sequencer/controller
Arrow up and down the steps (32max)	*n/a*
Arrow up and down BPM/tempo box	*n/a*

Table 6.3: GeoSEEq: Control Logic

6.2.3 Euclidean Visualizer

Toussaint cites [129] that in Birth of Tragedy, Friedrich Nietzsche states:

> "Music is like geometric figures and numbers, which are the universal forms of all possible objects of experience. The geometric characteristics alone of a rhythm can provide new understandings of that aspect of a particular piece of music that one couldnt view easily in traditional notation. This novel facet of musical rhythmic analysis reveals new ways of perceiving intrinsic qualities of music that would otherwise not be revealed through established conventions alone. Rhythm is arguably the most fundamental aspect of music, as such, this new discovery of geometric presentation is a potentially profound contribution to musicologists, composers, and producers, alike."[95]

Polyrythm

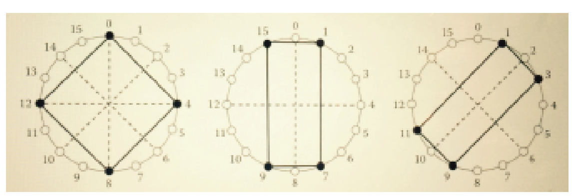


Figure 6.27: Common polyrhythms in 4/4 = pop music

Here (Fig. 6.27) one can observe three distinct rhythms that could easily be found in pop music. When stacked on top of one-another they create a polyrhythm that in we are viewing in geometric notation, thus revealing aspects about the polyrhythmic ostinato that is imperceivable otherwise. Symmetry is obviously inherent to this rhythmic loop, as in many cases with popular music.

Polymeter

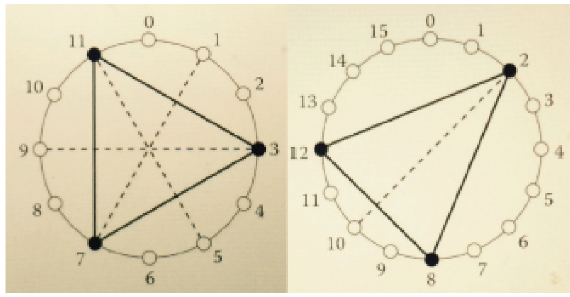


Figure 6.28: Example of polymeter

In (Fig. 6.28) two timelines with the same BPM (denoted by the circles being the same size representing a full cycle of a bar of music) are represented. In the left example three are 12 steps, in the right there are 16. Considering the full representation of a cycle is equal in each of the examples in terms of the time elapsed, one could observe how and when parts align and when they dont relative to their geometric characteristics. It is also possible to easily see when the meters realign at their respective mutual beginnings/RTP [10] (Fig. 6.19).

Odd Meter

Below (Fig. 6.29) is an ancient Greek rhythm in box notation with 5 groups of 5 pulses:

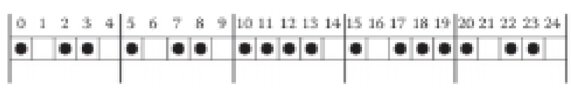


Figure 6.29: Greek rhythm - 5 groups of 5 pulses

In popular music we have examples of odd meter:

Above are two variations (Fig. 6.30) of a nine-pulse rhythm from Dave Brubecks Blue Rhono a la Turk. Below (Fig. 6.31) are three different 11-pulse rhythms from Dave Brubecks *Countdown* (left), Burt Bacharach and Hal Davids chorus of *I Say a Little Prayer* (center), and the meter of the Bulgarian folk dance *Krivo Horvo* (right). Symmetry is present in all of these rhythms revealing characteristics that make them appealing as odd meters that we would not otherwise perceive in conventional or box notation.

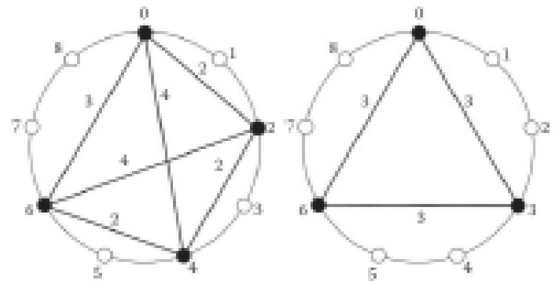


Figure 6.30: Two variations of Dave Brubecks Blue Rhondo a la Turk

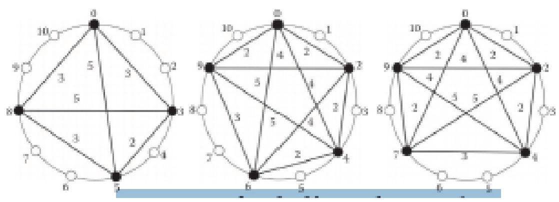


Figure 6.31: 3 vaariations of Brubecks 11-pulse Countdown

6.2.4 Musicality

Groove vs. no Groove- What makes a rhythm good?

When a rhythm is described as good by Toussaint, the word is intended to denote that it is effective as a timeline, as judged by cultural traditions and the test of time- these rhythms are found across cultural lines and thusly have different names where they are used. One aspect regarding Groove that cant be visualized with the current configuration of Gq are the rhythmic offsets where a humanized performance of a particular rhythm is slightly displaced- either a few milliseconds before or after the intended positive pulse event causing a type of rushed or behind-the-beat feel that some iconic percussionists incorporate into their signature feel, or personal groove language. Being able to view/edit these timing modulations during analysis of a particular performance could reveal new perceptual understandings about timing, feel, and interconnectedness of polyrhythm through the lens of a geometric global clock. The way Gq (and typical sequencers/visualizers) quantize the beat to the metric grid with precise timing to the BPM, respectively, is universal within the MIDI protocol. This prevents analysis of slight timing modulations (within the designated metric division) being visualized and handicaps the effectiveness of using Gq as an analysis tool. This feature is slated for future work. If the timeline could stretch in a similar fashion as elastic audio[13], one could compensate for metric offsets and fine-tune to adjust for accurate timing modulations of a particular performance from a particular player in a specific style being observed (Fig. 6.32).

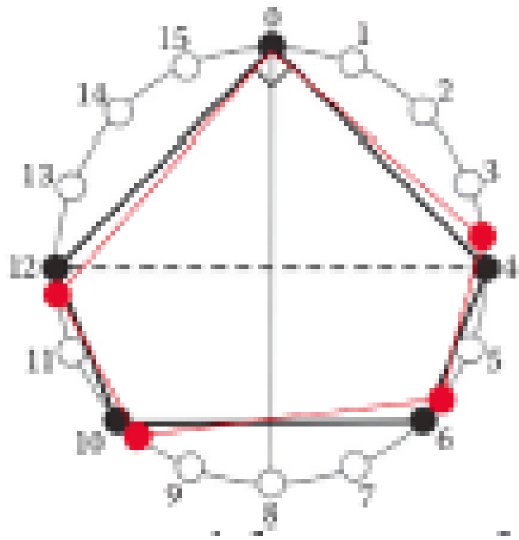


Figure 6.32: Timing modulation visualized in red

Symmetry

Symmetry suggests groove. There are many rhythms possible within the time-space of 16 equidistant rhythmic pulses. Of six commonly found rhythms around the world are observed having pure symmetry, but Rumba (Fig. 6.33).

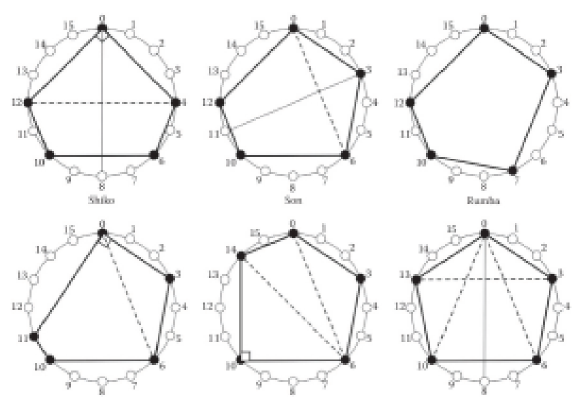


Figure 6.33: Common rhythms around the world

[13]https://www.ableton.com/en/manual/audio-clips-tempo-and-warping/

Toussaint presents a formula [129] that evaluates all possible rhythms producible within 16 steps. This formula yields the number 4,368. The figure below shows, in box notation, a dozen arbitrary members of this large family of rhythms (Fig. 6.34). He illustrates that not every rhythm in this family is considered good- meaning that somewhere on the planet it has been adopted into traditional repertoire as a timeline or a recognized, socially meaningful and distinct rhythm.

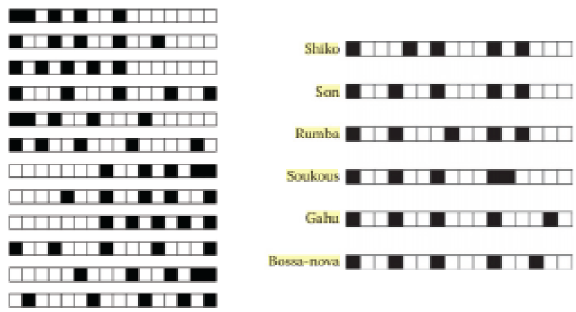


Figure 6.34: 12 arbitrary 16 pulse rhythms (l); previous 6 common rhythms (r)

It is clear that in simply glancing above at the figure on the right where the common set of global rhythms are written in box-notation, that one would not be able to denote the geometric symmetry inherent in each of the rhythms and that they all actually share a common shape event- though their accents and phrasing dont convey it clearly. In contrast, one of the main goals is to design a set of algorithms for generating good rhythms using simple mathematical principles observed above in the geometry to help psychologists and musicologists better understand why certain rhythms are embraced and become hypnotically seductive, while others seem dissonant and obtuse. This can been observed on a surface level simply from viewing the inherent shapes of the rhythms belonging to the possible overall group of 4,368, 16-step cycles. While some are recognizable as adopted timelines with their shapes being distinctly symmetrical, the obtuse rhythms of the group tend to produce no consistantly distinct, symmetrical shapes whatsoever and sound rhythmically dissonant from a social standpoint when looped on their own.

Experiments

1. Sounds: Default sounds are such and can be changed by changing the samples in the patch bank or controlling other external sound engines/robotic instruments using MIDI Out. They are called by Processing and stored in the according

project folder.

2. **eMarimba:** In this experiment I developed a pitch and amplitude envelope tracker in Max/MSP that would generate MIDI notes per acoustic bar hit fed into Gq. A marimba equipped with an electro-acoustic pickup system is used to provide the patch with a direct signal. I used this interface to program Gq in realtime in efforts to create a performance based format for the geometric visualizer that would display inherent geometry of the marimba performance rhythms- specific bars trigger specific percussion sounds in realtime.

3. **Robot:** This Max patch takes the MIDI notes produced by Gq and routes them to the Notomoton using a simple MIDI note scaler. This way, one can program/visualize robotic percussion sequences in geometric notation.

Part III

CONCLUSION

Chapter 7

Final Thoughts and Future Work

In conclusion the presented research has proven to be an inspiring platform for the author to pursue, with plans to build on and further the work in post-doctoral research, pedagogical, and artistic capacities. It is intended to expand the work into a framework that can be adopted by other players. A user study with feedback from other users would offer additional information on ways to improve the system. Intercultural collaborations using the framework present the possibility of new music being explored through cultural lenses that might afford insight into important social meanings of new interfaces for musical expression.

Interfaces

Wiikembe: The Wiikembe presents the possibility of being able to continually change filter parameters at will while being fully engaged in playing the acoustic aspect of the instrument in a traditional context without interruption. This allows for the possibility of the continual re-contextualization of the relationship of the acoustic signal vs. filter modulation output in real-time. Being able to interface the filters this way greatly diversifies the potential for sonic possibilities, improving the control aspect of interacting with filters in performance. As a result, this also increases the improvisational capacity of digital effects processing, evolving it into a musically dynamic practice, analogous and symbiotic with the traditional techniques of the acoustic instrument.

This work is meant to establish a rapid prototyping framework for idiomatic, intimate musical-DSP interaction so that artists can design custom instruments and computer music interfaces with minimal engineering overhead. The Wiikembe has

been used in concert with Trimpins robotic prepared pianos- (CanonX+4:33=100), at NIME 2012 with the Notomoton [65], and at Sound and Music Computing 2012 with EMMI[1]. In all of these concerts the author used the instrument in its native capacity, along with the hyperinstrument extensions to control parameters modulating filters used for creative signal processing, as well as control the robotic instruments parameters: velocity, sequence launching, and signal modulation. The instrument has proved stable, robust and compelling to use in concert: achieving the artists goals; and easy to reconfigure. The Wiikembe presents new avenues for sound design, sonic interaction, and musical-HCI in an idiomatic context specific to lamellophones, making technology that would have previously required custom engineering available to the conventional percussionist using common, low-cost, easily sourced contemporary components that are simple to work with. Further engineering experiments are intended, including the development of an open source, autonomous, and proprietary framework that eschews the laptop and Wiimote dependencies. This work is in development, results prove promising. Work with MEMS microphones and sensors are being explored to reduce the footprint of the embedded circuitry on the instrument.
El-Lamellophone: The El-la system has been tested and deployed in performance and put to use in rigorous contexts. It has also been used extensively on an album exclusively featuring the instrument in duo with the Range Guitar recorded at STEIM[2] in Amsterdam during an Artist Residency[3]. The instrument proves to be stable, robust, and remains inspiring to the artist. Future work includes designing a PCB and proper housing for the interface and microcontroller that can easily mount onto the instrument, while remaining easily removable. A custom chromatic Lamellophone design has been explored to ascertain whether or not a proprietary version with embedded interface and controller would be more intuitive. A user study is planned to gain valuable feedback about the instrument's augmentations. Solar cells for power could potentially make the instrument truly autonomous and ready for deployment in the field where collaboration with traditional instrumentalists could be established.

Physical Model

The physical and sonic characteristics of the Gyil have been analyzed and a physical model of the gourds, which produce its characteristic sound, has been developed. The

[1]http://expressivemachines.com/dev/wordpress/
[2]http://steim.org/
[3]https://southwoods.bandcamp.com/releases

performance of several different variants of the model using both comparative analysis and through listening tests with experts have been documented. The creation of a physical model of the gourds presents immediate applications for performers. The gourd model can be used in real time with any audio input as an effect. In particular, it can be used to process audio produced by other members of the idiophone family. Given the rarity of the Gyil outside of western Africa, this model presents performers with a means to access the characteristic Gyil sound without requiring the sourcing or construction of a Gyil. It also enables efficient traveling as packing the wooden bars is easy and the gourds, which are fragile and cumbersome, can be emulated digitally.

Because the physical model uses control parameters, which are clearly linked to the physical properties of the gourds on which they are modeled, users have an intuitive way to modify the sound of the model to suit their own purposes. The parameters and parameter ranges of the model have been specifically selected for use with mallet instruments (particularly idiophones), based on the pitch range and timbre produced by the Gyil, which makes them a uniquely useful tool for musicians and composers working with these instruments.

While this is a good model of the Gyil gourds, there are improvements, which could be implemented in future developments of the model. The primary difference remaining between the model and the physical instrument is the spectral characteristics of the distortion produced by the gourd. As was demonstrated in the comparative and qualitative listening tests, the Gyil model produces output with less high-frequency energy than the physical instrument. In future work a more detailed model of the physics of the spider-silk membranes could address this. While the complexity of the physics of the membranes makes them a difficult object to model, this does not mean that efforts in this area will be fruitless. Future experiments could include a nonlinear waveguide mesh. Furthermore, comments from the expert listening tests suggested that the buzz produced by the physical Gyil is much less predictable than the buzz produced by the model. In future work, stochastic methods could introduce random variation.

Surrogate Sensing

A direct sensing apparatus specifically designed for the African wooden xylophone called the Gyil has been described. An array of piezo-electric pickups mounts on the wooden bars enables the capture of detailed performance data. The raw audio data produced can be used to drive a physical model of the gourds. I also describe how the direct sensors can be used to obtain ground truth information for evaluation audio transcription approaches tailored to the particular instrument. There are many directions for future work. It is intended to study the variations in technique among different players in the context of performance analysis. The physical model can be improved by more detailed modeling of the gourd resonators as well as including coupling between neighboring gourds. Also I plan to create a full physical model in which the wooden bar excitation is simulated. Based on the existing algorithms, indirect acquisition is not sufficiently accurate to obtain performance data. I plan to investigate several variations of factorization methods. The surrogate sensor approach to obtaining ground truth will be essential in continuously improving the algorithms. Finally, it is intended to include the developed model system in live performances of electro-acoustic music.

Actuation

Musical Information Robotics: Techniques from Music Information Retrieval (MIR) can be adapted and used to solve practical problems in music robotics. More specifically audio classification can be used for automatic mapping, principal component analysis can be used for velocity/timbre calibration and dynamic time warping for gesture recognition. This system has not yet been tried in performance, yet it is intended to be improved and used in a live setting, many opportunities exist. In the future it is planned to extend this work utilizing more sensors including multiple microphones on both the robot and the performers. To obtain the maximum possible dynamic range I plan to have multiple actuators placed at different distances on the same drum so that the ones that are far are used for loud sounds and the ones that are near are used for soft sounds. The proposed calibration method will be used to drive seamlessly both actuators. I would also like to investigate how MIR techniques can be used to teach the robot to play and recognize rhythmic and melodic patterns.
Decay Sustain Release marimbA (DSRm): The DSRm is a platform for physically actuated/digitally controlled mallet instruments. It presents specific building blocks as techniques that can be expanded and serves as a foundation to explore various aspects of musical robotics [66]. However, room for improvement exists. Further work needs to be done to improve the digital audio signal fidelity. Optical sensors for audio pickups will be explored, as introduced by the Guitarbot [119] and the Mech-

bass [85], which eliminate unwanted system noise interference. This work has been initiated in EROSS [6], a system for control data. It is intended to expand it for audio acquisition. Other interfacing methods will also be explored. The robustness of the audio actuation system will be further developed as I attempt to implement a multi-channel version where many bars can be actuated independently with different types of signals on the various respective bars. I plan to introduce wireless capabilities in order to minimize the footprint and cabling requirements of the system.

In conclusion, the DSRm is an inexpensive, relatively easy to Implement/use, modular hyper-marimba framework prototype that presents vast potential for previously unattainable sounds and performance based interfacing solutions for digital pitched percussion electro-acoustic music. Using a small micro-computer that eliminates the need for speakers and the laptop, DSP capabilities that a conventional music computing platform would afford are retained while introducing an array of entirely new sound production facets to the instrument. The instrument itself (the bars) become the speakers and the instrument remains totally acoustic, able to be performed in its most simple, traditional capacity, as well as with the new digitally extended features. The author intends on developing a solar powered battery system so that the instrument can be played in a remote setting where no electricity is available. The whole system, including computer has minimal power requirements and could conceivably be battery powered. Since the bars would otherwise be the speakers, the DSRm presents a framework for a computer instrument that could be used in a folk capacity. A user study with results featuring the feedback of experienced musicians is planned to gather feedback about how to improve the instrument so that artists can begin realize repertoire for the system.

Tine Actuation: Some aspects of the current Lemellea actuation system prevents easy integration into EL-La without compromising the performer's ability to play the instrument normally. While the amplifier requires little-to-no heat-sinking, and the driver electronics are relatively compact, the magnets themselves are somewhat bulky, and require significant heat-sinking to avoid overheating. Ideally, the magnets would be positioned directly over the lamellophone tine, near the plucked end, but this configuration prevents normal playing technique. Nicolas Collins illustrates a similar idea (Figure 7.1) in [29] where the tines are also obscured from traditional playing.

The actuation of the magnets also tends to cause the pitch of the tines to lower slightly, for reasons that haven't yet been sufficiently explored by the author. For

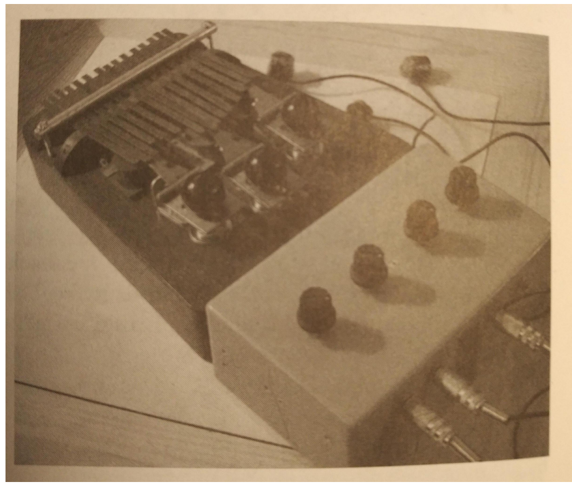


Figure 7.1: Thumb piano with electromagnetic drivers by Dan Wilson

these reasons, the current actuation system is a separate module from the rest of EL-La, able to be actuated by the Beaglebone, but not part of the same lamellophone system held by the performer. In this case one can employ the various sensors, such as the capacitive touch interface via the tines on the performers instrument to control the actuation of the tines on the electro-magnetic lamellophone, for instance. Further, the signal from the actuated instrument can be acquired in the same format, using a piezo, amplified, and processed autonomously via its own Beaglebone audio processing system. The two Beaglebones can be networked and synced via Ethernet or over a wireless connection using OSC.

Self-Tuning Auto-monochord Robotic Instrument (STARI): STARI presents a platform for physically actuated string instrument development. It offers specific building blocks as specific techniques that can be expanded and serves as a foundation to explore various aspects of musical robotics [66]. However, with this implementation, room for improvement exists. Further work needs to be done to improve audio signal fidelity. Optical sensors for audio pickups will be explored, as introduced by the Guitarbot [119] and the Mechbass [85], which eliminate unwanted system noise interference. This work has been initated in EROSS [69], and it is intended to expand it for audio acquisition. I intend to develop my own magnetic actuator system, along with solenoid beaters and dampeners similar to the Trimpins pianos and AMI. Other ideas include exploring a possible emulation of Trimpins pianos that had scrapers running laterally along the string, emulating a pick slide on the guitar. I intend to introduce multiple strings to introduce polyphony, as well as, researching the Autoharp to determine if it is possible to adopt its actuation techniques to achieve robotically actuated harmonic relationships. A simple, proprietary control interface was developed, allowing for wireless networking- this needs optimization to be reliable. STARI is intended for creative purposes, building on previous work so that it can be coupled with conventional computer music controllers, hyperinstruments, or used alone.

Abstracted Contributions

Reconfigurable Autonomous Novel Guitar Effects (RANGE): RANGE successfully presents a reconfigurable, autonomous, and novel DSP and sensing framework for the guitar. The RANGE system has been used in concert several times and in the studio, proving to be a novel, satisfying method of DSP interaction for the touring guitar player and computer musician. A user study contrasting the proposed approach with a traditional electric guitar effect setup is planned. It is also planned to investigate polyphonic transcription using a surrogate sensing approach [113]. An extended platform to support magnets for robotic guitar string actuation building on the previous work by Berdahl [42], McPherson [82], on electromagnetic actuation of stringed instruments is also intended.

Easily Reconfigurable Optical Sensing System (EROSS): The main goal of EROSS was to motivate minimally invasive and easy to remove digital sensing systems for musical instruments in general. Therefore, describing any explicit artistic mappings and performances with the system were refrained from, as it is hoped EROSS can be used by many artists in different and unique ways.

In the current configuration the added weight of EROSS (180g compared to the 880-1130g of a typical trumpet) still allows the trumpet to be held comfortably, but the high profile of the system, along with its low center of gravity affect the robustness of the EROSS by applying unwanted mechanical torque when tilting the trumpet. Replacing the AAA batteries with a slimmer LiPo battery, as well as designing a printed circuit board would reduce the systems profile, therefore reducing the torque and improving upon the robustness.

Future plans include providing versions of the system to other researchers experimenting with augmenting the trumpet as well as expert trumpet players that do not have a technical background. This way feedback about how intuitive the system is to use, how stable and robust it is, what are possible improvements, as well as more subjective aspects such as whether or not it is inspiring to use, whether it hinders performance, and whether it fits into a particular artists workflow can be ascertained. An easy to remove audio pickup in conjunction with real time pitch detection can be used to provide accurate pitch tracking information that can also be informed by the valve positions.

One interesting application of the proposed system planned to explore is real-time transcription and latency compensation. This involves the concept of what has been referred to as Negative Latency. This concept is associated with the time it takes for the valve to go down when is pressed. In order to play a note, a trumpeter generally has the valve(s) already fully pressed down (open valve positions are a special case) when they starts to blow into the mouthpiece. From previous tests, it was determined that a player is able to fully press a valve in 50ms. If the instant that he begins to blow is considered t = 0ms, because is when the sound is being produced, then at t = 50ms he would have been starting to press the valve(s). This negative instant could be used to give the system a head start to perform calculations and predictions in effect compensating for any system induced latency. The idea is to study how this Negative Latency could improve time accuracy in real-time transcription, as opposed to more discrete approaches of valve sensing [[31], [100]].

EROSS provides a low-cost and adaptable platform that artists can utilize without much background in engineering and electronics. This way they can realize their own specific vision while still taking advantage of a flexible framework for interactive physical computing. The use of minimally invasive and easy to remove augmentations to instruments has the potential to widen the adoption of hyperinstruments. EROSS demonstrates the potential of an easily removable, minimally invasive wireless sensing system for augmenting the trumpet with digital control capabilities.

Although in practice the dynamic range of the system is below the desired 10-bit range, it provides a fluid, dynamic and fairly linear response. The noise induced by the EROSS not being mechanically locked when mounted to the trumpet, plus the sensor's inherent noise, presented a decrease in the system's robustness. With a non-linearity measure of the system's transfer function in the order of 5percent, it still proved to be robust enough to give consistent output when detaching and attaching the system. In summary, the objectives of devising an easily removable system that provided continuous gesture control from the trumpet valves has been achieved and deployed in performance reliably and to great effect. The system was made possible by leveraging advances in sensors, micro-controllers, wireless transmission, and 3D printing. The system is stable and works as intended under the design constraints posed. It does not affect acoustic trumpet playing and is tightly integrated with the gestures used for playing. It is hoped that because of the open design of the system it will be adopted, used, and improved by others. Iterations are beginning to emerge as evidenced by MIGSI (Minimally Invasive Gesture Sensing Interface)[116].

Geometric Sequncer (GeoSEEq): GeoSEEq works well as a prototype for its intended use. Future work will include eventually making an autonomous, dedicated application with all bugs corrected. Notation could be improved by using the new notation functionality in Max, currently meter is displayed incorrectly. Being able to visualize groove offsets (elastic MIDI), multiple rhythms in differing meters, and dynamics related control parameters are all future features to be implemented. An open-source dedicated, idiomatic hardware controller using a touch screen for visualization and interaction will also be explored and applied towards live performance using a matrix spatial speaker system reflecting the GUI [9]. Toussaint discusses isomorphism regarding applying the rhythmic analysis to a paralleled interpretation concerning melody. The Notomoton/Emarimba relationship could be explored in this capacity and is a compelling avenue for future work.

Impact on the Field

This work has had a proven impact in the NIME community. The work has been cited in leading conference articles, journals, Masters theses, Doctoral dissertations, and book chapters on electronic music. There are too many projects to list that cite this work (107 citations to date). Relevant notable projects citing this work branches into other instrumental paradigms, research that analyzes orchestra gestures, primary education technology resources and more. For a full list of publications with citations and accompanying med visit the author's website[4]

Final Thoughts

In conclusion, the system described herein presents important advancements specific to the art of pitched percussion hyper-instrument performance that can be abstracted to other instruments/contexts. The author intends to formalize the framework into a comprehensive, dedicated instrument with redesigned frame, embedded electronics, along with the addition of an acoustic damper pedal similar to the vibraphone.

[4] www.shawntrail.co

The author would like to further the research into areas including, but not limited to: 4D spatial audio, AR/VR[5], and cross-cultural telematic collaborations. Additionally, the author seeks to commission new works for the platform through collaborations with composers and performers where new etudes, methods for study, a gesture library, and system manual can be developed towards formalizing a framework for use by others.

[5]http://store.steampowered.com/app/778430/OSCKit/

Appendix A

Author publications related to this thesis

In reverse chronological order

1. **GeoSEEq: a new musical sequencer that conveys the geometry of rhythmic cycles**. S Trail. 2015 (unpublished)

   Abstract: *This paper outlines the motivation, design and development of the DSRmarimba (DSRm.). This work presents a portable, autonomous mechanically actuated, digitally controlled musical mallet instrument intended for creative and pedagogical use. Detailed tests performed to optimize technical aspects of the DSRmarimba are described to highlight usability and performance specifications for artists and educators. The DSRm. (Figure 1) is intended to be open-source so that the results are reproducible and expandable using common components with minimal financial constraints and little engineering overhead. Because the field of musical robotics is so new, standardized systems need to be designed from existing paradigms. Such paradigms are typically singular in nature, solely reflecting the idiosyncrasies of the artist and often difficult to reproduce. The DSRm. is an attempt to standardize certain existing and novel robotic pitched-percussion techniques in order to establish a formal system for experimentation.*

2. **El-Lamellophone- A Low-cost, DIY, Open Framework for Acoustic Lemellophone Based Hyperinstruments**. S Trail, D MacConnell, L Jenk-

ins, J Snyder, G Tzanetakis, PF Driessen. In Proc. Intl. Conf. On New Interfaces for Musical Expression (NIME), 2014.

Abstract: *The El-Lamellophone (El-La) is a Lamellophone hyper-instrument incorporating electronic sensors and integrated DSP. An embedded Linux micro-computer supplants the laptop. A piezo-electric pickup is mounted to the underside of the body of the instrument for direct audio acquisition providing a robust signal with little interference. The signal is used for electric sound-reinforcement, creative signal processing and audio analysis developed in Puredata (Pd). This signal inputs and outputs the micro-computer via stereo 1/8th inch phono jacks. Sensors provide gesture recognition affording the performer a broader, more dynamic range of musical human-computer interaction (MHCI) over specific DSP functions. The instruments metal tines (conventionally used for plucking- traditional lamellophone sound production method) tines have been adapted to include capacitive touch in order to control a synthesizer. Initial investigations have been made into digitally-controlled electromagnetic actuation of the acoustic tines, aiming to allow performer control and sensation via both traditional Lamellophone techniques, as well as extended playing techniques that incorporate shared human/computer control of the resulting sound. The goal is to achieve this without compromising the traditional sound production methods of the acoustic instrument while leveraging inherent performance gestures with embedded continuous controller values essential to MHCI. The result is an intuitive, performer designed, hybrid electro-acoustic instrument, idiomatic computer interface, and robotic acoustic instrument in one framework.*

3. **DSRmarimba: low-cost, open source actuated acoustic marimba framework**. S Trail, L Jenkins, P Driessen. Proceedings of the International Computer Music Conference (ICMC), 2014

Abstract: *This paper outlines the motivation, design and development of the DSRmarimba (DSRm.). This work presents a portable, autonomous mechanically actuated, digitally controlled musical mallet instrument intended for creative and pedagogical use. Detailed tests* *performed to optimize technical aspects of the DSRmarimba are described to highlight usability and performance specifications for artists and educators. The DSRm. (Figure 1) is intended to be open-source so that the results are reproducible and expandable using common components with minimal financial constraints and little engineering overhead. Because the field of musical robotics is so new, standardized systems need to be designed from existing paradigms. Such paradigms are typically singular in nature, solely reflecting the idiosyncrasies of the artist and often difficult to reproduce. The DSRm. is an attempt to standardize certain existing and novel robotic pitched-percussion techniques in order to establish a formal system for experimentation.*

4. **STARI: A self tuning auto-monochord robotic instrument**. S Trail, L Jenkins, D MacConnell, G Tzanetakis, M Cheng, P Driessen. IEEE Intl. Conf. on Pacific Communications, Computers and Signal Processing (PacRim), 2013

Abstract: *This paper outlines the motivation, design and development of a self-tuning, robotic monochord. This work presents a portable, autonomous musical robotic string instrument intended for creative and pedagogical use. Detailed tests performed to optimize technical aspects of STARI are described to highlight usability and performance specifications for artists and educators. STARI is intended to be open-source so that the results are reproducible and expandable using common components with minimal financial constraints. Because the field of musical robotics is so new, standardized systems need to be designed from existing paradigms. Such paradigms are typically singular in nature, solely reflecting the idiosyncrasies of the artist and often difficult to reproduce. STARI is an attempt to standardize certain existing actuated string techniques in order to establish a formal system for experimentation and pedagogy.*

5. **The Wiikembeperformer designed lamellophone hyperinstrument for idiomatic musical-DSP interaction**. S Trail, G Tzanetakis. PacRim, 2013.

Abstract: *The Wiikembe is an augmented Likembe from Zaire, believed to be 100+ years old. A Wiimote affords 3D gesture sensing for*

musical-HCI. An Arduino interface offers explicit control over DSP functions. Puredata (Pd) scales, converts, and routes control data into Ableton Live. A contact mic is used to acquire a direct audio signal from the Likembe. The audio inputs into a conventional computer audio interface and routed into Live which handles event sequencing, DSP, and audio bussing. The result is a compact and intuitive, robust lamellophone hyperinstrument. The Wiikembe extends the sonic possibilities of the acoustic Likembe without compromising traditional sound production methods or performance techniques. We chose specific sensors and their placement based on constraints regarding the instrument s construction, playing techniques, the authors idiosyncratic compositional approach and sound design requirements. The Wiikembe leverages and combines inherent performance gestures with analogous embedded gestural sensing to achieve unprecedented intimate musicalDSP interaction. Specific gesture recognition techniques and mapping strategies have been standardized using easily sourced and implementable, low-cost components. This work is in efforts to establish an implementable pitchedpercussion hyperinstrument framework for experimentation and pedagogy with minimal engineering requirements.

6. **An Easily Removable, wireless Optical Sensing System (EROSS) for the Trumpet**. L Jenkins, S Trail, G Tzanetakis, PF Driessen, W Page. Proceedings of the Int. Conf. On New Interfaces for Musical Expression (NIME), 2013.

   Abstract: *This paper presents a minimally-invasive, wireless optical sensor system for use with any conventional piston valve acoustic trumpet. It is designed to be easy to install and remove by any trumpeter. Our goal is to offer the extended control afforded by hyperinstruments without the hard to reverse or irreversible invasive modifications that are typically used for adding digital sensing capabilities. We utilize optical sensors to track the continuous position displacement values of the three trumpet valves. These values are transmitted wirelessly and can be used by an external controller. The hardware has been designed to be reconfigurable by having the housing 3D printed*

so that the dimensions can be adjusted for any particular trumpet model. The result is a low cost, low power, easily replicable sensor solution that offers any trumpeter the ability to augment their own existing trumpet without compromising the instruments structure or playing technique. The extended digital control afforded by our system is interweaved with the natural playing gestures of an acoustic trumpet. We believe that this seamless integration is critical for enabling effective and musical human computer interaction.

7. **Reconfigurable autonomous novel guitar effects (range)**. D MacConnell, S Trail, G Tzanetakis, P Driessen, W Page. Proceedings of the Int. Conf. On Sound and Music Computing (SMC), 2013.

Abstract: *The RANGE guitar is a minimally-invasive hyperinstrument incorporating electronic sensors and integrated digital signal processing (DSP). It introduces an open framework for autonomous music computing eschewing the use of the laptop on stage. The framework uses an embedded Linux microcomputer to provide sensor acquisition, analog-to-digital conversion (ADC) for audio input, DSP, and digital-to-analog conversion (DAC) for audio output. The DSP environment is built in Puredata (Pd). We chose Pd because it is free, widely supported, flexible, and robust. The sensors we selected can be mounted in a variety of ways without compromising traditional playing technique. Integration with a conventional guitar leverages established techniques and preserves the natural gestures of each players idiosyncratic performing style. The result is an easy to replicate, reconfigurable, idiomatic sensing and signal processing system for the electric guitar requiring little modification of the original instrument.*

8. **Physical modeling and hybrid synthesis for the Gyil African xylophone**. D Godlovitch, S Trail, TF Tavares, G Tzanetakis. Proceedings of the Int. Conf. On Sound and Music Computing (SMC), 2012.

Abstract: *We propose a physical model for the Gyil, an African pentatonic idiophone with wooden bars that have similar sonic characteristics to the western marimba. The primary focus is modeling* *the gourds that are suspended beneath each bar and have a similar role to the tuned tubes below the bars in western mallet instruments. The prominent effect of these resonators is the added buzz that results when the bar is struck. This type of intentional sympathetic distortion is inherent to African instrument design as it helps unamplified instruments be heard above crowds of people dancing and singing. The Gyils distortion is created by drilling holes on the sides of each gourd and covering them with membranes traditionally made from the silk of spider egg casings stretched across the opening. By analyzing the sonic characteristics of this distortion we have found that the physical mechanisms that create it are highly nonlinear and we have attempted to model them computationally. In addition to the fully synthetic model we consider a hybrid version where the acoustic sound captured by contact microphones on the wooden bars is fed into a virtual model of the gourd resonators. This hybrid approach simplifies significantly the logistic of traveling with the instrument as the gourds are bulky, fragile and hard to pack. We propose several variants of the model, and discuss the feedback we received from expert musicians.*

9. **Direct and surrogate sensing for the Gyil African xylophone**. S Trail, TF Tavares, D Godlovitch, G Tzanetakis. Proceedings of the Int. Conf. On New Interfaces for Musical Expression (NIME), 2012.

Abstract: *The Gyil is a pentatonic African wooden xylophone with 14-15 keys. The work described in this paper has been motivated by three applications: computer analysis of Gyil performance, live improvised electro-acoustic music incorporating the Gyil, and hybrid sampling and physical modeling. In all three of these cases, detailed information about what is played on the Gyil needs to be digitally captured in real-time. We describe a direct sensing apparatus that can be used to achieve this. It is based on contact microphones and is informed by the specific characteristics of the Gyil. An alternative approach based on indirect acquisition is to apply polyphonic transcription on the signal acquired by a microphone without requiring the instrument to be modified. The direct sensing apparatus we have developed can be used to acquire ground truth for evaluating different approaches to poly-*

phonic transcription and help create a surrogate sensor. Some initial results comparing different strategies to polyphonic transcription are presented.

10. **Non-invasive sensing and gesture control for pitched percussion hyper-instruments using the Kinect**. S Trail, M Dean, G Odowichuk, TF Tavares, PF Driessen, WA Schloss, G Tzanetakis. Proc. of the Int. Conf. On New Interfaces for Musical Expression (NIME), 2012.

    Abstract: *Hyper-instruments extend traditional acoustic instruments with sensing technologies that capture digitally subtle and sophisticated aspects of human performance. They leverage the long training and skills of performers while simultaneously providing rich possibilities for digital control. Many existing hyper-instruments suffer from being one of a kind instruments that require invasive modifications to the underlying acoustic instrument. In this paper we focus on the pitched percussion family and describe a non-invasive sensing approach for extending them to hyper-instruments. Our primary concern is to retain the technical integrity of the acoustic instrument and sound production methods while being able to intuitively interface the computer. This is accomplished by utilizing the Kinect sensor to track the position of the mallets without any modification to the instrument which enables easy and cheap replication of the proposed hyper-instrument extensions. In addition we describe two approaches to higher-level gesture control that remove the need for additional control devices such as foot pedals and fader boxes that are frequently used in electro-acoustic performance. This gesture control integrates more organically with the natural flow of playing the instrument providing user selectable control over filter parameters, synthesis, sampling, sequencing, and improvisation using a commercially available low-cost sensing apparatus.*

11. **Sensor fusion: Towards a fully expressive 3d music control interface**. G Odowichuk, S Trail, P Driessen, W Nie, W Page. IEEE International Conference on Pacific Communications, Computers and Signal Processing (PacRim), 2011.

Abstract: *This paper describes a study into the realisation of a new method for capturing 3D sound control data. It expands on our established practice with the radiodrum 3D input device by incorporating a computer vision platform that we have developed using the Xbox Kinect motion sensing input device. It also extends our previous work in systemizing a low-cost, open-source 3D gesture sensor system, as a novel method for musically oriented human computer interaction. We discuss in detail the development process and the system performance in different scenarios and outline the future directions of this development.*

12. **Music Information Robotics: Coping Strategies for Musically Challenged Robots**. SR Ness, S Trail, PF Driessen, WA Schloss, G Tzanetakis. International Society for Music Information Retrieval, 2011.

Abstract: *In the past few years there has been a growing interest in music robotics. Robotic instruments that generate sound acoustically using actuators have been increasingly developed and used in performances and compositions over the past 10 years. Although such devices can be very sophisticated mechanically, in most cases they are passive devices that directly respond to control messages from a computer. In the few cases where more sophisticated control and feedback is employed it is in the form of simple mappings with little musical understanding. Several techniques for extracting musical information have been proposed in the field of music information retrieval. In most cases the focus has been the batch processing of large audio collections rather than real time performance understanding. In this paper we describe how such techniques can be adapted to deal with some of the practical problems we have experienced in our own work with music robotics. Of particular importance is the idea of self-awareness or proprioception in which the robot(s) adapt their behavior based on understanding the connection between their actions and sound generation through listening. More specifically we describe techniques for solving the following problems: 1) controller mapping 2) velocity calibration, and 3) gesture recognition.*

Appendix B

Deployment

An important aspect of this research is from the perspective of the end user, which is also the design engineer. The author has developed techniques, repertoire, and performance contexts for the platform. This includes multi-media contexts for these new instruments- improvisatory scenarios; notated music and graphical scores for new extended digital techniques; and abstractions into other formats. This work has been deployed in the form of concerts, installations, videos, and presentations along with several compilations of original recordings/album appearances. A full-length, professionally produced album accompanies this dissertation in efforts to encapsulate this works potential. Deployment, discography appearances, and media assets can be found on the author's website[1].

1. CanonX+4:33=100 12 -Wiikembe, Range (MacConnell), robotic pianos (Trimpin)
2. TEDx Victoria 12 -Wiikembe, Notomoton
3. NIME 12 -Wiikembe, Fantom Faders, Notomoton
4. SMC 12 -Wiikembe, EMMI robots
5. Music Maker Festival Vancouver 12 -Wiikembe, Notomoton
6. STEIM 13 (Southwoods Vol. 1 - Gary Cassettes) -El-la, Range (MacConnell)

[1] http://www.shawntrail.co

Appendix C

Potential Patent Claims

This section outlines the innovations of this study.

What is claimed is:

1. A digitally interactive pitched percussion system comprising:

- a plurality of augmented pitched idiophones;
- electro-piezos proximate to idiophone;
- a plurality of gesture sensing interfaces;
- gesture and signal processing software;
- a physical model;
- a custom computing platform;
  - 2. The system of claim 1, wherein each of said plurality of components are selected from the group of gesture sensing interfaces consists of:

    foot switches, computer vision based 3D skeletal tracking system, 9DOF sensor position tracking system, membrane sensor linear position tracking array, capacitive touch and proximity sensing array, pressure sensor array and optical position tracking sensor array.
  - 3. The system of claim 2, wherein said generates a plurality of digital control information.
  - 4. The system of claim 3, wherein each of said plurality of controller information selected from the group consists of:

assignable momentary and toggle footswitch data; multinode skeletal tracking including but not limited to mallet differentiation in 3 assignable axis, assignable pitch, yaw, roll and acceleration of position in space; assignable physical linear position tracking; direct and discrete touch; vertical proximity and touch pressure.

- 5. The system of claim 4, wherein all of said plurality of gestures derived from the interface are reassignable
- 6. The system of claim 5, wherein said plurality of assignments is converted to various mapping logic.
- 7. The system of claim 6, wherein one of said plurality of mapping logic is routed to said gesture and signal processing software.

8. The system of claim 1, wherein each of said plurality of components are selected from the group of gesture and signal processing software comprising:

- gesture recognition;
- gesture prediction;
- pattern differentiation;
- dynamics calibration;
- proprioception.

  - 9. The system of claim 3, a gesture recognition component that analyzes said plurality of 3D gestures;
  - 10. The system of Claim 4, a prediction component that anticipates the plurality of control data generated from said gestures;
  - 11. The system of Claim 7, a mapping logic automation component;
  - 12. The system of Claim 1, wherein said mapping logic is applied to actuation control.

13. The system of claim 1, wherein a physical model is described comprising:

- a physical model that replaces material resonators
- resonators of xylophone

- resonators of a traditional Ghanaian xylophone
- surrogate sensor system is used for dynamic control of model
  - 14.said model of Claim 13 reacts to user input.
  - 15. said model Claim 13 employs piezo system of claim 1.
  - 16. said piezo system of Claim 1 is coupled with microphone input.
  - 17. said system of Claim 17 com prises said surrogate system of Claim 13.
  - 17. algorithm analyzes audio input data
  - 18. algorithm analysis of Claim 17 is correlated to training library.
  - 19. said library of Claim 18 is converted to control data for said model system.

20. The system of claim 1, wherein a computing platform is described comprising:

- a single board computer
- assignable GPI/O
- open architecture
- low latency audio I/O
  - 21. said single board computer of Claim 19 has GPI/O for sensor acquisition devices described in Claim 2.
  - 22. said system of Claim 20 has open architecture running Linux OS.
  - 23. said system of Claim 20 performs low latency audio I/O supporting software system of Claim 1.

Bibliography

[1] et al. A. Hollinger, J. Thibodeau. *An embedded hardware platform for fungible interfaces*. Proc. the Int. Computer Music Conference (ICMC), pages 5659, 2010.

[2] P. Davidson R. Wilson A. Kapur, A. Lazier and P. Cook. *The electronic sitar controller*. NIME, 2004.

[3] P.R. Cook P. Driessen A. Kapur, P. Davidson and W.A. Schloss. *Digitizing North Indian Performance*. In Proc. of the Int. Computer Music Conf. (ICMC), 2004.

[4] George Tzanetakis Peter Driessen Andrew Schloss Adam R. Tindale, Ajay Kapur. *A comparison of sensor strategies for capturing percussive gestures*. Proceedings of the Conference on New Interfaces for Musical Expression (NIME), Pages 200- 203, 2005.

[5] George Tzanetakis Adam Tindale, Ajay Kapur. *Training surrogate sensors in musical gesture acquisition systems*. Multimedia, IEEE Transactions on, vol. 13, no. 1, pp. 50 59, Feb. 2011.

[6] J. M. Adrien. *The missing link: Modal synthesis*. Representations Of Musical Signals, Cambridge, 1991.

[7] R.M. Aimi. *Hybrid percussion: extending physical instruments using sampled acoustics*. Massachusetts Institute of Technology, 2007.

[8] A. Akl and S. Valaee. *Accelerometer-based gesture recognition via dynamic-time warping, affinity propagation, and compressive sensing*. ICASSP, 2010.

[9] David Wessel Ali Momeni. *Characterizing and Controlling Musical Material Intuitively with Geometric Models*. NIME, 2003.

[10] TW. Anku. *Circles and Time: A Theory of Structural Organization of Rhythm in African Music.* The Online Journal of the Society for Music Theory, Vol. 6, Number 1, 2000.

[11] W.A. Schloss A.R. Tindale, A. Kapur and G. Tzanetakis. *Indirect acquisition of percussion gestures using timbre recognition.* In Proc. Conf. On Interdisciplinary Musicology (CIM), 2005.

[12] W. A. Schloss B. Nevile, P. Driessen. *Radio drum gesture detection system using only sticks, antenna and computer with audio interface.* Proceedings of the International Computer Music Conference (ICMC), 2006.

[13] S. Barrett and J. Kridner. *Bad to the Bone Crafting Electronic Systems with BeagleBone.* Morgan and Claypool, 2013.

[14] Francis Bebey. *African Music: A People's Art.* Lawrence Hill Books, 1969.

[15] Judith Becker. *Earth, Fire, akti, and the Javanese Gamelan.* Ethnomusicology Vol. 32, No. 3 (Autumn, 1988), pp. 385-391, 1998.

[16] S. Bilbao. *EPercussion synthesis based on models of nonlinear shell vibration.* Audio, Speech, and Language Processing, IEEE Transactions on , vol. 18, no. 4, pp. 872880, 2010.

[17] James Blades. *Percussion Instruments and Their History, pg. 409.* The Bold Strummer, Ltd., 2005.

[18] Q. David Bowers. *Encyclopedia of automatic musical instruments: Cylinder music boxes, disc music boxes, piano players and player pianos... Incl. a dictionary of automatic musical instrument terms.* Vestal Press, 1988.

[19] C. Britt and et al. *The emvibe: An electromagnetically actuated vibraphone.* Int. Conf. on New Interfaces for Musical Expression (NIME), 2012.

[20] J. Brown. *Buzz and rumble: Global pop music and utopian impulse.* Social Text, 2010.

[21] M. Burtner. *The metasaxophone: concept, implementation, and mapping strategies for a new computer music instrument.* Organised Sound, 7(2):201213, 2002.

[22] M. Burtner. *A theory of modulated objects for new shamanic controller design.* Proceedings of the 2004 conference on New interfaces for musical expression, 2004.

[23] David Byrne. *How Music Works.* McSweeney's Books, 2012.

[24] E. Vincent C. Fevotte and R. Gribonval. *Performance measurement in blind audio source separation.* IEEE Trans. Audio, Speech and Language Processing, 2006.

[25] E. Vincent C. Fevotte and R. Gribonval. *Towards the evaluation of automatic transcription of music.* Anais do VI Congresso de Engeharia de Audio, 2008.

[26] R. Carl. *Terry Rileys In C (Studies in Musical Genesis, Struc- ture, and Interpretation).* Oxford University Press; 1 edition, 2009.

[27] Samuel Charters. *The Roots of the Blues: An African Search.* M. Boyars, 1981.

[28] John Collins. *Highlife Time- Ch. 51 (A) The Musical Ghost in the Machine.* Anemsesem Publications Limited, 1994.

[29] Nicolas Collins. *Handmade Electronic Music.* Routledge, 2006.

[30] P. R. Cook. *Physically informed sonic modeling (phism): Synthesis of percussive sounds.* Computer Music Journal , vol. 21, no. 3, 1997.

[31] Perry Cook. *Principles for Designing Computer Music Controllers.* CHI, 2001.

[32] T. Craig and B. Factor. *The trumpet midi controller.* Cornell, 2008.

[33] T. Tavares D. Godlovitch, S. Trail and G. Tzanetakis. *Physical modeling and hybrid synthesis for the gyil African xylophone.* In Proc. Int. Conf. Sound and Music Computing (SMC), 2012.

[34] S. Dahl and A. Friberg. *Expressiveness of musician's body movements in performances on marimba.* Gesture-Based Communication in Human-Computer Interaction, pages 361-362. Springer Berlin/Heidelberg, 2004.

[35] S. Dahl and A. Friberg. *Expressiveness of musicians body movements in performances on marimba.* Gesture-Based Communication in Human-Computer Interaction, vol. 2915, pp. 361362. Springer Berlin/Heidelberg, 2004.

[36] Matthew Wright David Wessel. *Problems and Prospects for Intimate Musical Control of Computers.* Proceedings of the CHI'01 Workshop on New Interfaces for Musical Expression (NIME-01), Seattle, USA, 2008.

[37] M. Davis. *Live-evil.* Music Album from Columbia Records, 1971.

[38] M. Demoucron and R. Causs. *Sound synthesis of bowed string instruments using a gesture based control of a physical model.* in Proc. of the International Symposium on Musical Acoustics, 2000.

[39] D.Overholt. *The overtone violin.* In Proc. New Interfaces for Musical Expression (NIME), 2005.

[40] C. Drioli and D. Rocchesso. *Learning pseudophysical models for sound synthesis and transformation.* in Proc. of IEEE Int. Conf. on Systems, Man, and Cybernetics pp. 10851090, 1998.

[41] D.Wessel and M. Wright. *Problems and prospects for intimate musical control of computers.* Computer Music Journal , vol. 26, no. 3, pp. 1122, 2002.

[42] G. Niemeyer E. Berdahl and J. O. S. III. *Active control of a vibrating string.* Acoustical Society of America (ASA) and European Acoustics Association (EAA), 2008.

[43] Elizabeth England. *Using Three-Dimensional Modeling to Preserve Cultural Heritage.* Library of Congress, 2017.

[44] C. Palacio-Quintin et al. *The hyper-flute.* NIME, 2003.

[45] N. H. Fletcher. *The nonlinear physics of musical instruments.* Reports on Progress in Physics, vol. 62, no. 5, p. 723, 1999.

[46] D. Font-Navarrete. *File under import Musical distortion, exoticism, and authenticity in congotronics.* Ethnomusicology Review, 2011.

[47] A. Freed. *Novel and forgotten current-steering techniques for resistive multitouch, duotouch, and polytouch position sensing with pressure.* Proc. Int. Conf. New Interfaces for Musical Expression (NIME), 2009.

[48] A. Freed. *Kalimba controller.* research blog, unknown.

[49] P. Driessen W. Nie G. Odowichuk, S. Trail and W. Page. *Sensor fusion: Towards fully expressive 3d music control interface.* In IEEE Pacific Rim Conf. PACRIM), 2011.

[50] W.A. Schloss G. Tzanetakis, A. Kapur and M. Wright. *Computational ethnomusicology.* Journal of interdisciplinary music studies, 2007.

[51] S. Gelineck. *A quantitative evaluation of the differences between knobs and sliders.* Proc. Int. Conf. New Interfaces for Musical Expression (NIME), 2009.

[52] S. Gelineck and S. Serafin. *A practical approach towards an exploratory framework for physical modeling.* Computer Music Journal , vol. 34, no. 2, pp. 51 65, 2010.

[53] R. J. Hanson and C. L. Lawson. *Solving least squares problems.* Philadelphia, 1995.

[54] R. J. Hanson and C. L. Lawson. *Solving least squares problems.* Philadelphia, 1995.

[55] I. Hattwick. *Mbira controller.* unknown, unknown.

[56] S. Heise and J. Loviscach. *A versatile expressive percussion instrument with game technology.* In Proc. IEEE Int. Conf. on Multimedia and Expo (ICME), 2008.

[57] L. L. Henrique and J. Antunes. *Optimal design and physical modelling of mallet percussion instruments.* Acta Acustica united with Acustica , vol. 89, no. 6, pp. 948963, 2003.

[58] G. Hoffman and G. Weinberg. *Gesture-based humanrobot jazz improvisation.* Int. Conf. on Robotics and Automation (ICRA), 2010.

[59] G. Hoffman and G. Weinberg. *Shimon: an interactive improvisational robotic marimba player.* CHI Extended Abstracts, 2010.

[60] Donald R. Hill (Translator) Ibn al-Razzaz al Jazari. *The Book of Knowledge of Ingenious Mechanical Devices: (Kitb f ma 'rifat al-iyal al-handasiyya).* Springer, 1974.

[61] L. Zhong J. Wickramasuriya J. Liu, Z. Wang and V. Vasudevan. *uwave: Accelerometer-based personalized gesture recognition and its applications*. IEEE Int. Conf. on Pervasive Computing and Communications , pages 19, 2009.

[62] et el. John Beck. *Encyclopedia of Percussion, pg. 401*. Routledge, 2007.

[63] I.T. Jolliffe. *Principal Component Analysis*. Springer, 2002.

[64] P. Cook S. Psaroudaks D. Kamarotos K. Tsachalinas, K. Tzedaki and T. Rikakis. *Physical modeling simulation of the ancient greek elgin auloi*. in Proc. s of the Int. Computer Music Conf. (ICMC), 1997.

[65] A. Kapur. *The karmetik notomoton: A new breed of musical robot for teaching and performance*. Proc. Int. Conf. New Interfaces for Musical Expression (NIME), 2011.

[66] A. Kapur and et al. *Ensembles of digitally controlled acoustic instruments*. Computer Music Journal, 2011.

[67] Ajay Kapur. *A HISTORY OF ROBOTIC MUSICAL INSTRUMENTS*. ICMC, 2005.

[68] G. Kubik. *Lamellophone, in: The New Grove Dictionary of Music and Musicians*. Macmillan Publishers, 1981.

[69] S. Trail G. Tzanetakis L. Jenkins, W. Page and P. Driessen. *An easily removable, wireless optical sensing system (eross) for the trumpet*. Proc. Int. Conf. on New Interfaces for Musical Expression (NIME), 2013.

[70] M. Boch C. Southworth L. S. Pardue, A. Boch and A. Rigopulos. *Gamelan elektrika: An electronic Balinese Gamelan*. In Proc. Int. Conf. on New Interfaces for Musical Expression (NIME), 2011.

[71] H. Leeuw. *The electrumpet, a hybrid electro-acoustic instrument*. Proc. Int. Conf. on New Interfaces for Musical Expression (NIME), 2009.

[72] L. F. Ludwig. *Real-time floor controller for control of music signal processing, mixing, video, lighting, and other systems*. Patent, 2004.

[73] R. Nikolaidis M. Bretan, M. Cicconet and G. Weinberg. *Developing and composing for a robotic musician using different modes of interaction.* Proceedings of the International Computer Music Conference (ICMC), 2012.

[74] R. Lunding M. Brynskov and L. S. Vestergaard. *Dul radio: A light-weight, wireless toolkit for sketching in hardware.* Tangible and Embedded Interaction Work-in- Progress Workshop Proceedings, 2011.

[75] C. Erkut M. Karjalainen and L. Savioja. *Compilation of unified physical models for efficient sound synthesis.* IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP), 2003.

[76] W.-M. Lai M. Macon, A. McCree and V. Viswanathan. *Efficient analysis/synthesis of percussion musical instrument sounds using an allpole model.* in IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP) , vol. 6, 1998.

[77] T. Apel M. Puckette and D. Zicarelli. *Real-time audio analysis tools for pd and msp.* In International Computer Music Conference, 1998.

[78] R. Cassidy M. Wright and M. F. Zbyszynski. *Audio and gesture latency measurements on linux and osx.* CCRMA, 2004.

[79] D. MacConnell and et al. *Reconfigurable autonomous novel guitar effects (range).* Proc. Int. Conf. Sound and Music Computing (SMC), 2013.

[80] T. Machover. *Hyperinstruments: A composer's approach to the evolution of intelligent musical instruments.* Organized Sound, pages 67-76, 1991.

[81] M. Mathews. *The Radio Drum as a synthesizer controller.* In Proc. Int. Computer Music Conference (ICMC), 1989.

[82] A. McPherson. *The magnetic resonator piano: electronic augmentation of an acoustic musical instrument.* Journal of New Music Research, 2010.

[83] A McPherson. *Bela: An embedded platform for low-latency feedback control of sound.* The Journal of the Acoustical Society of America, 2017.

[84] J. McVay. *Mechbass: A systems overview of a new four-stringed robotic bass guitar.* School of Engineering Victoria University, Technical Report, 2012.

[85] J. McVay. *Mechbass: A systems overview of a new four-stringed robotic bass guitar.* School of Engineering Victoria University, Tech. Rep., 2012.

[86] Pat Metheny. *The Orchestrion Project.* Nonesuch, 2013.

[87] J.W Beak M.J Yoo and I. K. Lee. *Creating musical expression using kinect.* Proc. New Interfaces for Musical Expression (NIME), Oslo, Norway, 2011.

[88] J. Murphy. *The helio: A study of membrane potentiometers and long force sensing resistors for musical interfaces.* Proc. Int. Conf. on New Interfaces for Musical Expression (NIME), 2010.

[89] V. Narnjo. *The dagari music of ghana.* Percussive Notes, 1992.

[90] V. Narnjo. *My introduction to the gyil.* Percussive Notes, 1996.

[91] B. Neill and B. Jones. *Posthorn.* Proceedings of the 28th of the international conference extended abstracts on Human factors in computing systems, 2010.

[92] B. Neill and B. Jones. *Posthorn.* Proceedings of the 28th of the international conference extended abstracts on Human factors in computing systems, pages 31073112. ACM, 2010.

[93] S. Ness and et al. *Music information robotics: Coping strategies for musically challenged robots.* Proc. International Society for Music Information Retrieval Conf. (ISMIR), 2011.

[94] Bernhard Niedermayer. *Non-negative matrix division for the automatic transcription of polyphonic music.* In Proc. of the ISMIR, 2008.

[95] Friedrich Nietzsche. *Die Geburt der Tragdie aus dem Geiste der Musik.* E. W. Fritzsch, 1872.

[96] J.H. Kwabena Nketia. *The music of Africa.* Norton, 1974.

[97] J. A. Olivares. *Electric Kalimbatronique.* unknown, unknown.

[98] H. F. Olson. *Music, Physics and Engineering.* 2nd ed. Dover Publications Inc., 1967.

[99] H. F. Olson. *Music, Physics and Engineering.* Dover Publications Inc., 1967.

[100] D. Morrill P. Cook and J. Smith. *A midi control and performance system for brass instruments.* Proc. of the International Computer Music Conference (ICMC), 1993.

[101] T. E. Darcie P. F. Driessen and B. Pillay. *The effects of network delay on tempo in musical performance.* Computer Music Journal, vol. 35, no. 1, pp. 7689,, 2011.

[102] J. Paradiso and N. Gershenfeld. *Musical applications of electric field sensing.* Computer music journal, 21(2):6989, 1997.

[103] Steven Ness George Tzanetakis Peter van Kranenburg, Daniel Peter Biro. *A COMPUTATIONAL INVESTIGATION OF MELODIC CONTOUR STABILITY IN JEWISH TORAH TROPE PERFORMANCE TRADITIONS.* 12th International Society for Music Information Retrieval Conference (ISMIR), 2011.

[104] Tim Ponting. *Simmons Silicon Mallet.* Music Technology Magazine, Jan., 1988.

[105] M. S. Puckette. *Pure data: another integrated computer music environment.* Proceedings Second Intercollege Computer Music Concerts, 1996.

[106] A. Schloss R. Jones, P. Driessen and G. Tzanetakis. *Digitizing North Indian Performance: Extension and Preservation Using Multimodal Sensor Systems, Machine Learning and Robotics.* Dissertation, University of Victoria, 2007.

[107] A. Schloss R. Jones, P. Driessen and G. Tzanetakis. *Advancing the Art of Percussion.* Dissertation, University of Victoria, 2009.

[108] A. Schloss R. Jones, P. Driessen and G. Tzanetakis. *A force-sensitive surface for intimate control.* Proceedings of the Conference on New Interfaces for Musical Expression (NIME), pp. 236241, 2009.

[109] Don Randel. *Cycle.* The Harvard Dictionary of Music, Cambridge, MA: Belknap Press, p. 218, 1986.

[110] George Tzanetakis Robert Van Rooyen, Andrew Schloss. *Voice Coil Actuators for Percussion Robotics.* NIME, 2017.

[111] F. Rocha and J. Malloch. *The hyper-kalimba: Developing an augmented instrument from a performers perspective.* In Proc. Int. Conf. Sound and Music Computing (SMC), 2009.

[112] Robert Rowe. *Machine Musicianship.* MIT Press, 2001.

[113] D. Godlovich S. Trail, T. F. Tavares and G. Tzanetakis. *Direct and surrogate sensing for the gyil African xylophone.* proceedings of NIME, 2012.

[114] T. Tavares G. Odowichuk P. Driessen A. Schloss S. Trail, M. Dean and G. Tzanetakis. *Non-invasive sensing and gesture control for pitched percussion hyperinstruments using the kinect.* Proc. Int. Conf. New Interfaces for Musical Expression (NIME), 2012.

[115] H. Sakoe and S. Chiba. *Dynamic programming algorithm optimization for spoken word recognition.* IEEE Transactions on Acoustics, Speech and Signal Processing, 1978.

[116] Colin Honigman Ajay Kapur Sarah Reid, Ryan Gaston. *MIGSI (Minimally Invasive Gesture Sensing Interface).* NIME, 2016.

[117] D. Schlessinger and J. O. S. III. *The kalichord: A physically modeled electroacoustic plucked string instrument.* In Proc. Int. Conf. New Interfaces for Musical Expression (NIME), 2009.

[118] W. Andrew Schloss and David A. Jaffe. *Intelligent musical instruments: The future of musical performance or the demise of the performer?* Journal of New Music Research Pages 183-193, 2008.

[119] E. Singer. *Lemur guitarbot: Midi robotic string instrument.* Proc. Int. Conf. New Interfaces for Musical Expression (NIME), 2003.

[120] P. Smaragdis and J.C. Brown. *Non-negative matrix factorization for polyphonic music transcription.* IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, pages 177-180, 2003.

[121] H.-C. Steiner. *Firmata: Towards making microcontrollers act like extensions of the computer.* Proceedings of NIME 2009 New Interfaces for Musical Expression, 2009.

[122] M. Sterling and M. Bocko. *Empirical physical modeling for bowed string instruments.* in IEEE International Conference on Acoustics Speech and Signal Processing pp. 433436, 2010.

[123] O. Cornelis F. Gouyon A. Rauber C.A.A. Kaestner T. Lidy, C.N. Silla Jr and A.L. Koerich. *On the suitability of state-of-the-art music information retrieval methods for analyzing, categorizing and accessing non-western and ethnic music collections*. Signal Processing, 2010.

[124] M. Laurson T. Takala, J. Hiipakka and V. Vlimki. *An Expressive Synthesis Model for Bowed String Instruments*. ICMA, 2000.

[125] J. Thibodeau. *Trumpet augmentation: Rebirth and symbiosis of an acoustic instrument*. Masters thesis, McGill University, 2010.

[126] A. R. Tindale. *A hybrid method for extended percussive gesture*. in Proc. New Interfaces for Musical Expression (NIME), 2007.

[127] A.R. Tindale. *A hybrid method for extended percussive gesture*. In Proc. New Interfaces for Musical Expression (NIME), 2007.

[128] T.Machover and J.Chung. *Hyperinstruments: Musically Intelligent and Interactive Performance and Creativity Systems*. SProceedings of the International Computer Music Conference (ICMC), 1989.

[129] G. Toussaint. *The Geometry of Musical Rhythm: What Makes a Good Rhythm Good?* Chapman and Hall/CRC, January 11, 2013.

[130] S. Trail. *El-lamellophone - a low-cost diy open framework for acoustic lemellophone based hyperinstruments*. Proc. Int. Conf. on New Interfaces for Musical Expression (NIME), 2014.

[131] S. Trail and et al. *Non-invasive sensing and gesture control for pitched percussion hyper-instruments using the kinect*. Proceedings of the Conference on New Interfaces for Musical Expression (NIME), 2012.

[132] S. Trail and G. Tzanetakis. *The Wiikembe — performer designed lamellophone hyperinstrument for idiomatic musical-dsp interaction*. In IEEE Pacicfic Rim Conf. On Communications, Computers and Signal Processing (PacRim), 2013.

[133] S. Trail and G. Tzanetakis. *The Wiikembe: performer designed lamellophone hyperinstrument for idiomatic musical-dsp interaction*. IEEE Pacicfic Rim Conf. On Communications, Computers and Signal Processing (PacRim), 2013.

[134] C. Traube and J.O. Smith. *Estimating the plucking point on a guitar string*. In Proc. Digital Audio Effects, 2000.

[135] Joseph Kobom. Trevor Wiggins. *Xylophone Music From Ghana*. White Cliffs Media Company, 1992.

[136] Ajay Kapur; Eric Singer; Manj Benning; George Tzanetakis; Trimpin. *Integrating hyperinstruments, musical robots and machine musicianship for North Indian classical music*. Proceedings of the Conference on New Interfaces for Musical Expression (NIME), 2007.

[137] Steven Kemper Troy Rogers and Scott Barton. *MARIE: Monochord-Aerophone Robotic Instrument Ensemble*. Proceedings of the 15th International Conference on New Interfaces for Musical Expression (NIME), 2015.

[138] G. M. Tucker and Roger Parker. *Cyclic Form*. The Oxford Companion to Music, edited by Alison Latham, Oxford and New York: Oxford University Press, 2001.

[139] Luca Turchet, Andrew McPherson, and Carlo Fischione. *Smart Instruments: Towards an Ecosystem of Interoperable Devices Connecting Performers and Audiences*. Proceedings of the Sound and Music Computing Conference, 2016.

[140] TD. Tymoczko. *A Geometry of Music: Harmony and Counter-point in the Extended Common Practice*. Oxford University Press,1 edition, 2011.

[141] Tiago F. Tavares ; Gabrielle Odowichuck ; Sonmaz Zehtabi ; George Tzanetakis. *Audio-visual vibraphone transcription in real time*. IEEE 14th International Workshop on Multimedia Signal Processing (MMSP), 2012.

[142] unknown. *Electric lemellophone*. Wikipedia, unknown.

[143] D. Matignon V. Doutaut and A. Chaigne. *Numerical simulations of xylophones. ii. time-domain modeling of the resonator and of the radiated sound pressure*. Journal of the Acoustical Society of America, 104(3):1633-1647, 1998.

[144] Mark Vail. *Vintage Synthesizers*. Backbeat Books, 1993.

[145] Michael B Vercelli. *Ritual Communication Through Percussion: Identity and Grief Governed by Birifor Gyil Music*. unknown, unknown.

[146] G. Weinberg and S. Driscoll. *Toward robotic musicianship.* Computer Music Journal, 2006.

[147] D. Wessel and M. Wright. *Problems and prospects for intimate musical control of computers.* Computer Music Journal , pp. 11-22, 2001.

[148] Tim Wilson. *Mikael Seifu- ZELALEM.* Quietus, 2016.

[149] E. L. Wong and et al. *Designing wii controller: a powerful musical instrument in an interactive music performance system.* Proc. of the 6th Int. Conf. On Advances in Mobile Computing and Multimedia, 2008.

[150] MATTHEW WRIGHT. *Open Sound Control: an enabling technology for musical networking.* unknown, 2005.

[151] Diana Young. *The Hyperbow Controller: Real-Time Dynamics Measurement of Violin Performance.* Proceedings of the Conference on New Interfaces for Musical Expression (NIME), 2007.

[152] Ernst Zacharias. *Hohner: Guitaret Manual.* Hohner, Trossingen Germany, 1963.

[153] A. Zorana and J. A. Paradiso. *The chameleon guitar guitar with a replaceable resonator.* Journal of New Music Research, 2011.