HUMAN-INFORMED ROBOTIC PERCUSSION RENDERINGS:

Acquisition, Analysis, and Rendering of Percussion Performances Using Stochastic Models and Robotics

by

Robert Martinez Van Rooyen
B.S.C.p.E, California State University Sacramento, 1993
M.S.C.S, California State University Chico, 2000

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

DOCTOR OF PHILOSPHY

in the Department of Computer Science

Supervisory Committee

HUMAN-INFORMED ROBOTIC PERCUSSION RENDERINGS:

Acquisition, Analysis, and Rendering of Percussion Performances Using Stochastic Processes and Robotics

by

Robert Martinez Van Rooyen
B.S.C.p.E., California State University Sacramento, 1993
M.S.C.S., California State University Chico, 2000

Supervisory Committee

Dr. George Tzanetakis, Supervisor
Department of Computer Science, Electrical Engineering, and Music

Dr. Andrew Schloss, Co-supervisor
School of Music, Department of Computer Science

Dr. Peter Driessen, Outside Member
Department of Electrical Engineering, Computer Engineering, and Music

Abstract

A percussion performance by a skilled musician will often extend beyond a written score in terms of expressiveness. This assertion is clearly evident when comparing a human performance with one that has been rendered by some form of automaton that expressly follows a transcription. Although music notation enforces a significant set of constraints, it is the responsibility of the performer to interpret the piece and "bring it to life" in the context of the composition, style, and perhaps with a historical perspective. In this sense, the sheet music serves as a general guideline upon which to build a credible performance that can carry with it a myriad of subtle nuances. Variations in such attributes as timing, dynamics, and timbre all contribute to the quality of the performance that will make it unique within a population of musicians. The ultimate goal of this research is to gain a greater understanding of these subtle nuances, while simultaneously developing a set of stochastic motion models that can similarly approximate minute variations in multiple dimensions on a purpose-built robot. Live or recorded motion data, and algorithmic models will drive an articulated robust multi-axis mechatronic system that can render a unique and audibly pleasing performance that is comparable to its human counterpart using the same percussion instruments. By utilizing a non-invasive and flexible design, the robot can use any type of drum along with different types of striking implements to achieve an acoustic richness that would be hard if not impossible to capture by sampling or sound synthesis. The flow of this thesis will follow the course of this research by introducing high-level topics and providing an overview of related work. Next, a systematic method for gesture acquisition of a set of well-defined percussion scores will

be introduced, followed by an analysis that will be used to derive a set of requirements for motion control and its associated electromechanical subsystems. A detailed multidiscipline engineering effort will be described that culminates in a robotic platform design within which the stochastic motion models can be utilized. An analysis will be performed to evaluate the characteristics of the robotic renderings when compared to human reference performances. Finally, this thesis will conclude by highlighting a set of contributions as well as topics that can be pursued in the future to advance percussion robotics.

Table of Contents

Supervisory Committee ii

Abstract iii

Table of Contents v

List of Tables x

List of Figures xi

Acknowledgments xviii

CHAPTER 1: Introduction 1

Overview 3

Motion Capture 4

Motion Dataset 6

Motion Analysis 7

Voice Coil Actuators 9

Mechatronics 11

Contributions 13

Summary 14

CHAPTER 2: Related Work 15

Gesture Acquisition 16

Expressive Performances 18

Percussion Robotics 20

Non-musical Robotics 26

Cyber-Physical Systems 28

Summary 28

CHAPTER 3: Gesture Acquisition 30

Capture system 31

Calibration 36

Motion Data 39

Summary 41

CHAPTER 4: Performer-specific Analysis of Percussion Gestures 43

Stochastic Model 44

Setup of Study 45

Analysis Application 50

Parameter Vector 52

Timing 57

Velocity 60

Location 62

Performer Analysis 63

Comparative Analysis 68

Summary 69

CHAPTER 5: Mechatronic System 70

System Requirements 71

System Design 73

Mechanical Design 74

Voice Coil Actuators 76

Vertical Axis 94

Horizontal Axis 95
Enclosure 96
Assembly 97
Electronic Design 102
Field Programmable Gate Array 104
Optical Quadrature Encoders 107
H-Bridge Drivers 107
Closed Loop Control 109
Strike Detection 111
Strain Detection 112
User Interface 112
Audio 113
Peripheral Ports 113
Power 114
Temperature Monitor 114
Development Features 115
Software 116
PetaLinux 116
Bare-metal Firmware 117
Interprocess Communications 117
Abstraction Library 118
Open Sound Control 120
Musical Instrument Digital Interface 122

Calibration 124
Diagnostics 125
Configuration 126
Patches 127
User Interface 129
Summary 130
CHAPTER 6: System Evaluation 132
Overview 133
Speed 133
Velocity 134
Location 135
Bounce 136
Timing Compensation 137
Orchestral Rendering 138
Musical Instrument Digital Interface 139
Open Sound Control 140
Performances 143
Summary 147
CHAPTER 7: Conclusion 149
Future Work 151
Bibliography 153
Appendices 167
Appendix A 167

Appendix B 170

Appendix C 193

Appendix D 194

Appendix E 195

Appendix F 196

Appendix G 210

Appendix H 215

Appendix I 228

List of Tables

Table 1. Percussive Arts Society rudiments. 47

Table 2. Parameter vector example. 53

Table 3. Drum study participants attributes. 64

Table 4. High-level requirements matrix. 71

Table 5. Voice coil actuators electrical parameters. 77

Table 6. Dynamic consistency percentage comparison. 85

Table 7. Diagnostic command listing. 126

Table 8. Field Programmable Gate Array pinout and signals information. 203

Table 9. Debug/expansion port signals. 204

Table 10. Carrier module BOM. 206

Table 11. Keyboard module BOM 208

Table 12. Mechanical BOM. 210

Table 13. MIDI Mapping chart. 215

Table 14. MIDI continuous controller chart. 216

Table 15. Field programmable gate array register summary. 218

List of Figures

Figure 1. Motion capture recording session. 9

Figure 2. Voice coil actuator mounted in initial prototype. 11

Figure 3. Top down diagonal view of the MechDrum[tm]. 13

Figure 4. Panharmonicon orchestral machine. 21

Figure 5. The Machine Orchestra of CalArts. 22

Figure 6. The Logos robot orchestra. 23

Figure 7. Robotic marimba player named "Shimon." 24

Figure 8. The Pat Metheny Orchestrion album. 25

Figure 9. Recording session showing the participant, calibrated backdrop, lighting, and video camera. 32

Figure 10. Diagram showing the motion capture system profile that highlights the cameras field of view and instrument angle. 33

Figure 11. Camera motion capture view of striking implements and drum with calibrated backdrop. 34

Figure 12. Power spectrum of microphone and transducer. 36

Figure 13. Timbre score composed of quarter notes per measure for three impact regions. 36

Figure 14. Strike impact regions on the drum head surface. 37

Figure 15. Timbre impact region motion plot. 38

Figure 16. Dynamics score composed of eight quarter note crescendo. 38

Figure 17. Normalized dynamics curve showing peak sample and polynomial curve. 39

Figure 18. Double stroke open roll rudiment showing the left and right striking implement tip elevation over time. 40

Figure 19. Double stroke open roll rudiment notation. 40

Figure 20. Double stroke open roll rudiment showing a detailed view of left and right striking implement tip elevation over time with their associated drum surface strike events. 41

Figure 21. Diagram showing an external trigger, tempo value, and dynamic/static parameters driving stochastic model that generates an onset, velocity, and position tuple. 44

Figure 22. Motion capture frame in the Tracker application. 49

Figure 23. Single stroke roll left and right striking implement plot 50

Figure 24. Custom application that extracts the onset, velocity, and elevation parameter vector from recorded motion data. 52

Figure 25. Onset sample in comparison to literal performance for Double Stroke Open Roll rudiment that was performed by participant #1. 54

Figure 26. Position sample showing the variability in strike locations for Double Stroke Open Roll rudiment that was performed by participant #1. 55

Figure 27. Velocity sample showing the intentional and nuanced strike dynamics for Double Stroke Open Roll rudiment that was performed by participant #1. 56

Figure 28. Renderings of literal, dynamics, timing, and timing/dynamics performances. 56

Figure 29. Tempo drift signal derived from onset time versus literal time for the Double Stroke Open Roll rudiment performed by participant #1. 58

Figure 30. Tempo drift histogram showing bimodal tendency for Double Stroke Open Roll rudiment that was performed by participant #1. 59

Figure 31. Visible strike drift when compared to metrical time. 60

Figure 32. Velocity histogram showing frequency and magnitude for Double Stroke Open Roll rudiment that was performed by participant #1. 61

Figure 33. Hand position histogram showing frequency and locations for Double Stroke Open Roll rudiment that was performed by participant #1..... 63

Figure 34. Parameter vector plot showing the mean and standard deviation of onset, velocity, and elevation parameters for participant #1. 65

Figure 35. Parameter vector plot showing the mean and standard deviation of onset, velocity, and elevation parameters for participant #2. 66

Figure 36. Parameter vector plot showing the mean and standard deviation of onset, velocity, and elevation parameters for participant #3. 67

Figure 37. Parameter vector plot showing the mean and standard deviation of onset, velocity, and elevation parameters for participant #4. 68

Figure 38. Performance comparison of onset standard deviation, mean velocity, and mean elevation with related standard deviation error bars. 69

Figure 39. Mechatronic system block diagram. 74

Figure 40. MechDrum^{tm} promotional photo collage..... 75

Figure 41. Voice coil actuator cutaway diagram. 78

Figure 42. Optical code wheel and quadrature encoder diagram. 80

Figure 43. PID controller flow diagram. 81

Figure 44. Actuator latency comparison chart. 83

Figure 45. Maximum loudness comparison chart. 84

Figure 46. Voice Coil Actuator loudness compared to elevation. 84

Figure 47. Maximum strike repetition rate chart. 86

Figure 48. First prototype system schematic. 87

Figure 49. First mechanical prototype showing VCA in relation to striking implement. 88

Figure 50. First integrated prototype showing VCA and electronics. 89

Figure 51. First prototype used to bring-up second hardware platform. 90

Figure 52. PID closed loop control tuning plot. 91

Figure 53. Motion captured position compared to playback location. 92

Figure 54. Multiple strike plot showing implement tip bounce. 93

Figure 55. Vertical and horizontal motion planes. 95

Figure 56. X axis rotating clevis connector. 96

Figure 57. 3D model of enclosure with airflow path. 97

Figure 58. Pre-assembly test fit, orthogonal view. 98

Figure 59. Pre-assembly test fit, voice coil mount. 98

Figure 60. Final assembly, voice coils and linkages. 99

Figure 61. Final assembly, right linkage and wiring. 100

Figure 62. Final assembly, linkages and wiring complete. 100

Figure 63. Final assembly, voice coil actuators and connector bulkhead. 101

Figure 64. Completed assembly under development in the lab. 101

Figure 65. AVNET MicroZed SBC hosted by custom PCBA. 102

Figure 66. Custom carrier module multi-view model. 103

Figure 67. Custom keyboard module multi-view model. 104

Figure 68. Xilinx Zynq internal architecture (image provided by the Xilinx Corporation). 105

Figure 69. Logical system architecture diagram. 106

Figure 70. Location and closed loop controller output during a strike event. 108

Figure 71. Digital to analog conversion board. 109

Figure 72. Closed loop control system. 110

Figure 73. Percussive surface plot. 111

Figure 74. Internal microphone frequency response. 113

Figure 75. Example multi-threaded system logging output. 117

Figure 76. OpenAMP system sequence diagram (image provided by Xilinx Corporation). 118

Figure 77. Abstraction library API listing. 120

Figure 78. OSC packet data bundle. 121

Figure 79. Partial parameter section file listing. 127

Figure 80. Partial human patch section file listing. 128

Figure 81. User interface display and buttons. 129

Figure 82. Pilot study SPL range comparison. 135

Figure 83. Strike location spectrum. 136

Figure 84. Position and location tracking. 137

Figure 85. Timing compensation as a surface for normalized velocity and elevation. 138

Figure 86. Cross-correlation of human and mechatronic drummer orchestral performance. 139

Figure 87. MIDI event latencies. 140

Figure 88. Whack and strike detection. 141

Figure 89. MechDrum[tm] performance at the 2018 Guthman musical instrument competition. 143

Figure 90. Exhibition at the 2018 NIME conference. 144

Figure 91. Dr. Andrew Schloss providing a demonstration during his lecture at the International Symposium of New Music in Brazil. 145

Figure 92. 2018 Interactive Art, Science, and Technology symposium at Lethbridge University. 147

Figure 93. Carrier board schematic, page 1 of 6. 197

Figure 94. Carrier board schematic, page 2 of 6. 198

Figure 95. Carrier board schematic, page 3 of 6. 199

Figure 96. Carrier board schematic, page 4 of 6. 200

Figure 97. Carrier board schematic, page 5 of 6. 201

Figure 98. Carrier board schematic, page 6 of 6. 202

Figure 99. Wiring diagram. 205

Figure 100. MicroZed block diagram (image provided by AVNET, Inc.). 209

Figure 101. MicroZed functional overlay (image provided by AVNET, Inc.) 209

Figure 102. Y axis voice coil actuator specifications (provided by MotiCont). 213

Figure 103. X axis voice coil actuator specifications (provided by MotiCont). 214

Figure 104. Software/hardware architecture. 217

Figure 105. FPGA version register. 219

Figure 106. FPGA general purpose control register. 219

Figure 107. FPGA PID control and status regsiter. 220

Figure 108. FPGA PID interrupt status register. 220

Figure 109. FPGA PID interrupt mask register. 221

Figure 110. FPGA PID interrupt pending register. 221

Figure 111. FPGA PID strike register. 221

Figure 112. PID strike register. 222

Figure 113. PID post-strike register. 222

Figure 114. PID position regsiter. 222

Figure 115. PID location register. 223

Figure 116.PID proportional gain register. 223

Figure 117. PID integral gain register. 224

Figure 118. PID proportional gains register. 224

Figure 119. PID bias register. 224

Figure 120. PID clock divisor register. 225

Figure 121. PWM duty cycle margin register. 225

Figure 122. PWM duty cycle timeout register. 225

Figure 123. PWM clock divisor register. 225

Figure 124. Key LWD intensity register. 226

Figure 125. LCD backlight intensity register. 226

Figure 126. LED PWM divisor register. 226

Figure 127. LED control register. 227

Acknowledgments

My pursuit of a doctoral degree in computer science would not have been possible without the tremendous and sustained support of my wife Lisa, my son Chase, and my daughter Lindsay. Through it all they managed to work around the rigours of my academic and consulting business responsibilities, while offering encouragement and constructive feedback along the way. I will forever be in their debt for the patience and dedication they have shown me during my years of study and research. I am very grateful for the guidance, support, and recommendations of my supervisor George Tzanetakis and my co-supervisor Andrew Schloss. George's idea of using voice coils in the context of a percussion instrument led me on a unique path of discovery and creativity that has been truly rewarding on both an academic and personal level. I am especially indebted to Andrew, for his enthusiasm in my research and willingness to participate at a level that not only made my journey possible, but celebrated the results with unique and inspiring performances using my robotic instrument. As my teacher and enthusiast, Kirk McNally shared his vast knowledge of audio recording, helped arrange a pilot drum study, and encouraged me to enter my instrument in an international competition that had a significant impact on my trajectory as a graduate student. I would also like to thank each participant in my pilot drum study for their time, quality of their performances, and the dedication to the pursuit of knowledge in the area of percussion musicianship. My successes in the area of mechanical design is the direct result of working with my friend Max Rukosuyev, whose multiple rounds of feedback, expertise in CNC machining, and dedication to precision engineering yielded an exceptional outcome. I would like to

extend a very special thank you to Georgia Tech University and the Margret Guthman musical instrument competition under the direction of Gil Weinberg, and the judges, Perry Cook, Suzanne Ciani, and Jesper Kouthoofd for their time and consideration of my mechatronic instrument. I also wish to thank the Office of Research Services at the University of Victoria, Human Research Ethics Board at the University of Victoria, and the University of Victoria School of Music for their contributions to the pilot study. I would like to express my gratitude to the University of Victoria Industry Liason Officer Aislinn Sirk for her generous time and interest in patenting my intellectual property, and the Coast Capital Savings Innovation Centre team of Jerome Etwaroo and Tyler West for their dedication in supporting the entrepreneurial spirit. Finally, I want to thank my parents for providing the environment and opportunities in my formative years that sparked my interest in engineering within a backdrop of music. As a life-long primarily self-taught professional accordionist from Amsterdam, my father's interest in being able to play a Hammond B3 organ was the genesis for my creation of one of the first hybrid MIDI accordions in 1984. I had achieved this rather ambitious undertaking at the age of nineteen after receiving the MIDI specification I had ordered by mail [1]. I used an Intel 8080 processor trainer from Diablo Valley College, a Roland Juno-106 MIDI keyboard that we had borrowed from a local music store, and my Father's Italian made electronic accordion, along with endless hours of pouring over datasheets, prototyping, wire wrapping, and hand writing machine code without access to a personal computer.

CHAPTER 1: Introduction

"When I lost the use of my hi-hat and bass drum legs, I became basically a singer. I was a drummer who did a bit of singing, and then I became a singer who did a bit of percussion." – Robert Wyatt

This famous quote by Robert Wyatt, a founding member of the influential Canterbury scene, reveals the unfortunate consequences of an accident that led him on an alternate path from his natural talent as a drummer [2]. Although his trajectory as a musician continued to gain momentum, his work as a percussionist was limited to instruments that did not require the use of his legs. In this context, imagine the impact of a system that could remap his remaining mobility in a manner that could effectively replace what he had lost [3]. The concept of accessibility is critically important in the modern world for people with a wide array of disabilities that need direct or at least indirect access to physical objects [4]. This of course applies to all of the items one would need to conduct the daily business of living, but this list must include devices that enable self-expression and the creation of art in its infinite forms [5]. Going beyond accessibility, the therapeutic value of creating music has been demonstrated in numerous qualitative studies [6, 7]. Playing music contributes to self-identity and connecting with other people while simultaneously building self-esteem, competence, independence, spiritual expression, and avoiding isolation [8].

What if a percussionist had the ability to play a traditional acoustic instrument from a distance to enhance or augment their performance? The remote instrument could be

across the stage, on the ceiling of a concert hall, or possibly on the other side of the world [9]. This line of thinking can expand both the size and colour capacity of the contemporary percussionist's palette. Live and nuanced performances with musicians in different physical locations create the potential for increasing collaborations and fostering creative productions [10].

Whether in the recording studio or in an interactive art installation, the ability to record and playback highly expressive percussion pieces opens the door to new methods of capturing and reproducing performances [11]. For example, a recording session can capture the motion of a performance that can be edited for an optimal rendering by a mechatronic system. The same performance can be played back over and over until the recording equipment, room acoustics, and motion has been optimized for multi-track audio recording. Similarly, the recording and playback of rudiments, and musical pieces at variable rates can be used in a pedagogical context [12].

The central question in my research is whether or not it is possible to develop a robotic percussionist that is capable of comparable expressiveness to a human performer. The goal of this dissertation is to describe the research that led to a process and mechatronic system that can have a positive impact on the human experience with respect to artistic expression. Whether for accessibility, remote control, recording, education, or artistic exhibition, the research and development presented in this thesis may serve as a case study for an entirely new approach to percussion. To achieve this goal, significant software design and development will be required in the context of data analysis, signal processing, multi-core embedded system infrastructure, device drivers, multi-threaded applications, and closed loop control with a subset of constructs contained within a

hardware definition language as opposed to code running on a microprocessor. The aforementioned list of software components implies a cutting edge electronic and mechanical infrastructure that can deliver the performance and physical motion needed for a purpose-built mechatronic drummer.

As an introduction to my research, this chapter takes a conceptual view of major topics that ultimately resolve to a set of concrete contributions. After a brief overview, the material introduces the process of motion capture, a dataset, and subsequent analysis. This is followed by a primer on voice coil actuators and the remaining mechatronic system that is composed of mechanical, electronic, and software components.

Overview

The production of a quality snare drum performance can take many years of instruction and practice to achieve [13]. Subtle nuances in timing, dynamics, and timbre can separate musicians despite using the same instrument, striking implements, and sheet music. How is this possible? It is a well-known fact that each artist develops their own style, but can this property be quantified in a tangible and reproducible fashion [14]? By recording a performance in a consistent and non-invasive manner, one can in fact discover what makes a particular performer's work unique among their peers.

In order to establish a reference recording, a standardized and well documented score needs to be created or identified. Further, a repeatable data acquisition process is required to create not only the baseline, but subsequent recordings for the comparison set. An analysis of the recordings can lead to new discoveries in human motion as well as highlighting the subtle differences between musicians.

Artistic expression can come in many forms, but a common thread is the notion of individuality. In the case of a musical performance, the uniqueness can be subtle when comparing the work of two highly skilled musicians. Nevertheless, their technique, virtuosity, and style can still set them apart to the trained listener. You might wonder what it takes to become a trained listener. The obvious answer of course is years of formal musical training coupled with an ability to concentrate on subtle sonic detail such as timing, timbre, and dynamics. It comes as no surprise however that computers are also particularly good at differentiating stylistic attributes with the application of expressive models [15, 16].

Extending the concept of automated performance analysis a bit further, it is conceivable that computers can begin to learn what a uniquely human performance encompasses. By comparing musicians against each other and formal notation, statistical patterns and other performance traits such as latency and drift begin to emerge. By applying these elements, robotic musicians can start to incorporate nuances into their own renderings, which can dramatically improve the quality of an otherwise sterile, although technically accurate performance.

Motion Capture

Musicians interact with their instruments both in generalized and subtly nuanced ways. The former is part and parcel to learning how to play the instrument given standard instruction in the context of the associated physics. The latter is a fine tuning of the physical interaction that brings out the best musical performance and sound of the instrument. What is the distinction between these two components of a performance? By

what methods can one begin to analyze the musician's competence, and by extension, their uniqueness when compared to other performers? To answer these questions in a quantitative manner we need access to real-world data, which in this context is defined as multi-dimensional temporal data from the unencumbered musical performance of a score.

A highly trained and experienced musician can evaluate the quality of a performance purely by ear. This is of course a qualitative and highly subjective measure, but consensus within a population of experts is achievable due to the application of learning constraints [17]. If we breakdown a performance into attributes that can be graded on a scale such as timing, dynamics, and timbre, we can compute the mean, average, and standard deviations for each attribute in a survey. We could further establish a weight based on the experience of the individual evaluator, but ultimately we will derive an informed opinion on the quality of the performance.

In contrast, with access to real-world data as suggested earlier, we can begin to critically evaluate a performance against expected values and in comparison to other musicians. The expected values for attributes can be derived from an original score and can be further adjusted by genre experts with the goal of establishing a reference performance [18]. Although this adjustment can be interpreted as another form of subjective evaluation, the intent is to define a reference, which will serve as the basis for a subsequent quantitative analysis. The definition of reference is a dataset that is representative of a quality performance that can be used for comparative studies. A thorough analysis of other performances can lead to tangible conclusions about the quality of a given performance and how it is unique within a population of musicians. Further, we will be able to quantify what a high quality performance looks like in

addition to what it sounds like. In order to achieve this objective, a practical and entirely non-invasive data acquisition method for the recording and interpretation of striking implement tip motion will be established. By capturing the motion of performances using an actual acoustic instrument rather than a MIDI drum pad, one can begin to uncover the subtle nuances beyond timing that includes dynamics and timbre. Further, a study of drum head properties such as deformation and rebound can be conducted to gain a better understanding of impact events that result in a bounce.

Motion Dataset

Studying the complexities of human percussive performance can lead to a deeper understanding of how musicians interpret a musical score while simultaneously imparting personal expressiveness. This knowledge can serve to not only educate other musicians on mastering technique, but also to quantifiably describe what an exceptional performance looks like from a multi-dimensional scientific perspective. Moreover, scalable motion models and machine learning techniques can be developed to render more expressive performances in other mediums, such as robotic instruments. Although this research is being conducted in the context of music, it is conceivable that other branches of study may find elements of the dataset applicable, such as animation or cognitive sciences [19, 20].

The dataset is composed of the 40 rudiments as defined by the Percussive Arts Society, which serves as a contemporary vocabulary for the modern percussionist [21]. Rudiment classes such as drum rolls will inform the researcher on the timing, velocity, dynamics, and strike locations that comprise unique human performances. Armed with

this knowledge, one may build mechatronic or synthesis systems that can render compelling performances that move beyond precision and often stale interpretations. Other uses may include the derivation of metrics that can help beginning percussionists understand where they need to focus their training.

Motion Analysis

A percussion performance is composed of striking events that are spread over time and that include timing variations (relative to the beat), positional variations on the drum head, and changes in striking velocity, which is proportional to sound-pressure level. Taken together as a set, these multi-dimensional variables are unique to each performance, even given the same musician, score, and tempo. The differences of these fundamental variables can be even more profound across a set of musicians with comparable competence performing the same piece [22]. The former is primarily due to stochastic processes throughout the human body and brain, whereas the latter can be attributed to the addition of subtle stylistic attributes that are expressed in micro-timing [23, 24, 25, 26, 27].

The goal of this work is to explore a single representative percussion rudiment as a case study in the context of timing, velocity, and position to identify stochastic and intentional micro-timing components. Furthermore, by applying a statistical analysis to onset, velocity, and position, a parameter vector can be derived and used to render a unique performer-specific expressive performance of a score by using a stochastic model that can be coupled with an equally capable robotic or synthesis system. To be clear, the objective is parameter identification, model definition, and a basic understanding of range

rather than optimization or generalizations across a large population of musicians and/or performances.

From a research perspective, one of the primary goals is to gain a deeper understanding of human motion with respect to the striking implement so that a plurality of attributes can be infused into synthesized or robotic percussion renderings. Although a robotic rendering can be technically accurate for a piece being played, it often lacks emotion and the subtle variety that comes with a multi-dimensional human performance. Despite the fact that robotic musicians may never approach the richness and spontaneity of a human musician, it is possible to direct mechatronic or synthesis systems to render performances that are more life-like and thus more pleasing to the listener. Sam Phillips, who arguably invented Rock 'n' Roll, embraced the idea of "perfect imperfection" [28]. As the creator of Sun Records in Memphis Tennessee, he realized that subtle flaws in a performance gave songs a soul. Artists such as Elvis Presley and Johnny Cash capitalized on this approach in countless recordings that proved beneficial to their success and the nearly universal enjoyment of their music.

A fine example of rendering a human-nuanced performance on an actual instrument is the Steinway & Sons Spirio high-resolution player piano[1] . This system was designed to be capable of recording all the subtle keyboard and pedal work that takes place in a live performance. Once a given performance has been recorded using their proprietary system, it can be played again and again without losing its virtuosity or the most subtle emotion. Reproducing this capability with a percussion instrument represents a unique challenge as the physical constraints and instrument response are quite different. Going

[1] A press release describing the Steinway & Sons Spirio piano is available at http://tinyurl.com/lcbwjxn.

beyond the playback of a recording and introducing the use of stochastic models is yet another level of sophistication that can potentially open new avenues of musical expression. It is important to capture the details of human percussive performance by analysing the actual motion patterns used to create the sound rather than solely capturing the sound itself, which implies the application of a motion-capture system and process for acquiring performer-specific percussion motion data [29]. As shown in Figure 1, a calibrated commodity non-invasive motion capture system can be utilized to capture performer-specific motion data.

Figure 1. Motion capture recording session.

Voice Coil Actuators

Human percussion performances involve extremely complex biomechanics, instrument physics, and musicianship. The development of a robotic system to closely approximate the complexity of performance of its human counterpart not only requires a deep understanding of the range of motion, but also a set of technologies with a level of performance that can match or exceed empirical measurements in multiple dimensions.

There are a variety of electro-mechanical devices and configurations to choose from that offer the level of performance required for percussion robotics. However, only subsets of the devices represent viable options in practice.

Through calibrated non-invasive acquisition and analysis of the 40 percussive rudiments [21], a typical range of motion with respect to striking implement tip motion was determined. This was accomplished by developing a simple low cost motion capture system using an off-the-shelf consumer grade high frame rate video camera [29]. The actual motion data was extracted from the video footage using open source tools, which enabled subsequent analysis of timing, velocity, and position.

With an understanding of the range and speed of striking tip implement motion, a mechatronic system was designed using an industrial voice coil actuator (VCA) [30]. Unlike solenoids and DC motors, VCAs offer high-precision continuous linear control of motion with minimal power when coupled with an adequate position encoder and application-specific closed-loop servo controller. This thesis will discuss the related technologies and how they were fashioned into a basic prototype for evaluation. As shown in Figure 2, the VCA is at the very heart of the mechatronic system, but controlling this type of actuator requires a significant amount of supporting software and electronics.

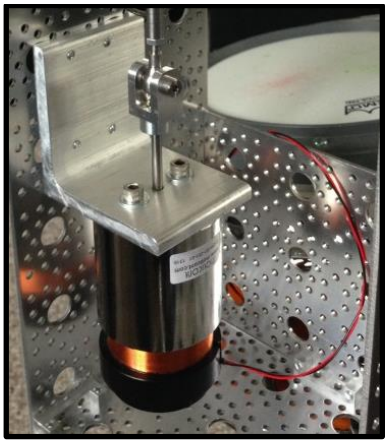


Figure 2. Voice coil actuator mounted in initial prototype.

Mechatronics

A skilled percussionist is capable of incredible speed, a large dynamic range, precision metrical timing, and subtle stylistic micro-timing that can culminate in a virtuoso performance across a variety of musical genres. What are the attributes that separate a good performance from an outstanding one? What are the expectations of timing accuracy, dynamic range, and timbres across a group of percussionists? To gain a better understanding of musicianship and technique, a pilot drum study was conducted at the University of Victoria with the goal of collecting real-world performance motion data from a group of participants using a common framework for both individual and comparative analysis [31]. Each participant was asked to play a series of calibration and rudiment fragments in the absence of any further direction. Aside from the presence of a video camera, microphone, calibrated backdrop, and lighting, both the musician and instrument were completely unencumbered from any type of data collection apparatus [29]. The experienced participants were encouraged to warm up, practice a given musical fragment for up to five minutes, and perform the piece using any type of grip (American,

French, or German) and technique that felt natural to them. Research conducted by Bouenard, et al. provides evidence that the type of grip used can have a correlation to gesture amplitude [32].

Unpacking the discoveries derived from a pilot study, is it possible to create a mechatronic drummer using voice coil actuators (VCAs) that could potentially meet and/or exceed the abilities of experienced musicians [33]? What would the architecture of such a device look like in terms of software, hardware, and mechanical components? Are there any existing technologies and/or algorithms that can be leveraged for a pragmatic and extensible design? These and many other questions formed the basis of moving forward with a conceptual prototype. Tasks such as initial component selection, regression testing, and performance analysis all contributed to the long-term vision of building a compelling mechatronic drummer platform that was capable of reproducing human motion from a striking implement tip perspective.

After several years of research and development, the MechDrum™ as shown in Figure 3 has become a reality with performance characteristics that approach and in some cases exceed its human counterparts. This goal was achieved by a careful application of the scientific method, a full multi-discipline engineering development cycle, rigorous testing and data analysis, and numerous international papers, presentations, and performances while continually applying feedback from experts in the field of electronic instruments and the arts. Many lessons were learned over the course of this research and a variety of improvements have been identified; however the general approach informed by the study and embodied in a working prototype have shown a viable instrument that is capable of extraordinary and characteristically non-robotic performances.

Figure 3. Top down diagonal view of the MechDrum^{tm}.

Contributions

The research presented in this thesis includes several key contributions to the field of musical robots. Specifically, a system and method for calibrated gesture acquisition using a single commodity camera that is non-invasive to the musician and instrument. This is followed by the specification for a stochastic model that utilizes performer-specific parameters to generate a real-time performance data stream. Finally, a unique and highly expressive multi-axis robotic platform is presented that can be utilized with any drum and a variety of striking implement types. Further innovations include scaled pressure control and strike detection, low-latency motion control over a network, instrument calibration using virtual planes, human inclusive dynamic range, strain detection, and timing compensation.

Summary

The contents of this thesis will cover related work in the field of percussion robotics as well as the use of voice coil actuators in other industries. This will be followed by a description and demonstration of gesture acquisition and performer-specific data analysis that yield parameters for a generative stochastic performance model. Finally, a highly expressive robotic platform will be defined and evaluated in terms of software, electronics, and mechanical systems that can render authentic acoustic performances that are on par with its human counterparts. This will be followed by a conclusion and enumeration of future work towards the advancement of related research.

CHAPTER 2: Related Work

There has been much work in the area of musical robot research and development [34]. It presents an interesting interdisciplinary environment where individuals or teams harness their creativity and knowledge in music, art, physics, mechanical design, electronics, software, material science, and possibly other areas of expertise. Given the orthogonal nature of the related tasks, one must often acquire an academic or at least a pragmatic multi-discipline understanding along with skills to execute on the vision of a new and truly unique mechatronic instrument. For the lay person, the question of why quickly comes to mind. The set of answers is as diverse as the individuals who develop musical robots, but a common theme may be the innate need in all of us for artistic expression along with the desire to push the boundaries of what is possible [7].

In this chapter we will review some of the seminal and ongoing work that has been done with gesture acquisition. By using a variety of sensors, cameras, and innovative techniques, crucial data has been collected that can assist a host of academic and pragmatic endeavors. Some of the work related to expressive performance research will be explored with the notion that such concepts can be leveraged into mechatronic systems. Finally, several fine examples of percussion robots will be presented that demonstrate the creativity, ingenuity, and craftsmanship of their respective researchers and inventors.

Gesture Acquisition

A major challenge of acquiring real-world data is its effect on the system being measured. One can easily postulate that attaching physical sensors to an instrument, musician, or both has the potential to adversely affect the quality of the performance and sound. As an example, previous research for capturing percussive gestures has included attaching pressure sensors, contact leads, and accelerometers to sticks and/or drumheads [35]. These sensors inadvertently cause modifications to the sound and perhaps more importantly, the playability of the instrument can be compromised, which can negatively impact the quality of a performance. Moreover, the inclusion of cables and other related hardware can significantly diminish the dynamics of the instrument and the musician's range of motion [36].

The work of S. Dahl, et al. uncovered remarkable detail associated with percussionist motion using high-speed optical motion capture [37]. This approach required the attachment of LED markers on both the participants and the striking implement that worked in concert with the commercially available Selcom Selspot[2] system. Additionally, strain gauges were added to the implements along with an electrically conductive path that provided contact force and duration measurements respectively [38]. As was noted previously, modifications to the striking implements can negatively affect playability. In addition, the cost and complexity associated with this type of motion capture system can be prohibitive. It is important to note however that each approach is motivated by

[2] The Selcom Selspot system was first introduced in 1975 and has been used in a wide variety of multi-plane motion capture applications.

different research questions, which implies that subsequent results can have equally different applications.

A comprehensive multi-dimensional percussive dataset created by Gillet, O. and Richard, G. known as the “ENST-Drums” was released in 2006 [39]. This dataset offers a rich set of audio/video data spanning three professional drummers. All of the data was collected non-invasively and manually annotated with respect to onset time and instrument type. The primary difference in comparison to the dataset presented in this thesis is that it provides a macro view of an entire drum kit. Further, the research team used two normal speed (25 frames per second) video cameras as opposed to a high-speed camera on a single instrument with distance calibration.

The research conducted by Bouenard, et al. used gesture acquisition of timpani performances to develop a virtual percussionist [40, 32, 41]. The motion data tracked key points on the upper body while performing different styles and nuances. In total, the dataset included forty-five sequences of drumming for three percussionists using five playing modes and three impact locations [40]. The performances were made up of both individual training exercises and improvisational sequences. Data collection was achieved by using a Vicon 460 system that is based on Infra-Red camera tracking and a standard DV camera [32].

In comparison to prior work, the approach presented in this thesis has its own unique set of advantages and disadvantages. One of the key benefits is non-invasive data acquisition, which allows the performer to play the instrument in a natural setting without being encumbered with sensors or augmented striking implements. Additionally, the use of low cost commodity equipment provides accessibility to researchers with limited

funding and/or access to specialized equipment. As a consequence however, positional accuracy is dictated by the resolution of the camera and the quality of the motion tracking algorithms. Furthermore, manual intervention when extracting motion data due to occlusions or motion blur can result in the introduction of positional errors, which could be manifested as discontinuities or outliers in the data.

In the context of this research, timing information that is collected through the recording of MIDI events from a drum pad is not sufficient to capture all of the subtleties of a performance. Further, different performers have unique expression and micro-timing characteristics that need to be taken into account to create performer-specific generative models of percussion motion.

Expressive Performances

A large body of work exists in the analysis of expressive musical performances on a variety of instruments [42, 43, 44, 45, 16, 46, 23, 26]. Research conducted by Berndt and Hehnal explored the degrees of freedom with respect to timing over several feature classes, including human imprecision [47]. Formal timing models were designed and implemented within a MIDI environment; however, other attributes such as dynamics and timbre were considered as future work. With regard to randomness, the team cited psychological and motor conditions as the primary contributors to timing accuracy that followed a Gaussian distribution. This was further broken down into macro and micro timing components, with the former being long-term tempo drifts and the latter being onset delays. The magnitude of timing deviations were quantified by using a normal

distribution with a mean of the exact note event time and a standard deviation in milliseconds.

The rule system defined by the Department of Speech Communication and Music Acoustics at the Royal Institute of Technology (KTH) in Stockholm was created to model the performance principles of musicians in the context of western classical, jazz, and popular music. A detailed overview of the system by Friberg et al. demonstrates how it is applied to phrasing, micro-timing, patterns, grooves, and many other attributes, including performance noise [48]. At a high level, the system takes a nominal score as its input and produces a musical performance as output by applying rules whose behaviours are dictated by k values that specify the magnitude of the expressiveness. There have been several practical MIDI implementations of the system that have iterated on rule design and have shown great promise in humanizing an otherwise sterile score at macro and micro temporal levels. In the context of the present work, the simulations of inaccuracies in human motor control are of particular interest. Perception experiments conducted by Juslin et al. have shown that the introduction of a noise rule results in higher ratings by listeners in the category of "human" likeness [49].

The concept of expressivity in robotic music has been explored by Kemper and Barton [50]. The use of sound control parameters is a common technique when attempting to develop expressive instruments. The intent is to convey an emotional communication to the listener by presenting an "intransitive" experience, where the robot is perceived to be expressive. Each instrument has a unique vocabulary of expressive gestures as a result of their components and construction that can be utilized as "mechatronic expressions" in a composition.

A virtuoso Hi-Hat timing and dynamics performance has been shown to contain long-range fluctuations in musical rhythms that lead to favored listening experiences [51]. By using onset detection and time series analysis of amplitude and temporal fluctuations of a performance by Jeff Porcaro's "I Keep Forgettin", both long-range correlations and short-range anti-correlations separated by a characteristic time scale in the 16^{th} note pulse were found. There were also small drifts in the 16^{th} note pulse and complex two-bar patterns in amplitudes and intervals that offered a subtle nuance to the performance.

Percussion Robotics

One of the first musical machines was called the "Panharmonicon" and it was invented in 1805 by Johann Nepomuk Malzel, who was a contemporary of Ludwig van Beethoven [52]. The massive mechanical orchestral organ illustrated in Figure 4 included percussive elements that could mimic gunfire and cannon shots. Beethoven's piece entitled "Wellington's Victory" (Op. 91) was composed with idea that it would be played on the Panharmonicon to commemorate Arthur Wellesley's victory at the Battle of Victoria in 1813. Since then there have been countless other mechanical, and later, mechatronic musical machines that included percussion instruments

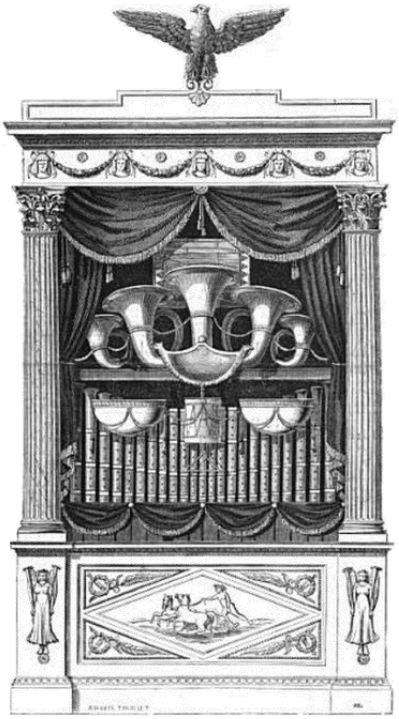


Figure 4. Panharmonicon orchestral machine.

Musical robots have used a variety of actuators that include solenoids [53, 54, 55], brushed/brushless DC motors [53, 54, 55, 56], and stepper motors [54]. These electromechanical devices offer a simple low cost solution that can be adapted to a wide variety of applications. The Machine Orchestra developed by Kapur et al. (circa 2010) used seven sophisticated and expressive percussive instruments [54]. The Machine Orchestra was developed as part of pedagogical vision to teach related technical skills while allowing human performers to interact with the instruments in real-time. The robots used a variety of actuators over a low-latency network to render unique and highly technical performances. A later version of the robotic ensemble can be seen in Figure 5.

Figure 5. The Machine Orchestra of CalArts.

The world's largest robot orchestra as shown in Figure 6 is Logos [55]. There are over 45 individual mechatronic devices in the orchestra that include organs, wind, string, and percussion instruments. Each instrument uses a musical instrument digital interface (MIDI) controller that has been tailored to control specific instrument features [1]. Precise timing and pulse width modulation (PWM) are used to control the activation and dynamics of each instrument actuator from a central point. Some of the instruments also included closed loop control for positioning and modified loud speakers to drive monophonic wind instruments.

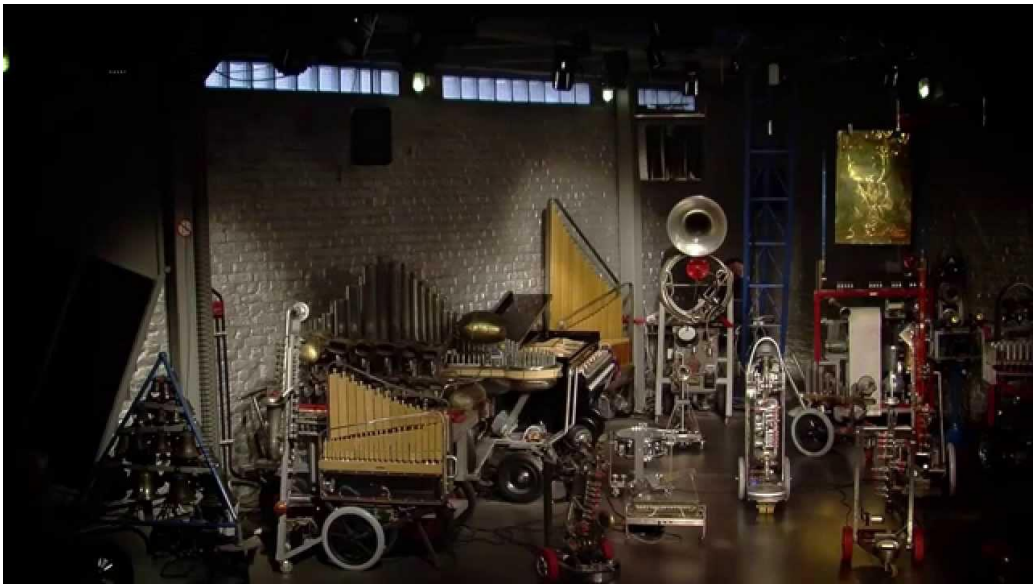


Figure 6. The Logos robot orchestra.

Extensive research and development of a percussion robot named “Haile” by Weinberg and Driscoll links a mechatronic system with improvisation in order to promote human-robot interaction [57]. The robot is designed to interpret human performances in real-time and provide an accompaniment in an improvised fashion by utilizing both auditory and visual cues. The robot was designed to embody human characteristics in terms of its form and uses a linear motor and solenoids. The left arm uses a motor and solenoid for precise closed loop positioning of a strike, which yields greater control over volume and timbre. In contrast, the right arm uses a single solenoid which can strike at a higher rate than the left arm. Each arm is controlled by a dedicated microprocessor that is directed by a networked single board computer that enables low-latency communication with a laptop computer.

Voice coil actuators were used for the improvisational robotic marimba player named “Shimon” that was developed by Hoffman and Weinberg [53]. Like Haile, this robot explored human interaction that included visual elements. As shown in Figure 7, the

robot is composed of four arms with solenoids for the striking implements and voice coils for lateral arm movement. The mechatronic marimba player was developed to explore human interaction by using machine learning to accompany human musicians [58]. In addition, the robot includes visual elements such as a bouncing head that provided feedback to its human counterparts in a hybrid band.

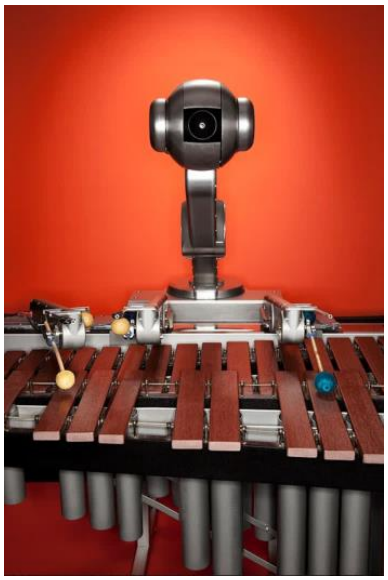


Figure 7. Robotic marimba player named "Shimon."

The world renowned jazz guitarist Pat Metheny created a studio album called Orchestrion in 2010 as show in Figure 8 with the help of Eric Singer and the League of Electronic Musical Urban Robots or LEMUR [59]. The robots orchestra was composed of a piano, marimba, string instruments, and a large array of percussion instruments that could be activated through an interface to Pat's guitar [60]. Although the process of developing music for all of the individual instruments was daunting, he found the experience to be "exhilarating, educational, and a musical journey to a new place."

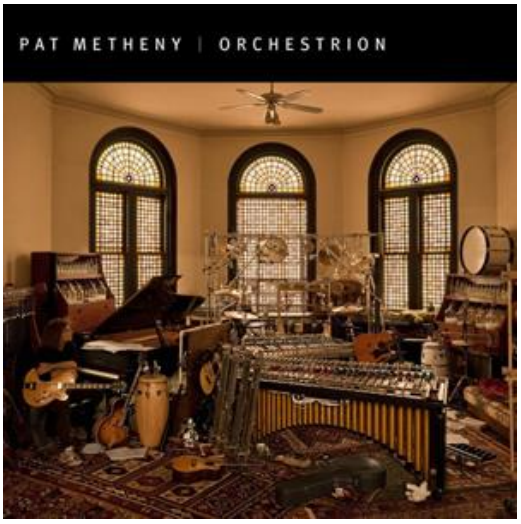


Figure 8. The Pat Metheny Orchestrion album.

Motion in the context of percussion has also been explored by developing virtual characters and using inverse kinematics, inverse kinetics, and PID closed loop control with sound synthesis [61]. Motion data from timpani performances were used to generate computer animations that not only visually illustrate the complex multi-axis motion associated with drumming, but generate synthesize sounds based on the attributes of the striking events, such as velocity. A motion database was created from multiple gesture acquisition sessions. A visual simulation was then generated from the motion data that was subsequently analyzed to derive an instrument interaction visualization and sound synthesis parameters.

Non-percussive musical robots have been developed to play instruments in a humanized way [62]. In this instance, a specialized robot was developed to play a traditional Chinese musical instrument that is similar to a dulcimer. By using an inverse kinematics control algorithm, the robot was able to perform a composition based on the contents of a human performance recorded MIDI file by striking the strings with hammers in real-time.

Non-musical Robotics

Robotic systems have used voice coil actuators as a replacement for servos, hydraulics, and other types of actuators in order to take advantage of their unique characteristics. Researchers in the MIT media lab have worked with long-travel voice coil actuators in the context of human-robot interaction [63]. The team of McBean, J. et al. designed and built a 6DOF direct-drive robotic arm using VCAs and discovered both advantages and shortcomings, but highlighted their controllability, ease of integration, geometry, biomimetic characteristics, high power capability, and low operational noise. Voice coils have also been used in a 3DOF parallel manipulator with a direct drive configuration for use in soft mechanical manipulations that includes human-robot interaction [64]. In this embodiment, positional control using optical quadrature encoders in the absence of force sensors yielded a satisfactory outcome in robot-assisted assembly and haptics.

In the medical field, voice coils have been used to precisely manipulate instruments both in dentistry and in a surgical setting. A team of researchers led by Dangxiao Wang created a miniature robotic system to manipulate a laser that prepares a tooth to receive a crown [65]. The robot uses three voice coil motors to drive the 2D pitch/yaw of a vibration mirror and protruding optical lens, which enabled high-resolution control of the laser beam. A set of experiments revealed a robotic system that delivered the level of accuracy required for dental operations with an appropriate physical size for the narrow workspace of the oral cavity. For endoscopic surgery, maintaining a desired contact force during laser endomicroscopy is important for image consistency [66]. In this context, an instrument was created using a voice coil and closed loop control to ensure a

predetermined contact force between the probe tip and tissue with compensation for involuntary operator hand movement. This technology will be integrated into endoscopic and robotic platforms in the future.

Voice coil actuators have also been used for a micro gripper, visual orientation, and pneumatic actuator force control. The research team Bang Young-bong, et al. developed a 1mm micro gripper that uses a VCA to generate linear motion with an adjustable stiffness that can also measure an externally applied force [67]. This type of gripper has applications in micro machining in the context of assembly on a micrometer scale. In the aerospace industry, a monocular visual system was developed to hold on a fixed target despite severe random perturbations from a ducted fan [68]. In this application, a voice coil actuator was used to control ocular orientation. Lastly, in order to improve agility, accuracy, and fine-motion control, a voice coil actuator was used to control single-stage flapper valves for two frictionless pneumatic cylinders [69]. A major advantage of using a voice coil over a conventional electromagnetic torque motor was the absence of any measureable hysteresis.

Sound has also been used to augment expressive robotic movement [70]. A study conducted by Dahl, et al. demonstrates a qualitative effort to map movement to sound with the goal of enhancing expressiveness as perceived by users of such robotic systems. Although a percussion robot generates sound as a by-product of its function, the mapping to movement is inherent and reinforces the concept of movement and sound being tightly coupled, and a general expectation by the observer.

Cyber-Physical Systems

This type of system controls or monitors some type of mechanism using computer-based algorithms that are tightly integrated with the internet and end users. By this definition, a robotic percussionist can be one of these systems where its foundations include linear temporal logic, continuous-time controllers, event based control, sharing of computing resources, feedback control systems, time triggered architecture, and real-time scheduling [71]. Although cyber is often used in a nefarious context, as in cyberattack, it generally implies computers and perhaps some form of virtual reality. In the context of mechatronic drummer, the virtual component is the consolidation of knowledge from gesture acquisition and expressive performances into an algorithmic representation of a human performer.

Summary

Gesture acquisition has been explored extensively in a variety of settings that include performances on percussion instruments. Whether it is in support of understanding and developing robotic instruments or controlling the parameters of a live performance, the use of real-world data connects the artist, audience, researcher, and inventor in a relatable way that results in control, observation, understanding, and innovation respectively.

What is the distance between a good performance and one that receives a standing ovation? Given the same material surely all of the correct notes were played in the context of a chord progression and rhythm, but the difference is strikingly palatable to the audience. One could argue that the best performance was the most expressive. Perhaps

the musicians were more animated or dramatic in some significant way. Maybe the arrangement was more colorful in terms of instrumentation. Although these are high-level observations, the underlying corollary is that expressive performances are preferred over flat and lifeless ones.

From a historical perspective, all of the aforementioned robotic musical instruments trace an evolution of creativity and engineering towards a common goal of rendering performances that are pleasing to both the active listener and casual observer. The research presented in this thesis builds upon these earlier breakthroughs by adding capabilities and features that have been further informed by human performances, with the objective of moving expressive robotic renderings along the continuum of artistic and technical achievement [72, 47, 73]. In the next chapter we will become acquainted with a rather simple gesture acquisition system that produced surprisingly good data that ultimately informed unique renderings and the requirements for a mechatronic drummer.

CHAPTER 3: Gesture Acquisition

In order to understand the range of human motion and the nuances of a given performance, one must acquire data directly or indirectly for analysis. Depending on the desired outcome, it may be sufficient to quantize the data to a discrete set of values that answer specific questions such as impact zones on a drum. In other cases, the sample resolution must be very high in order to extract reasonably accurate values for velocity or acceleration. In each case, one must determine the requirements of the gesture acquisition system so that it can be designed to deliver the information needed in an efficient and repeatable manner.

Acquiring real-world performance data in a non-invasive manner does impose limitations on the type of data that can be captured [35]. In some cases however, it is possible to either derive non-measureable values from measureable quantities or infer weighted correlations using calibrated references. For example, a calibrated sound pressure level (SPL) meter can be used at a fixed distance to establish a baseline reference that is synchronized with the audio recording and becomes a correlation for striking force.

In this chapter a cost-effective calibrated motion capture system will be defined that yields quality data for subsequent analysis. The discussion will include all of the details needed to reproduce the system along with sample data that is used extensively in the following chapter.

Capture system

A study conducted by A. Hajian, et al. concluded that the upper bound of the impact rate for a drum roll performed by an accomplished musician is on the order of 30Hz [74]. In this case however, the signal of interest is not the impact rate, but rather the motion that results in the impact rate. To capture the related motion sufficiently one must have a video frame rate that is "high enough" to produce reasonably smooth data [75, 76, 35].

A variety of cameras have been used for percussion motion analysis that ranged widely in cost, features, and size [75, 18, 37, 77, 61]. The camera that was selected in this instance was the GoPro HERO3+ Black Edition, which is a very versatile camera that is intended for rugged outdoor use when contained within its protective housing. The GoPro supports a variety of resolutions and frame rates that includes 848x480 at 240 frames per second. By rotating the wide angle field of view by 90° the relatively inexpensive camera can produce the desired quality and sampling rate for the motion capture system.

To understand the nuances of a human percussionist, a system was devised to capture raw video in multiple dimensions with sufficient spatial resolution that was synchronized with audio and vibration data [29]. Furthermore, it was critically important to avoid encumbering the musician and instrument with sensors and/or other equipment that could potentially influence the performance [35]. With this in mind, a specific configuration and process was created to capture and extract the motion of the tip of the striking implement along with pertinent audio and vibration data. The photograph as shown in Figure 9 documents the data-acquisition system in action during one of many recording sessions.

Figure 9. Recording session showing the participant, calibrated backdrop, lighting, and video camera.

The video data acquisition elements are composed of the snare drum, foam board backdrop, video camera, and key light as shown in Figure 10. As depicted in the profile diagram, the drumhead is tilted at an obtuse angle θ of 110°, which results in a camera relative Z-axis projection that enables depth perception. This is an important attribute of the configuration as it eliminates the need for a second camera in order to resolve drumhead region mapping. It must be noted however that tilting the drum in a specific manner has the potential to influence and/or reduce the playability of the instrument for musicians who prefer an alternate orientation.

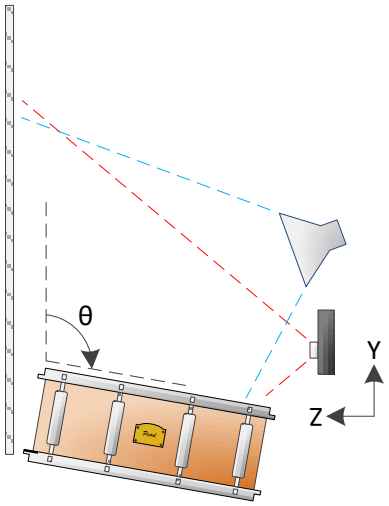


Figure 10. Diagram showing the motion capture system profile that highlights the cameras field of view and instrument angle.

The video camera has a wide angle lens and is used in portrait mode in order to utilize the full 848 vertical lines of resolution, which results in a field of view that encompasses a typical range of motion for a performance. Finally, an inexpensive project lamp outfitted with a 100W compact florescent bulb is used as the key light with the resulting view from the cameras perspective shown in Figure 11. In this frame grab, the striking elements, drum surface, and backdrop are visible. In addition, the backdrop contains a dimensional reference placard that enables video tracking software distance calibration and radial distortion compensation.

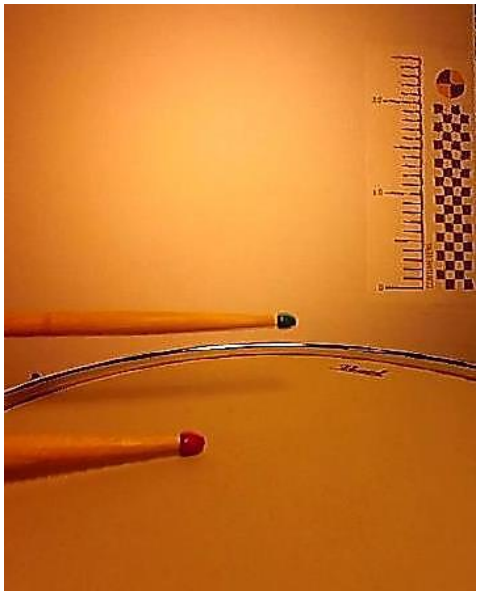


Figure 11. Camera motion capture view of striking implements and drum with calibrated backdrop.

Each tip of the striking implements was coated with an enamel paint, which in this case is green for the left stick, and red for the right stick. These colours were used by the tracking software as unique contrast to distinguish individual motion. The only caveat is that the red tip must not occlude the green tip, which can be accomplished by using a slight horizontal offset when performing, or potentially adjusting the lateral camera position with angle compensation using calibration references. Note that in some configurations the musician may be placed on the opposite side of the drum in which case the left and right sticks and their associated tip colors would be reversed. The software that was used to extract tip motion is the "Tracker" application[3] for Windows, which is a project of the Open Source Physics organization. This program provides a rich set of

[3] Detailed information or the cross-platform Tracker v4.85 application, © 2014 Douglas Brown, is available at http://bit.ly/1qWLHem.

tools for importing content, filtering video, calibrating distance, tracking motion, and extracting a variety of time referenced values.

Audio recording is naturally at the very heart of capturing a musical performance. In this instance, an industry standard Shure SM57[4] dynamic microphone was used along with an Edirol UA-25[5] (24-bit/96kHz) audio interface, which includes an integrated dual channel preamp. The audio interface was connected to a high-performance laptop running Windows 7 that was configured with the Sonar X1 Digital Audio Workstation[6] software.

In contrast to capturing variations in sound pressure through the air, recording the vibration of the drumhead can reveal other subtleties related to a given performance, such as timbre and tuning characteristics [35]. The transducer used in the audio recording system was the Roland RT-10S acoustic drum trigger[7] . Drum triggers are normally used with an electronic "brain" that interprets the analog signal in order to determine MIDI onset and velocity data for sound samples while filtering out false triggers. In this configuration however, one can benefit from the raw analog signal coming directly from the transducer, which is mechanically connected to the drumhead surface. A spectral analysis of the signal data can reveal the resonant frequency of the drum along with key harmonics that are directly influenced by the mechanical composition of the drum. The signal from the acoustic drum trigger is treated as a standard microphone input.

---

[4] Additional information on the Shure SM57 dynamic microphone can be found at http://bit.ly/1uCQMbC.

[5] Detailed specifications for the Edirol UA-25 audio interface are located at http://bit.ly/1s2yRNL.

[6] The Sonar X1 Digital Audio Workstation software is a product of Cakewalk, which produces a variety of audio production tools: http://bit.ly/1wlfPjy.

[7] Information on the Roland RT-10S acoustic drum trigger can be found at http://bit.ly/1ri2rMV.

The result of plotting a power spectrum for both the vibration transducer and dynamic microphone can be seen in Figure 12. Despite a common source, the spectral profile is quite different given wave propagation through air versus a solid material, where the latter is a complex combination of the transducer location, drumhead, and all of the physical components of the drum.

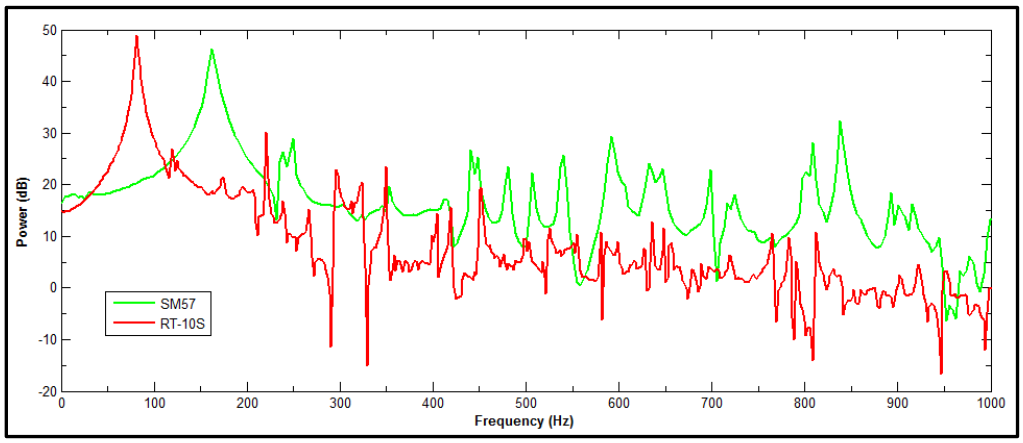


Figure 12. Power spectrum of microphone and transducer.

Calibration

Calibrating the data acquisition system is a critical step in understanding and categorizing performance data [35]. Aside from validating equipment configuration and position specifications, two of the key calibration elements are timbre and dynamics. A component of snare drum timbre is dependent on the location of drumhead impact. A performance of the score depicted in Figure 13 with motion capture enables the quantization of impact locality.

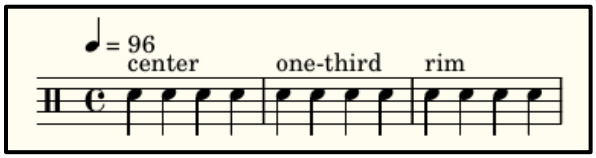


Figure 13. Timbre score composed of quarter notes per measure for three impact regions.

As illustrated in Figure 10, the obtuse angle of the drumhead in relation to the camera field of view results in a Z-axis projection. This projection is quantized into three regions, which are labeled as center, one-third, and rim [61] as illustrated in Figure 14.

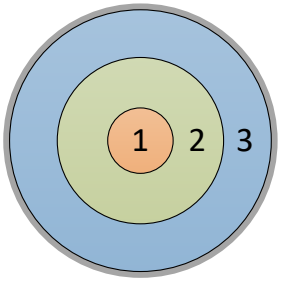


Figure 14. Strike impact regions on the drum head surface.

To identify these regions, one can process the Y-axis data from the calibration performance by subtracting the global minimum as defined in Equations 1**,** 2.

$$c = \min(Y) \quad \textbf{(1)}$$

$$z(t) = y_t - c \quad \textbf{(2)}$$

This is followed by defining static thresholds a and b that ideally delimit the impact regions. The point of impact represents a local minimum, which can be simplistically determined by evaluating a given sample with adjacent values that meet a minimum distance criterion m as depicted in Equation 3[8]. Identification is achieved by locating the point of impact and mapping the elevation value to a region based on the established thresholds as shown in Equation 4. The plot in Figure 15 demonstrates detected hits from

[8] A more robust min/max search algorithm for multivariate data should be used in practice.

a performance of the score in Figure 13 and how they were categorized as center (C), one-third (O), and rim (R).

$$h(n,t) = \begin{cases} Z_t, & Z_t < Z_{t-1} - m \,\Lambda\, Z_t < Z_{t+1} - m \\ undefined \end{cases} \quad \textbf{(3)}$$

$$c(n) = \begin{cases} 'C', & h_n < a \\ 'O', & a < h_n < b \\ 'R', & h_n > b \end{cases} \quad \textbf{(4)}$$

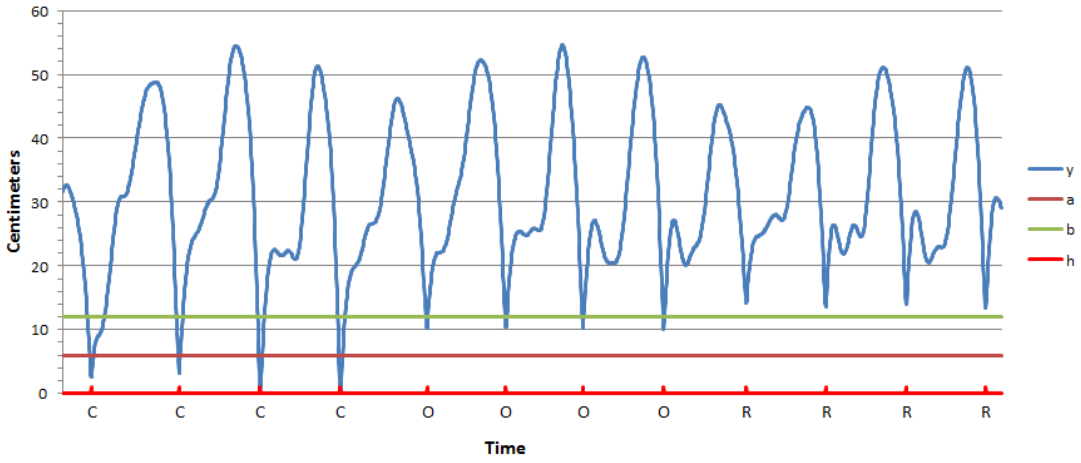


Figure 15. Timbre impact region motion plot.

Calibration of dynamics is an attempt to identify a range of sound pressure levels (SPL), from very quiet to very loud. The score in Figure 16 defines a performance that is a two measure crescendo from piano pianissimo to forte fortissimo, for which the related audio signal from a performance can be seen in Figure 7.

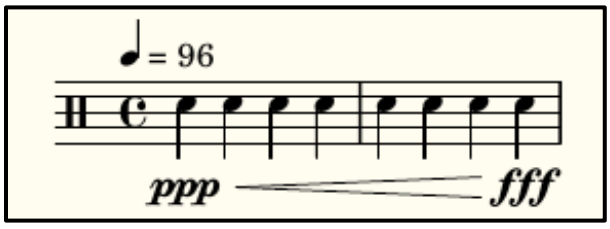


Figure 16. Dynamics score composed of eight quarter note crescendo.

As depicted in Figure 17, the peak and hold function of Equation 5 captures the successive dynamic level increase of each impact [37]. The regression curve was derived in MATLAB using a second order polynomial as shown in Equation 6 with coefficients {-0.0286, 0.3542, -0.1047}.

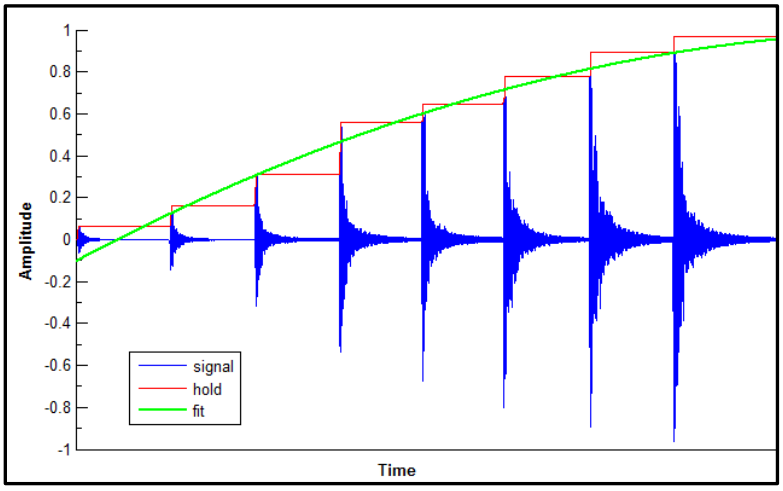


Figure 17. Normalized dynamics curve showing peak sample and polynomial curve.

$$h(x,t) = \begin{cases} x_t & h_{t-1} < x_t \\ h_{t-1} & \end{cases} \quad \textbf{(5)}$$

$$f(x) = c_1 x^2 + c_2 x + c_3 \quad \textbf{(6)}$$

Motion Data

The motion data for a performance of the "Double Stroke Open Roll" rudiment appears in Figure 18; the data is the result of an expert-level performance of the score shown in Figure 19. Referring to Figure 18, the higher-amplitude plot is the elevation-calibrated Y-axis motion of the tip of the left-hand striking implement. Conversely, the

lower-amplitude plot captures the motion of the tip of the right-hand striking implement. The entire performance spans an excess of three measures at 110 beats per minute (bpm), which amounts to approximately 6.5 s in real time along the X axis.

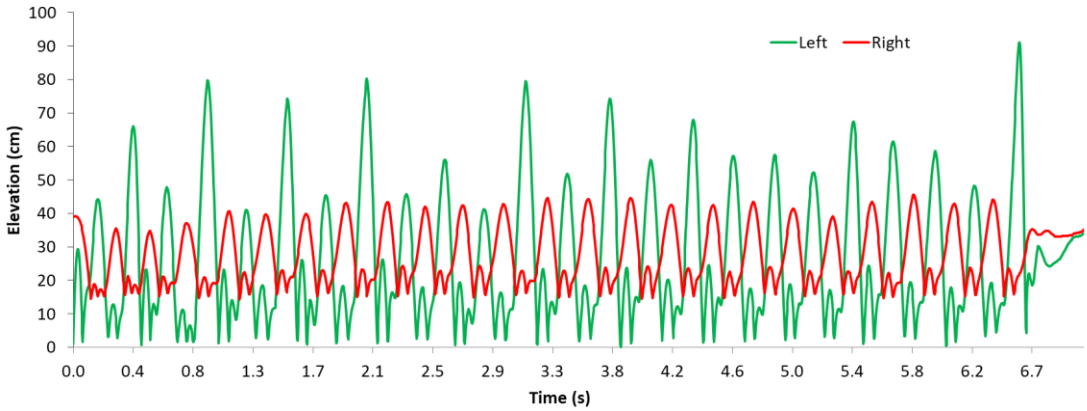


Figure 18. Double stroke open roll rudiment showing the left and right striking implement tip elevation over time.

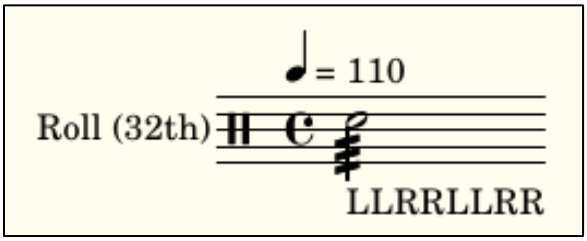


Figure 19. Double stroke open roll rudiment notation.

The graph in Figure 20 shows a magnified view of the rudiment along with correlated strike locations. At this level of detail, several triple strikes are visible; however, the magnitude of the third strike in each instance is markedly smaller given the prior peak elevation and the reduced discontinuity of the curve at the striking point. From this perspective, the third strike could be construed as gracing the drumhead in preparation for the next intentional striking event.

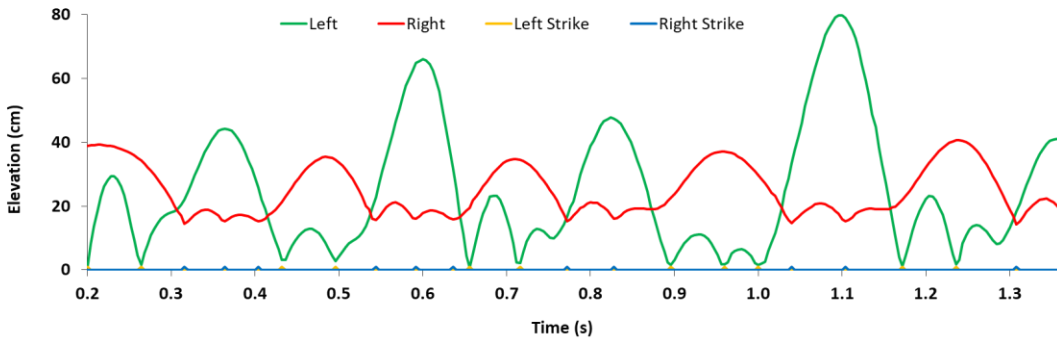


Figure 20. Double stroke open roll rudiment showing a detailed view of left and right striking implement tip elevation over time with their associated drum surface strike events.

The elevation amplitude of the tips of the left and right striking implement in Figure 18 and Figure 20 show a distinct difference that can be attributed to hand dominance [76]. In this example, the left hand is dominant and, given its variation in elevation amplitude, is far more expressive than the right hand, which has relatively constant amplitude. Since the drum surface is actually tipped towards the camera, the absolute elevation can be mapped to a Z-axis projection that provides a measurement of the X-axis position from the musician's perspective [29]. This component of motion is defined as the position for each striking implement, which has implications with respect to timbre.

Summary

In this chapter a motion capture system was defined based on the basic requirements of simplicity, low cost, and reproducibility. The resulting motion data was presented along with methods for determining the impact region and dynamics curve. This was followed by a detailed view and description of a Double Stroke Open Roll rudiment in the context of the gesture acquisition system. The following chapter will present the concept of a

stochastic model that can generate unique performance data streams based on parameters that have been derived from acquired rudiment data.

CHAPTER 4: Performer-specific Analysis of Percussion Gestures

After collecting data through the process of gesture acquisition, one must define one or more methods to analyze the resulting information in order to inform subsequent processes. By starting with an objective, such as finding the limits of motion, the analysis can proceed with mathematical operations like computing the minimum, maximum, and first derivative of motion to determine the lower and upper position bounds, and the average velocity respectively. The results of such an analysis would serve to define the minimum operating limits of a mechatronic system that would be called upon to reproduce similar gestures.

In addition to defining the operating limits of a system, one might also want to explore the concept of generative models. An example of such a model would include the ability to play a drum roll with parameters that were derived from an actual human performance. Going beyond basic timing of the strikes, it would be interesting to pursue more subtle nuances in a performance like dynamics and timbre that can help bring a traditional robotic performance to life.

In this chapter a stochastic model will be presented that can reproduce unique rudiment data streams on demand that are based on performer-specific parameters. In addition, the details of a pilot drum study will be introduced. This will be followed by the introduction of a custom software application that generates parameter vectors (performer-specific parameters) from motion data. A detailed description of timing,

velocity, and location analysis will be presented by utilizing motion data from a pilot drum study. Finally, a cursory review of each performer will be conducted that also includes a brief comparative analysis.

Stochastic Model

A compelling alternative to playing back a pre-recorded motion profile in a mechatronic system for a given rudiment is to create a parameter-driven model that can create a unique performance on demand. This approach presents an opportunity to identify a set of parameters that, when coupled with a stochastic model, results in a unique and "human-like" rendering. Given a standard set of rudiments and the infinite variety of other potential articulations, it is important to create a plug-in model that is extensible and tuneable, as illustrated in Figure 21. In this conceptual drawing, a collection of models can exist within a system that, when triggered, generates a stream of onset-, velocity-, and position-selection controls based on the current tempo and set of related parameters.

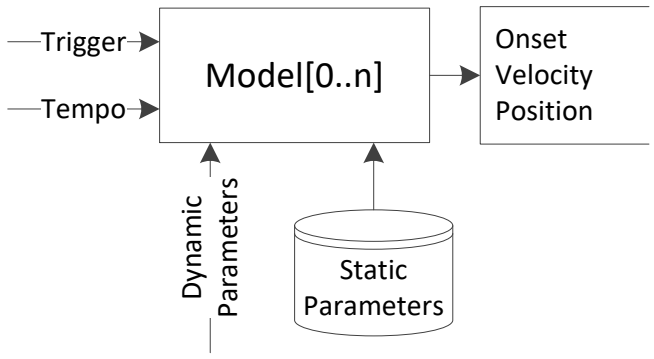


Figure 21. Diagram showing an external trigger, tempo value, and dynamic/static parameters driving stochastic model that generates an onset, velocity, and position tuple.

In this context, the trigger can for example, be a MIDI “note on” event, an Open Sound Control (OSC) message, or a physical switch. When a particular model is selected as the “patch” and then triggered, the resulting series of onset, velocity, and position values must be subsequently rendered by a robotic or synthesis system. With regard to the former, a mechatronic implementation must faithfully reproduce the striking-implement motion that results in the directed sound without introducing significant additional multi-dimensional noise. The combination of actuators, mechanical coupling, and electrical timing will certainly introduce some level of noise; however, great care should be taken to minimize this component in order to render an accurate performance.

Setup of Study

A pilot drum study was conducted at the University of Victoria in order to gain a deeper understanding of human motion in the context of percussion performances [31, 78]. A total of four experienced percussionist participants were selected to participate in the study. The participants were asked to perform a series of identical performances that were recorded using a commercial-off-the-shelf (COTS) high frame rate video camera as discussed in the Gesture Acquisition chapter [29]. The performances included a dynamics exercise, a double stroke open roll rudiment at 60 and 110 beats per minute (bpm), a total of 39 additional rudiments as defined by the Percussive Arts Society, and an optional short form musical piece entitled “Shostakovic 10 mvt II [21].” All of the rudiment performances were recorded with a reference click track to enable precision timing analysis. The motion data was then analyzed using a custom application that was

developed to extract the mean and standard deviation of the timing, velocity, and strike locations for individual and comparative analysis.

By analysing the onset, position, and velocity of each strike over time for a given rudiment, one can begin to derive a set of parameters for a performance composed of stochastic and intentional micro-timing components. As the musicians performed the piece, they were interpreting the score in the context of a background click track as heard in a pair of monitor headphones. This absolute timing reference provides a means to evaluate each strike in relation to the beat and score. Tendencies to lead or lag the beat were discovered from both a statistical and temporal perspective.

With the drum tilt resulting in a Z-axis projection, the tips of the left- and right-hand striking-implement positions can be evaluated over time [29]. Representing the timbre of the strike, this data reveals some of the frequency spectrum components that make a performance unique. As with onset timing, statistical distributions and intention can provide insight into spectral diversity.

Finally, the dynamics of a given performance is proportional to the velocity of the striking implement just prior to impact. As a result, accents become evident on specific beats, which can be stylistic in nature. An example of this would be an emphasis on beat one and three of a four-beat measure, which is common in western music.

As an internationally recognized standard compiled by the Percussive Arts Society, the 40 rudiments in Table 1 represents a “vocabulary for contemporary percussionists [21].” With a stated goal of capturing a standardized set of recordings that offer rich nuanced performance opportunities, the rudiments provided a well-documented and natural choice. In addition to the rudiments, calibration sequences were recorded for

striking implement position and sound pressure level, which are also included in the dataset.

Table 1. Percussive Arts Society rudiments.

1	Single Stroke Roll	21	Flam Accent
2	Single Stroke Four	22	Flam Tap
3	Single Stroke Seven	23	Flamacue
4	Multiple Bounds Roll	24	Flam Paradiddle
5	Triple Stroke Roll	25	Single Flammed Mill
6	Double Stroke Open Roll	26	Flam Paradiddle-diddle
7	Five Stroke Roll	27	Pataflafla
8	Six Stroke Roll	28	Swiss Army Triplet
9	Seven Stroke Roll	29	Inverted Flam Tap
10	Nine Stroke Roll	30	Flam Drag
11	Ten Stroke Roll	31	Drag
12	Eleven Stroke Roll	32	Single Drag Tap
13	Thirteen Stroke Roll	33	Double Drag Tap
14	Fifteen Stroke Roll	34	Lesson 25
15	Seventeen Stroke Roll	35	Single Dragadiddle
16	Single Paradiddle	36	Drag Paradiddle #1
17	Double Paradiddle	37	Drag Paradiddle #2
18	Triple Paradiddle	38	Single Ratamacue
19	Single Paradiddle-diddle	39	Double Ratamacue

20	Flam	40	Triple Ratamacue

A methodology was described that is associated with capturing and processing multi-dimensional performance data, which is composed of video, audio, transducer data, and annotated temporal position information. This process was used for each rudiment in Table 1, which resulted in a large collection of files. Each file in the data set can be reconciled by file name. The naming convention is of the form: "<device> <rudiment> <bpm> [left | right].{mp4, wav, csv}." The videos are in MPEG4 format with a resolution of 848x480 at 240 frames per second. The audio and transducer tracks were recorded at 16-bit/44.1Khz. Finally, the annotated temporal position information was archived into comma separated value (CSV) files for simple import into MATLAB, Excel, and many other environments given its ubiquity.

In reference to the first rudiment list in Table 1, a "Single Stroke Roll" frame capture of the video, shown in Figure 22, depicts the first strike of the left striking implement. With regard to the camera coordinate systems, the Y axis is up and down, the X axis is left and right, and the Z axis is in and out. Note that the red tip implement was accidentally used for the left hand in the example photo, which resulted in re-recording the session in order to follow the established norm of using the green tip for the left hand.

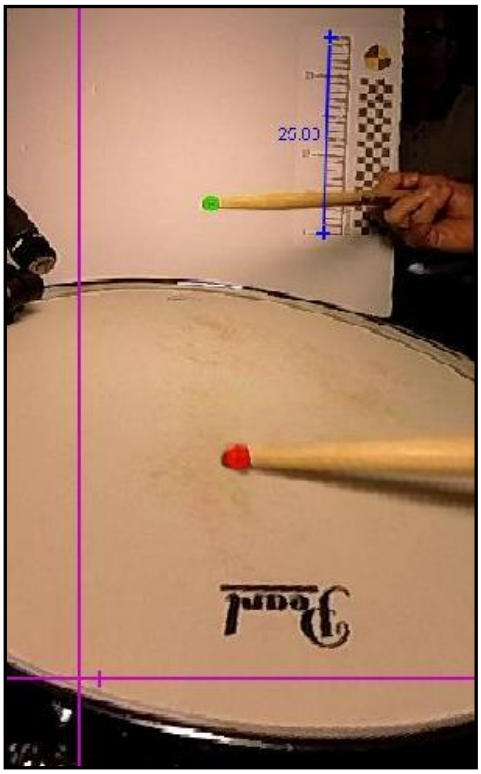


Figure 22. Motion capture frame in the Tracker application.

A partial plot of the left and right annotations for the first rudiment appears in Figure 23. An offset representing the minimum of the left (green) striking implement has been subtracted from both signals in order to show the relative positions. This is illustrated by the right (red) striking implement, whose minimum is approximately 10cm above the left minimum.

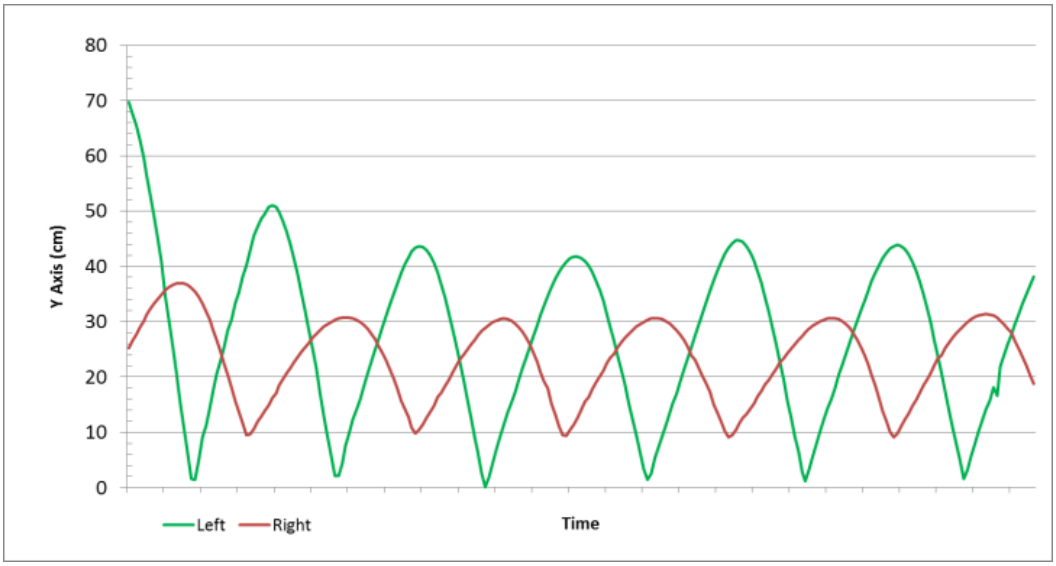


Figure 23. Single stroke roll left and right striking implement plot

Analysis Application

The screenshot in Figure 24 shows a custom C++ application that was developed by using the JUCE[9] framework to visualize, inspect, and auto-generate a parameter vector for the stochastic model. The RGen (Rudiment Generator) application allows the user to load independent motion data for the left and right striking implements along with a score for the rudiment, specifications for the bpm, and the motion data sample rate (fps). The waveforms can be visually inspected by using zoom and pan sliders. The editor group includes dedicated sliders that drive a real-time signal processing algorithm to auto-detect the onset, velocity, and position of impact events, as indicated by the crosshair symbols in the plots in this example. Each individual impact event can be manually inspected and/or edited by using the timestamp, velocity, elevation, and implement fields.

[9] JUCE is a partially open-source cross-platform C++ application framework that is used primarily for GUI and audio plug-in development. Detailed information on the framework is available at https://juce.com.

Additionally, the user can use the right-click button feature on the mouse to manually add or delete events in the waveform display area. The score slider skews the position of the measures in time for proper alignment with the performance data. The audition, load, and save buttons allow the user to render the rudiment by using a general MIDI snare drum synthesis, loading a previous session, or saving the current session, respectively. The graphical display includes stats at the top on the left and right motion data that also shows the number of theoretical strikes as defined by the score over the detected strikes, which must be the same and determined to be "locked" to create a parameter vector. The parameter vector group uses a set of sliders to scale the magnitude of the onset time, velocity, and elevation values with buttons for auditioning and loading (saving) parameters from (to) a file. Finally, the MIDI group allows the user to generate unique performances by specifying the number of measures, bpm, and general MIDI instrument type before pressing the export button, which creates a standard MIDI file. The user may optionally review the auto-generated MIDI performance by pressing the audition button before using it with other devices, such as a DAW with a synthesis instrument, drum machine, or robotic percussionist.

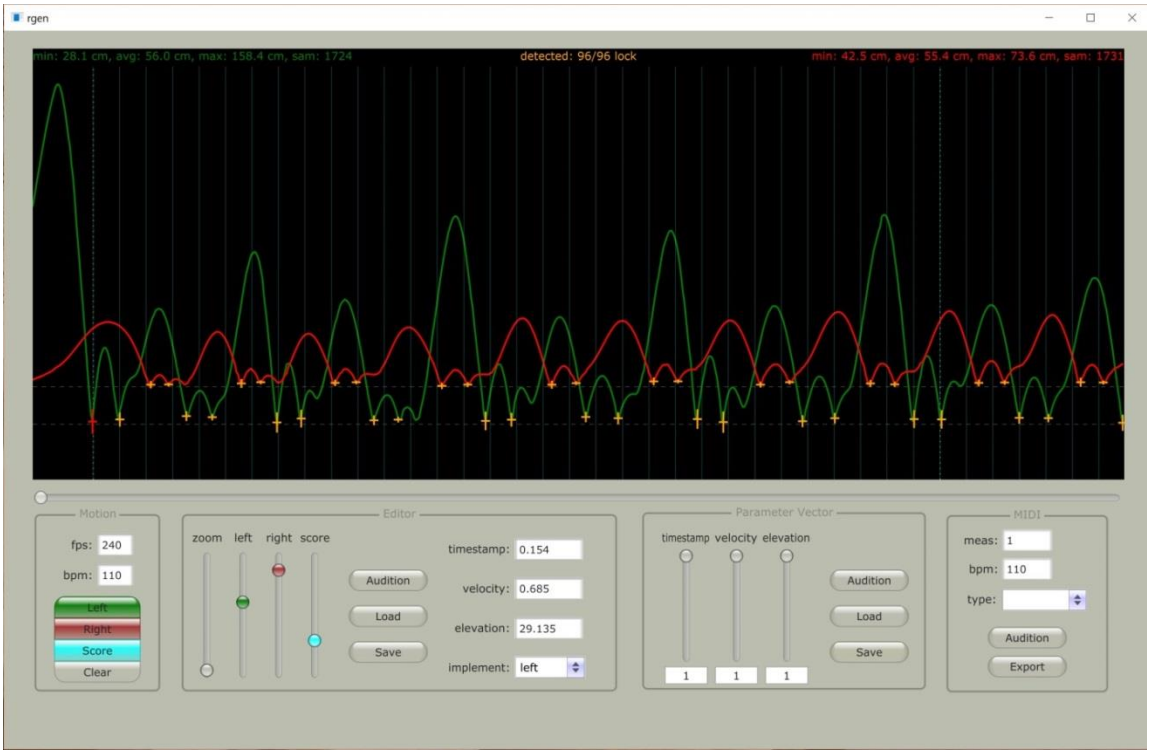


Figure 24. Custom application that extracts the onset, velocity, and elevation parameter vector from recorded motion data.

Parameter Vector

By incorporating the statistical measures for each strike in terms of timing, position, and velocity, one can create a parameter vector that generally describes the rudiment performance. Although a single measure of a Double Stroke Open Roll contains 32 notes, an example of the first eight notes is shown in Table 2. The entries in the parameter vector are used to generate a unique strike by using the parameter tuple that consists of onset, velocity, and position.

Table 2. Parameter vector example.

Note	Onset (μ sec)	Onset (σ sec)	Velocity (μ cm/s)	Velocity (σ cm/s)	Position (μ cm)	Position (σ cm)	Hand
1	-0.018	0.004	-1324.101	131.453	29.318	0.161	L
2	0.053	0.004	-1029.712	112.868	29.799	0.053	L
3	0.118	0.004	-515.806	66.100	42.962	0.341	R
4	0.197	0.004	-227.061	80.203	43.666	0.323	R
5	0.252	0.004	-761.860	26.731	30.598	0.510	L
6	0.321	0.004	-620.436	111.393	30.669	0.150	L
7	0.379	0.005	-424.778	136.845	44.004	0.250	R
8	0.457	0.005	-226.775	77.906	44.024	0.099	R

Equation 7 is used to calculate each onset mean value in the parameter vector as shown in Table 2:

$$\mu = o_i + \frac{t_r}{t_d}\left[\frac{1}{n}\sum_{0}^{n}(o_n - o_i)\right]. \quad \textbf{(7)}$$

Because a rudiment is captured at a specific beats per minute, a scaling factor can also be applied to the onset time to translate it to any desired tempo; however it should be noted that the parameters will not shift linearly in the same way that an individual's gate changes when they transition from walking, to jogging, and finally running. This implies that rudiments should be captured and analysed for discrete ranges where linear scaling generally remains intact. The variables of Equation 7 are defined as follows: o_i =

expected note onset time, t_r = reference tempo in bpm, t_d = desired tempo in bpm, n = measure aligned note samples, and o_n= measure aligned onset note time.

The plot in Figure 25 shows the result of computing the onset by using a normal distribution with a specified mean and standard deviation for each strike in the measure as directed by the partial parameter vector depicted in Table 2. The solid straight line (yellow) indicates the precise metrical position for each note in the score.

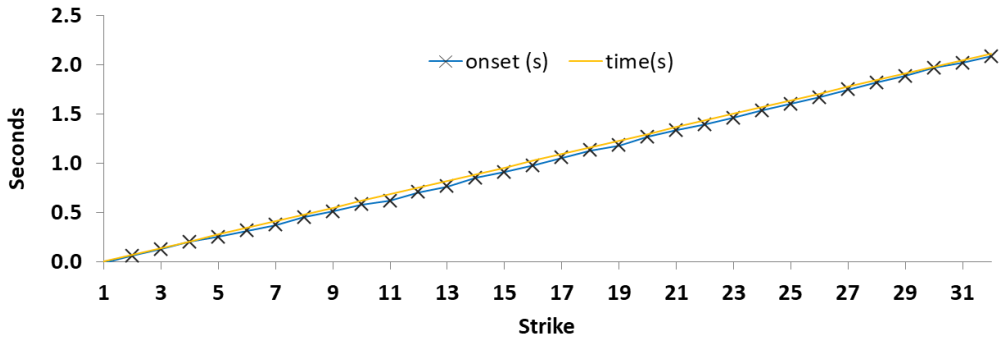


Figure 25. Onset sample in comparison to literal performance for Double Stroke Open Roll rudiment that was performed by participant #1.

By introducing the concept of a normal distribution derived from a live performance, one can see the resulting effect on the sample implement strike plot in the onset line [79]. By using Equation 8 with the data plotted in Figure 18, a computation of each unique onset sample element given the related parameters in the parameter vector is shown in Table 2. The equation returns an onset sample by using the probability density function for a normal distribution as directed by the mean μ and standard deviation σ arguments, along with a global drift sinusoid term, which is scaled by the average peak magnitude A.

The remaining variables for Equation 8 are defined as o_i = expected note onset time, t_r = reference tempo in bpm, and f_d = drift frequency:

$$o_s = N(\mu, \sigma) + A\sin\left[\frac{o_i t_r f_d \pi}{240}\right]. \quad \textbf{(8)}$$

In a similar fashion, the position of each note can be rendered as depicted in Figure 26 by using the normal-distribution term of Equation 8 with the related mean and standard deviation parameters in the partial parameter vector of Table 2.

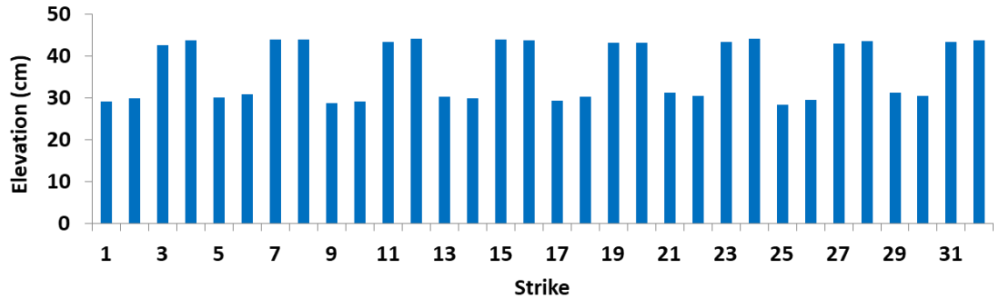


Figure 26. Position sample showing the variability in strike locations for Double Stroke Open Roll rudiment that was performed by participant #1.

Lastly, the velocity including all of the individual accents can be generated as shown in Figure 27 by using the normal-distribution portion of Equation 8 with the related mean and standard deviation for each note in Table 2. The velocity sample illustrated in Figure 27 also demonstrates the preservation of accent notes, which not surprisingly occur on quarter note boundaries {1, 9, 17, 25}.

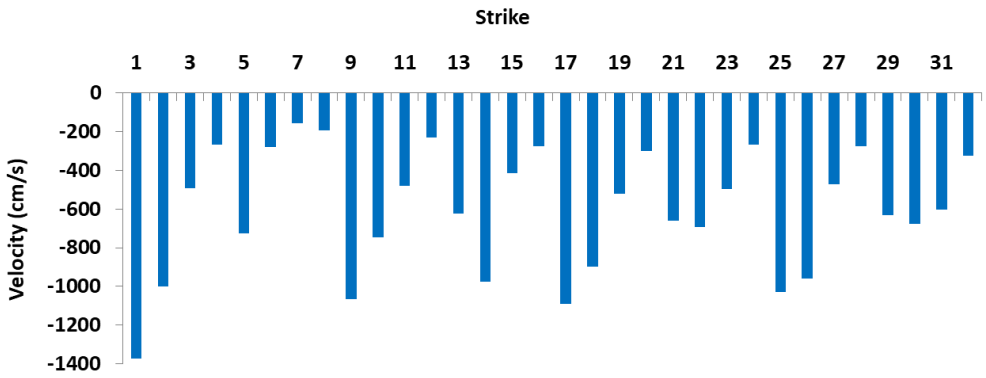


Figure 27. Velocity sample showing the intentional and nuanced strike dynamics for Double Stroke Open Roll rudiment that was performed by participant #1.

A software application was developed to generate N measures of a given rudiment as a MIDI file. For the Double Stroke Open Roll rudiment, the stochastic model used a 32 note parameter vector, as partially depicted in Table 2, with the exclusion of the position or hand parameters, since MIDI "note on" events are restricted to timing and velocity values only. The resulting MIDI event timing (horizontal axis) and velocity (height of bar) of the first eight notes for a set of variants can be seen in the Sonar Producer X3 DAW screen capture of Figure 28. Top to bottom, the panels show four single-bar renderings labelled literal, dynamics, timing, and timing with dynamics.

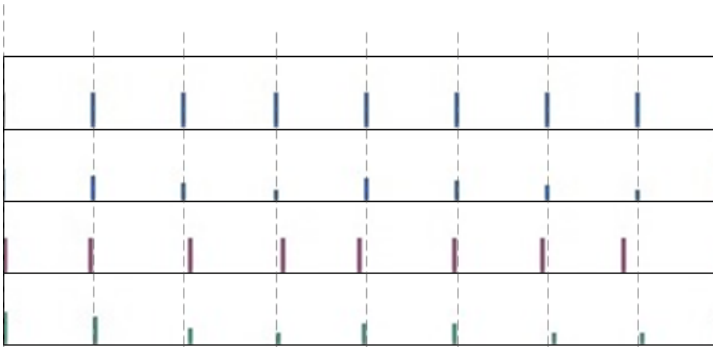


Figure 28. Renderings of literal, dynamics, timing, and timing/dynamics performances.

When the literal performance was played through a snare drum sample, the result was predictably cold and completely devoid of any emotion. In fact, it sounded more like a machine gun! In contrast, when the stochastic model variants were used with the same snare sample, the rendering sounded much more human. By adding pattern and stochastic elements to the velocity in addition to timing, the performance came to life in a way that clearly demonstrated the value of a model that uses multiple dimensions and whose basic parameters are derived from a real performance. The reader is encouraged to listen to the original recording and renderings by visiting the companion website at http://tinyurl.com/gu2wwlx.

Timing

From a timing perspective, the onset mean and standard deviation were derived by comparing the strike event with an ideal score at the same bpm using note alignment as shown in Figure 24. Individual performance analysis yielded mean and standard deviation values for each note that were understandably unique; however they were also generally comparable as shown in Figure 34 and Figure 38 respectively. The data suggested both individual and generalized parameters that could be used in a stochastic model to reproduce comparable performances in an autonomous fashion.

At 110 bpm, a Double Stroke Open Roll rudiment, which is composed of 32nd notes, should have a strike every 68ms. By aligning the first strike of the performance with metrical time zero and measuring the delta of subsequent strikes with expected strikes, a plot was derived as shown in Figure 29. As can be seen from the first set of strikes, the

player is attempting to synchronize with the beat; however, a progressive lead is evident with a peak of nearly 100ms. As time goes on, however, one can see that the tempo drifts towards zero, followed by an arc that lags the beat with a peak of 80ms. This artifact provides evidence of a low-frequency oscillation in onset timing as the musician chases the beat. The smooth trend line in Figure 29 highlights the general shape of the onset drift over three measures.

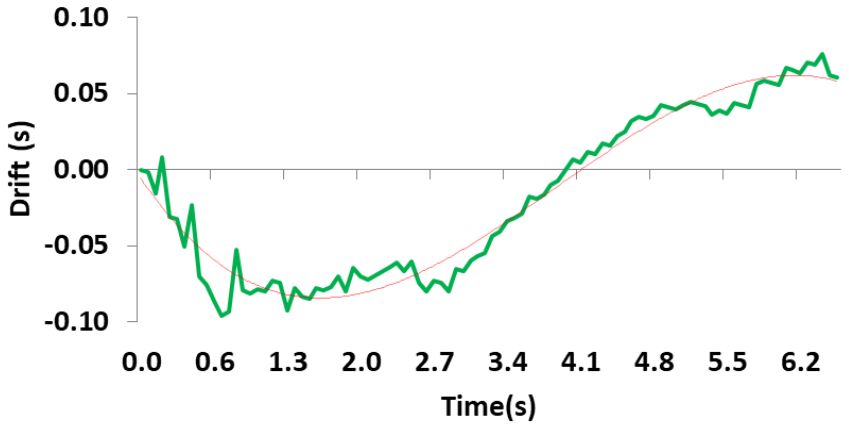


Figure 29. Tempo drift signal derived from onset time versus literal time for the Double Stroke Open Roll rudiment performed by participant #1.

Incorporating the tempo drift data into a histogram, as shown in Figure 30, reveals a bimodal distribution with a stronger tendency towards leading the beat. Although the sample points are insufficient to draw any generalized conclusions about tempo drift, they do illustrate a potential periodic behaviour that, on the surface, seems reasonable.

By applying Equation 9 over three measures of drift data, the mean tempo μ delta is determined to be -19ms. A standard deviation σ of 54ms for the Double Stroke Open Roll rudiment performed by participant #1 is calculated by using Equation 10:

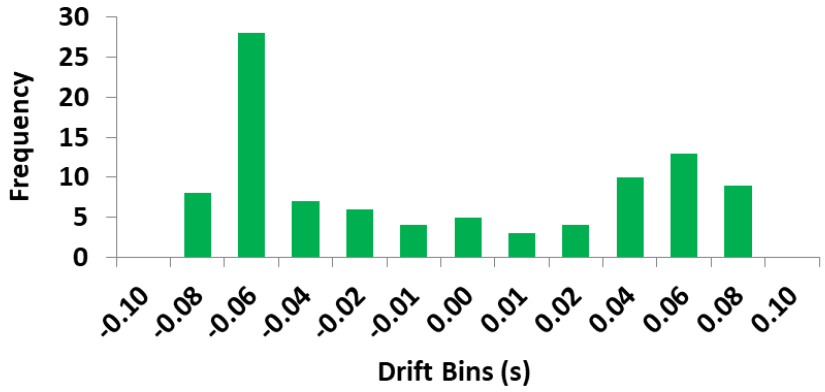


Figure 30. Tempo drift histogram showing bimodal tendency for Double Stroke Open Roll rudiment that was performed by participant #1.

$$\mu = \frac{1}{n}\sum_{i=0}^{n} d_i, \tag{9}$$

$$\sigma = \sqrt{\frac{\sum(d_i - \mu)^2}{n}}. \tag{10}$$

To preserve the association of the strike parameters, it is important to use the average and standard deviation of the same note in each successive bar as opposed to using the contiguous drift data. A notable side effect of this approach is the decimation of the tempo drift curve, as shown in Figure 29. However, the curve can be re-created by adding to onset events a global sinusoidal oscillation of the appropriate frequency and amplitude, which for the plot shown in Figure 29 is 0.12Hz with an average peak magnitude of 0.086. Note that vastly different macro-timing characteristics can exist among a group of musicians, and even within a series of performances by the same musician.

In addition to many other attributes, it is possible to detect the onset time for each strike as shown in Figure 31 using transient selection mode in Sonar X1. The recording of

eighth note strikes (red onset) illustrates instances of both leading and lagging the reference beat (vertical black bands) at 96bpm. By examining the delta between actual onset time and meter, statistical metrics can be computed for the performance. In this case, the median delta is -33ms, which indicates a bias towards leading the beat. The tendency towards leading is clearly visible in Figure 31 when comparing onsets to the established meter.

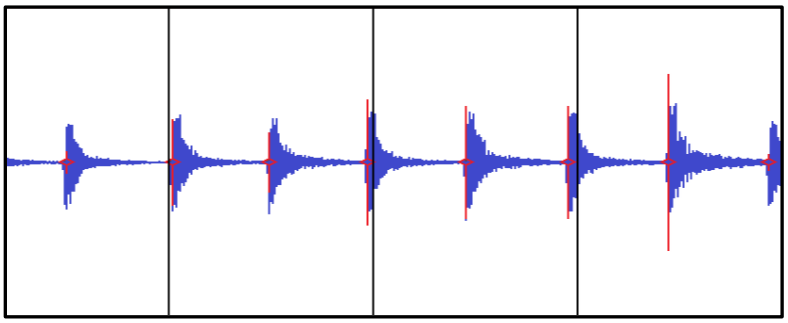


Figure 31. Visible strike drift when compared to metrical time.

Velocity

The velocity of each strike represents a profile in dynamics that tended to follow standard patterns in western music in terms of emphasis on quarter note beats within a measure as illustrated in Figure 27. The double stroke open roll rudiment is composed of 32 individual strikes in a measure using a repeating “LL, RR” striking implement pattern. A statistical analysis of the velocity data spanning several measures resulted in a mean and standard deviation for each individual strike that can be used as parameters to drive a stochastic model in a similar manner as the aforementioned timing parameters.

With respect to dynamics, an average velocity of −600 cm/s has the highest incident frequency for the left hand in Figure 32, whereas values near −200 cm/s are more common for the right hand. A statistical analysis of the respective histograms using an

absolute value reveals a mean of 844 cm/s and a standard deviation of 240 for the left hand, and a mean of 379 cm/s and standard deviation of 130 for the right hand. This again provides a consistent story that the left-hand striking implement is far more expressive than that of the right hand in terms of amplitude and range for this participant.

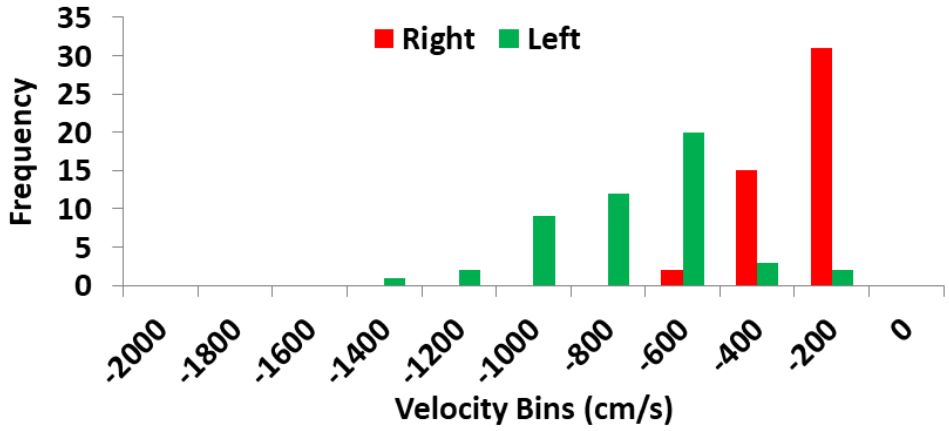


Figure 32. Velocity histogram showing frequency and magnitude for Double Stroke Open Roll rudiment that was performed by participant #1.

As was the case with position, the velocity histograms indicate substantial skews. The left-hand skew using Equation 11 for the data plotted in Figure 11 is −0.541 and the skew for the right-hand velocity histogram is −0.202. Compared with the position histograms, the velocity skewness is much stronger given the emphasis on accented strikes. It is also interesting to note that the left-hand skew is greater than that of the right hand. This again correlates with the discovery of a dominant left hand where magnitude, range, and intentional micro-timing are greater than for the right hand.

Location

The location of impact on a snare drum plays a significant role in timbre [80]. A strike in the middle of the drum has a characteristically lower frequency and shorter dampening time compared with a strike towards the edge of the drum. This is the result of a complex multi-modal vibration between the drum heads, shell, and snare [81]. Research conducted by Tindale et al. has also shown that striking locations result in distinct timbres that can be classified by training artificial neural networks [82].

A histogram of strike locations shown in Figure 33 reveals a mean of 30.05cm for the left hand as compared with 43.6cm for the right hand when referenced to the lower rim of the drum that is nearest to the camera. This separation is intuitive given the Z-axis projection towards the camera, because the right hand is at a greater distance and thus a higher elevation on the tilted surface. Aside from the relative positions of the tips of the left and right striking implements, the higher standard deviation of 0.85 for the left hand compared to 0.52 for the right hand shows that the potential spectrum range is larger for the dominant hand.

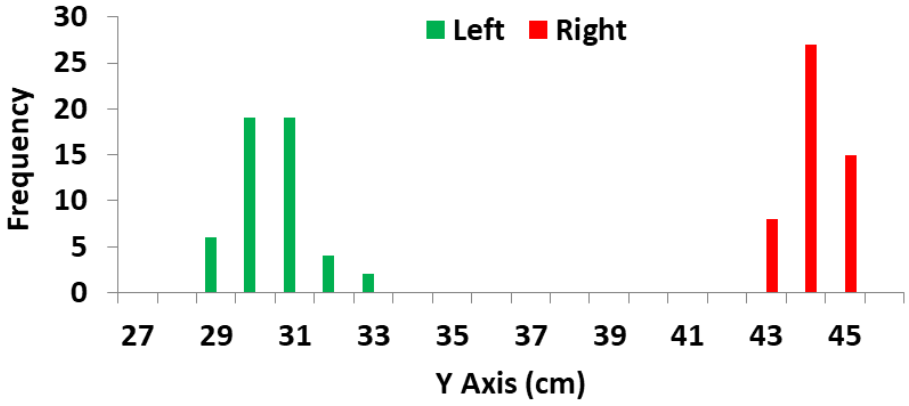


Figure 33. Hand position histogram showing frequency and locations for Double Stroke Open Roll rudiment that was performed by participant #1.

The histograms of both the left- and right-hand positions have significant skew of 0.171 and −0.174, respectively, when using Equation 11 on the X-axis motion data. The lack of symmetry strongly suggests an intentional component that coexists with stochastic processes:

$$\text{skew} = \frac{n}{(n-1)(n-2)} \sum \left(\frac{d_i - \mu}{\sigma} \right)^3 . \quad \textbf{(11)}$$

Performer Analysis

To gain a deeper understanding of performance attributes across a group of percussionists, a pilot study following a standard ethics protocol was conducted at the University of Victoria. A total of three participants were included in the study with data-collection methods and analysis being identical to the initial participant data discussed at the outset of this thesis. The primary attributes of each participant such as years of experience and grip are shown in Table 3.

Table 3. Drum study participants attributes.

Participant	Years of experience	Classically trained	Dominant hand	Grip	Genres
1	40	Yes	Left	Traditional	Orchestral
2	20	Yes	Right	Matched	Orchestral, Jazz
3	13	Yes	Right	American	Orchestral
4	30	No	Right	American	Jazz

After collecting and extracting motion data, the RGen application was used to derive a parameter vector, which is illustrated graphically in Figure 34 for participant #1. The ascending onset line shows the mean onset timing as the musician performed 32 strikes in a Double Stroke Open Roll rudiment at 110bpm. The oscillation in this line illustrates the concept of "chasing the beat" as discussed earlier [83]. The onset standard deviation error bars indicate that the performer was 'loose' with regard to strike timing. The mean velocity bars show a pattern of emphasis on quarter note boundaries and the associated standard deviation error bars imply a wider range of dynamics for some equivalent note positions across several measures. Finally, the individual elevation strikes demonstrates both the separation and subtle variance in strike position on the drum surface that lead to changes in timbre throughout the performance.

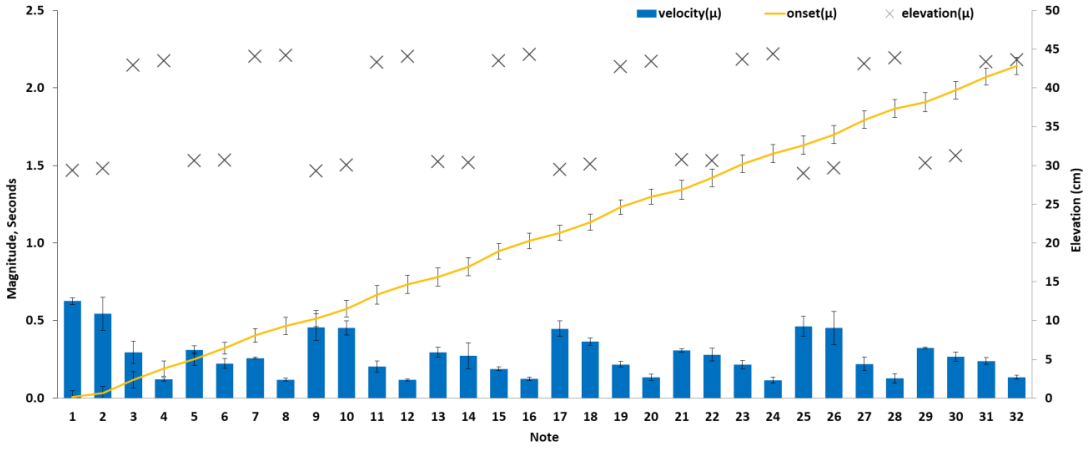


Figure 34. Parameter vector plot showing the mean and standard deviation of onset, velocity, and elevation parameters for participant #1.

The parameter vector plot shown in Figure 35 for participant #2 indicates good onset timing in reference to the click track given the negligible visible oscillation. Further, the small onset standard deviation error bars illustrate a performance that was remarkably consistent across multiple measures. The velocity and related error bars shows a distinct four note dynamic pattern that follows “LL, RR” strikes sequences as opposed to metrical positions. The strike elevations show a relatively large separation and the leading strike is on the right-hand instead of the left-hand.

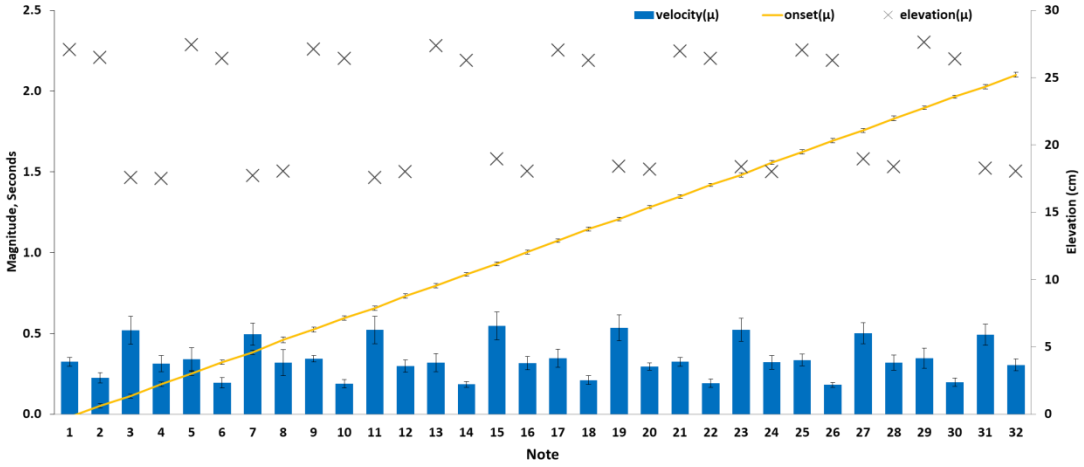


Figure 35. Parameter vector plot showing the mean and standard deviation of onset, velocity, and elevation parameters for participant #2.

For participant #3, the parameter vector plot in Figure 36 indicates excellent onset timing with respect to the click track. In contrast to the other participants in the study, the onset standard deviation error bars are exceptionally small, which highlights the musician's ability to maintain a consistent roll over several measures. Although there is a basic four note dynamic pattern associated with "LL, RR" strikes, there is evidence of accents on quarter note metrical boundaries. With regard to elevation, the proximity of the left and right strikes is close, which implies a timbre that is unique to this performance. As was the case with other participants, the musician started the measure with their right-hand.

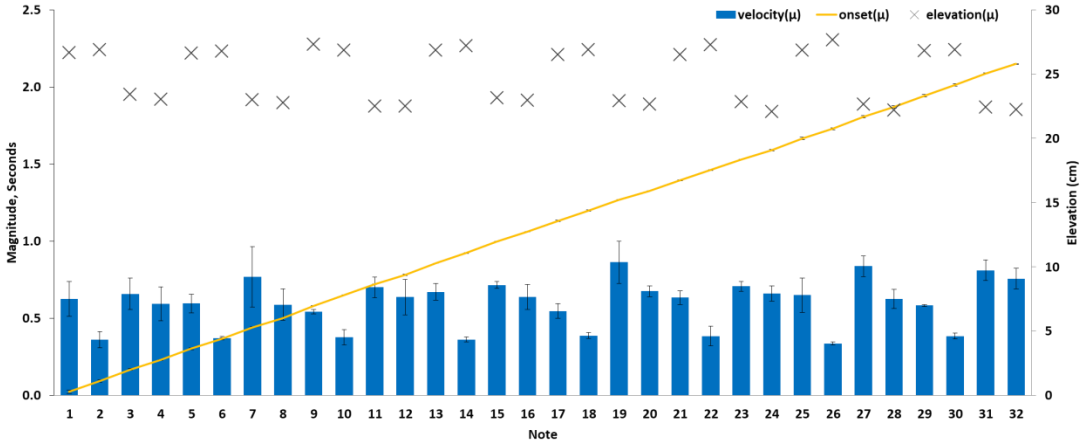


Figure 36. Parameter vector plot showing the mean and standard deviation of onset, velocity, and elevation parameters for participant #3.

The performance of participant #4 as shown in Figure 37 is nearly linear in regard to onset timing, which indicates tight tracking of the click track. Onset standard deviation error bars are very small and thus the rudiment is consistent from measure to measure. The velocity and related error bars show the familiar four note pattern with little evidence of metrical accents or large dynamic variation. Lastly, the strike locations are extremely tight between the left and right hand as compared to the other participants in the study.

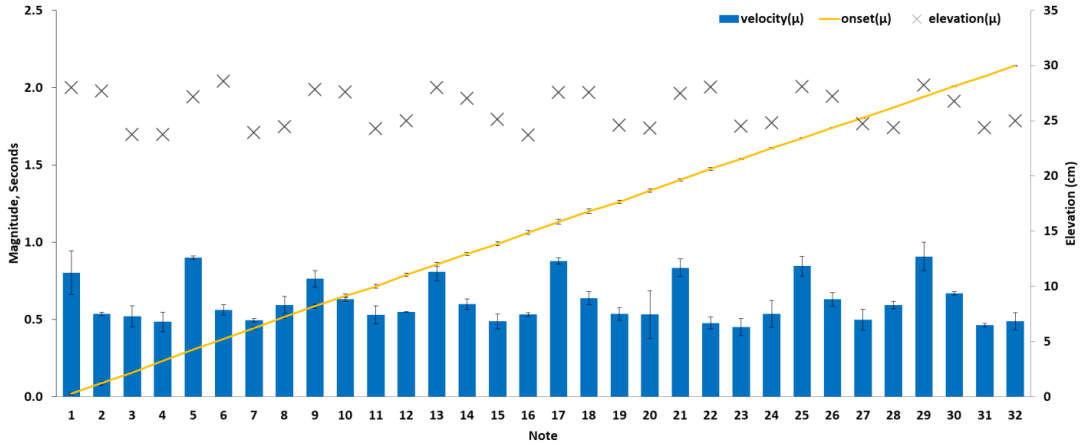


Figure 37. Parameter vector plot showing the mean and standard deviation of onset, velocity, and elevation parameters for participant #4.

Comparative Analysis

A comparison of parameter vectors across the participants in the drum study as shown in Figure 38 highlight both the generalized and nuanced performances of each participant. Participants 2–4 have an average onset standard deviation that is significantly less than that of participant #1, who shows an inclination for loose timing with the click track during the recording. In contrast, the mean velocity and related average standard deviation bars demonstrate that participants #3 and #4 were much more aggressive in their style in terms of dynamics as compared with the other participants. The minimal average standard deviation of the elevation indicates a general trend towards a small regional position on the drum head for both hands that led to a rather consistent timbre. The mean elevation however shows that each participant opted to strike the surface in a unique region with characteristic separation, ostensibly to acquire a specific sound from the drum or to strive for a comfortable position based on their individual performance styles.

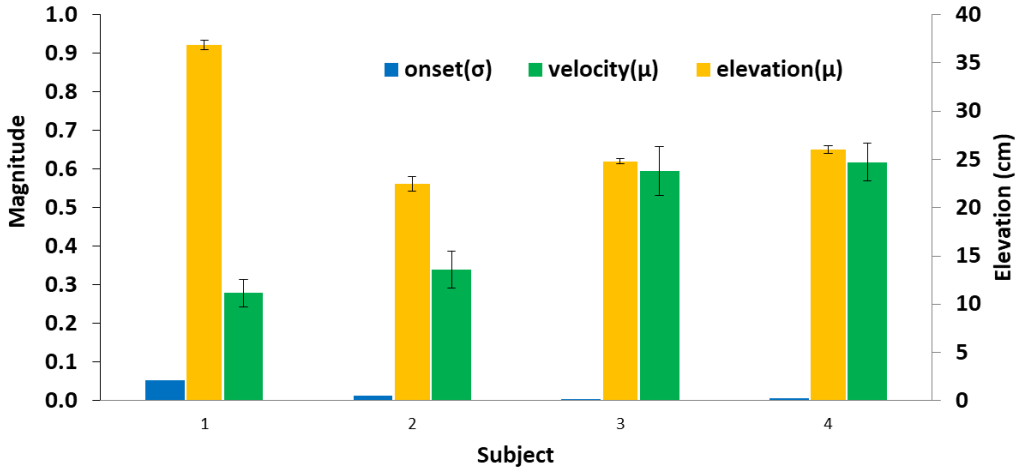


Figure 38. Performance comparison of onset standard deviation, mean velocity, and mean elevation with related standard deviation error bars.

Summary

This chapter began with a description of a stochastic model that is capable of generating unique performances on demand. After a review of the pilot drum study that was conducted at the University of Victoria, an analysis application was presented that greatly aids in the creation of parameter vectors for the stochastic model. Next, a parameter vector was described for use with a stochastic model. This was followed by a detailed analysis of timing, velocity, and location information for participant #1. Lastly, a cursory observation was provided for each additional participant along with a comparative analysis. In the next chapter we will take a deep dive into voice coil actuators, which serve as the motion generators in the mechatronic system. This will be followed by a high-level design walkthrough of the mechanical, electronic, and software subsystems.

CHAPTER 5: Mechatronic System

System design is driven by a set of requirements that are derived from the feature set of the device to be manufactured. From electric toothbrushes to lunar landers, each device has capabilities that have been tailored for the tasks it must undertake within its intended operating environments. The process of defining the feature set and distilling it down to individual requirements can be a long and daunting one, but it must be methodical in order to ensure complete coverage while avoiding unnecessary elements that can inadvertently add complexity.

Once a set of requirements has been defined, the task of system design begins, which typically includes subsystems in a hierarchical fashion. Many considerations are taken into account that include cost, component availability and lead time, schedule, performance, development environment[s], testability, and system maintenance. For example, the Linux operating system may be chosen due its low cost, ubiquitous deployment, and rich development environment. However, in other cases a real-time operating system (RTOS) may be selected to meet specific performance requirements.

This chapter will enumerate the high-level system requirements and follow with a multi-discipline design that meets the needs of the system while simultaneously adhering to the objectives of the research, which is to create an extensible purpose-built percussion robot that can render expressive performances that are comparable to its human counterparts.

System Requirements

Based on the analysis of human motion in the context of percussion performances, cost, usability, leverage, interoperability, and research objectives, a high-level requirements matrix was created as shown in Table 4. As is indicated in the table, there is a direct correlation between a requirement and a design decision. Given the large array of technology options across several disciplines, there are certainly multiple methods to fulfill specific requirements; however the design decisions are often influenced by pragmatic constraints that effectively reduce the range of options.

Table 4. High-level requirements matrix.

Requirement	Design
Autonomous self-contained system	Integrated software, electronics, and mechanical systems
Four degrees of freedom	X and Y axis motion for left and right side
High-resolution closed loop position control	Quadrature encoder and PID controller
Current magnitude and direction control	H-Bridge driver and PWM generator
Current monitoring and fault detection	Current sensing amplifier
Standard interfaces for live or studio environment	MIDI, OSC, USB, Ethernet
User interface for configuration and status	LCD with illuminated buttons
International commercial off-the-shelf power supply	120-240VAC 50-60Hz adapter
Interchangeable striking implements	Collet system for drumsticks, mallets, and brushes
High acceleration zero backlash actuators	Industrial voice coil actuators
High-performance low-latency extensible computing platform	Xilinx Zynq FPGA w/ Dual ARM9 cores
Audio inputs for recording and machine learning	Electret and external microphone inputs
Temperature management	Heatsink, fan, temperature sensor
Non-invasive to percussion instrument	Tripod or scaffold mounting system
High-leverage ubiquitous operating system platform	Linux operating system
Development support	JTAG, Serial, and debug/expansion port
Persistent storage for operating system and configuration files	MicroSD card

Before a robotic system can be developed to model a human performance, research must be conducted to understand the associated range of motion on the target instrument. Attributes such as timing, velocity, and position in multiple dimensions represent a complex interaction between the musician and the instrument. Seminal work by Sofia Dahl has served to lay the groundwork of understanding with respect to striking implement motion [76]. In addition, a pilot drum study was conducted to acquire and analyze the unencumbered human performance motion in a non-invasive manner [78]. The results of this study informed a set of constraints for a robotic system that can closely approximate human motion while maintaining a fundamental goal of a robust and practical design.

The key ingredients in creating multiple strikes through drum head bounce are impact event detection, pressure adjustment, and precision timing. Impact events can be detected by calibrated absolute position or haptic feedback as discussed previously. Once the impact has been detected, the controller must reduce the downward pressure by adjusting the PWM signal to the VCA driver. It is the reduced downward pressure that allows the striking tip implement to “bounce” up from the taught drum head.

From a usability perspective, the instrument should be very simple to setup, configure, and operate in a performance setting. This implies minimal physical dependencies on the user, such as installing striking implements and mounting the system to a rigid body located next to the percussion instrument. Configuration should be local to the device and remain persistent across power cycles. A minimalist and self-explanatory menu system will allow the user to complete a task with little effort in an efficient manner. When it comes to a compatibility, the musician/composer should have the

expectation of simple integration into their existing electronic music environment that includes DAWs, keyboards, sound modules, and standard power sources.

As a researcher, the platform must provide industry standard methods for software development and the application of upgrades. A ubiquitous programming environment with generous resources will allow the researcher to explore entirely new areas of functionality by utilizing an underlying framework that abstracts multi-core application boot, system configuration, motion control, data acquisition, and supporting hardware manipulation.

System Design

In addition to purely mechanical components, the robot is composed of four independent voice coil actuators with dedicated drive and quadrature encoder circuits as shown in Figure 39. The electronics spans three separate PCBAs that include the custom carrier board, keyboard, and the AVNet MicroZed single board computer. A modular design was chosen in order to take advantage of an established commercial off-the-shelf system while simultaneously providing an application-specific substrate.

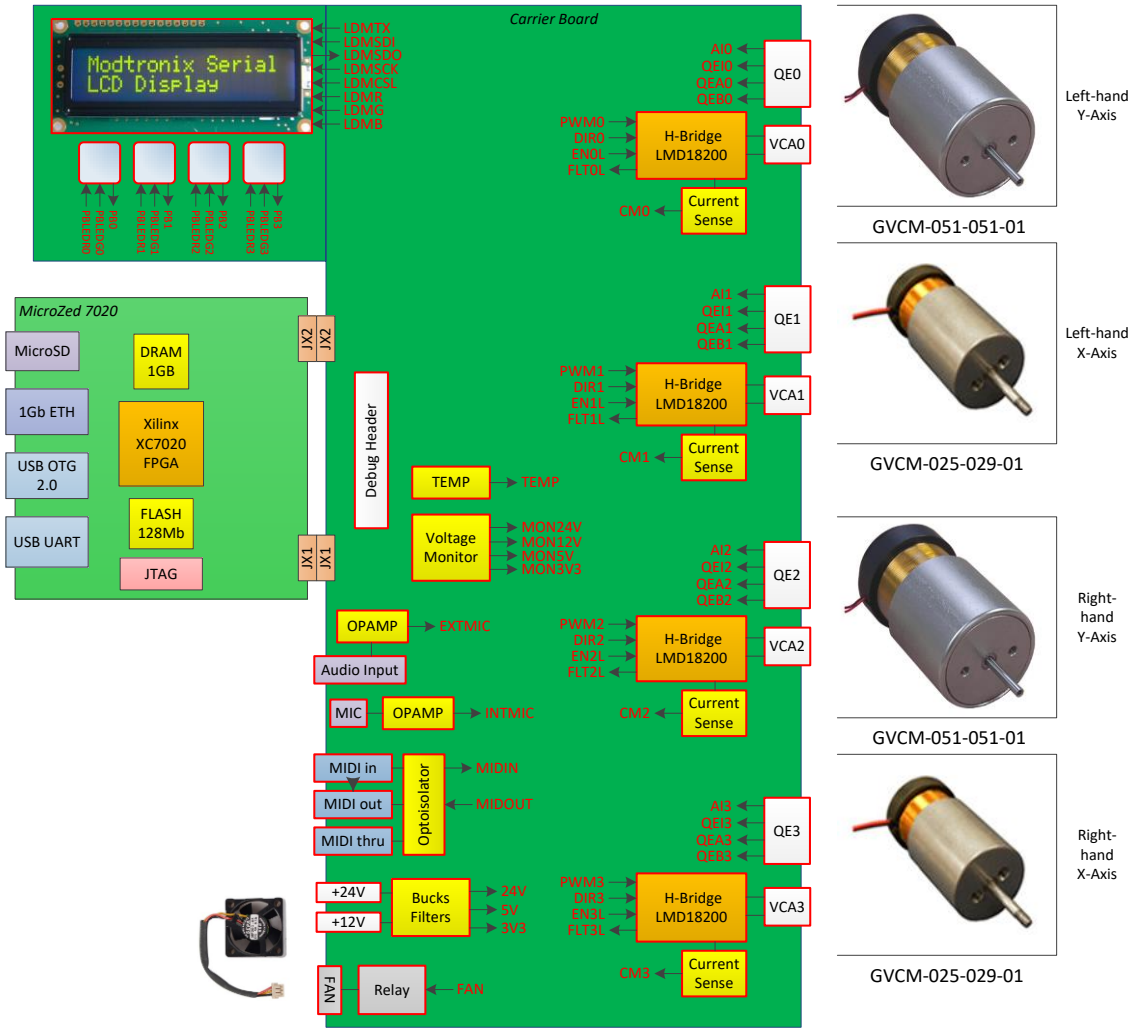


Figure 39. Mechatronic system block diagram.

Mechanical Design

The mechatronic drummer as shown in Figure 40 offers four degrees of freedom (4DOF), a robust enclosure, an intuitive user interface, and a set of peripheral ports that were all designed to meet, and in some cases, exceed human ability as informed by pilot drum study data. In general, the mechanical design reflects the concept of form following function with the primary objective being musical fidelity. This has led to a compact and

optimized design that is intentionally non-anatomical in nature due to the imposed constraints of such an embodiment. By focusing on striking tip implement motion as opposed to mimicking human biomechanics, this unique approach could be simplified in a manner that offers exceptional performance with reduced complexity.

Figure 40. MechDrum^{tm} promotional photo collage.

With 4DOF, the mechatronic drummer has both X and Y axis motion for the left and right striking implements, but does not support motion in the Z axis. The lack of a third axis of motion stems from the discovery that there was minimal activity on this axis during rudiment and musical piece performances in the pilot drum study [31]. Further, since percussion instruments tend to be circular, the X axis can be used to explore similar timbres to extreme Z axis locations by moving outwards from the center of the instrument. Given the typical static nature of Z axis motion as discovered in the study, the position of the mechatronic drummer relative to the percussion instrument can be adjusted as defined by an individual rendering rather than burdening the design with an additional 2DOF, which would significantly impact the system complexity and

potentially reduce the performance potential of the primary motion planes. It should be noted however that motion in the Z axis plane does occur in other contexts. For example, when using brushes the musician will typically use a circular motion pattern that involves significant motion in the Z axis.

Voice Coil Actuators

High-resolution, powerful, quiet, and fast motion demands actuators with low position quantization, high acceleration, low mass, low friction, low hysteresis, and no backlash. Although solenoids possess some of these characteristics, there are inherent limitations that include a lack of precision motion control. Solenoids are essentially Boolean devices that are meant to be in one of two states, i.e., open or closed. Although PWM can be used to set the average current through a solenoid to scale its force, it cannot be used in a closed loop system for the precise positioning required to facilitate expressive dynamics. In contrast, VCAs, when instrumented with position encoders, can achieve a remarkable level of performance when coupled with appropriate driver electronics and closed loop servo control. Other types of actuators such as DC motors and servos have also been used effectively in the context of percussion robots [56]. However, VCAs offer a unique combination of characteristics that make them a compelling alternative for use in musical robots. They represent the simplest form of non-commutated (single phase, brushless) motors, which increases robustness, reliability, and performance. The basic VCA design has been in use since it was first invented by Oliver Lodge in 1898 in the creation of a moving coil dynamic loudspeaker, which is the origin of the “Voice Coil” moniker [84].

Actuator Fundamentals

Voice coil actuators are continuous analog devices composed of a permanent magnet, coil winding on a shaft, and a linear bearing as shown in the cross section of Figure 41. The technology is similar to a loudspeaker; however instead of the bobbin modulating a suspended paper cone to effect localized air pressure, it moves a linear shaft over a stroke range. Electrical characteristics include a back EMF constant, coil resistance, coil inductance, and maximum continuous power. The parameters for the actual vertical and horizontal axis actuators used in the mechatronic drummer are enumerated in Table 5.

Table 5. Voice coil actuators electrical parameters.

parameter	axis	value
back EMF constant	Y	0.20 V/IN/S (8.0 V/M/S)
	X	0.10 V/IN/S (3.75 V/M/S)
coil resistance	Y	2.7 ohms
	X	4.9 ohms
coil inductance @ 1 KHz	Y	1.3 mH
	X	1.6 mH
max continuous power	Y	33 W
	X	9 W

VCAs are linear motors with a permanent magnet and coil winding whose motion is dictated by the Lorentz force principle [85]. This force is a product of the current and magnetic flux as defined in Equation 12, where F is the force in Newtons, k is a constant, B is the magnetic flux density in Teslas, L is the length of wire in meters, I is the current in amperes, and N is the number of conductors.

$$F = kBLIN \tag{12}$$

Since all of the variables in Equation 12 for a given motor are fixed with the exception of I, the generated force is directly proportional to the input current. In

addition, a change in the direction of current results in a change in direction of force. As shown in the cutaway of Figure 41, the use of a stationary permanent magnet with a moving coil attached to a linear bearing yields an actuator with low mass and low friction.

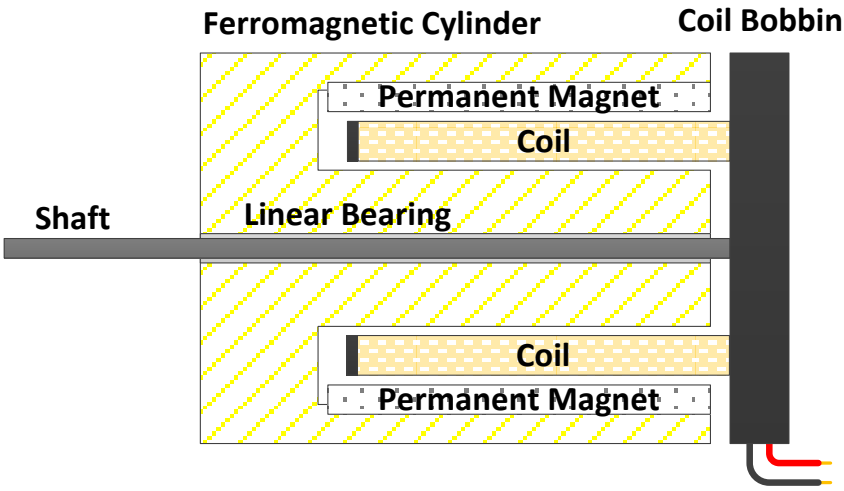


Figure 41. Voice coil actuator cutaway diagram.

The stroke of a typical VCA can range from 0.125 inches to 6 inches with a relatively constant force that drops off at less than 5 percent at the stroke extremes. Since it is a direct drive device, there is no backlash, which enables precision positioning and high acceleration rates.

Rapid coil movement has the side effect of generating back EMF that is proportional to the speed, current, and magnetic field strength. This phenomenon reduces the current and limits the acceleration. However, these limitations are often acceptable in many applications. Further, the VCA can be used to detect motion that is directed by an external force as defined by Equation 13, where E is in Volts, k is constant, B is the magnetic flux density in Teslas, L is the length of the conductor, v is the velocity of the

conductor, and N is the number of conductors. Motion detection can be used to detect haptic feedback events such as striking implement tip impact with a drum head.

$$E = kBLvN \quad \textbf{(13)}$$

Actuator Control

Since the VCA coil bobbin and shaft are free to move along a single axis, there is nothing to hold the assembly in place. In fact, without any mechanical stops, the bobbin assembly will simply fall out of the permanent magnet housing. In a similar fashion, if current is applied to the coil, the assembly will either shoot out of the housing or be driven closed based on the direction of current flow. Therefore, the key to controlling a VCA is a closed control loop, which also implies some form of position sensing.

There are several forms of position sensors that include optical disk linear quadrature encoders and proximity devices such as Hall Effect sensors [86]. The former typically uses dual infrared LED emitter and detector pairs as illustrated in Figure 42 to generate quadrature output signals, whereas the latter relies on a localized magnetic field strength detector. Other forms of position sensors include optical or sonic time of flight, mechanical systems, and variable resistive elements. Sensorless position control has also been explored by measuring position dependent impedance of the voice coil at high frequencies [87].

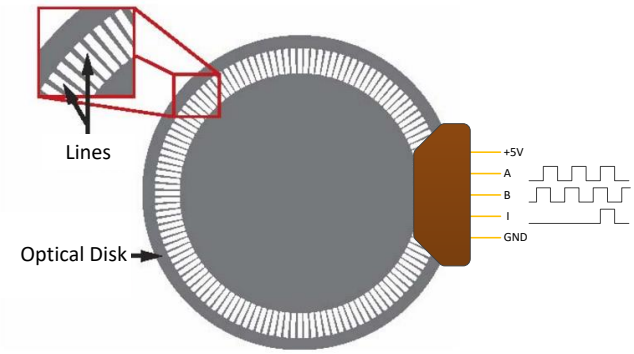


Figure 42. Optical code wheel and quadrature encoder diagram.

The quadrature encoder outputs A, B, and I as depicted in Figure 42 provide a set of signals that can be decoded for speed, direction, and index. The frequency of A or B are a proportional indication of speed in terms of pulses per second, which can be converted to units such as RPM since the diameter and the total number of lines are known. The phase relationship will change between A and B to leading or lagging depending on the direction of rotation. Finally, an index pulse I can be generated once for every full rotation of the disk. In many cases, a quadrature decoder chip is used to present an absolute position value, which can be used directly for closed loop control.

A proportional, integral, and derivative or PID controller is a common motion control method to set and maintain the absolute position of a closed loop system [88]. As illustrated in Figure 43, the weighted sum of $u(t)$ in Equation 14 is a control variable that is used to minimize the error value between the set point $r(t)$ and the measured process value $y(t)$.

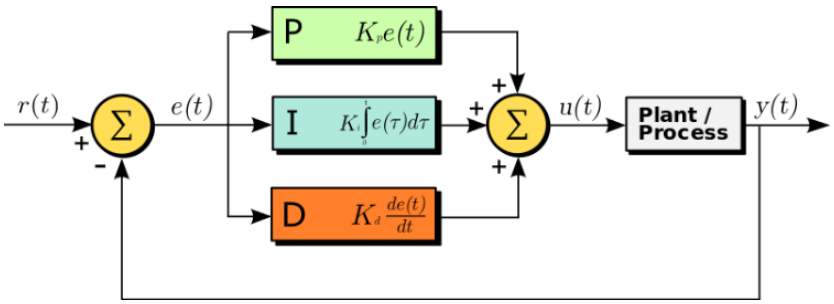


Figure 43. PID controller flow diagram.

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{de(t)}{dt} \quad \textbf{(14)}$$

Referring to Equation 14, the first term accounts for the present value of the error, the second term integrates past values of the error, and the third term embodies future trends based on the current rate of change. Each of the constants K_p, K_i, and K_d are non-negative coefficients that must be tuned for a specific application in order to optimize responsiveness while eliminating oscillations and reducing overshoots. With proper tuning, the PID controller offers a fast and stable means to move a VCA between positions or simply maintain its current position.

Actuator Driver

An H-Bridge circuit is typically used to control the speed and direction of a VCA from a microprocessor. A good example of such a device is the Texas Instruments LMD18200. Common binary logic control inputs include PWM, DIR, BRAKE, and ENABLE. The PWM signal controls the speed of the motor based on the duty cycle. The DIR input dictates which pair of power MOSFET devices is enabled. For example, to move the motor forward a diagonal set of MOSFETs would be driven by the input PWM signal whilst the opposite set remain off. Switching to the opposite diagonal set would result in

reverse motion. In order to stop the motor quickly, the BRAKE signal effectively shorts the motor inputs by enabling the appropriate pair of MOSFETs based on the DIR input. Lastly, de-asserting the ENABLE signal will turn off all of the MOSFET devices, allowing the motor to move freely. Integrated H-Bridge drivers will often include other features such as thermal sensing, automatic overcurrent detection and shutdown, and current sensing.

Actuator Comparison

When compared to other actuator technologies such as solenoids and DC motors, VCAs offer compelling performance characteristics as demonstrated by comparative metrics. J. Long, et al. proposed a methodology for evaluating striking mechanisms in the context of musical instruments [89]. This very informative work added to some of the earlier seminal research conducted by Ajay Kapur, Trimpin, and others towards the goal of developing high-quality robotic drumming systems [90].

The principal performance metrics in the aforementioned study were composed of latency, maximum loudness, dynamic consistency, and maximum repetition rate. Although an exact reproduction of the test environment was not available, a relatively similar approach was used to perform comparable measurements. Notable differences include the use of a 14” Pearl Session Studio Classic drum with snare disabled, a Roland RT-10S vibration transducer attached to the top of the drum head located 150mm from the strike position, and a 24V VCA driver voltage.

Latency

The plot in Figure 44 shows linear VCA actuator to drum head impact latency in comparison to several solenoid designs that had been previous evaluated by J. Long, et al. after compensating for MIDI and microcontroller latencies [89]. This shows that VCA latency performance is roughly on par with the "Linear Solenoid with Pivot" design that has been used extensively in robotic percussion systems. It is very important to note that the mechanical design with respect to transferring actuator motion to a striking implement plays a large role in the performance characteristics of the overall solution. Improper mechanical translation of actuator motion can result in poor performance despite the selection of an adequate electromechanical actuator for a given application.

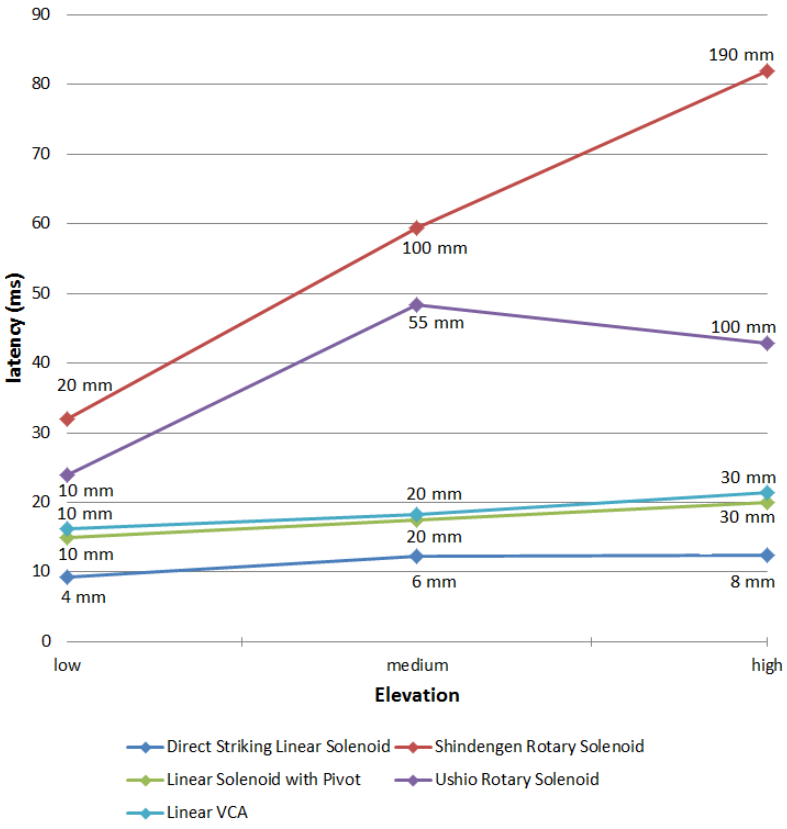


Figure 44. Actuator latency comparison chart.

Maximum Loudness

The bar chart in Figure 45 shows the comparative maximum loudness of a linear VCA in the context of the study conducted by J. Long, et al. [89]. Although the drum type was fundamentally different as indicated previously, the VCA loudness measurement should generally indicate excellent relative performance.

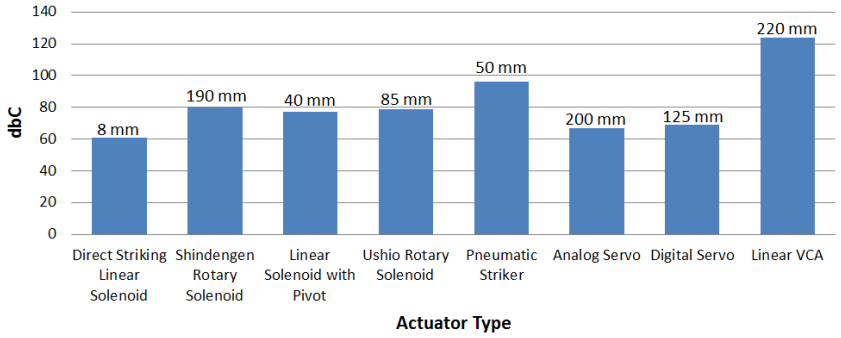


Figure 45. Maximum loudness comparison chart.

As shown in Figure 46, the loudness level is proportional to the preparatory height as expected. However, the level plateaus beyond 220 mm. Further, one can see that that the VCA achieves 92% of maximum loudness at a height of 20 mm, which would indicate very high acceleration within a short stroke.

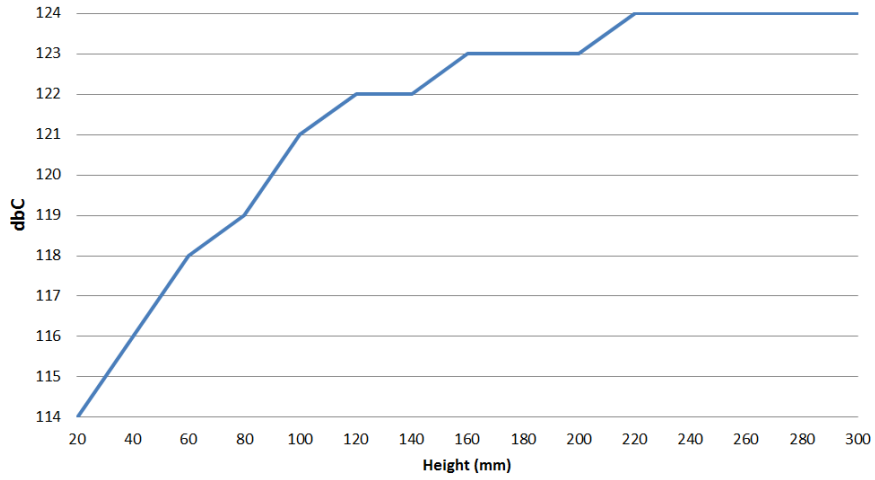


Figure 46. Voice Coil Actuator loudness compared to elevation.

Dynamic Consistency

Consistency in dynamics as a function of height can be expressed in terms of percentage standard deviations for a fixed number of strikes. The consistency of the linear VCA is lower than most of the other actuators that have been characterized by J. Long et, al. as enumerated in Table 6 [12]. Further investigation will be needed to understand and potentially improve this metric. Potential sources of inconsistency include low tolerance linear to rotary coupling components and bearing play in the first prototype.

Table 6. Dynamic consistency percentage comparison.

type	low	medium	high
Direct Striking Linear Solenoid	5.05 (4 mm)	7.12 (6 mm)	9.52 (8 mm)
Shindengen Rotary Solenoid	6.99 (10 mm)	7.4 (100 mm)	3.22 (190 mm)
Linear Solenoid with Pivot	3.5 (10 mm)	3.53 (20 mm)	4.25 (30 mm)
Ushio Rotary Solenoid	4.34 (10 mm)	2.36 (55 mm)	1.44 (100 mm)
Pneumatic Striker	N/A	10.83 (50 mm)	N/A
Analog Servo	4.34 (10 mm)	5.21 (65 mm)	6.08 (120 mm)
Digital Servo	2.93 (10 mm)	3.71 (65 mm)	3.16 (120 mm)
Linear VCA	8.72 (10 mm)	8.66 (20 mm)	11.22 (30 mm)

Maximum Repetition Rate

As shown in Figure 47, the linear VCA achieved 34 strikes per second at a height of 16mm, which ranked a close third when compared to the set of actuators that had been evaluated by J. Long et, al. [89].

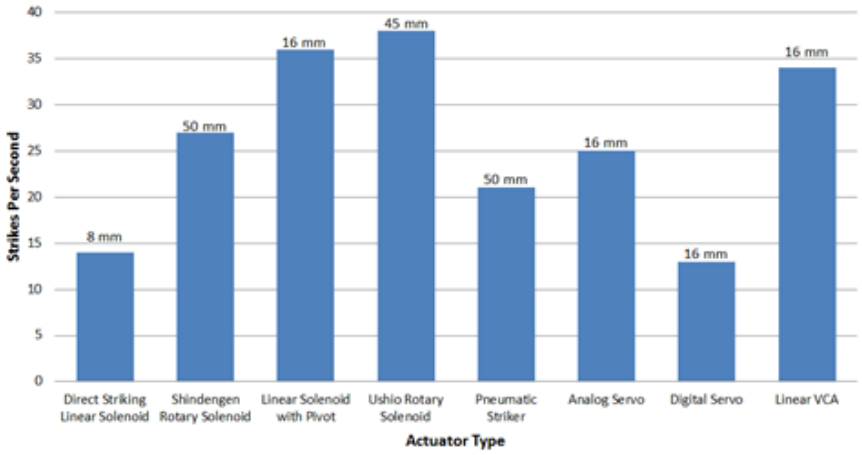


Figure 47. Maximum strike repetition rate chart.

Initial Prototype

A 1DOF prototype was developed using an industry standard VCA, off-the-shelf PWM driver, optical position encoder, custom interface board, general purpose USB control board, and custom console application hosted on a PC as illustrated in Figure 48.

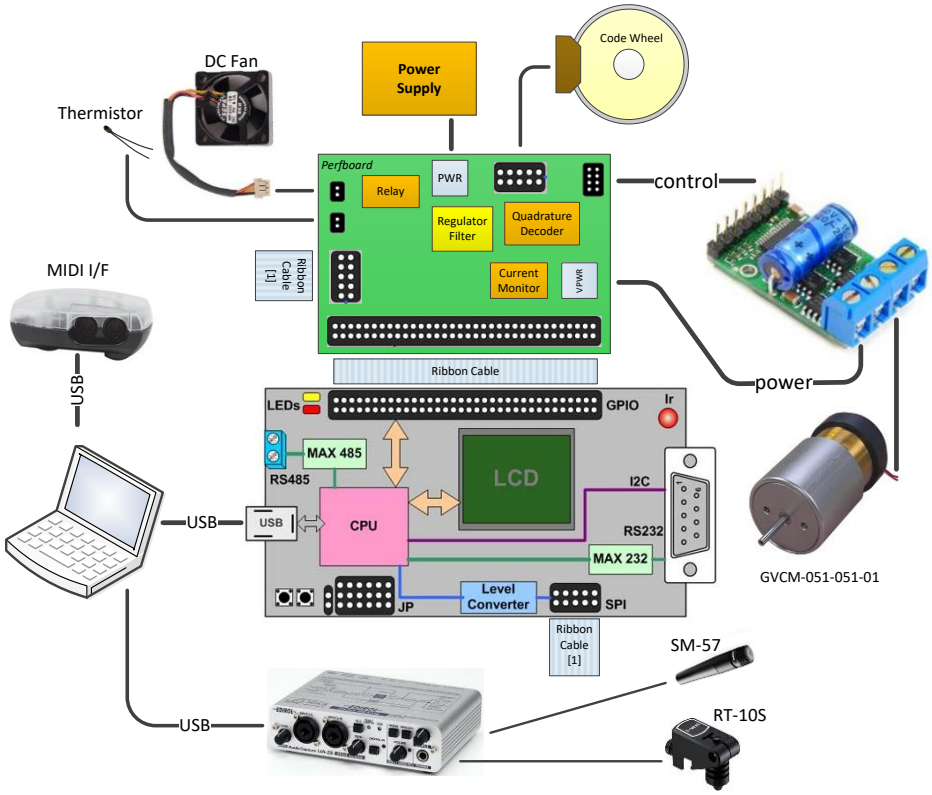


Figure 48. First prototype system schematic.

The mechanical system as shown in Figure 49 was composed of commercially available general purpose components that included aluminum panels, bearings, fasteners, rod ends, a hollow shaft, and a tripod mount. In addition, several custom components were designed and machined to translate linear VCA motion to appropriate rotary striking implement tip motion that included position monitoring using an optical disk quadrature encoder.

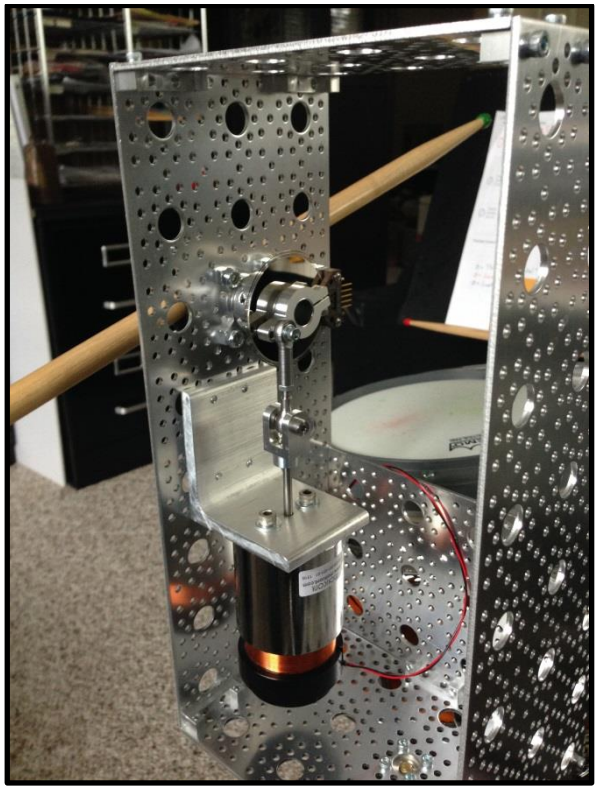


Figure 49. First mechanical prototype showing VCA in relation to striking implement.

The integrated prototype can be seen in Figure 50, which also includes a PC laptop, 24V 90W power supply, and USB MIDI interface as illustrated in Figure 48. A demonstration of the prototype is available at http://tinyurl.com/jlvavta.

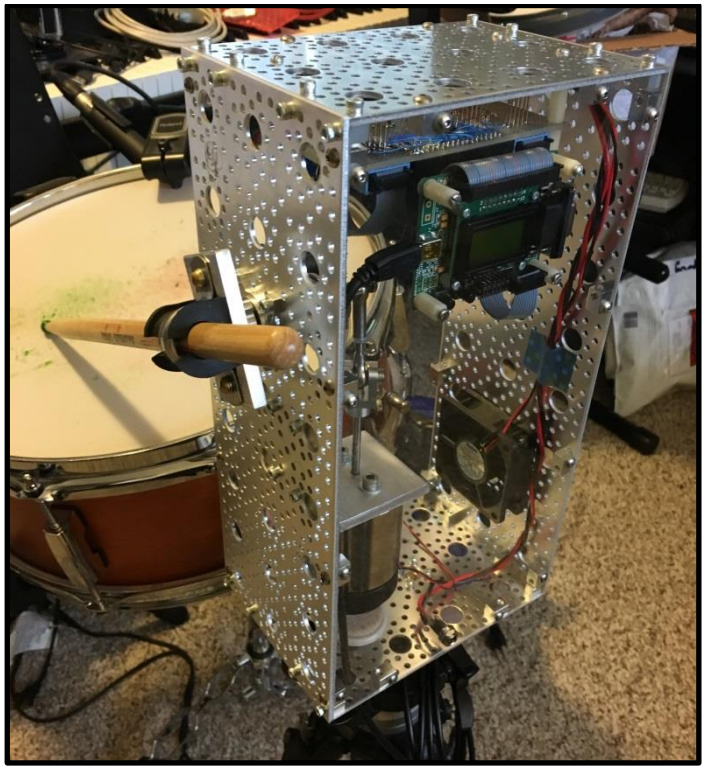


Figure 50. First integrated prototype showing VCA and electronics.

As shown in Figure 51, the first mechanical prototype was leveraged to bring up and test a subsequent hardware platform, which was far more capable then the PC driven experimental configuration in Figure 50.

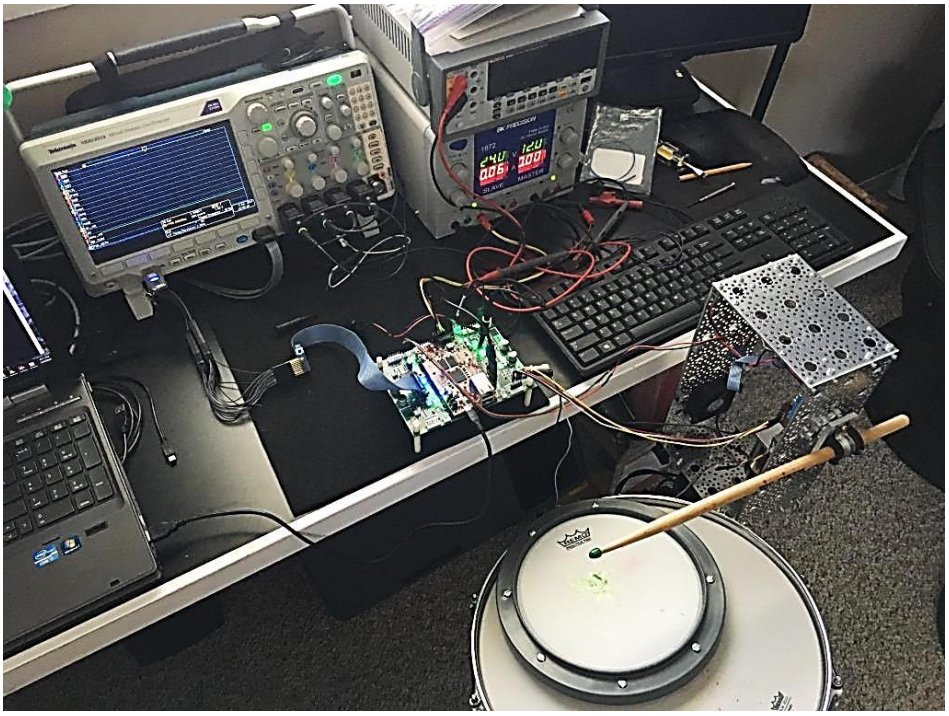


Figure 51. First prototype used to bring-up second hardware platform.

PID Tuning

Adjusting the application-specific PID gain constants in Equation 14 requires an iterative approach that starts with proportional gain. By setting the integral and derivative term gains to zero, the proportional component can be adjusted for a basic response. If the proportional gain is too high, the system will oscillate, whereas if it is too low, the response time will be unacceptably slow and inaccurate. As a general rule of thumb, one should start with low proportional gain and no integral or derivative gain [91]. The proportional gain is then increased until it starts to oscillate then reduced by a factor of two. At this point, a small integral gain can be introduced until the system begins to oscillate after which it should also be reduced by a factor of two. Finally, derivative gain is introduced to anticipate and react quickly to positional changes that are internally or externally induced. Ultimately, fine tuning of all three gains will result in a stable loop

with minimal overshoot and a quick set point response time, which must be verified both at rest and in motion. The set of plots in Figure 52 illustrate examples of PID tuning behaviors.

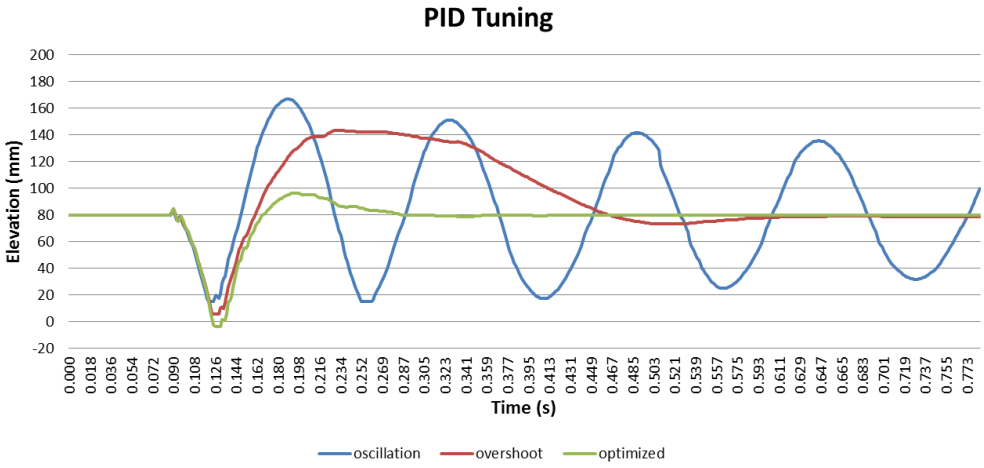


Figure 52. PID closed loop control tuning plot.

Calibration

Positioning the striking implement tip requires a point of reference, which ostensibly should be the drum head surface. Discovering the geometry of the drum as a calibration step is critical to subsequent motion control algorithms that make assumptions about striking implement tip elevation, timing, and expected impact position [92, 93]. Calibration is achieved by gently lowering the striking implement tip until it contacts the drum head surface. This is followed by briefly applying additional force while resetting the quadrature encoder position counter to become the "home" position. Next, the striking implement tip is gently raised until it contacts the upper extent of the mechanics, which is then followed by applying increased force while recording the quadrature encoder position counter value for the "top" position. The typical values for the home and top position on the prototype were 750 and 0 respectively. Given the location at which

the striking implement is mounted to the center of rotation, the tip sweeps a vertical height of 18”, which yields an effect quadrature encoder counter resolution of 0.024” per bit.

Playback

The ability to accurately set the striking implement position in real time inherently enables the playback of pre-recorded or live motion data. The blue motion capture plot shown in Figure 53 is the extracted normalized Y axis left hand motion of the original performance plot that was introduced in **Error! Reference source not found.**. The red playback plot in Figure 53 shows the normalized Y axis optical encoder position of the 1DOF prototype as it rendered the normalized captured performance data. Although they are similar, there are notable distortions that can be attributed to sample rate, PID controller bandwidth, and the overall performance of the mechatronic system. Further mechanical improvements and PID performance tuning will ideally assist in reducing undesirable artifacts.

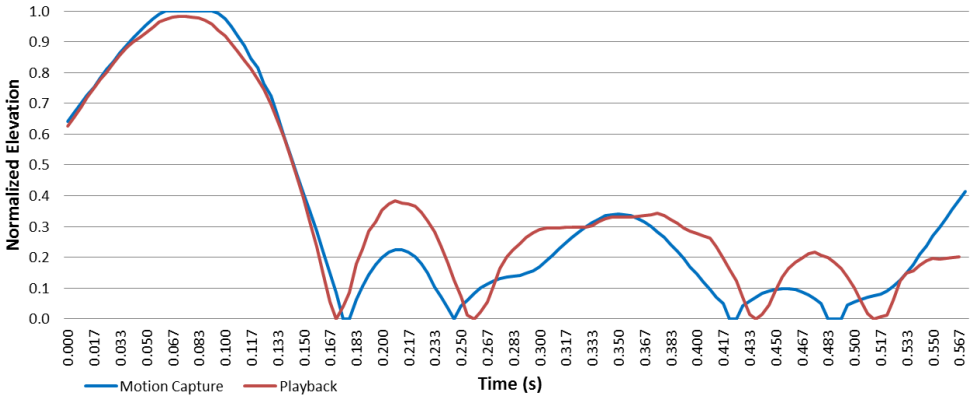


Figure 53. Motion captured position compared to playback location.

Bounce

Percussionists take advantage of a drum head impact by allowing the striking implement to bounce from the first strike through a timely reduction in downward pressure and then reapplying pressure at the right time to induce a second strike. This method is used successively for the formation of triple and quadruple strikes, which form the foundation of drum rolls between the alternating left and right hands.

Since VCA driver strength is controlled by a PWM signal, the prototype can also increase or reduce downward pressure at will, in addition to controlling the absolute position of the striking implement tip. As shown by the optical encoder position plot in Figure 54, this method was used to affect double, triple, and quadruple strikes in a similar fashion to a human performer.

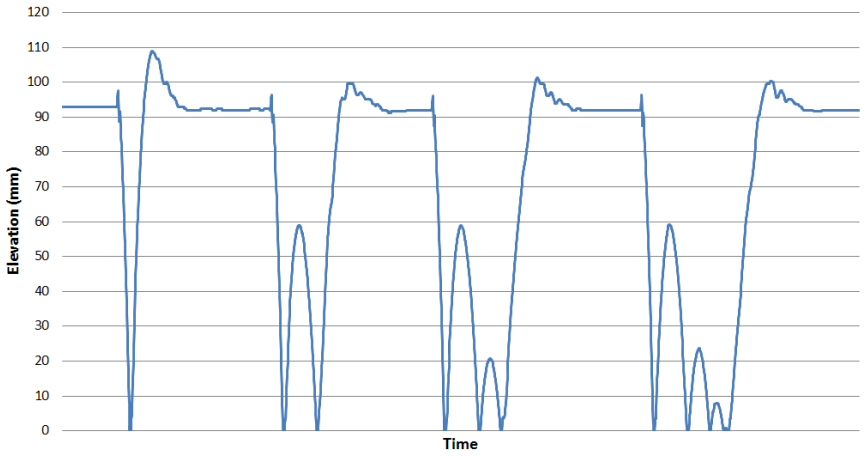


Figure 54. Multiple strike plot showing implement tip bounce.

The final component of a multi-strike articulation is the timing of when downward pressure should be increased again to force a second impact event. As suggested previously, this process is simply repeated with appropriate pressure and timing values to form triple and quadruple strikes. The pressure magnitude changes and timing associated

with multiple strikes can either be precomputed profiles or adaptive in nature through the use of training algorithms on specific drums. The latter method is highly recommended since the type of drum and head tuning will impact the characteristics of the successive strikes in terms of strength and timing as it relates to the BPM of a particular performance.

Vertical Axis

The vertical or Y axis from the mechatronic drummer's perspective in red as shown in Figure 55, consists of an industrial voice coil actuator (VCA), 2DOF coupler, lever arm, rotational shaft through precision bearings, implement forks/shuttle, and finally the striking implement, with the left side mirroring the right. The VCA has an intermittent force at 10% duty cycle of 19.9 lb (88.6 N) that is amplified through a lever arm to produce a striking force at the implement tip that exceeds human ability when measured indirectly using sound pressure level (SPL). The system has considerably less mass then the human arm; however the acceleration is far greater, which results in a larger striking force and lower latency as defined by Newton's second law of motion [94]. As shown in Figure 8, the series of mechanical components that connect to the VCA translates a relatively small 0.750 inch (19.1 mm) linear motion stroke to a much larger rotational motion that traces in excess of 18.0 inches (45.7 cm) at the striking implement tip for a motion magnification factor (MMF) of 24.

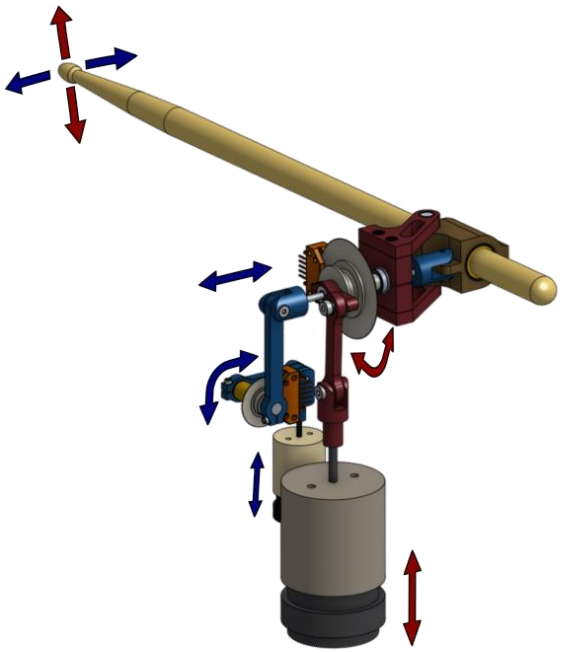


Figure 55. Vertical and horizontal motion planes.

Horizontal Axis

The horizontal or X axis from the mechatronic drummer's perspective in blue as shown in Figure 55, consists of an industrial VCA, 1DOF coupler, slotted 90° offset lever arm and shaft through a bronze bushing, 1DOF slotted coupler, pushrod, and shuttle, with the left side mirroring the right. The VCA has an intermittent force at 10% duty cycle of 57.5 oz (16.0 N) with a lever ratio that magnifies a 0.625 inch (15.9 mm) stroke to a 10.0 inch (25.4 cm) span at the striking implement tip resulting in 16 MMF. The force of the VCA is relatively small when compared to the mass of the mechanism and striking implement, which necessitated a 10 millisecond slew rate limit to improve closed loop settling time. The pilot study revealed relatively slow movement in the X axis, which made slew rate limiting an acceptable damping compensation strategy [31]. The image shown in Figure

56 highlights the rotating clevis design that enables 2DOF by allowing simultaneous X and Y axis motion.

Figure 56. X axis rotating clevis connector.

Enclosure

The enclosure of the mechatronic drummer consists of four computer numerical controlled (CNC) machined 6160 aluminum panels with webbing in order to reduce the total mass to approximately 10 lbs (4.5 kgs) whilst maintaining structural integrity as illustrated in Figure 57. In addition to housing the mechanics, electronics, and pressed bearings, the enclosure also serves as a passive heatsink for four high-power H-bridge MOSFET drivers. The bottom plate provides a standard ¼”-20 tripod mounting point, whereas the top plate contains the buttons and LCD display that comprise the user interface. The front and back covers are made of 0.100” (2.54 mm) transparent polycarbonate that serve as bulkheads for a microphone, fan, and a variety of external connectors. In addition to showcasing the internal mechatronics, the covers create a positive air-pressure chamber that strategically exhausts fresh ambient temperature air

around the H-bridge drivers and supporting electronics when the fan is enabled as depicted in Figure 57.

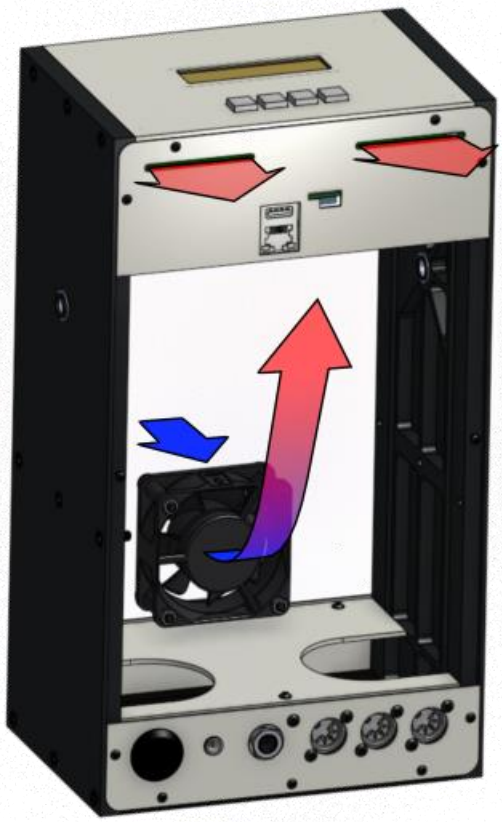


Figure 57. 3D model of enclosure with airflow path.

Assembly

After primary component CNC machining had been completed, elements of the robot were pre-assembled in order to test fit critical parts prior to the anodizing process and bearing presses. The image shown in Figure 58 documents the initial joining of the enclosure, LCD panel, keyboard, and right shuttle assembly.

Figure 58. Pre-assembly test fit, orthogonal view.

As can be seen in Figure 59, the right Y axis voice coil actuator is bolted to the horizontal motor mount plate and the 2DOF linkage has been connected to the output shaft. The vertical X axis fence is also in place to validate general linkage motion.

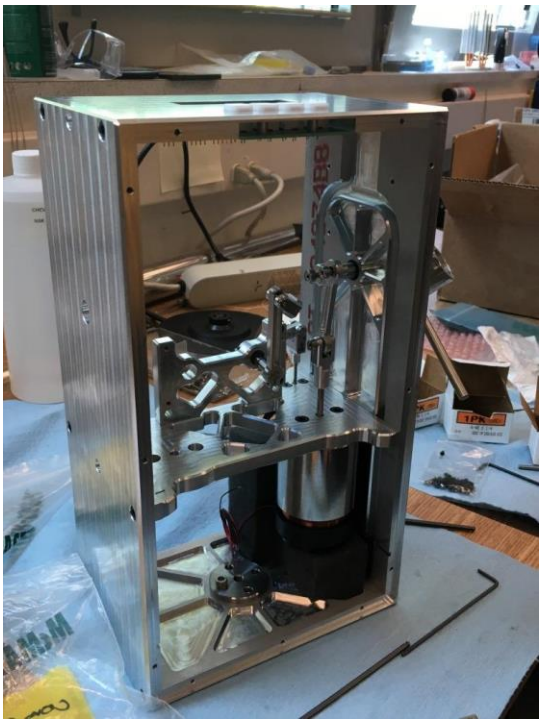


Figure 59. Pre-assembly test fit, voice coil mount.

The final assembly of the robot took place after the anodizing process had been completed and the bearings were pressed into the left and right walls of the enclosure. The stepwise process began with the enclosure, motor mount, voice coils, and linkages as shown in Figure 60.

Figure 60. Final assembly, voice coils and linkages.

The robot contains four main wiring harnesses that are encased in a stainless steel braid for signal shielding [95]. As shown in Figure 61, the left harness is embedded within a channel on the side wall of the enclosure and is affixed with small dabs of two-part epoxy. The Y axis code wheel and quadrature encoder is also in place with further clarity of the X (blue) and Y (red) axis linkages.

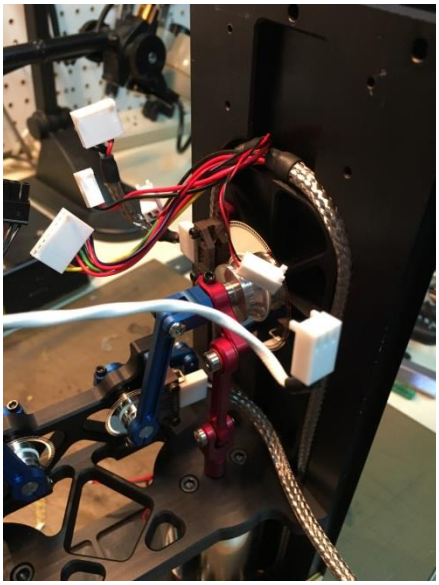


Figure 61. Final assembly, right linkage and wiring.

Figure 62 shows the completed wire harnesses with connectors that attach to the custom carrier PCBA. The harnesses includes power, an external audio input, MIDI ports, and four independent channels of voice coil leads and quadrature encoder interfaces.

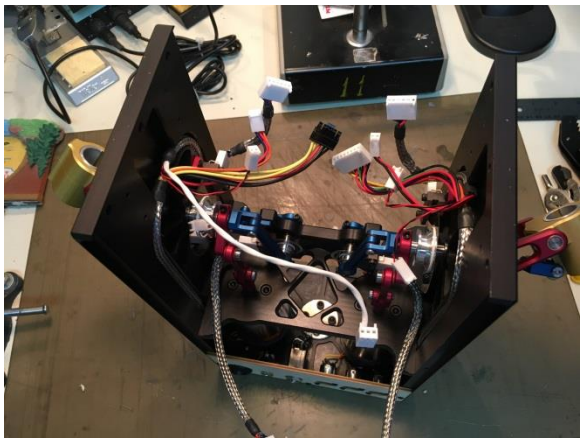


Figure 62. Final assembly, linkages and wiring complete.

The image in Figure 63 captures the four voice coils in full extension with their associated leads and bulkhead terminations for the external audio input and MIDI output port.

Figure 63. Final assembly, voice coil actuators and connector bulkhead.

Figure 64 shows the completed assembly on a tripod in a lab setting with connections to a workstation in support of software development and signal analysis. Individual subsystems were brought up and unit tested in a stepwise fashion to ensure system integrity.

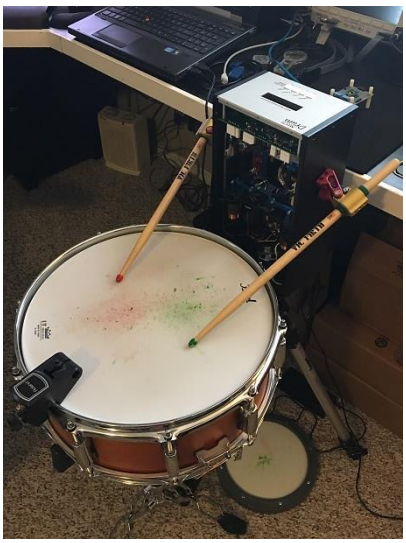


Figure 64. Completed assembly under development in the lab.

Electronic Design

The system electronics is composed of a field programmable gate array (FPGA), optical quadrature encoders, H-bridge drivers, voice coil actuators (VCA), buttons, LCD panel, microphone pre-amplifiers, power regulation, temperature monitoring, and a variety of peripheral ports. As shown in Figure 65 the custom printed circuit board assembly (PCBA) hosts a commercial off-the-shelf (COTS) AVNET MicroZed (part number: AES-Z7MB-7Z010-SOM-G/REV-F) single board computer (SBC). The SBC contains the FPGA, 1Gb of DDR3 memory, MicroSD card slot, and a variety of other peripherals along with general purpose input/output (GPIO) lines that are utilized by circuitry on the custom SBC carrier PCBA, with connectivity through two high-density fine pitch board-to-board connectors.

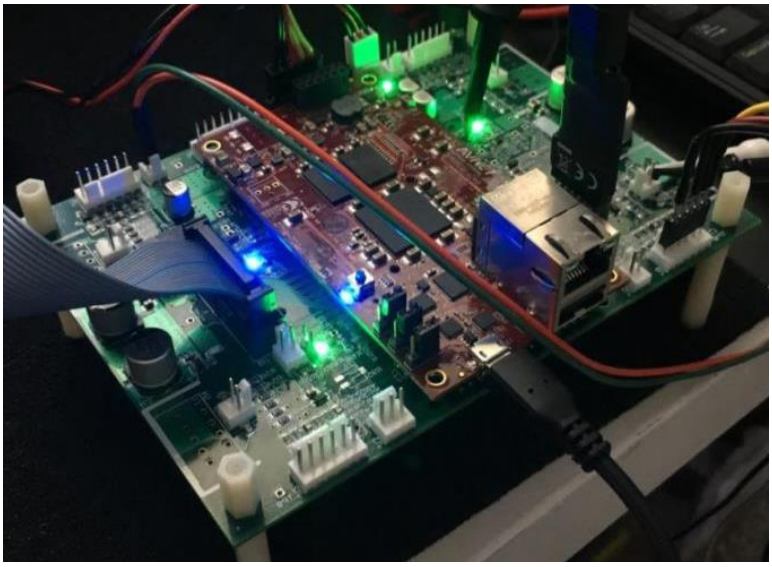


Figure 65. AVNET MicroZed SBC hosted by custom PCBA.

3D models of the carrier and keyboard modules were created from the PCB artwork as shown in Figure 66 and Figure 67 respectively. The models provided a means to validate the physical constraints of components relative to the mechanical design.

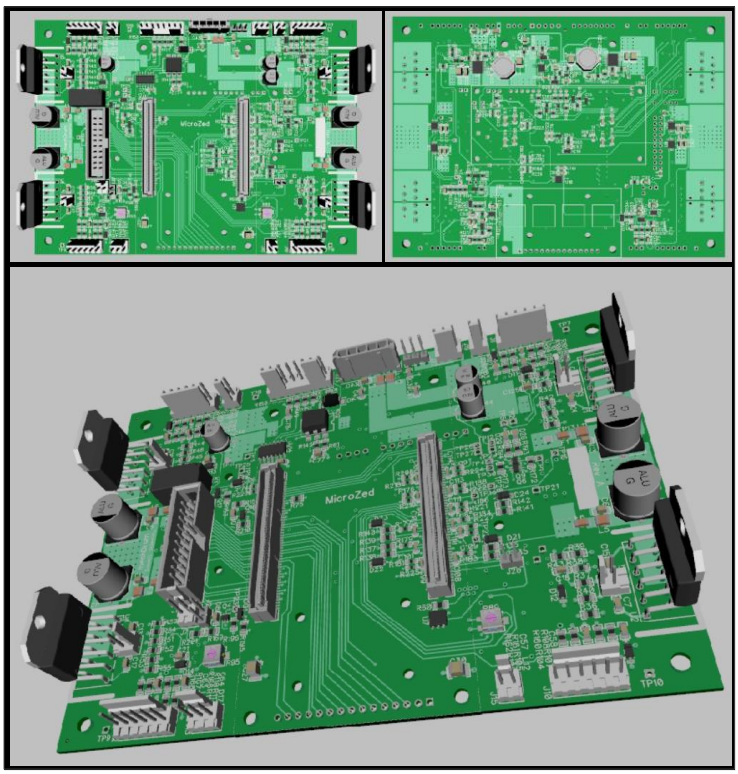


Figure 66. Custom carrier module multi-view model.

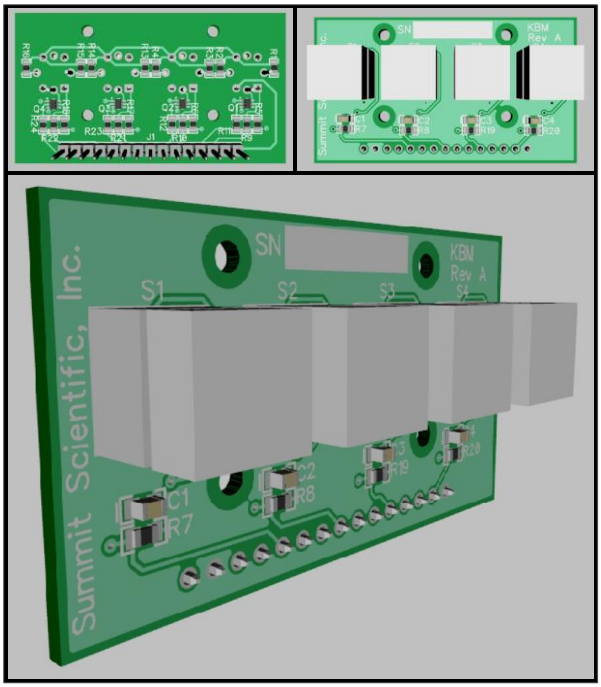


Figure 67. Custom keyboard module multi-view model.

Field Programmable Gate Array

The Xilinx Zynq 7Z010 FPGA is housed on the MicroZed SBC and offers a dual ARM Cortex-A9 processing system (PS) along with an Artix-7 programmable logic (PL) fabric as depicted in Figure 68 [96]. The PS includes a 32KB level 1 cache and a 512KB level 2 cache, on-chip boot ROM, 256KB on-chip RAM, eight channel DMA controller, and high-bandwidth connectivity between the PS and PL. The PL has 28K programmable logic cells, 17,600 Look-up tables (LUTs), 35,200 Flip-Flops, 2.1Mb block RAM (BRAM), and 80 DSP slices yielding a peak performance of 100 Giga multiply-accumulates per second (GMACs).

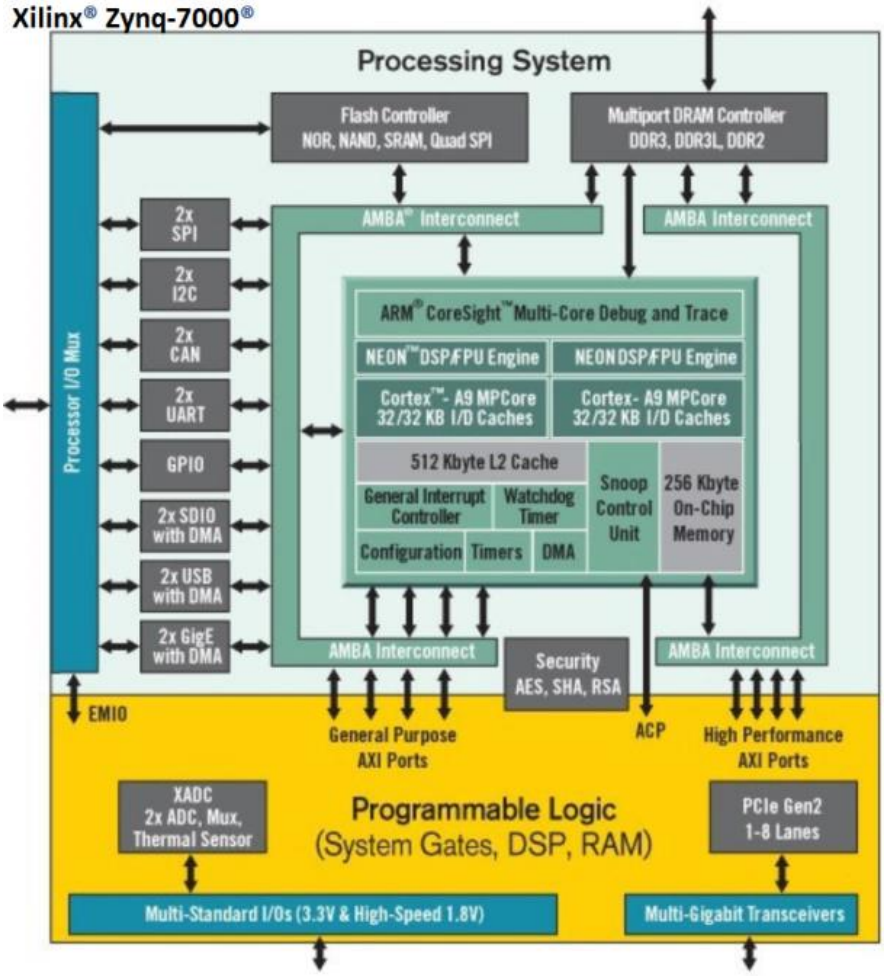


Figure 68. Xilinx Zynq internal architecture (image provided by the Xilinx Corporation).

The dual CPU cores of the PS execute the PetaLinux operating system and bare-metal firmware, whereas the PL contains four custom independent floating point proportional integral derivative (PID) closed loop controllers, quadrature decoders, strike and strain detection circuits, PWM LED controller, digital measurement port, debug port, and additional support logic. The block diagram in Figure 69 shows a logical view of the CPU cores and custom FPGA circuits.

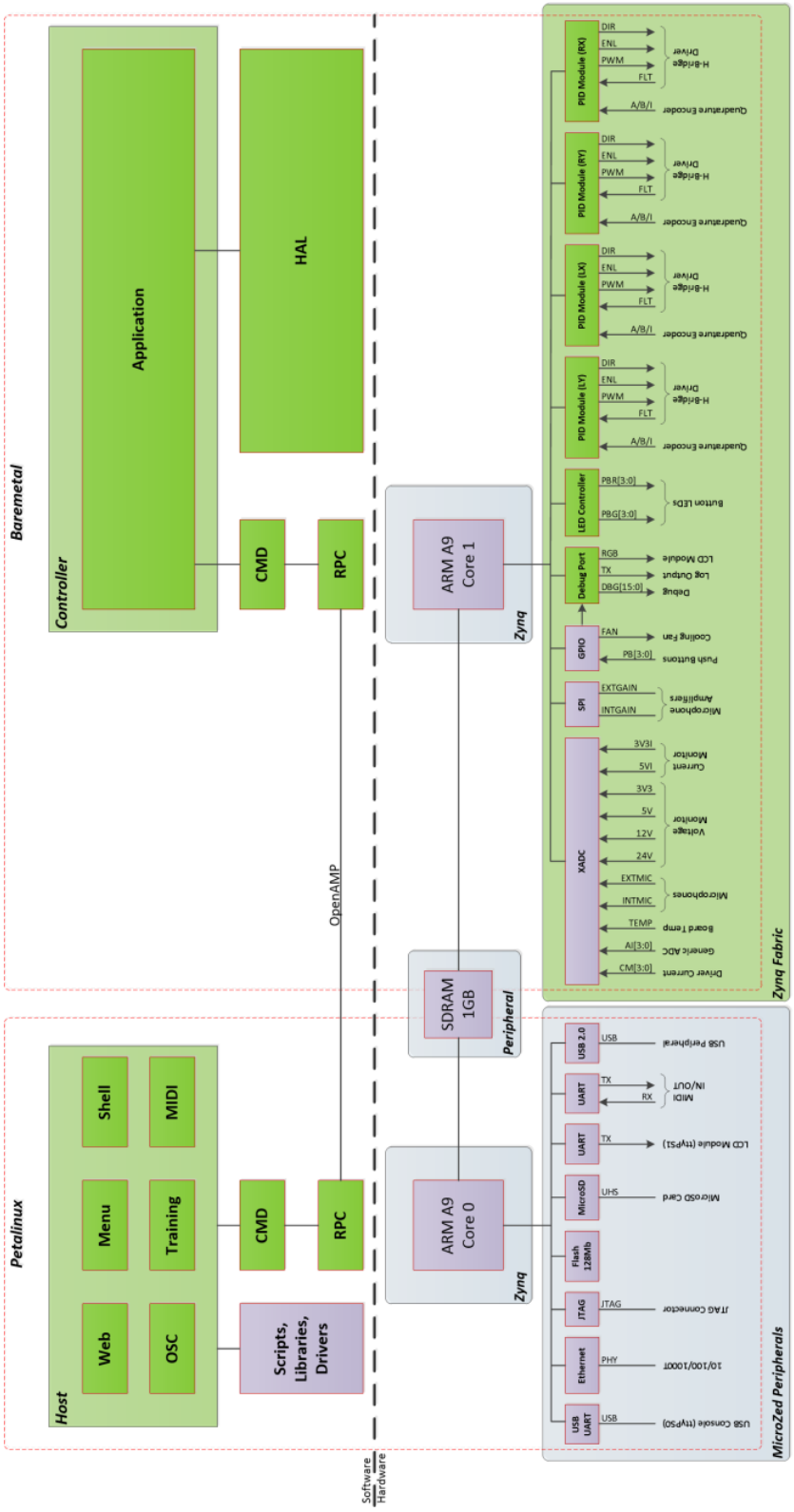


Figure 69. Logical system architecture diagram.

Optical Quadrature Encoders

The 4DOF positioning system utilizes independent optical quadrature encoders for each motion plane. The left and right vertical or Y axis points of rotation use 2 inch diameter 1000 count per revolution (CPS) code wheels and three channel optical encoders [97, 98]. In contrast, the left and right horizontal or X axis points of rotation employ 1 inch diameter 500 CPS code wheels [98]. Each of the optical encoders has three channels composed of A, B, and I outputs as shown in Figure 42. Speed and direction are determined by the frequency and phase of signals A and B, whereas the I, or "index", is used as a home position to reset the quadrature decoders contained in the PL.

With a Y axis striking implement tip range of 18.0 inches (45.7 cm) and an absolute quadrature decoder count of ~750, the vertical resolution is 0.024 inches (0.6mm). In contrast, the X axis striking implement tip range of 10.0 inches (25.4 cm) with an absolute quadrature decoder count of ~100 yields a horizontal resolution of 0.1 inches (2.54 mm). By evidence derived from a pilot drum study and measured human cognitive/muscle processes in the literature, the 4DOF resolution is within the noise floor of human motion [99].

H-Bridge Drivers

The VCAs are controlled by high-power MOSFET H-bridge drivers with dedicated 3A surface mount technology (SMT) fuses [100]. The four independent drivers provide a means to vary the magnitude and direction of the current for each VCA using a digital PWM frequency of 25 KHz as sourced by the PL. The oscilloscope trace in Figure 70 shows the relationship of the PWM and direction signal along with consequential VCA

motion, which is provided through the digital measurement port and companion multi-channel digital to analog (DAC) converter protoboard as seen in Figure 71.

The traces in Figure 70 show the location (yellow) and closed loop controller output signal (cyan) in action as the system reacts to a strike event. In this example, the strike was issued over a MIDI interface specifying the maximum velocity with a static preparatory height of 1 inch (25.4 mm) between the striking implement tip and drum surface. The latency from the start of motion to the impact event is 16.6 milliseconds, whereas the recovery time from the impact event to the pre-strike position is 19.1 milliseconds. With tuned proportional, integral, and derivative (PID) controller gains, one can see the resulting duty cycle signal and how it relates to the physical position of the striking implement as seen in Figure 70. This method was used to tune the PID controller gains empirically, which is a viable alternative to simulating the system with a first order plant model [101].

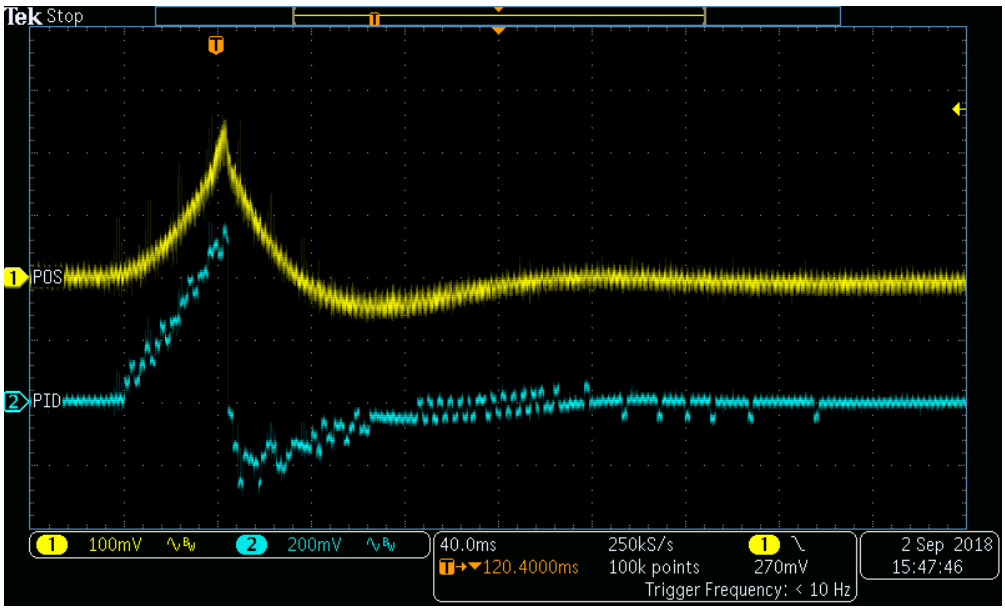


Figure 70. Location and closed loop controller output during a strike event.

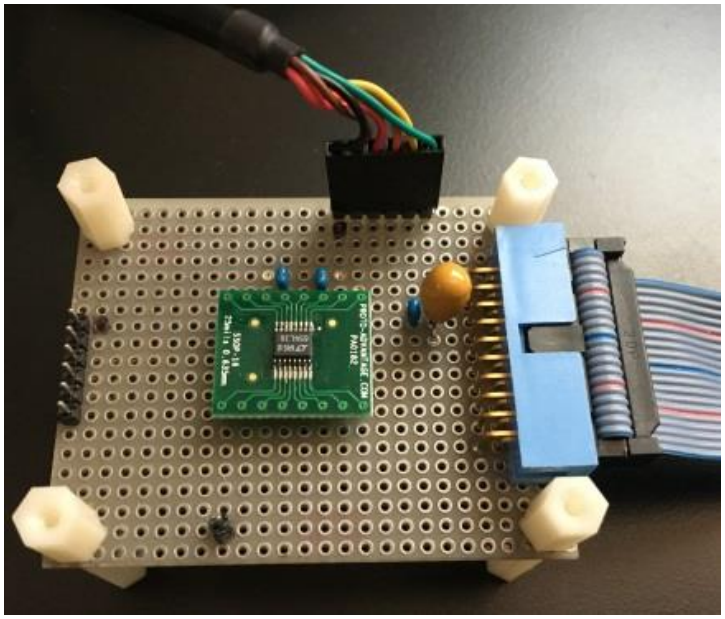


Figure 71. Digital to analog conversion board.

Closed Loop Control

In order for the VCA and associated mechanics to maintain an arbitrary position, an external sensing system must be used to determine the current position [3]. The diagram in Figure 72 shows the generalized primary components that comprise closed loop control for each of the motion planes. Starting at the top of the loop, the striking implement is affixed to shaft that retains a code wheel. The code wheel is read by an optical quadrature encoder that generates three channels of data for a quadrature decoder circuit in the PL. The absolute location is then fed forward to the PID controller that generates PWM and direction signals for the H-Bridge driver. In addition to the absolute location, the PID controller in the PL accepts digital set point and bias inputs from registers that are programmed by the PS, specifically by the secondary core running bare-metal firmware.

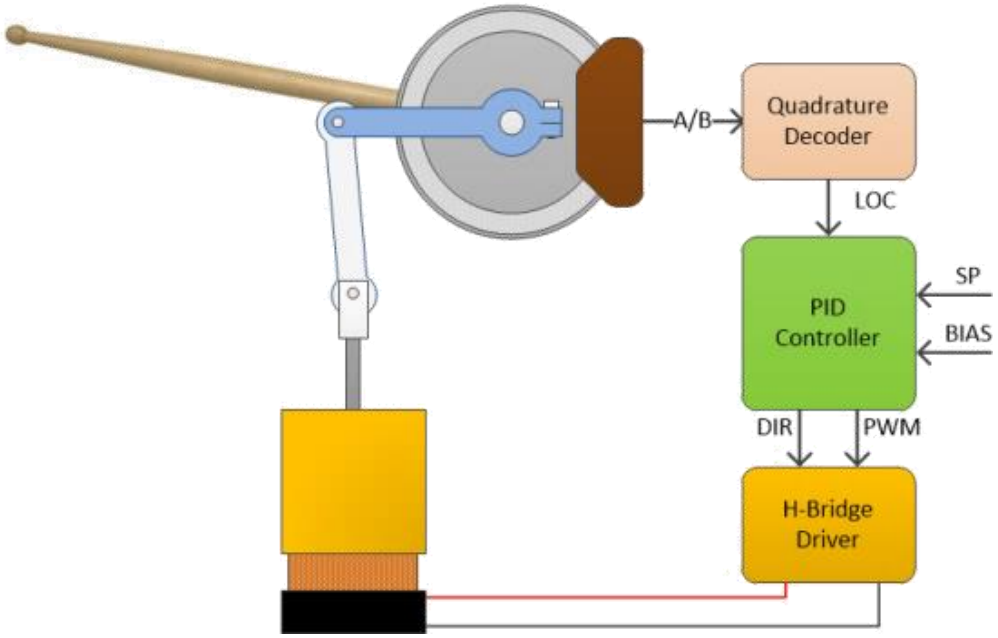


Figure 72. Closed loop control system.

The H-bridge MOSFET driver controls the current magnitude and direction through the VCA, which in turn couples the resulting motion through a 2DOF linkage to a lever arm that connects to the shaft, to which the code wheel and striking implement are affixed. The static PID controller parameters include the proportional, integral, and derivative gain coefficients as well as the PWM clock rate, and duty cycle timeout threshold. The static values have been tuned and calibrated for optimal performance. The dynamic parameters consist of the position (or set point), bias, and strike detection threshold. The bias and strike detection parameters are utilized and set respectively during calibration, whereas the position is used to specify the desired location as directed by the Open Sound Control (OSC) or MIDI interfaces. The custom PID controller circuit additionally supports the concept of a strike event with velocity, which when received, will disable closed loop control, set the PWM duty cycle to a level that results in a

proportional strike based on the specified velocity, and resumes closed loop control when the strike has been detected.

Strike Detection

During system calibration, a value is established in decoder units for each X axis sampled position as the percussive surface is probed with the striking implement across the Y axis. This results in a virtual representation of the surface, which can be further interpreted for discontinuities to determine the physical range of motion within the plane as shown in Figure 73. The visible discontinuity in the plot for a tilted drum surface demonstrates that the rim of the snare drum was discovered at 71 decoder units. Upon receiving a strike event, the PS will set the strike threshold based on the current X axis position. When the strike detection circuit located in the PL detects that the actual location exceeds the strike threshold, it will signal the PID controller circuit to re-establish closed loop control, which will restore the position of the striking implement tip prior to receiving the strike event.

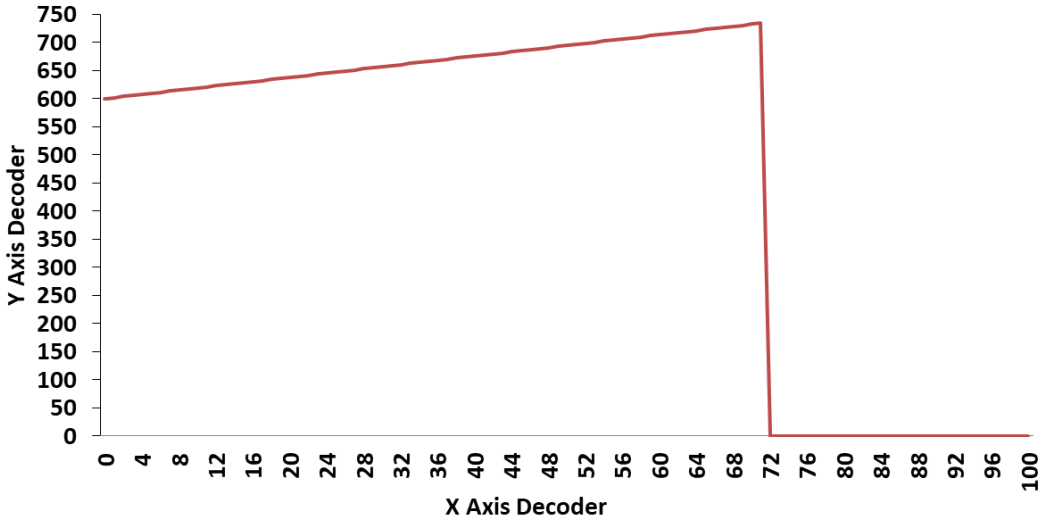


Figure 73. Percussive surface plot.

Strain Detection

If any of the striking implements encounter an obstacle such as the rim of a drum, the strain detection circuit in the PL will automatically disable the associated PID controller, which results in immediate disengagement of force to the related plane of motion. This is used as a safety mechanism to prevent drawing excessive current through the H-bridge driver and/or VCA that can result in overheating or a brownout condition. Strain detection is also used to determine when a striking implement has encountered an obstacle during calibration. The design uses a timer and programmable thresholds in the PL that continually and independently monitors the duty cycle of all four PID controllers. For example, if the duty cycle exceeds a threshold of 80% for 500 milliseconds, the strain detection circuit will automatically disable the related PID controller.

User Interface

The user interface is composed of four bi-LED (red/green) illuminated tactile buttons and a single 16 character by two line backlit display as shown in Figure 40 [102]. Each button is pulled up by a resistor and when pressed, shorts the resistor to ground, which generates a Boolean signal on the general purpose I/O (GPIO) inputs for de-bouncing in software by the primary processor core running PetaLinux [103]. Each of the bi-LEDs in the buttons is driven by the LED controller circuit in the PL that allows the secondary processor core running the bare-metal firmware to adjust the intensity of each color, providing a range of hues to convey status information. One of the available status displays modes translates the PWM duty cycle and current direction for each motion plane to red or green color intensities within the PL for a dynamic representation of

motion activity. In a similar fashion, the backlighting on the LCD allows the PS to set a range of colors using the red, green, and blue LEDs.

Audio

The mechatronic drummer has two independent audio input channels to receive real-time synchronized audio streams on the primary core. The first channel is utilized by a built-in electret microphone with a corresponding frequency response plot in Figure 74. The secondary channel is routed to a single ended input connector for use with an external microphone or vibration transducer. The audio channels can be utilized for a variety of high-level tasks that include audio recording, frequency analysis, beat detection, and machine learning.

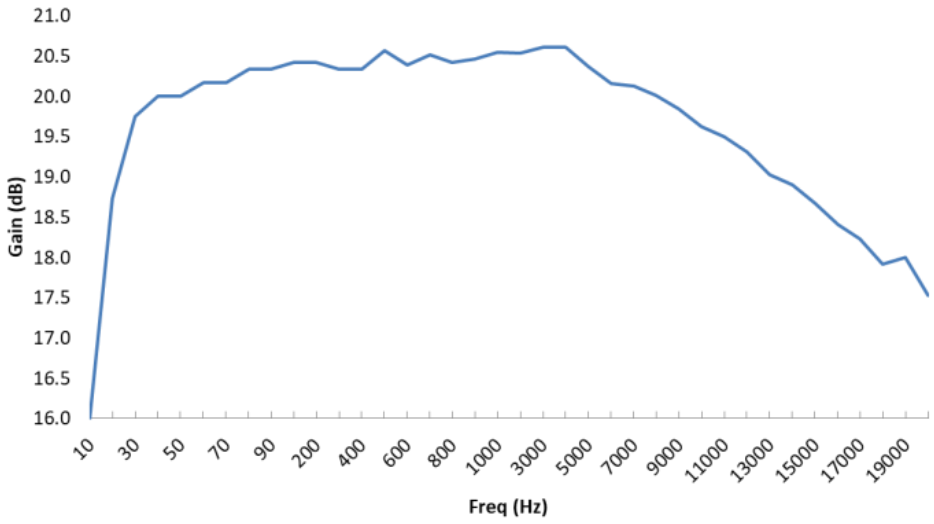


Figure 74. Internal microphone frequency response.

Peripheral Ports

There are a variety of peripheral ports on the mechatronic drummer to facilitate connectivity with external devices. The set of ports include 10/100/1000T Ethernet, USB 2.0 type A, USB 2.0 Micro-B 5pin, MicroSD card, and MIDI input, output, and thru

ports. The Ethernet port offers standard connectivity and supports the OSC, HTML, SSH, FTP, and DHCP protocols. The USB ports enable general system expansion such as storage and audio I/O as well as access to the Linux console. The boot image, kernel, FPGA configuration bitstream, configuration file, and patches all exist on the MicroSD card, which offers extensibility and simple upgrades. Finally, the MIDI interface is at the core of interoperability with other digital musical instruments, controllers, and digital audio workstations. As per the standard, the MIDI interface includes the “thru” port as the established method for creating a daisy chain bus topology [1].

Power

The mechatronic drummer power source is a standard 24VDC 3.75A 90W AC adapter with international input voltage support. The PCBA uses buck converters to regulate the incoming 24V to 5V and 3.3V power rails that share a common 5A SMT fuse [104]. In addition to sourcing the switching regulators, the 24V rail also powers four MOSFET H-bridge drivers through dedicated 3A SMT fuses. The system design supports an additional 12V input that can be used to power the PCBA, which offers the opportunity to isolate and thus dedicate the 24V rail exclusively to the H-bridge drivers. Each of the four power rails are independently monitored by a 12-bit analog to digital converter (ADC) for compliance monitoring by either the primary or secondary processor cores.

Temperature Monitor

There are two temperature monitors in the system. The first is contained directly on the Xilinx Zynq 7Z010 FPGA silicon die and the second is located on the PCBA [96, 105]. This configuration allows the software to monitor chip core and board temperatures,

which are the key inputs to automatic temperature regulation with hysteresis. By monitoring the temperatures and enabling the fan for cooling when needed, the system remains within its normal operating range. In practice, the fan is rarely engaged given the highly efficient passive heatsink capability of the enclosure.

Development Features

Development is supported by a JTAG connector for access to debug cores and device programming, a serial USB port for the PetaLinux console, a serial USB port for the bare-metal firmware, and a general purpose debug/expansion port. A standard JTAG probe such as the Digilent HS2 can be used to program and/or debug the Xilinx Zynq 7Z010 FPGA from the Xilinx Vivado development toolchain [106]. A terminal emulation program such as TeraTerm can be used to connect to the PetaLinux console and bare-metal firmware logging ports running at 115.2Kbps , 8-bits , no parity, and 1 stop bit.

The general purpose debug/expansion port provides a means to route selected FPGA signals to a 0.100” dual row 20-pin header for connection to a logic analyzer or oscilloscope. By using the diagnostic console, the developer can specify which set of debug signals are routed to the port. Additionally, the port can be configured in expansion mode, which allows an external digital to analog converter (DAC) to be connected for real-time visibility of the selected PID controller channel data. This is feature is especially useful for empirical measurements and tuning as described in the PID Tuning section.

Software

The mechatronic drummer is entirely self-contained from a software perspective and is distributed between two ARM9 cores that are contained within the FPGA [96]. The primary core hosts PetaLinux and the secondary core executes bare-metal firmware that manages the logic and closed loop controllers embedded within the PL. The block diagram shown in Figure 69 illustrates the high-level system software architecture. Inter-process communication between the two cores utilizes shared memory and interrupts, which is foundational to a thread-safe custom motion and system control abstraction library. The abstract library is used by the OSC server, MIDI server, system diagnostic, and temperature regulation threads of execution.

PetaLinux

The PetaLinux distribution is a software development kit (SDK) managed by Xilinx that contains all of the components needed to develop, build, test, and deploy an embedded Linux image for Xilinx hardware target platforms [107]. Given its Linux operating system lineage, PetaLinux offers a rich standardized environment for software development that includes operating system, file system, and networking primitives with direct access to source code via the associated open source community.

Configuration of the PetaLinux operating system for a given target is achieved using typical kernel and file system configuration utilities that provide a large set of component inclusion/exclusion and parameter specifications. Furthermore, a detailed description of the hardware is imported directly from the Xilinx Vivado hardware description language (HDL) development toolchain along with user specifications to build a target image that

fully supports the custom mechatronic drummer hardware, which is composed of the SBC and motherboard along with a variety of peripheral devices.

Bare-metal Firmware

Motion and general hardware control/monitoring is provided by custom single-threaded firmware running on the secondary processor core. The firmware communicates with the closed loop controllers in the PL as well as a variety of other custom hardware components as shown in Figure 69. This separate thread of execution outside of the PetaLinux environment offers real-time control of the 4DOF hardware, which is a key contributor to the low-latency system design. The firmware also includes a timer that is used for several lower priority housekeeping tasks such as event logging. The event logging subsystem uses a circular buffer with a global level filter that minimizes calling thread and serial port character drain overhead. In addition, the event logging API provides a flexible and convenient method for capturing detailed timestamped information as shown in Figure 75.

```
DBG[0.00:00:00.00,rpc.c,174,rpcInit]: RPC initialized
INFO[0.00:00:00.00,main.c,57,banner]: Mechatronic Drummer (tm)
INFO[0.00:00:00.00,main.c,58,banner]: Summit Scientific, Inc.
INFO[0.00:00:00.00,main.c,59,banner]: (c) 2016-2017, All Rights Reserved.
INFO[0.00:00:00.00,main.c,61,banner]: firmware: 0.1.6 omega
DBG[0.00:00:01.99,rpc.c,108,created]: RPC channel created
MAX[0.00:53:353,midi.c,679,on]: note: 36, velocity: 100
MAX[0.00:59.368,midi.c,679,on]: note: 36, velocity: 0
```

Figure 75. Example multi-threaded system logging output.

Interprocess Communications

The open asymmetric multi-processing OpenAMP framework was utilized to facilitate bi-directional communications between the two processor cores within the FPGA as illustrated in Figure 69 [108]. The implementation uses shared memory and an interrupt

per core for event handling. The diagram in Figure 76 illustrates the sequence between the master and remote processor for booting, channel setup, messaging, channel tear down, and shutdown. The shared message buffer is a bit less than 512 bytes due to protocol overhead, which is sufficient for the vast majority of command/response transactions. Larger low-priority data transfers are broken up into fragments to accommodate shared buffer size limitations or utilize dedicated out-of-band buffers as defined by the hardware configuration.

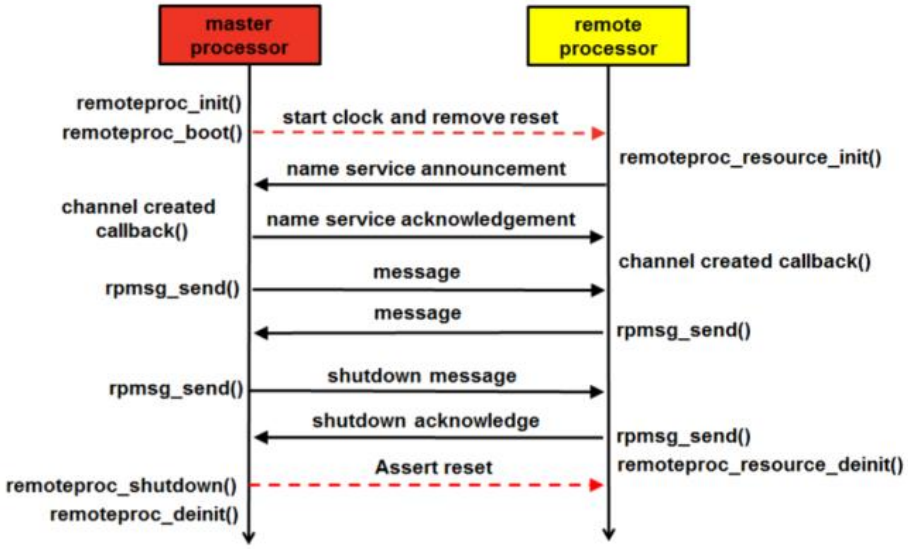


Figure 76. OpenAMP system sequence diagram (image provided by Xilinx Corporation).

Abstraction Library

Control and monitoring of the mechatronic drummer motion and utility hardware is achieved through an abstraction library that exists in the bare-metal firmware. Calls to the library within the bare-metal firmware on the secondary core are direct, whereas calls from the primary core in the context of PetaLinux use a thread-safe veneer API that utilize the inter-process communications infrastructure as illustrated in Figure 69.

The creation of an abstraction library as listed in Figure 77 keeps the logical and physical domain knowledge separated between the two cores. As an example, the logical act of issuing a strike event at a point in time from the primary core does not require any knowledge on how the strike is actually initiated and managed. In contrast, providing a striking primitive by managing the related hardware at a lower level on a secondary processing core removes the complexity of such an action from the primary core, which can focus on high-level constructs such as the timing, velocity, and location of strikes as directed by MIDI or OSC messages.

```
teError cmdInit(thCmd *phCmd);
teError cmdShutdown(thCmd hCmd);

teError cmdButton(thCmd hCmd, teCmdButton eButton, bool *pState);
teError cmdConfigGet(thCmd hCmd, teCmdPlane ePlane, teCmdConfig eConfig,
                     uint32_t *pValue);
teError cmdConfigSet(thCmd hCmd, teCmdPlane ePlane, teCmdConfig eConfig,
                     uint32_t value);
teError cmdCurrent(thCmd hCmd, teCmdCurrent eCurrent, float *pAmps);
teError cmdDac(thCmd hCmd, teCmdDacPort eDacPort);
teError cmdDebug(thCmd hCmd, teCmdDebugPort eDebugPort);
teError cmdFan(thCmd hCmd, bool enable);
teError cmdGainGet(thCmd hCmd, teCmdGain eGain, uint8_t *pValue);
teError cmdGainSet(thCmd hCmd, teCmdGain eGain, uint8_t value);
teError cmdLed(thCmd hCmd, teCmdLed eLed, uint8_t value);
teError cmdLedMode(thCmd hCmd, teCmdLedMode eMode);
teError cmdLedDivisor(thCmd hCmd, uint32_t value);
teError cmdPidConstGet(thCmd hCmd, teCmdPlane ePlane, teCmdPidConst eConst,
                       float *pValue);
teError cmdPidConstSet(thCmd hCmd, teCmdPlane ePlane, teCmdPidConst eConst,
                       float value);
teError cmdPidDuty(thCmd hCmd, teCmdPlane ePlane, uint8_t *pValue);
teError cmdPidEnable(thCmd hCmd, teCmdPlane ePlane, bool value);
teError cmdPidFault(thCmd hCmd, teCmdPlane ePlane, bool *pValue);
teError cmdPidLocation(thCmd hCmd, teCmdPlane ePlane, int16_t *pValue);
teError cmdPidPositionSet(thCmd hCmd, teCmdPlane ePlane, int16_t value, uint16_t microseconds);
teError cmdPidPositionGet(thCmd hCmd, teCmdPlane ePlane, int16_t *pValue);
teError cmdPidReset(thCmd hCmd, teCmdPlane ePlane);
teError cmdStrike(thCmd hCmd, teCmdPlane ePlane, uint8_t velocity, int16_t threshold,
                  uint16_t timeout);
teError cmdStrikeTimeout(thCmd hCmd, teCmdPlane ePlane, bool *pValue);
teError cmdStrikeDetected(thCmd hCmd, teCmdPlane ePlane, bool *pValue);
teError cmdStrikePreTimer(thCmd hCmd, teCmdPlane ePlane, float *pValue);
teError cmdStrikePostTimer(thCmd hCmd, teCmdPlane ePlane, float *pValue);
teError cmdTemp(thCmd hCmd, teCmdTemp eTemp, float *pValue);
teError cmdTime(thCmd hCmd, trCmdTime *prTime);
teError cmdTune(thCmd hCmd, teCmdPlane ePlane, uint8_t velocity, int16_t threshold,
                uint16_t timeout, trCmdSample *prSample);
teError cmdVersion(thCmd hCmd, teCmdVersion eVersion, trCmdVersion *prVersion);
teError cmdVoltage(thCmd hCmd, teCmdVoltage eVoltage, float *pVolts);
```

Figure 77. Abstraction library API listing.

Open Sound Control

The Open Sound Control (OSC) user datagram protocol (UDP) server provides a means of receiving low-overhead telemetry data that define the desired position within 4DOF [109]. Each OSC packet is composed of a data bundle as shown in Figure 78. The header contains the address and type, where the former identifies the unique namespace for the payload and the latter specifies the data types for the seven parameters, which in this case are all 32-bit floating point numbers. The timestamp is in units of seconds and is used to

determine the sample rate of the data, which can remain fixed or varied over time depending on the application. The remaining tuple specifies the normalized X and Y axis position for the left and right striking implement tips along with a whack value that is unused at this time since the mechatronic drummer derives its own striking strength value from the normalized Y axis position data.

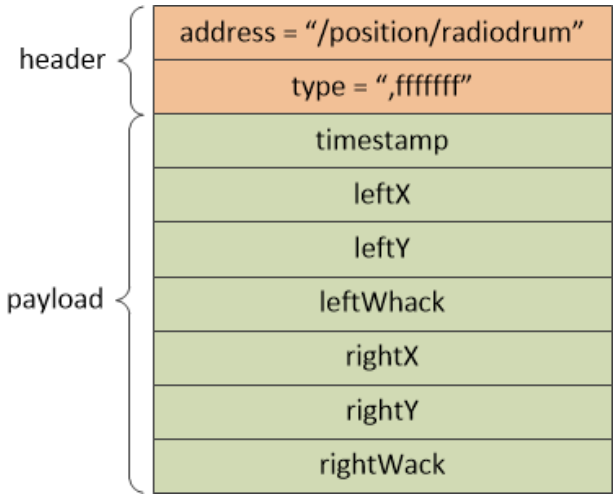


Figure 78. OSC packet data bundle.

Since the sample rate can exceed the bandwidth of the actuators and related infrastructure, the server employs a software whack detection scheme that determines the strength of a strike by calculating the average velocity of adjacent samples that exceed a predefined normalized threshold using the first derivative of motion as defined by Equation 15. If the normalized vertical position sample y is less than the threshold t, the whack strength s is equal to zero, otherwise it is equal to the constant k multiplied by the difference between the current and previous samples of y. As shown in Equation 16, the final PWM duty cycle d of the strike is clamped between zero and a maximum 8-bit value.

$$s = \begin{cases} 0, & y < t \\ k(y_{n-1} - y_n), & y \geq t \end{cases} \quad \textbf{(15)}$$

$$d = \begin{cases} 0, & s < 0 \\ 2^8 - 1, & s > 2^8 - 1 \\ s & \end{cases} \quad \textbf{(16)}$$

When a whack event is detected, the closed loop controller is temporarily disabled by driving the voice coil actuator directly with the scaled PWM duty cycle d that is immediately disabled in hardware upon detection of a physical strike, after which closed loop control is automatically re-enabled in order to return to the striking implement Y axis position that was set prior to the strike, or as defined by a subsequent OSC message depending upon the packet arrival time. This feature is critically important in reducing latency given the typical low-pass behavior of mechanical closed loop control systems.

By supporting flexible rate high resolution 4DOF discrete position samples, the mechatronic drummer can faithfully reproduce human or algorithm driven motion. This fundamental capability can be used in the context of live or playback renderings to closely meet and in some cases exceed the ability of a human percussionist.

Musical Instrument Digital Interface

The MIDI specification was standardized in 1983 and is maintained by the MIDI Manufacturers Associated (MMA) as a technical standard that describes a communication protocol, digital interface, and electrical connectors to interconnect musical instruments [1]. The mechatronic drummer includes a MIDI interface and implementation that enables instruments such as keyboards, drum machines, and digital audio workstations (DAW) to play a variety of rudiments and utilize continuous controllers. Individual strikes can be issued to each implement in a polyphonic manner,

or continuous dual implement strike rudiments can be controlled as specified in the MIDI Mapping chart located in the Appendix (Table 13). All of the notes are velocity sensitive and repeated rudiments include modulation, pressure, and pitch that alter the timing, velocity, and strike locations respectively.

The addition of general and specialized continuous controllers provide a means to modify the global beats per minute (bpm), preparatory height, onset/velocity/location standard deviation gains, and individual striking implement bpm, location, and velocity values as enumerated in

MIDI Continuous Controller chart located in the Appendix (Table 14). The global bpm value is a persistent setting accessible through the user interface that allows one to specify the frequency of quarter notes for repeating rudiments. For example, at 60 bpm, a repeating strike interval of 0.25 seconds will be observed regardless of preparatory height or velocity by utilizing timing compensation. The standard deviation gain values are applicable to patches that use stochastic models, whereby the user can specify the degree of uncertainty with respect to timing, velocity, and strike location, which serves to add a distinctive human quality to performances.

Calibration

The mechatronic drummer is entirely non-invasive to its companion percussive instrument. This is accomplished by first establishing a proximity to the instrument that ideally supports the full range of 4DOF motion. The second step is to run a calibration session from the user interface on the mechatronic drummer, which establishes the quadrature decoder zero position on each motion axis plane and attempts to discover the extent of the percussive surface. Discovery is achieved by dropping each striking implement independently near the proximal center of the surface and sweeping it outward while collecting elevation samples until sufficient resistance has been detected, such as encountering the rim of a drum or simply reaching the mechanical limit X axis motion. Depending on the selected class of percussion instrument, the calibration algorithm will evaluate the elevation samples in order to determine the contiguous surface of the instrument. Discontinuities in the surface such as cymbal edges or drum rims would be construed as the limit of X axis motion for the calibrated session as shown in Figure 73. After calibration, the mechatronic drummer can be controlled via MIDI and/or OSC

interfaces using normalized event data that maps directly to the previously discovered percussive surface.

When used with a rimmed instrument, such as a snare drum, the height and angle of the striking implements relative to the drum can be adjusted to support a rimshot. This offers the musician or composer a richer palate in terms of timbre that can greatly enhance a performance. The mechatronic system is unaware that it can strike the rim as a consequence of its strike alignment, so no additional configuration is needed.

Diagnostics

A diagnostic shell is available on the USB port or over the network that allows the operator to run a set of commands to explore and tune the state of the system. The command set includes reading the position of the quadrature encoders, adjusting PID controller gains, evaluating voltage and current levels, and interacting with other features of the system. In addition, there are a series of specialized calibration and data collection commands for performance analysis and optimization. A complete set of commands is listed in Table 7.

Table 7. Diagnostic command listing.

Command	Arguments	Description
b		Print button state
cal		Calibrate system
cfg	<ly|lx|ry|rx> <pdiv|mdty|mtim|mdiv> [value]	PID controller configuration
cls		Clear terminal
dac	<p0|p1|p2|p3>	Set DAC port
demo	<number>	Activate specific demonstration
dp	<none|p0|p1|p2|p3|kbdm|lcdm|misc>	Set debug port
et	<ly|ry>	Elevation timing measurements
fan	<on|off>	Set cooling fan state
gain	<int|ext> [value]	Set amplifier gain
i	<cm0|cm1|cm2|cm3|3v3|5>	Print current monitor measurement
info		Print system information
help	[command]	Print information on all or specific command
led	<g<0..3>|r<0..3>|l<r|g|b>|div>	Set LED parameter
ledm	<man|key|loc|duty>	Set LED display mode
loc		Print 4DOF location
mon	<us> <cmd>	Monitor command for specified interval
param	<r|w> <section> <key> [value]	Read or write persistent parameter
pid	<ly|lx|ry|rx> <pos|duty|loc|kp|ki|kd|bias|rst|en|flt|sd|to|pre|post> [value]	PID controller parameter
pos	<ly|lx|ry|rx> <pos> <us>	Set specified axis position
quit		Exit diagnostics
rep	<count> <us> <cmd>	Repeat command for specified interval
r	<velocity> <hits> <gap> <alt>	Roll rudiment
rr	<velocity> <vstd> <hits> <gap> <gstd> <x> <xstd> <alt>	Random roll rudiment
rs	<ly|ry> <velocity> <vstd> <hits> <gap> <gstd> <x> <xstd>	Random strike
rud	<number> <bpm>	Rudiment
s	<ly|ry> <velocity> [hits <gap>]	Strike
temp	<board|core>	Print temperature
tim	<ly|ry>	Timing measurements
tune	<ly|lx|ry|rx> <velocity>	Tune closed loop controller
tw	<speed>	Twirl sticks
vt	<ly|ry>	Velocity timing measurements
ver		Print version information
v	<ai0|ai1ai2|ai3|24|12|5|3>	Print voltage measurement
exp		Execute experimental code

Configuration

Configuration parameters are persisted on a MicroSD card, which also retains the boot loader, PetaLinux image, bare-metal firmware image, FPGA fabric configuration bitstream, and rudiment patch files. The parameters are stored in section files as partially listed in Figure 79. A subset of the parameters can be modified via the user interface, while others are statically configured during development and testing using a text editor. The MicroSD card uses the FAT32 file system and can be mounted on a computer for firmware upgrades, patch uploads, or configuration parameter modifications as needed.

```
#
# MechDrum(tm) Parameter File
#

[general]
patch = precision.patch

[midi]
bpm = 60
channel = 0

[lefty]
kp = 0.5
ki = 0.035
kd = 10.0
biasup = -15.0
biasdown = 5.0
biasdelay = 2000000
pidclkdivisor = 208333
pwmdutymargin = 75
pwmdutytimeout = 50000000
pwmclkdivisor = 15
slew = 1000
strikethreshold = 0.96
strikevelocitygain = 5.0
height = 0.6
timeout = 500
min = 0.0
max = 1.0
calstart = 500
calduty  = 50
caldelay = 100000
calcomp  = 5
caloc0 = 0.0403244682530018
caloc1 = 0.510255135113519
caloc2 = -0.520992291273044
```

Figure 79. Partial parameter section file listing.

Patches

Patch files are formatted as section files and contain both global and MIDI note meta-data as shown in Figure 80. The global data contains the patch name as displayed on the LCD panel when selected and the bpm upon which the meta-data is referenced. Each MIDI note contains one or more tuples that define the nature of the strike with respect to a stochastic model as an embodiment of a parameter vector. In the partial example listed in Figure 80, the entries include the striking events for a double stroke open roll. The first

data pair per line is the mean and standard deviation timing of the striking event. This is followed by two addition pairs containing the normalized mean and standard deviation of the velocity and location. Finally, the specific striking implement is defined as left or right. The last line for the note spans the remainder of the measure before the rudiment is repeated. There is no limit to the number of patches that can be created to perform time compensated rudiments or rhythmic patterns with varying degrees of precision.

```
#
# Human Drummer
#

[general]
name = Human
bpm = 110

[36]
name = Left Strike
repeat = 0
notes = 1
note0 = 0.0,0.0,1.0,0.0,0.5,0.0,0

[45]
name = Right Strike
repeat = 0
notes = 1
note0 = 0.0,0.0,1.0,0.0,0.5,0.0,1

[59]
name = Double Stroke Roll
repeat = 1
notes = 33
note0 = 0.0,0.04,1.0,0.3,0.5,0.3,0
note1 = 0.053,0.04,0.778,0.3,0.5,0.3,0
note2 = 0.118,0.04,0.390,0.3,0.5,0.3,1
note3 = 0.197,0.04,0.171,0.3,0.5,0.3,1
note4 = 0.252,0.04,0.575,0.3,0.5,0.3,0
note5 = 0.321,0.04,0.469,0.3,0.5,0.3,0
note6 = 0.379,0.04,0.321,0.3,0.5,0.3,1
note7 = 0.457,0.04,0.171,0.3,0.5,0.3,1
note8 = 0.511,0.04,0.826,0.3,0.5,0.3,0
note9 = 0.577,0.04,0.602,0.3,0.5,0.3,0
note10 = 0.640,0.04,0.366,0.3,0.5,0.3,1
note11 = 0.711,0.04,0.189,0.3,0.5,0.3,1
note12 = 0.779,0.04,0.527,0.3,0.5,0.3,0
note13 = 0.863,0.04,0.368,0.3,0.5,0.3,0
note14 = 0.921,0.04,0.303,0.3,0.5,0.3,1
note15 = 0.990,0.04,0.193,0.3,0.5,0.3,1
note16 = 1.063,0.04,0.802,0.3,0.5,0.3,0
note17 = 1.133,0.04,0.673,0.3,0.5,0.3,0
note18 = 1.206,0.04,0.387,0.3,0.5,0.3,1
note19 = 1.274,0.04,0.231,0.3,0.5,0.3,1
note20 = 1.342,0.04,0.502,0.3,0.5,0.3,0
note21 = 1.420,0.04,0.502,0.3,0.5,0.3,0
note22 = 1.485,0.04,0.364,0.3,0.5,0.3,1
note23 = 1.553,0.04,0.209,0.3,0.5,0.3,1
note24 = 1.629,0.04,0.753,0.3,0.5,0.3,0
note25 = 1.697,0.04,0.734,0.3,0.5,0.3,0
note26 = 1.768,0.04,0.371,0.3,0.5,0.3,1
note27 = 1.843,0.04,0.238,0.3,0.5,0.3,1
note28 = 1.907,0.04,0.588,0.3,0.5,0.3,0
note29 = 1.985,0.04,0.496,0.3,0.5,0.3,0
note30 = 1.046,0.04,0.431,0.3,0.5,0.3,1
note31 = 2.115,0.04,0.247,0.3,0.5,0.3,1
note32 = 2.184,0.04,0.0,0.3,0.5,0.05,0
```

Figure 80. Partial human patch section file listing.

User Interface

The mechatronic drummer provides a menu system for viewing and configuring operational parameters utilizing the LCD display and multi-function illuminated buttons as shown in Figure 81. Upon system power-up, the display briefly shows the version of the firmware that is currently running, after which it displays the IPV4 network address as a result of being assigned an address by a DHCP server on the network. By pressing the ‘select’ button repeatedly, the user is able to access a series of parameters that include temperature readings, SPMTE timecode, beats per minute, patch, calibration, and the MIDI channel. For parameters that are adjustable, such as beats per minute, patch, and the MIDI channel, the ‘-1/no’ and ‘+/yes’ buttons can be used to decrement or increment the respective values. Once a value has been modified an ‘*’ will appear next to the value that will only be persisted if the ‘enter’ button is pressed. The dual mode illuminated buttons also serve as status indicators for activity on the four planes of motion named ‘ly’, ‘lx’, ‘rx’, and ‘ry’, which equate to ‘left Y-axis’, ‘left X-axis’, ‘right X-axis’, and ‘right Y-axis’ respectively.

Figure 81. User interface display and buttons.

Summary

A lot of ground was covered in this chapter, which reflects the level of multi-discipline engineering effort that went into creating the mechatronic drummer. From a mechanical design perspective, a detailed review of voice coil actuators was presented along with a description of an initial prototype platform for evaluating a candidate VCA. The specifications and design for vertical and horizontal axis motion was also described. The requirement of thermal management was addressed and included a design for moving ambient air through the system by taking advantage of positive air pressure, and aluminum enclosure heat dissipation. Finally, a preliminary test fit and assembly was documented to illustrate a key milestone in the mechanical design flow.

With regard to the electronics, the primary building blocks were enumerated that directly map to the requirements in Table 4. The details of the field programmable gate array and its capabilities were reviewed in a manner that ties it to fundamental software requirements, such as performance, low-latency, extensibility, and interoperability. In order to control the voice coil actuators in the mechanical design, the quadrature encoders, H-bridge drivers, closed loop control, and strike/strain detection schemes were presented. The remaining hardware associated with the user interface, audio inputs, peripheral ports, power, temperature sensing, and development features rounded out the hardware design.

From a computer science perspective, the software was presented in functional blocks at a high-level. A description of the primary operating system and secondary thread of execution across two processing cores along with an inter-process communication

method was covered. The concept of abstracting motion control from the high-level software was discussed that includes the advantage of adopting such an architecture. The dedicated open sound control and musical instrument digital interface tasks were described along with how they interact with the abstraction library. In addition, a process for system calibration and configuration was presented that uses a file system for persistent configuration and patch libraries. Finally, the user interface was documented to show how a musician can interact with the system to read status information and adjust persistent configuration settings.

In the next chapter we will evaluate the mechatronic drummer in an effort to compare its capabilities to that of its human counterparts. Metrics such as speed, dynamics, and timing compensation will be presented in a manner that allows us to critically evaluate its capabilities as an expressive instrument. We will also explore some the live performances and related feedback that reinforce the general claim that the mechatronic drummer has met the main objectives of this research.

CHAPTER 6: System Evaluation

In order to evaluate a system you must have a basis for comparison. This can be in the form of specifications and/or representative output that the system must be able to generate. Straight forward comparative data can be measured values such as voltage and frequency. Complex comparisons may take the form of discrete signal correlations and spectral plots. In all cases, the goal is to evaluate a system in a manner that delivers a set of metrics that can relate the generated output to the baseline data with conclusions about the degree to which they agree or disagree. Moving beyond quantitative results, there are qualitative evaluations with respect to devices that interact with people. With regard to musical instruments, how does a rendering sound when compared to a similar human performance across timing, dynamics, and timbre? Is the system easy to setup and use, or is it cumbersome and unreliable? Both the musician/composer and audience have a role to play in evaluating the quality of the mechatronic drummer and its applicability to their respective interests.

In this chapter we will discover both quantitative and qualitative evaluations that speak to the effectiveness of the mechatronic drummer as an expressive instrument. Raw speed, playability, and feedback from live performances will help to form our impression of the mechatronic drummer and how it can be utilized as both an instrument and a platform for further study.

Overview

As discussed previously, the motion analysis of professional percussionists in a pilot study is the baseline by which the mechatronic drummer system has been evaluated in terms of accuracy in reproducing and exceeding human ability. Multi-axis velocity and timbre driven by location represent key components of being able to recreate human motion beyond onset timing. Timing compensation for preparatory height and velocity is also critical to maintaining an accurate beat, and when coupled with a stochastic model, can deliver unique performances with a distinctive human feel.

Live or pre-recorded streaming of position telemetry or events via OSC, or MIDI messages respectively, enables augmentation of traditional percussive performances. Position telemetry offers rich motion control with minimal position/strike latency, whereas MIDI events offer ubiquitous controller options with increasing strike latency depending on the preparatory height and velocity.

Speed

Since 2003, the World's Fastest Drummer/Extreme Sport Drumming competition has held annual competitions to select a champion[10] . Over the years the world record has been broken several times with the current leader being Tom "Tommy Gun" Grosset at 1208 strikes in 60 seconds that do not include bounces, presses, or buzz. This is an extraordinary accomplishment for a human performer. In contrast, the mechatronic drummer has achieved a maximum rate of 34 strikes per second, or 2040 strikes per

[10] World's Fastest Drummer, Extreme Sport Drumming®: https://bit.ly/2xJH2oa

minute per striking implement, for a total of 4080 strikes. This represents a speed increase of 338% over the current world record holder. Of course since the mechatronic drummer is a robot, it can maintain this speed indefinitely in contrast to its human counterpart.

Velocity

The bar chart in Figure 82 shows a comparison of pilot study participants and mechatronic drummer striking velocity with the resulting sound pressure level (SPL) in decibels as measured at a distance of 10 centimeters using 'A' weighting, which is the relative loudness as perceived by the human ear [110]. As is evident in the data, a strike is capable of exceeding human velocity when measured indirectly via SPL. This enhanced performance can be used by the musician and/or composer to potentially explore musical renderings that have previously been unattainable through conventional means. The chart also shows that the robotic drummer is capable of very low velocity strikes that demonstrate its ability to achieve a large dynamic range that is inclusive of human range. This is critically important for expressive performances and it represents a significant contribution when compared to other robotic percussion systems.

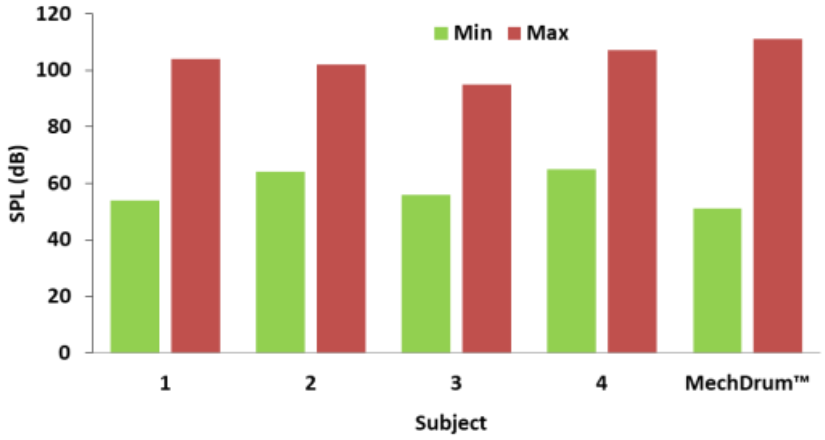


Figure 82. Pilot study SPL range comparison.

Location

The location of a strike directly affects the timbre of the sound [80]. This phenomenon is evident when listening to live or recorded performances. The scaleogram plot in Figure 83 shows a comparison between a human and mechatronic drummer strike at the center, middle radius, and outer rim of a snare drum when recorded with a Shure SM57 dynamic microphone. The primary frequency, secondary harmonic, and general overtones are visually comparable, which suggests that the mechatronic drummer can reproduce human strike timbre with respect to strike location.

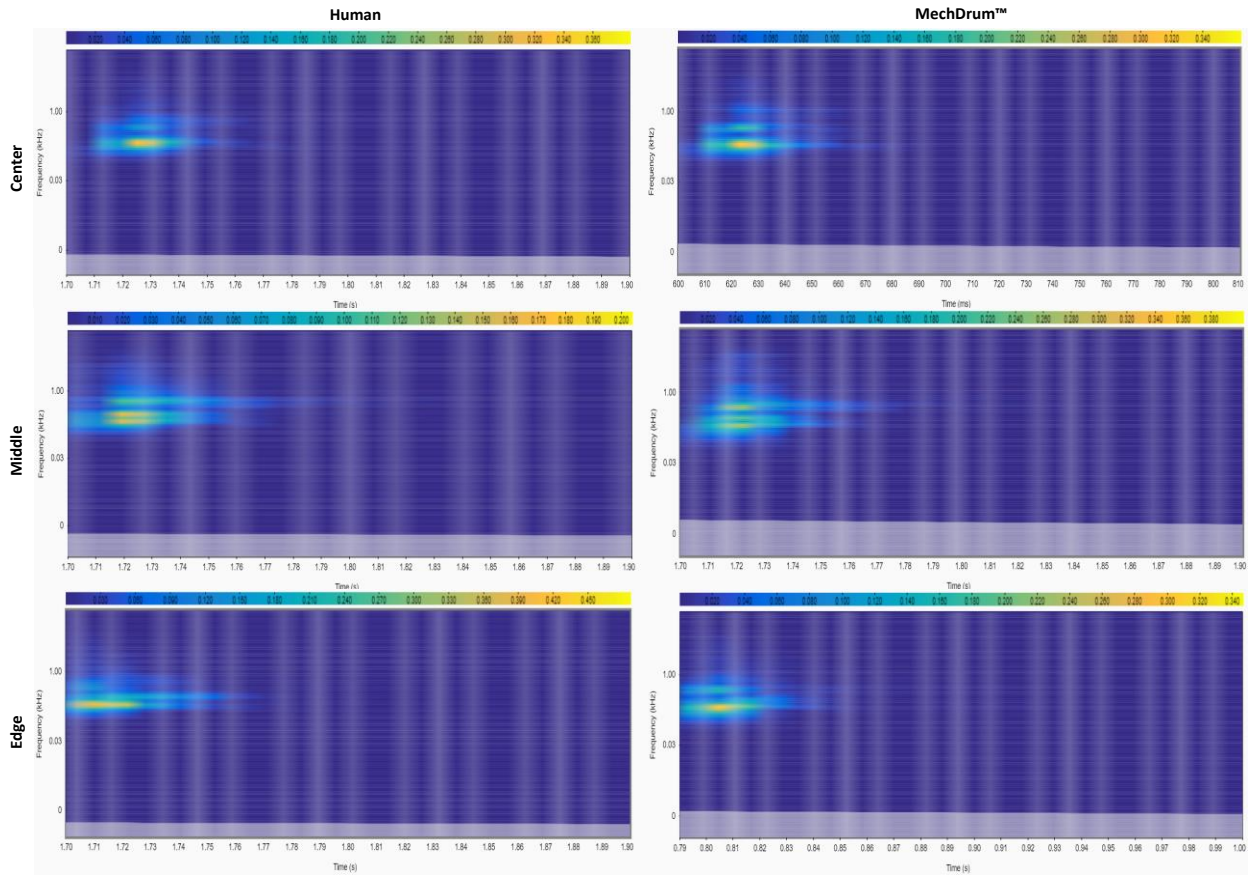


Figure 83. Strike location spectrum.

Bounce

Percussionists take advantage of the inherent bounce upon striking implement impact for a variety of rudiments such as rolls [73]. After impact, the percussionist reduces stick pressure to allow the rebound before applying downward pressure again for a secondary strike. In a similar manner, the mechatronic drummer is able to leverage a bounce by immediately controlling the force to the striking implement after an impact event has been detected. A comparison of this phenomenon can be seen in Figure 84. In the plot, the human position bounce results in two strikes with a normalized preparatory height

and timing interval that is closely approximated by the mechatronic drummer given similar input control input via OSC or the MIDI interfaces. The visible time lag and negative going location samples are the result of overall system latency (including network communications overhead) and physical drum surface deflection respectively.

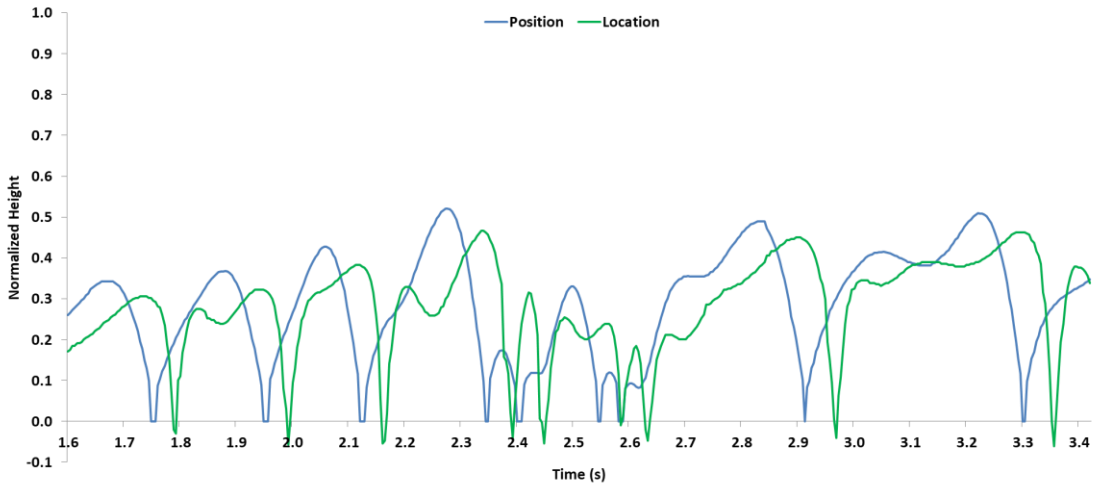


Figure 84. Position and location tracking.

Timing Compensation

The two dimensional surface shown in Figure 85 illustrates the relationship between velocity, elevation, and strike latency for a normalized elevation. By using the multi-variable curve fitting power function in Equation 17 along with coefficients derived from the aforementioned empirically measured data, the system is able to compensate for latency by adjusting the onset time to deliver a consistent beat as defined by the global beats per minute value. Referring to Equation 17, y is the timing compensation in seconds, a is a constant coefficient, E is the normalized elevation, b is a constant coefficient, V is the normalized velocity, and c is a constant coefficient. The empirically

derived values of a, b and c depicted in Figure 85 were 0.040324, 0.510255, and -0.520992 respectively.

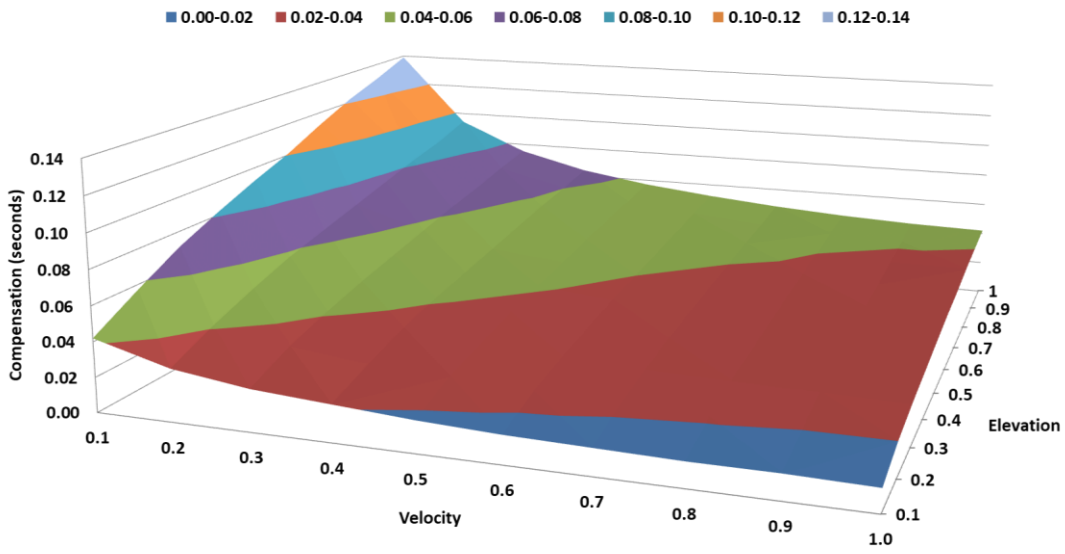


Figure 85. Timing compensation as a surface for normalized velocity and elevation.

$$y = aE^bV^c \quad \textbf{(17)}$$

Orchestral Rendering

An audio recording of an orchestral piece entitled “Shostkovic 10 mvt II” as performed by a human drum study participant was compared with an audio recording of the same piece being played by the mechatronic drummer using the 4DOF motion data from the original performance. The comparison was made by applying a cross-correlation of two discrete-time sequences, which measures the similarity between two signals whilst additionally identifying the time lag. The graph in Figure 86 shows a normalized peak of 1.0 at a time lag of zero seconds, which indicates that the audio recordings are in fact

highly correlated and thus implying the faithful reproduction of the human performance by the mechatronic drummer in the acoustic domain.

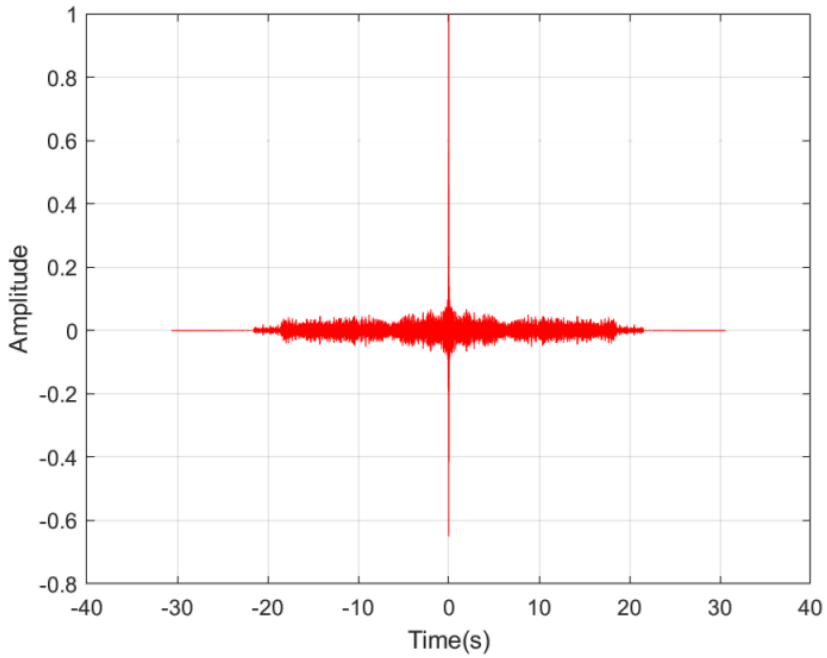


Figure 86. Cross-correlation of human and mechatronic drummer orchestral performance.

Musical Instrument Digital Interface

The MIDI protocol is inherently limited by bandwidth and latency due to its 31,250 bits per second (bps) serial bit rate, “note on” message overhead, source event detection time, and destination event reaction time [1]. As a consumer of the message, the mechatronic drummer receives it via the MIDI input port, decodes the message, and issues a strike from the primary processor core to the secondary core, which in turn initiates a strike in the PL fabric. At this point, the PL will respond by disabling the PID controller and applying scaled PWM current to the VCA via the H-bridge driver that will put the striking implement in motion until the actual strike is detected. This high-level chain of events is shown in Figure 87 with annotated average latencies for an implement tip

elevation from the drum surface of 1 inch. Clearly the primary contributor to latency is the mechanical subsystem, which is limited by its components and the physics of the system. As shown previously, the latency increases as a function of striking implement elevation and velocity, therefore a live MIDI performance must take this into account by anticipating the strike delay. For MIDI playback performances, one can use timing compensation in a score to minimize the inherent latency.

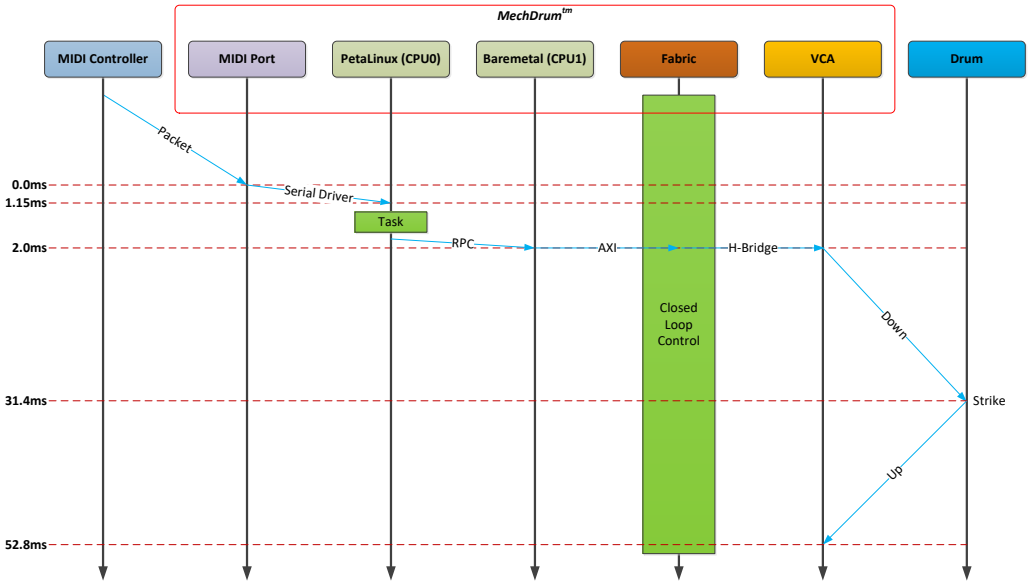


Figure 87. MIDI event latencies.

Open Sound Control

As opposed to the MIDI interface, OSC messages are received from the Ethernet port at a much higher data rate. In addition, the OSC messages contain discrete positions samples instead of a single trigger event, which results in minimal system latency. This latency is further reduced by detecting the whack event that disables the PID controller and applies scaled PWM current to the VCA via the H-bridge driver as illustrated in Figure 88. Instead of latency increasing as a function of elevation and velocity, it is constrained to a

relatively small constant of approximately 20 milliseconds that is generally tolerable by the musician in the context of a live performance.

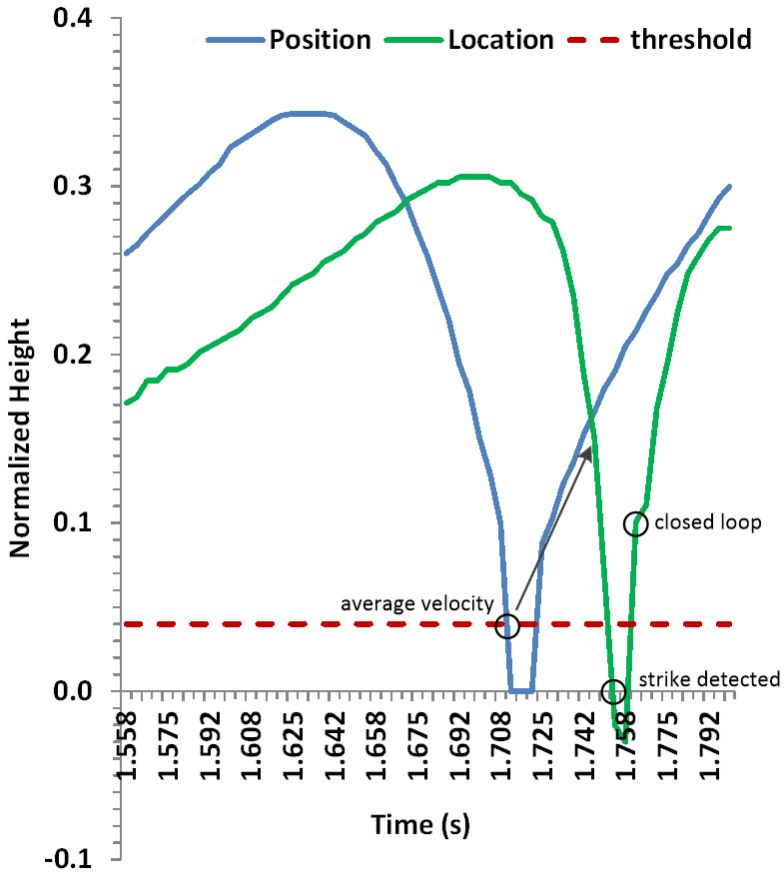


Figure 88. Whack and strike detection.

Referring to Figure 88, the position signal is a discrete series of timestamped normalized Y-axis samples that are received as OSC messages from the network. The location signal is a real-time normalized measurement of the physical position of the striking implement on the Y-axis as the closed loop controller follows the position set point stream. When the position signal crosses the static threshold, a single average velocity value is calculated from adjacent samples, that when multiplied by a static

constant gain, becomes the overriding duty cycle of the PWM channel. The resulting current through the VCA accelerates the striking implement location towards the drum head surface until a strike detection event occurs. Upon detecting this event, the overriding duty cycle is removed from the PWM channel, which allows the closed loop controller to restore the location to the position set point stream. As is evident from the location signal in the plot after the strike, the location dips below the surface as it deforms the drum head and subsequently rebounds with a high velocity until the point of visual discontinuity, where the closed loop controller resumes its tracking of the position set point stream. Note that the position and location delay is artificially amplified in Figure 88 due to the inherent network latency associated with the UDP location return packet and as such it does not truly represent the actual delay, which is markedly smaller.

The method described above is essentially a workaround for the fact that closed loop systems are effectively low pass filters. In the absence of such an approach, the VCA could not keep up with strike discontinuities within the data stream. Strike events would never be heard or at best become soft grazing hits on the surface of the drum head. The VCA itself is capable of the velocity required and therefore by determining the data streams strike intention as described above by using a static threshold, the system is able to drive the VCA current magnitude directly. This bypasses the low pass filter impact of the PID controller until the strike has been detected when there is plenty of time to resume tracking the position in the data stream.

Performances

The first official performance of the MechDrum[tm] occurred at the 2018 Margaret Guthman Musical Competition at Georgia Tech University in Atlanta as shown in Figure 89. The competition included an initial evaluation by a panel of judges with deep technical and artistic domain knowledge. After a presentation of the mechatronic drummer and a question/answer round, it was selected to compete in a final concert that showcased its capabilities in front of a live audience that included streaming to social media platforms. The performance was well received and the judges awarded the MechDrum[tm] with a Technical Achievement award.

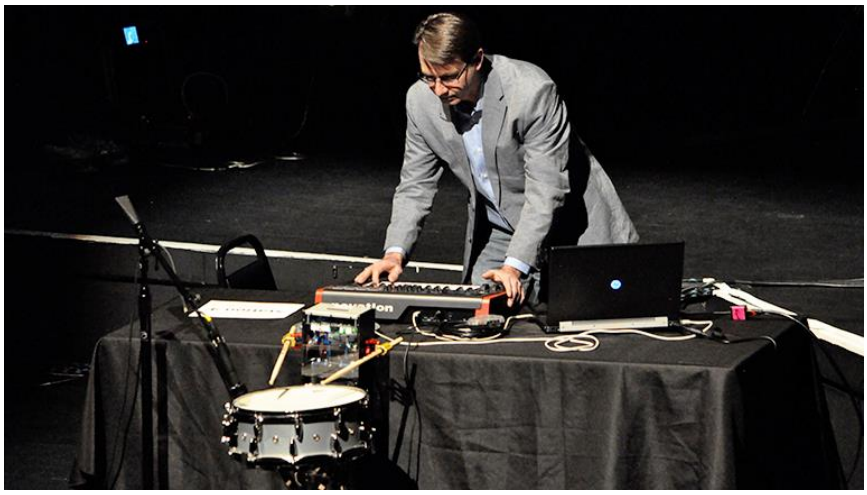


Figure 89. MechDrum[tm] performance at the 2018 Guthman musical instrument competition.

The second performance occurred at the 2018 New Interfaces for Musical Expression (NIME) conference that was held in Blacksburg, Virginia. This was the first performance of the Sonicpair #2 duo piece for the SDRdrum and MechDrum[tm] by Dr. Andrew Schloss, a Professor of Electronic and Computer Music, Musical Acoustics, and Ethnomusicologist at the University of Victoria. Andrew spent the better part of several

weeks exploring the capabilities and limitations of the mechatronic drummer prior to the concert in an effort to showcase the instrument in a live performance that highlighted its ability to follow his gestures on the SDRdrum directly and indirectly via parameter driven MIDI excerpts. The piece received immediate praise from the large NIME audience and was the focus of a yet to be published article by Annie Stevens, who is an accomplished percussionist and assistant professor at the School of Performing Arts at Virginia Tech University. The NIME conference included an exhibition of the SDRdrum and the MechDrum^{tm} as depicted in Figure 90, which received much attention from musicians, artists, and makers that asked many questions regarding how the system worked and what its capabilities were. A subset of attendees was also allowed to try the SDRdrum and MechDrum^{tm} so they could get first-hand experience as a performer. Both Andrew and I received positive feedback on the instruments and a genuine interest in seeing our work prosper in the future.

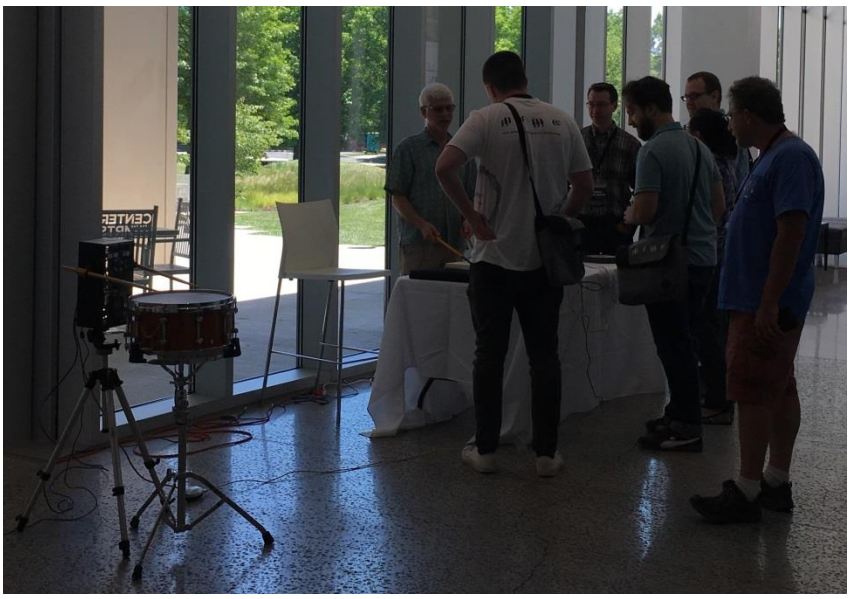


Figure 90. Exhibition at the 2018 NIME conference.

An additional performance of the Sonicpair #2 piece by Andrew was conducted at the Collective Response Concert that took place in Charlottesville, Virginia. This concert was very well received by the audience and fellow performers. In addition, the MechDrum[tm] was paired with the McBlare robotic bagpipe instrument for a remarkable duo performance by Dr. Roger Dannenberg, who is a professor of Computer Science and Art & Music at Carnegie Mellon University. The MIDI piece really pushed the MechDrum[tm] to its limit in terms of timing and dynamic virtuosity.

The third performance of Sonicpair #2 by Dr. Andrew Schloss occurred at the 2018 International Symposium of New Music in Curitiba, Parana, Brazil. This festival of contemporary music included concerts, sound and art exhibitions, and lectures. Andrew presented the SDRdrum and MechDrum[tm] in both a lecture and concert setting. As shown in Figure 91, Andrew is demonstrating the capabilities of the instruments.

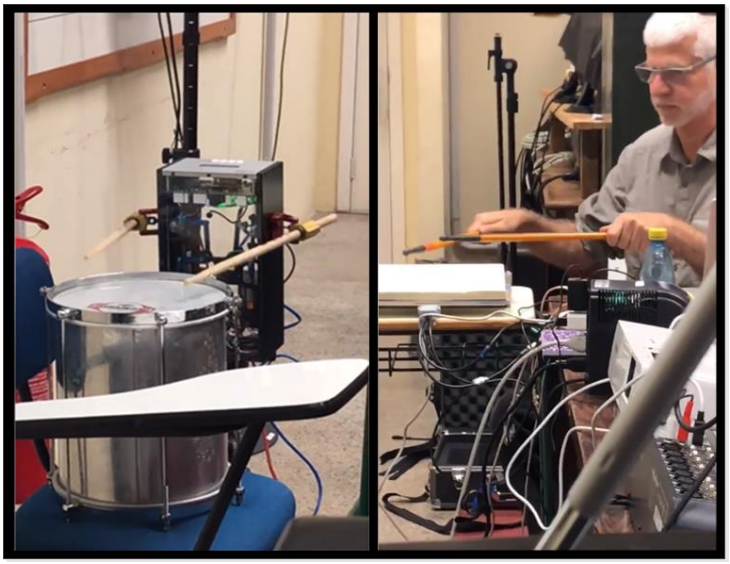


Figure 91. Dr. Andrew Schloss providing a demonstration during his lecture at the International Symposium of New Music in Brazil.

The latest performance of Sonicpair #2 as shown in Figure 92 took place at the 2018 Interactive Art, Science, and Technology symposium that was held at Lethbridge University, in Alberta, which also included a student exhibition. As before, the performance was very well received and included many comments from attendees and fellow performers on the quality of the renderings and extraordinary nature of the instrument, and its construction. Several of the attendees and Lethbridge School of Music faculty played the instrument during the exhibition and commented on the crisp response, ease of use, and natural sound of the MechDrum^{tm} when coupled with a world class drum. Many inquired as to when it will be available for purchase and if it would be possible to build one from a set of plans. A professional jazz drummer was also very interested in the MechDrum^{tm}, and after playing the instrument, he commented that he would be able to use it in a professional setting given the quality and acceptable latency of the mechatronic drummer.

Figure 92. 2018 Interactive Art, Science, and Technology symposium at Lethbridge University.

Summary

As this chapter demonstrates, the mechatronic drummer has been evaluated from both a quantitative and qualitative perspective. The raw speed of the instrument greatly exceeds the capabilities of a human and the dynamics are inclusive of the professional percussionists range. The ability to alter the timbre of a performance in real-time is also comparable and its bounce capability matches the motion of drummer's strike. With regard to rudiments, the mechatronic drummer is able to perform timing compensation to accommodate real-time elevation and velocity changes. The MechDrum^{tm} is also able to reproduce orchestral pieces to a high degree of precision as indicated by a cross-correlation of recordings. Both the MIDI and OSC interfaces offer low-latency control over the mechatronic drummer's articulations with superior performance in an OSC

environment. Finally, multiple conference performances and exhibitions have displayed interest and enthusiasm for the instrument, which speaks to its qualitative appeal to musicians, artists, educators, and the maker community.

The next chapter will summarize a set of conclusions about the research and highlight areas of future work.

CHAPTER 7: Conclusion

Capturing the motion associated with percussion instrument performances requires an infrastructure of equipment, software, analytical processes, skilled musicians, and a formal calibration technique. The results tangibly illustrate the ability to evaluate a performance from multiple vantage points in a non-invasive manner with respect to the musician and the instrument. It is also evident that meaningful motion capture can be accomplished with minimal resources and at a relatively low cost. As a consequence, a real world dataset has been established for current and future research in the field of human motion. By creating models of human motion, electromechanical devices can begin to move in a more fluid and relatable manner, which is the ultimate goal of this work. Designing algorithms and electromechanical systems that can reproduce a truly human-informed performance represents an exciting new branch of research that extends an existing body of work.

Through the novel use of a calibrated data-acquisition system, motion extraction, and analysis methods, performances have been distilled into parameter vectors from a pilot user study using four expert level percussionists that can be used to describe independently unique performances. The pragmatic approach to data acquisition enables the extraction of parameters such as average velocity and position of a strike in an unencumbered manner, which would not be possible by using a typical percussive MIDI trigger or other types of non-positional impact sensors. To render a convincing robotic performance however, it is clear that a mechatronic percussionist must be designed and

implemented with the appropriate degrees of freedom, spatial resolution, and agility to faithfully reproduce a parameter-vector-driven stochastic model.

Extending the approach discussed herein across the entire set of rudiments and other fundamental percussive articulations would result in a library of parameter vectors, or "patches" that can be used to render unique and life-like performances on demand. Although this represents a daunting manual task, it is possible to automate many, if not all of the steps illustrated in this document and its references. The use of a passive 3D motion-tracking system coupled with an enhanced version of the stochastic analysis application would make it trivial to create large parameter-vector libraries that could include multiple percussionists as well as generalizations. A number of open questions remain, including whether tails and endings have notable features. Answering this type of question will require the acquisition and analysis of considerably more data. In addition, acquiring data in a non-laboratory setting could add further insight into statistical vectors associated with live performances as musicians tend to "rise to the occasion" in front of a live audience.

This research has shown that the use of VCAs for percussion robots is a viable alternative to traditional solenoids and DC motors. Although VCAs are expensive, the combination of high acceleration, low latency, low hysteresis, and precision positioning result in an effective means to control striking implement tip motion. Through comparative analysis and experimentation it has been demonstrated that the application of VCAs and associated control algorithms can successfully contribute to the evolution of percussion robotics.

The research and development that was conducted in the context of this dissertation has demonstrated that a robotic percussionist using voice coil actuators with proper software, hardware, and mechanical infrastructure, can yield high quality renderings that closely match, and in some cases, exceed human percussionist abilities [72]. Potential use cases include analog drumming for musicians with varying degrees of disabilities, live performance augmentation over a distance, pedagogical studies, percussion sample library creation, and playback of pre-recorded motion performances in a recording studio or artistic installation. Although VCAs are complex devices to control, it is clear that their peak force and velocity can be used to great effect in a musical context.

Future Work

With regard to improved performance, larger VCAs should be used to increase the force and speed of X axis motion. Although this would necessitate mechanical changes to the current prototype, the general approach can largely remain the same. In addition, tighter tolerances using ball joints and other refined linkage strategies along the X axis force transfer path would serve to improve closed loop stability and help reduce noise stemming from the apparatus. Adding software support for multiple striking implement types along with percussion instrument classes would serve to extend the timbre range and of the robotic drummer. Finally, an effort to reduce the cost of the current prototype design would make the robot accessible to individuals and organization that wish to use the instrument in a variety of new and interesting settings.

Although in-depth research in the areas of micro-timing, ethnomusicology, and machine learning were beyond the scope of the research presented here, the robotic drummer is well positioned as an experimental platform for further exploration. By developing new algorithms and training the robot with relevant data sets, new and exciting automated accompaniment should begin to emerge that would have a positive impact on robotic music and musicianship [111, 112].

Bibliography

[1] *The Complete MIDI 1.0 Detailed Specification,* La Habra: MIDI Manufacturers Association, 2014.

[2] M. O'Dair, "Different Every Time: The Authorised Biography of Robert Wyatt," *Popular Music,* vol. 34, no. 3, p. 531, 2015.

[3] C. Pierella, F. Abdollahi, A. Farshcjiansadegh, J. Pedersen, E. B. Thorp, F. A. Mussa-Ivaldi and M. Casadio, "Remapping residual coordination for controlling assistive devices and recovering motor functions," *Neuropsychologia,* vol. 79, no. B, pp. 364-376, 2015.

[4] J. d. Millan, R. Rupp, G. R. Muller-Putz, R. Murray-Smith, C. Giugliemma, M. Tangermann, C. Vidaurre, F. Cincotti, A. Kubler, R. Leeb, C. Neuper, K. -R. Muller and D. Mattia, "Combining brain-computer interfaces and assistive technologies: state-of-the-art and challenges," *Frontiers in Neuroscience,* vol. 4, p. 161, 2010.

[5] T. Luca, "Therapeutic Application of The Kokas-Method in Music Therapy for People with Severe Disabilities," *Journal of Russian & East European Psychology,* vol. 55, no. 1, pp. 85-105, 2018.

[6] S. Shannon, "The Therapeutic Power of Music," in *Handbook of Complementary and Alternative Therapiesin Mental Health,* Colorado, Academic Press, 2002, pp. 517-537.

[7] T. Lomas, "Positive art: Artistic expression and appreciation as an exemplary

vehicle for flourshing.," *Review of General Psychology,* vol. 20, no. 2, pp. 171-182, 2016.

[8] T. Hays and V. Minichiello, "The meaning of music in the lives of older people: a qualitative study," *Psychology of Music,* vol. 33, no. 4, pp. 437-453, 2005.

[9] H. Riley, R. B. MacLeod and M. Libera, "Low Latency Audio Video: Potentials for Collaborative Music Making Through Distance Learning," *National Association for Music Education,* vol. 34, no. 3, pp. 15-23, 2016.

[10] Yamaha Corporation of America, "Yamaha Unveils "Remote Live"," *Music Trades,* June 2011.

[11] Q. Yang and G. Essl, "Evaluating Gesture-Augmented Keyboard Performance," *Computer Music Journal,* vol. 38, no. 4, pp. 68-79, 2014.

[12] Information Gatekeepers, Inc., "Legendary Pianist Byron Janis Conducts First-Ever Master Class Linking the U.S. and Russia Via Disklavier Remote Lesson," *Russian Telecom Newsletter,* pp. 9-10, September 2016.

[13] M. Ballard, "Developing the Total Percussionist," *The Journal of the Canadian Band Association,* vol. 11, no. 1, pp. 19-20, 2012.

[14] Z. Paris, "Making the Connection A Guide to Big Band Drumming for the Jazz Educator," *The Canadian Music Educator,* vol. 59, no. 1, pp. 38-40, 2017.

[15] E. Stamatatos and G. Widmer, "Automatic identification of music performers with learning ensembles," *Artificial Intelligence,* vol. 165, pp. 35-56, 2005.

[16] G. Widmer and W. Goebl, "Computational Models of Expressive Music Performance: The State of the Art," *Journal of New Music Research,* vol. 33, no. 3,

pp. 203-216, 2004.

[17] K. Ericsson and A. Lehmann, "Expert and exceptional performance: evidence of maximal adaptation to task constraints.," *Annual Review of Psychology,* vol. 47, pp. 273-305, 1996.

[18] M. Miura, "Inter-Player Variability of a Roll Performance on a Snare-Drum Performance," in *International Conference of the Forum Acusticum*, Budapest, 2005.

[19] P. McAleer, F. E. Pollick, S. A. Love, F. Crabbe and J. M. Zacks, "The role of kinematics in cortical regions for continuous human motion perception," *Cognitive, Affective, & Behavioral Neuroscience,* vol. 14, no. 1, pp. 307-318, 2014.

[20] A. S. Etemad, A. Arya, A. Parush and S. DiPaola, "Perceptual validity in animation of human motion," *Computer Animation and Virtual Worlds,* vol. 27, no. 1, pp. 58-71, 2016.

[21] "Percusive Arts Society," [Online]. Available: http://www.pas.org/index.aspx. [Accessed 27 December 2014].

[22] L. Holm, O. Karampela, F. Ullen and G. Madison, "Executive control and working memory are involved in sub-second repetitive motor timing," *Experimental Brain Research,* vol. 235, no. 3, pp. 787-798, 2017.

[23] A. Friberg and A. Sundstrom, "Swing Ratios and Ensemble Timing in Jazz Performance: Evidence for a Common Rythmic Pattern," *Music Perception,* vol. 19, no. 3, pp. 333-349, 2002.

[24] A. Danielsen, C. H. Waadeland, H. G. Sundt and M. A. Witek, "Effects of

instructed timing and tempo on snare drum sound in drum kit performance," *Acoustical Society of America,* vol. 138, no. 4, pp. 2301-2316, 2015.

[25] M. Davies, G. Madison, S. Pedro and F. Gouyon, "The Effect of Microtiming Deviations on the Perception of Groove in Short Rhythms," *Music Perception,* vol. 30, no. 5, pp. 497-510, 2013.

[26] M. Wright and E. Berdahl, "Towards Machine Learning of Expressive Microtiming in Brazilian Drumming," in *International Computer Music Conference*, New Orleans, 2006.

[27] V. Iyer, "Embodied Mind, Situated Cognition, and Expressive Microtiming in African-American Music," *Music Perception,* vol. 19, no. 3, pp. 387-414, 2002.

[28] P. Guralnick, Sam Phillips: The Man Who Invented Rock 'n' Roll, Little, Brown and Company, 2015.

[29] R. Van Rooyen and G. Tzanetakis, "Pragmatic Drum Motion Capture System," in *New Interfaces for Musical Expression*, Baton Rouge, 2015.

[30] X. M. Feng, J. Z. Duan, Y. Fu, L. A. Sun and W. D. Zhang, "The technology and application of voice coil actuator," in *Second International Conference on Mechanic Automation and Control Engineering*, Hohhot, 2011.

[31] R. Van Rooyen, A. Schloss and G. Tzanetakis, "Snare Drum Performance Motion Analysis," in *New Instruments for Musical Expression*, Brisbane, 2016.

[32] A. Bouenard, M. M. Wanderley, S. Gibet and F. Marandola, "Virtual Gesture Control and Synthesis of Music Performances: Qualitative Evaluation of Synthesized Timpani Exercises," *Computer Music Journal,,* vol. 35, no. 3, pp. 57-

72, 2011.

[33] R. Van Rooyen, A. Schloss and G. Tzanetakis, "Voice Coil Actuators for Percussion Robotics," in *New Instruments for Musical Expression*, Copenhagen, 2017.

[34] A. Kapur, "A History of Robotic Musical Instruments," in *International Computer Music Conference*, Barcelona, 2005.

[35] A. R. Tindale, A. Kapur, G. Tzanetakis, P. Driessen and A. Schloss, "A Comparison of Sensor Strategies for Capturing Percussive Gestures," in *International Conference of New Interfaces for Musical Expression*, Vancouver, 2005.

[36] M. Collicut, C. Casciato and M. Wanderly, "From Real to Virtual: A Comparison of Input Devices for Percussion Tasks," in *International Conference of New Interfaces for Musical Expression*, Pittsburgh, 2009.

[37] S. Dahl, M. Grossbach and E. Altenmuller, "Effect of Dynamic Level in Drumming: Measurements of Striking Velocity, Force, and Sound Level," in *Conference of the Forum Acusticum*, Denmark, 2011.

[38] S. Dahl and E. Altenmuller, "Motor Control in Drumming: Influence of movement pattern on contact force and sound characteristics," in *International Proceedings of Acoustics*, Paris, 2008.

[39] O. Gillet and G. Richard, "ENST-Drums: an extensive audio-visual database for drum signals processing," GET / ENST, CNRS LTC1, Paris, 2006.

[40] A. Bouenard, S. Gibet and M. M. Wanderley, "Hybrid Inverse Motion Control for

Virtual Characters Interacting with Sound Synthesis - Application to Percussion Motion," *The Visual Computer,* vol. 28, no. 4, pp. 357-370, 2012.

[41] A. Bouenard, M. M. Wanderley and S. Gibet, "Gesture Control of Sound Synthesis: Analysis and Classification of Percussion Gestures," *Acta Acustica united with Acustica, Special issue on Natural and Virtual Instruments: Control, Gesture and Player Interaction,* vol. 96, no. 4, pp. 668-677, 2010.

[42] H. Honing and W. Bas De Haas, "Swing Once More: RelatingTiming and Tempo in Expert Jazz Drumming," *Music Perception,* vol. 25, no. 5, pp. 471-476, 2008.

[43] A. Danielsen, Musical Ryhthm in Age of Digital Reproduction, Oslo: Ashgate Publishing Limited, 2010.

[44] J. Fruhauf, Kopiez, Reinhard and F. Platz, "Music on the Timing Grid: The Influence of Microtiming on the Perceived Grove Quality of a Simple Drum Pattern Performance," *Musicae Scientiae,* vol. 17, no. 2, pp. 246-260, 2013.

[45] L. Naveda, F. Gouyon, C. Guedes and M. Leman, "Microtiming Patterns and Interactions with Musical Properties in Samba Music," *Journal of New Music Research,* vol. 40, no. 3, pp. 449-460, 2011.

[46] S. Dixon, W. Goebl and E. Cambouropoulos, "Perceptual Smoothness of Tempo in Expressively Performed Music," *Music Perception,* vol. 23, no. 3, pp. 195-214, 2006.

[47] A. Berndt and T. Hahnel, "Expressive Musical Timing," in *Audio Mostly: 4th Conference on Interaction with Sound - Sound and Emotion,* Glasgow, 2009.

[48] A. Friberg, R. Bresin and J. Sundberg, "Overview of the KTH Rule System for

Musical Performance," *Advances in Cognitive Psychology*, vol. 2, no. 2-3, pp. 145-161, 2006.

[49] P. N. Juslin, A. Friberg and R. Bresin, "Toward a Computational Model of Expression in Performance: The GERM Model.," *Musicae Scientiae Special Issue*, vol. 5, no. 1, pp. 63-122, 1 September 2001.

[50] S. Kemper and S. Barton, "Mechatronic Expression: Reconsidering Expressivity in Music for Robotic Instruments," in *New Interfaces for Musical Expression*, Blacksburg, 2018.

[51] E. Rasanen, O. Pulkkinen, T. Virtanen, M. Zollner and H. Henning, "Fluctuations of Hi-Hat Timing and Dynamics in a Virtuoso Drum Track of a Popular Music Recording," *PLOS ONE*, vol. 10, no. 6, 2015.

[52] J. De Souza, "Orchestra Machines, Old and New," *Organised sound: an international journal of music technology*, vol. 23, no. 2, pp. 156-166, 2018.

[53] G. Hoffman and G. Weinberg, "Shimon: an interactive improvisational robotic marimba player," in *ACM Conference on Human Factors in Computing Systems*, Atlanta, 2010.

[54] A. Kapur, M. Darling, D. Diakopoulos and J. W. Murphy, "The Machine Orchestra: An Ensemble of Human Laptop Performers and Robotic Musical Instruments," *Computer Music Journal*, vol. 34, no. 4, pp. 49-63, 2011.

[55] L. Maes, G.-W. Raes and T. Rodgers, "The Man and Machine Robot Orchesta at Logos," *Computer Music Journal*, vol. 35, no. 4, pp. 28-48, 2011.

[56] J. Murphy, D. Carnegie and A. Kapur, "Little Drummer Bot: Building, Testing, and

Interfacing With a New Expressive," in *ICMC*, Athens, 2014.

[57] G. Wienberg and S. Driscoll, "The interactive robotic percussionist: new developments in form, mechanics, perception and interaction design," in *Proceedings of the ACM/IEEE international conference on Human-robot interaction*, Arlington, 2007.

[58] G. Hoffman and G. Weinberg, "Interactive improvisation with a robotic marimba player," *Autonomous Robots*, vol. 31, no. 2-3, pp. 133-153, 2011.

[59] Wikipedia, "Orchestrion (album)," Wikipedia, 26 October 2018. [Online]. Available: https://en.wikipedia.org/wiki/Orchestrion_(album). [Accessed 29 October 2018].

[60] P. Metheny, "About Orechstrion," Pat Metheny, November 2009. [Online]. Available: http://www.patmetheny.com/orchestrioninfo/. [Accessed 29 October 2018].

[61] A. Bouenard, S. Gibet and M. M. Wanderley, "Enhancing the Visualization of Percussion Gestures by Virtual Character Animation," in *International Conference of New Interfaces for Musical Expression*, Genova, 2008.

[62] T. Fei, X. Chen, C.-X. Jiang, L. Zhou and Z.-T. Liu, "Performance control system of dulcimer music-playing robot," in *Asian Control Conference*, Gold Coast, 2017.

[63] J. McBean and C. Breazeal, "Voice coil acutators for human-robot interaction," in *International Conference on Intelligent Robots and Systems*, Sendai, 2004.

[64] A. D. Udai, D. H. Salunkhe, A. Dutta and S. Mukherjee, "Force/Position Control of 3 DOF Delta Manipulator with Voice Coil Actuator," in *Proceedings of the*

Advances in Robotics, New Delhi, 2017.

[65] D. Wang, L. Wang, Y. Zhang, P. Lv, Y. Sun and J. Xiao, "Preliminary study on minature laser manipulation robotic device for tooth crown preparation," *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 10, no. 4, pp. 482-494, 2014.

[66] W. T. Latt, R. C. Newton, M. Visentini-Scarzanella, C. J. Payne, D. P. Noonan, J. Shang and G.-Z. Yang, "A Hand-held Instrument to Maintain Steady Tissue Contact during Probe-Based Confocal Laser Endomicroscopy," *IEEE Transactions on Bimedical Engineering*, vol. 58, no. 9, pp. 2694-2703, 2011.

[67] Y.-b. Bang, K.-m. Lee, J. Kook, W. Lee and I.-s. Kim, "Micro parts assembly system with micro gripper and RCC unit," *IEEE Transactions on Robotics*, vol. 21, no. 3, pp. 465-470, 2005.

[68] L. Kerhuel, S. Viollet and N. Franceschini, "Steering by Gazing: An Efficient Biomimetic Control Strategy for Visually Guided Micro Aerial Vehicles," *IEEE Transactions on Robotics*, vol. 26, no. 2, pp. 307-319, 2010.

[69] D. Ben-Dov and S. E. Salcudean, "A force-controlled pneumatic actuator," *IEEE Transactions on Robotics*, vol. 11, no. 6, pp. 906-911, 1995.

[70] L. Dahl, J. Bellona, L. Bai and A. LaViers, "Data-Driven Design of Sound for Enhancing the Perception of Expressive Robotic Movement," in *Proceedings of the 4th International Conference on Movement Computing*, London, 2017.

[71] R. Rajkumar, D. de Niz and M. Klein, Cyber-Physical Systems, Addison-Wesley Professional, 2016.

[72] *Margret Guthman Musical Instrument Competition.* [Performance]. Georgia Tech, 2018.

[73] D. Gopinath and G. Weinberg, "A generative physical model appraoch for enhancing the stroke palette for robotic drummers," *Robotics and Autonomous Systems*, vol. 86, pp. 207-215, 2016.

[74] A. Z. Hajian, D. S. Sanchez and R. D. Howe, "Drum Roll: Increasing Bandwidth Through Passive Impedance Modulation," in *International Conference on Robotics and Automation*, Albuquerque, 1997.

[75] E. Schoonderwaldt, N. Rasamimanana and F. Bevilacqua, "Combining Accelerometer and Video Camera: Reconstruction of Bow Velocity Profiles," in *International Conference of New Interfaces for Musical Expression*, Paris, 2006.

[76] S. Dahl, "Striking Movements: A Survey of Motion Analysis of Percussionists," Acoustic Science and Technology, Denmark, 2012.

[77] H. Kawakami, Y. Mito, R. Watanuma and M. Marumo, "Analysis of Drum Player's Motion," in *International Proceedings of Acoustics*, Paris, 2008.

[78] R. Van Rooyen, A. Schloss and G. Tzanetakis, "Snare Drum Motion Capture Dataset," in *New Interfaces for Musical Expression*, Baton Rouge, 2015.

[79] Wikipedia, "Normal distribution," Wikipedia, 10 October 2018. [Online]. Available: https://en.wikipedia.org/wiki/Normal_distribution. [Accessed 29 October 2018].

[80] A. R. Tindale, A. Kapur, G. Tzanetakis and I. Fujinaga, "Retrieval of Percussion Gestures Using Timbre," in *International Conference on Music Information*

Retrieval, Barcelona, 2004.

[81] T. D. Rossing, Z. Ingolf and D. O. Fystrom, "Acoustics of Snare Drums," *Acoustical Society of America*, vol. 92, no. 1, p. 84, 1992.

[82] A. Tindale, "Classification of snare drum sounds using neural networks," McGill University, Montreal, 2004.

[83] F. C. Manning and M. Schutz, "Trained to keep a beat: movement-related enhancements to timing perceptionin percussionists and non-percussionists," *Psychological Research*, vol. 80, no. 4, pp. 532-542, 2016.

[84] O. J. Lodge, "Improvements in magnetic telegraphy and telephony". UK Patent GB189729505, 1898.

[85] D. Collins, "Voice Coil Actuator Basics," Linear Motion Tips, A Design World Resource, 25 August 2016. [Online]. Available: http://www.linearmotiontips.com/voice-coil-actuator-basics/. [Accessed 13 November 2016].

[86] X. Song, J. Fang and B. Han, "High-Precision Rotor Position Detectionor High-Speed Surface PMSM Drive Based on Linear Hall-Effect Sensors," *IEEE Transactionson Power Electronics*, vol. 31, no. 7, pp. 4720-4731, 2015.

[87] J. W. Mckeage, R. M. J. Williams, B. P. Ruddy, P. M. F. Nielsen and A. J. Taberner, "Sensorless position control of voice-coil motors for needle-free jet injection," in *6th International Conference on Automation, Robotics and Applications*, Queenstown, 2015.

[88] Wikipedia, "PID Controller," Wikipedia, 13 November 2016. [Online]. Available:

https://en.wikipedia.org/wiki/PID_controller. [Accessed 3 November 2016].

[89] J. Long, J. W. Murphy, J. Kapur and D. Carnegie, "A Methodology for Evaluating Robotic Striking Mechanisms for Musical Contexts," in *New Instruments for Musical Expression*, Baton Rouge, 2015.

[90] A. Kapur, Trimpin, E. Singer, A. Suleman and Tzanetakis, George, "A comparison of solenoid-based strategies for robotic drumming," in *International Computer Music Association*, Copenhagen, 2007.

[91] C. Hardy, "The Basics of Tuning PID Loops," Integrated Systems, 9 April 2014. [Online]. Available: http://innovativecontrols.com/blog/basics-tuning-pid-loops. [Accessed 25 November 2016].

[92] J. Murphy, A. Kapur and D. Carnegie, "Better Drumming Through Calibration: Techniques for Pre-Performance Robotic Percussion Optimization," in *New Interfaces for Musical Expression*, Michigan, 2012.

[93] S. R. Ness, S. Trail, P. F. Driessen, A. W. Schloss and G. Tzanetakis, "Music Information Robotics: Coping Strategies for Musically Challenged," in *International Society for Music Information*, Miami, 2011.

[94] M. L. Waddell, J. M. Fine, A. D. Likens, E. L. Amazeen and P. G. Amazeen, "Perceived heaviness in the context of Newton's Second Law: Combined effects of muscle activity and lifting kinematics.," *Journal of Experimental Psychology: Human Perception and Performance*, vol. 42, no. 3, pp. 363-374, 2016.

[95] S. Palanisamy, V. Tunakova and J. Militky, "Fiber-based structures for electromagnetic shielding - comparison of different materials and textile

structures," *Textile Research Journal,* vol. 88, no. 17, pp. 1992-2012, 2018.

[96] Xilinx, Inc., *Zynq-7000 SoC Data Sheet: Overview,* San Jose: Xilinx, inc., 2018.

[97] Avago Technologies, *HEDS-9040/9140: Three Channel Optical Incremental Encoder Modules,* Avago Technologies., 2014.

[98] Avago Technologies, *HEDS-51X0/61X0 Series, HEDG-512X/612X Series, HEDM-512X/61XX Series: Two and Three Channel Codewheels,* Avago Technologies, 2009.

[99] A. F. C. Hamilton, K. E. Jones and D. M. Wolpert, "The scaling of motor noise with muscle strength and motor unit number in humans," *Experimental Brain Research,* vol. 157, no. 4, pp. 417-430, 2004.

[100] Texas Instruments Incorporated, *LMD18200 3A, 55V H-Bridge,* Dallas: Texas Instruments Incorporated, 2013.

[101] V. Alfaro and R. Vilanova, Model-Reference Robust Tuning of PID Controllers, Switzerland: Springer International Publishing, 2016.

[102] Newhaven Display International, Inc., *NHD-0216K1Z-NS(RGB)-FBW-REV1: Character Liquid Crystal Display Module,* Elgin: Newhaven Display International, Inc., 2016.

[103] W. Gay, Custom Raspberry Pi Interfaces, Ontario: Apress Media, LLC, 2017.

[104] Texas Instruments Incorporated, *LM43603 3.5-V to 36-V, 3A Synchronous Step-Down Voltage Converter,* Dallas: Texas Instruments Incorporated, 2017.

[105] Microchip Technology, Inc., *MCP9700/9700A:,* 2016: Microchip Technology, Inc.,

Low-Power Linear Active Thermistor ICs.

[106] Xilinx Corporation, *Vivado Design Suite User Guide (UG893),* San Jose: Xilinx Corporation, 2017.

[107] Xilinx Corporation, *PetaLinux Tools Documentation,* San Jose: Xilinx Corporation, 2017.

[108] Xilinx Corporation, *OpenAMP Framework for Zynq Devices,* San Jose: Xilinx Corporation, 2017.

[109] M. Wright, A. Freed and A. Momeni, "OpenSound Control: State of the Art 2003," in *New Interfaces for Musical Expression*, Montreal, 2003.

[110] Wikipedia, "A-weighting," Wikipedia, 17 October 2018. [Online]. Available: https://en.wikipedia.org/wiki/A-weighting. [Accessed 29 October 2018].

[111] M. Krzyzaniak, "Timbral Learning for Musical Robots," Arizona State University, Tempe, 2016.

[112] M. Krzyzaniak, "Interactive Learning of Timbral Rhythms for Percussion Robots," *Computer Music Journal,* vol. 42, no. 2, pp. 35-51, 2018.

Appendices

Appendix A

Research and Development Process

The research and development that was conducted in the context of this dissertation took place from September 2013 to December 2018. The research started with a literature review to gain an understanding of what had been done to date with respect to motion capture systems, generative models, and percussion robotics that continued through the end of 2013.

The creation of a motion capture system and process to acquire the performance data needed for analysis started in the last quarter in 2013 and continued through first quarter of 2014. By the end of the first quarter of 2014, the concept of a parameter vector was proposed to describe a rudiment in statistical form. This was followed by the creation of an initial robotic prototype in the 3rd quarter that contained a single VCA encased in a commodity robotic hobbyist housing with 1DOF. The system included a custom wire-wrapped driver board with a USB interface module. A Microsoft Visual C/C++ console application was written to communicate with the prototype and develop closed loop control along with a rudimentary set of features for basic performance validation. A rudiment dataset was recorded using the established motion capture system and process with the first participant in the pilot drum study towards the end of 2014.

The “Pragmatic Drum Motion Capture System” [29] and “Drum Motion Capture Dataset” [78] NIME conference papers were written in the latter part of 2014 and submitted in January of 2015. This was followed by a sustained effort in dataset analysis

that culminated in the completion of a custom software application for parameter vector generation in the second quarter of 2015. Design of the final robotic system based on the learnings from the first prototype began in mid-2015 that included detailed mechanical and electronic design using SolidWorks/OnShape (3D modeling) and Dip Trace (schematic capture and PCB layout) respectively.

January of 2016 began with the submission of the "Snare Drum Performance Motion Analysis" [31] NIME conference paper. This was followed by software design/development and a manufacturing cycle for the mechanical and electronic subsystems that spanned the remainder of 2016. The electronics subsystems were brought up and validated in a stepwise fashion over the course of several weeks using the first prototype platform while the mechanical components were being CNC machined.

The "Voice Coil Actuators for Percussion Robotics" [33] NIME conference paper was submitted at the beginning of January 2017 and software development/testing continued throughout the remainder of the year. The development included FPGA and software development in the context of the Xilinx Vivado Toolchain, PetaLinux, and bare-metal firmware. Numerous performance tests were conducted along with the development of the OSC prototype in support of the SDRdrum that was developed by Dr. Andrew Schloss.

A number of exhibitions and concerts occurred over the course of 2018 and an "Acquisition and modeling of performance-specific percussion motion" paper was submitted to the Journal of New Music Research in September of 2018. A "Mechatronic Drummer; a New Approach to Percussion" paper for submission to IEEE Transactions in

Robotics has been in development with a tentative submission date in early 2019. Lastly, the creation of this dissertation effectively spanned the fourth quarter of 2018.

Appendix B

Patent Pending

Content begins on following page.

US 20180326588A1

(19) **United States**

(12) **Patent Application Publication** (10) **Pub. No.: US 2018/0326588 A1**

Van Rooyen et al. (43) **Pub. Date: Nov. 15, 2018**

(54) **ROBOTIC DRUMMER**

(71) Applicant: **UVic Industry Partnerships Inc.**, Victoria (CA)

(72) Inventors: **Robert Van Rooyen**, Victoria (CA); **George Tzanetakis**, Victoria (CA)

(73) Assignee: **UVic Industry Partnerships Inc.**, Victoria (CA)

(21) Appl. No.: **15/980,640**

(22) Filed: **May 15, 2018**

Related U.S. Application Data

(60) Provisional application No. 62/506,539, filed on May 15, 2017.

Publication Classification

(51) **Int. Cl.**

B25J 11/00	(2006.01)
G10H 1/32	(2006.01)
G10H 1/00	(2006.01)
B25J 9/16	(2006.01)

(52) **U.S. Cl.**

CPC ***B25J 11/004*** (2013.01); ***G10H 1/32*** (2013.01); ***Y10S 901/09*** (2013.01); ***B25J 9/1653*** (2013.01); ***B25J 9/1674*** (2013.01); ***G10H 1/0008*** (2013.01)

(57) **ABSTRACT**

Robotic drummers include voice coil actuators that are coupled to linear-to-rotary motion convertors to produce drumstick rotations so as to strike a drum head. Such rotations can be triggered via a microprocessor using stored performance data, by a user with a mouse, trackpad, joystick, or other user input device. Performances are enhanced by driving the VCA with drive signals have random variations associated with strike timing, amplitude, location, and speed. Multiple strikes are provided by reducing, eliminating, or reversing drumstick rotation with a corresponding drive signal upon detection of drumstick contact with the drum head,

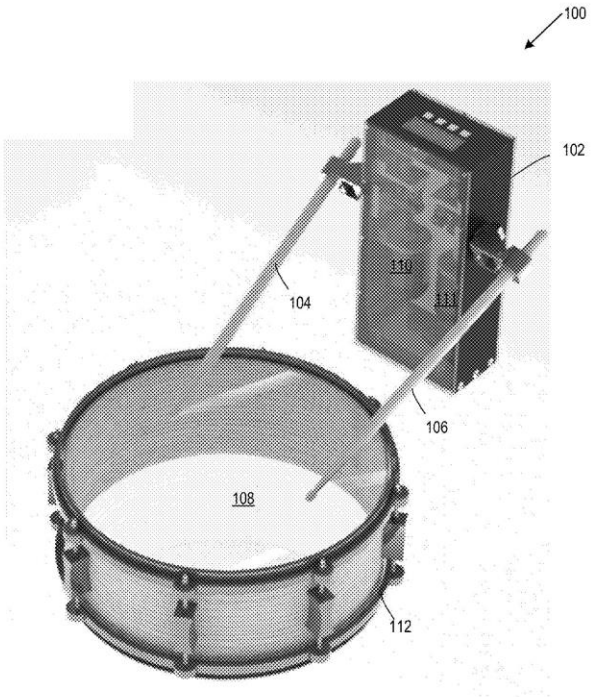


FIG. 1

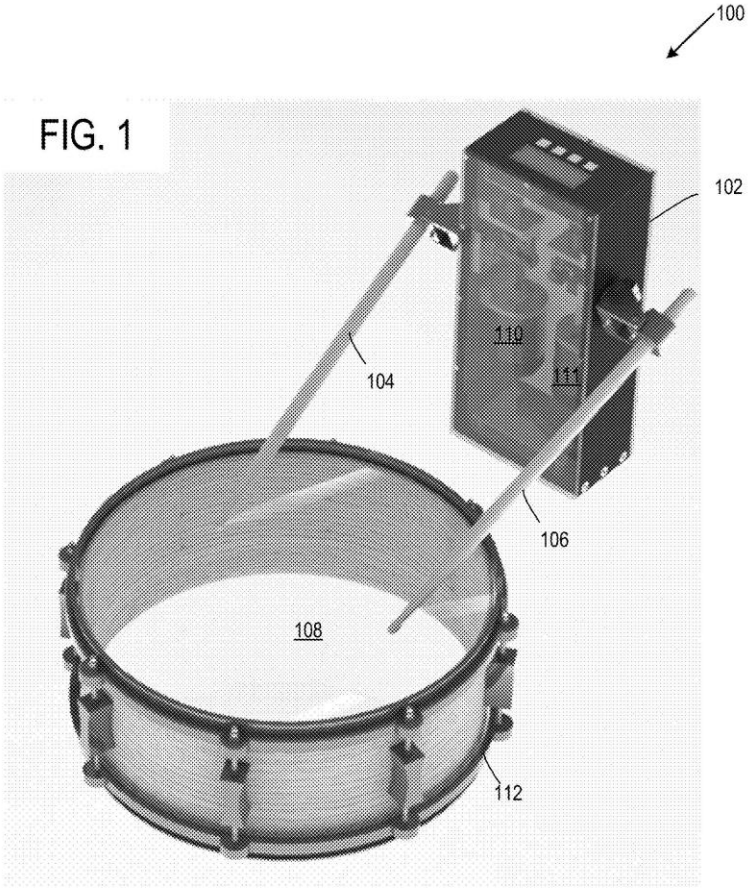
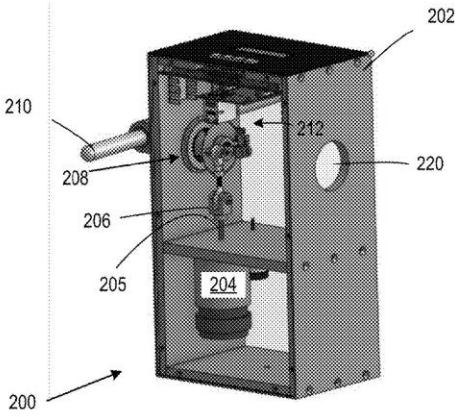


FIG. 2A



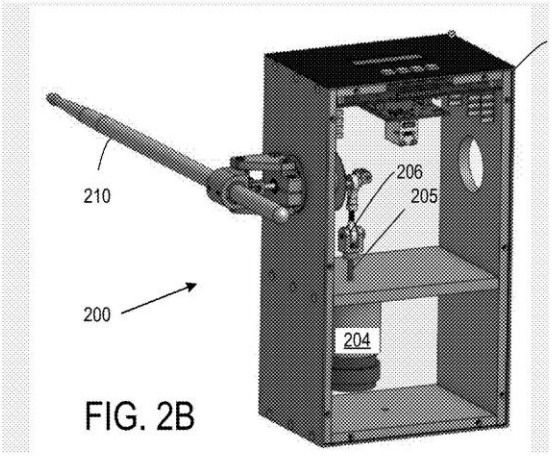


FIG. 2B

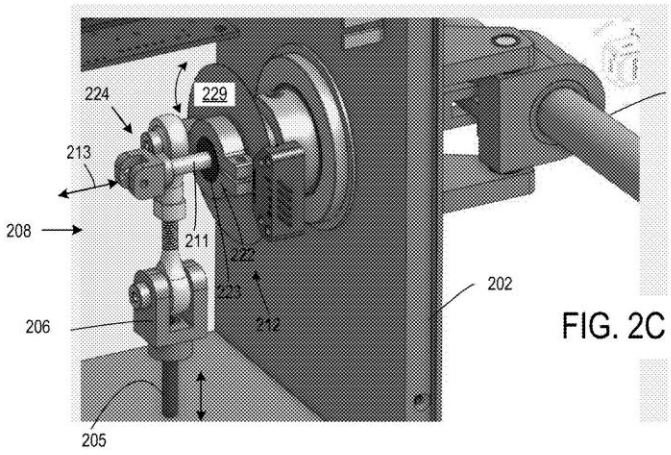

224
229
210
213
208
211
222
202
223
206
212
205

FIG. 2C

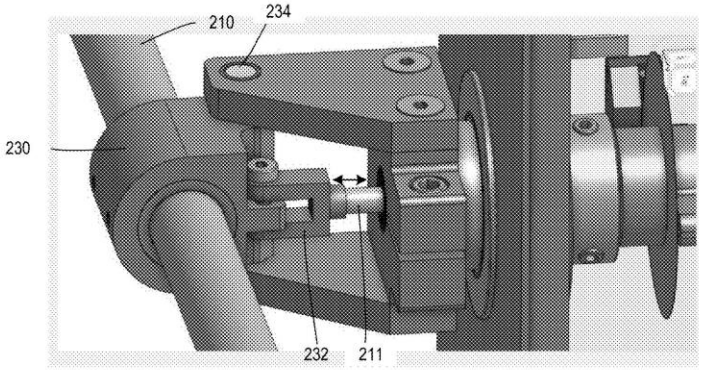
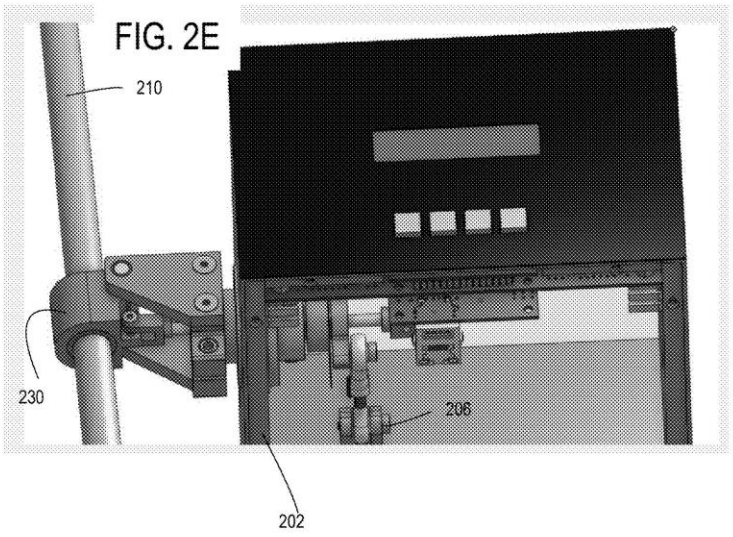

210
234
230
232
211

FIG. 2D

FIG. 2E



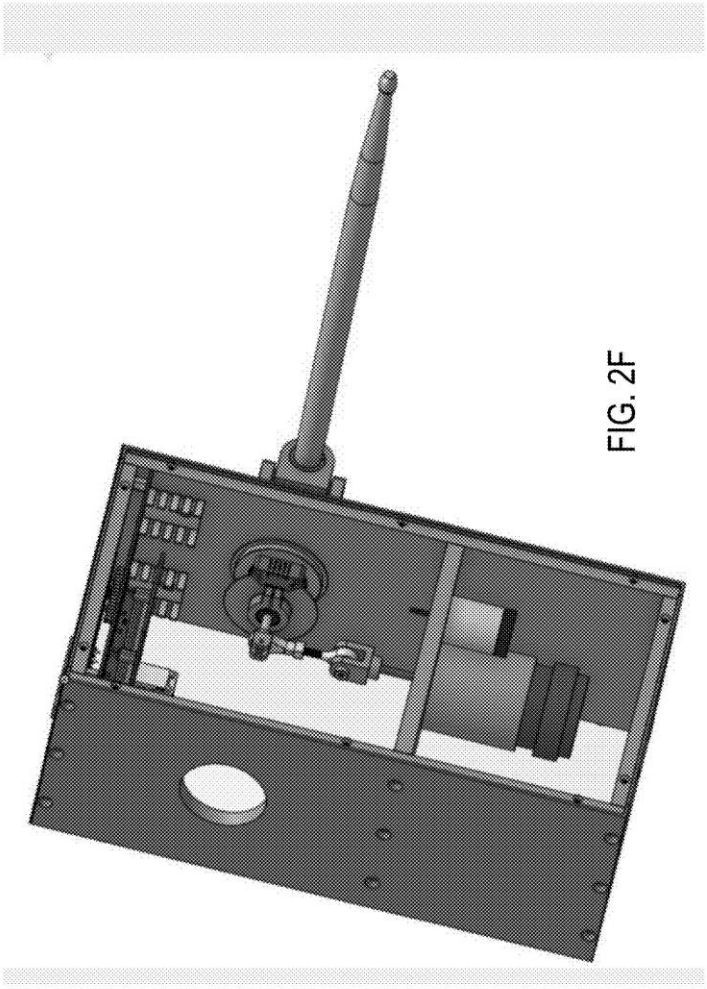


FIG. 2F

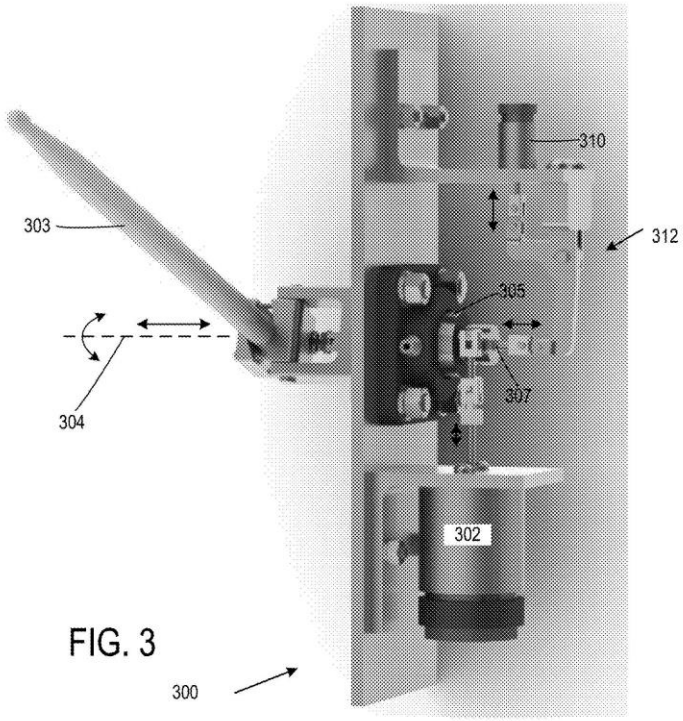


FIG. 3

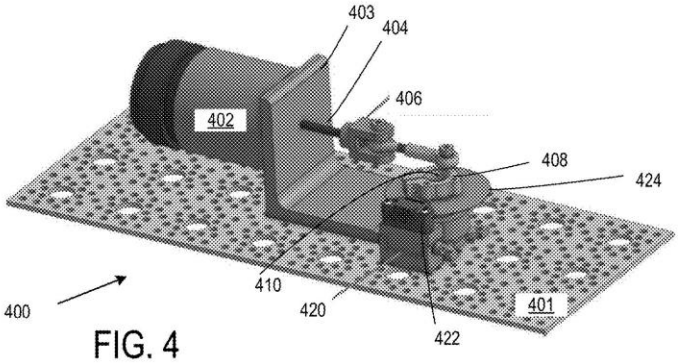


FIG. 4

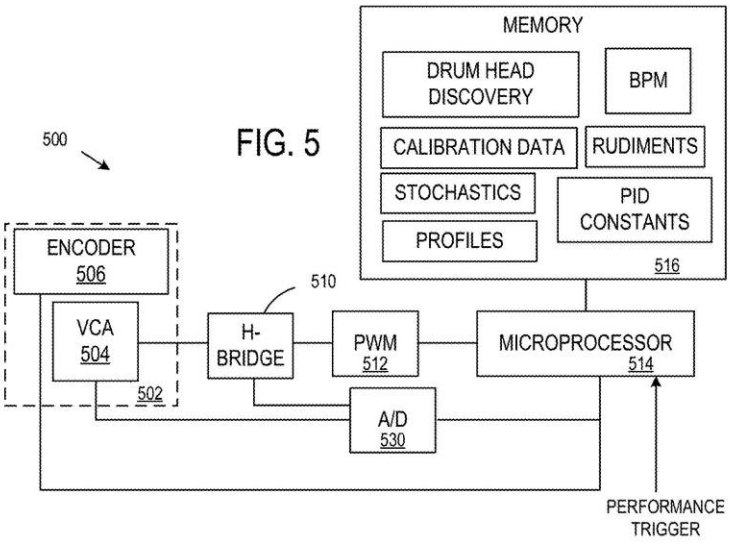

500
MEMORY
DRUM HEAD DISCOVERY
BPM
CALIBRATION DATA
RUDIMENTS
STOCHASTICS
PID CONSTANTS
PROFILES
516
ENCODER 506
VCA 504
502
510
H-BRIDGE
PWM 512
MICROPROCESSOR 514
A/D 530
PERFORMANCE TRIGGER

FIG. 5

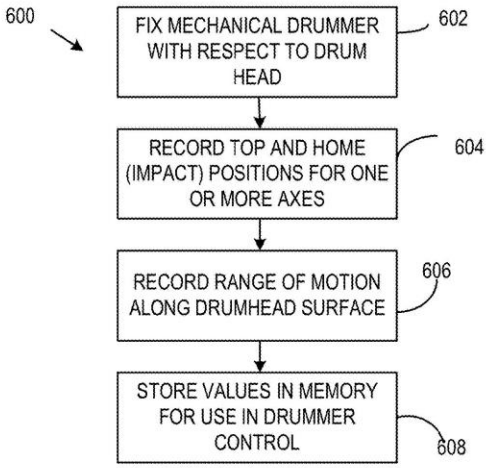

600
FIX MECHANICAL DRUMMER WITH RESPECT TO DRUM HEAD
602
RECORD TOP AND HOME (IMPACT) POSITIONS FOR ONE OR MORE AXES
604
RECORD RANGE OF MOTION ALONG DRUMHEAD SURFACE
606
STORE VALUES IN MEMORY FOR USE IN DRUMMER CONTROL
608

FIG. 6

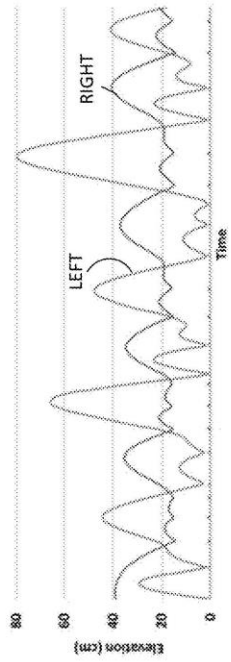


FIG. 7A

FIG. 7B

Note	Onset (μ)	Onset (σ)	Velocity (μ)	Velocity (σ)	Position (μ)	Position (σ)	Hand
1	-0.018	0.004	-1324.101	131.453	29.318	0.161	L
2	0.053	0.004	-1029.712	112.868	29.799	0.053	L
3	0.118	0.004	-515.806	66.100	42.962	0.341	R
4	0.197	0.004	-227.061	80.203	43.666	0.323	R
5	0.252	0.004	-761.860	26.731	30.598	0.510	L
6	0.321	0.004	-620.436	111.393	30.669	0.150	L
7	0.379	0.005	-424.778	136.845	44.004	0.250	R
8	0.457	0.005	-226.775	77.906	44.024	0.099	R

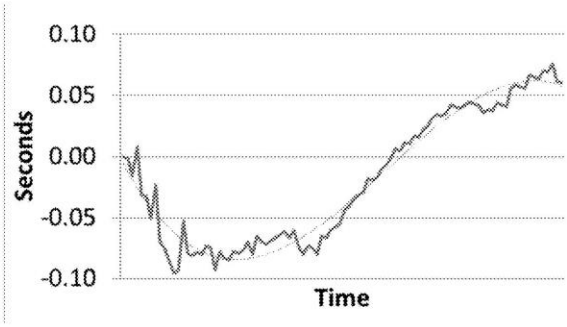


FIG. 8

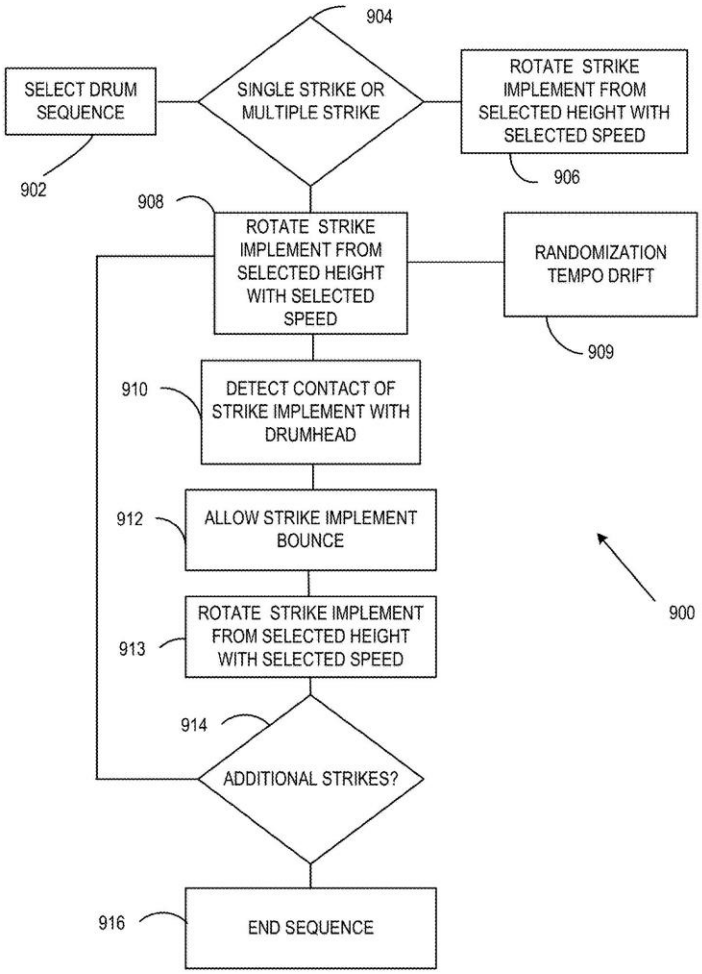

904
SELECT DRUM SEQUENCE
SINGLE STRIKE OR MULTIPLE STRIKE
ROTATE STRIKE IMPLEMENT FROM SELECTED HEIGHT WITH SELECTED SPEED
902
906
908
ROTATE STRIKE IMPLEMENT FROM SELECTED HEIGHT WITH SELECTED SPEED
RANDOMIZATION TEMPO DRIFT
909
910
DETECT CONTACT OF STRIKE IMPLEMENT WITH DRUMHEAD
912
ALLOW STRIKE IMPLEMENT BOUNCE
900
913
ROTATE STRIKE IMPLEMENT FROM SELECTED HEIGHT WITH SELECTED SPEED
914
ADDITIONAL STRIKES?
916
END SEQUENCE

FIG. 9

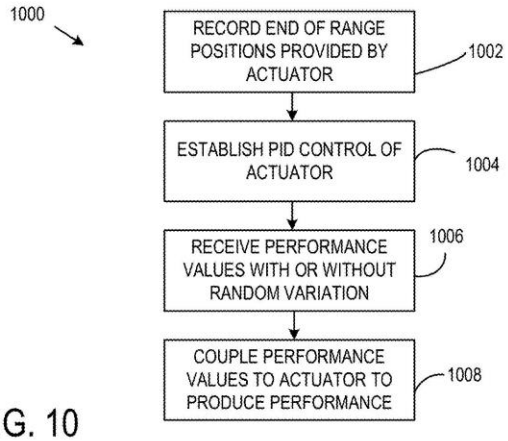

1000
RECORD END OF RANGE POSITIONS PROVIDED BY ACTUATOR
1002
ESTABLISH PID CONTROL OF ACTUATOR
1004
RECEIVE PERFORMANCE VALUES WITH OR WITHOUT RANDOM VARIATION
1006
COUPLE PERFORMANCE VALUES TO ACTUATOR TO PRODUCE PERFORMANCE
1008

FIG. 10

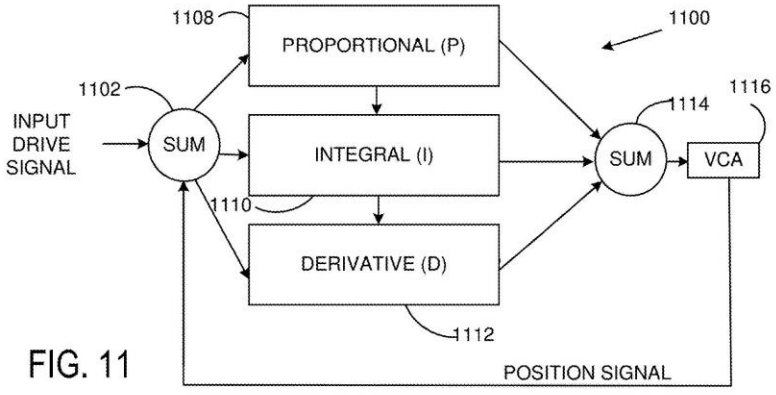

1108
PROPORTIONAL (P)
1100
1102
INPUT DRIVE SIGNAL
SUM
INTEGRAL (I)
1110
DERIVATIVE (D)
1112
1114
SUM
1116
VCA
POSITION SIGNAL

FIG. 11

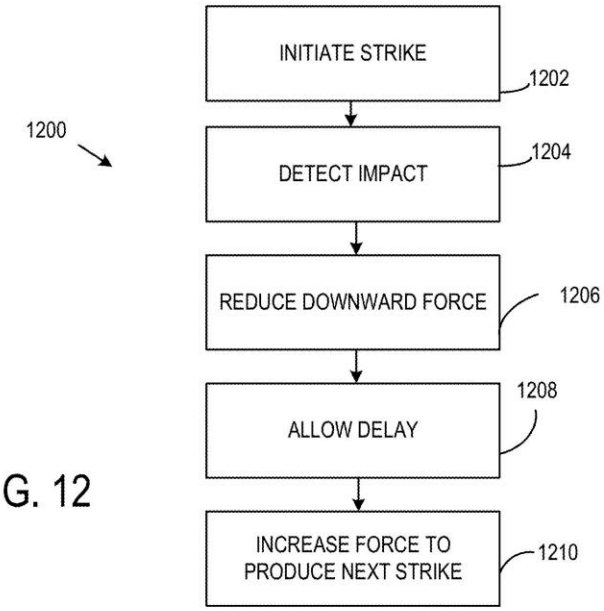

1200
INITIATE STRIKE
1202
DETECT IMPACT
1204
REDUCE DOWNWARD FORCE
1206
ALLOW DELAY
1208
INCREASE FORCE TO PRODUCE NEXT STRIKE
1210

FIG. 12

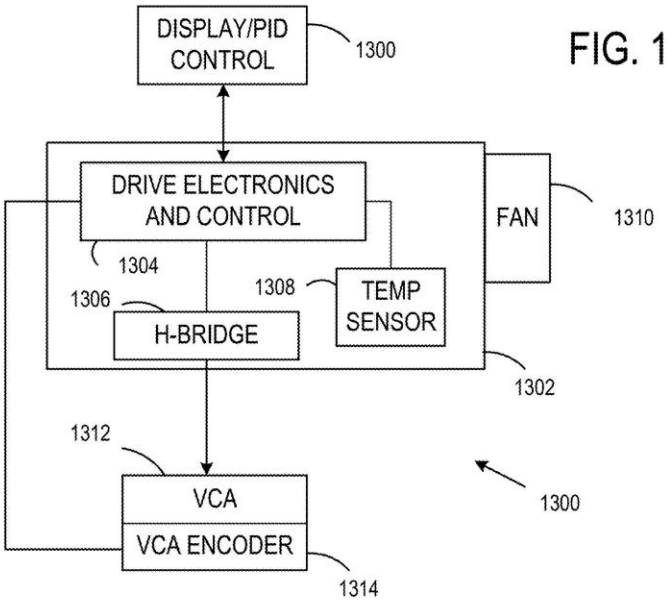

DISPLAY/PID CONTROL
1300
DRIVE ELECTRONICS AND CONTROL
1304
FAN
1310
1306
H-BRIDGE
1308
TEMP SENSOR
1302
1312
VCA
VCA ENCODER
1314
1300

FIG. 13

ROBOTIC DRUMMER

CROSS REFERENCE TO RELATED APPLICATION

[0001] This application claims the benefit of U.S. Provisional Application No. 62/506,539, filed May 15, 2017, which is hereby incorporated by reference in its entirety.

FIELD

[0002] The disclosure pertains to mechanical drumming systems.

BACKGROUND

[0003] Mechanical systems for use with percussion instruments generally do not provide sounds that are satisfactory. Such systems also tend to be complex and expensive, and are not adapted for use by persons with disabilities. Improved robotic systems and methods are needed.

SUMMARY

[0004] Robotic drumming apparatus comprise a voice coil actuator (VCA) that includes a shaft and a linear-to-rotary motion convertor coupled to the VCA shaft. A rotary encoder is coupled to the linear-to-rotary motion convertor so as to indicate a rotation produced by the linear-to-rotary motion convertor. A rotatable shaft is coupled to the linear-to-rotary motion convertor so as to rotate in response to a linear motion of the shaft of the VCA. In typical examples, a striking implement retainer is secured to the rotatable shaft. In some embodiments, the linear-to-rotary motion convertor includes a crank arm secured to a rotatable wheel, wherein the crank arm is coupled to the VCA shaft so as to rotate the rotatable wheel in response to the linear motion of the VCA shaft. In other examples, an H-bridge is coupled to the VCA so as to apply control signals to the VCA. In one example, a microprocessor and a digital-to-analog convertor are configured to generate the control signals applied to the VCA. In still further examples, a PID controller on a chip is situated to generate the control signals applied to the VCA. In additional representative examples, the PID controller reduces VCA drive current in response to detection of contact of a striking implement with a surface. In still further examples, the controller is configured to increase the VCA drive current as the striking implement retainer is rotating away from the surface so as to produce a multistrike. In still further examples, the microprocessor provides a random offset to the VCA current and is associated with a random variation in amplitude, tempo, or strike position. In another example, a second VCA is situated to move a striking implement across a surface, wherein the rotatable shaft is coupled to the linear-to-rotary motion convertor so as to rotate a striking implement tip so as to contact the surface. In a specific example, a bell crank is coupled to the second VCA so that linear motion of a shaft of the second VCA moves the tip of the striking implement across the surface. shaft.

[0005] Methods comprise applying a control current to a VCA so as to manipulate a striking implement to contact a drum head and detecting contact of the striking implement with the drum head. The control current is reduced in response to the detecting of the contact. In some examples, a subsequent contact of the striking implement is initiated after the contact is detected after a bounce time elapses. According to some examples, the contact is detected based on an image of the drum head and the striking implement tip. In other examples, the contact is detected based on an electrical signal received from the VCA or based on a comparison of a predetermined home coordinate associated with the drum head and a current coordinate of the striking implement. In still additional examples, the VCA is coupled to produce a rotation, and the contact is detected based on a comparison of a predetermined home rotation associated with rotation of the striking implement so as to contact the drum head and a current rotation of the striking implement.

[0006] Methods of robotic drumming comprise applying a drive signal to a voice coil actuator and converting a linear motion of a shaft of the VCA to a rotary motion. A drumstick is coupled so as to be rotated in response to the rotary motion, wherein the drumstick is rotated so as to strike a surface corresponding to a drum head. In other examples, the drive signal is associated with a series of contacts with the drum head surface, and is based on drumstick displacements and speeds stored in a tangible computer readable medium. According to other examples, the series of contacts includes average values and deviations for one or more of drumstick tip speed, drumstick displacement above the drum head, and drumstick movement initiation time. In further examples, upon detecting contact of the striking implement with the drum head, the drive signal is reduced to produce a bounce. In some examples, the drumstick is set to strike the surface based on an acceleration associated with the VCA.

[0007] In other examples, methods comprise applying an EMF to a VCA and detecting contact with an object based on detection of a back EMF produced in response to the EMF applied to the VCA. In some embodiments, the EMF is applied to the VCA as a pulse width modulated (PWM) EMF, at least one of a pulse width, a pulse amplitude, or a duty cycle of the PWM EMF is adjusted in response to detection of contact with the object. In other examples, the object is contacted at a plurality of locations in response to the EMF applied to the VCA. A back EMF associated with the plurality of locations is detected, and based on detection of the back EMF, a surface shape of at least a portion of the object is determined. In typical examples, the surface shape includes at least one of a surface extent and a surface tilt and in further examples, a timing of the EMF applied to the VCA is adjusted.

[0008] Apparatus comprise a voice coil actuator (VCA) and a PID controller coupled to activate the VCA. A user interface is configured to display PID control values and permit user adjustment of one or more of the PID control values. In representative examples, a pulse width modulator (PWM) is situated to be responsive to the PID controller so as to activate the VCA. In still other examples, an H-bridge is situated to couple the PWM to the VCA. In a representative embodiment, the H-bridge is thermally coupled to a thermally conductive enclosure and a temperature controller is situated to adjust a temperature associated with the H-bridge or the enclosure. In other representative implementations, a fan is coupled to produce circulation in the enclosure, wherein the fan is coupled to the temperature controller so that the circulation is variable based on the temperature associated with the H-bridge or the enclosure. In some cases, the H-bridge includes a plurality of transistors that are thermally coupled to the thermally conductive enclosure.

[0009] The foregoing and other objects, features, and advantages of the disclosed technology will become more apparent from the following detailed description, which proceeds with reference to the accompanying figures.

BRIEF DESCRIPTION OF THE DRAWINGS

[0010] FIG. **1** illustrates a representative robotic drummer that uses two drumsticks.

[0011] FIGS. **2**A-**2**F illustrates a representative robotic drummer with a single drumstick installed.

[0012] FIG. **3** illustrates a representative robotic drummer that includes voice coil actuator for use in striking a drum surface with a striking implement and a voice coil actuator for moving the drumstick to different locations on the drum head surface.

[0013] FIG. **4** illustrates a portion of another representative robotic drummer.

[0014] FIG. **5** illustrates a representative drive system coupled to a robotic drummer. Only a single VCA is shown in FIG. **5**, but additional VCAs can be provided for additional drumsticks or to provide different types of motion such as striking motion and motion across a drum head.

[0015] FIG. **6** illustrates a calibration method.

[0016] FIG. **7**A illustrates recorded data for left and right drumstick tips.

[0017] FIG. **7**B illustrates a portion of a vector data associated with a double roll.

[0018] FIG. **8** illustrates tempo variation.

[0019] FIG. **9** illustrates a method of applying multiple or single strikes to a drum head.

[0020] FIG. **10** illustrates a method of controlling an actuator such as a VCA.

[0021] FIG. **11** illustrates PID control.

[0022] FIG. **12** illustrates a method of producing drumstick bounce.

[0023] FIG. **13** illustrates a representative VCA control system.

DETAILED DESCRIPTION

[0024] Mechanical drummers are disclosed that incorporate electromechanical systems such as voice coil actuators (VCAs) that are suitably controlled to approximate human drumming. Human motion can be captured and modeled in a compressed statistical vector representation for rendering by a mechanical drummer when triggered by, for example, a MIDI interface or other network interface. The disclosed approaches can permit musicians and composers to conduct live and preprogrammed performances with a wide degree of control, and can enable performances by musicians with varying degrees of disabilities. Parameter driven stochastic model[s] can be provided to render unique human inspired performances.

[0025] VCA's can provide linear and continuous performance characteristics that meet or exceed human motion given appropriate sensors, driver electronics, closed loop control, and mechanical systems, and can be adapted to human motions associated with drumming. PID control of a VCA using various position sensors can permit positioning a VCA accurately with minimal damping time when tuned. Application specific PID control couples position control with momentary direction and force curves that take full advantage of VCA acceleration. This innovative approach to motion control is key to delivering a performance that is comparable to a human performance. Random variation in strikes can provide a more realistic, human feel. Stochastic models for timing, dynamics, and position can provide deliver unique performances that adhere to generalized percussion patterns or individual performance traits. Representative drumming mechanical implementations are provided that are controlled to produce sound associated with both single and multiple strikes. Motion capture analysis can be conducted to determine appropriate ranges of motion. The ability to create multiple strikes that is consistent with a human performance is enabled by application specific closed loop control, but is not limited by mechanics in terms of load and/or friction.

[0026] VCAs are linear motors that produce forces proportional to an applied current. A total range of motion is generally referred to as a "stroke," and strokes of up to 6 inches can be produced with a relatively constant forces. Rapid movement produces a back EMF proportional to speed, current, and magnetic field strength. Measurement of the back EMF can be used for motion detection or to provide motion control. In a typical VCA, a coil bobbin and shaft are freely movable, and closed loop control is generally needed. In the examples below, position sensors such as optical disk linear quadrature encoders or other optical encoders or Hall Effect proximity sensors are used.

[0027] Quadrature optical encoders provide sets of signals that can be decoded to determine speed and direction, and if needed, an index signal can be provided to indicate a complete rotation. A PID controller can be used to control VCA motion, and a VCA can be driven using an H bridge circuit so as to control speed and direction. PID control uses estimated or measured values associated with present error values, an integral of previous error values, and a current rate of change of error.

[0028] Examples are generally described with reference to vertical (i.e., striking motions) and not to lateral motions, such as drumstick motion across a drum head. This is for convenient explanation, and in other applications, such motions can be provided as well or instead. In some applications, drums are struck using both hands, but in some examples, only a single drumstick is used, In general, performances associated with a left hand, a right, or both hands can be produced.

[0029] As used in this application and in the claims, the singular forms "a," "an," and "the" include the plural forms unless the context clearly dictates otherwise. Additionally, the term "includes" means "comprises." Further, the term "coupled" does not exclude the presence of intermediate elements between the coupled items.

[0030] The systems, apparatus, and methods described herein should not be construed as limiting in any way. Instead, the present disclosure is directed toward all novel and non-obvious features and aspects of the various disclosed embodiments, alone and in various combinations and sub-combinations with one another. The disclosed systems, methods, and apparatus are not limited to any specific aspect or feature or combinations thereof, nor do the disclosed systems, methods, and apparatus require that any one or more specific advantages be present or problems be solved. Any theories of operation are to facilitate explanation, but the disclosed systems, methods, and apparatus are not limited to such theories of operation.

[0031] Although the operations of some of the disclosed methods are described in a particular, sequential order for

convenient presentation, it should be understood that this manner of description encompasses rearrangement, unless a particular ordering is required by specific language set forth below. For example, operations described sequentially may in some cases be rearranged or performed concurrently. Moreover, for the sake of simplicity, the attached figures may not show the various ways in which the disclosed systems, methods, and apparatus can be used in conjunction with other systems, methods, and apparatus. Additionally, the description sometimes uses terms like "produce" and "provide" to describe the disclosed methods. These terms are high-level abstractions of the actual operations that are performed. The actual operations that correspond to these terms will vary depending on the particular implementation and are readily discernible by one of ordinary skill in the art.

[0032] In some examples, values, procedures, or apparatus' are referred to as "lowest", "best", "minimum," or the like. It will be appreciated that such descriptions are intended to indicate that a selection among many used functional alternatives can be made, and such selections need not be better, smaller, or otherwise preferable to other selections.

[0033] Examples are described with reference to directions indicated as "above," "below," "upper," "lower," and the like. These terms are used for convenient description, but do not imply any particular spatial orientation.

[0034] Time varying electric signals and voltages used to activate VCAs or other actuators are referred to generally as signals. In some cases, VCAs are driven with current signals, but suitable voltages can also be used

[0035] Referring to FIG. **1**, a representative apparatus **100** includes a frame **102** that retains a first striking implement **104** and a second striking implement **106** so as to be rotatable to strike a drum head **108**. Striking implements are typically drumsticks or brushes, but other striking implements can be used. VCAs **110**, **111** are mechanically coupled to the first striking implement **104** and the second striking implement **106**, respectively, so that actuation of the VCAs **110**, **111** can be used to strike the drum head **108**. The frame **102** is generally fixed with respect to a drum **112** using an external stand, or can be fixed to the drum **112**.

[0036] A representative drumming apparatus **200** is illustrated in detail in FIGS. **2**A-**2**E. Referring to FIGS. **2**A-**2**B, a VCA **204** is secured to a frame **202** so that linear motion of a VCA shaft **205** is coupled via a clevis **206** to a linear-to-rotary motion convertor **208**. Rotary motion resulting from linear motion of the VCA shaft **205** produces a rotation of a striking implement **210** that can be used to strike a drum head. An encoder **212** is situated to measure a rotation associated with the linear-to-rotary motion convertor **208** so that striking implement rotation is measured. The frame **202** also includes an aperture **220** for use by a second VCA and associated motion convertors for a second striking implement.

[0037] Referring to FIGS. **2**C-**2**E, the linear-to-rotary motion convertor **208** includes a rotary member **222** secured to a hollow shaft **223** through which a shaft **211** extends. A linkage **224** coupled to the VCA shaft **205** by the clevis so that linear motion of the VCA shaft **205** rotates the rotary member **222**. The rotary member **222** shown corresponds to a disk or wheel but with portions not used for attachment of the linkage **224** removed. An encoder disk **229** is secured for rotation measurement. In this example, the striking implement **210** is secured to the shaft **211** with a collar **230** that is coupled with a rotatable linkage **232** that provides rotation about a shaft **234** so that a tip of the striking implement **210** is movable across a striking surface in response to linear motion of the shaft **211** as shown at **213**. Such linear motion can be provided with an additional VCA as discussed below, but is not shown in FIGS. **2**C-**2**E.

[0038] Referring to FIG. **3**, a representative two axis drum assembly **300** includes a first VCA **302** coupled to rotate striking implement **303** about an axis **304** based on rotation of a shaft **305**. A second VCA **310** is coupled to a bell crank **312** so that a shaft **307** is translatable along the axis **304** in response to activation of the VCA **310**. Encoders can be provided to determine rotation and translation of the shaft **305**, but are not shown for clarity in FIG. **3**.

[0039] A representative mechanical arrangement **400** is illustrated in FIG. **4**, and includes a VCA **402** having a shaft **404** that is coupled via a flexible coupling such as a clevis **406** to a crank **408** to convert linear motion of the shaft **404** to rotary motion of a shaft that is inserted into an aperture **410** so as to be coupled to the crank **408**. The VCA **402** is secured to a base **401** with a bracket **403**. An encoder assembly **420** includes an encoder readout **422** and an encoder wheel **424** that are coupled so as to indicate rotation of the crank **408**.

[0040] With reference to FIG. **5**, a representative drum system **500** includes a mechanical drum assembly **502** having a VCA **504** and an encoder **506**. The VCA **504** is driven with an H-bridge **510** that is coupled to a pulse width modulator (PWM) **512** that receives drive signals produced by a microprocessor **514** based on computer executable instructions stored in a memory **516**. The PWM **512** can be a separate component or provided by the microprocessor **514**. The memory **516** typically stores calibration data, procedures and data for random or stochastic variation of drum strikes, and profiles associated with drumming patterns obtained from one or more human drummers. In some examples, the memory stores MIDI or other instructions for percussion rudiments such as roll, diddle, flam, and drag rudiments. In other alternatives, storage is remote, and procedures and data are provided via a network connection. Control of the VCA **504** is provided using PID control based on PID constants stored in the memory **516**. A calibration procedure may be needed to obtain suitable PID constants. Some or all functions provided by the microprocessor **514** can be provided with dedicated analog or digital circuitry, and in many cases, such functions can be provided with a single integrated circuit, such as a custom ASIC. The H-bridge **510** is provided as it permits bi-directional control of VCA shaft extension so that the shaft can be driven in two directions (or allowed to move without a drive current applied). An analog-to-digital convertor (ADC) **530** is coupled to the VCA **504** to receive and digitize back EMFs to facilitate VCA control. Alternatively, an optical system can be used that determines striking implement position with respect to a drum head or other surface to be struck. The A/D converter **530** can also be connected to the H-Bridge **510** in order to monitor current.

[0041] The memory **516** can also include processor-executable instructions for determining position and orientation of a drum head, i.e., for drum head discovery with our without user intervention. One or more values of beats per minute (BPMs) can be received from a user and/or stored in the memory **516** for establishing timing for rudiments based on the stored rudiment data.

[0042] While mechanical drumming can be controlled via a MIDI or other network interface as shown in FIG. **5**, "live" control can be provide using a wide variety of inputs and input devices such as motion tracking systems (using, for example, magnetic, radio, video, audio signals), buttons, sliders, air pressure, blink detection, audio, input generated in response to other musical instruments or cues.

[0043] Referring to FIG. **6**, a method **600** includes fixing a mechanical drum assembly with respect to a drum head at **602**. At **604**, a top position (i.e., greatest distance from a drum head) and a home position (i.e., an impact position) are determined for each striking implement based on optical or other measurements with respect to suitable axes. For example, positions can be determined with an optical system that includes a camera. At **606**, associated ranges of motion across the drum head surface are measured. At **608**, the measured values are stored in a memory or otherwise stored for use in controlling striking implements. In some cases, ranges of striking implement accelerations and speeds are measured along with striking implement movement resolution. Acceleration can be measured or establish to provide an approximate strength of strike that can be scaled to PWM duty cycle for VCA drive. In other examples, ranges and resolution can be obtained based on characteristics of an actuator and/or striking implement dimensions without additional measurements. Establishment of home and top position permits improved control of timing, dynamics, and timbre. Typically striking implement tip movement resolutions can be 0.10 inch, 0.05 inch, 0.025 inch, or other larger or smaller resolutions. Small enough resolution also permits relatively smooth motion of the striking implement.

[0044] While basic striking implement motion can be provided using the data obtained by the method of FIG. **5**, calibration typically also uses adaptive learning to establish the extent and angle of the drum head surface and drum head performance characteristics. Rebound performance measurements can be used to populate a dynamic table, and calibration can be based on measurements of A-weighted sound pressure (SPL), but other weightings such as C-weightings can be used if appropriate. Loudness as a function of striking implement distance from the drum head prior to activation is determined and stored, with a maximum loudness typically associated with the top position.

[0045] In some examples, motion profiles are obtained by recording striking implement positions from human performances. These profiles can then be used to determine VCA drive signals. Vector data can be used to represent strike events whose timing, velocity, and location are derived statistically from many repeated measures of a human performance. As an example, with a Double Stroke Open Roll rudiment shown in FIG. **7**A, there are 32 striking events per measure. A first vector entry in the table of FIG. **7**B equates to a first note in a single measure rudiment whose timing, velocity, and location and associated statistical parameters (standard deviations) are computed by evaluating the first strike of ten or more repeated measures. FIG. **7**B contains only a subset of a full vector data table. Mean values are noted as μ and standard deviations as σ. Entries for second through the thirty-second notes contain parameters for rendering the specific note. With the statistical parameters for a complete measure, a unique performance can be rendered at will on a capable robotic or synthesis system. This concept can be extended to a large set of articulations that occupy sub-measures, measures, or entire scores. Each of the articulations should be thought of as a patch that can be loaded and triggered in the same manner as a sound or sample on a synthesis system. As shown in FIG. **8**, tempo can drift during a performance, and such drift can vary among performers. To provide more natural sound, triggers for strike events can be allowed to drift as well.

[0046] Referring to FIG. **9**, a representative method **900** includes selecting a sequence at **902**, and determining if the sequence is a single or multiple strike sequence at **904**. For a single strike, the strike implement is rotated with a selected time, height, and speed, at **906**, and selected values can correspond to predetermined performances with or without inclusion of random variation. For multiple strikes, the strike implement is similarly rotated at **908**. Contact of the strike implement with a drum head is detected at **910**, and in response, actuator drive force is reduced to permit the strike implement to bounce at **912**. After the bounce and at a suitable time, the strike implement is rotated for the next bounce at **913**. At **914**, it is determined if additional strikes are requested. If so, the method returns to **908**, otherwise the sequence ends at **916**. Random variations can be supplied at **909**.

[0047] With reference to FIG. **10**, a control method for musical instrument control **1000** for a drum or other instrument includes recording actuator range limits at **1002**, and establishing PID actuator control at **1004**, typically by determining coefficients associated with proportional, integral, and derivative signals. At **1006**, performance values are received with or without introduction of random variation to tempo, pitch, amplitude, drum head striking position or other performance related variables. At **1008**, the performance values are applied to the actuator to produce the requested sounds.

[0048] A PID control system **1100** is illustrated in FIG. **11**, and can be implemented in hardware, software, or a combination thereof. A summing junction **1102** is coupled to receive a VCA drive input and couple the VCA drive input to proportional, integral, and derivative control components **1108**, **1110**, **1112** that are in turn coupled to a summing junction **1114**. The sum is then coupled to drive a VCA **1116** or other actuator, and a position signal associated with an actuator position is returned to the summing junction **1102**.

[0049] A method **1200** for providing bounce is shown in FIG. **12**. At **1202**, a strike is initiated, and at **1204** contact with a drum head is detected. A camera or other optical system can be used, an actuator voltage or current change associated with contact can be detected, or a predetermined angle of rotation associated with contact can be detected. At **1206**, downward force on the striking implement is reduced, eliminated or reversed, and at **1208**, a delay period is allowed to elapse to provide time from the striking implement to move away from the drum head. At **1210**, force is increased in order to produce a next strike. As noted above, two VCAs (X and Y axis) can be provided for each side (left and right) in a mechanical drummer. Each of these VCAs can be calibrated and controlled as discussed above. Additional VCAs or other actuators can be used to move striking implements across a striking surface.

[0050] While the haptic feedback used by human drummers is not available to time variable pressure on a striking implement to produce multiple strikes, such variable pressures can be applied as shown above in FIG. **12**. Typically, VCA force is adjusted by adjustment of a duty cycle associated with pulse width modulation (PWM) after detect-

ing a collision with the drum head. VCA force can also be adjusted by adjusting a VCA drive signal magnitude. Collision can be detected by mapping the plane during calibration and monitoring the encoder position at which the striking implement reaches the drum head surface. Alternatively, the current through the VCA can be sampled at a sufficient rate to detect a spike in current as the VCA encounters physical resistance at impact. Since the VCA is composed of a permanent magnet and a coil, a voltage will be induced when the shaft is moved, or in the case of a collision, restricted.

[0051] As noted above, a robotic drummer can be triggered via a computer interface such as a MIDI interface, or a pointing device can be used. Use of various input devices permits a robotic drummer to be adaptable to the needs of persons with disabilities.

[0052] Referring to FIG. **13**, a VCA system **1300** includes drive electronic **1304**, an H-bridge **1306**, and a temperature sensor **1308** coupled to or contained by a conductive enclosure **1302**. A fan **1310** is situated to direct a cooling flow to the enclosure **1302**. The H-bridge **1306** is coupled to a VCA **1312** and VCA position can be determined based on an encoder **1314** that is coupled to drive electronics **1304**. Typically, the VCA **1312** is controlled using PWM and PID, and an external control **1320** can be provided that includes a display or other output device and one or more computer input devices such as a keyboard, touchscreen, or pointing device so that PID and/or PID values can be viewed and adjusted.

[0053] It will be appreciated that the disclosed methods and apparatus can be used in applications other than drumming. For example, these approaches can be used generally to control VCAs and to assess VCA status (such as VCA velocity, acceleration, and/or position). In some applications, positions of surfaces other than drum heads can be assessed, and positions with respect to the determined surface locations can be established. Variations in surface position can be accommodated using surface discovery such as described above, and an angular tolerance of a surface can be established. Timing of contact with a surface such as a drum head or other surface can be established based on velocity with respect to the surface, and repetitive motion with respect to a surface can be characterized and/or controlled. Timing can be compensated for use with tilted or displaced surfaces, and surface positions and tilts can be discovered and compensated automatically.

[0054] In view of the many possible embodiments to which the principles of the disclosed technology may be applied, it should be recognized that the illustrated embodiments are only preferred examples and should not be taken as limiting the disclosure. We claim as our invention all that comes within the scope and spirit of the appended claims.

We claim:

1. An apparatus, comprising:

a voice coil actuator (VCA) that includes a shaft;

a linear-to-rotary motion convertor coupled to the shaft of the voice coil actuator;

a rotary encoder coupled to the linear-to-rotary motion convertor so as to indicate a rotation produced by the linear-to-rotary motion convertor; and

a rotatable shaft coupled to the linear-to-rotary motion convertor so as to rotate in response to a linear motion of the shaft of the VCA.

2. The apparatus of claim **1**, further comprising a striking implement retainer secured to the rotatable shaft.

3. The apparatus of claim **1**, wherein the linear-to-rotary motion convertor includes a crank arm secured to a rotatable wheel, wherein the crank arm is coupled to the VCA shaft so as to rotate the rotatable wheel in response to the linear motion of the VCA shaft.

4. The apparatus of claim **3**, further comprising an H-bridge coupled to the VCA so as to apply control signals to the VCA.

5. The apparatus of claim **4**, further comprising a microprocessor and a PWM configured to generate the control signals to an H-Bridge to drive the VCA.

6. The apparatus of claim **25**, further comprising an optical quadrature encoder and PID controller on a chip situated to generate the control signals applied to the VCA.

7. The apparatus of any of claim **6**, wherein the PID controller reduces the VCA drive current in response to detection of contact of a striking implement with a surface.

8. The apparatus of claim **7**, wherein the controller is further configured to increase the VCA drive current as the striking implement retainer is rotating away from the surface so as to produce a multistrike.

9. The apparatus of claim **7**, wherein the PID controller terminates the VCA drive current in response to the detection of the contact.

10. The apparatus of claim **7**, wherein the microprocessor provides a random offset to the VCA current.

11. The apparatus of claim **1**, further comprising a second VCA situated to move a striking implement across a surface, wherein the rotatable shaft is coupled to the linear-to-rotary motion convertor so as to rotate a striking implement tip so as to contact the surface.

12. The apparatus of claim **11**, further comprising a bell crank coupled to the second VCA so that linear motion of a shaft of the second VCA is coupled as a corresponding linear motion to the rotatable shaft, a rotatable shaft coupled to the linear-to-rotary motion convertor so as to rotate in response to a linear motion of the shaft of the VCA.

13. A method, comprising:

applying a control current to a VCA so as to manipulate a striking implement to contact a drum head;

detecting contact of the striking implement with the drum head; and

reducing the control current in response to the detecting of the contact.

14. The method of claim **13**, further comprising initiating a subsequent contact of the striking implement after the contact is detected after a bounce time elapses.

15. The method of claim **5**, wherein the contact is detected by locating a striking implement tip at the drum head in an image.

16. The method of claim **14**, wherein the contact is detected based on an electrical signal received from the VCA.

17. The method of claim **14**, wherein the contact is detected based on a comparison of a predetermined home coordinate associated with the drum head and a current coordinate of the striking implement.

18. The method of claim **15**, wherein the VCA is coupled to produce a rotation, and the contact is detected based on a comparison of a predetermined home rotation associated with rotation of the striking implement so as to contact the drum head and a current rotation of the striking implement.

19. A method of robotic drumming, comprising:
applying a drive signal to a voice coil actuator;
converting a linear motion of a shaft of the VCA to a rotary motion; and
coupling a drumstick so as to be rotated in response to the rotary motion, wherein the drumstick is rotated so as to strike a surface correspond to a drum head.

20. The method of claim **19**, wherein the drive signal is associated with a series of contacts with the drum head surface, and is based on drumstick displacement stored in a tangible computer readable medium.

21. The method of claim **20**, wherein the series of contacts includes average values and deviations for one or more of drumstick tip speed, drumstick displacement above the drum head, and drumstick movement initiation time.

22. The method of claim **20**, further comprising detecting contact of the striking implement with the drum head, and in response, reducing the drive signal.

23. A method, comprising:
applying an EMF to a VCA; and
detecting contact with an object based on detection of a back EMF produced in response to the EMF applied to the VCA.

24. The method of claim **23**, wherein the EMF is applied to the VCA as a pulse width modulated (PWM) EMF, and further comprising adjusting at least one of a pulse width, a pulse amplitude, or a duty cycle of the PWM EMF in response to detection of contact with the object.

25. The method of claim **23**, further comprising:
contacting the object at a plurality of locations in response to the EMF applied to the VCA;
detecting at least one back EMFs associated with the plurality of locations; and
based on detection of the at least one back EMF, determining a surface shape of at least a portion of the object.

26. The method of claim **25**, wherein the surface shape includes at least one of a surface extent and a surface tilt.

27. The method of claim **23**, further comprising adjusting a timing of the EMF applied to the VCA.

28. An apparatus, comprising:
a voice coil actuator (VCA);
a PID controller coupled to activate the VCA; and
a user interface configured to display PID control values and permit user adjustment of one or more of the PID control values.

29. The apparatus of claim **28**, further comprising:
a pulse width modulator (PWM) responsive to the PID controller so as to activate the VCA.

30. The apparatus of claim **10**, further comprising an H-bridge situated to couple the PWM to the VCA.

31. The apparatus of claim **30**, further comprising a thermally conductive enclosure, wherein the H-bridge is thermally coupled to the thermally conductive enclosure.

32. The apparatus of claim **31**, further comprising a temperature controller situated to adjust a temperature associated with the H-bridge or the enclosure.

33. The apparatus of claim **20**, further comprising a fan coupled to produce circulation in the enclosure, wherein the fan is coupled to the temperature controller so that the circulation is variable based on the temperature associated with the H-bridge or the enclosure.

34. The apparatus of claim **25**, further comprising a thermally conductive enclosure, wherein the H-bridge includes a plurality of transistors that are thermally coupled to the thermally conductive enclosure.

35. The method of claim **19**, further comprising setting the drumstick to strike the surface based on an acceleration associated with the VCA.

* * * * *

Appendix C

Refereed Journal Papers

1. R. Van Rooyen, A. Schloss and G. Tzanetakis, “Acquisition and modelling of performance-specific percussion motion,” Journal of New Music Research, submitted September 29, 2018.
2. R. Van Rooyen, A. Schloss and G. Tzanetakis, “Mechatronic drummer: a new approach to percussion,” IEEE Transactions in Robotics, submission pending.

Refereed Conference Papers

1. R. Van Rooyen, A. Schloss and G. Tzanetakis, "Voice Coil Actuators for Percussion Robotics," in New Instruments for Musical Expression, Copenhagen, 2017.
2. R. Van Rooyen, A. Schloss and G. Tzanetakis, "Snare Drum Performance Motion Analysis," in New Instruments for Musical Expression, Brisbane, 2016.
3. R. Van Rooyen, A. Schloss and G. Tzanetakis, "Snare Drum Motion Capture Dataset," in New Interfaces for Musical Expression, Baton Rouge, 2015.
4. R. Van Rooyen and G. Tzanetakis, "Pragmatic Drum Motion Capture System," in New Interfaces for Musical Expression, Baton Rouge, 2015.

Appendix D

Exhibitions

1. R. Van Rooyen, Interactive Art, Science, and Technology, Lethbridge, AB, 2018.
2. R. Van Rooyen, New Interfaces for Musical Expression, Blacksburg, VA, 2018.
3. R.Van Rooyen, BC Tech Summit, Vancouver, BC, 2018.

Appendix E

Performances

1. A. Schloss, Interactive Art, Science, and Technology, Lethbridge, AB, 2018.
2. A. Schloss, International Symposium of New Music, Curitiba, Parana, Brazil, 2018.
3. A. Schloss, New Interfaces for Musical Expression Collective Response Concert, Charlottesville, VA, 2018.
4. R. Dannenberg, New Interfaces for Musical Expression Collective Response Concert, Charlottesville, VA, 2018.
5. A. Schloss, New Interfaces for Musical Expression, Blacksburg, VA, 2018.
6. R. Van Rooyen, Margret Guthman Musical Instrument Competition, Atlanta, GA, 2018

Awards, and Grants

1. University of Victoria Graduate Award, Victoria, BC, 2018.
2. Margret Guthman Musical Instrument Competition, Technical Achievement, Atlanta, GA, 2018.
3. Wighton Engineering Development Fund, Victoria, BC, 2017.
4. University of Victoria Graduate Award, Victoria, BC, 2016.
5. Wighton Engineering Development Fund, Victoria, BC, 2016.

Appendix F

Schematic

Content begins on next page.

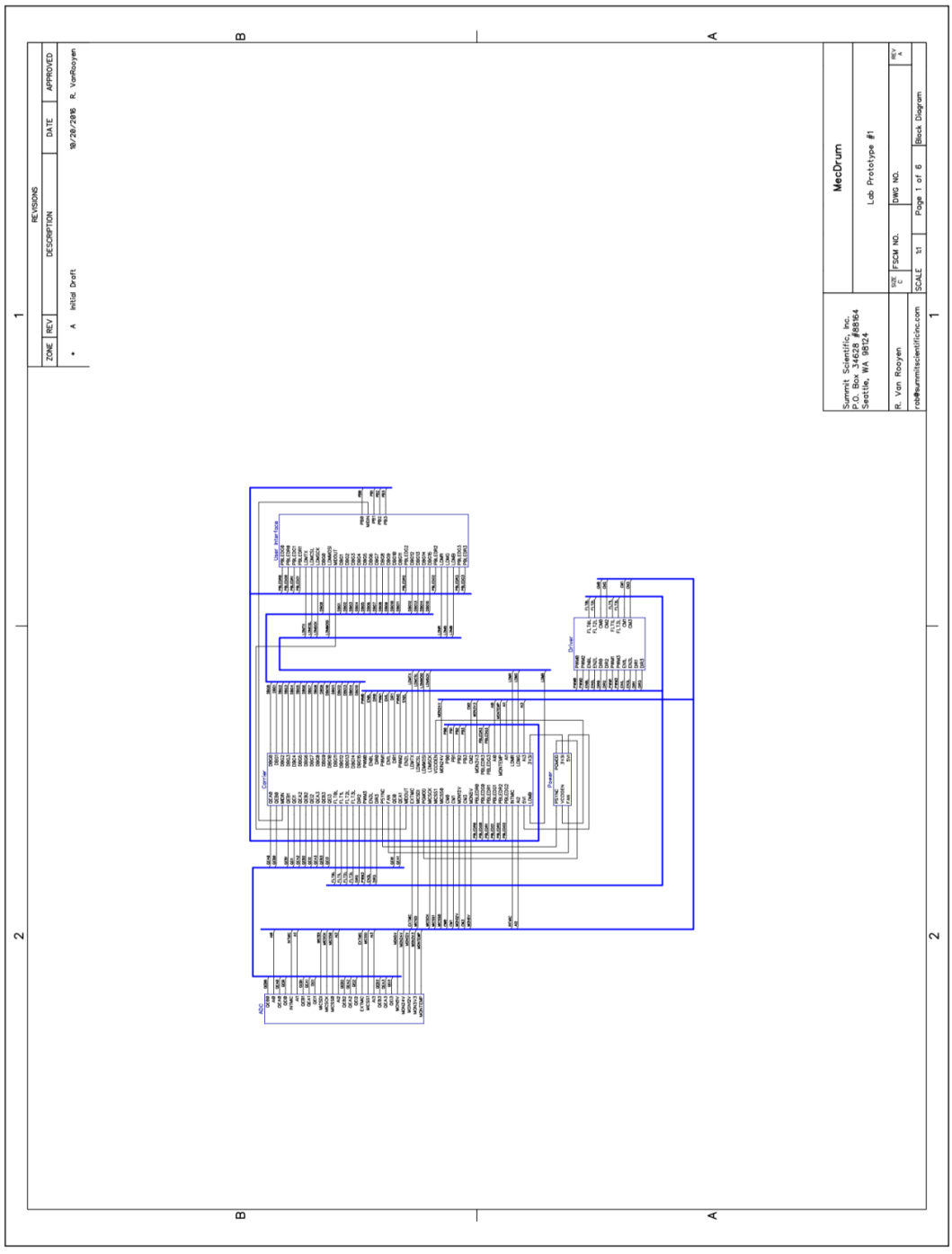


Figure 93. Carrier board schematic, page 1 of 6.

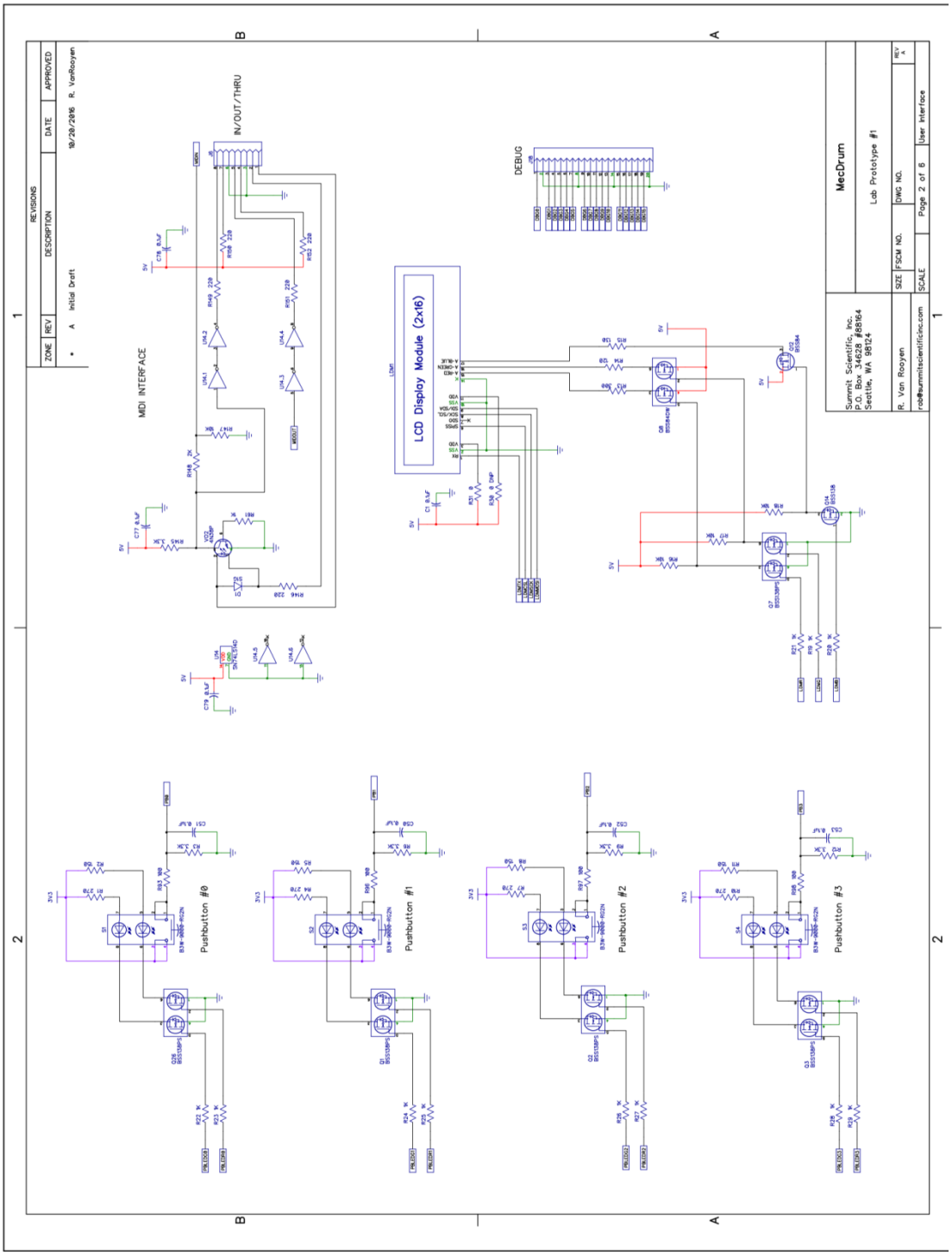


Figure 94. Carrier board schematic, page 2 of 6.

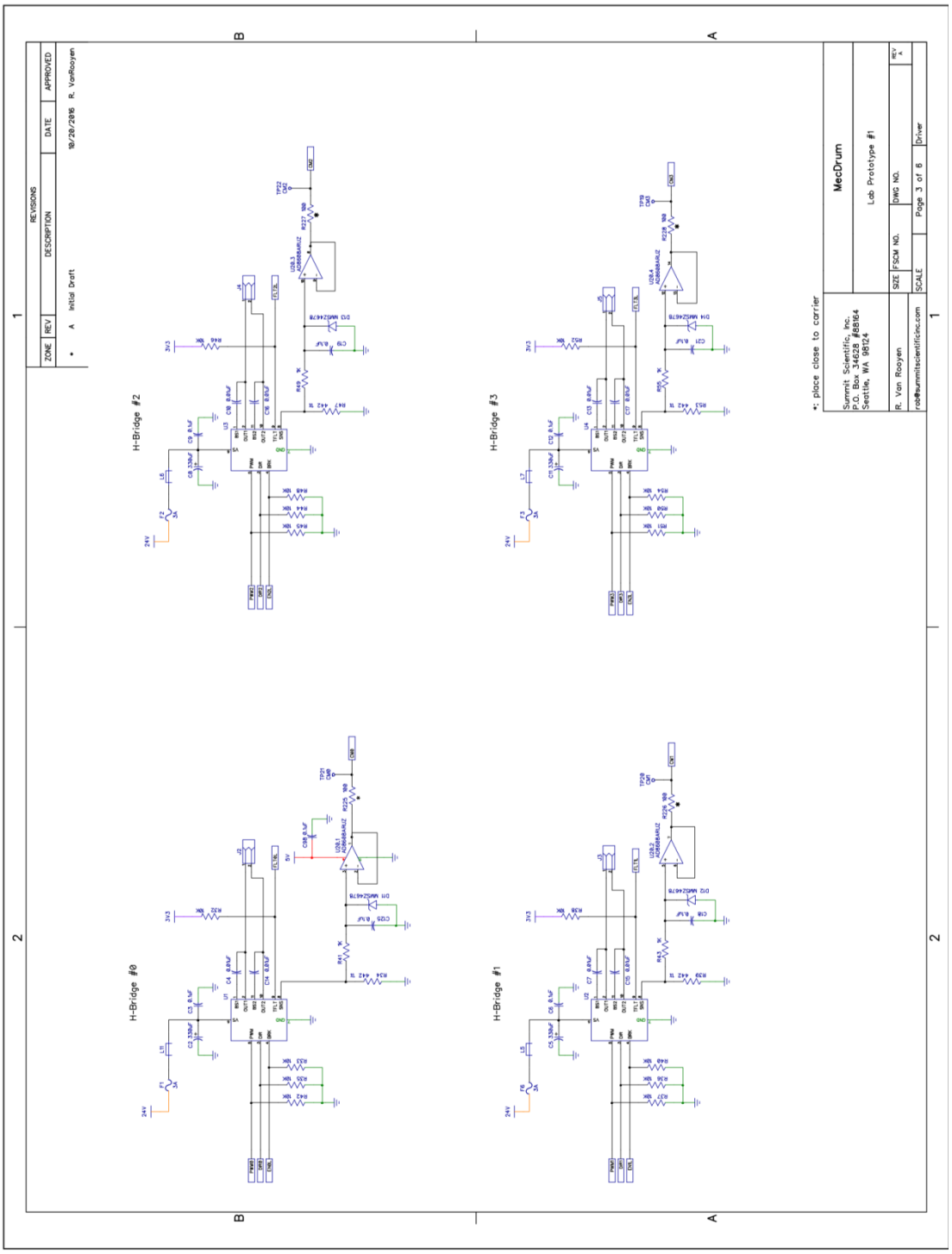


Figure 95. Carrier board schematic, page 3 of 6.

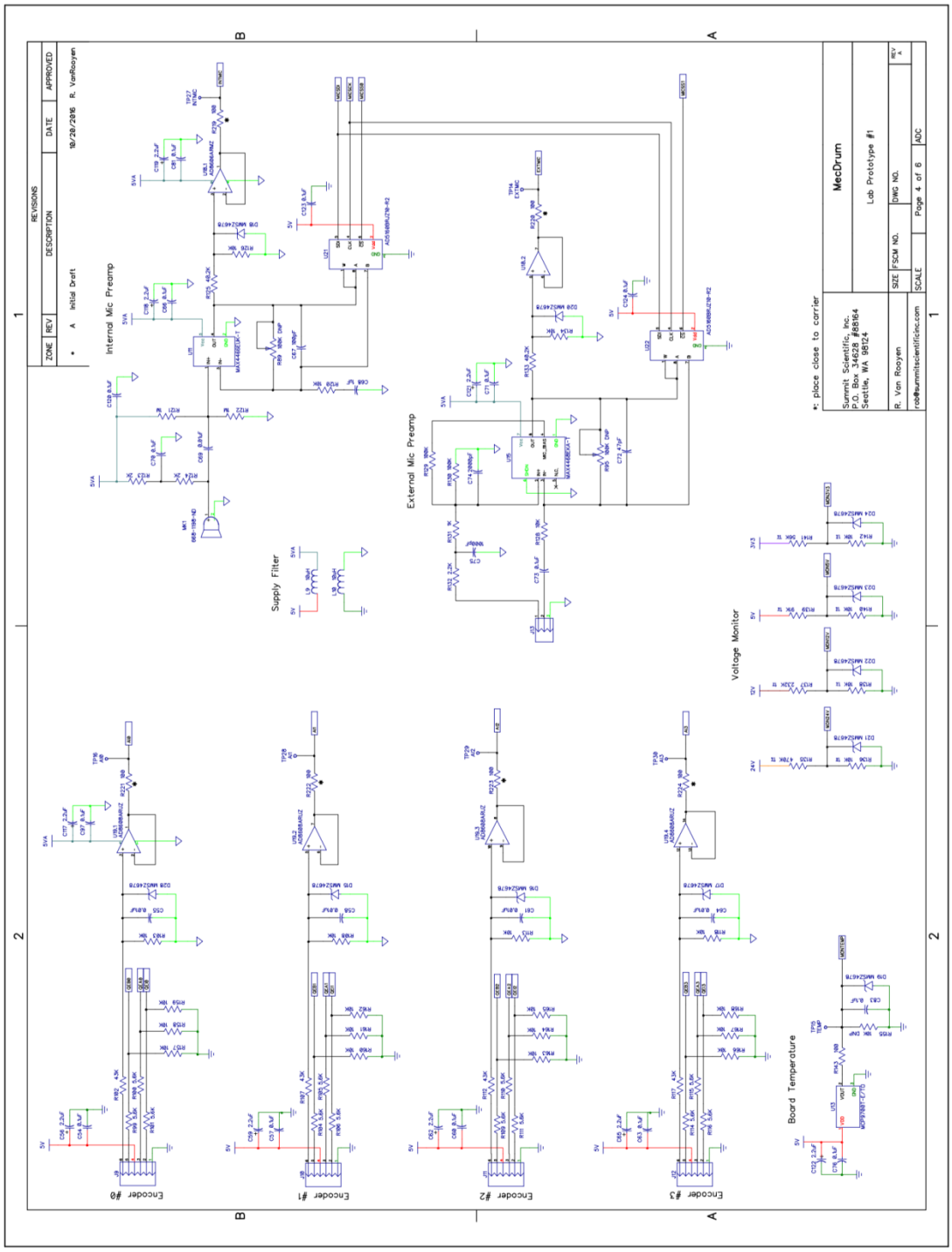


Figure 96. Carrier board schematic, page 4 of 6.

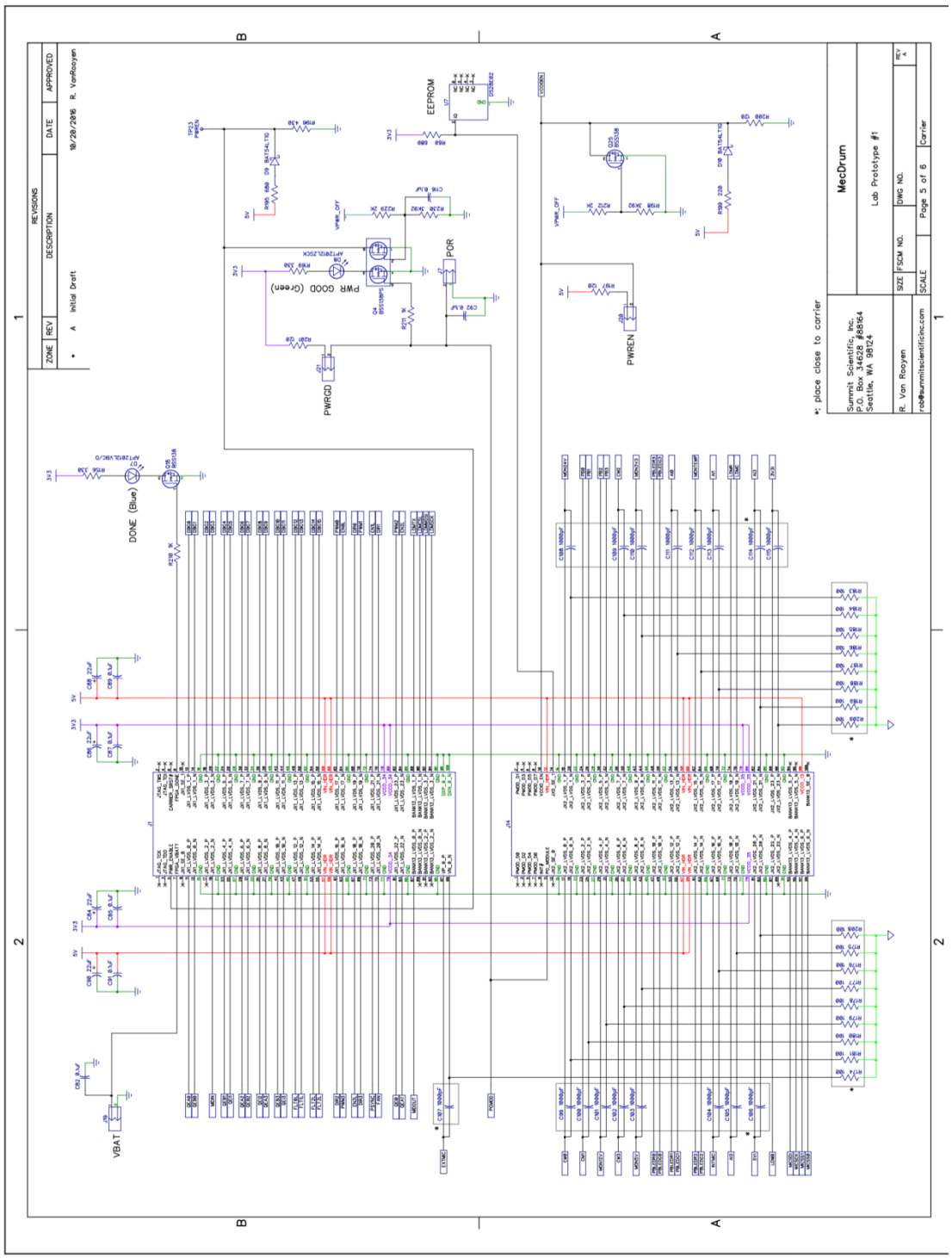


Figure 97. Carrier board schematic, page 5 of 6.

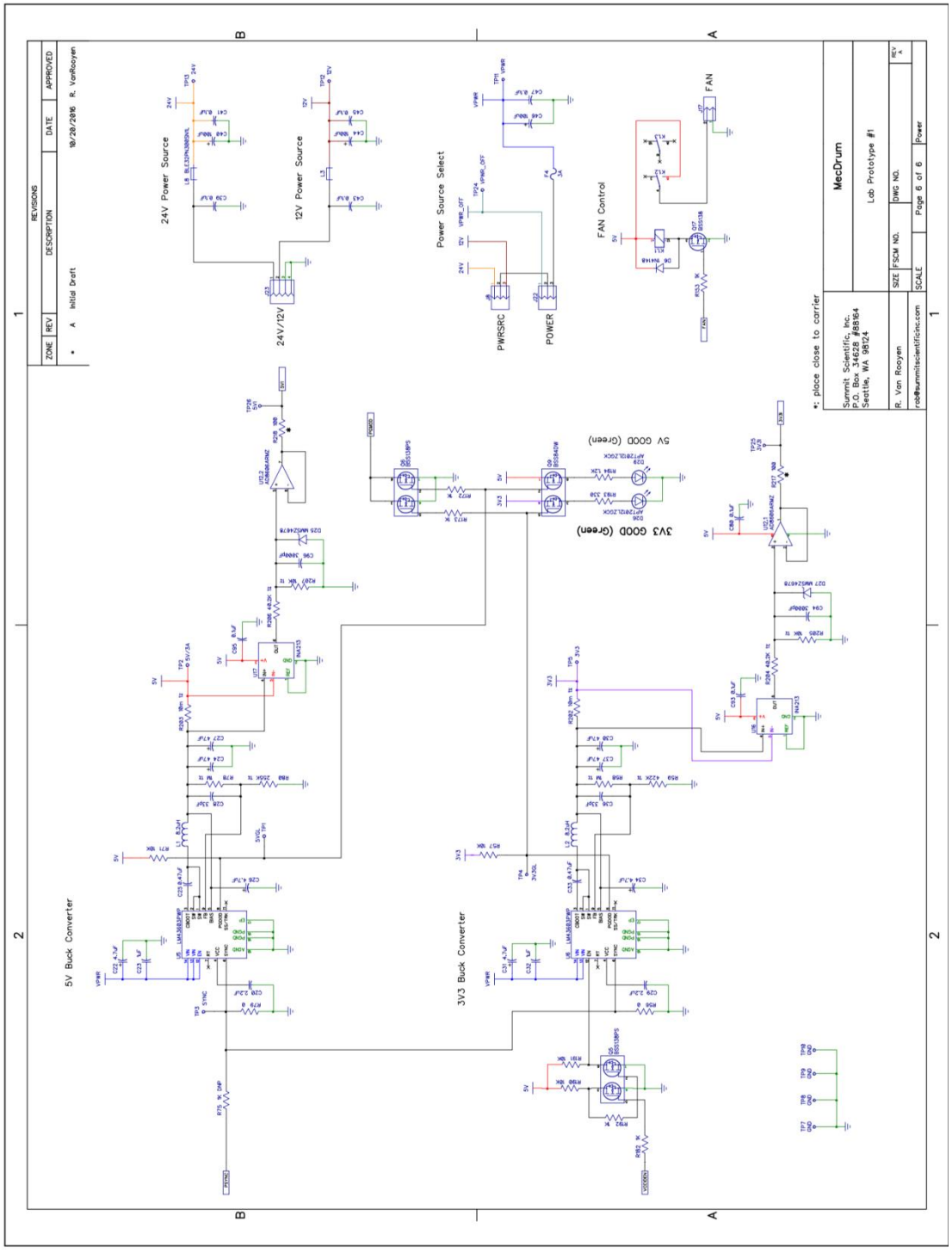


Figure 98. Carrier board schematic, page 6 of 6.

FPGA

Table 8. Field Programmable Gate Array pinout and signals information.

port	group	dir	standard	strength (mA)	slew	pull type	term	XADC	JXn	JXn pin	bank	pin	description
DBG0	DBG	out	LVCMOS33	4	FAST				JX1_LVDS_0_P	11	34	T11	debug port D[0]
DBG1	DBG	out	LVCMOS33	4	FAST				JX1_LVDS_0_N	13	34	T10	debug port D[1]
DBG2	DBG	out	LVCMOS33	4	FAST				JX1_LVDS_2_P	17	34	U13	debug port D[2]
DBG3	DBG	out	LVCMOS33	4	FAST				JX1_LVDS_2_N	19	34	V13	debug port D[3]
DBG4	DBG	out	LVCMOS33	4	FAST				JX1_LVDS_4_P	23	34	T14	debug port D[4]
DBG5	DBG	out	LVCMOS33	4	FAST				JX1_LVDS_4_N	25	34	T15	debug port D[5]
DBG6	DBG	out	LVCMOS33	4	FAST				JX1_LVDS_6_P	29	34	Y16	debug port D[6]
DBG7	DBG	out	LVCMOS33	4	FAST				JX1_LVDS_6_N	31	34	Y17	debug port D[7]
DBG8	DBG	out	LVCMOS33	4	FAST				JX1_LVDS_8_P	35	34	T16	debug port D[8]
DBG9	DBG	out	LVCMOS33	4	FAST				JX1_LVDS_8_N	37	34	U17	debug port D[9]
DBG10	DBG	out	LVCMOS33	4	FAST				JX1_LVDS_10_P	41	34	U14	debug port D[10]
DBG11	DBG	out	LVCMOS33	4	FAST				JX1_LVDS_10_N	43	34	U15	debug port D[11]
DBG12	DBG	out	LVCMOS33	4	FAST				JX1_LVDS_12_P	47	34	N18	debug port D[12]
DBG13	DBG	out	LVCMOS33	4	FAST				JX1_LVDS_12_N	49	34	P19	debug port D[13]
DBG14	DBG	out	LVCMOS33	4	FAST				JX1_LVDS_14_P	53	34	T20	debug port D[14]
DBG15	DBG	out	LVCMOS33	4	FAST				JX1_LVDS_14_N	55	34	U20	debug port D[15]
MIDIN	MIDI	in	LVCMOS33						BANK13_LVDS_5_P	94	13	V11	MIDI input port (7010: JX2_LVDS_22_P, Pin 87, Bank 34, M14)
MIDOUT	MIDI	out	LVCMOS33	4	FAST				BANK13_LVDS_5_N	96	13	V10	MIDI output port (7010: JX2_LVDS_22_N, Pin 89, Bank 34, M15)
LDMTX	LDM	out	LVCMOS33	4	FAST				BANK13_LVDS_1_P	88	13	T9	LCD module serial transmit (7010: JX1_LVDS_23_N, Pin 84, P16)
LDMCSL	LDM	out	LVCMOS33	4	FAST				BANK13_LVDS_1_N	90	13	U10	LCD module SPI chip select (7010: not used)
LDMMOSI	LDM	out	LVCMOS33	4	FAST				BANK13_LVDS_3_N	94	13	U5	LCD module SPI master out slave in (7010: not used)
LDMSCK	LDM	out	LVCMOS33	4	FAST				BANK13_LVDS_3_P	92	13	T5	LCD module SPI clock (7010: not used)
LDMR	LDM	out	LVCMOS33	4	QUIETIO				BANK13_LVDS_2_P	91	13	V8	LCD module green backlight LED (7010: JX1_LVDS_14_P, Pin 53, T20)
LDMG	LDM	out	LVCMOS33	4	QUIETIO				BANK13_LVDS_0_N	89	13	V7	LCD module red backlight LED (7010: JX1_LVDS_14_N, Pin 55, U20)
LDMB	LDM	out	LVCMOS33	4	QUIETIO				BANK13_LVDS_2_N	93	13	W8	LCD module blue backlight LED (7010: JX1_LVDS_12_N, Pin 49, P19)
PB0	PB	in	LVCMOS33						JX1_LVDS_17_P	62	34	V16	push button #0
PB1	PB	in	LVCMOS33						JX1_LVDS_13_N	50	34	P20	push button #1
PB2	PB	in	LVCMOS33						JX1_LVDS_11_P	42	34	U18	push button #2
PB3	PB	in	LVCMOS33						JX1_LVDS_7_N	32	34	Y14	push button #3
PBLEDR0	PB	out	LVCMOS33	4	QUIETIO				JX1_LVDS_17_N	64	34	W16	push button red LED #0
PBLEDG0	PB	out	LVCMOS33	4	QUIETIO				JX1_LVDS_19_P	68	34	T17	push button green LED #0
PBLEDR1	PB	out	LVCMOS33	4	QUIETIO				JX1_LVDS_15_P	54	34	V20	push button red LED #1
PBLEDG1	PB	out	LVCMOS33	4	QUIETIO				JX1_LVDS_15_N	56	34	W20	push button green LED #1
PBLEDR2	PB	out	LVCMOS33	4	QUIETIO				JX1_LVDS_11_N	44	34	U19	push button red LED #2
PBLEDG2	PB	out	LVCMOS33	4	QUIETIO				JX1_LVDS_13_P	48	34	N20	push button green LED #2
PBLEDR3	PB	out	LVCMOS33	4	QUIETIO				JX1_LVDS_9_P	36	34	V15	push button red LED #3
PBLEDG3	PB	out	LVCMOS33	4	QUIETIO				JX1_LVDS_9_N	38	34	W15	push button green LED #3
PWM0	DRV	out	LVCMOS33	4	FAST				JX2_LVDS_19_N	76	35	G15	H-bridge #0 PWM
EN0L	DRV	out	LVCMOS33	4	FAST				JX2_LVDS_11_N	50	35	K18	H-bridge #0 enable (active low)
DIR0	DRV	out	LVCMOS33	4	FAST				JX2_LVDS_19_P	74	35	H15	H-bridge #0 direction
FLT0L	DRV	in	LVCMOS33	4	FAST				JX1_LVDS_19_N	70	34	R18	H-bridge #0 fault (active low)
CM0	DRV	in	XADC					XADC_AD0_P	JX2_LVDS_0_P	17	35	C20	H-bridge #0 current monitor
PWM1	DRV	out	LVCMOS33	4	FAST				JX2_LVDS_5_N	32	35	F17	H-bridge #1 PWM
EN1L	DRV	out	LVCMOS33	4	FAST				JX2_LVDS_5_P	30	35	F16	H-bridge #1 enable (active low)
DIR1	DRV	out	LVCMOS33	4	FAST				JX2_LVDS_3_N	26	35	D20	H-bridge #1 direction
FLT1L	DRV	in	LVCMOS33	4	FAST				JX2_LVDS_11_P	48	35	K17	H-bridge #1 fault (active low)
CM1	DRV	in	XADC					XADC_AD1_P	JX2_LVDS_2_P	23	35	E17	H-bridge #1 current monitor
PWM2	DRV	out	LVCMOS33	4	FAST				JX1_LVDS_18_P	67	34	R16	H-bridge #2 PWM
EN2L	DRV	out	LVCMOS33	4	FAST				JX1_LVDS_18_N	69	34	R17	H-bridge #2 enable (active low)
DIR2	DRV	out	LVCMOS33	4	FAST				JX1_LVDS_20_P	73	34	V17	H-bridge #2 direction
FLT2L	DRV	in	LVCMOS33	4	FAST				JX1_LVDS_16_N	63	34	Y19	H-bridge #2 fault (active low)
CM2	DRV	in	XADC					XADC_AD2_P	JX2_LVDS_7_P	36	35	M19	H-bridge #2 current monitor
PWM3	DRV	out	LVCMOS33	4	FAST				JX1_LVDS_3_P	18	34	V12	H-bridge #3 PWM
EN3L	DRV	out	LVCMOS33	4	FAST				JX1_LVDS_1_N	14	34	U12	H-bridge #3 enable (active low)
DIR3	DRV	out	LVCMOS33	4	FAST				JX1_LVDS_1_P	12	34	T12	H-bridge #3 direction
FLT3L	DRV	in	LVCMOS33	4	FAST				JX1_LVDS_3_N	20	34	W13	H-bridge #3 fault (active low)
CM3	DRV	in	XADC					XADC_AD3_P	JX2_LVDS_6_P	35	35	L19	H-bridge #3 current monitor
QEA0	ADC	in	LVCMOS33						JX1_LVDS_21_N	76	34	W19	quadrature encoder A #0
QEB0	ADC	in	LVCMOS33						JX1_LVDS_23_P	82	34	P15	quadrature encoder B #0
QEI0	ADC	in	LVCMOS33						JX1_LVDS_21_P	74	34	W18	quadrature encoder index #0
AI0	ADC	in	XADC					XADC_AD4_P	JX2_LVDS_13_P	54	35	J18	analog input #0
QEA1	ADC	in	LVCMOS33						JX2_LVDS_14_N	63	35	G18	quadrature encoder A #1
QEB1	ADC	in	LVCMOS33						JX2_LVDS_3_P	24	35	D19	quadrature encoder B #1
QEI1	ADC	in	LVCMOS33						JX2_LVDS_14_P	61	35	G17	quadrature encoder index #1
AI1	ADC	in	XADC					XADC_AD5_P	JX2_LVDS_17_P	68	35	J20	analog input #1
QEA2	ADC	in	LVCMOS33						JX1_LVDS_22_P	81	34	N17	quadrature encoder A #2
QEB2	ADC	in	LVCMOS33						JX1_LVDS_20_N	75	34	V18	quadrature encoder B #2
QEI2	ADC	in	LVCMOS33						JX1_LVDS_22_N	83	34	P18	quadrature encoder index #2
AI2	ADC	in	XADC					XADC_AD6_P	JX2_LVDS_18_P	73	35	K14	analog input #2
QEA3	ADC	in	LVCMOS33						JX1_LVDS_5_N	26	34	R14	quadrature encoder A #3
QEB3	ADC	in	LVCMOS33						JX1_LVDS_7_P	30	34	W14	quadrature encoder B #3
QEI3	ADC	in	LVCMOS33						JX1_LVDS_5_P	24	34	P14	quadrature encoder index #3
AI3	ADC	in	XADC					XADC_AD7_P	JX2_LVDS_21_P	82	35	L14	analog input #3
MON24V	MON	in	XADC					XADC_AD8_P	JX2_LVDS_1_P	18	35	B19	24VDC monitor
MON12V	MON	in	XADC					XADC_AD9_P	JX2_LVDS_4_P	29	35	E18	12VDC monitor
MON5V	MON	in	XADC					XADC_AD10_P	JX2_LVDS_8_P	41	35	M17	5V rail monitor
MON3V3	MON	in	XADC					XADC_AD11_P	JX2_LVDS_9_P	42	35	K19	3.3V rail monitor
MONTEMP	MON	in	XADC					XADC_AD12_P	JX2_LVDS_15_P	62	35	F19	temperature monitor
INTMIC	AUD	in	XADC					XADC_AD13_P	JX2_LVDS_16_P	67	35	G19	internal microphone
EXTMIC	AUD	in	XADC					XADC_VP_0_P	VP_0_P	97	0	K9	external microphone
5VI	MON	in	XADC					XADC_AD14_P	JX2_LVDS_20_P	81	35	N15	5V rail current monitor
3V3I	MON	in	XADC					XADC_AD15_P	JX2_LVDS_23_P	88	35	K16	3.3V rail current monitor
PSYNC	PWR	out	LVCMOS33	4	FAST				JX1_LVDS_23_N	84	34	P16	power supply synchronization
MICSDI	MIC	out	LVCMOS33	4	FAST				JX2_LVDS_10_P	47	35	L16	microphone gain SPI data input
MICSCK	MIC	out	LVCMOS33	4	FAST				JX2_LVDS_12_P	53	35	H16	microphone gain SPI clock
MICSS1	MIC	out	LVCMOS33	4	FAST				JX2_LVDS_12_N	55	35	H17	microphone gain SPI slave select 0
MICSS0	MIC	out	LVCMOS33	4	FAST				JX2_LVDS_10_N	49	35	L17	microphone gain SPI slave select 1
FAN	FAN	out	LVCMOS33	4	QUIETIO				JX1_LVDS_16_P	61	34	Y18	FAN enable

Debug/Expansion Port

Table 9. Debug/expansion port signals.

Group	SEL	Signal	DBG	Description
NONE	0	LDMG	15	LCD module green LED
		LDMR	14	LCD module red LED
		LDMB	13	LCD module blue LED
		LogRx	1	log receive
		logTx	0	log transmit
PID0	1	QEA0	15	Quadrature encoder channel A
		QEB0	14	Quadrature encoder channel B
		QEI0	13	Quadrature encoder channel I
		FLTL0	12	PWM driver fault (active low)
		PWM0	11	PWM driver
		DIR0	10	PWM driver direction
		EN0L	9	PWM driver enable (active low)
		start0	8	PID start flag
		done0	7	PID done flag
		idle0	6	PID idle flag
		ready0	5	PID ready flag
		pdir_vld0	4	PID direction valid
		pdir0	3	PID direction
		pduty_vld0	2	PID duty valid
		QD_CLRN0	1	Quadrature decoder clear (active low)
		PID_RSTN0	0	PID reset (active low)
PID1	2	QEA1	15	Quadrature encoder channel A
		QEB1	14	Quadrature encoder channel B
		QEI1	13	Quadrature encoder channel I
		FLTL1	12	PWM driver fault (active low)
		PWM1	11	PWM driver
		DIR1	10	PWM driver direction
		EN1L	9	PWM driver enable (active low)
		start1	8	PID start flag
		done1	7	PID done flag
		idle1	6	PID idle flag
		ready1	5	PID ready flag
		pdir_vld1	4	PID direction valid
		pdir1	3	PID direction
		pduty_vld1	2	PID duty valid
		QD_CLRN1	1	Quadrature decoder clear (active low)
		PID_RSTN1	0	PID reset (active low)
PID2	3	QEA2	15	Quadrature encoder channel A
		QEB2	14	Quadrature encoder channel B
		QEI2	13	Quadrature encoder channel I
		FLTL2	12	PWM driver fault (active low)
		PWM2	11	PWM driver
		DIR2	10	PWM driver direction
		EN2L	9	PWM driver enable (active low)
		start2	8	PID start flag
		done2	7	PID done flag
		idle2	6	PID idle flag
		ready2	5	PID ready flag
		pdir_vld2	4	PID direction valid
		pdir2	3	PID direction
		pduty_vld2	2	PID duty valid
		QD_CLRN2	1	Quadrature decoder clear (active low)
		PID_RSTN2	0	PID reset (active low)
PID3	4	QEA3	15	Quadrature encoder channel A
		QEB3	14	Quadrature encoder channel B
		QEI3	13	Quadrature encoder channel I
		FLTL3	12	PWM driver fault (active low)
		PWM3	11	PWM driver
		DIR3	10	PWM driver direction
		EN3L	9	PWM driver enable (active low)
		start3	8	PID start flag
		done3	7	PID done flag
		idle3	6	PID idle flag
		ready3	5	PID ready flag
		pdir_vld3	4	PID direction valid
		pdir3	3	PID direction
		pduty_vld3	2	PID duty valid
		QD_CLRN3	1	Quadrature decoder clear (active low)
		PID_RSTN3	0	PID reset (active low)
KBDM	5	PB0	15	Pushbutton #0
		PB1	14	Pushbutton #1
		PB2	13	Pushbutton #2
		PB3	12	Pushbutton #3
		PBLEDG0	11	Pushbutton #0 green LED
		PBLEDR0	10	Pushbutton #0 red LED
		PBLEDG1	9	Pushbutton #1 green LED
		PBLEDR1	8	Pushbutton #1 red LED
		PBLEDG2	7	Pushbutton #2 green LED
		PBLEDR2	6	Pushbutton #2 red LED
		PBLEDG3	5	Pushbutton #3 green LED
		PBLEDR3	4	Pushbutton #3 red LED
LCDM	6	LDMTX	15	LCD module RS232 (SVTTL) transmit
		LDMCSL	14	LCD module SPI chip select
		LDMSCK	13	LCD module SPI clock
		LDMMOSI	12	LCD module SPI master out slave in
		LDMR	11	LCD module red LED
		LDMG	10	LCD module blue LED
		LDMB	9	LCD module green LED
MISC	7	MICSDI	15	Microphone SPI data
		MICSCK	14	Microphone SPI clock
		MICSS0	13	Microphone SPI select (internal)
		MICSS1	12	Microphone SPI select (external)
		EEPROM	11	1-Wire Serial EEPROM
		PSYNC	10	Power supply synchronization
		FAN	9	FAN enable
		MIDIN	8	MIDI in
		MIDOUT	7	MIDI out
		DACSYNC	5	DAC synchronization
		DACSDO	4	DAC SPI serial data out
		DACCSN	3	DAC SPI chip select
		DACSCK	2	DAC SPI clock

Wiring Diagram

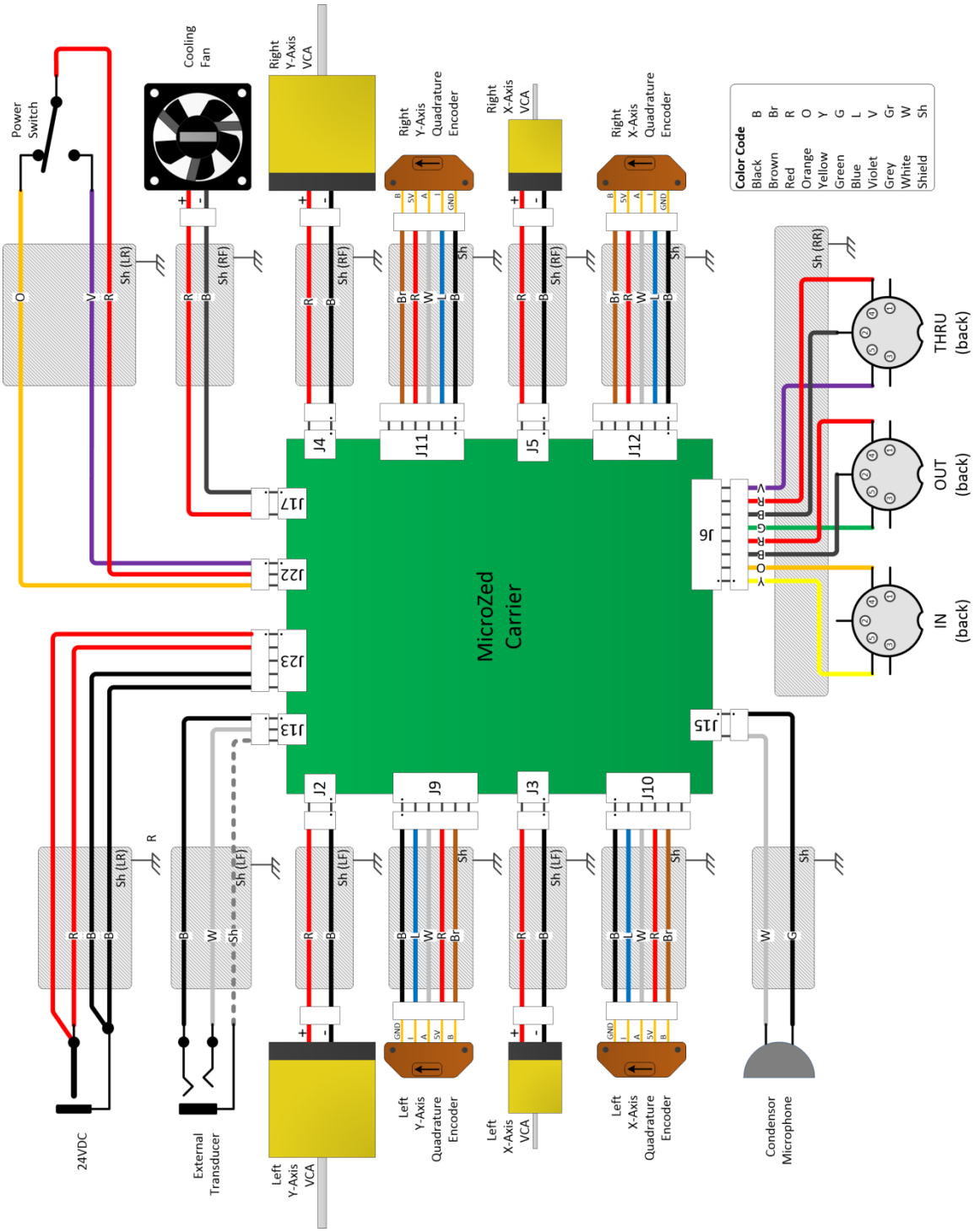


Figure 99. Wiring diagram.

Carrier Module Bill of Materials

Table 10. Carrier module BOM.

Item	Reference	Value	Mfg Part Num	Pins	Pattern	Qty	Attrition	Order	Vendor	Vendor Part Num	Description
1	C21, C39, C41, C43, C45, C47, C49, C54, C57, C60, C63, C66, C70, C71, C73, C76, C77, C78, C79, C80, C81, C82, C83, C85, C87, C89, C91, C92, C93, C95,	0.1uF	GCM21BR71H104KA37L	2	CAP_0805	45	10%	50	Digi-Key	490-6064-1-ND	0.10µF 50V Ceramic Capacitor X7R 0805 (2012 Metric) 0.079" L x 0.049" W (2.00mm x 1.25mm)
2	C2, C5, C8, C11	330uF	EEE-HA1V331P	2	EEE-HA1V331P	4	10%	4	Digi-Key	PCE4245CT-ND	330µF 35V Aluminum Capacitors Radial, Can - SMD 1000 Hrs @ 105°C
3	C4, C7, C10, C13, C14, C15, C16, C17, C55, C58, C61, C64, C69	0.01uF	GRM216R71H103KA01D	2	CAP_0805	13	10%	14	Digi-Key	490-1664-1-ND	10000pF 50V Ceramic Capacitor X7R 0805 (2012 Metric) 0.079" L x 0.049" W (2.00mm x 1.25mm)
4	C20, C29	2.2uF	GRM155R60J225ME15D	2	CAP_0402	2	10%	2	Digi-Key	490-4519-1-ND	2.2µF 6.3V Ceramic Capacitor X5R 0402 (1005 Metric) 0.039" L x 0.020" W (1.00mm x 0.50mm)
5	C22, C31	4.7uF	GRM32ER71H475KA88L	2	CAP_1210	2	10%	2	Digi-Key	490-1864-1-ND	4.7µF 50V Ceramic Capacitor X7R 1210 (3225 Metric) 0.126" L x 0.098" W (3.20mm x 2.50mm)
6	C23, C32	1uF	C3216X5R1H105K	2	CAP_1206	2	10%	2	Digi-Key	445-4044-1-ND	1µF 50V Ceramic Capacitor X5R 1206 (3216 Metric) 0.126" L x 0.063" W (3.20mm x 1.60mm)
7	C24, C27, C30, C37	47uF	GRM32ER61A476KE20L	2	CAP_1210	4	10%	4	Digi-Key	490-3887-1-ND	47µF 10V Ceramic Capacitor X5R 1210 (3225 Metric) 0.126" L x 0.098" W (3.20mm x 2.50mm)
8	C25, C33	0.47uF	GRM155C80J474KE19D	2	CAP_0402	2	10%	2	Digi-Key	490-6284-1-ND	0.47µF 6.3V Ceramic Capacitor X6S 0402 (1005 Metric) 0.039" L x 0.020" W (1.00mm x 0.50mm)
9	C26, C34	4.7uF	GRM21BR61E475KA12L	2	CAP_0805	2	10%	2	Digi-Key	490-3335-1-ND	4.7µF 25V Ceramic Capacitor X5R 0805 (2012 Metric) 0.079" L x 0.049" W (2.00mm x 1.25mm)
10	C28, C36	33pF	C0805C330J5GACTU	2	CAP_0805	2	10%	2	Digi-Key	399-1115-1-ND	33pF 50V Ceramic Capacitor C0G, NP0 0805 (2012 Metric) 0.079" L x 0.049" W (2.00mm x 1.25mm)
11	C40, C44, C46	100uF	EEE-1VA101XP	2	EEE-1VA101XP	3	10%	3	Digi-Key	PCE3951CT-ND	100µF 35V Aluminum Capacitors Radial, Can - SMD 2000 Hrs @ 85°C
12	C38, C42, C49, C117, C118, C119, C121, C122	2.2uF	GRM219R61H225KE15D	2	CAP_0805	8	10%	9	Digi-Key	490-10744-1-ND	2.2µF 50V Ceramic Capacitor X5R 0805 (2012 Metric) 0.079" L x 0.049" W (2.00mm x 1.25mm)
13	C67	100pF	GRM2165C1H101JA01D	2	CAP_0805	1	10%	1	Digi-Key	490-1615-1-ND	100pF 50V Ceramic Capacitor C0G, NP0 0805 (2012 Metric) 0.079" L x 0.049" W (2.00mm x 1.25mm)
14	C68	1uF	GRM219R61H105KA73D	2	CAP_0805	1	10%	1	Digi-Key	490-10743-1-ND	1µF 50V Ceramic Capacitor X5R 0805 (2012 Metric) 0.079" L x 0.049" W (2.00mm x 1.25mm)
15	C72	47pF	GQM2195C1H470JB01D	2	CAP_0805	1	10%	1	Digi-Key	490-3612-1-ND	47pF 50V Ceramic Capacitor C0G, NP0 0805 (2012 Metric) 0.079" L x 0.049" W (2.00mm x 1.25mm)
16	C74	2000pF	GRM2165C1H202JA01D	2	CAP_0805	1	10%	1	Digi-Key	490-1627-1-ND	2000pF 50V Ceramic Capacitor C0G, NP0 0805 (2012 Metric) 0.079" L x 0.049" W (2.00mm x 1.25mm)
17	C75	1000pF	GRM216R71H102KA01D	2	CAP_0805	1	10%	1	Digi-Key	490-9741-1-ND	1000pF 50V Ceramic Capacitor X7R 0805 (2012 Metric) 0.079" L x 0.049" W (2.00mm x 1.25mm)
18	C84, C86, C88, C90	22uF	GRM21BR60J226ME39L	2	CAP_0805	4	10%	4	Digi-Key	490-1719-1-ND	22µF 6.3V Ceramic Capacitor X5R 0805 (2012 Metric) 0.079" L x 0.049" W (2.00mm x 1.25mm)
19	C94, C96	3000pF	GRM216R71H302JA01D	2	CAP_0805	2	10%	2	Digi-Key	490-12555-1-ND	3000pF 50V Ceramic Capacitor X7R 0805 (2012 Metric) 0.079" L x 0.049" W (2.00mm x 1.25mm)
20	C99, C100, C101, C102, C103, C104, C105, C106, C107, C108, C109, C110, C111, C112, C113, C114, C115	1000pF	GRM188R71H102KA01D	2	CAP_0603	17	10%	19	Digi-Key	490-1494-1-ND	1000pF 50V Ceramic Capacitor X7R 0603 (1608 Metric) 0.063" L x 0.031" W (1.60mm x 0.80mm)
21	D1		S1G	2	DO214	1	0%	1	Digi-Key	S1GFSCT-ND	Diode Standard 400V 1A Surface Mount SMA (DO-214AC)
22	D6		1N4148	2	SOD-123	1	0%	1	Digi-Key	1N4148WTPMSCT-ND	Diode Standard 100V 150mA Surface Mount SOD-123

Item	Reference	Value	Mfg Part Num	Pins	Pattern	Qty	Attrition	Order	Vendor	Vendor Part Num	Description
23	D7		APT2012LVBC/D	2	DIODE_0805	1	0%	1	Digi-Key	754-1938-1-ND	Blue 470nm LED Indication - Discrete 2.65V 0805 (2012 Metric)
24	D8, D26, D29		APT2012LZGCK	2	DIODE_0805	3	0%	3	Digi-Key	754-1939-1-ND	Green 525nm LED Indication - Discrete 2.65V 0805 (2012 Metric)
25	D9, D10		BAT54LT1G	2	SOT23	2	0%	2	Digi-Key	BAT54LT1GOSCT-ND	Diode Schottky 30V 200mA (DC) Surface Mount SOT-23-3 (TO-236)
26	D11, D12, D13, D14, D15, D16, D17, D18, D19, D20, D21, D22, D23, D24, D25,		MMSZ4678	2	SOD-123	17	0%	17	Digi-Key	MMSZ4678-TPMSCT-ND	Zener Diode 1.8V 500mW ±5% Surface Mount SOD-123
27	F1, F2, F3, F6	3A	0685T3000-01	2	IND_1206	4	10%	4	Digi-Key	507-1896-1-ND	FUSE 3.0A 63VAC/DC SLOW 1206
28	F4	5A	0685T5000-01	2	IND_1206	1	10%	1	Digi-Key	507-1948-1-ND	FUSE BOARD MNT 5A 63VAC/63VDC
29	JX1, JX2		61083-103402LF	100	61083-103402LF	2	0%	2	Digi-Key	609-1695-1-ND	100 Position Connector Plug, Center Strip Contacts Surface Mount Gold
39	K1		TXS2-4.5V	8	TXD2	1	0%	1	Digi-Key	255-1896-ND	General Purpose Relay DPDT (2 Form C) Through Hole
40	L1, L2	8.2uH	SRU1048-8R2Y	2	SRU1048-8R2Y	2	10%	2	Digi-Key	SRU1048-8R2YCT-ND	8.2µH Shielded Wirewound Inductor 4.6A 15 mOhm Nonstandard
41	L4, L9, L10, L12, L13, L14		BLM18PG471SN1D	2	IND_0603	6	10%	7	Digi-Key	490-5223-1-ND	FERRITE BEAD 470 OHM 0603 1LN
42	L3, L5, L6, L7, L8, L11		BLE32PN300SN1L	2	IND_1210	6	10%	7	Digi-Key	490-9566-1-ND	FERRITE BEAD 30 OHM 1210 1LN
43	Q4, Q5, Q6, Q7		BSS138PS	6	SOT363	4	0%	4	Digi-Key	568-8393-1-ND	MOSFET 2N-CH 60V 0.32A 6TSSOP
44	Q8, Q9		BSS84DW	6	SOT363	2	0%	2	Digi-Key	BSS84DW-FDICT-ND	MOSFET 2P-CH 50V 0.13A SC70-6
45	Q12		BSS84	3	SOT23	1	0%	1	Digi-Key	BSS84PH6327XTSA2CT-ND	MOSFET P-CH 60V 170MA SOT-23
46	Q14, Q17, Q18, Q25		BSS138	3	SOT23	4	0%	4	Digi-Key	BSS138CT-ND	MOSFET N-CH 50V 220MA SOT-23
47	R145	3.3K	ERJ-6GEYJ332V	2	RES_0805	1	10%	1	Digi-Key	P3.3KACT-ND	RES SMD 3.3K OHM 5% 1/8W 0805
48	R13	300	ERJ-6GEYJ301V	2	RES_0805	1	10%	1	Digi-Key	P300ACT-ND	RES SMD 300 OHM 5% 1/8W 0805
49	R14, R197, R200, R201	120	ERJ-6GEYJ121V	2	RES_0805	4	10%	4	Digi-Key	P120ACT-ND	RES SMD 120 OHM 5% 1/8W 0805
50	R15	130	ERJ-6GEYJ131V	2	RES_0805	1	10%	1	Digi-Key	P130ACT-ND	RES SMD 130 OHM 5% 1/8W 0805
51	R16, R17, R18, R32, R33, R35, R36, R37, R38, R40, R42, R44, R45, R46, R48, R50, R51, R52, R54, R57, R71, R103, R108, R113, R118, R120, R126, R128, R134, R157, R158,	10K	ERJ-6GEYJ103V	2	RES_0805	43	10%	47	Digi-Key	P10KACT-ND	RES SMD 10K OHM 5% 1/8W 0805
52	R19, R20, R21, R41, R43, R49, R55, R61, R131, R153, R172,	1K	ERJ-6GEYJ102V	2	RES_0805	16	10%	18	Digi-Key	P1.0KACT-ND	RES SMD 1K OHM 5% 1/8W 0805
54	R31, R56, R79	0	ERJ-6GEY0R00V	2	RES_0805	3	10%	3	Digi-Key	P0.0ACT-ND	RES SMD 0.0 OHM JUMPER 1/8W 0805
55	R34, R39, R47, R53	442 1%	ERJ-6ENF4420V	2	RES_0805	4	10%	4	Digi-Key	P442CCT-ND	RES SMD 442 OHM 1% 1/8W 0805
56	R58, R78	1M 1%	ERJ-6ENF1004V	2	RES_0805	2	10%	2	Digi-Key	P1.00MCCT-ND	RES SMD 1M OHM 1% 1/8W 0805
57	R59	422K 1%	ERJ-6ENF4223V	2	RES_0805	1	10%	1	Digi-Key	P422KCCT-ND	RES SMD 422K OHM 1% 1/8W 0805
58	R60, R195	680	ERJ-6GEYJ681V	2	RES_0805	2	10%	2	Digi-Key	P680ACT-ND	RES SMD 680 OHM 5% 1/8W 0805
60	R80	255K 1%	ERJ-6ENF2553V	2	RES_0805	1	10%	1	Digi-Key	P255KCCT-ND	RES SMD 255K OHM 1% 1/8W 0805
62	R99, R100, R101, R104, R105, R106, R109, R110, R111,	5.6K	ERJ-6GEYJ562V	2	RES_0805	12	10%	13	Digi-Key	P5.6KACT-ND	RES SMD 5.6K OHM 5% 1/8W 0805
63	R102, R107, R112, R117	43K	ERJ-6GEYJ433V	2	RES_0805	4	10%	4	Digi-Key	P43KACT-ND	RES SMD 43K OHM 5% 1/8W 0805
64	R121, R122	1M	ERJ-6GEYJ105V	2	RES_0805	2	10%	2	Digi-Key	P1.0MACT-ND	RES SMD 1M OHM 5% 1/8W 0805
65	R123, R124, R212, R229	2K	ERJ-6GEYJ202V	2	RES_0805	4	10%	4	Digi-Key	P2.0KACT-ND	RES SMD 2K OHM 5% 1/8W 0805
66	R125, R133, R204, R206	40.2K	ERJ-6ENF4022V	2	RES_0805	4	10%	4	Digi-Key	P40.2KCCT-ND	RES SMD 40.2K OHM 1% 1/8W 0805
67	R129, R130	100K	ERJ-6GEYJ104V	2	RES_0805	2	10%	2	Digi-Key	P100KACT-ND	RES SMD 100K OHM 5% 1/8W 0805
68	R132	2.2K	ERJ-6GEYJ222V	2	RES_0805	1	10%	1	Digi-Key	P2.2KACT-ND	RES SMD 2.2K OHM 5% 1/8W 0805
69	R135	470K 1%	ERJ-6ENF4703V	2	RES_0805	1	10%	1	Digi-Key	P470KCCT-ND	RES SMD 470K OHM 1% 1/8W 0805
70	R136, R138, R140, R142, R205, R207	10K 1%	ERJ-6ENF1002V	2	RES_0805	6	10%	7	Digi-Key	P10.0KCCT-ND	RES SMD 10K OHM 1% 1/8W 0805
71	R137	232K 1%	ERJ-6ENF2323V	2	RES_0805	1	10%	1	Digi-Key	P232KCCT-ND	RES SMD 232K OHM 1% 1/8W 0805
72	R139	91K 1%	ERJ-6ENF9102V	2	RES_0805	1	10%	1	Digi-Key	P91.0KCCT-ND	RES SMD 91K OHM 1% 1/8W 0805
73	R141	56K 1%	ERJ-6ENF5602V	2	RES_0805	1	10%	1	Digi-Key	P56.0KCCT-ND	RES SMD 56K OHM 1% 1/8W 0805

Item	Reference	Value	Mfg Part Num	Pins	Pattern	Qty	Attrition	Order	Vendor	Vendor Part Num	Description
74	R143, R174, R175, R176, R177, R178, R179, R180, R181, R183, R184, R185, R186, R187, R188, R189, R208, R209, R217, R218, R219, R220, R221, R222, R223, R224, R225, R226, R227, R228	100	ERJ-3GEYJ101V	2	RES_0603	30	10%	33	Digi-Key	P100GCT-ND	RES SMD 100 OHM 5% 1/10W 0603
75	R146, R149, R150, R151, R152, R199	220	ERJ-6GEYJ221V	2	RES_0805	6	10%	7	Digi-Key	P220ACT-ND	RES SMD 220 OHM 5% 1/8W 0805
77	R156, R169, R193	330	ERJ-6GEYJ331V	2	RES_0805	3	10%	3	Digi-Key	P330ACT-ND	RES SMD 330 OHM 5% 1/8W 0805
78	R194	1.2K	ERJ-6GEYJ122V	2	RES_0805	1	10%	1	Digi-Key	P1.2KACT-ND	RES SMD 1.2K OHM 5% 1/8W 0805
79	R196	430	ERJ-6GEYJ431V	2	RES_0805	1	10%	1	Digi-Key	P430ACT-ND	RES SMD 430 OHM 5% 1/8W 0805
80	R198, R230	3K92	ERJ-6ENF3921V	2	RES_0805	2	10%	2	Digi-Key	P3.92KCCT-ND	RES SMD 3.92K OHM 1% 1/8W 0805
81	R202, R203	10m 1%	ERJ-6BWFR010V	2	RES_0805	2	10%	2	Digi-Key	P.01AQCT-ND	RES SMD 0.01 OHM 1% 1/2W 0805
83	U5, U6		LM43603PWP	17	TSSOP-16EP	2	0%	2	Digi-Key	296-44477-1-ND	Buck Switching Regulator IC Positive Adjustable 1V 1 Output 3A 16-TSSOP (0.173", 4.40mm Width) Exposed Pad
84	U7		DS28E01P-100+T	6	TSOC-6	1	0%	1	Digi-Key	DS28E01P-100+CT-ND	IC EEPROM 1KBIT 6TSOC
85	U11		MAX4466EXK+T	5	SOT23-5	1	0%	1	Digi-Key	MAX4466EXK+CT-ND	Amplifier IC 1-Channel (Mono) Class AB SC-70-5
86	U8, U9, U10, U12, U18, U19		AD8606ARMZ	8	MSOP-8	6	0%	6	Digi-Key	AD8606ARMZREELCT-ND	General Purpose Amplifier 2 Circuit Rail-to-Rail 8-MSOP
87	U13		MCP9700T-E/TT	3	SOT23	1	0%	1	Digi-Key	MCP9700T-E/TTCT-ND	Temperature Sensor Analog, Local -40°C ~ 125°C 10mV/°C SOT-23-3
88	U14		SN74LS14D	12	SOIC-14/150mil	1	0%	1	Digi-Key	296-1212-1-ND	Inverter IC 6 Channel Schmitt Trigger 14-SOIC
89	U15		MAX4468EKA-T	8	SOT23-8	1	0%	1	Digi-Key	MAX4468EKA+CT-ND	Amplifier IC 1-Channel (Mono) Class AB SOT-23-8
90	U16, U17		INA213AIDCKR	6	SOT363	2	0%	2	Digi-Key	296-26072-1-ND	Current Sense Amplifier 1 Circuit SC-70-6
91	U21, U22		AD5160BRJZ10-R2	8	SOT23-8	2	0%	2	Digi-Key	AD5160BRJZ10-R2CT-ND	Digital Potentiometer 10k Ohm 1 Circuit 256 Taps SPI Interface SOT-23-8
92	VO2		4N38SR2M	5	6-SMD	1	0%	1	Digi-Key	4N38SR2MCT-ND	Optoisolator Transistor with Base Output 4170Vrms 1 Channel 6-SMD
				92		63		417			

Keyboard Module Bill of Materials

Table 11. Keyboard module BOM.

Item	Reference	Value	Mfg Part Num	Pins	Pattern	Qty	Vendor	Vendor Part Num	Description
1	C1, C2, C3, C4	0.1uF	GCM21BR71H104KA37K	2	CAP_0805	4	Digi-Key	490-8049-1-ND	0.10µF ±10% 50V Ceramic Capacitor X7R 0805 (2012 Metric)
3	Q1, Q2, Q3, Q4		BSS138PS	6	SOT363	4	Digi-Key	568-8393-1-ND	MOSFET 2N-CH 60V 0.32A 6TSSOP
4	R1, R3, R13, R15	270	ERJ-6GEYJ271V	2	RES_0805	4	Digi-Key	P270ACT-ND	RES SMD 270 OHM 5% 1/8W 0805
5	R2, R4, R14, R16	150	ERJ-6GEYJ151V	2	RES_0805	4	Digi-Key	P150ACT-ND	RES SMD 150 OHM 5% 1/8W 0805
6	R5, R6, R17, R18	100	ERJ-6GEYJ101V	2	RES_0805	4	Digi-Key	P100ACT-ND	RES SMD 100 OHM 5% 1/8W 0805
7	R7, R8, R19, R20	3.3K	ERJ-6GEYJ332V	2	RES_0805	4	Digi-Key	P3.3KACT-ND	RES SMD 3.3K OHM 5% 1/8W 0805
8	R9, R10, R11, R12, R21, R22, R23, R24	1K	ERJ-6GEYJ102V	2	RES_0805	8	Digi-Key	P1.0KACT-ND	RES SMD 1K OHM 5% 1/8W 0805
						32			

MicroZed Board

MicroZed™ is a low-cost development board based on the Xilinx Zynq®-7000 All Programmable SoC. Its unique design allows it to be used as both a stand-alone evaluation board for basic SoC experimentation, or combined with a carrier card as an embeddable system-on-module (SOM). MicroZed contains two I/O headers that provide connection to two I/O banks on the programmable logic (PL) side of the Zynq®-7000 All Programmable SoC device. In stand-alone mode, these 100 PL I/O are inactive. When plugged into a carrier card, the I/O are accessible in a manner defined by the carrier card design [source: zedboard.org/product/microzed].

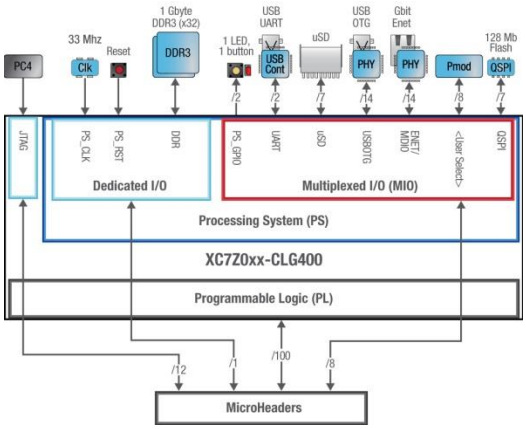


Figure 100. MicroZed block diagram (image provided by AVNET, Inc.).

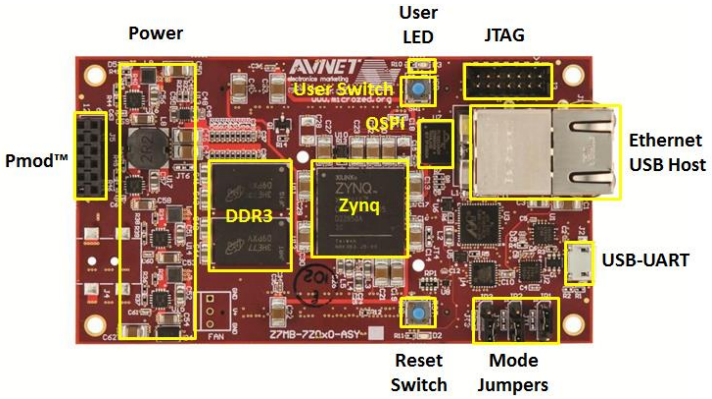


Figure 101. MicroZed functional overlay (image provided by AVNET, Inc.)

Appendix G

Mechanical Design Bill of Materials

Table 12. Mechanical BOM.

item	part	mfg	mfg #	description	specification	material	qt
1	100001	Summit Scientific, Inc.	100001	Enclosure, Top		Aluminum 6061 T6, 0.375"	1
2	100002	Summit Scientific, Inc.	100002	Enclosure, Left		Aluminum 6061 T6, 0.375"	1
3	100003	Summit Scientific, Inc.	100003	Enclosure, Right		Aluminum 6061 T6, 0.375"	1
4	100004	Summit Scientific, Inc.	100004	Enclosure, Bottom		Aluminum 6061 T6, 0.375"	1
5	100005	Summit Scientific, Inc.	100005	Bulkhead Panel, Top Rear		Aluminum 6061 T6, 0.032"	1
6	100006	Summit Scientific, Inc.	100006	Bulkhead Panel, Bottom Rear		Aluminum 6061 T6, 0.032"	1
7	100007	Summit Scientific, Inc.	100007	Panel, Rear		Acrylic Lexan, 0.125"	1
8	100008	Summit Scientific, Inc.	100008	Panel, Front		Acrylic Lexan, 0.125"	1
9	100009	Summit Scientific, Inc.	100009	Bulkhead Panel, Top Front		Aluminum 6061 T6, 0.032"	1
10	100010	Summit Scientific, Inc.	100010	Bulkhead Panel, Bottom Front		Aluminum 6061 T6, 0.032"	1
11	100011	Summit Scientific, Inc.	100011	Cover, Bottom		Aluminum 6061 T6, 0.025"	1
12	100012	Summit Scientific, Inc.	100012	Motor Mount		Aluminum 6061 T6, 0.375"	1
13	100013	Summit Scientific, Inc.	100013	Pivot Block		Aluminum 6061 T6, 0.375"	1
14	100014	Summit Scientific, Inc.	100014	Encoder Mount, X-Axis		Aluminum 6061 T6, 0.125"	2
15	100015	Summit Scientific, Inc.	100015	Tripod Mount		Stainless Steel 304	1
16	100016	Summit Scientific, Inc.	100016	Pivot Arm, Upper		Aluminum 6061 T6, 0.1875"	2
17	100017	Summit Scientific, Inc.	100017	Pivot Arm, Lower		Aluminum 6061 T6, 0.1875"	2
18	100018	Summit Scientific, Inc.	100018	Linkage, Y-Axis		Aluminum 6061 T6, 0.1875"	2
19	100019	Summit Scientific, Inc.	100019	Coupler, VCA Y-Axis		Aluminum 7075 T6, 0.625" diameter	2
20	100020	Summit Scientific, Inc.	100020	Coupler, VCA X-Axis		Aluminum 7075 T6, 0.625" diameter	2
21	100021	Summit Scientific, Inc.	100021	Coupler, Control Rod		Aluminum 7075 T6, 0.625" diameter	2
22	100022	Summit Scientific, Inc.	100022	Coupler, Implement Shuttle		Aluminum 7075 T6, 0.625" diameter	2
23	100023	Summit Scientific, Inc.	100023	Lever Arm		Aluminum 6061 T6, 0.250"	2
24	100024	Summit Scientific, Inc.	100024	Control Rod		Stainless Steel 303	2
25	100025	Summit Scientific, Inc.	100025	Shaft, Y-Axis		Stainless Steel 316	2
26	100026	Summit Scientific, Inc.	100026	Shaft, Pivot		Stainless Steel 303	2
27	100027	Summit Scientific, Inc.	100027	Fork, Base		Aluminum 6061 T6, 0.375"	2
28	100028	Summit Scientific, Inc.	100028	Fork, Left		Aluminum 6061 T6, 0.250"	2
29	100029	Summit Scientific, Inc.	100029	Fork, Right		Aluminum 6061 T6, 0.250"	2
30	100030	Summit Scientific, Inc.	100030	Implement Shuttle		Aluminum 6061 T6, 1.125"	2
31	100031	Summit Scientific, Inc.	100031	Implement Collet		Aluminum 6061 T6, 0.750" diameter	2
32	100032	Summit Scientific, Inc.	100032	Implement Insulator		Rubber Tubing	2
33	100033	Summit Scientific, Inc.	100033	Pinslot Bushing, Linkage		Delrin	4
34	100034	Summit Scientific, Inc.	100034	Pinslot Washer, Linkage		Delrin	8
35	100035	Summit Scientific, Inc.	100035	Pinslot Bushing, X-Axis		Delrin	6
36	100036	Summit Scientific, Inc.	100036	Pinslot Washer, X-Axis		Delrin	12
37	100037	Summit Scientific, Inc.	100037	Pin, Implement Shuttle		Stainless Steel 304	2
38	100038	Summit Scientific, Inc.	100038	Washer, Implement Shuttle		Delrin	4
39	100039	Summit Scientific, Inc.	100039	Axial Washer, Y-Axis		Delrin	2
40	100040	Summit Scientific, Inc.	100040	Washer, Pivot		Delrin	4
41	100041	Summit Scientific, Inc.	100041	Shaft Bushing, Y-Axis		Delrin	4
42	100042	Summit Scientific, Inc.	100042	PCBA, Carrier Module		FR4	1
43	100043	Summit Scientific, Inc.	100043	PCBA, Keyboard Module		FR4	1
44	100044	Summit Scientific, Inc.	100044	PCBA, MicroZed		FR4	1
45	100045	Summit Scientific, Inc.	100045	PCBA, LED (16 x 2) Display		Assembly	1

item	part	mfg	mfg #	description	specification	material	qt
46	100046	MotiCont	GVCM-051-051-01	VCA, Y-Axis			2
47	100047	MotiCont	GVCM-025-029-01	VCA, X-Axis			2
48	100048	Avnet	HEDM-5120#B06	Codewheel, Pivot			2
49	100049	Avnet	HEDS-6100#B08	Codewheel, Y-Axis			2
50	100050	Avnet	HEDS-9100#B00	Encoder, Pivot			2
51	100051	Avnet	HEDS-9000#B00	Encoder, Y-Axis			2
52	100052	McMaster-Carr	1688K2	Bushing, Implement Shuttle		Oil Impregnated Bronze	4
53	100053	McMaster-Carr	7804K145	Bearing, Y-Axis		Stainless Steel	4
54	100054	McMaster-Carr	57155K364	Bearing, Control Rod		Stainless Steel	2
55	100055	McMaster-Carr	1688K500	Bushing, Pivot		Oil Impregnated Bronze	2
56	100056	McMaster-Carr	91590A113	Retaining Clip, Pivot Shaft		Stainless Steel	2
57	100057	McMaster-Carr	93330A571	Standoff, MicroZed	Aluminum Female Threaded Round Standoff 1/4" OD, 9/16" Length, 4-40 Thread Size	Aluminum	4
58	100058	McMaster-Carr	93505A112	Standoff, Carrier Module	Aluminum Male-Female Threaded Hex Standoff 3/16" Hex Size, 3/8" Length, 6-32 Thread Size	Aluminum	4
59	100059	McMaster-Carr	94355A330	Implement Shuttle	18-8 Stainless Steel Flat-Tip Set Screws 10-32 Thread, 1/4" Long, 25pk	Stainless Steel 18-8	1
60	100060	Digikey	HS-002	Heatsink			4
61	100061	Digikey	36-621-ND	Cover Bracket			3
62	100062	Digikey		DIN (5-pin) Connector			3
63	100063	Digikey		1/4" Jack			1
64	100064	Digikey		Power Jack			1
65	100065	Digikey		Power Switch			1
66	100066	Digikey		Electret Microphone			1
67	100067	Digikey		Fan			1
68	100068			Drumstick			2
69	100069	Summit Scientific, Inc.	100069	Display Bezel		Acrylic Lexan, 0.125"	1
70	100070	Summit Scientific, Inc.	100070	Encoder Mount, Y-Axis		Aluminum 6061 T6, 0.125"	2
71	100071	McMaster-Carr	96006A613	Screw, Enclosure and Pivot Block	Black Oxide 18-8 Stainless Steel Socket Head Screw 8-32 Thread Size, 3/8" Long, 50pk	Stainless Steel 18-8	1
72	100072	McMaster-Carr	91251A146	Screw, Fork Left/Right	Black-Oxide Alloy Steel Socket Head Screw 6-32 Thread Size, 3/8" Long, 100pk	Alloy	1
73	100073	McMaster-Carr	96006A253	Screw, Fork Base	Black Oxide 18-8 Stainless Steel Socket Head Screw 6-32 Thread Size, 3/8" Long	Stainless Steel 18-8	1
74	100074	McMaster-Carr	90318A203	Screw, Coupler X-Axis	Ultra-Low-Profile 316 Stainless Steel shoulder Screw Socket Drive, 1/8" Diameter x 3/8" Long Shoulder, 4-40 Thread Size	Stainless Steel 316	6
75	100075	McMaster-Carr	94035A146	Screw, Linkage	18-8 Stainless Steel Tight-Tolerance Screw Socket Drive, 5/32" Diameter x 9/32" Long Shoulder, 6-32 Thread Size	Stainless Steel 18-8	4
76	100076	McMaster-Carr	97763A315	Screw; Panel Front/Rear, PCBAs	18-8 Stainless Steel Hex Drive Rounded Head Screws Black-Oxide, 2-56 Thread Size, 1/4" Long 50pk	Stainless Steel 18-8	1
77	100077	McMaster-Carr	96006A694	Screw; Motor Y-Axis, Tripod Mount	Black Oxide 18-8 Stainless Steel Socket Head Screw 10-32 Thread Size, 3/8" Long 50pk	Stainless Steel 18-8	1
78	100078	McMaster-Carr	97763A102	Screw; Encoder Mount, Heatsink, PCBA, cover bracket	18-8 Stainless Steel Hex Drive Rounded Head Screws Black-Oxide, 4-40 Thread Size, 1/4" Long 100pk	Stainless Steel 18-8	1
79	100079	McMaster-Carr	93791A411	Screw, Motor X-Axis	Black-Oxide 18-8 Stainless Steel Hex Drive Flat Head Screws 6-32 Thread Size, 5/16" Long 100pk	Stainless Steel 18-8	1

item	part	mfg	mfg #	description	specification	material	q
80	100080	McMaster-Carr	97763A317	Screw, Encoder	18-8 Stainless Steel Hex Drive Rounded Head Screws Black-Oxide, 2-56 Thread Size, 1/2" Long 50pk	Stainless Steel 18-8	1
81	100081	McMaster-Carr	96006A210	Screw, Lever Arm	Black Oxide 18-8 Stainless Steel Socket Head Screw 4-40 Thread Size, 1/4" Long 100pk	Stainless Steel 18-8	1
82	100082	McMaster-Carr	96006A217	Screw; Motor Y-Axis	Black Oxide 18-8 Stainless Steel Socket Head Screw 4-40 Thread Size, 3/4" Long 100pk	Stainless Steel 18-8	1
83	100083	McMaster-Carr	92196A081	Screw, Shaft Y-Axis	18-8 Stainless Steel Socket Head Screw 2-56 Thread Size, 1/2" Long 100pk	Stainless Steel 18-8	1
84	100084	McMaster-Carr	91864A005	Screw, Upper/Lower Arm	Black-Oxide Alloy Steel Socket Head Screw 2-56 Thread Size, 5/16" Long 25pk	Stainless Steel 18-8	1
85	100085	McMaster-Carr	91255A201	Screw, Fan	Hex Drive Rounded Head Screw Black-Oxide Alloy Steel, 8-32 Thread Size, 1-1/4" Long 10pk	Alloy Steel	1
86	100086	McMaster-Carr	92362A130	Nut, Fan	18-8 Stainless Steel Narrow Hex Nut Black-Oxide, 8-32 Thread Size, 1/4" Wide 100pk	Stainless Steel 18-8	1
87	100087	McMaster-Carr	97763A103	Screw, DIN Connector	18-8 Stainless Steel Hex Drive Rounded Head Screws Black-Oxide, 4-40 Thread Size, 3/8" Long 100pk	Stainless Steel 18-8	1
88	100088	McMaster-Carr	90730A005	Nut, DIN Connector	18-8 Stainless Steel Narrow Hex Nut 4-40 Thread Size 100pk	Stainless Steel 18-8	1
89	100089	McMaster-Carr	92949A144	Screw, Carrier Module	18-8 Stainless Steel Hex Drive Rounded Head Screw 6-32 Thread Size, 1/4" Long 100pk	Stainless Steel 18-8	1
90	100090	McMaster-Carr	95601A295	Washer, MicroZed	Electrical-Insulating Hard Fiber Washer for Number 4 Screw, 0.125" ID, 0.25" OD, 0.028"-0.034" Thickness 100pk	Fiber	1
91	100091	McMaster-Carr	1478T1	Cable Sleeve	Super-Abrasion-Resistant Expandable Sleeving Corrosion Resistant, 1/4" ID	Stainless Steel	1

Voice Coil Actuators

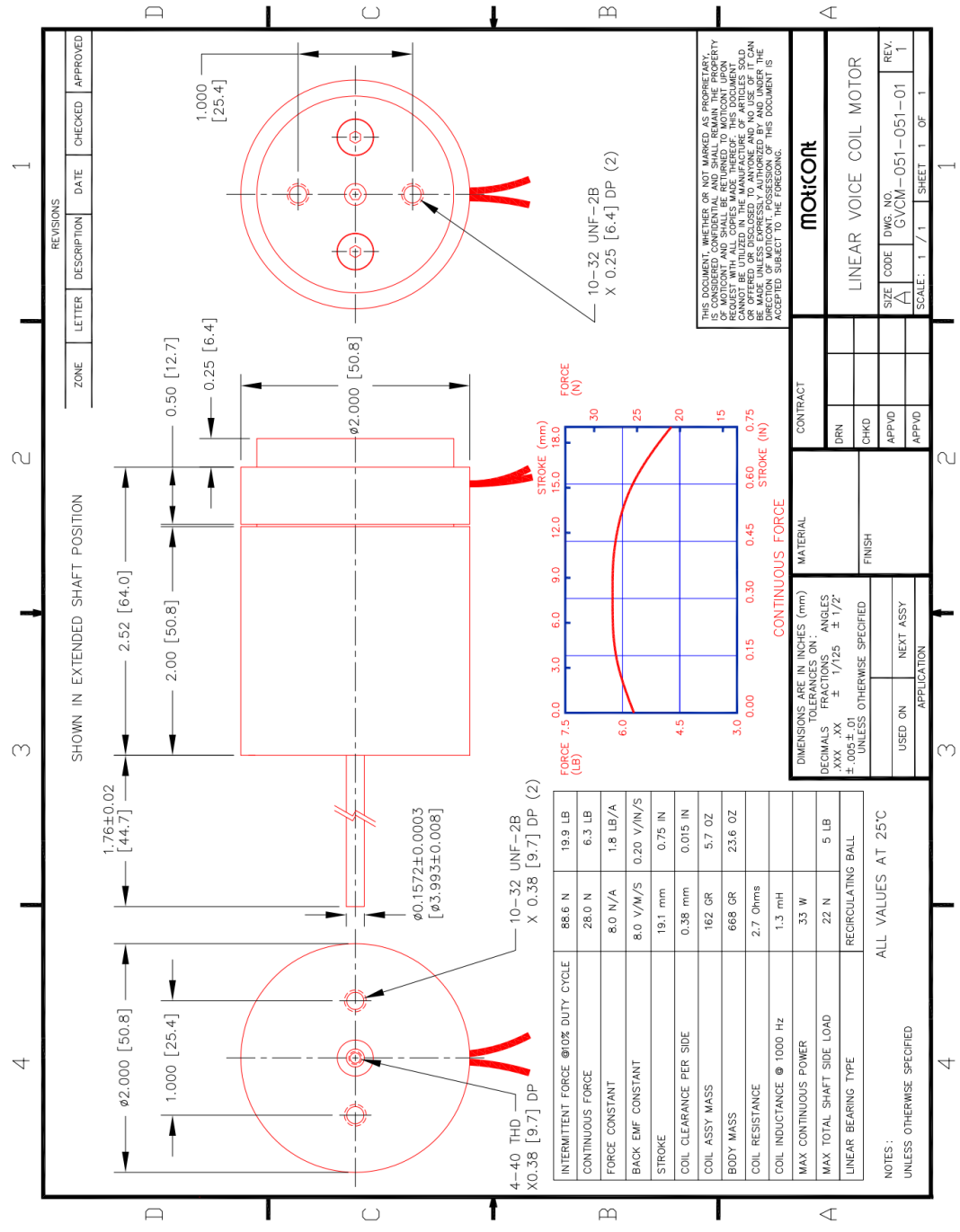


Figure 102. Y axis voice coil actuator specifications (provided by MotiCont).

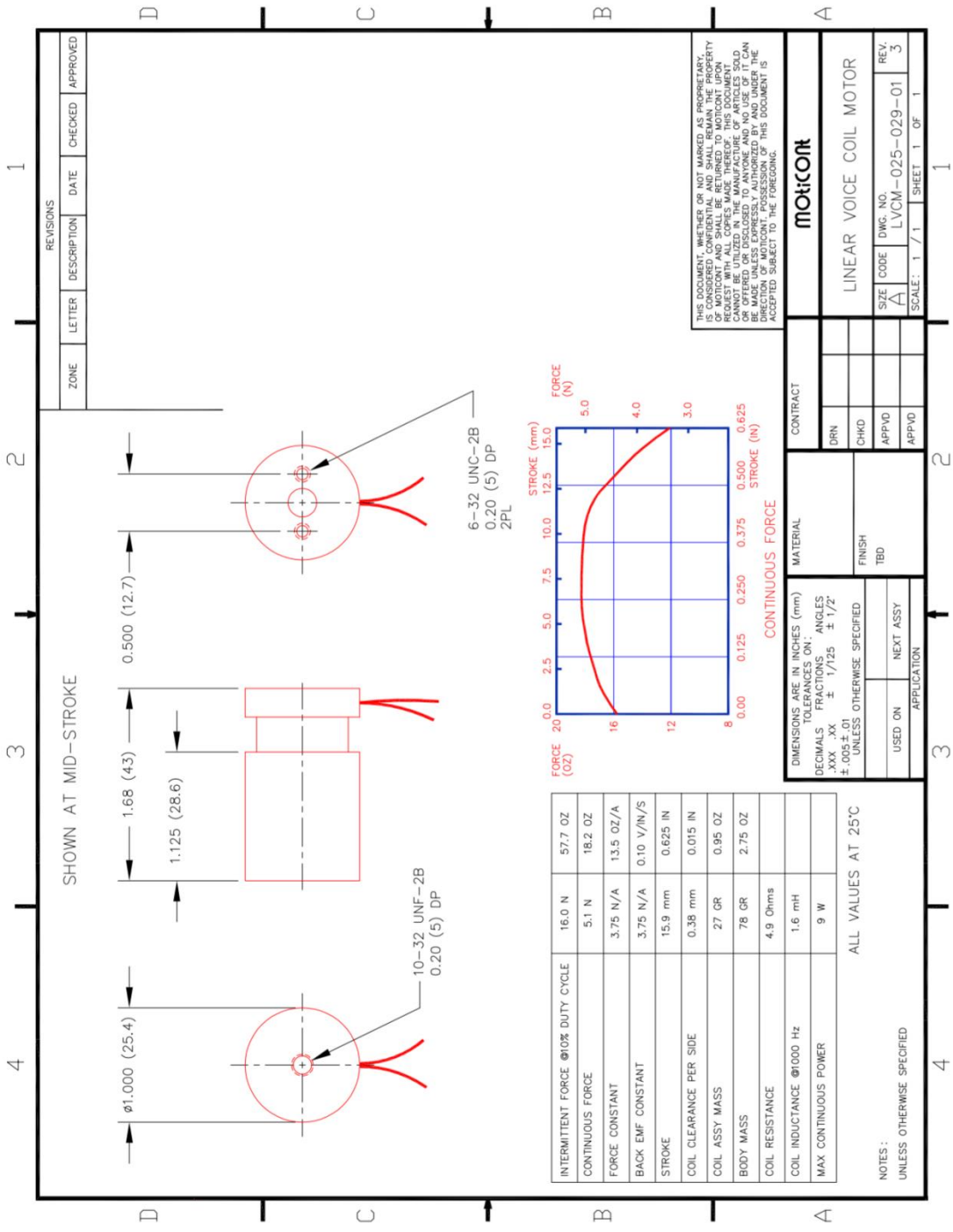


Figure 103. X axis voice coil actuator specifications (provided by MotiCont).

Appendix H

MIDI Mapping

Table 13. MIDI Mapping chart.

note	value	map	velocity	pressure	modulation	volume	pitch
C3	36	Left Strike	0-127	-	-	0-127	0-16383
C#3	37	Left Quarter Roll	0-127	0-127	0-127	0-127	0-16383
D3	38	Left Eighth Roll	0-127	0-127	0-127	0-127	0-16383
D#3	39	Left Sixteenth Roll	0-127	0-127	0-127	0-127	0-16383
E3	40	Left Thirtysecond Roll	0-127	0-127	0-127	0-127	0-16383
F3	41	Left Sixtyfourth Roll	0-127	0-127	0-127	0-127	0-16383
F#3	42	Left Triplet Roll	0-127	0-127	0-127	0-127	0-16383
G3	43	Left Flam	0-127	0-127	0-127	0-127	0-16383
G#3	44	Left Drag	0-127	0-127	0-127	0-127	0-16383
A3	45	Right Strike	0-127	-	-	0-127	0-16383
A#3	46	Right Quarter Roll	0-127	0-127	0-127	0-127	0-16383
B3	47	Right Eighth Roll	0-127	0-127	0-127	0-127	0-16383
C4	48	Right Sixteenth Roll	0-127	0-127	0-127	0-127	0-16383
C#4	49	Right Thirtysecond Roll	0-127	0-127	0-127	0-127	0-16383
D4	50	Right Sixtyfourth Roll	0-127	0-127	0-127	0-127	0-16383
D#4	51	Right Triplet Roll	0-127	0-127	0-127	0-127	0-16383
E4	52	Right Flam	0-127	0-127	0-127	0-127	0-16383
F4	53	Right Drag	0-127	0-127	0-127	0-127	0-16383
F#4	54	-	-	-	-	-	-
G4	55	Single Stroke Roll	0-127	0-127	0-127	0-127	0-16383
G#4	56	Single Stroke Four	0-127	0-127	0-127	0-127	0-16383
A4	57	Single Stroke Seven	0-127	0-127	0-127	0-127	0-16383
A#4	58	Multiple Bounce Roll	0-127	0-127	0-127	0-127	0-16383
B4	59	Double Stroke Roll	0-127	0-127	0-127	0-127	0-16383
C5	60	Triple Stroke Roll	0-127	0-127	0-127	0-127	0-16383
C#5	61	Five Stroke Roll	0-127	0-127	0-127	0-127	0-16383
D5	62	Six Stroke Roll	0-127	0-127	0-127	0-127	0-16383
D#5	63	Seven Stroke Roll	0-127	0-127	0-127	0-127	0-16383
E5	64	Nine Stroke Roll	0-127	0-127	0-127	0-127	0-16383
F5	65	Ten Stroke Roll	0-127	0-127	0-127	0-127	0-16383
F#5	66	Eleven Stroke Roll	0-127	0-127	0-127	0-127	0-16383
G5	67	Thirteen Stroke Roll	0-127	0-127	0-127	0-127	0-16383
G#5	68	Fifteen Stroke Roll	0-127	0-127	0-127	0-127	0-16383
A5	69	Seventeen Stroke Roll	0-127	0-127	0-127	0-127	0-16383
A#5	70	Single Paradiddle	0-127	0-127	0-127	0-127	0-16383
B5	71	Double Paradiddle	0-127	0-127	0-127	0-127	0-16383
C6	72	Triple Paradiddle	0-127	0-127	0-127	0-127	0-16383
C#6	73	Single Paradiddle-diddle	0-127	0-127	0-127	0-127	0-16383
D6	74	Flam	0-127	0-127	0-127	0-127	0-16383
D#6	75	Flam Tap	0-127	0-127	0-127	0-127	0-16383
E6	76	Flam Accent	0-127	0-127	0-127	0-127	0-16383
F6	77	Flamacue	0-127	0-127	0-127	0-127	0-16383
F#6	78	Flam Paradiddle	0-127	0-127	0-127	0-127	0-16383
G6	79	Single Flammed Mill	0-127	0-127	0-127	0-127	0-16383
G#6	80	Flam Paradiddle-diddle	0-127	0-127	0-127	0-127	0-16383
A6	81	Swiss Army Triplet	0-127	0-127	0-127	0-127	0-16383
A#6	82	Inverted Flam Tap	0-127	0-127	0-127	0-127	0-16383
B6	83	Flam Drag	0-127	0-127	0-127	0-127	0-16383
C7	84	Pataflafla	0-127	0-127	0-127	0-127	0-16383
C#7	85	Drag Ruff	0-127	0-127	0-127	0-127	0-16383
D7	86	Single Drag Tap	0-127	0-127	0-127	0-127	0-16383
D#7	87	Double Drag Tap	0-127	0-127	0-127	0-127	0-16383
E7	88	Lesson 25	0-127	0-127	0-127	0-127	0-16383
F7	89	Single Dragadiddle	0-127	0-127	0-127	0-127	0-16383
F#7	90	Dragadiddle #1	0-127	0-127	0-127	0-127	0-16383
G7	91	Dragadiddle #2	0-127	0-127	0-127	0-127	0-16383
G#7	92	Single Ratamacue	0-127	0-127	0-127	0-127	0-16383
A7	93	Double Ratamacue	0-127	0-127	0-127	0-127	0-16383
A#7	94	Triple Ratamacue	0-127	0-127	0-127	0-127	0-16383
B7	95	Metronome	0-127	-	-	0-127	0-16383
C8	96	Conductor	-	-	-	-	-

MIDI Continuous Controller

Table 14. MIDI continuous controller chart.

cc	purpose	range	description
1	modulation	0-127	global beats per minute
7	volume	0-127	prepatory height
16	onset	0-127	onset sigma gain
17	velocity	0-127	velocity sigma gain
18	location	0-127	location sigma gain
20	left modulation	0-127	left implement beats per minute
21	left location	0-127	left implement location
22	left velocity	0-127	left implement velocity
23	right modulation	0-127	right implement beats per minute
24	right location	0-127	right implement location
25	right velocity	0-127	right implement velocity

Software/Hardware Architecture

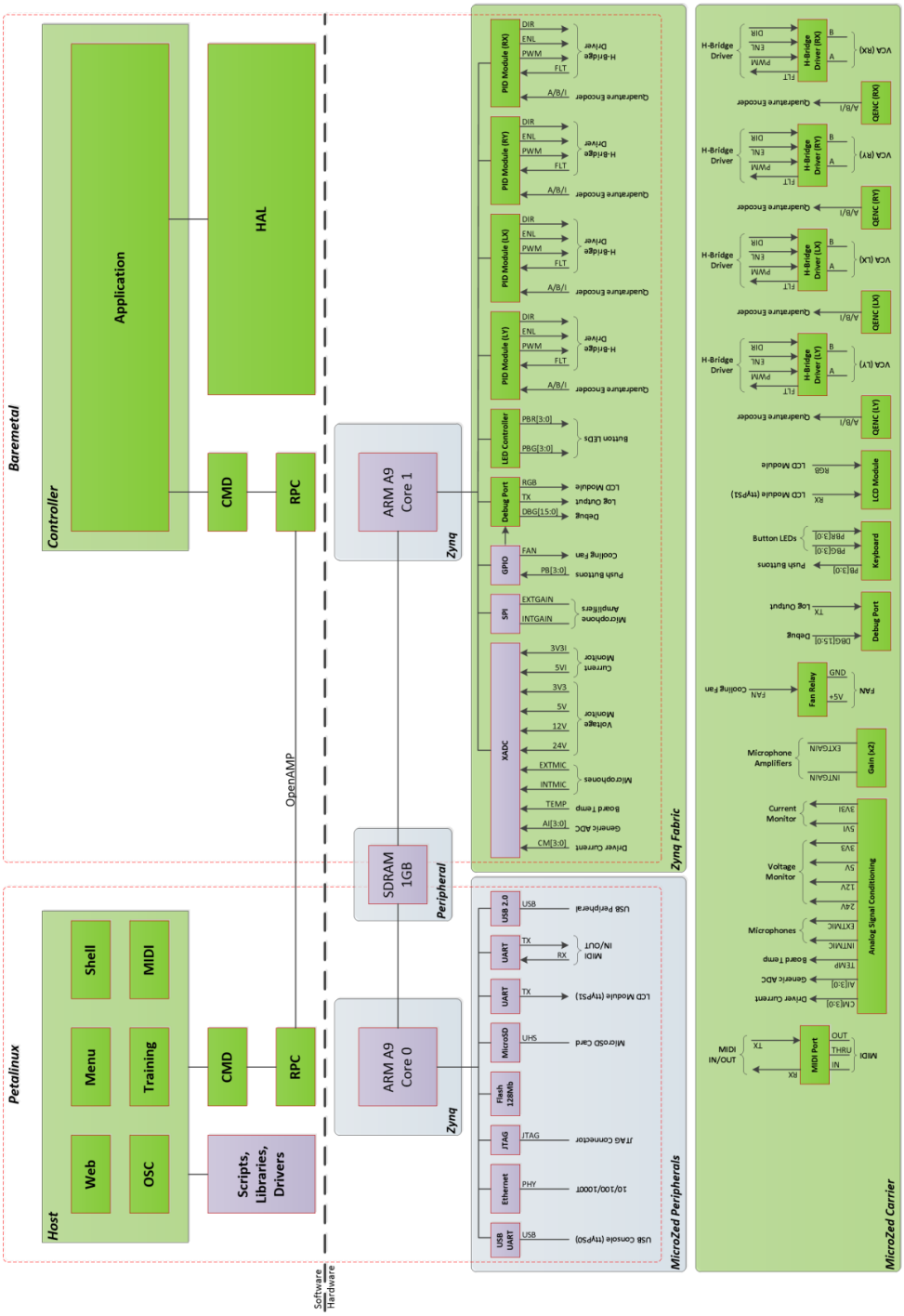


Figure 104. Software/hardware architecture.

FPGA Registers

Table 15. Field programmable gate array register summary.

Acronym	Base	Offset	Description
GP_VER	0x41200000	0x0	FPGA bitstream version
GP_CTRL	0x41200000	0x0	general purpose control
PID0_CSR	0x43C00000	0x0	control/status
PID0_ISR		0x4	interrupt status
PID0_IMR		0x8	interrupt mask
PID0_IPR		0xC	interrupt pending
PID0_STRK		0x10	strike and pre-strike timer
PID0_POST		0x14	post-strike timer
PID0_POS		0x18	position
PID0_LOC		0x1C	location
PID0_KP		0x20	proportional gain
PID0_KI		0x24	integral gain
PID0_KD		0x28	derivative gain
PID0_BIAS		0x2C	duty cycle bias
PID0_PDIV		0x30	PID clock divisor
PID0_MDTY		0x34	PWM duty cycle margin
PID0_MTIM		0x38	PWM duty cycle timeout
PID0_MDIV		0x3C	PWM clock divisor
PID1_CSR	0x43C20000	0x0	control/status
PID1_ISR		0x4	interrupt status
PID1_IMR		0x8	interrupt mask
PID1_IPR		0xC	interrupt pending
PID1_STRK		0x10	strike and pre-strike timer
PID1_POST		0x14	post-strike timer
PID1_POS		0x18	position
PID1_LOC		0x1C	location
PID1_KP		0x20	proportional gain
PID1_KI		0x24	integral gain
PID1_KD		0x28	derivative gain
PID1_BIAS		0x2C	duty cycle bias
PID1_PDIV		0x30	PID clock divisor
PID1_MDTY		0x34	PWM duty cycle margin
PID1_MTIM		0x38	PWM duty cycle timeout
PID1_MDIV		0x3C	PWM clock divisor
PID2_CSR	0x43C30000	0x0	control/status
PID2_ISR		0x4	interrupt status
PID2_IMR		0x8	interrupt mask
PID2_IPR		0xC	interrupt pending
PID2_STRK		0x10	strike and pre-strike timer
PID2_POST		0x14	post-strike timer
PID2_POS		0x18	position
PID2_LOC		0x1C	location
PID2_KP		0x20	proportional gain
PID2_KI		0x24	integral gain
PID2_KD		0x28	derivative gain
PID2_BIAS		0x2C	duty cycle bias
PID2_PDIV		0x30	PID clock divisor
PID2_MDTY		0x34	PWM duty cycle margin
PID2_MTIM		0x38	PWM duty cycle timeout
PID2_MDIV		0x3C	PWM clock divisor
PID3_CSR	0x43C40000	0x0	control/status
PID3_ISR		0x4	interrupt status
PID3_IMR		0x8	interrupt mask
PID3_IPR		0xC	interrupt pending
PID3_STRK		0x10	strike and pre-strike timer
PID3_POST		0x14	post-strike timer
PID3_POS		0x18	position
PID3_LOC		0x1C	location
PID3_KP		0x20	proportional gain
PID3_KI		0x24	integral gain
PID3_KD		0x28	derivative gain
PID3_BIAS		0x2C	duty cycle bias
PID3_PDIV		0x30	PID clock divisor
PID3_MDTY		0x34	PWM duty cycle margin
PID3_MTIM		0x38	PWM duty cycle timeout
PID3_MDIV		0x3C	PWM clock divisor
LED0_DUTY	0x43C60000	0x0	key #0 LED intensity
LED1_DUTY		0x4	key #1 LED intensity
LED2_DUTY		0x8	key #2 LED intensity
LED3_DUTY		0xC	key #3 LED intensity
LCD_DUTY		0x10	LCD backlight LED intensity
LED_PDIV		0x14	PWM frequency
LED_CTRL		0x18	control
XADC	0x43C10000		Xilinx ADC

Version register (GP_VER)
Address offset: 0x0
Reset value: 0x0000 1000 (subject to change)
This register contains the version information for the FPGA bitstream.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																MAJ[3:0]				MIN[3:0]				BLD[3:0]				Reserved		CNF[3:0]	
																ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro			ro	ro

Bits 31:16 Reserved, must be kept at reset value.
Bits 15:12 **MAJ[3:0]:** Major version number.
Reports the major version number of the FPGA bitstream.
Bits 11:8 **MIN[3:0]:** Minor version number.
Reports the minor version number of the FPGA bitstream.
Bits 7:4 **BLD[3:0]:** Build number.
Reports the build number of the FPGA bitstream.
Bits 1:0 **CNF[1:0]:** Confidence value.
Reports the confidence value of the FPGA bitstream.
00: omega (for developer use only)
01: alpha (for lab use only)
10: beta (for customer evaluation)
11: release (for production)

Figure 105. FPGA version register.

General purpose control register (GP_CTRL)
Address offset: 0x0
Reset value: 0x0000 0000
This register provides a mechanism for general purpose control of the system.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																										DAC[1:0]		FAN		DP[2:0]	
																										rw	rw	rw	rw	rw	rw

Bits 31:4 Reserved, must be kept at reset value.
Bits: 5:4 **DAC:** Dac selection.
0: left Y
1: left X
2: right Y
3: right X
Bit 3 **FAN:** Fan control.
0: off
1: on
Bits 2:0 **DP:** Debug port select.
0: none
1: PID 0
2: PID 1
3: PID 2
4: PID 3
5: keyboard module
6: LCD module
7: misc

Figure 106. FPGA general purpose control register.

PID control and status register (PIDn_CSR)
Address offset: 0x00
Reset value: 0x0000 0000
This register provides a mechanism to control the state and read the status of the PID subsystem.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
POS[15:0]																Reserved													RST	DIS
ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro														rw	rw

Bits 31:16 **POS:** Position value
Bits 15:3 Reserved, must be kept at reset value.
Bit 2 **RST:** Reset
0: normal
1: reset
Bit 1 **DIS:** Disable
0: enable
1: disable
Bit 0 **QCLR:** Quadrature decoder clear
0: normal
1: clear

Figure 107. FPGA PID control and status regsiter.

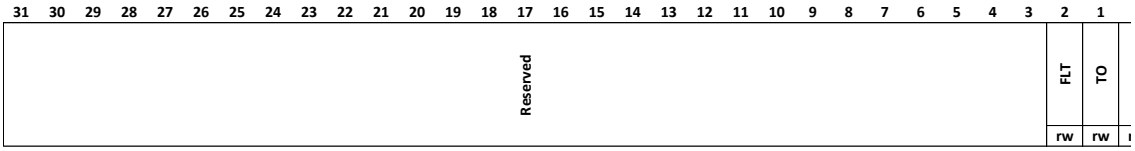
PID interrupt status register (PIDn_ISR)
Address offset: 0x04
Reset value: 0x0000 0000
This register provides a mechanism to read the interrupt status of the PID subsystem.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																													FLT	TO	SD
																													ro	ro	ro

Bits 31:3 Reserved, must be kept at reset value.
Bit 2 **FLT:** Fault
A fault event occurred.
Bit 1 **TO:** Timeout
A timeout event occurred.
Bit 0 **SD:** Strike detection
A strike detection event occurred.

Figure 108. FPGA PID interrupt status register.

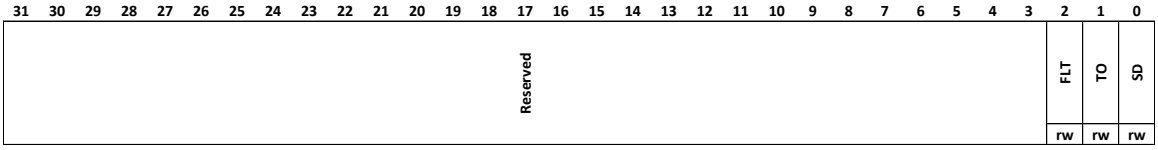
PID interrupt mask register (PIDn_IMR)
Address offset: 0x08
Reset value: 0x0000 0000
This register provides a mechanism to set the interrupt mask of the PID subsystem.



Bits 31:3 Reserved, must be kept at reset value.
Bit 2 **FLT:** Fault
0: mask interrupt
1: unmask interrupt
Bit 1 **TO:** Timeout
0: mask interrupt
1: unmask interrupt
Bit 0 **SD:** Strike detection
0: mask interrupt
1: unmask interrupt

Figure 109. FPGA PID interrupt mask register.

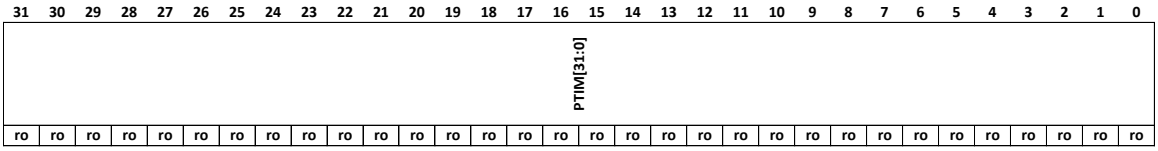
PID interrupt pending register (PIDn_IPR)
Address offset: 0x0C
Reset value: 0x0000 0000
This register provides a mechanism to read and dismiss interrupts of the PID subsystem. Write a '1' to dismiss a pending interrupt event.



Bits 31:3 Reserved, must be kept at reset value.
Bit 2 **FLT:** Fault
0: no interrupt pending
1: interrupt pending
Bit 1 **TO:** Timeout
0: no interrupt pending
1: interrupt pending
Bit 0 **SD:** Strike detection
0: no interrupt pending
1: interrupt pending

Figure 110. FPGA PID interrupt pending register.

PID Strike (PIDn_STRK)
Address offset: 0x10
Reset value: 0x0000 0000
This register provides a mechanism to measure the pre-strike time.



Bits 31:0 **PTIM:** Pre-strike timer.
Pre-strike time in units of 10ns.

Figure 111. FPGA PID strike register.

PID Strike (PIDn_STRK)
Address offset: 0x10
Reset value: 0x0000 0000
This register provides a mechanism to issue a strike.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			TIM[9:0]										DIR	VEL[7:0]								DET[9:0]									
			wo	wo	wo	wo	wo	wo	wo	wo	wo	wo	wo	wo	wo	wo	wo	wo	wo	wo	wo	wo	wo	wo	wo	wo	wo	wo	wo	wo	wo	wo

Bits 31:29 Reserved, must be kept at reset value.
Bits 28:19 **TIM:** Timeout
Timeout in milliseconds.
Bits 18 **DIR:** Direction
Strike direction.
Bits 17:10 **VEL:** Velocity
Strike velocity.
Bits 9:0 **DET:** Detection Threshold
Impact detection location threshold.

Figure 112. PID strike register.

PID post-strike register (PIDn_POST)
Address offset: 0x14
Reset value: 0x0000 0000
This register provides a mechanism to measure the post-strike time.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OTIM[31:0]																															
ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro	ro

Bits 31:0 **OTIM:** Post-strike timer.
Post-strike time in units of 10ns.

Figure 113. PID post-strike register.

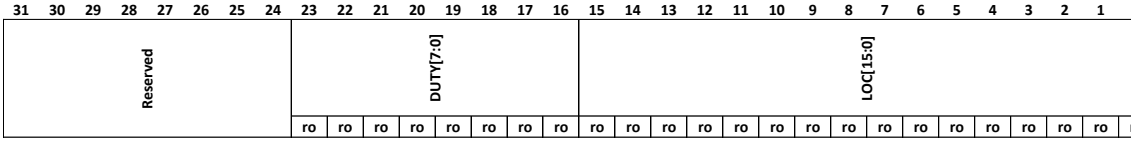
PID position register (PIDn_POS)
Address offset: 0x18
Reset value: 0x0000 0000
This register contains the target setpoint position.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SLEW[15:0]																POS[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 15:0 **SLEW[15:0]:** Slew limit.
Target rate of change limit in units of microseconds.
Bits 15:0 **POS[15:0]:** Signed value.
Target setpoint position.

Figure 114. PID position regsiter.

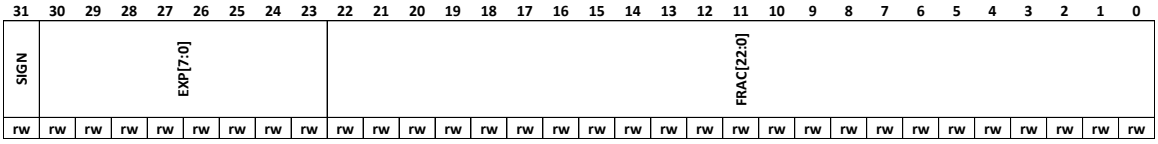
PID location register (PIDn_LOC)
Address offset: 0x1C
Reset value: 0x0000 0000
This register contains the current duty cycle and location.



Bits 31:24 Reserved, must be kept at reset value.
Bits 23:16 **DUTY[7:0]:** Unsigned value.
Current duty cycle value.
Bits 15:0 **LOC[15:0]:** Signed value.
Current location.

Figure 115. PID location register.

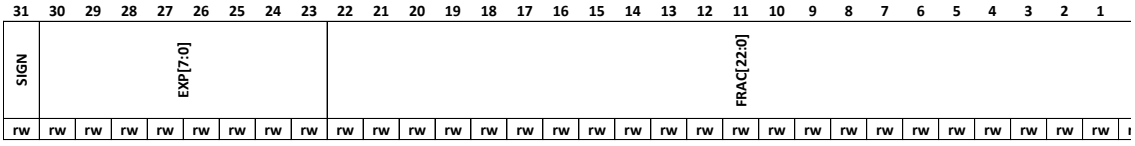
PID proportional gain register (PIDn_KP)
Address offset: 0x20
Reset value: 0x0000 0000
This register contains the proportional gain as a IEEE 754 32-bit floating point value.



Bit 31 **SIGN:** Sign indicator.
Sign value.
Bits 30:23 **EXP[7:0]:** Unsigned value.
Exponent value.
Bits 22:0 **FRAC[22:0]:** Unsigned value.
Fractional value.

Figure 116.PID proportional gain register.

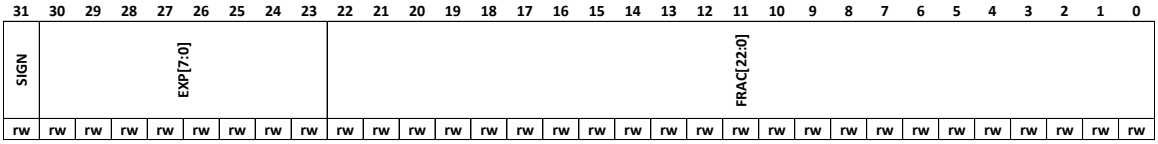
PID integral gain register (PIDn_KI)
Address offset: 0x24
Reset value: 0x0000 0000
This register contains the integral gain as a IEEE 754 32-bit floating point value.



Bit 31 **SIGN:** Sign indicator.
Sign value.
Bits 30:23 **EXP[7:0]:** Unsigned value.
Exponent value.
Bits 22:0 **FRAC[22:0]:** Unsigned value.
Fractional value.

Figure 117. PID integral gain register.

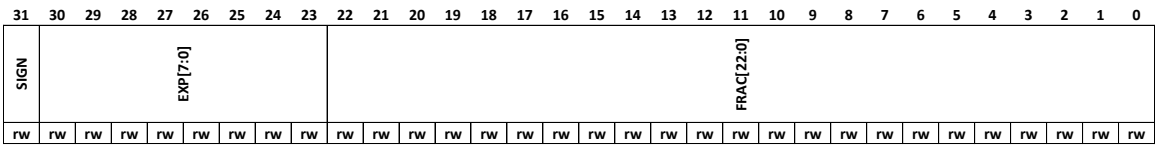
PID proportional gain register (PIDn_KD)
Address offset: 0x28
Reset value: 0x0000 0000
This register contains the derivative gain as a IEEE 754 32-bit floating point value.



Bit 31 **SIGN:** Sign indicator.
Sign value.
Bits 30:23 **EXP[7:0]:** Unsigned value.
Exponent value.
Bits 22:0 **FRAC[22:0]:** Unsigned value.
Fractional value.

Figure 118. PID proportional gains register.

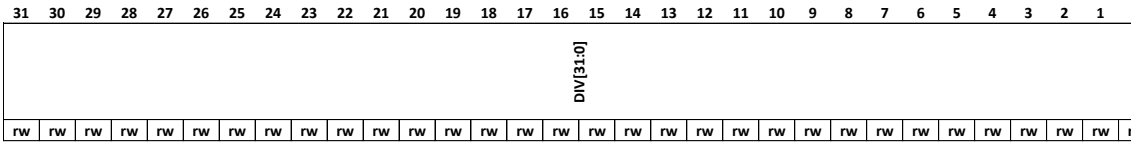
PID bias register (PIDn_BIAS)
Address offset: 0x2C
Reset value: 0x0000 0000
This register contains the bias as a IEEE 754 32-bit floating point value.



Bit 31 **SIGN:** Sign indicator.
Sign value.
Bits 30:23 **EXP[7:0]:** Unsigned value.
Exponent value.
Bits 22:0 **FRAC[22:0]:** Unsigned value.
Fractional value.

Figure 119. PID bias register.

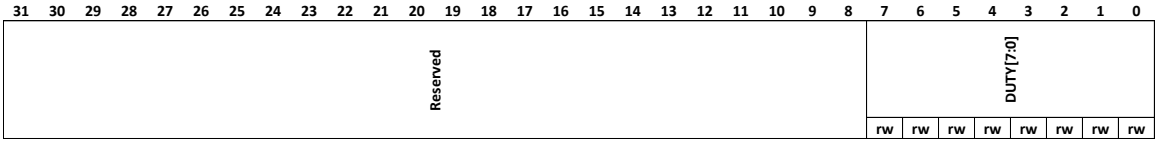
PID clock divisor (PIDn_PDIV)
Address offset: 0x30
Reset value: 0x0001 86A0
This register provides a mechanism to set the PID clock divisor.



Bits 31:0 **DIV:** Divisor.
100MHz clock divisor value (default = 1KHz).

Figure 120. PID clock divisor register.

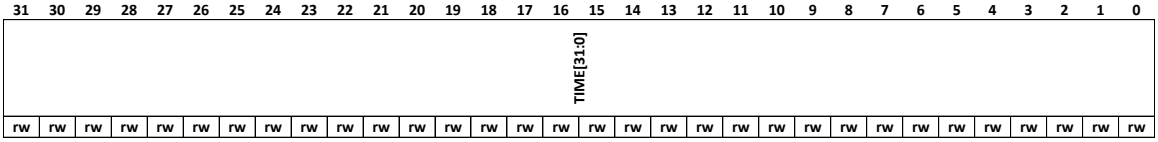
PWM duty cycle margin (PIDn_MDTY)
Address offset: 0x34
Reset value: 0x0000 0019
This register provides a mechanism to set the PWM duty cycle margin.



Bits 31:8 Reserved, must be kept at reset value.
Bits 31:0 **DUTY:** Duty cycle margin threshold.
Fault detection threshold (default = 10%).

Figure 121. PWM duty cycle margin register.

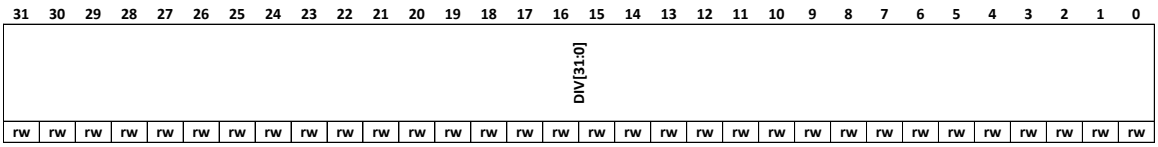
PWM duty cycle timeout (PIDn_MTIM)
Address offset: 0x38
Reset value: 0x0098 9680
This register provides a mechanism to set the PWM duty cycle timeout.



Bits 31:0 **TIME:** Duty cycle timeout.
Fault detection timeout threshold (default = 100ms).

Figure 122. PWM duty cycle timeout register.

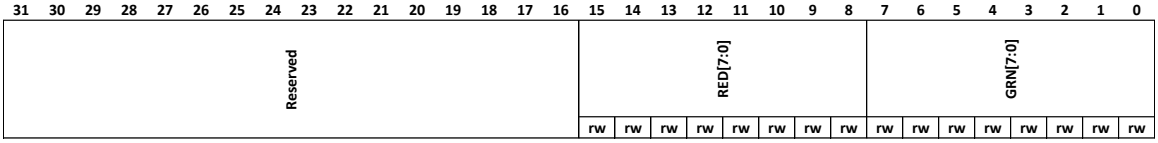
PWM clock divisor (PIDn_MDIV)
Address offset: 0x3C
Reset value: 0x0000 1388
This register provides a mechanism to set the PWM clock divisor.



Bits 31:0 **DIV:** Divisor.
100MHz clock divisor value (default = 20KHz).

Figure 123. PWM clock divisor register.

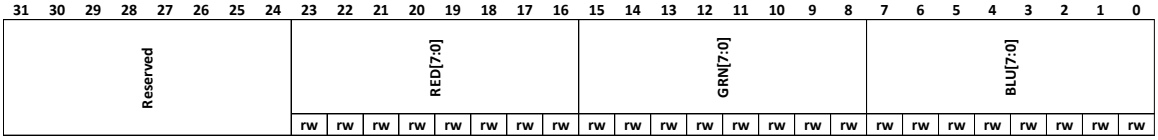
Key LED intensity (LEDn_DUTY)
Address offset: 0x00, 0x04, 0x08, 0x0C
Reset value: 0x0000 0000
This register provides a mechanism to set the keyboard LED intensity.



Bits 31:16 Reserved, must be kept at reset value.
Bits 15:8 **RED:** Red duty cycle.
Intensity, 0 = 0%, 255 = 100%.
Bits 7:0 **GRN:** Green duty cycle.
Intensity, 0 = 0%, 255 = 100%.

Figure 124. Key LWD intensity register.

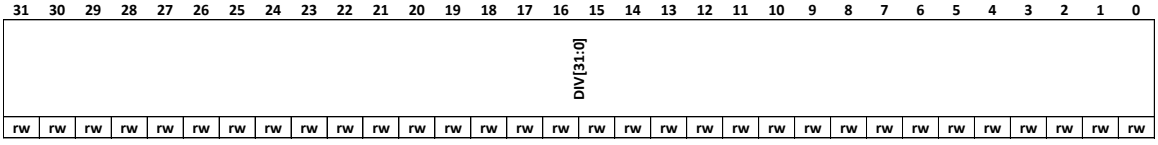
LCD backlight intensity (LCD_DUTY)
Address offset: 0x10
Reset value: 0x0000 0000
This register provides a mechanism to set the keyboard LED intensity.



Bits 31:16 Reserved, must be kept at reset value.
Bits 23:16 **RED:** Red duty cycle.
Intensity, 0 = 0%, 255 = 100%.
Bits 15:8 **GRN:** Green duty cycle.
Intensity, 0 = 0%, 255 = 100%.
Bits 7:0 **BLU:** Blue duty cycle.
Intensity, 0 = 0%, 255 = 100%.

Figure 125. LCD backlight intensity register.

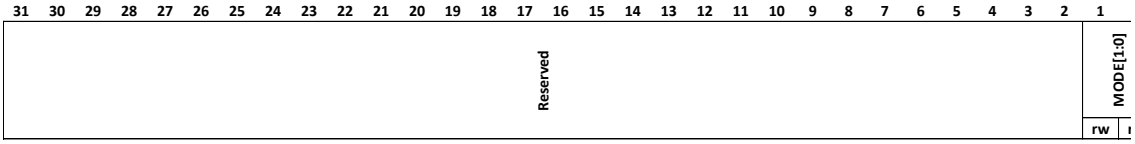
LED PWM divisor (LED_PDIV)
Address offset: 0x14
Reset value: 0x000F 4240
This register provides a mechanism to set the LED PWM clock divisor.



Bits 31:0 **DIV:** Divisor.
100MHz clock divisor value (default = 100Hz).

Figure 126. LED PWM divisor register.

LED control (LED_CTRL)
Address offset: 0x18
Reset value: 0x0000 0000
This register provides a mechanism to set the LED control register.



Bits 31:2 Reserved, must be kept at reset value.
Bits 1:0 **MODE:** Mode.
00: manual
01: keyboard
02: location
03: duty

Figure 127. LED control register.

Appendix I

Curriculum Vitae

CONTACT

Robert M. Van Rooyen
rvanrooy@uvic.ca
rob@summitscientificinc.com

RESEARCH INTERESTS

Human Percussion Performance Analysis, Robotics, Embedded Systems, Video Motion Capture, Voice Coil Actuators, Haptic Feedback, Digital Signal Processing, Generative Models, Machine Learning

EDUCATION

- **Ph.D. in Computer Science**, University of Victoria, 2013-2018
- **Certificate Program (Master Theory, Harmony, & Ear Training)**, Berklee College of Music, 2007-2009
- **MS in Computer Science**, California State University Chico, 1995-2000
- **Vocational (Recording Arts)**, American River College, 2000
- **BS in Computer Engineering**, California State University Sacramento, 1991-1993
- **General Education, Mathematics, Physics, Chemistry**, Los Medanos College, 1990-1991
- **Certificate Program (Industrial Electronic Technician)**, Diablo Valley College, 1982-1985

RECENT COURSES

- **CSC 699 PhD Dissertation**: May 2015 – Dec 2018
- **CSC 693 PhD Candidacy**: Sep 2013 - Apr 2015
- **CSC 561 Multimedia Systems**: Jan - Apr 2015
- **CSC 575 Music Information Retrieval Techniques**: Jan - Apr 2014
- **CSC 406B Sound Recording Seminar**: Jan - Apr 2014
- **CSC 546 Operations Research II**: Sep - Dec 2013
- **CSC 595 Research Skills**: Sep - Dec 2013

PUBLICATIONS (PEER REVIEWED)

- **Mechatronic drummer: a new approach to percussion:** submission pending, IEEE Transactions in Robotics.
- **Acquisition and modelling of performance-specific percussion motion:** submitted September 29, 2018, Journal of New Music Research.
- **Voice Coil Actuators for Percussion Robotics**: May 16, 2017, New Interfaces for Musical Expression.
- **Drum Performance Motion Analysis**: July 14, 2016, New Interfaces for Musical Expression.
- **Pragmatic Drum Motion Capture System**: April 17, 2015, New Interfaces for Musical Expression.
- **Snare Drum Motion Capture Dataset**: April 17, 2015, New Interfaces for Musical Expression.

PUBLICATIONS (REPORTS/WHITEPAPERS)

- **Securing Consumable Components in an Embedded System Desig**n: October 19, 2015, rtcmagazine.com.
- **Verbrauchsmittel für medizinische Geräte elecktronisch schützen**: October 1, 2015, all-electronics.de.
- **Percussion Detection in Polyphonic Music: Final Report**, April 30, 2014, University of Victoria.
- **Percussion Detection in Polyphonic Music: Progress Report**, March 16, 2014, University of Victoria.
- **Percussion Detection in Polyphonic Music: Project Design Specification**, February 3, 2014, University of Victoria.
- **Reproducing Musical Performances: An exploration of automated transcription and robotics in the context of percussion instruments**, November 15, 2013, University of Victoria.
- **Reproducing Musical Performances**: Research Proposal, November 1, 2013, University of Victoria.
- Robert Van Rooyen, **Retrofitting Embedded Systems to Enhance Security**, January 5, 2013, Barr Group article.
- Robert Van Rooyen, **How to Secure Consumable Components of Medical Devices and Other Embedded Systems**, March 3, 2013, Barr Group article.

PATENTS

- Van Rooyen, Robert Martinez. Tzanetakis, George. 2018. **Robotic Drummer.** U.S. Patent 2018/0326588 A1 Pending November 2018. University of Victoria.
- Rollins, Richard Jensen. Tupin, Paul Joe Jr. Van Rooyen, Robert Martinez. 2010. **System and method for extracting physiological data using ultra-wideband radar and improved signal processing techniques**. U.S. Patent 7,725,150 filed June 4, 2003, issued May 25, 2010. Life Wave, Inc.
- Van Rooyen, Robert M. 2010. **System for multi-profile boot selection of an embedded device**. U.S. Patent 7,676,671 filed October 31, 2006, issued March 9, 2010. Hewlett Packard Development Company, L.P.
- Van Rooyen, Robert Martinez. Crawford, Kenneth Edward. 2003. **Method and system for failsafe recovery and upgrade of an embedded operating system**. U.S. Patent 6,591,376 filed March 2, 2000, issued July 8, 2003. Hewlett Packard Development Company, L.P.

CERTIFICATIONS

- **Wind River Systems (VxWorks BSP and Device Driver Workshop)**, Alameda, CA
- **Wind River Systems (VxWorks Operating System Workshop)**, Alameda, CA
- **Green Hills Software, Inc. (Integrity BSP and Device Driver Training)**, Roseville, CA
- **Green Hills Software, Inc. (Integrity Operating System Training)**, Roseville, CA
- **Netrino (Embedded Software Boot Camp)**, Elkridge, MD
- **Barr Group (Embedded Security Boot Camp),** Germantown, MD

CORE SKILLS

- **Embedded System Architecture, Object Oriented Analysis & Design, Multithreading, Digital Signal Processing, Music Technology, Video/Audio, Multimedia, Medical Devices, Robotics, Networking, Security, and Industrial Control**
- **Researcher, Technical Leadership, and Project Management**
- **Academic Writing, White Papers, and Engineering Documentation**
- **Audio Engineering, Video Post-Production, and Content Creation**
- **Programming languages**: C++/Java, C, Assembly, Verilog, XML, and Scripting

- **Hardware Design**: microprocessors, FPGAs, peripherals, analog circuits, schematic capture, BOMs, SPICE, Simulation, and PCB layout
- **Hardware Platforms**: ARM, PowerPC, MIPS, x86, TI/Analog Devices DSPs, Xilinx FPGAs, Xilinx Zynq SOC, and numerous Microcontrollers (8/16/32-bit)
- **Hardware Interfaces**: USB, SPI, I2C, 1-Wire, CAN, SPORT, FSMC, PCI, PCI Express, RS232/422, ATA/IDE/SCSI, Ethernet, and others
- **Diagnostic/Development Tools**: Oscilloscope, Spectrum Analyzer, Digital Multi-meter, Logic Analyzer, Network Protocol Analyzer, USB Analyzer, JTAG Probes, Bench Power Supply, and SMD rework
- **OS**: Windows, Linux, Embedded Linux, and other Unix variants
- **RTOS**: Integrity, VxWorks, uC/OS-II, FreeRTOS, TI-RTOS, VisualDSP++, and others
- **Protocols**: TCP/IP, UDP, Multicast, SNMP, Telnet, FTP, NFS, DHCP, DNS, SLP, iSCSI, and others
- **Build Environments**: Linux, Windows, Greenhills Multi-IDE, WindRiver VxWorks, IAR, MPLAB-X, GNU, TI Code Composer, Visual DSP++, MS Visual Studio, MATLAB/Simulink, GIT, Subversion, and others
- **Mechanical Design**: enclosures, fixtures, motion control, and assemblies using Solidworks, and OnShape

EXPERIENCE

President, Senior Engineer/Scientist, Summit Scientific, Inc., Seattle, WA, 2008-Present

Research, system architecture, software/firmware development, electronic design, mechanical engineering, and project management

Sample Projects

- **Virus Detection Lab System**: Architecture and implementation of image processing FPGA for rapid data analysis and system monitoring. Tech: Xilinx Artix-7, FMC, 12-bit Camera Interface, DRAM. Tools: Xilinx Vivado 2017.3, Xilinx Vivado HLS 2017.3, oscilloscope, logic analyzer, Bench Power Supply.
- **Lithotripsy System**: Architecture and implementation of safety monitor FPGA to prevent injury to patients during treatment by actively monitoring and constraining frequency, temperature, power, and digital conditions. Tech: Xilinx Spartan-6, SPI, PWM. Tools: Xilinx PlanAhead, oscilloscope, DVM, logic analyzer, Bench Power Supply.
- **NRAM Test Fixture**: Architecture and implementation of parameter driven data integrity test fixture for leading-edge NRAM device in Low Earth Orbit. Tech: Xilinx Spartan-6 FPGA, FSMC, DCM, ODDR2, FIFO. Tools: Xilinx PlanAhead, oscilloscope, logic analyzer, Bench Power Supply.
- **Robotic Drummer**: research/software/firmware/hardware/mechanical architecture and design of percussion robot that closely approximates human motion. Tech: Xilinx Zynq-7000 FPGA, ARM, AXI, XADC, LCD, MIDI, H-Bridge, amplifiers, MicroZed SBC, buck converter, PID controller, quadrature decoder, motion control algorithms, PWM generator, SPI, JTAG. Tools: Tracker, Xilinx Vivado, Xilinx SDK, PetaLinux, JUCE framework app, schematic capture, PCB layout, oscilloscope, DVM, logic analyzer, JTAG probe, bench power supply, SMD rework.
- **Brain Infusion Pump**: Architecture and implementation of safety monitor FPGA to prevent accidental injury to patients during treatment by actively constraining fluid pump and valve activation though continuous synchronous monitoring of real-time analog pressure and digital conditions. Tech: Xilinx Spartan-6, A/D converters. Tools: Xilinx PlanAhead, oscilloscope, DVM, logic analyzer, Bench Power Supply.
- **Atomic Clock**: Architecture and implementation of multiple FPGAs providing temperature, voltage, and current servo control of complex laser subsystems that included hardware reviews, board bring-up,

engineering documentation, vendor selection/management. Tech: Xilinx Kintex-7 and Spartan-6, AXI, A/D converters, DACs, Thermistor, Peltier Cooling. Tools: Xilinx Vivado, Xilinx PlanAhead, oscilloscope, DVM, logic analyzer, Bench Power Supply.

- **Quantum Random Number Generator**: Architecture and implementation of PCI Express board logic that included hardware reviews, board bring-up, engineering documentation, vendor selection/management, the creation and integration of FPGA IP cores, embedded firmware, host device driver, and host application. The development work also included research and design of algorithms to manage data integrity in real time. Tech: Xilinx Zynq-7000 FPGA, ARM, AXI, PCIe, DRAM, PLL, SHA-512, A/D converter, DAC, Thermistor, Peltier Cooling. Tools: Vivado, Xilinx SDK, CentOS (Linux), oscilloscope, DVM, logic analyzer, Bench Power Supply.
- **MIDI Interface**: Conception, Architecture, and implementation of MIDI (Musical Instrument Digital Interface) controller for an Electronic Accordion. The feature set included a high-level command processor over a serial JTAG port, A/D converter for analog slider level monitoring, timer interrupt driven keyboard scanning, MIDI command generation, and MIDI serial communications. Tech: Dynamic C, Rabbit 2000 Microcontroller, A/D converter, opamp. Tools: ExpressPCB, JTAG controller, digital volt meter, oscilloscope, logic analyzer, serial data simulation breadboard circuit.
- **Medical device prototype**: Requirements analysis, electronic design, board layout, documentation. Tech: ARM Processor, LED driver, opamps, analog filters. Tools: DipTrace.
- **Li-Ion Charge/Debug Fixture**: Requirements analysis, electronic design, board layout, mechanical design, and documentation. Tech: Li-Ion IC, JFET, USB/External power, pogo pins. Tools: Alibre, ExpressPCB, Spice, DVM

Senior Engineering Consultant, Barr Group, Germantown, MD, 2012-2014

Firmware development, electronic design, system architecture, technical leadership, project management, intellectual property research, and embedded system training content creation

Sample Projects

- **Closed Loop Control Webinar**: Conception, content creation, and video production.
- **Mobile Prescription Drug Distribution System**: Multiple embedded systems code refactoring to established coding standard compliance and industry best practices. Tech: PIC, MPLAB-X.
- **Embedded System Security Audit**: Detailed security analysis and report publication of automotive industry product. Tech: ARM Cortex M4, Atmel ATSHA204, C, Public Key Cryptography, Authentication, Encryption, Signing Authority, Secure Update, JTAG, Boot Loader.
- **Industrial Automation Controller**: Cross compile, test, and configure open source VPN server, and Javascript JIT compiler for embedded Linux ARM platform. Tech: Linux, pptpd-1.3.4, Node v0.10.28, Tools: CentOS 5.0 Linux Distribution, GNU 4.3 ARM Cross Compiler, Python 2.6, Make, ssh, ftp, Artila PAC-4010 Controller.
- **Invasive Surgical Medical Device**: Architecture and implementation of 3rd generation disposable/reusable medical device for the surgical treatment of Gastroesophageal reflux disease. Tech: C, PIC18, AD7193, SPI, 1-Wire, EEPROM, Tools: MPLAB IDE v8.50, DipTrace, Microchip Pickit 3 JTAG probe, Oscilloscope, Logic Analyzer, Fluke High-precision Digital Multimeter, NIST Traceable RTD Temperature Probe, Bench Power Supply, Scientific Hotplate, Subversion.
- **Training Content and Exercises for Advanced Embedded Programming Course**: Conception and implementation of topical content and hands-on exercises for embedded systems course. Tech: C, Assembly, uC/OS-II, FreeRTOS, TI-RTOS, ARM Cortex-M on TI Single Board Computer, Tools: IAR Embedded Workbench for ARM, TI Code Composer, Logic Analyzer, Subversion.
- **Training Content and Exercises for Embedded Security Course**: Conception and implementation of topical content and hands-on exercises for embedded systems training course. Tech: C, Elliptical Curve Cryptography, PolarSSL, uC/OS-II, ARM Cortex-M on TI Single Board Computer, Tools: IAR Embedded Workbench for ARM, Subversion.

Software Design Engineer, Hewlett Packard, Roseville, CA, 2004-2012

Firmware development, embedded system bring-up, system architecture, technical leadership, and mentoring

Sample Projects

- **Operating System Abstraction Layer and Device Driver Framework**: Conception, architecture, and implementation of standardized operating system abstraction layer for enterprise networking equipment. Tech: C, Assembly, Integrity RTOS, Redhat Linux, Microsoft Windows, PowerPC, ARM, MIPS, x86, Tools: Greenhills Multi IDE, GNU Toolchain, Microsoft Visual Studio, Logic Analyzer, Network Analyzer, Greenhills JTAG Probe, GIT.
- **Triple Speed Ethernet Controller Driver**: Architecture and implementation of proprietary network stack triple speed Ethernet device driver for enterprise networking equipment. Tech: C, Assembly, PowerPC, TSEC controller, Tools: WindRiver Workbench, Greenhills Multi IDE, Greenhills JTAG Probe, Network Protocol Analyzer, ClearCase, GIT.
- **Network Switch Boot ROMs & BSPs**: Architecture and implementation of platform bootROMs and BSPs. Tech: C, Assembly, PowerPC, ARM, MIPS, WindRiver VxWorks RTOS, Greenhills Integrity RTOS, file system, SPI, I2C, FPGA, ATA, Tools: WindRiver Workbench, Greenhills Multi IDE, WindRiver JTAG Probe, Greenhills JTAG Probe, ClearCase, GIT.
- **IDE/ATA Flash Storage Driver**: Architecture and implementation of IDE Flash storage device using ATA command set on 16-bit local bus. Tech: C, Assembly, ATA, IDE, PowerPC, WindRiver VxWorks RTOS, Greenhills Integrity RTOS, file system, Tools: WindRiver Workbench, Greenhills Multi IDE, WindRiver JTAG Probe, Greenhills JTAG Probe, Logic Analyzer, Oscilloscope, ClearCase, GIT.
- **USB Mass Storage Driver**: Architecture and implementation of hot-swap USB mass storage driver. Tech: C, FPGA, PowerPC, ARM, WindRiver VxWorks RTOS, Greenhills Integrity RTOS, file system, Tools: WindRiver Workbench, Greenhills Multi IDE, WindRiver JTAG Probe, Greenhills JTAG Probe, USB Protocol Analyzer, ClearCase, GIT.

Senior Software and Hardware Engineer/Architect, Life Wave, Inc., Roseville, CA, 2002-2004

Medical device research, algorithm development, firmware development, electronic design, and system architecture

Primary Project

- **Cardio Pulmonary Medical Device**: Conception, design, and implementation of digital signal processing algorithm and support framework for extracting cardiopulmonary data from ultra-wideband radar returns in a non-invasive handheld device. The feature set included an object oriented RTOS framework, DSP computational framework abstraction, multi-phase DSP algorithm (transpose, convolution, normalization, FFT), adaptive “bin” chasing on returns, A/D device driver, DMA device driver, USB class protocol driver, LCD driver, serial driver, logging task, and runtime configuration database. Co-design and implementation of UWB RF transceiver, CPLD, A/D, filter, and embedded system electronics. This project led to US patent #7,725,150. Tech: C++, C, Assembly, Blackfin DSP, TI DSP, Sport Port, RS232, USB, VisualDSP++ RTOS, SYS/BIOS, Tools: VisualDSP++, TI Code Composer, Oscilloscope, Logic Analyzer, Spectrum Analyzer, USB protocol analyzer, Matlab, Labview, Perforce, OrCAD, Xilinx Webpack.

Software Design Engineer/Architect, Seven Systems Technologies, Inc., Rocklin, CA, 2001

Research, technical leadership, firmware development, system architecture, mentoring, and vendor management

Sample Projects

- **Optical Networking Concentrator**: Conception, design, and implementation of enterprise networking switch. Tech: C++, C, Assembly, x86, VxWorks RTOS, Tools: Wind River Tornado Development Environment.
- **Network Storage Device**: Conception, design, and implementation of NAS device. Tech: C++, C, x86, Linux, Tools: GNU.

Audio Engineer and Small Business Owner, Your Tempo, Antelope, CA, 1999-2003

Audio recording/mixing/mastering, performance, marketing, equipment acquisition/maintenance, studio construction, talent interaction, and business administration

Sample Projects

- **Demo tracks**: multiple recording sessions with country artist and back band that included lead vocals, backup vocals, lead guitar, rhythm guitar, bass, piano/keyboard, and drums. The engineering work included timing corrections, vocal pitch tuning, sound reinforcement, additional performance tracks, and automation. Tech: Logic Audio, Ramsa DA7 Digital Mixer, Sure SM57/57A/94 dynamic microphones, Rode NT2 condenser microphone, ElectroVoice RE-20 dynamic microphone.
- **Band rehearsal/gigs**: performance, audio engineering/recording, arrangement, music selection, and talent direction. Tech: Logic Audio, Ramsa DA7 Digital Mixer, Alesis QS8.1 Keyboard, Fender Jazz Bass, Pearl Drum Kit, amplifiers, speakers, and cabling.
- **Training video**: concept and content creation for audio engineering course. Tech: Adobe Premiere, Logic Audio, PowerPoint, and Excel.
- **County library presentation**: content creation and audio/video editing. Tech: Adobe Premiere, Logic Audio, and PowerPoint.

Senior Software Engineer/Architect, Hewlett Packard, Roseville, CA, 1995-2001

Firmware development, embedded system bring-up, system architecture, technical leadership, and mentoring

Sample Projects

- **Network appliance**: Conception, design, and implementation of enterprise network printing appliance. Tech: C++, C, Assembly, x86, Linux, Tools: GNU.
- **Network printing installation/configuration/management applications**: Conception, design, and implementation of enterprise network printer applications. Tech: C++, C, Assembly, x86, Windows, Tools: Microsoft Visual Studio C/C++, network sniffer
- **Network device drivers**: Design and implementation of network device drivers. Tech: C++, C, Assembly, x86, Windows, Tools: Microsoft Visual Studio C/C++, network sniffer
- **Network reliability tools**: Conception, design, and implementation of network test application. Tech: C++, C, Assembly, x86, Windows, Tools: Microsoft Visual Studio C/C++, network sniffer

Software Engineer, Optivision, Davis, CA, 1994-1995

Software development, hardware bring-up, system architecture, source control, and documentation

Sample Projects

- **MPEG Digital Video Encoder/Decoder**: Design and implementation of non-linear editor and playback software. Tech: MPEG1, C++, C, Assembly, x86, Windows, Tools: Microsoft Visual Studio C/C++
- **Video Transport Control Driver**: Design and implementation of Sony standard serial protocol. Tech: C, x86, Windows, Tools: Microsoft Visual Studio C/C++
- **Application Installation Software**: Design and implementation of Windows application installation programs. Tech: C, Windows, Tools: InstallShield
- **High Speed Optical Network Controller/Driver Software**: Conception, design, and implementation of windows software that controlled a proprietary optical network communication system. Tech: C, Windows, Tools: Microsoft Visual Studio C/C++.

Computer Engineering Consultant, Leiden Logic, Sacramento, CA, 1991-1995

Software and hardware development, field engineering, and documentation

Sample Projects

- **System software**: Design and implementation of multiple applications and utilities. Tech: C++, C, Assembly, 68K, x86, BSD Unix, Tools: BSD Toolchain, Borland Turbo C
- **Voice Recognition Prototype**: Conception, design, and implementation of speech recognition system for Windows. Tech: C, x86, Windows, A/D Converter, PLD, Tools: Microsoft Visual Studio C/C++
- **Financial Processing System Maintenance**: Field maintenance of document transport control systems, BSD Unix servers, optical character recognition system, barcode reader, and CCD camera.

Industrial Electronic Technician, TRW Financial Systems, Berkeley, CA, 1985-1991

Analog/digital design, software/firmware development, mechanical design and fabrication

Sample Projects

- **System diagnostics**: Conception, design and implementation of multiple applications and utilities. Tech: C, Assembly, 68K, x86, BSD Unix, MSDOS, Tools: BSD Toolchain, Borland Turbo C
- **CCD Camera**: Design and implementation of analog/digital front-end, and image processing board. Tech: CCD chips, OPAMPs, analog delay lines, TTL digital logic, PLDs, Tools: Oscilloscope, logic analyzer, DVM, spectrum analyzer, PCB layout
- **Barcode Reader**: Incremental design improvements and standalone MCU decoder. Tech: OPAMPs, TTL digital logic, PLDs, Tools: Oscilloscope, logic analyzer, DVM, 8048 assembler, milling machine, lathe

ORGANIZATIONS

Victoria Innovation, Advanced Technology and Entrepreneurship Council

National Academy of Recording Arts and Sciences

Audio Engineering Society

Association for Computing Machinery

HONORS & AWARDS

2018 University of Victoria Graduate Award

2018 Margret Guthman Musical Instrument Competition Technical Achievement (Georgia Tech)

R&D 100 Award (individual contributor to Los Alamos National Laboratory development team)

Wighton Engineering Development Fund (University of Victoria)

2016 University of Victoria Graduate Award

Tau Beta Pi National Engineering Honor Society

Golden Key National Academic Honor Society

LANGUAGES

English (fluent)

Dutch (limited working proficiency)