California Institute of the Arts

Effusions of Fancy:

An Exploration into Improvisatory Live Computer Performance

by

David Howe

A thesis submitted in partial fulfillment for
the degree of Master of Fine Arts

in the
Herb Alpert School of Music
Music Technology: Interaction, Intelligence & Design

Supervisory Committee

Dr. Owen S. Vallis

Mentor

Dr. Ajay Kapur

Committee

Dr. Jordan Hochenbaum

Committee

Abstract

Improvisation is one of the innate, fundamental behaviors exhibited by humankind. The art of spontaneity is our instinctual method of "staying in the present moment," as opposed to "dwelling in the past." In logically assuming that one must experience the present before they can then dwell in that past, it stands that extemporaneous musical creation must predate all other musical compositional/performance techniques, rendering improvisation likely the oldest musical approach known to man. The affinity for improvisation in music has never ceased its considerable influence over the development of musical forms, systems, and instruments to date. This thesis aims to explore improvisation in music as it stands currently, through a context of live computer music performance. How has improvisation influenced modern electronic musical paradigms? What effusions of fancy are afforded by computer-driven performance that was previously otherwise impractical, if not impossible, or even unimaginable? Moreover, how can computer-driven live performance potentially revolutionize such an archaic, long-held musical infatuation as improvisation?

Computers are sought after for their superhuman abilities, namely, precision speed and multitasking. They have essentially incited a musical modus operandi of complete user specification and separation of inputs (instruments/controllers), outputs (type of sound), and their mapping (connection between the two). Computers now allow for a performing musician to "wear many (if not all) musical hats" besides just that of "performer." With the aid of digital systems, the performer can now be the performer-composer-conductor-mixer-digital luthier-system designer-visual artist and more if he so chooses. It is the scope of this research to analyze current and newly developed software and hardware tools as they apply to improvisation in computer music performance; to determine methods for more complete spontaneous control by applying these superpowers previously unavailable without computers; to explore the boundary between maximum musical control and tasteful effective performance; finally, to wear as many hats as the average computer and (average) musician mind can tolerate, and document it, for the fate of improvisation.

Presented in this work is (1) a brief historic account of the improvisational foundations that have been laid and still adhered to in Western Music, a demonstration of the somewhat symbiotic correlation between the evolution of music technology and musical improvisation, up through a modern account of improvisation in computer-driven audio and audiovisual performance; (2) a handful of select efficient mapping strategies and advanced mapping tools for the performing electronic musician, with the intention of providing some ease and clarity to dauntingly controllable modern musical systems; (3) a couple software and hardware tools built specifically for refining and hopefully redefining improvisation as it stands in different electronic audiovisual performance contexts; (4) an analyses of said tools through live improvisational performance by the author, as well as other musicians. Technologists, musicians, and researchers have tackled these ongoing questions of improvisation periodically throughout Western Music, especially recently as we evolve through a computer-aided society. It is not the expectation of this document to have one correct answer to these improvisational queries and how they apply to the digital age, but it is the hope that researching them academically, historically, and experientially will provide insight for not only current and future electronic musicians infatuated with the effusion of fancy, but more selfishly for myself and my own future improvisational endeavors. This thesis serves to account that despite the prevalence of pre-composed, pre-sequenced, algorithmic, and other popular computer-oriented pre-programed musical processes, innovation in improvisation is still at the forefront of musical performance.

Acknowledgments

To my mentors, Ajay Kapur, and Owen Vallis, and professors, Jordan Hochenbaum, Martijn Zwartes, Trimpin, Perry Cook, John Baffa, Tom Jennings, Sara Roberts, and all the others, for continuously blowing my mind with some musical idea, concept, knowledge, performance, technology, or other musical weirdness.

To past teachers like John Schneider and Putter Smith Adam Howe that kept me musically inspired, and to electronic musical inspirations like Daedelus, Brian Crabtree, and Sergi Jordà (and far too many others to name), that have kept me music-technologically inspired.

To my classmates, who have provided me an outlet to vent when my mentors/professors overloaded me with too much work!

To my old band mates, like my brother Adam Howe, Timothy Deschaine, and Rob Petricca, Greg Ripes, and Daniel Rizik-Baer, and so many more collaborators, that have given me an outlet to express myself through music.

And ultimately to my mom and dad, Carole and Jack Howe, and my brother, Adam Howe, and all of my relatives and friends who have given me more support than I could ever have hoped for through my eccentric endeavors.

Thank You!

Table of Contents

Chapter 1 Introduction 1

Motivation 1

Computers Extending Humans 1

The Art Of Improvisation 3

Considerations For Improvisation 4

Levels Of Improvisation 4

Contexts Of Improvisation 5

Forms Of Improvisation 6

Instruments Of Improvisation 7

Improvisational Focus of This Work 7

Overview 8

Tools Of The Trade 9

Contributions 9

Outline 10

Chapter 2 A Brief History of Improvisation in Western Music Technology 13

Improvisation Pre-electronics 13

Improvisation post-electronics 18

The Hardware angle 19

The Software Angle 29

The Performer Angle 34

The Audio-Visual Angle 45

Chapter Summary 49

Chapter 3 Mapping Strategies 51

Mapability 52

Importance of Mapping 52

The State of Things 53

Considerations And Contributions 58

Considerations 59

Contributions 63

Chapter Summary 70

Chapter 4 Inputs and Outputs 73

LiveMLR 73

Motivation 74

A Bettering of Both Worlds 78

LiveMLR explained 80

Related/Future Work 83

LightBalloons 85

Motivations 86

LightBalloons Hardware 88

LightBalloons Software 89

Chapter Summary 91

Chapter 5 Hands On Research 93

Sonic Original Improvisation 93

Machine Orchestra 93

Grids Beats Groups – Roy O. Disney Hall, CalArts 95

Hamsterstyle – CalArts Coffeehouse Theatre 99

March 20th, 2014 - In Relation To The Universe, We Are All Sleeping In The Same Room, Black & White Gallery - CalArts 102

Visual Improvisation 105

July 19th-21st, 2013 – ConCon 2013: The Controller Convention, ... 105

MIA Gallery – Los Angeles 105

January 27th, 2013 – LightBalloons @ Control Freqs, Ham & Eggs Tavern, Los Angeles 106

February 20th, 2014 – Zaptra EP Release Show, Los Globos – Los Angeles 109

Chapter Summary 111

Chapter 6 Conclusion 115

Summary 115

Primary Contributions 116

Other Discussions 117

Limits 117

Final Thoughts 118

On Technology 118
On technique 120
On Modern Improvisation 121
Future Work 121

APPENDIX A LightBalloons: A Technical Glance 123
A.1 Concepts 123
A.2 Mapping Specifics 124
A.3 Hardware Revisions 125
A.3.1 Gas Leakage 125
A.3.2 Plug and Play 126
A.3.3 Brightness 126
A.4 Software Specifics 127

Appendix B LiveMLR Technical Files 131

Appendix C Some Poetry 133
C.1 Playing With JAZZ 134
C.2 Hands 135
C.3 Jazz 2 136

Bibliography 137

List of Figures

Figure 1. Thesis Map 8
Figure 2. Wegmen's definition of improvisation 14
Figure 3. Baroque-style figured bass (left) vs. Jazz-style chord chart (right); different symbology with almost identical meaning 18
Figure 4. Edison and the first phonograph 19
Figure 5. Leon Theremin (inventor), playing the Theremin 21
Figure 6. Diagram of Sensor Control for Mathews' Radio Batons 24
Figure 7. Turning point electronic instruments: MPC60 (left); Kaoss Pad (right) 26
Figure 8. Grid Controllers: Monome, Chronome, Launchpad, Push 29
Figure 9. Example of Graphical Musical Programming 30
Figure 10. Acoustic Guitar as a 5-Way Shared Instrument 36
Figure 11. League Of Automatic Composers flyer/Map of Networking 37
Figure 12. PLOrk (left); Machine Orchestra (right) 37
Figure 13. Simian Mobile Disco's extensive live rig 40
Figure 14. Tim Exile's live rig 43
Figure 15. Joshua Light Show live (left, center); materials (right) 47
Figure 16. Cyclo Live Visuals 49
Figure 17. Input and output between different families of instruments 52
Figure 18.one-to-many mapping (left); many-to-one mapping (right) (Hunt and Kirk 2000)...... 53
Figure 19. Nosaj Thing Playing an MPD 55
Figure 20. A Progression of digital controllers 58
Figure 21. Example of Hardware Radio buttons 66
Figure 22. MIDIRadio User Control Panel 67
Figure 23. MIDIChain GUI (against Ableton Live background) 69
Figure 24. Sidechaining in Ableton Live is common, easy, and intuitive 69
Figure 25. MLRV – Hypersampling Software Instrument Version of MLR 74
Figure 26. Ableton Live Clip Control Window 78
Figure 27. MLRReaktor GUI 80
Figure 28. LiveMLR GUI (right 2/3) laid over Ableton Live Session View (left 1/3) 82

Figure 29. LiveMLR Signal Routing for regular MLR button press in blue, and pattern record loop in red ... 83
Figure 30. LightBalloons: A Visual Instrument ... 85
Figure 31. ATOM (left); Particles (middle); AntiVJ (right) ... 87
Figure 32. LightBalloons Essential Parts: LEDs, Airtight tube, locking plug ... 88
Figure 33. Neopixel RGB LED (left); Adafruit's Neopixel Circuit Diagram (right) ... 89
Figure 34. LightBalloons signal flow ... 91
Figure 35. The Machine Orchestra at REDCAT Theatre, 2012 ... 94
Figure 36. Boma Happy at CalArts ... 97
Figure 37. Grids Beats Groups at CalArts 2013 ... 97
Figure 38. Flyer for Hamsterstyle pt. 0 ... 100
Figure 39. Flyer for Hamsterstyle pt. I (left); Flyer for Hamsterstyle pt. II (right) ... 102
Figure 40. Flyer for Hill Jaeger's Graduate Recital ... 103
Figure 41. Flyer for ConCon Convention 2013 ... 106
Figure 42. Flyer for Control Freqs ... 106
Figure 43. LightBalloons @ Control Freqs ... 109
Figure 44. Zaptra EP Release Show Flyer (left); Zaptra Pre-Show Balloon Hang (right) ... 110
Figure 45. Zaptra Performance with LightBalloons ... 113
Figure 46.LightBalloons Live Mapping ... 125
Figure 47.Ableton Live Plugin Window of Max4Live Serial Protocol for LightBalloons ... 128
Figure 48. Full Max4Live Serial Protocol Patch (Expanded View) ... 128

Chapter 1

Introduction

"There is nothing more risky than improvisation, but there is nothing more devastating to music's emotional message than avoidance of risk"

(Levin 2009, 147)

Improvisation within music is an act of bravery; it is a choice to boldly explore the unfamiliar, in the hopes that something really gratifyingly magical will emanate. Often it does, but there is no guarantee. Such is the same with the evolution of Music Technology, a constant inventing of something new, in the hopes that it will open a world of unexpected sonic sensations. In this way, improvisation and Music Technology are (and will probably always be) intrinsically tied. This work has set out to demonstrate that music with the aid of computers has offered a dramatic increase in control over methods/tactics/techniques/technology with which one can improvise, providing performers with a newfound freedom over an entire musical system.

Motivation

Computers Extending Humans

As our technological threshold is pushed closer toward our idea of exactitude, we continue increasingly to depend upon our tools of the age. Thus far, we have instruments that can measure almost every aspect of our physical world far more precisely than we could ever dream possible through our senses alone. We have existed for millennia without computers, and not more than a century with them, so is it not wrong to ask, what specifically can computers do that humans cannot? This question underlies any further discussion, exploration, and experimentation of why and how computers should or should not be used in musical performance. It is without question that computers have influenced, eased, and for many even given a sense of purpose to most aspects of modern everyday lives, including music. By making the previously humanly impossible possible, computers have infiltrated our way of life to the point of dependability, where it is difficult to think of what life would be like without them. Of the seemingly infinite tasks computers are capable of accomplishing, their actual super powers that separate them above mere "manual" human capability are two-fold: (1) accuracy and speed of data manipulation, which equate to tasks being executed with more precision, and at exorbitantly faster rates than humanly possible–fast enough to appear simultaneous by human standards, which in turn gives rise to another far superior power: multitasking; and (2) more memory and accurate memory accessibility than any savant or person utilizing a photographic memory or pneumonic device can produce. With these powers alone, computers (in conjunction with the mechanical devices they operate) can do almost everything better (or at least more quickly and exactly) than a human can. It is easier for a human to make a computer simulate a human, than for a human to simulate a computer. As an example, a computer can simultaneously record and play multiple channels of audio data at a rate of 192 thousand times per second, while multitasking to listen for and be processed by input data from a controller, like a mouse, while also spectrally analyzing the audio data and sending it out through a network to other computers to process. As is a common stereotype toward live computer performers, a computer even provides the means to execute what used to comprise the efforts of say a 100 piece orchestra, just by pressing "play," while concurrently navigating through email or some online social media.

Some specific useful powers for audiovisual improvisational performance that render a computer far more capable then the humans that created them are an ability to:

- Be compact and multifaceted through virtual software, without very much hardware equipment
- Control multiple instruments, voices, or arrangements at once
- Use any input source–any change in the outside world–as a control source to control any parameters
- Easily generate complex indirect mapping relationships between input source and output source that are not bound by the laws of acoustics
- Recreate, redesign, and re-customize its musical system up to any/every time it is played
- Vary in amount of precision, from undetectably fast and/or quantized, to extremely latent and/or "humanized"

- Vary in control over randomness from completely random to not random at all.[1]
- Sync any pertinent data, e.g., time signature, rhythm, pitch, etc. between devices or programs
- Allow the performer-composer to be his own digital luthier, and designer of his entire musical system

The Art Of Improvisation

What is it about tastefully executed improvisational risks that usurp a listener's attention, leaving it in an insatiable, enthusiastic state of wanting more? How has improvisation managed to permeate all genres and cultures, and all levels of musical complexity? Is it the high demonstration of artistry? Is it a sonic window into the inner workings of the human mind? Is it just a refreshing break from monotony? Answers to questions like these would require a considerable philosophical undertaking, and unfortunately are far beyond the scope of this thesis. If we can leave these questions to wonder and take for given our mysterious long held affinity for improvisation, we can at least scratch the surface by understanding what improvisation is, how and where it has influenced Western Music–genre by genre–, and ultimately, how modern computer-aided music has refined as a product of, and has influenced improvisation itself.

It is interesting that the origins of music technology and the origins of improvisation go hand in hand, as some of the first musical criteria in existence, dating back to when the earliest humans designed and shaped their surrounding materials into sound producing mechanisms, or, instruments (first with grooved scrapers and "bullroarers," and later with flutes made of animal bone) (Geiringer and Miall 1945, 37). Only after their instruments were sufficiently technologically designed, were people then capable of generating new sounding rhythms and tones, initially in somewhat spontaneously random combinations–which we would call improvisation–and eventually (after fiddling enough to fall into a worthwhile likable pattern) in organized repeatable groupings–which we would call composition. It is this perseverance through developing Music Technology of the ages that has directly fostered advancements in

[1] Randomness is only simulated in computers with algorithms, but far more accurately than can be executed by a person (oxymoronically), since true randomness is a conceptual impossibility in practice, similar to a concept like Infinity.

improvisation and composition[2], yet the drive to advance improvisation and composition also fostered said technological perseverance. All of this to note that pushing the boundaries of Music Technology, be it as a performer, composer, luthier, or otherwise, is the most appropriate realm to explore where new improvisation, and hence, new music can most easily emanate.

Considerations For Improvisation

Before any improvisational evaluation can proceed, a basis must be established; criteria must be formulated with which to reflect upon. It is my opinion that the following categories and subcategories present a loose taxonomy, or general overview of improvisation as it has stood and currently stands. The following categories serve as a foundation for which this work derives its ideas and contributions.

Levels Of Improvisation

Levels of improvisation refer to the differentiating of which musical "realm" performers are improvising within, ranging from micro to macro. Are they improvising monophonic melodies? Or are they improvising with timbre or applying effects to the instrument or sound? Or perhaps are they improvising chord progressions of different song sections? In an attempt to classify different types of control for digitally controlled live performance, Wanderley and Orio furnish a taxonomy of different levels one can operate upon, the first three of which directly apply to improvisation, digital or not. Any type of musically based improvisation can ultimately be reduced into one of these three categories (M. M. Wanderley and Orio 2002):[3]

1. Note-level control, or musical instrument manipulation (performer-instrument interaction), i.e., the real-time gestural control of sound synthesis parameters, which may affect basic sound features as pitch, loudness and timbre.
2. Sound processing control, or post-production activities, where digital audio effects or sound spatialization of a live performance are controlled in real time, typically with live-electronics.[4]

[2] The other major factor being the sociological influences on cultures and their art.
[3] These categories have been re-ordered from their original listing to show a micro to macro type of progression, whereas the original was ordered from the most traditional to the most recent in a live computer music context.
[4] Often referred to as effect-level control.

3. Score-level control, for instance, a conductor's baton used to control features to be applied to a previously defined–possibly computer generated–sequence.

These three improvisational levels are administered for evaluation throughout the course of this thesis. All three levels of improvisational control have existed throughout Western Music's past, so although they are described to pertain to digital instruments, note-level can for example just as easily pertain to a violin playing a melody, and sound processing can pertain to the amount of violinist's vibrato, and similarly score-level refer to the impromptu chord progression the melody is being improvised within. Interestingly however, on a continuum of micro to macro, these control levels loosely outline the evolutionary progression of improvisational control of musical instrument design–a major focus of Music Technology. As instruments were in their infancy, they could produce notes, then, as they became more refined, development of timbre was of utmost importance. Then, a jump all they way to computer music, note, effect, and score-level control have all been an equal playing field.

Contexts Of Improvisation

In any given performance situation while operating on these various levels of improvisation, the improviser is filtering their performance through some (or multiple) type(s) of construct(s), almost none of which are mutually exclusive.[5] Common constructs to improvise within are:

- **Tonal –** adhering to some type of established harmonic rules[6] (scales, keys, chord progressions, etc.)
- **Rhythmic –** adhering to some type of established rhythmic and temporal rules (time signature, tempo, etc.)
- **Timbral –** focusing on the timbre of the sonic spectrum
- **Dynamic –** focusing on the volume of the sonic input(s)/output(s)
- **Programmatic/Procedural –** following within the context of some sort of instructions, pattern or procedure.
- **Contour –** focusing more broadly on direction and shape of notes/pitches

[5] The only mutually exclusive ones are tonal/atonal, and rhythmic/arrhythmic.

[6] 'Adhering to' also includes being able to intentionally break said rules in a comprehensive fashion, as 'breaking rules' implies that 'rules' have already been established.

- **Atonal –** adhering to no harmonic rules whatsoever[7]
- **Arrhythmic –** adhering to no rhythmic rules whatsoever

Usually all of these contexts are considered and utilized as a filter in a given improvisational performance, many in simultaneity. Because there are so many (and likely more not touched upon), it is also possible for the performer or their instrument to omit some of them, whether intentionally, or because of some type of limitation.

Forms Of Improvisation

Operating on levels and filtering through contexts are merely observational methods of categorizing and differentiating what improvisation is on a conceptual plane, in the hopes that it can be objectively evaluated. The actual form improvisation takes when it creeps out of the intent of the performer can also be classified and categorized. Most of the more notable ones include:

- **Solo –** an improvised passage within a larger body of work or song, often with many of its contexts already set about by that work
- **Ornamentation –** a usually slight and momentary departure from the fixed work or song within any improvisational context or level
- **Free Fantasy –** a completely improvised entire work or song
- **Timbral Manipulation –** an improvisational manipulation of timbre of a sound
- **Buffer Manipulation –** an improvisational manipulation of (usually the playback of) pre-recorded audio
- **Live Coding –** an improvisational reprogramming of one's musical system via on-the-fly computer programming
- **Controller Mapping –** an improvisational remapping of one's controller
- **Autonomous –** utilization of computer algorithms, randomness and machine learning for a musical system to improvise on its own with little-to-no human input

[7] Whereas a tonal context can temporarily break tonality, an atonal context is differentiated in that it implies there is no adherence to tonality at all.

This is by no means an exhaustive taxonomy of every form of improvisation, and like improvisational contexts, they can be mixed and matched. As will be discussed in the following chapter, the first four are introduced toward the beginning of (pre-Baroque) Western Music, and still persist, while the rest came about only after the advent of computers, mid-20th Century (except for Buffer Manipulation, which began shortly before computers with the first recorded audio mediums: records, tape, etc.). Interesting to note is the huge gap in time in origination of different types of improvisation.

Instruments Of Improvisation

There are not really any instruments that are made more specifically for improvisation rather than for more structured performance. It can be argued that improvisational ability lies solely in the creativity and technique of the improviser. In that sense, a harpist can be an equally if not more capable improviser than say a pianist. But consider the piano versus the harp as a performable instrument. Yes, they both have one control source (ultimately both a string) for each note, but a piano is more mechanically suited for volume control; a piano is more ergonomically suited for ease and speed in performance; a piano has additional foot pedals programed for macro shifts in the effect-level of performance; Maybe a piano was not specifically crafted for improvisation, but is it more appropriately suited than a harp? Quite probably so. Whether we are talking about acoustic instruments, or digital systems, if the instrument promotes ease in playing, if it provides ample variation (in note, effect, or score level), if it eliminates bandwidth (mental or computational) without sacrificing control, or if the amount of control added is proportionally larger than the amount of mental bandwidth required to operate, it is likely an instrument better suited for improvisation.

Improvisational Focus of This Work

This thesis aims to address each of Wanderley and Orio's first three Levels of musical control. Possibly out of familiarity, maybe out of relatability, perhaps purely out of personal musical preference, this work tends to center around tonal and rhythmic Contexts to improvisation, rather than atonal and arrhythmic. Though ripe with potential, the last three listed Forms of improvisation will be addressed little at best; instead, this work will attempt to utilize and explore the first five more traditional outlets in a modern context.

Overview

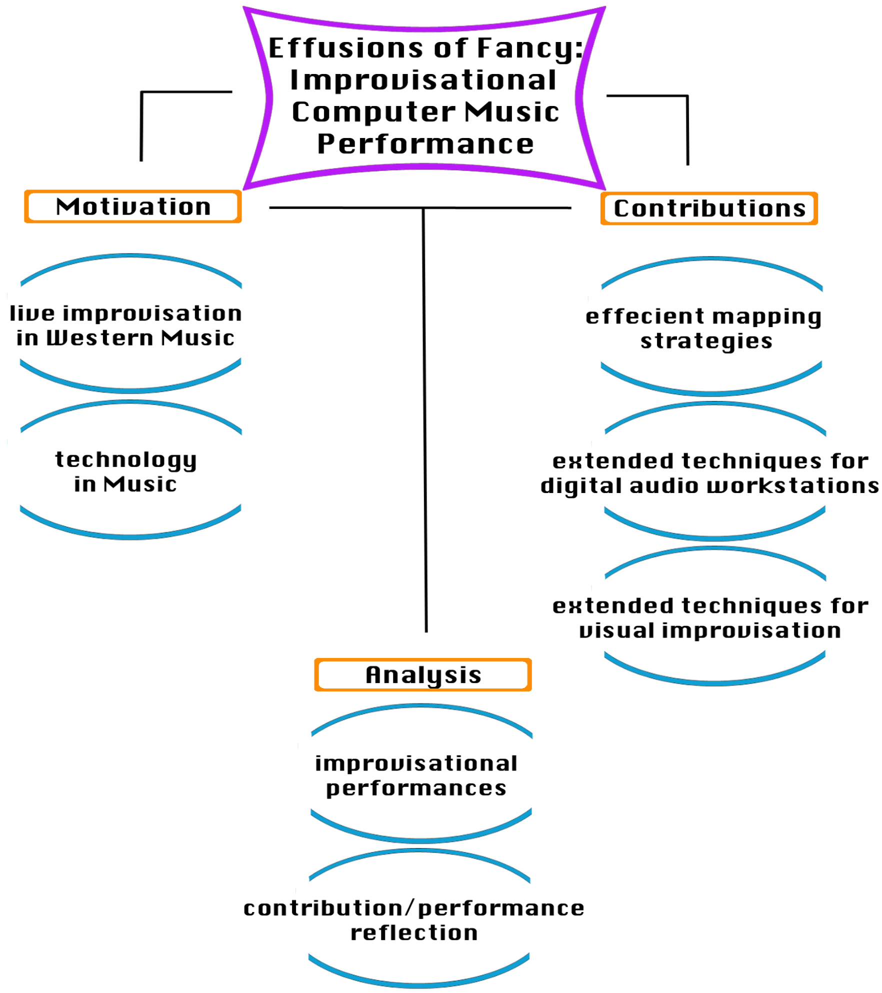


Figure 1. Thesis Map

The above diagram (Figure 1) is a map of how this thesis is organized. Understanding notable feats in improvisation and technology, and combining that with my own contributions, I then plan to perform live in different situations, and document my reflection through an improvisational mindset. Before this study, I have already played close to, if not more than 1000 performances in most styles, on different instruments–most often bass, various computer driven systems, usually improvisationally in some capacity. Digital systems are often designed and performed quite differently from performance to performance. So despite experience, I still commonly get nervous, doubtful, anxious, and feel underprepared in most live computer-music situations. The tools and improvisational performance documented in this work will ideally make improvisational computer performance more intuitive to help alleviate some of these woes, which will hopefully be of use to other improvisers.

Tools Of The Trade

It is the approach of this work to research, evaluate, and hopefully extend live performance tools that promote improvisation. Though contributions often need to be built from scratch (as many are and are documented throughout) either to better understand them or because they do not already exist, it can also be advantageous to leverage what is already in existence, assuming it has potential to be enhanced. In this sense, certain modern consumer grade and peer-contributed hardware and software tools are utilized throughout for testing and potential enhancement, and should be noted. Some of the more consistently used tools (discussed in detail later on) are: live performance software, Ableton Live Suite 9.1[8]; microcontroller, Arduino[9]; hardware MIDI controller, QuNeo[10]; hardware OSC/MIDI controller, Monome[11].

Contributions

Tools built for improvisational case studies of this work include:

LightBalloons – A new visual interface of balloons filled with full color LEDs, each individually addressable, created primarily for improvisational light-based visual live control, in accompaniment with music.

LiveMLR – A hypersequencing software add-on for Ableton Live makes MLR[12] part of Ableton's workstation. Unlike any attempt at previous versions of MLR to work inside Ableton Live, this is a fully functional MLR, with its own Graphical User Interface, that operates directly on Ableton's clips.

MIDI Organizational Patches – Computers grow more powerful and software more functional, which usually amounts to more fun for the performer, but can also lead to a need for a level of omniscience that performers do not possess. Although controllers are continuing to evolve, there is still much room for them to become more efficient and functional in the way of

[8] https://www.ableton.com/
[9] http://arduino.cc/
[10] http://www.keithmcmillen.com/QuNeo/tour
[11] http://monome.org/
[12] The hypersampling software instrument/musical system for the Monome.

organization, which could allow performers to reserve their mental bandwidth for more crucial performance matters.

Outline

Background - The history of improvisation in Western Music in the Background portion of this work should serve to demonstrate just how key the necessity for complete unrestricted control over real-time spontaneous sound creation is to the pursuit of advancement in musical technology (by direct intention, or not), and inversely, how technology has directly served to advance–even redefine–such fundamental musical concepts, like improvisation.

Improvisational Contributions – The technical chapters of this work highlight some of my major contributions to improvisation in electronic computer music to date, starting with advantageous MIDI routing and control programs built to help increase overall improvisational potential (like MIDIChain), and to help save an improviser from not depleting his mental bandwidth where it should not be needed (such as MIDISnap and MIDIRadio). Then, some case studies in more complete performance software (LiveMLR for Ableton Live and the Monome,) and hardware (LightBalloons), as they apply to improvisation.

Integration – Chapter 5 is a journal of live improvisational performance experiments within the duration of my Master's Curriculum at CalArts; a documentation of testing the improvisational suitability of my own contributions, various researched modern techniques, and other consumer gear/software, for the purpose of evaluating and understanding first hand what works and is worth perusing, and what is not.

Finally chapter 6 marks my conclusions from my research, and my account of modern improvisation in a digital age.

Along the way, many issues specific to this thesis will be addressed, such as: Is it better for the electronic improviser to be a virtuoso at their instrument/performance style, or a jack of many trades (even if we agree to define the arbitrary scapegoat of a word, better, for this context as: more spontaneous, enlivening, and fulfilling, is that pertaining to the performer, or the audience, or are they one in the same)? Are there general software or hardware tools that every electronic improviser should have? What are the pros and cons to using more all-purpose mass-produced tools versus designing/building one's own? Do certain tools, approaches, or styles promote more spontaneity? Or versatility? Can, or better yet, should an electronic performer incorporate as many differing performance tools/outlets into one system as possible, or is there a maximum amount of spontaneity or versatility that is aesthetically pleasing?

Of course, the simple answer to these questions is, "Its art; there is no right or wrong way for one to express their creativity through music." That being true, this art "is not an analytically impenetrable romantic chaos of emotions" (Michel Waisvisz, from Wanderley and Battier 2000); there are sure-fire ways to refine one's craft through analysis and experience. And though each analyzer might arrive at different answers to all of these questions, it is the hope that researching them academically, historically, and experientially through this work will provide insight for anyone seeking any kind of newness through modernity.

Chapter 2

A Brief History of Improvisation in Western Music Technology

In demonstrating a close relationship between improvisation and technology as they currently stand, it is one hope of this background to make abundantly clear a direct connection between Music Technology and musical innovation throughout time. It is also important to start back to the beginnings of Western Music so as to have a basis for what improvisation has been over time, in order to properly evaluate what it currently is, and via technology, what it is capable of being. If one traces far enough they will find that the groundwork laid for musical performance techniques, including improvisation, arguably predate Western Music's beginnings, basically remaining unchanged ever since (until recently with computer music). What has refined throughout Western Music performance in direct consequence from the field of Music Technology is the control that performers have had over their sonic output, input and more recently, over their whole musical paradigm (including enough control to relinquish any or all control whatsoever and set music to autopilot). To demonstrate what a lively and healthy component improvisation has been throughout the whole of Western Music, we will start with some of its earliest direct footprints. As this section progresses through time, the irrefutable interdependent relationship between technology and improvisation will begin to emerge.

Improvisation Pre-electronics

What constitutes as improvisation? Rob C. Wegmen, elegantly describes it as, "differences between what is played and what is notated on the page,"[13] as illustrated by Figure 2. Of course, this description does promote vagaries, as it does not account for things like accidents, or whole

[13] www.oxfordmusiconline.com

styles of music that are not notated, or that are notated in some kind of shorthand code. However, the description's effectiveness is in its simplicity. Essentially this definition implies that although there are possibly many different types of improvisation, they are quantitative, or measurable, assuming you have an account of "the page" and "the differences" played from that page. Today, everything is documented. People can (and do) record anything and everything. It is easy to study how a jazz musician like Miles Davis improvises, as we have numerous recordings of him playing; we have his own sheet music and charts, recorded interviews, biographies, even autobiographies. More than likely, this is the reason that the association between improvisation and Classical Music seems to have escaped our minds: because those historic unrepeatable moments of spontaneity happened before the advent of audio recording technology. Hence, before the 20th century, we have had "the page," but lacked its "differences."

Figure 2. Wegmen's definition of improvisation

As audio recording succeeds the height of the Classical Age of Western Music, the only possible ways improvisations of the past could have been preserved were by the words/writings of the composers, or of those who listened (critics, fans, students, contemporaries). As scarce as this type of documentation appears to be, there is enough to know that although improvisation has swept through high and low ebbs of popularity throughout various eras, it was never completely lost or forgotten, and is by no means a recent invention.

The Baroque era brings us some of the earliest evidence of formal improvisation with figured bass, or, variations on continuos. These notation techniques could be equated to a jazz chord chart of today wherein chord by chord, the lowest voice (bass note), chord quality, and relation to tonal center are all implied by a cryptic (but logical roman numeric) symbol; because only the bass note is specified, the performer had no choice but to improvise the rest of the voicing for each chord. Continuing into the Classical era, musicologist Robert Levin interprets Mozart's notated piano accompaniment of his orchestral tutis, akin to a Duke Ellington or Count Bassie swing band, where the pianist accompanies by tastefully choosing a riff or passage, provoking delight or otherwise intensifying the character of the music, preparing the audience for the piano solo about to come (Levin 2009, 147).

Perhaps of improvisation's more enticing qualities was one referred to in Romantic times as *intimate disclosure*–the revealing of ones deepest thoughts and feelings (Hatten 2009, 281). Performers often have a false persona when performing, but when improvising, a performer's true state of mind tends to creep out. During a Beethoven improv in 1790 at the Viennese Salons (a dwelling place for socialites of the day), "Beethoven became so engrossed in his inner world while improvising at the piano that upon glancing up and witnessing tears of his listeners, he suddenly insulted them and charged out of the room," feeling exposed to a room full of strangers (Hatten 2009, 283). It is ironic that sometimes the most profound artistic communication happens when a performer is immersed, even lost in his own private reflection.

In relation to false persona, contrary to contemporary popular belief, the association with the solo extends beyond the stereotype of a shirtless male, oiled-chiseled-bodied guitarist, with long windblown hair, standing under a spotlight, one foot on his monitor, and with his instrument pouring out his soul. The solo is one of the more common manifestations of improvisation prevalent in most styles of music, globally, as it has been for centuries (possibly millennia). Roughly 200 years ago Classical Music referred to the solo as the Cadenza, which Merriam-Webster describes as "a technically brilliant sometimes *improvised* solo passage toward the close of a concerto."[14]

Like a solo, the Free Fantasy–called, "Fantasieren", by Beethoven (Kinderman 2009, 297)–was a popular and virtuosic form of improvisation in the Classical and Romantic period. Unlike a cadenza, it was a full piece on its own, and there were no constraints; the length, key, time signature, and tempo of the piece were no holds barred. The themes incorporated were are all subject to the whim and spontaneity of the performer-composer. This ultimate control over not just the note and effect levels of performance, but over the score level, as well as the entire musical system, is the earliest precursor that most closely resembles the more advanced paradigmatic and score-level type control that modern music of the digital domain can offer.
Ferdinand Reis recounts his teacher, Beethoven's affinity for extemporization, from witnessing his Quintet for Piano Forte and Wind Instruments:

[14] http://www.merriam-webster.com/dictionary/cadenza

> In the last allegro there are several pauses. At one of these Beethoven suddenly began to improvise, took the rondo for theme and entertained himself and the others for a considerable time, but not the players. They were displeased...even very angry. It was really comical to see them, monetarily expecting the performance to be resumed, put their instruments to their mouth, only to put them down again. At length Beethoven was satisfied, and dropped into the rondo, the whole company was transported with delight.
>
> (Kinderman 2009, 289)

Improvisatory preludes before a larger work were customary for performers during the Classical and early Romantic era. Preludes were intended as pleasant introductions and interludes, but also served practicality. They allowed the performer to warm up, set the mood, adjust for any necessary tuning, test the instrument, and give pitch and tempo for a vocalist if there was one, among other functions. Philosopher and composer, Jean Jacques Rousseau, in a published entry for the Encyclppedie (1765) noted that "the art of preluding is something more significant; it consists of composing and playing on the spot, charged with the most aspects of composition, in fugue, imitation, and harmony" (qtd. Temperley 2009, 325).

Philip Anthony Corri in London published, Original System of Preluding, with analyses and over 200 preludes. In it he said, "In the performance of preludes, all formality of precision of time must be avoided; they must appear to be the birth of the moment, the effusion of the fancy" (qtd. Hatten 2009, 321).

It was during and after Chopin's time that the art of preluding began to die out and be replaced with canonization, as pianists did not want to stray from the page (Temperley 2009, 327). By now, preludes and cadenzas have been given notated realizations, usually by the composers themselves. The profound impact of improvisation is exemplified in that during its peak moments of introspective exploration, new discoveries can be made that can transcend, or redefine a work, a composer's style, and even a whole genre of music altogether. This fits the case for improvisation's decline in Western Art Music around this time, "as the bold liberties (...) that once characterized [a piece] became compositional norms, thus undermining its special status."[15] And so it is that very affinity for improvisation, which expanded Western musical

[15] Rink, John. "Improvisation." *Grove Music Online. Oxford Music Online.* 30 Apr. 2012

compositional norms, in turn brought about improvisation's own demise (or rather, hibernation).

Whereas an innate goal of improvisation is an attempt to unite the roles of performer and composer in real-time, as the turn of the 20th century waned, Western Art Music began to show its first strives to free the composer again from the notated musical work. Performance utilizing chance, algorithmic, serial, procedural, and atonal/timbral/spectral type compositions (Cage, Xennakis, Stockhausen, Zorn, etc.) became the fad. Although these styles could loosely fall into a category of spontaneous sound creation, they usually contained strict performance instructions, and lacked intent from the composer, making them a sort of improvisational-grey area. However, like a virulent virus, improvisation found a new suitable host to preserve its vitality in the emerging and increasingly popular Western styles of Jazz and Blues.

As time marched on, the styles changed, but the types of improvisation remained the same. Though improvisation is basically the integral component of Jazz and Blues, if anything, the differing types of improvisation utilized in said styles dwindled as compared to earlier Classical music. From about the beginning of the 20th century until about the 1970s, Jazz and Blues were restricted mostly to a note-level type of improvisation. The height of the Jazz/Blues era (the 1950s-1970s) can rightfully be described as the height of improvisation in Western music to date, but the improvised solos, and chord voicings and such, are conceptually nothing different from their Classical predecessors (Figure 3 compares Baroque figured bass to a 20th century jazz chord chart, showing different symbols that essentially imply the same improvisation).[16] [17] During and after the 1970s, forms of Jazz like early Jazz Fusion and Freejazz

[16] It may seem peculiar that this work lists no examples of improvisational Jazz/Blues performers and performances. This work makes the assumption that the main association with Jazz/Blues is improvisation, and inversely the main association with improvisation is likely Jazz/Blues to a general public. Needless to mention, examples of improvisation in Jazz/Blues consist of almost every Jazz/Blues performance that has transpired, and so examples are so numerous that providing just a few would not do justice to the history of improvisation in jazz. This work is interested in how the ability to improvise has progressed over time. Other than becoming the forefront of a style of music, the way in which one improvises in Jazz/Blues has posed no enhancement toward improvisation as a technique (see Figure 3). For great examples of improvisation in Jazz/Blues, just listen to any decent Jazz/Blues (Duke Ellington, Miles Davis, John Coltrane, Bill Evans, Stevie Ray Vaughan, etc.).

[17] To further exemplify the pre-existence of note-level improvisation, we should briefly turn toward Indian Music–one of the oldest fully developed musical styles in human history. This thesis is centered on Western Music, but it should be noted that the similarities in improvisatory style and technique between Indian music and Jazz is uncanny. The primary difference is that in

(Ornette Coleman, Sun Ra, Miles Davis), which had little-to-no score or structure allowed for more of a potential focus on score and effect-level-type spontaneity. Jazz and Blues fully embraced electronics into their repertoire, with amplified guitars, electronic organs, vibraphones, and the like.[18] The overlap makes for a great transition to how the incorporation of electronics in music influenced improvisation

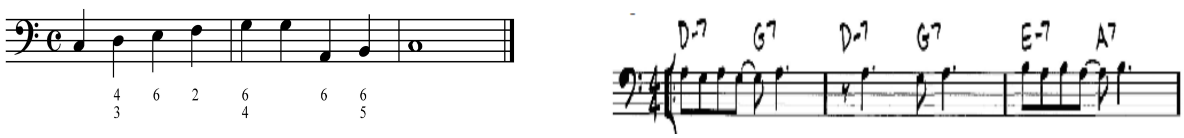


Figure 3. Baroque-style figured bass (left) vs. Jazz-style chord chart (right); different symbology with almost identical meaning

Improvisation post-electronics

"Technology helps to transform sounds, and make something new out of the old."

Alex Ridha (Boys Noize)[19]

One of the most obvious and noticeable evolutions of Music Technology in the grand scheme of Western Music was its incorporation of electronics. Most profoundly, electronics provided a means to physically store memory, one product of which was recording sound, beginning with Thomas Edison and the phonograph (Figure 4), then eventually via records, tape, and digital storage. The technology of audio recording has provided for the most practicable analysis of sound, which has been most salutary in effect on the proliferation of (the affinity for) improvisation in Western Music. It provided a means for emerging styles predicated on improvisation to become more widely listened to and enjoyed, and hence, popular. This is in all likeliness why we equate popular styles post recording technology like Jazz and Blues to improvisation: because we have been able to listen back to the style since its beginnings, as it has developed through and from improvisation. This bit of technologic and stylistic history implies that as technology advances, improvisation proliferates.

Indian music the harmonic singularity of the Rag (key center)–hence, lack of harmonic movement–allow musicians to place a greater emphasis into *rhythmic* improvisation focusing on more improvisationally complex phrasing within complex time signatures.

[18] Though they fit the technical definition of electronic, these instruments are often distinguished as 'electric', from later, more coupled electronic, less acoustically driven devices like synthesizers, samplers, and computers.

[19] http://thecreatorsproject.vice.com/creators/boys-noize

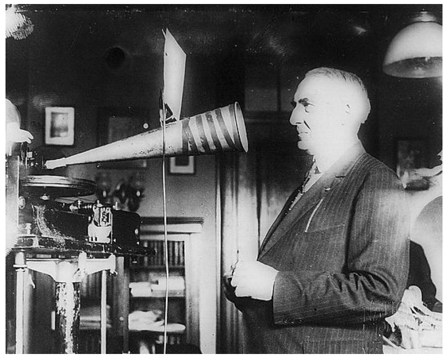


Figure 4. Edison and the first phonograph

More directly applicable to musical performance, electronics in music essentially allowed sound for the first time to not be enslaved by the physics of acoustics; no longer was the need for direct energy to be placed into a resonating body to sustain a tone, as with acoustic instruments. Instead, control sources could be sent out to trigger oscillators, or indirectly control other parameters of sound, creating for the first time the notion that an input source could be decoupled from an output source.

The beginnings of electronics incorporated into musical performance, being a new science, did not automatically revolutionize the improvisational ability of electronic instruments over night. Most of the history of Music Technology post electronics-to-now involved decades of trying to make electronic instruments as improvisationally capable as the millennia spent developing their acoustic counterparts. It was not until computers and digital hardware came about in more recent times that set about a significant evolution in improvisational ability, by placing complete control in the hands of the performer, as will be discussed further in this section. Until then, in what appears to be out of a continued need for more improvisational control, the history of electronic instruments underwent constant refinement, and technological advancements to make something new out of the old.

The Hardware angle

> The controller is the first component of the digital instrument chain. Controllers constitute the interface between the performer and the music system inside the computer, and they do so by sensing and converting continuous and discrete analog control signals coming from the exterior into digital messages or data understandable by the digital system.

(Sergi Jordà 2005, 29)

The hardware side of Music Technology generally refers to the instrument itself. Whereas acoustic instruments always have a coupled input and output source, as we shift further through time, electronic musical instruments tend to exploit their decoupled nature by dividing into separate input and output devices. As computers and their software have grown robust enough to enter the real-time musical performance paradigm, they predominantly usurp the position of outputs, relegating the bulk of modern musical hardware to (highly decoupled) input controller devices (for computers). To show how electronic instruments can differ from acoustic ones, Sergi Jordà provides a concise list of restrictions and rules that have governed the lutherie of acoustic instruments (Sergi Jordà 2005, 23–24).

- Acoustic instruments, with few exceptions (including organs and several other keyboard instruments), impose by construction their own playability rules, allowing listeners to infer the type and form of the gesture from the generated sound. There may still be room for some control design improvements but scarcely for capricious decisions or radical changes.
- No instrument allows for a complete control over its whole sonic richness. Microevolutions of their parameters are often uncontrollable and subject to a combination of (a) complex correlations (b) structural and morphological constraints and (c) additional random -or ungraspable- factors.
- With few exceptions (as it is the case of the organs) the excitation energy needed for the instrument to sound has to be provided by the performer.
- With few exceptions, not only energy but also all type of temporal control variations or modulations of parameters (e.g. vibrato or tremolo) have to be explicitly and permanently addressed physically by the performer.

This list establishes a comprehensive technological jumping point for electronic instrument design. With electronics, all of these restrictions can be (and have been) overcome.

2.1.1.1 Music Becomes Decoupled

The first explorations into electronic hardware were not made to be decoupleable by the performer, nor did they necessarily supersede acoustic instruments in improvisatory control. They did however explore the uniqueness of electronically generated synthesized sounds, essentially placing a spin on the effect-level of improvisation right from the start, as well as provide new methods of physically generating sound. As previously mentioned, the piano (and its derivatives) were a major turning point in enhancing control over a musical performance. The first developers of performable electronic musical instruments in the 1920s were interested in cracking this keyboard-dominated institution. Invented in Russia by Lev Sergeivitch Termen (Leon Theremin), the Theremin was the first fully-fledged spatially controlled gestural controller

(Gallin and Sirguy 2009, 199). Nothing is touched in the controlling of the Theremin; instead, hands are waved about, picked up by two antennae that measure body capacitance for outputting a continuum of pitch and amplitude. The way in which it is controlled makes the Theremin a very continuous gestural instrument almost opposite to the discreteness of the keys of a piano (see Figure 5).

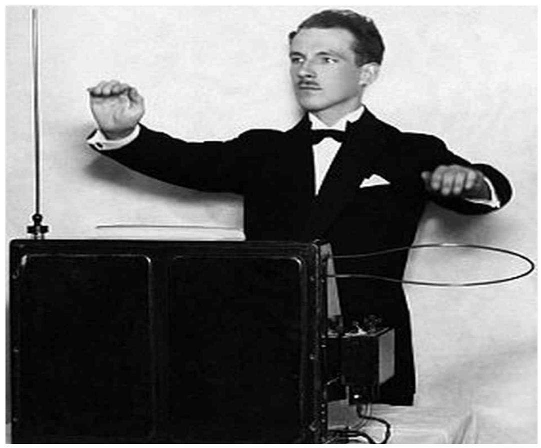


Figure 5. Leon Theremin (inventor), playing the Theremin

The next 20 years saw a movement back toward piano-like electronic instruments with examples like the Martenot, the Ondioline, the Trautonium, respectively, which all offered different input control sources, and progressively more refined synthesized output sources (Gallin and Sirguy 2009, 199). The Martenot, by Maurice Martenot, had the same timbre and continuous control as a Theremin, but could be played more accurately by sliding over keys,[20] providing a melodic glissandi (portamento) not possible with an acoustic piano. This invention effectively enhanced both the note and effect-level potential for improvisation for keyboard instruments. The Trautonium and the Ondioline came soon later with more robust synthesis, the Ondioline was also especially unique in that its keyboard bed was laid on a spring which provided a controllable vibrato effect also previously acoustically unachievable on a keyboard instrument (Fourier, Roads, and Perrey 1994, 19). The 1940s brought the first voltage-controlled synthesizer that used different waveforms as timbre, The Electronic Sackbut by Hugh Le Caine, a precursor to the popular analog synths that followed[21] (Buchla, Moog, Serg, etc.). It was built

[20] Done basically by sliding about a ribbon connected to a potentiometer that affects pitch.
[21] www.hughlecain.com

with ergonomic effect-level control under the notion that an added timbral flexibility was something a performer should be able to "feel without carefully watching the controls."[22]

Though the input sources of these first electronic instruments were always hard-linked to the output sources by their creators and never technically decoupled, the breakaway between input control source and its respective output sound demonstrated a potential for modular linkage, inciting the decoupled instrument tradition in electronic music that was soon to follow.

By the 1960s, electronics in musical performance were commonplace. They had infiltrated the popular styles like Jazz and Rock and Blues, as well as the more academic styles, collectively referred to as 20th Century Western Art Music. Instrument signal chains were now commonly running through a multitude of effects either from pedals or built into amplifiers, allowing for on-the-spot effect-level control (Beatles, Jimi Hendrix, Pink Floyd, Grateful Dead, Stevie Ray Vaughan, to name a few). Electric pianos, and electric organs were from then on extremely popular in Jazz and Rock (Jimmy Smith, James Brown, etc.). And the new wave of analog synthesizers, invented independently in 1964 in different parts of the world by Robert Moog, Donald Buchla, and Paul Ketoff, were just begging to be improvised upon, more on a note-level in popular music with Moog who offered a traditional keyboard to control traditional sonic parameters (pitch and volume); and more on an experimental effect-level with Buchla and Serge, who offered less conventional pitch generating control sources. At this point in Music Technology, what was improvisationally possible with effect-level control via these new electronic devices was surpassing the possibilities of the acoustic realm. Meanwhile, modular synthesizers around this time were pioneering the first user definable decoupled instruments by allowing any synthesis parameter to be routed into any other synthesis parameter, in any combination. Many things could control one thing, or one thing could control many things, introducing a concept of complex patching, called mapping. By the 1970s, the ability to sequence rhythm and pitch in analog Buchla and Serge synthesizers allowed for a user to create harmonically and rhythmically rich, improvised performances by providing automation–in the form of sequences–on the note and score-levels, while simultaneously offering vast improvisatory timbral freedom.

[22] Similarly today, the trend is to create a musical system that does not require staring at a computer screen.

2.1.1.2 Going Digital

> "As we moved from the transistors of the '60s to the integrated circuits of the '70s, computers and analog synthesizers became less expensive and easier to use, and they were often joined together in what were called hybrid systems."

(Joel Chadabe, Electronic Musician)[23]

During the late 1970s, personal computers were just barely capable of limited real-time control over sonic parameters.[24] The first digital synthesizers (Dartmouth Digital Synthesizer, Synclavier), standalone drum machines/sequencers (Roland, Sequential Circuits, Oberheim, EMS), and samplers (Fairlight, Emu) were becoming publicly available, at progressively more affordable prices.[25] These types of devices dawned an awakening for ease in performance of Wanderley and Oreo's third type of improvisational technique: score-level improvisation. After the standardization of digital communication between these devices in the mid 1980s (MIDI–discussed more in the software section of this chapter), all three levels of improvisation were beginning to be explored more equally, in isolation, and in conjunction with one another. At this point, the control over the decoupling of musical inputs and outputs were trickling down to the users of them, and not just the designers; if a performer had the proper equipment, he could finally in real-time alter and automate full arrangements of music with very little input control; a sort of musical super power not previously available before this time.

Around the mid 1980s, fringe music technologists–digital luthiers[26] were cropping up and exploring less conventional and commercial means of digital instrument design. One of the first instruments in this category, subject to much subsequent iterative development by other designers there afterward, was the Radio Baton, invented by Max Mathews and Bob Boie (Lawson and Mathews 1977; Mathews 1991). The original controller consisted of two percussive mallets, called batons, each with radio transmitting antennae that when struck upon an electroplate, could track each baton's location in 3D, generating a total of 6 independent data

---

[23] http://www.emusician.com/gear/0769/the-electronic-century-part-iv-the-seeds-of-the-future/145415

[24] Exhibited by early computer bands like The League of Automatic Composers and The Hub.

[25] http://www.emusician.com/gear/0769/the-electronic-century-part-iv-the-seeds-of-the-future/145415

[26] Term gathered from (Sergi Jordà 2005).

points (see Figure 6). Mathews demonstrates its ease of use in a video[27] where he controls tempo and multiple channel volumes, while stepping through a fixed classical score of music, beat by beat, requiring no skill other than hand-eye coordination of striking a plate with a mallet. With radio batons, Mathews was among the first to use a computer to digitally improvise within a score-level, driven to show the world how easily a computer can accomplish any musical heavy lifting, in his vision rendering "the many hours of practice most performers have to put in to learn technique unnecessary" (Mathews 1991).

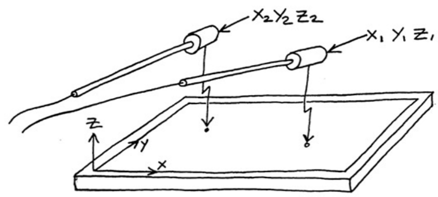


Figure 6. Diagram of Sensor Control for Mathews' Radio Batons

2.1.1.3 Further Decoupled

"A contrasting approach to predefining interface behaviors for a particular performer's needs is to create an interface with a selection of basic inputs and undefined behaviors."

(Vallis, Hochenbaum, and Kapur 2010, 1)

As the 1970s drew to a close and the separation between input and output source of a musical system became one of the clear-cut benefits of working in the electronic realm, a focus on creating new types of controllers for improvisational performance was on the rise. In what almost seems like a direct attempt to show that electronic/digital instruments could improvisationally "surpass" their acoustic counterparts, instrument design in Music Technology of the 1980s and 1990s branched into either (1) methods of controlling and outputting sounds and performance tactics that are unique and not replicable through acoustic means, and (2) methods simulating or enhancing with electronics the improvisational breadth of acoustic instruments themselves, creating hyperinstruments.

Michael Waisvisz and Nicolas Collins are two pioneers in the early stages of digital controller interface design for music. Waisvisz–virtuosic live performer and improviser–is most known for the musical controller, The Hands, described as: "Two aluminum ergonomically shaped plates with sensors, potentiometers and switches strapped under the hands of the

[27] http://www.youtube.com/watch?v=3ZOzUVD4oLg

performer. The analog information generated by finger and hand movements are sent out as MIDI messages to the sound and music generator (Sergi Jordà 2005, 69).” More closely adhering to the second branch of electronic/acoustic hyper-instrument design, Collins could be found in his early days playing (not traditionally, but the buttons he wired into) an electronically enhanced trombone. Both were demonstrating that adding some “digital-ness” to their system distinctly expanded the scope of their musical palate.

The capability for improvisation was magnified in the 1990s with a unique digital musical device unlike any acoustic (or electronic) performance instrument before it, the MPC60 (and future successive models), created jointly by Roger Linn[28] and hardware manufacturer, Akai. The MPC (Figure 7 left), short for Music Production Center, is a portable digital on-the-fly sequencing/sampling station, designed to be easy to use out of the box, with no additional equipment required, complete with button pads and knobs for live performance. Samples could be triggered, hence performed on a score level, or broken up and/or pitched for more melodic note-level improvisation. Though the all-inclusiveness of this popular digital instrument initially appears to break the decoupled trend of its predecessors, its back panel since its inception came complete with MIDI In and Out ports, allowing it to be just an input controller, or just an output sampler, if so desired.[29] The popularity of the MPC set a precedent for an ongoing intrinsic linkage between button pushing and live electronic performance.

One of the more innovative musical interfaces of this same period for the effect-level domain was the Korg Kaoss Pad. Unlike a traditional foot-controlled stomp box, and unlike a rack-mount studio effects processor, this was a hand-controlled XY trackpad effects processor-sampler-controller. Like the MPC, the Kaoss Pad came decoupleable with a MIDI port for sending out continuous control data. Both this device and the MPC, when viewed as a purely uncoupled MIDI controller, became a precursor to the uncoupled computer controller revolution that we have currently grown fond of.

Although decoupled, most devices of the pre/early 1990s were designed with pre-built functionality; they all “had brains of their own,” so to speak. The increasing self-sufficiency of laptops around this time were showing that a controller hooked up to them did not need much for brains; that control and modularity could be enhanced if these musical interfaces had no

[28] Roger Linn was the creator of the LM-1, first drum sequencer to use digital samples of real drums

[29] (“Feature: Industry Interview -Roger Linn @ SonicState.com” 2013; “Roger Linn On Swing, Groove & The Magic Of The MPC’s Timing - Attack Magazine” 2013a)

preset functionality other than to merely digitize and transmit the data from its sensors; that much more functionality could be specified and personalized if left to the central processing hub of a computer.

Figure 7. Turning point electronic instruments: MPC60 (left); Kaoss Pad (right)

2.1.1.4 Current Paradigm: (Micro) Controller Revolution

As music technologists and musicians of today's computer age, we have been able to live the dream of real-time computer music performance, which in turn lays witness to noticeable advancements in improvisation. The newly anointed performer-composer-conductor-mixer-even digital luthiers are now more limited by their own creativity and mental bandwidth than computational bandwidth.

An underlying goal in Music Technology of the past 40 or so years has been the determination to relieve digital musical systems of their inhuman rigidity, basically devising super-humanly precise ways of being imprecise. Ample research and development has been put forth into making digital instruments, not equally, but vastly more sensitive and encompassing than their acoustic counterparts. A major contributor to this cause was the increasing proliferation of microcontrollers like the Atmel AVR ATMega series,[30] (Kapur, Davidson, et al. 2004), which convert analog sensor data (like the change in resistance of a potentiometer) into digital data, to be interpreted by a computer. These microcontrollers made possible augmented instruments–acoustic instruments enhanced with sensors and microcontrollers. Early masters of this domain include Waisvisz, Collins, Trimpin[31]–having built an automatic "player" for almost every major acoustic instrument–, and Perry Cook–who set forth guidelines for digital instrument design in papers like, Principles For Designing Computer Music Controllers (Cook 2001). A more recent master and disciple of Trimpin and Cook, is Ajay Kapur, with an ever increasing body of hybrid acoustic/digital traditional world instruments, such as his ETabla and

---

[30] www.atmel.com

[31] Trimpin – The Sound of Invention, is a great documentary for more detail into the life and incredible inventions of Trimpin (Esmonde 2009).

ESitar (Kapur, Davidson, et al. 2004; Kapur et al. 2003; Kapur, Lazier, et al. 2004). Much of the focus of Kapur's extended instruments is to preserve "centuries of tradition passed on from guru (teacher) to shishak (student), while enabling a curious performer to experiment with new tools for expression" (Kapur, Davidson, et al. 2004, 7). With these electronic acoustic hybrid instruments doubling as instrument and controller, Kapur is able to improvise traditionally, while also triggering samples, controlling effects such as delays, ring modulation, and sympathetic pitches, even recording of his gestural data to play back and modify.[32]

As digital versions and hybrids of more traditional instruments have been approaching perfection, a need for new instruments–like the MPC and Kaoss pad–to adequately satisfy the performance of newer styles and performer (now performer-composer-arranger)-driven roles also exists. An influential invention carrying on this tradition that really grasped the potential might of computers coupled with extremely decoupled controllers was the button-grid controller, Monome, invented in 2005 by Brian Crabtree.[33] Its form, function, and essence is eloquently detailed by Vallis, Hochenbaum, and Kapur:

> Monome is both a two-layer uncoupled NxN device consisting of a matrix of silicon buttons situated over a matrix of LEDs. (…) Monome's minimalist design philosophy manifests in the company's production of interfaces that avoid complexity in order to promote greater possible versatility.
>
> (Vallis, Hochenbaum, and Kapur 2010, 2)

Brian Crabtree alludes to the minimalist potential of a highly uncoupled interface in noting, "We see flexibility not as a feature, but as a foundation."[34] The user-configurable button mapping possibilities, in accordance with completely programmable (coupled or decoupled) light feedback, allow for performer/composers to take complete command over their inputs and mapping of their musical system to accomplish more advanced improvisational feats. The

[32] Notable demonstrations of Kapur's ESitar can be found in his performances with the Machine Orchestra (discussed further in the performer section of this chapter).

[33] Tenori-on is another grid controller developed independently at the same time by Japanese artist Toshio Iwai and sold to Yamaha (Nishibori and Iwai 2006). Its corporate marketing strategies, closed source philosophy and less pleasant aesthetic than the Monome contributed to a lack of demand for it in the consumer market. Most if not all subsequent grid controllers have been modeled after the Monome.

[34] www.monome.org

minimalist decoupled concept of the Monome combined with its fully open source documentation inspired a community-driven tradition of "iterative musical interface design" (Vallis, Hochenbaum, and Kapur 2010) process to make Monomes more performer-friendly, and make the type of controller, a grid controller, a staple in computer performance. Among the first was the Arduinome by Flip Mu (Owen Vallis and Jordan Hochenbaum). The Arduinome requires a bit more technical knowledge in order to build and use, but provides a generic cheap and available[35] means of owning a Monome-like grid controller. Interestingly, the different components of each 'nome are compatible to be mixed and matched, kind of like a technological Mr. Potato Head. Future iterations led to things like full color RGB LEDs for increased feedback and hence, ease in improvisational performance, as in the Octinct, by Brad Hill, Jonathan Guberman, and Devon Jones. A different grid controller with pressure sensitive continuous buttons, the LUMI, developed at Stanford by Adrian Freed and Mike Gao (Gao and Hanson 2009; Vallis, Hochenbaum, and Kapur 2010) served to increase continuous controllability and functionality. Eventually RGB feedback and pressure sensitivity were combined, first by Owen Vallis with the Chronome, and eventually by more commercial companies like Ableton Live with their Push. Many major musical hardware manufacturers now sell their version of a (non-open source) highly uncoupled grid controller (as in the Novation Launchpad, Akai APC40, more recently Ableton/Akai Push, Keith McMillan QuNeo). Furthermore, the decoupled and user-programmable nature of the controller has become an expectation out of contemporary computer-based musical controllers. All it took was one technologically sound idea to fuel an iterative revolution in instrument design. See **Figure 8** below for but a fraction of the grid controller family.

Whether on a commercial or homemade interface, there exists a sensor to recognize any gesture, and anything in our real world that represents a change in data can in itself become a sensor to be used in performance. As abundant as sensing technology now is, it is the opinion of the author that furthering improvisatory potential in musical hardware currently lies less in the innovation of new input-based sensing technology, and more in in how that data is organized

[35] Monomes are handmade by Brian and his wife, which greatly limit their supply (its difficulty to obtain rendering it even more desirable); since the inception of Monome as a company, acquiring a Monome was/is similar to buying concert tickets to your favorite band/festival that sells minutes after they go on sale; should someone want one, he/she would have to wait until the minute (realistically, hour or two) that the latest batch went on sale, hoping they had not yet sold out before payment sent.

and output within a controller. One company currently realizing this in their products is Keith McMillan,[36] responsible for numerous silicon button pad controllers (QuNeo, QuNexus, Softstep, etc.), all of which can output multiple MIDI commands from one sensor, and allow most all sensor output to be user defined/tweaked. Like Monome they also provide comprehensive user-modifiable visual LED feedback to indicate the state of its sensors. This beginning of selectable complex functionality that is user defined seems to be the next evolution of computer interfacing for ease in improvising. It is from this point that Chapter 3 will diverge.

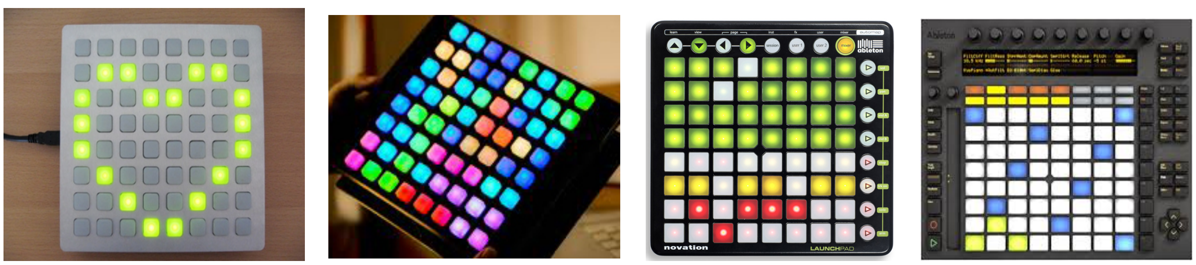


Figure 8. Grid Controllers: Monome, Chronome, Launchpad, Push

The Software Angle

The Computer's competence and acceptance in the field of real-time musical performance is extremely recent in the grand scheme of Western Music, dating not much farther back than the dawn of the 21st century. The possibility of computers controlling improvisational techniques in a real-time musical context is history-in-the-making. Just as improvisational freedom has been intrinsic to the iterative development of the hardware side of Music Technology, it has also been a major influence throughout software development for computer music.

2.1.1.5 A Shift Toward Real Time Control

The field of Digital Music Technology basically started in 1957 with Max Mathews' development of the musical software, MUSIC. Successive revisions (MUSIC II–MUSIC IV), and developments like the C programming language, and its families, (C++, etc.) brought major innovations to digital music (CARL, Cmix, Csound, DSP libraries for synthesis and audio control, etc.), yet real-time audio control at this point was still considerably lacking. Major innovations in real-time music performance came around the mid 1980s with music software built specifically for real-time manipulation of control data called, Max, by Miller Puckett. This

[36] http://www.keithmcmillen.com/

program was unique in that it was controlled and manipulated using graphical representation (as opposed to text based code), as shown in Figure 9, which made the virtual visualization of signal flow much more intuitive for musical programming, and computer-based performing. "A Max patch is at once a program and (potentially) a graphical user interface" (Wang and Adviser-Cook 2008).

At around this time, a standardization of data transmission just for musical instruments, known as MIDI (Musical Instrument Digital Interface), allowed for a plethora of digital musical input and output devices (previously mentioned sequencers, samplers, synthesizers, etc.) to flourish, and aided in digitally performed music's spotlight into the mainstream. Before MIDI, live digital music performance consisted primarily of "computer music *sound* creation." With MIDI being a note–or pitch–centric system, "computer *music* creation was able to become a real-time activity" (Sergi Jordà 2005; S Jordà 2002).

Just after the standardization of MIDI in the mid-1980s, personal computers of the age were being built with musical manipulation in mind. The Apple II Plus was prebuilt with advanced digital to analog audio converters of the age, and AtariST actually featured MIDI in and out ports built into its hardware (Manning 2013). At this point, so long as an aspiring digital performer was proficient enough with computer science, they could create an improvisational system all to their own.

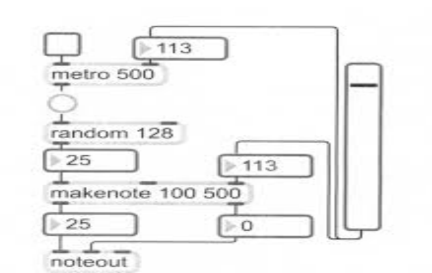


Figure 9. Example of Graphical Musical Programming

2.1.1.6 Programmer-Oriented Software

As the 1990s progressed, computers became more user friendly, and hence, commonplace; it was becoming no longer necessary to have computer science knowledge to operate one. A live performance software tool was still in demand that performers could improvise within, rather than needing to spend the time learning to program. Digital Audio Workstations originated either for composing and organizing with MIDI (Cubase, Logic), or for recording audio (Protools), eventually merging into different flavors of the same virtual system for routing, controlling, processing, and recording audio and MIDI. Still yet, a true harnessing of the potential of digital improvisation was relegated to those who could build, or at least operate custom-built software.

By the late 1990s, major developments had been made in audio performance-based programming environments, making live computer music more feasible. SuperCollider, a text based environment designed for real-time synthesis, and algorithmic composition, was created (McCartney 1996; McCartney 2002). Additionally, Puckett added audio DSP to his data control environment, Max, making it Max/MSP (Wang and Adviser-Cook 2008, 20).[37] With user-friendliness in mind, Max eventually created extremely intuitive user help/reference documentation, a model that future programming environments soon followed. As more programs made just for audio control/manipulation proliferated, musicians obtained more and more control over designing their improvisatory musical systems; whereas electronic instruments increased what is improvisationally achievable effect-level-wise, computer instruments vastly increased what is possible via the score-level, ultimately bringing all three of Wanderley and Orio's improvisational levels to a performable equilibrium.

Electronic music grew to become more improvisable as software design grew to become more prevalent in the hands of the performer. Be it the expertly designed DSP (by Vadim Zavalishin and Martijn Zwartjes) of Reaktor, or the strongly timed and conveniently "syntaxed" coding of ChucK, or the efficient and powerful low-level robustness of JUICE, the next and more current family of audio programming environments seem to take audio software design to a pedagogical domain, providing modules, objects, examples and thorough documentation, abstracted to make audio programming not only easier to code, but easier to learn, and hence, more available to a general musical public.

2.1.1.7 Necessity For a Live DAW

Although laptops for live performance were gaining popularity by the onset of the 21st century, there was still a lack of any standard live performance software. To use a computer exclusively for a performance, one still had to either have some computer science chops and make (or use) a program in Max/MSP, Reaktor, Supercollider, or something similar, or be forced with the very

[37] http://www.emusician.com/gear/0769/the-electronic-century-part-iv-the-seeds-of-the-future/145415

limited live real-time control of the studio oriented DAWs available.. More commonly, electronic musicians would use their laptop as a limited part of a larger musical system (usually involving digital hardware, and/or effects processors).[38] In 2001, Berlin based Bernt Roggendorf and Gerhard Behles sought to change all of that with their live electronic performance oriented DAW, Ableton Live. Unlike Protools' recording and Cubase/Logic's MIDI beginnings, Ableton Live found its beginnings somewhere in between, as an audio loop handler/trigger/manipulator, setting itself apart with its intuitive workflow and user interface.[39] By 2004, Ableton not only incorporated MIDI, but also made almost every parameter in its graphical user interface (GUI) MIDI mappable. This infinitely increased its input and output potential, allowing for MIDI effects and control over internal and third party soft synths, as well as assignable control over the whole program by any controller that could communicate via MIDI. Synchronizing, or using Live as a host or slave in conjunction with other software is also quite intuitive. These and plenty of other features have evolved Ableton Live to become the leader, with what could seem like a monopoly over live performance software, which, to an extent affirms the common theme throughout this work: more control amounts to more improvisational ability, which makes a performance more "live".

Though using Ableton Live can solve most of the complex issues that can arise out of concocting a new musical system for performance, one can still run into some limitations. Of course, Ableton is unable to satisfy every electronic musician and their wish list of features. Instead, they have recently offered ways of "hacking" into their program for further customization. With a feature called remote scripting, Ableton allows users/companies to include scripts of code (written in Python scripting language) that will automatically provide functionality for their controller as soon as it is plugged in. Users can concoct a whole system unto their own of automatically mapped functionality. In a recent partnership with Cycling 74 (Max/Msp) to make Max For Live, Ableton has also allowed users to build their own plugins right inside the program for effects, synths, sequencers, routing schemes,[40] as well as internal control over Ableton's robust mapping, essentially providing another back door. Similar to–maybe even influenced by–the Monome aesthetic, by the late 2000s, Ableton Live and its programmable "back doors," allowed users to jump in and design their live performance system

---

[38] This is still common today, only currently more out of choice rather than necessity as before.
[39] http://www.musicradar.com/us/tuition/tech/a-brief-history-of-ableton-live-357837
[40] Very similar to the concept of Reaktor when used as a VST

at whatever level, from right-out-of-the-box beginner, all the way to intricately customized advanced programmer-performer-improviser.

2.1.1.8 Community Based Tools

We have currently reached an age in music where the performer-composer is hardly limited by the physical confines of his instrument; where musical personality emanates not only from the notes performed or composed, but also from the musical systems individually constructed. Audio programming environments are now abstracted at high enough levels to not require a computer science degree to operate (though it doesn't hurt). As Ge Wang puts it, "the homebrew aesthetic has encouraged personal empowerment and artistic independence from established traditions and trends" (Wang and Adviser-Cook 2008, 29). Using these types of high level environments, many of these sort of musician-computer programmer half-breeds who encourage advancement and proliferation through collaborative open source contributions, have shared a wide variety of tools for live performance, and the improvisation there of. The following are a list of a few interesting improvisationally minded user-contributed software tools.

Probably the most advanced and unique recent performance software tool, which constitutes as an entirely new complete performance system unto its own, is MLR, written in Max/Msp by Brian Crabtree as the first program to accompany the Monome. It exploits buffer manipulation[41]–one of a computer's many superpowers–by making a multi-voiced sequencer out of (multiple) pre-recorded, or live-recorded audio loops. Due to that MLR only deals with recorded audio loops (as opposed to MIDI triggered synths), the program by nature also exploits a musical trend of exploring ways to make new music out of old music (not necessarily oldies or classical, but pre-recorded music).[42] The high degree of improvisational freedom over note-level, and more so, score level control that MLR offers is an ideal solution to many issues an innovative solo electronic improviser must tackle (though, as discussed later, it presents some more problems as well).

One of the more groundbreaking digital audio tools to enhance effect-level type improvisation was built in Reaktor by Tim Exile, The Finger. Everyone who likes to tinker with added audio effects in their signal chain (be it hardware or virtual) has probably discovered the

[41] The processing and manipulating of an audio file, often in reference to how and where it is being played back

[42] –A trend that extends back from the first experiments with Music Concrete-style recorded audio buffer manipulation.

exciting revelation of what new different sounds emanate when the effects are reordered in different combinations. The Finger's purpose is essentially to be able to route, order, and build an effects chain on-the-fly, quickly and easily. Using (only) provided effects internal to the plugin, one is able to assign each effect to a button; the order in which each button is held comprises the order of the effects chain, allowing for seemingly endless on-the-fly combinations of potential effect-level improvisation.

In addition to their hardware and performance contributions, Flip Mu[43] has contributed a hefty bundle of Reaktor and JUCE based tools for the live computer musician, which they openly share with the world. Their tools run the gamut of type-of-control (note-level/effect-level/score-level), as in an audio chopper, shuffler, wow/flutter, Monome MIDI handler, audio/control data analyzer, machine learning based autonomous agents, i.e., virtual co-performers (Hochenbaum 2013; Vallis 2013).[44] Another fellow Monome enthusiast, Raymond Weitekamp, has invented a Java program called, Smacktop, which harnesses the accelerometer of one's laptop and turns each of it's the laptop's corners into a smackable MIDI control.

Taking advantage of Ableton Live's remote scripting in one of the most practical and creative fashions are programs like LiveOSC, and Clyphx. These programs are not coded to automatically map a controller as remote scripts were originally intended. Instead, they make all of Ableton's "back door" functionality extremely easy to access and use. LiveOSC is a way of accessing these commands through OSC (a digital communication advancement from MIDI), essentially rendering Ableton Live an OSC compatible DAW. Clyphx does the same thing, except all commands are controlled through creating, titling, and launching Ableton Live "dummy" clips, for in-the-box musical system re-programming. This in particular can be extremely advantageous for live performance because the command-filled clips are already organized and quantized within the flow of Ableton's musical system, requiring no additional external programs.

The Performer Angle

We have now briefly seen how the technology of live computer music has progressed to present day, in light of improvisational performance. We have seen how the need for spontaneous controllability has been a crucial factor in the development of Music Technology. It is now time

[43] Owen Vallis and Jordan Hochenbaum; http://flipmu.com
[44] http://flipmu.com

to pay some attention to how that technology has been used live, when used for the purpose of improvisation. This section will briefly cover some improvisational-minded electronic performers and their approaches. It of course is not an exhaustive taxonomy of every approach and performer possible, but an observation of some of the more obvious and common modern practices that are still utilized today, further addressed throughout this work. There is much overlap in the techniques utilized, as differing techniques amount to different "musical flavors" that go well with one another.

2.1.1.9 The Band Approach

The band approach involves improvisation amongst a group, where each musician is collaborating on their own instrument. Of the earlier improvisationally driven bands that incorporated electronics, like The Grateful Dead, there were definitely advancements to improvisation within the effect-level, but for the most part these artists followed the improvisational rules laid out for them by their predecessors of classical times. As music became more recently digital, the band approach is still in use, but with more powerful instruments, like sequencers, and samplers, or computer versions thereof, often utilizing some type of tempo syncing mechanism between them. These modern digital Band Approach bands are able to dwell in all three levels of improvisation more equally. Some worth investigating include Monolake, Magic Mountain High, Flip Mu, Glitch Mob, and Meat Beat Manifesto.

2.1.1.10 The Shared Instrument Approach

The more computers enter the picture, the more examples of shared instrument-based collaborative approaches emerge. This approach can be easily summed up as two or more users controlling the same output(s). The shared instrument approach is not impossible without computers, but is very uncommon, considering how intrinsically the input and output of acoustic instruments are tied. The piano (and other keyboard instruments) is one of the rare few acoustic instruments that can easily accommodate the role of a shared instrument. Another quirky acoustic example is Figure 10, showing an example of a band, Walk off The Earth, uncommonly using a common guitar as a 5-way shared instrument. Improvisation within computer-networked based shared instruments can get interesting, as there is a blending of two separate input sources controlled by different users.

Figure 10. Acoustic Guitar as a 5-Way Shared Instrument

The earliest computer band, The League of Automatic Music Composers, introduced the possibility of a shared instrument approach with computers, creating performance methods that still seem advanced today. The League took advantage of the first publicly available and affordable microprocessors, creating a network between them; each was able to play his own instrument, as well as send control data to play certain parameters of each other's instruments (Figure 11 shows an example of their network routing). John Bischoff, a leading member of the League of Automatic Composers describes the networking of his band. "When the elements of the network are not connected the music sounds like three completely independent processes, but when they are interconnected the music seems to present a 'mindlike' aspect" (Bischoff, Gold, and Horton 1978). Every League performance was a product of collaborative improvisation for an unexpected outcome.

The League and successors, The Hub, have served as inspirations for many networked computer groups to come; one academically oriented branch includes PLOrk, SLOrk, and the Machine Orchestra. These ensembles are unique in that they are each a product of a university, and so double as classes for students. The first of which, PLOrk (Princeton Laptop Orchestra), founded in 2005 by Dan Trueman and Perry Cook, explored improvisation in large networked groups (Trueman et al. 2006). Next was SLOrk, created by Ge Wang as the official laptop orchestra of Stanford University;[45] These eventually gave way to a number of laptop orchestras cropping up throughout the world (Oxford Laptop Orchestra [OxLOrk], Boulder Laptop Orchestra [BLOrk], City Laptop Orchestra [CLOrk], Tokyo Laptop Orchestra [TOrk]) (Kapur

[45] Not that there were any rivaling unofficial Stanford Laptop Orchestras.

et al. 2010). The third notable pedagogically based laptop orchestra is the KarmetiK Machine Orchestra, by Ajay Kapur and Michael Darling at California Institute of The Arts.[46] The Machine Orchestra makes the shared instrument concept further abundantly clear with robotics by allowing performers to collectively improvise with electromechanical (mostly percussive) instruments built by Kapur, Darling, and their students. For further investigation into these and other LOrks, please take a look at (Wang and Adviser-Cook 2008; Kapur et al. 2010; Vallis 2013; Kapur et al. 2011; Kapur and Darling 2010).

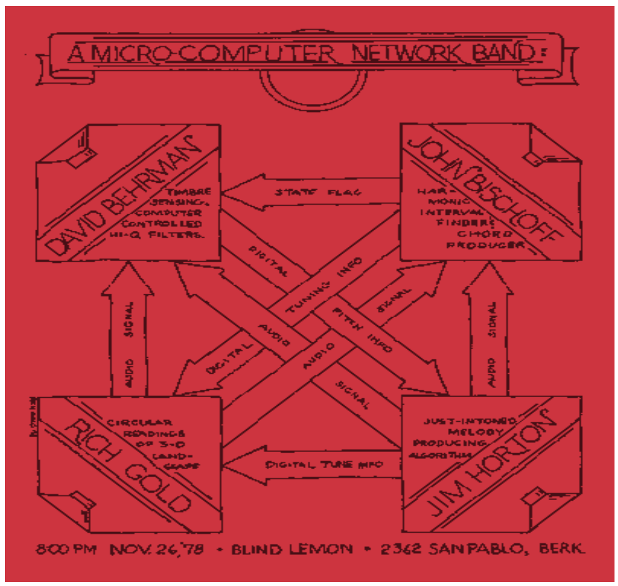


Figure 11. League Of Automatic Composers flyer/Map of Networking

Figure 12. PLOrk (left); Machine Orchestra (right)

[46] http://www.karmetik.com/artist/machine-orchestra

Shared instruments also crop up in modern pop and dance music, one example being the electro-pop group, Soulwax. Soulwax makes electronic dance music, which they perform at festivals and raves, but as a live band with drums, bass, vocals, synths, an MPC, and of course computer driven tracks and MIDI sequences. It would seem at surface value that they would fit in the band approach category (which they do), but, as they acknowledge in an interview with Future Music Magazine,[47] "Ableton runs everything."[48] The front man of the band, Stephen Dewaele is a man of many hats; he is the singer, the synthesist, and with the use of his computer, is networked with every other band member, as well as with the band's lighting system, rendering him also the band's improvisational conductor. Through Ableton Live, Dewaele controls the arrangements of Soulwax's songs spontaneously by sending MIDI sequences or program changes to each performer's instrument rig, improvisationally at will, creating a collection a shared instruments at his and his bandmates' fingertips. Being an electronic live band, they value their ability to be improvisational. The incorporation of a computer (networked to the rest of the band) into their setup allows for a score-level improvisation that would be rendered incapable by a computer-less band. David elaborates, "The cool thing is there is a tight order when you look at things in detail, but in the grand scheme of things, actually my brother can decide if we do twenty minutes of the same thing, or two seconds."[49]

2.1.1.11 The All Inclusive Approach

"I'd rather [that I am] having too many choices and sometimes losing control, than having too little choices and being bored by myself and feeling this translates to the audience"

Robert Henke[50]

The All Inclusive technique involves a setup that contains more instruments (or at least control) than can be played by the performer at one time. This approach is well suited for improvisation, as the artist is free to then rotate throughout whatever gear available seems as though it should be controlled at that moment; it becomes even more applicable and versatile in the digital

47 http://www.musicradar.com/futuremusic/
48 http://www.youtube.com/watch?v=FbFXxKGeD0E
49 http://www.youtube.com/watch?v=FbFXxKGeD0E
50 http://createdigitalmusic.com/2009/07/video-interview-atom-by-robert-henke-christoph-bauder-musical-balloon-sculpture/

domain in that instruments not being operated upon at that particular moment can be set to run automatically in simultaneity.

Original electronic artists, Simian Mobile Disco (SMD) exhibit tendencies akin to a shared instrument, but their improvisational prowess more so lies in their musical All Inclusive approach. They set up their plethora of gear in an island, so they can encircle it easily. Their main setup consists of three analog synths, all heavily tweaked during the course of each performance, as well as a Roland 909 for drums, two iPads and a Novation Launchpad for trigger patterns in Ableton which are sent to the synths, some Ableton clip launching for necessary audio tracks like vocals, all of which sent to a mixer for live mixing and remixing.[51] Whether they feel it is their duty as electronic musicians, or they just have an inkling toward keeping things fresh, new, and exciting for themselves from night to night, like Soulwax, Simian Mobile Disco has put much thought into how to make (what usually has a tendency to be monotonous) dancy club type electronic music as improvisational as they can. The band alludes to this in many comments given during an interview with the Creators Project,[52] stating,

> Our live setup is quite flexible and quite different each night, (and is) exactly halfway in between DJing and playing live. (…) The live versions have come to sound quite different from the recorded versions. (…) We record patterns on-the-fly, so we can put in whatever we think is right for that moment. (…) Although we have an idea of what tracks we're gonna play, the transitions between them and how long we play them for is pretty much up for grabs. Some nights it goes well, and we play them for a long time, and some times not so well.

In reference to their massive amount of mostly analog gear, they acknowledge this approach's one potential flaw (which they turn into a positive), "Pretty much every night something goes wrong. And part of what is interesting is letting it go wrong, and following it, rather than worrying about it and trying to make it exactly like it was the previous night."[53] In addition to improvisation being fun and fresh for them and their fans, its almost as if Simian Mobile Disco is saying that it is easier for them to improvise than not!

Artist, Four Tet exemplifies a solo improvisational electronic artist's attempt at the All Inclusive approach. By having a lot of gear and control at his disposal, a Four Tet performance is a prime example of spontaneously creating new music out of pre-existing music as an

[51] http://thequietus.com/articles/12640-simian-mobile-disco-interview-live-dance-music
[52] http://thecreatorsproject.vice.com/en_us
[53] All SMD quotes were gathered from: http://www.youtube.com/watch?v=FwMjr5m80fo

improvisational performance technique, a trend mentioned briefly throughout this chapter. He is one of the more improvisational live computer musicians in that he does not plan out his sets before he plays; instead, his setup is comprised of building blocks of seemingly infinite choices for control and manipulation. His setup is more computer-friendly than SMD. Everything is routed down to a four channel DJ mixer, with ample room for control and manipulation within each channel. While each channel can be thought of to constitute the efforts of one band member (of which he can control all or any at once), the channels are not simply divided into drums, bass, melodies, and so fourth. Instead, his channels are divided into essentially different techniques, one for Ableton clip launching, one for rhythmic Monome sequencing, and two for different live looping approaches. Four Tet realizes his technique's uniqueness in stating, "I really like this, because its not a plugin; its not a synthesizer; its not an instrument or anything, but its so live; its never ever gonna be remotely the same at any show."[54]

Figure 13. Simian Mobile Disco's extensive live rig

2.1.1.12 Clip Launching/Clip Manipulating Approach

If we relegate Clip Launching as merely an advanced version of playing audio tracks, it ties back to one of the earliest electronic musical approaches, DJing, which actually originated in the 1930s -1940s.[55] The first DJs later to use their rig as an improvisational instrument are called turntablists, as they used vinyl turntables to rhythmically scratch records, "beat juggle" and "beat match" between usually two different turntables that are playing songs in different tempos/keys. Because DJs typically have only finished songs at their disposal, it is one the first prime examples of an approach for score-level improvisation in electronic music. It is also probably the first example of blurring the lines of using score-level techniques to actuate note-level improvisation

[54] http://createdigitalmusic.com/2013/05/four-tet-walks-through-his-unique-live-rig-for-red-bull-music-academy-video/

[55] For a decent brief overview, consult: http://www.radiosolution.info/newsletter/website_pages/history_of_DJ.html

by using finished songs to scratch a rhythmic or even melodic sequence. Because it comes from a live effort of beat matching, and basing the decision of what to play next on that live reciprocating audience/artist energy many call vibe, DJing comes from improvisational roots. Some of the more innovative improvisationally minded DJs of fairly recent times include DJ Q-Bert, DJ Shadow, DJ Nu-Mark, Mix Maser Mike, Cut Chemist, Boys Noize, Justice, Busy P.

Clip launching in its more modern, computer-driven sense usually incorporates advanced tempo quantizing and sequencing, as well as buffer manipulation. It was, and still is the approach given most credence by live performance software, Ableton Live by having the majority of one of their two main windows devoted to it. Every live set from the prominent monomer, Daedelus is a non-stop clip manipulating rollercoaster of improvisation. His performance material tends to be an equal blend of his own songs and others', all of which he combines in new ways, shuffles around, and makes new loops out of. Daedelus is a key figure in regards to pushing the boundaries of and showing the potential for improvisation in contemporary live computer music: (1) His ability to make new music out of the old by strictly improvising with score-level clip manipulation utilizes a technique previously impossible without a computer; (2) he was among the first to perform on a grid controller like a Monome, and among the first to perform with accompanying software, MLR, and to date has become somewhat of a virtuoso of his instrument/performance system; (3) and because of his original performance rig, he is a pioneer of his unique style of performance, which blends elements of DJing, original performance, live/looping, remixing, mashups, as well as incorporates a wide range of musical genres throughout each set, almost exclusively through clip manipulation.

Seeing as how their audio tools are geared for live performance (Arduinome, Chronome, Freeze Delay, Shuffler, Chopper, etc.) it is fitting that Flip Mu (Owen Vallis, Jordan Hochenbaum) would also double as an electronic performance duo. Their setup involves mostly clip launching in Ableton, but by manipulating their score-level material (clips) with their effects, they are able to produce a lot of interesting note-level type improvisation. Nosaj Thing is another electronic artist that uses a similar approach to Flip Mu, but with an MPD[56] controller and commercial effects.

[56] http://www.akaipro.com/product/mpd241

2.1.1.13 Live Looping Approach

Live looping in performance has a rich history beginning with tape music[57] in experimental Western Art Music of the late sixties, with composers such as Edgard Varese, Karlheinz Stockhausen, Pauline Oliveros, Terry Riley, Steve Reich, and plenty more, who used reel-to-reel tape machines to perform with loops of material.[58] [59] Live looping is just as it sounds: a method consisting of recording some aspect of a performance live and playing, or, looping it back. Usually more performance is then executed on top of these looping tracks, often in an iterative re-looping process where loops are stacked and play simultaneously.

Tonal and beat based versions of this approach tend to be utilized by smaller groups and solo artists to allow them to build up an impromptu looped rhythm section from scratch to then perform their more solo note-level type material over. Using digital pedals designed for this specific purpose, Reggie Watts, and Dub Fx are but two masters of building complex music with live looping. Both essentially emulate instruments with their mouth, which they loop in order to sing over, or solo over with their mouth, emulating another instrument.

A more computer-friendly master of creating full arrangements through live looping is Tim Exile. Exile dwells in every level of improvisational performance, and belongs in most of the other approaches listed in this section, namely the All Inclusive Approach (just as Four Tet and Daedelus also belong here in live looping). His entirely computer-based setup is exponentially more massively encompassing and elaborate than the average electronic musician; so much so, that he has to place controllers on top of controllers to fit everything on one table[60]

[57] Looping actually started with records, but these were generally for recording purposes as opposed to performance.
[58] http://www.loopers-delight.com/history/Loophist.html
[59] For an in depth look into tape music's influence over music technology please read (Mee, Daniel, and Clark 1999) and Joel Chadabe's article in Electronic Musician: The Electronic Century Part II: Tales of the Tape (http://www.emusician.com/gear/0769/the-electronic-century-part-ii-tales-of-the-tape/140461)
[60] With more modern, smarter controllers than his bulky Behringer B-Control Series controllers he would be able to fit everything more compactly on his table. The fact that he chooses to stick with outdated clunky controllers implies that he has spent a lot of time practicing with this particular setup, and prefers to stick with what he is used to, to perform at the best of his ability (although contractual spokesperson type obligations could also be a factor).

(see Figure 14). That said, this much improvisational control could not be possible in such a compact setup without the aid of computers.[61]

Figure 14. Tim Exile's live rig

Generally, much of a live looper's performance is rehearsed and executed at least similarly from show to show, so exactly how much of it is improvised is hard to say. Something about the building of loops from nothing in the effusion of fancy teeters a similar line to the spontaneity of improvised music, and is at least very well suited for the potential incorporation of improvisation. An interesting twist possible with live looping and computers is the looping of control data, as opposed to audio. As already mentioned previously, Ajay Kapur discusses the recording and playing back of control data of his North Indian extended instruments (Kapur et al. 2003; Kapur, Lazier, et al. 2004; Kapur et al. 2010). Daedelus is as well a proponent for live looping of control data; as a function of MLR, there exist "pattern records" that loop button press performance data over a designated amount of time which–as will be discussed further later on–can offer advantages over mere audio looping. The live looping of control data is yet an underexplored improvisational tactic full of potential.

2.1.1.14 Note Playing Approach

Like the band approach, the note playing approach traces it origins to acoustic music. It refers to that one-to-one correlation of using a controller like a normal acoustic instrument. As the playing of notes has been scrutinized documented and analyzed since the dawn of music, there is no need to delve into it in detail for this work; it is only pertinent to mention that note playing is a fundamental, practical and useful tool for all improvisers of music, acoustic, electronic, digital,

[61] As all of his self-programmed software is done in Reaktor, Exile has made an informative video on behalf of Native Instruments explaining and demonstrating his setup that is worth seeing at: http://www.youtube.com/watch?v=9r38r3BIgew.

or otherwise. Nevertheless, what kind of notes that can be played can surely be expanded via digital means.

Some electronic artists have taken improvisational note playing further than could be done without computer driven or digital instruments. Out of a sort of progression from turntablism, and a progression of technology, with clever mapping strategies and a lot of practice time, a few MPC virtuosi of the 1990s including Gel and DJ shadow, among others, were able to play multiple individual parts of full song arrangements on button pads simultaneously, stepping through full arrangements in real-time, beat for beat, note for note. More recently, with multiple Native Maschine controllers and a computer, electronic performer, Jeremy Ellis has continued the live electronic note-level one-man-band tradition ever more astoundingly into the new millennium.

2.1.1.15 Other Approaches

Some other live approaches that can get highly improvisational, but do not necessarily pertain to this work are live coding, and autonomous agents, or machine learning. If the coding of musical software comprises or represents the musical system one uses or has created, then live coding–called "Art Oriented Programming" by prominent live coder, Alex McLean–represents an ability to control and modify a whole musical system with just a few lines of code. For notable examples of improvisationally based live coding, check Ge Wang, Nic Collins, Slub, and Alex McLean. Improvisationally speaking, autonomous agents are virtual performers with which to improvise, that can essentially "listen, learn, lead and react." This type of performance has been utilized in music as early as the late 1970s with George Lewis (Sergi Jordà 2005, 65), but has undergone hefty advancements since with the work of (Kapur 2008) (Collins 2006)(Vallis 2013).

Not every live improvisational approach has been discussed, but (hopefully more than) enough to show that there are many new ways to tackle improvising in today's digital age. Nowadays, it can be equally if not more intriguing, interesting, and beneficial for a musician to consume time normally spent practicing, now designing his performance system with software hardware tools, clever mapping, and various techniques. Part of what it means to be a digital improviser today is how to create and personalize a musical system. The other part is how to achieve mastery over it.

The Audio-Visual Angle

Every musical performance has a visual component. It cannot be circumvented or avoided. Every bit of data perceived by the eyes during a musical performance contributes to a visual aesthetic, be it the clothes artists wear; whether they dance, bob their head or remain frozen; whether they stand or sit or hang upside down (as in Flea of the Red Hot Chili Peppers); whether they smile or intensely glare or shoegaze; progressing with technology, whether their set contains projections, or moving/blinking lights of any sort; and most recently with computers, even how these modern visuals are controlled, be it pre-programed, audio reactive, live improvised (by a lighting tech, or the performers themselves). Like the sonic equivalent of Cage's piece, *4:33*, even the absence of anything visually perceivable–such as electronic duo Autechre turning out all lights during their performance[62]–constitutes a visual aesthetic. The lights from Monome buttons in addition to constructive visual feedback provide excellent visual stimulus for an audience during a performance–especially when used with sequencing type software, like MLR–as the audience sees some relationship between the moving lights and the sound, which makes the audiovisual performance more mesmerizing. Ajay Kapur's percussive robots in The Machine Orchestra are another great example of the audio and visual element intertwining, as digital computer music becomes more embodied through an electromechanical physical entity. And with digital technology it is now within the scope of modern performance for generated or light-based visuals to be improvised upon along with music. Needless to note, the audiovisual angle is a broad one, which could comprise volumes of research unto its own. Although at first glance it might appear tangential to be analyzing visuals in a highly music-centric document, instead, please understand this work as an exploration into improvisation of musical performance, which more so over time includes a visual component. As this thesis will demonstrate, improvising with visuals has many similarities to that of audio. Here, we will examine a mere fraction of audio-visual relationships and the visual element as it applies to live light control or visual imagery as a performance, as well as a brief history into where these ideas came from.

[62] http://www.theguardian.com/music/2005/apr/18/electronicmusic

2.1.1.16 Early Electronic Audiovisual Performance

"You could actually compose for the room, as if it were an instrument. You could play it like an instrument or you could compose pre-programed shows," exclaims David Rosenboom, at the time one of the resident DJs and assistants to one of the earliest multimedia sound and image social environments, The Electric Circus (Gluck 2009, 2). Started in 1967 in NYC, the Electric Circus was an audiovisual art discotheque with flame throwers, jugglers, tight rope walkers, big rock bands (The Velvet Underground, Sly and the Family Stone, Deep Purple, Allman Brothers, etc.), live and DJed dance music, audiovisual art pieces (John Cage, Alvin Lucier, Pauline Oliveros, Morton Subotnick, David Rosenboom, etc.), and a whole visual environment of carousel, film and overhead projectors, as well as projected liquid and light manipulation[63](Gluck 2009).

Visual performance art using light as a medium–called Light Shows–grew exceptionally popular in the late 1960s and onward. Among the many light show artists of the time (Mark Boyle, Mike Leonard, Brotherhood Of Light, Little Princess 109, etc.), live visual artist residents of the Fillmore West from 1968-1971,[64] The Joshua Light Show, embodied most of the light producing techniques of the day. Their performances back then involved improvisational visual accompaniment for top rock bands (The Who, The Doors, Jimi Hendrix, Janis Joplin, etc.), using projection of colored light through various devices, liquid projection of dyes glycerins and oils, hundreds of color wheels, motorized reflectors to shape and direct light,[65] all rear projected onto a 30-40 foot canvas[66] (for a look at their live show and light manipulating materials, see Figure 15). The Joshua Light Show, still performing today, is especially pertinent to this work, as they are performers who have always placed an emphasis on improvisation, and over time adapted to new technology, adding digital imagery and modern projection equipment. Joshua White (creator) describes the then and now in an interview for gothamist.com.

---

[63] http://bedfordandbowery.com/2013/09/a-look-back-at-the-electric-circus-the-greatest-show-on-st-marks-place/

[64] [65] http://www.joshualightshow.com/about-classic/joshua-light-show-1967-68

[66] http://gothamist.com/2007/04/02/interview_joshu.php

> The show at the Fillmore was perfect for its time and I don't wish to recreate it. I'm interested in regenerating some of the visual techniques and forms that I thought were very important when I was doing it back then and that I haven't had a great deal of opportunity to do in the present. One of those is improvisation. We had a knowledge of their music but we weren't following any score, we were just improvising, making something visual using our tools. So improvisation is the key. So what we will do at The Kitchen is the same idea with different musicians and different music. Bec Stupak, my collaborator, and I will improvise. That's number one.
>
> Number two is adapting new technology.
>
> (Joshua White, 2007)[67]

Figure 15. Joshua Light Show live (left, center); materials (right)

2.1.1.17 Visual Performance Software

The history of digital visual software followed a similar path as that of digital audio. A protocol, called DMX–almost exactly like MIDI (but each note/ID with double the continuous control range)–was created as a standard to operate digitally controlled live lighting. Visual programming software evolved both in text based form with pedagogical free programs like Processing[68] (in Java) and libraries like OpenFameworks[69] for C++, and graphically with programs Quartz Composer and Max/MSP's eventual add-on, Jitter. Given the history of digital visual development being so similar to that of the audio world, it is likely predictable that the next inevitable step for visual control would be some kind of digital video workstation for live improvisational performance. A few, like Resolume,[70] and VDMX,[71] have followed closely in

[67] http://gothamist.com/2007/04/02/interview_joshu.php
[68] http://www.processing.org/
[69] http://www.openframeworks.cc/
[70] http://resolume.com/
[71] http://vidvox.net/

workflow and structure to Ableton Live,[72] created specifically to cater to an audiovisual market. A wide variety of effects are provided, allowing for plenty of improvisational variation over even very basic or minimal source material. Like Ableton, Resolume is made to receive MIDI, OSC data, and sync with other programs and controllers, and most of the features in the GUI are MIDI/OSC mappable. It also acts as a video mixer, capable of splitting up the output image to be mapped to conform to a 3D surface (or multiple 2D surfaces), called projection mapping. Programs like Resolume and VDMX and their predecessors have provided a means for the live audiovisual improviser to possess more complete control over his craft on all levels of improvisation (note/effect/score); the performer-composer-x-x- can now easily be the performer-composer-x-x-visual artist!

2.1.1.18 Reactive Visuals

Reactive visuals are another grey area in the continuum of improvisation as they are generally spontaneously generated with intentionality, however, the intentionality behind their generation is a preset group of instructions actually controlled through audio (or some other kind of) analysis, as opposed to being manipulated live. By this description, with reactive visuals it is justifiable to determine that if the audio is being improvised, then so is the visual element.

Using information gathered from waveforms through an oscilloscope, the live electronic computer duo, Cyclo (Ryoji Ikeda and Carsten Nicolai), has created a reactive audiovisual set so visually captivating that their reactive projected visuals almost overshadow their audio that they are controlling live. Cyclo has been obsessed with visually realizing audio data by running their stereo output through an oscilloscope to generate beautiful patterns called, Lissajous curves.[73] Ryoji Ikeda elaborates in an interview from Museum of Modern Art (MOMA), "We now have a great number of sounds that we use as our artistic alphabet, but the musical grammar we decided to take for the performance is more accessibly groovy rather than statically academic."[74] By archiving a database of waveforms based out of their naturally correlated beautiful visual

---

[72] Resolume more so than VDMX.

[73] In a gross oversimplification, Lissajous curves are the visual representation of two waveforms stacked orthogonally on an X,Y axis of a polar coordinate system. The phase, frequency and amplitude all factor to crate beautiful geometric patterns. For a more mathematical comprehension over Lissajous curves, refer to (Merino 2003).

[74] http://www.moma.org/explore/inside_out/2013/10/01/an-interview-with-cyclo-ryoji-ikeda-and-carsten-nicolai/

representation, Cyclo's audio is reactionary to visual stimulation in their compositional process, but then for performance flip around the roles: improvising music, which then generates their beautifully reactive visuals.

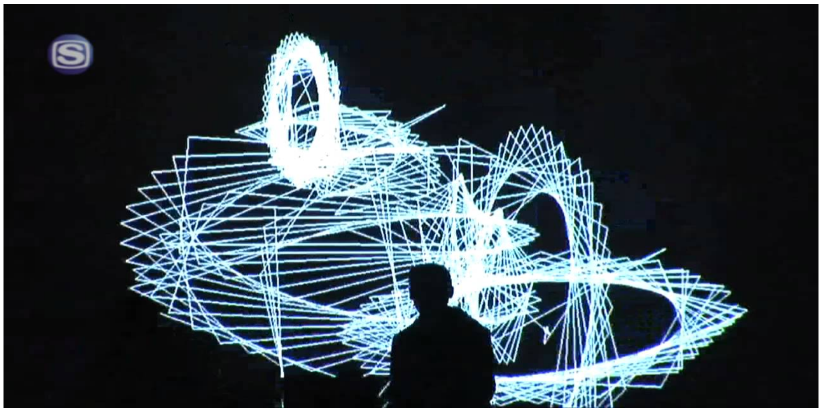


Figure 16. Cyclo Live Visuals

Chapter Summary

Improvisation has a long, rich, and vibrant past, arguably tracing its origins to the beginnings of music whatsoever. Interestingly, Music Technology, in big part appears to develop out of a drive for more improvisational control within musical execution. Early Western Art Music even as far back as Baroque and Classical seem to have embodied the same musically improvisational approaches as that of recent improvisational styles like Jazz and Blues, using solos, ornamentation, and the like. This is an indication that, although music has definitely evolved over time, the rules of improvisation were laid early on and have been stuck with ever since (see Figure 3) until computers entered the musical picture. Over a few decades of their development, computers demonstrated that when used with music, much of the rules and constraints placed upon music, especially upon improvisational musical performance, were no longer present. Complex routing of a musical system, the ability to formulate sound out of programmatic and algorithmic means instead of being bound by the laws of acoustics, and probably the biggest turning point for modern digital performers, the decoupled nature of controllers, enabling users to specify their own improvisational system. These were all superpowers offered by the computer that were previously unattainable. Ever since, the improvisational methods and techniques of yesteryear are still completely intact, but expanded upon, enhanced, and evolved.

With computers we have reached and age in music where performers with a mere technical curiosity can be designers of their own musical controllers and/or software.

Conversely, very technical minded individuals with a need to tickle their creativity can create and cater their musical systems all to their own, supplementing any lacking musical prowess with electronic and digital slight of hand (or slight of electronics, rather). Thanks to the doors that computer music has opened, musicians have developed many intriguing capabilities and personalizable approaches to their improvisational performance, e.g., virtual sharing of instruments, ways to jam pack an overwhelming amount of functionality into a performance setup, and live looping of data to name a few. These kinds of approaches are all products of a goal to technologically push further what is possible with music further; to make inhumanly intricate or complex tasks performable–even improvisable.

There is and always has been a visual component to every musical performance. It has been recently with the development of computer based audio-visual systems that a visual component has been more plausibly performable. It was the mid-20th century when the concept of "improvising with visuals" in relation or accompaniment to music began to thrive, with various types of light shows. This has ultimately evolved (similarly to music) into digitally controlled light shows, and digitally manipulated visuals, brining us now to a forefront of complete improvisational audio-visual improvisational potential.

Chapter 3

Mapping Strategies

A digital musical system's mapping ability is essentially the prime exploitable factor that separates it from music that came before it. Before electronics in music, though the instrument luthiers of the past had some level of command over their input control source and sonic output, the 'mapping' between the two were generally always subject to the laws of physics of acoustics. This capability of decoupling in recent times has provided a means for complex and selective routing between any control source (input) and its chosen control (output). This notion of user-defined 'mapping' was introduced with electronic analog instruments like modular synthesizers, but its potential was more fully harnessed and commonplace with musical systems becoming digitized. A simple relationship between how inputs and outputs are tied differently among acoustic and digital instruments through mapping is shown in Figure 17; acoustic instruments generally demonstrate a holistic unified relationship, whereas electronic instruments generally exploit a complete separation between its inputs and outputs, connected via mapping.

For the improviser, thoughtful and intricate mapping within a digital system is like a futuristic laser gun compared to a physically constrained bow and arrow that is acoustic mapping. When used advantageously, complex mapping can serve a level of efficiency and complexity unrivaled by acoustic instruments, but when not fully grasped or thought through, can bring about confusion and undesired limitation. Furthermore, as customizable and functional as digital interfaces have become, there still exists plenty of room for controllers themselves to behave more functionally and efficiently, to allow significant reduction in mental bandwidth for the performer. As mapping from controller to computer–or within the computer itself–to this day is still monopolized by MIDI, this chapter is devoted to efficient, clever, and effective MIDI-based mapping strategies and tools to potentially make the computer music improviser's job more practicable, and enjoyable.

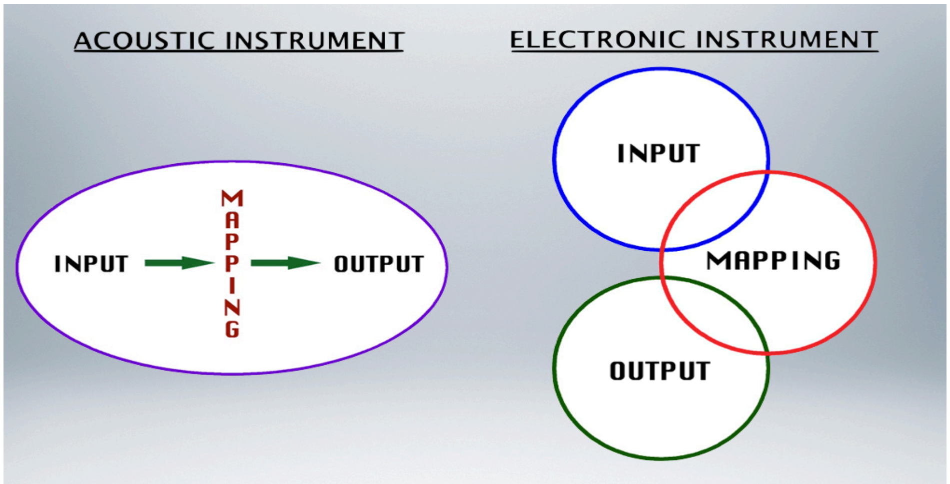


Figure 17. Input and output between different families of instruments

Mapability

Importance of Mapping

Mapping is integral to live computer music for abstracting complexities that allow a performer to improvise with more effortless ease than can be otherwise unachievable, or would require a lifetime of practice to execute manually. For instance, with the press of one mapped button, a performer can unmute multiple channels, while simultaneously triggering (launching/playing) particular clips within those channels, while also looping through an arpeggio on a MIDI controlled software synth (often called, a one-to-many mapping–Figure 18 [left]). In fact, although it would probably not be very improvisational, one button could be pressed to engage an elaborately shifting completely automated musical system, with no further control required. Similarly for example, one could map a group of buttons to shift between different chord qualities and key centers for an arpeggio of a particular soft synth (conversely called, many-to-one mapping–Figure 18 [right]). The acoustic equivalent in either example of just arpeggiating alone would require a strike, press, pluck, or some kind of energy placed into every single note arpeggiated, as compared to the one button press from digital mapping. These examples also illustrate another advantage afforded to the live computer musician out of complex mapping, in that the performer can use his input to control more macro, conditional parameters instead of (or in addition to) directly tied correlations. Of course, acoustic instruments exhibit a one-to-many mapping in that one input can control multiple output parameters at once; e.g. one strike/pluck/etc. can control the volume, pitch, and the timbre of a sound altogether. The main difference is that all of these acoustic mappings are tied to the instrument, none of which can be omitted or reconfigured as they can in a digital system. In essence, with specifically computer-driven live performance, one can take more control over their entire musical system with considerably less effort (physical and mental), then without.

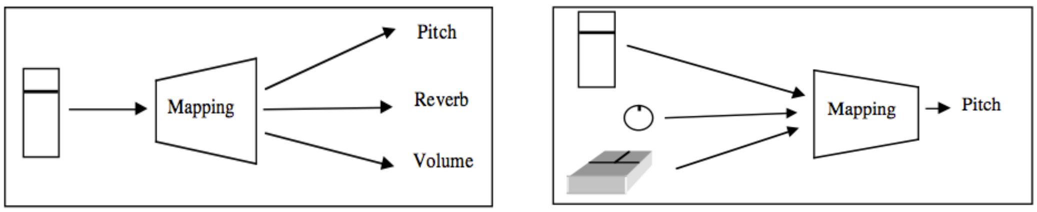


Figure 18.one-to-many mapping (left); many-to-one mapping (right) (Hunt and Kirk 2000)

The State of Things

"MIDI, we have seen, is the lingua franca in the controller's world"

(Sergi Jordà 2005, 49)

3.1.1.1 If It Ain't Broke, Don't Fix It

Almost 10 years after the above quote by Sergi Jordà, and over 30 since its origination, MIDI still reigns as the unchanged standard for musical digital communication. Other technology from that era, like vintage digital synths and drum machines of the 1980s holds a niche fondness and relevance today for its sonic value, nostalgic value, and novelty, most of the valuable gear that still proves useful for today's electronic performance has been updated, enhanced, and modernized, the rest relegated to fun collector's items. Yet MIDI, a technologically antiquated methodology, still operates as it has since its standardization. As technology has progressed, what once were MIDI's strengths, are now thought to be its weaknesses; its 128 point resolution, and fixed equally tempered tuning were but two selling points that showed how expressive and musical MIDI based electronic music could be, now only serve as limitations to musical expression in comparison to the technological potential of today.

For better or worse, the history of musical digital data transmission embodies the epitome of the phrase, "If it ain't broke, don't fix it," However, the desire for more robust musical instrument digital interfacing has not been completely swept under the rug. There have been plenty of attempts to update, upgrade, and surpass MIDI, a few of which have served to increase compatibility, organization, and musicality, ultimately enhancing improvisational ability in a live situation. Some features added to the MIDI standard in the early 1990s include a standardization of which note number comprises middle C (on a piano),[75] the incorporation of time stamping (MIDI Time Stamp) and time coding (MIDI Time Code) for ease in syncing multiple midi devices, and a MIDI Tuning Standard for a standardizing of alternate tunings beyond equally tempered. More recently, MIDI protocol has been ported to work over USB, providing ease in connecting MIDI devices to a computer. As previously noted, a newer data protocol, OSC, capable of transmitting a greater resolution and amount of messages, has been the first major valiant attempt to replace MIDI. Although many open-minded software manufacturers are increasingly beginning to understand and implement OSC's potential (or at least willing to acquiesce), OSC has remained in an unstandardized stasis, and has by no means "taken over" or rendered MIDI obsolete. By 2008, the MIDI Manufacturers Association (responsible for upholding the standardization of MIDI) was vocal about addressing the seemingly archaic issues with MIDI in today's fast paced high definition world. According to the MMA, by 2011 a backwards-compatible HD prototype protocol for HD-MIDI was established, and by 2013 private demonstrations and tests were carried out for MMA members.[76] Should this new standard stick, we could very soon be on the cusp of a revolution in digital musical control.

Chastising aside, MIDI is still useful for today's musical systems. Sixteen channels of 256 controls each with 128 discrete steps of resolution for parameters like volume, or pan, or effect amount, still go a long way for live performance. Because most controllers output MIDI, and software accepts MIDI, and even virtual MIDI is built into some computers (e.g. the IAC bus in Macintosh), in the interest of time, ease, and compatibility, it can prove still beneficial to work with MIDI rather than make painstaking attempts to circumvent it.

3.1.1.2 MIDI Mapping of Late

Of all the amenities afforded by digitizing live performance, and more specifically by digital mapping complexities, probably the most useful is ease in customization. Gear utilizing MIDI

[75] This standard is one not necessarily adhered to by all equipment manufacturers and varies among some companies.

communication did not start out as user-programmable as it has become. Early MIDI sequencers, synths, samplers and the like, though decoupled, usually output one specific MIDI message per control preset by its manufacturers, which was hardly or at least painstakingly user definable. Once controllers began to be intended more toward use in conjunction with computers in progressively personalized fashions, a need for the performer to be able to designate his controller's MIDI messages became pertinent.

The MPD is an example of an early MIDI controller made specifically and exclusively for control with a computer. It is modeled after the MPC family with button pads, and some knobs and sliders (see Nosaj Thing playing on one in Figure 19). The MPD is highly customizable in setting which sensors send what MIDI messages, which at its peak made for a fairly accommodating computer controller. The controller itself is not extremely customizable because it cannot receive MIDI information, only send. This means the LED feedback provided with each pad was not definable and was always hard-linked to its button press. This is not normally an issue until one desires more complex uncoupled functionality than a simple indication of "pressed, or not" (as will be discussed shortly).

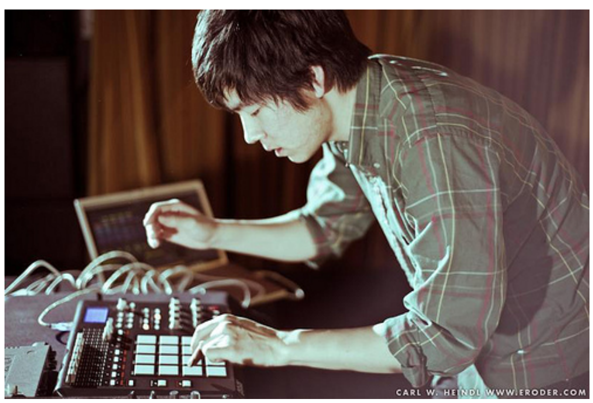


Figure 19. Nosaj Thing Playing an MPD

As the Monome grew in popularity, not only did its hardware and accompanying software help to guide performable Music Technology in a particular direction, its concept of mapping did the same. Having a minimalistic, almost initially functionless controller placed the versatility and creativity of a musical system almost entirely in the hands of the performer-programmer. For the first time with a Monome, one could have not only full control over the message data output by its sensors, but also independent control over each button's LED, creating not only a customizable highly functional button mapping potential, but also the proper visual feedback necessary to interpret it. Users could make their buttons toggle–i.e. one press sending a Note On, or high value and toggling button light on; the next press doing the opposite–, one could make button groupings of various sorts–e.g. for sequencing and radio buttons (discussed more later in this chapter)–, one could even design more complete musical systems with the Monome's buttons all set to whatever desired user-specified functionality.[77] Every improviser of electronic music has given some thought to how they would like to improvise, and most if not all arrive at their own differing conclusions. The need for customization in uncoupled visual feedback within a controller has been met with products like the Monome and most successive MIDI controllers, creating more freedom and personalization for its users, essentially allotting more potential for distinct unique personality to emanate from electronic performers' customized musical systems.

Monome's mapping concept toward future controller mapability was evolutional to say the least. To fully utilize the maximal potential of the Monome's minimalistic output, however, requires a considerable amount of programming proficiency.[78] As much as this was a useful evolution to the state of mapping in live computer music, being a computer music programmer is a quality that not all computer improvisers possess, nor should they need to, and hence can prove to be a major hurdle or limitation. Even if a computer improviser does possess enough programming ability to complete the desired task at hand, programming the proper task very often proves costly in time consumption, potentially stealing time away from efficiently building other aspects of a musical system, or practicing. Needed next were controllers that implicitly, yet *optionally* provided some of this useful complex programmed functionality embeddable into the controller itself.

To help counter the issue of needing to be a programmer to use a Monome, the helpful and active Monome community has shared a plethora of software, and basic MIDI/OSC handlers that provide many of the functions one might need a grid of buttons to accomplish. Moreover, most commercial controller manufacturers since Monome now provide a software editor with their controller that allow users to delineate exactly which sensor outputs what message. Many even allow its own light feedback to be uncoupled. In more recent times, controllers are providing more features that allot for user customization, and software is picking

[77] MLR is but one example.
[78] This definitely seems like a valiant attempt by its creator, Brian Crabtree, to garner interest in Music Technology among the Monome's users; to teach them how gratifying it is to have the knowledge and power to create and modify digital musical systems on one's own.

up the slack. Ableton live has been riding the customization bandwagon regarding mapping, in that most of its parameters within its DAW are mappable to any MIDI control source within its G.U.I. One advanced complex feature is its ability to set thresholds or ranges on the parameters mapped within the software, providing a means for highly complex routing schemes by mapping certain ranges of the same continuous MIDI control to multiple different parameters of a musical system.

Regarding controllers post-Monome, a company like Keith McMillan is making products with not only advanced sensing capability (like multidimensional XYZ pads), but with complex mapping options as well. Most of their simple button-type sensors can send out a simultaneous MIDI Note and CC message; more complex sensors like their touchpad slider sensor pads will simultaneously output one MIDI CC message for location, one for pressure, and a MIDI Note On/Off message, all from just one touch of the pad. Having simultaneous MIDI messages from sensing different aspects of one press can significantly increase as well as hone one's mapping ability, amounting to ease and extension of performance. Another advanced feature offered on most of their gear, like their QuNeo–a controller with a grab bag of differently shaped silicon buttons which output continuous control in interesting ways–is the inclusion of (optionally) dedicated buttons for bank switching. A grouping of other buttons' MIDI Notes on the device are shifted, allowing from the press of one "bank button," this grouping of buttons to control completely separate banks of parameters. For the improviser that likes to have a lot of control at his fingertips, this one-to-many complex mapping firstly simplifies the controller real-estate issue of having too many overwhelming buttons/knobs/sliders/etc. into a more compact system (unlike Tim Exile's massive setup in Figure 14), and secondly makes each grouping of bank buttons as many times more functional as the number of banks they shift between. Another complex function with the QuNeo controller is that users are able to completely switch the sensing functionality of its pads between directional messages or messages per each corner of the pad. Ample LED feedback persists throughout the QuNeo controller, which can be set to couple its MIDI message or be decoupled to be programed independently. Keith McMillan controllers demonstrate that they are a leader in understanding and implementing the importance of user-definable mapping in MIDI controllers. In the opinion of the author, they and controller manufacturers like them represent the future paradigm in controller mapability for performance, providing optional built-in advanced functionality alongside complete customization; an advancement of saving effort in programming the programmer-heavy

Monome custom that preceded. The inclusion of complex MIDI routing in this newer paradigm allows the improviser to just be the improviser–or still delve into creating complex MIDI middle-ware, but at the choice of the improviser. This digital controller evolution is simplified and highlighted in Figure 20, beginning with semi-decoupled digital hardware on the left–such as synths and drum machines–to computer controlling devices–the first, fully decoupled controllers like the MPD second from left–to full user programmability with the Monome–second from right, and currently to controllers with complete user programmability, but also pre-built with already usable complex functionality like the QuNeo on the right.

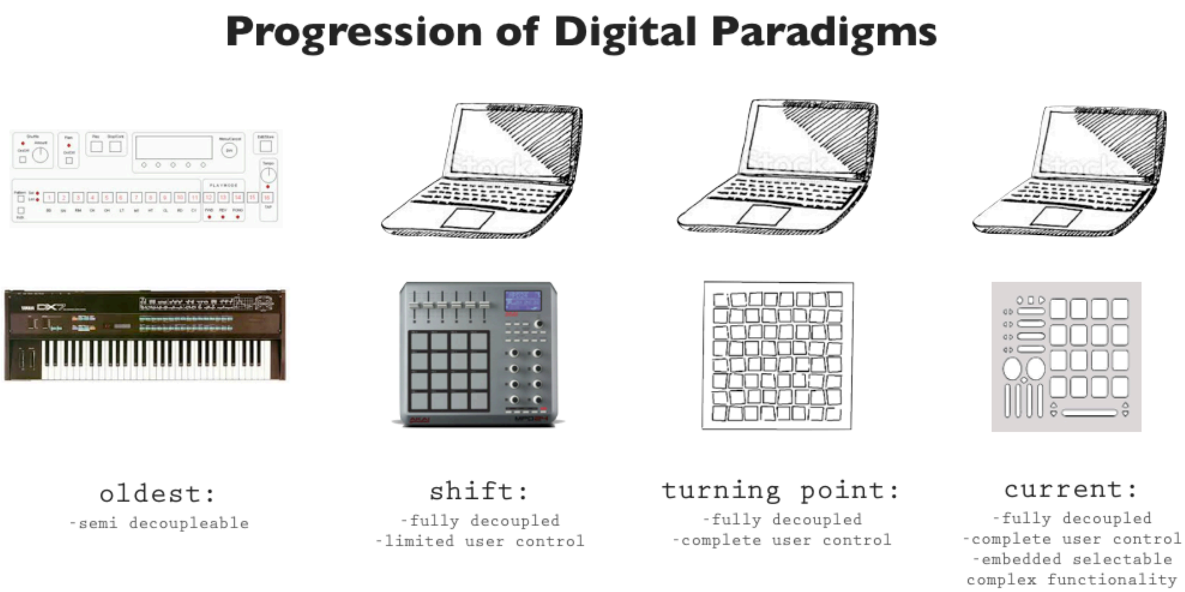


Figure 20. A Progression of digital controllers.

Considerations And Contributions

The following section delves into questions, considerations, and contributions involving complex mapping schemes for the digital improvising artist. For the purposes of this work, a non-complex mapping scheme would amount to any direct one-to-one mapping within the system, e.g., one key on a digital keyboard programed to play one note, or one digital knob or slider mapped to control one channel volume. Most everything else will be considered a complex mapping: many-to-one (e.g. multiple different keys set to play the same note); one-to-many (e.g. one slider/knob modulating four different channel volumes); and many-to-many (e.g. basically tied groupings of many-to-ones and one-to-manys).

Considerations

For complex mappings to be effective to an improviser, they need to strike a delicate balance between promoting simplicity and promoting freedom in control. Without enough simplicity, a digital system can get too overwhelming to control in its entirety, but without enough freedom in control, an improvisational performance is more likely to stagnate or grow dull.

In a study on complex mapping on different types of musical interfaces, music technologists, Hunt and Kirk propose their conclusions:

1. Real-time control *can* be enhanced by the multiparametric interface
2. Mappings that are not one-to-one are more engaging for users
3. Complex tasks may need *complex* interfaces
4. The "mouse interface" is good for simple tests and for little practice time
5. Some people prefer to think in terms of separate parameters

(Hunt and Kirk 2000)

Having been conducted almost 15 years ago, it is still relevant to considerations regarding mapping of today. The study was carried out on a "general public" for the purpose of trying to evaluate which types of control and mapping are more intuitive than others. Unfortunately, the study was not carried out on musicians (digital musicians at that), nor was it for the purpose of specifically exploring improvisation; had these been the conditions, it is likely that they would have arrived at differing conclusions. Although more recent study/documentation to suggest any new evidence-based conclusions appears to be lacking, recounting from personal performance experience and observation of other computer improvisers sheds a similar but slightly tweaked augmentation of the previous conclusions.

1. Real-time control *can* be enhanced by the multiparametric interface*, but it can also be limited if not mapped effectively.*
2. Mappings that are not one-to-one are more engaging for users *that are not well aware of what and how they are controlling; one-to-one mappings can ultimately be more engaging to an experienced improviser because of their independent control.*
3. Complex tasks may need *complex* interfaces*, but a complex task on a simple interface is often more useful than a simple task on a complex interface.*
4. The "mouse interface" is good for simple tests and for little practice time
5. Some people prefer to think in terms of separate parameters

Conclusions four and five seem to still pertain to the improvisational side of things without an addendum. As the original conclusions are for the goal of making tasks easier (for the novice), improvisers often require an awareness of the amount of control they are sacrificing by making tasks simpler.

The potential possible permutations of complex mappings within a digital system are numerous enough to appear infinite. From a general performance perspective, if we were comparing these tools to acoustic systems, the first evaluation of their worth is likely how much added complexity they offer that acoustic systems cannot? Next is a comparison of how much added complexity a particular digital system's advanced mapping/routing can offer on its output versus how much reduction in complexity it places on the input? The evaluation of these two factors alone should make a certain mapping scheme stand above others. When considering from an improvisational perspective, however, there is an added factor of, how much freedom in control is offered versus how much limitation it places on the system. This is similar to the previous criteria, the difference being that when inputs or outputs are complexly tied via mapping, some degree of limitation is implied, usually for convenience, which is advantageous in general performance, but with improvisation can result in an undesirable decreased amount of potential control.

Regarding amount of control, Sergi Jordà raises a notable point:

> How much control is humanly possible? The limitations are both motoric and cognitive. Traditional monophonic instruments frequently have three or four degrees of control. Polyphonic ones have two or three per voice while always showing limited continuous control possibilities. The reasons for these restrictions are to be found in the physical and cognitive limitations of humans and not in the inherent properties of the instruments.
>
> (Sergi Jordà 2005, 138)

Essentially, Jordà is saying the restrictions we place on instrument mapping (regardless of if they are digital or not) are imposed by our own mental and physical ability, a compelling argument for placing a cap on amount of control. The important thing to point out however is that this idea pertains to amount of control at one moment in time. An improvisational performance can last minutes, if not hours. With digital systems not all control parameters need to be wielded in simultaneity, and so more control than a human can handle at a given time is not necessarily a viable limitation or restriction to place upon a digital musical system. Where parameters can be decoupled and energy not always required for parameters to produce sound, a performer can span through an exponentially greater amount of control over the course of a performance than can be physically or cognitively controlled in one moment. Although adhering to mastery over a limited amount of control in an improvisational performance can often result in a more virtuosic performance, it is equally arguable that having more control than can be operated at a given time can lead to more potential diversity and range over the course of a performance.

Different ways to route and program complex mappings seem endless; even similar ones built by different people are tweaked just a little differently to whatever specifications. How then could any of these functions become standard as to be selectable within a commercial controller's software editor? Given that there are so many possibilities, would it even help if there were standard middleware mapping functions? Just like LED decoupled control and bank switching, which are common complex functions on most modern interfaces, there are many more useful mapping functions that prove advantageous enough to become standard over time by themselves. Moreover, this work sets about to (at least begin to) establish a quantifiable criteria for evaluating various complex mapping functions, in order to start an ongoing discussion of potential standard complex mapping tools. The degree to which a complex mapping function is useful for improvisation can be factored by:

- How much it can do that could not have been done prior.
- How much the desired added or reduced complexity outweighs the limitations it places on the system.
- How versatile of a function it is; how many different problems it can solve.

One example of a complex mapping on a (reasonably complex) push button knob sensor that would *not* be extremely conducive for improvisation would be to map the amounts of multiple effects equally to the range of the knob, and the effects' on/off to the push button. This example would create a one-to-many mapping from one knob, to multiple simultaneous parameters. Although a one-to-many mapping such as this can be quite versatile, hence, meets the third criteria, it severely limits functionality by coupling multiple effects which therefore cannot be controlled independently, and because of this relegates it more akin to the principles of an acoustic instrument which exhibits some complex many-to-many mappings that cannot be uncoupled, like a violin. This type of coupled example then fails to meet the first two criteria, and hence, is not extremely advantageous for improvisation.

Another example with the same sensor that can be equally if not more limiting in certain situations, but advantageous in others, would be inverting the mapping of one effect's On/Offs, essentially *alternating* between effects with the pushbutton (while still keeping the knob mapped to multiple parameters). This type of coupling ensures that these effects can then never be used together simultaneously, arguably limiting control more than the first example. Though it is rare, there are some instances where one would never want to use two effects at the same time, but would like the same control for them, for example, switching between an EQ that passes a band, versus an EQ that rejects it. In the case of switching EQs, this inverting-type one-to-many mapping becomes useful for improvisation. This brings about the point that any complex mapping can serve an integral function, but without thoughtful evaluation of what type of mapping serves best for freedom yet ease of control in each specific situation, improvising can easily grow increasingly cumbersome.

A third example of a complex mapping function of the same sensor that could be fairly improvisationally useful might be a 6D KnobSwitch, created by dividing a knob's continuous control range into three equal regions, and incorporating the push button to add an extra switch, or, dimension to each of the three regions. This creates a complex 6-way switch out of one sensor for ease in switching between related things. The 6D switch can, for example, be organized to mute and unmute drum tracks to form different drum groupings (e.g. kick & snare, kick & hi-hats, kick snare & hi-hats, etc.). This example sacrifices the full continuous range normally provided by a knob, but substitutes six discrete dimensions of interdependent control. This versatile interdependent grouping of control on one knob ensures an easily performable complex routing system, and saves controller real estate. In doing so, it provides a means for controlling in a way that was previously impossible, and promotes mental and cognitive performance ease, making for a great improvisational tool. Of course, again, it is only viable for certain situations, but this example is just one of numerous ways of creating MIDI middleware to get much more out of just a little.

Complex mapping strategies are key to efficient digital performance when the most advantageous mapping is chosen for the job, but the importance of un-complex direct one-to-one mappings should not be overlooked when mapping for improvisation. One-to-one mappings can be invaluable for having a plethora of potential parameter combinations because of the independence they impose over a musical system. In fact, because acoustic instruments usually involve some sort of complex coupled mapping (e.g. a hand position and bow position on a violin will control volume and timbre, and it takes two hands to play most violin pitches) to which all coupled parameters need to be involved to produce its sound, a digital system's direct one-to-one uncoupled mapping of one control to one parameter is even more independent than is possible with acoustic instruments. The parameter mapping in a virtual system actually can be more isolated and stripped down than in a physical/acoustic one, a sort of superpower of simplicity only available in an electronic system. In an effort to have utmost independent control over what is improvisationally possible in a digital system, one-to-one mappings usually promote the most freedom for improvisation.

Another interesting example of complex mapping that could prove widely useful for improvisational performance is an ability to analyze metadata of a digital system to then apply that within a mapping scheme. Sergi Jordà has put much research into the mapping ability of a digital system. Regarding this kind of mapping, he proposes an analysis of the history of actions and data during a performance to use as a control source. Some of them include:

- Smoothing responses (low-pass filter), amplifying changes (high-pass filter), modifying the instrument responsiveness in different manners (using output and/or input history)
- Measuring input speed (a differentiator using input history)
- Measuring average activity
- Detecting gestures (using input history)

(Sergi Jordà 2005, 148)

In this way, users are able to make the most out of their system by using data that is already available, ready to be tapped into and used creatively. Because this metadata-type mapping is usually a product of comparing or analyzing in a macro arena, its use as a control source will tend to give the parameter it is controlling a sense of sounding innately tied to the musical system.

Some complex mapping tools and meta-mapping tools that have proven useful in the course of performance are highlighted in the following section.

Contributions

As a computer music improviser, that often must rework his entire digital musical system to cater to each performance, it has been of utmost importance to provide myself with the most effective and mentally simplifying mapping strategies and tools for the job; basically to put the heavy thinking/fore-thought intro preparation of thoughtful mapping, rather than hoping that thought is abundantly available during performance, when it really counts. Often, the functionality provided by modern controllers is still not enough to obtain desired results. Somewhat resorting back to the Monome mapping paradigm, there has been a need to create a slew of MIDI-routing middleware for controllers to suit particular demands. None of these types of control enhance actual improvisational techniques or methods. Instead, these types of added controller functionality are suited to make a controller more "self-aware," organized, and functional, to allow an improviser to be less bogged down by mapping minutia, and have more time to devote to creative improvisation when in the moment.

The concept of middleware to handle control signals is nothing new. It was the reason for the initial creation of Max/Msp, by Miller Puckett, to more easily manipulate and route control data of audio signals. The DAW, Logic, in its early days also had a few macro methods of controlling and mapping MIDI data. Many computer performers like myself have created their own such mapping tools to make their own systems more effective. Many of these tools are unique to the performer, or his musical system; but many middleware type tools that people make for complex and organized mapping are very similar, if not exactly the same. Just like the LED decoupling of the Monome, and optional MIDI bank switching on modern Keith McMillan controllers, some more common home-made MIDI middleware should be *already existent* within future controllers as a selectable option within a controller's software editor, and not require creating a MIDI handling program, or any programming ability; tools that provide complex mapping and promote more of a controller self-awareness so there is less required user awareness. Most of these tools are likely nothing revolutional, as that is the point; they are commonly useful enough, and all of their functions outweigh their limitations enough for their implementation to be warranted.

The mapping middleware to be discussed has arisen from issues incurred first hand during live performance that could not be remedied via the functions of the controller alone. Most involve slightly-to-relatively complex MIDI routings, set about to save time and eliminate confusion or second-guessing during performance. For example, being a proponent of the previously discussed All Inclusive Approach to modern improvisation, it is a personal preference to have more potential control than can be handled at any one time, in order to be able to dwell in whatever level of improvisation the spontaneous moment beckons. Often when engaging in effect-level control, many different combinations of effects are explored live, sometime to obtain a specific sound, and sometimes to be intentionally surprised with the unforeseen sonic result of different combinations. Once it is time to revert back to the audio's unprocessed original state, depending how many effects need to be manually reset, it is often impossible to revert all one-to-one controls back quickly enough to give a drastic sonic shift (rather than a gradual one), or quick section change. Another issue incurred with a massive amount of effects is that when resetting, occasionally one effect gets accidentally neglected, or cannot be reset perfectly back to its original position in time. The problem these issues amount to in live computer performance is basically one of crossing that threshold of too much control at once, and not enough brain power or physical agility to stay on top of everything at once. The easiest, and most popular solution (highlighted by experts like Jordá and Cook, Weisvisz, etc.) would be to limit control, and master performance with less effects. Although this solution can lead to potentially more virtuosic performance, the key word that is the antithesis to many improvisers (that subscribe to the All Inclusive Approach) is *limit*. The more improvisationally fertile solution[79] would be to create a technological means to prevent these issues from arising while still able to engage in the manipulation of a potentially overwhelming amount of effects. To this, MIDISnap has been created, which allows a user to set 'original' states for each and any mapped parameters, all of which flush back to that 'original state' with the push of one button. This is essentially a preset to snap to any time, at a moments notice, with a one-to-many mapping that effectively reduces more control than a human can physically and mentally operate into one single control. More than just 'original states,' presets to any desired states can be created, allowing for mindless macro shifts to a variety of carefully designed complex sonic results. Furthermore, MIDISnap modules can be used in parallel to create subgroups of control for flushing out different parts of a musical system. The amount of "limitation" placed on a musical system from this tool is a sacrifice of one button (per flushed group), for a magnified increase in performable ability that is not accomplishable directly from simple mapping. That established, this tool conforms to the criteria of its added control far outweighing the limitations it places, rendering MIDISnap a fundamentally relevant improvisational performance tool.

1.1.1.1 Controller Self Awareness

The concept of LEDs in or around a sensor for feedback has undoubtedly become a standard in helping to unite the performer and digital instrument. By the controller being aware of its own MIDI state, it is able to provide a visual indication to better inform the performer of its current state and should be executed next. Controllers since the Monome mapping paradigm have

[79] Which I consider at the essence of why digital systems can be superior.

progressed in their design and implementation of user-definable LED feedback. One pertinent example lies in a Traktor remote script for QuNeo's slider pads, where instead of showing the value of the slider in light feedback, the volume meter of the channel is being continuously displayed, still while being affected by control of the slider's range. This useful feedback gives a sense of the controller being further aware of the musical system it is a part of.

One useful tool widely used within digital systems is the concept of radio buttons, as defined by the Oxford Dictionary:

Radio button: (in a graphical display) an icon representing one of a set of options, only one of which can be selected at any time.

Figure 21. Example of Hardware Radio buttons

Though described as a digital tool, radio buttons have existed long before computers (Figure 21). It just so happens that radio buttons are an effective, and hence commonly used method of selecting an item of a group within software. Essentially a group of radio buttons differs from a group of regular buttons in that radio buttons cannot be turned off by themselves (both in data and LED feedback), only instead by the next button pressed in the radio group. This makes all radio buttons connected in a complex routing scheme. In live musical performance they are effective for mute group type routing–as in MLR and MPC-style programs, for switching between banks of things like effects or synths, or for making "modes" to switch between for anything, really. Unfortunately, because there is yet no commercial controller that offers a user-definable option for radio button functionality, some type of middleware is still required to handle the mapping. Otherwise, though regular buttons can just as easily freely select within a group, unless they are a radio button group their LED feedback will never indicate their correct state, leading only to confusion in live performance.

MIDIRadio (Figure 22) was created by the author for easy implementation of radio buttons on any MIDI controller. It is a Reaktor-created MIDI middleware that allows a user to set any group of discrete control (e.g. buttons) as a group of radio buttons. The program outputs the correct On and Off messages, as well as updates the controller's visual feedback, providing a complex mapping functionality seemingly built into the controller itself. As complex mappings can serve to aid in improvisational potential, having the controller understand, execute, and correctly display these complexities with feedback saves the improviser's ever-so coveted mental energy when undergoing performance.

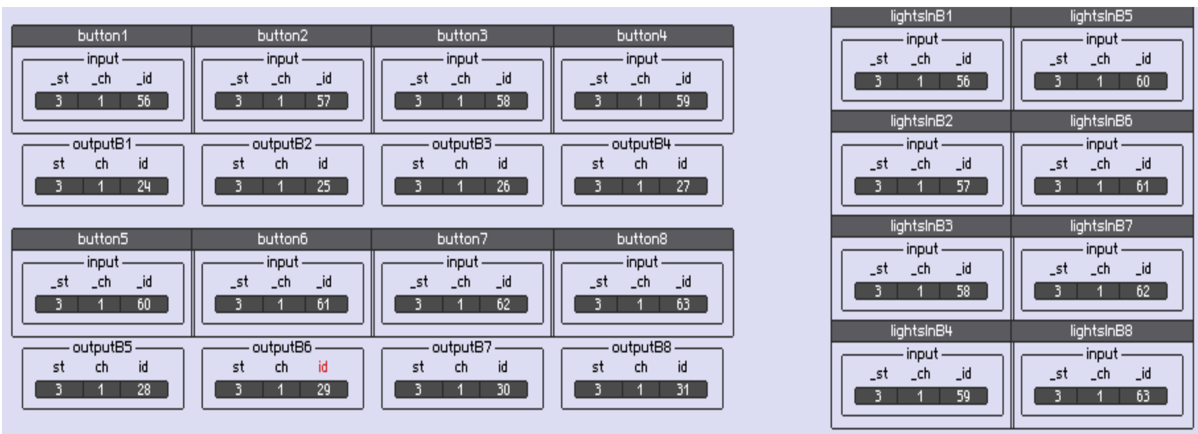


Figure 22. MIDIRadio User Control Panel

1.1.1.2 Increased Functionality

Mapping tools that are designed specifically to increase functionality can come in many flavors. A few designed by the author seem to correspond to the improvisational criteria, and could be prime candidates for commonplace embedded implementation into a controller via its software editor. One already listed, the 6D KnobSwitch, makes 6 interdependent–even radio-style–buttons out of a knob with a push button (or any continuous control with a discrete control superimposed over it). And it does not have to stop at 6D; it could theoretically be 12D, or more (the more dimensions, the smaller the range for each dimension, and the harder to control). Of the opposite many-to-one type flavor are things like bi-polar knobs, which are essentially a continuous control that actually splits its range into two full ranges of different MIDI control messages. The most common example of this type of complex mapping is the 'pan' knob in basically every DAW. Each half of the pan knob is set to control one speaker channel of the stereo field, essentially splitting the knob into two complete independent controls, but controlling one function, i.e., many-to-one. Bi-polar knobs and similar complex mappings are useful in any performance situation where the performer would like to map two full ranges of an output parameter to half of the range of the knob. Another highly functional complex control, as a sort of plan B to radio buttons,[80] is MIDIScroll, which takes two buttons– one scrolling up and one down–through a series of MIDI notes, mapped to different output parameters. This could work well for quickly sorting through different soft synthesizers, or through synth/effect presets with which to improvise. A common example in practice of another version of MIDIScroll built into a musical system is MLRV, where two buttons of its function row can be dedicated to navigating through presets of sample banks. Two dedicated buttons for scrolling can prove quicker and easier than locating software buttons on a screen with a mouse/trackpad. Plenty of other similar mapping tools have been created by the author and otherwise; tools that should already exist on a controller; many of which uphold the beginnings of the efficient mapping criteria proposed for improvisation. Hopefully those criteria and selected provided examples can begin to serve as a manifesto of more complex mapping functionality that should be included in future controller design.

Some kinds of control through mapping can exist without a controller whatsoever. Sergi Jordá's ideas for the meta-mapping of analyzed parts of a musical system can make for useful improvisational tools as well. These types of complex creative mappings are not built for controller efficiency, more so they are to enhance sonic possibilities that are otherwise impossible in improvisational performance without the number crunching ability of a computer. MIDIChain (Figure 23) is a VST within this mapping category that will analyze the volume being played on a channel in order to sidechain it and send data as MIDI to be mapped/used to "duck" any MIDI mappable parameter within a DAW whatsoever.

The task of sidechaining is most often a form of compression to reduce the volume of one sound source based from the volume of another, the most common modern example is reducing the volume of the bass when the kick drum strikes. In addition to allowing a kick drum to penetrate the audio mix, sidechaining in this fashion tends to give an audio reactive perfectly pumping, pulsating effect-level quality not really replicable manually. The past 20 years has seen sidechaining become a widely used compositional tool in contemporary electronic music. The prevalence of sidechaining in today's music is evidenced by how standard and easy it has become to utilize within major modern DAWs like Ableton Live. Figure 24 shows Ableton's standard compressor, built with its own section (to the left) dedicated just for sidechain control. The

[80] –Assuming you do not have enough buttons or appropriate middleware to orient in a radio button fashion–

audio running through this effect will be sidechained, or volume reduced, based from whatever audio is coming through the 'Audio From' section in the upper-left region of the window (in this case it would be 3-Audio).

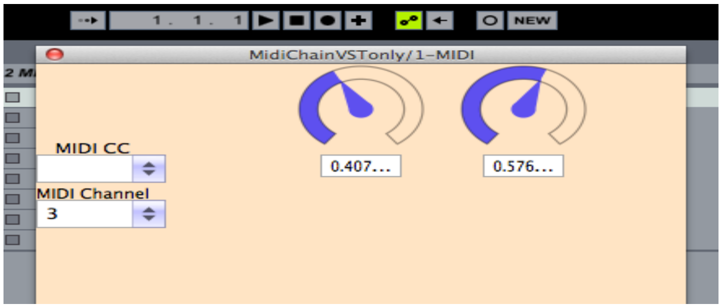


Figure 23. MIDIChain GUI (against Ableton Live background)

As sidechaining has become extremely commonplace in composition, for originality's sake, electronic music composers and performers try to be creative in their audio sources that are being sidechained from and sidechained to. Where it gets really creatively unique, is when this sidechain information is being routed to control another effect besides volume. Ableton Live begins to make this possible by adding sidechain functionality into some of their other effects, like their Gate (another type of volume-based effect) and Autofilter plugins. MIDIChain takes this idea one step further by supplying sidechain information in the form of a MIDI continuous control message, so that it can be mapped to any parameter that is MIDI mappable within a DAW (not just volume).

The ability to sidechain any (and multiple) musical parameter(s) to any audio source in a live improvisational capacity adds one extra flavor to a performer's palate; one that feeds off of its own system to reveal and exploit hidden layers of musical data; one that creates an automated sense of cohesion within a live musical system; and one that could not be done without conjunction of the superpowers we have bestowed upon our computer counterparts.

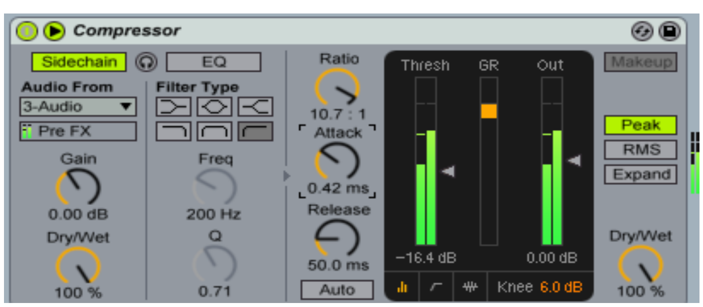


Figure 24. Sidechaining in Ableton Live is common, easy, and intuitive

Chapter Summary

"Mapping defines the personality of an instrument."

(Sergi Jordà 2005, 145)

Mapping defines the personality of a *musical system*. As digital musical performance systems have evolved, so have complex and clever mapping possibilities within them. When computers entered the realm of real-time performance, a desire for customization over the mapping of one's system and controller became popular, evidenced in a controller like the MPD. When the Monome entered the scene in the early 2000s, customization was at an all-time high in that LED feedback of its buttons could be decoupled and reprogrammed, in order to create more complex functionality amongst its buttons. Unfortunately, one had to basically be a programmer to some degree in order to program this type of functionality, or at least hope someone else's open source complex mapping worked for them, otherwise be stuck with more simple mapping schemes. Over the course of the next decade, this sense of customization remained an option within most commercial controllers. The ability to automatically, yet optionally set more complex functionality with a controller's editing software has grown increasingly more popular post Monome, entertaining another evolution in mapping ability of users not needing to know how to, or at least spend time creating any middleware mapping functionality. Out of this evolution, somewhat of a standard in complex functionality has inadvertently evolved, including functions such as bank switching, and QuNeo's simultaneous sending of multiple data signals from one sensor (as in its X,Y,Z_pads that send simultaneous Notes and CC messages for the same control). Although there continue to be advancements in this optionally selectable built-in complex mapping functionality paradigm, there is still plenty of room for more of this standard complex functionality to hard-set to a controller, requiring no middleware.

Complex mapping functions provide a means to make what would be originally exceptionally difficult–if not impossible to execute–possible. That said it is easy to map things that promote ease in execution, but severely limit improvisational control. A thoughtful and holistic approach needs to be tackled when undergoing mapping with complex sensors or routings, otherwise, from an improvisational perspective, the limiting of control can lead to undesired stagnancy. As an improviser trying to push boundaries through modern music technology, one's performance is only as complex and intricate as his creative and efficient mapping will allow. Although mapping can directly enhance feats of sonic complexity beyond acoustic means, often the best complex mapping strategies for improvisation are those that promote a sense of organization and "controller self-awareness," so one does not have to waste time, confusion, or mental bandwidth keeping track of data that can be accomplished automatically with efficiently mapped control and visual feedback. These tools become good for improvisation in this way because they allot more time, mental bandwidth, and physical ease in execution of effusions of fancy.

With complex routings, the concept of making the difficult-to-impossible possible, if not easy, is at the heart of what separates a digital musical system from an acoustic one. It should not be overlooked, however, the importance of simple, more direct one-to-one mappings. The ability to control one single sonic parameter of sound with one control is actually another valuable feature that comes easily with digital systems, while hardly, if ever present in acoustic ones. In a test conducted on mapping interfaces, mapping experts, Hunt and Kirk conclude that some people prefer independent control (Hunt and Kirk 2000); I submit that these people are natural born improvisers, aware of the independent control, and sonic combinational possibilities thereof.

Whatever type of mapping scheme is necessary for a particular musical system, if it is to be improvisationally suitable, and promote variety, it must exhibit an inexhaustible amount of control. This is not to test the limits to how much mental juggling can be done at one time, but to have more option to develop throughout the course of a performance, and be able to develop differently throughout different performances. It is my own preference that there must be an almost overwhelming amount of control; this creates a musical system that–like a characteristic of a good instrument–one can grow with, explore and hone over time.

In a related call for progression over simplification is a quote by Joel Ryan of STEIM[81] (Studio For Electro-Instrumental Music). When applying his phrases of physical effort to the physical and mental control from efficient and robust mapping, through his words, a need for a high amount of control is rationalized.

> Most computer instruments in use are those provided by the commercial music industry. Their inadequacy has been obvious from the start -emphasizing rather than narrowing the separation of the musician from the sound. Too often controllers are selected to minimize the physical, selected because they are effortless. Effortlessness in fact is one of the cardinal virtues in the mythology of the computer... Though the principle of effortlessness may guide good word processor design, it may have no comparable utility in the design of a musical instrument. In designing a new instrument it might be just as interesting to make control as difficult as possible. Physical effort is a characteristic of the playing of all musical instruments. Though traditional instruments have been greatly refined over the centuries the main motivation has been to increase ranges, accuracy and subtlety of sound and not to minimize the physical. Effort is so closely related to expression in the playing of traditional instruments. It is the element of energy and desire, of attraction and repulsion in the movement of music. But effort is just as important in the formal construction of music as for its expression: effort maps complex territories onto the simple grid of pitch and harmony. And it is upon such territories that much of modern musical invention is founded.

[81] http://steim.org/

Chapter 4

Inputs and Outputs

Being that the computer affords the performer to take on more–potentially all–sonic roles during a performance, the improvisational performer-composer must find ways of alleviating performative and compositional responsibility, such as the difficulties of playing multiple instruments at once. With non-computer music this might amount to the clichéd one-man-band street performer trying to juggle playing drums with his feet, guitar with his hands, and soloing on harmonica or singing. Or another acoustic score-level improvisation might be a composer constantly re-arranging the music throughout a piece, requiring him to somehow instantaneously re-orchestrate and re-notate and re-distribute the piece to his orchestra after each new arrangement decision during a performance. Both examples are impractical to say the least. The inclusion of computer-controlled hardware and software into musical systems, however, begins to make executing all sonic roles a more surmountable task.

Can some computer software or hardware be more improvisational than others? Can they evolve ways in which improvisation can be performed? Hopefully this work thus far has already successfully demonstrated that the answers to these questions are 'yes'. This chapter documents two explorations into improvisational inputs and outputs, one piece of software called, LiveMLR–an improvisational advancement/enhancement over the original MLR–, and one hardware instrument called, LightBalloons–an improvisationally controlled visual interface to accompany music, performed using musical concepts, techniques, and approaches. These examples serve as a case study for approaching improvisational tools–be it hardware or software–within a digital paradigm.

LiveMLR

LiveMLR seeks to enhance the original MLR–the already highly improvisationally conducive electronic music performance program/system/technique written for the Monome in

Max/MSP. LiveMLR is designed to work within Ableton Live. The potential benefits gained by this merger are considerable and worth mentioning. To first comprehend the benefits, a heightened understanding of the inner workings of the original MLR and its pros and cons is necessary.

Motivation

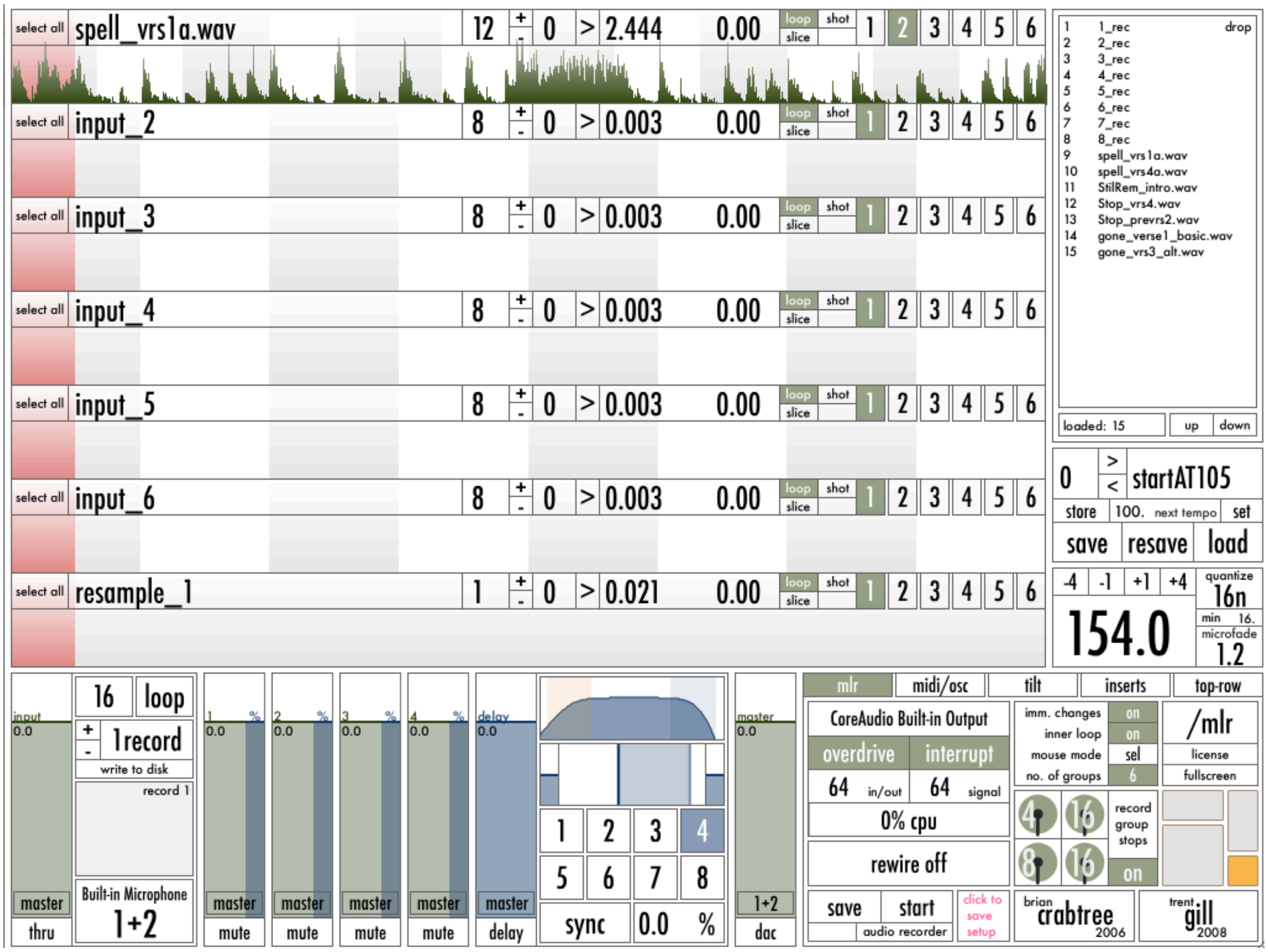


Figure 25. MLRV – Hypersampling Software Instrument Version of MLR

Think of MLR as an audio sample sequencer, or, perhaps more fittingly sleekly worded, a "hypersampling software instrument,"[82] made specifically for the Monome. The MLR GUI is shown in Figure 25.[83] Most of MLR's functionality lies in its audio sample region–the upper left box that comprises most of the GUI–where audio samples are placed and handled. This is also

[82] http://monome.org/docs/app

[83] This figure depicts MLRV, a later revision in 2008 by Brian Crabtree and Trent Gill for Max5; it operates exactly as the original, some updates being a larger, more readable GUI, a modular function row, and built-in delay.

the part that is displayed via LED feedback on the Monome. Each audio sample is laid across with equally subdivided trigger points amongst the buttons of each row. The general idea behind MLR is that all audio samples are quantized to one global tempo, so all audio samples are in sync with one another and easy to sequence; additionally, all retriggering of samples are set to trigger at the next quantized portion of the tempo, the level of quantization is configurable in the GUI. The adjustable quantization levels combined with type of audio samples used provides for a system that can be improved upon with practice and experience, an integral factor for any lasting instrument. On the Monome hardware itself, just like most sequencers, each row's audio play position is stepped through and indicated by LED feedback on its button row. The buttons being both control sources for buffer shuffling and LED feedback indicators make this system very intuitively conducive for live performance, as the performer rarely has to stray his gaze away from the instrument. By freeing the performer of otherwise necessary note-level performance constraints like exact rhythmic precision and correct pitch execution, MLR can allow the performer to spend more time dwelling in other improvisational arenas. Be it MLR or whatever improvisationally-minded software, the abstraction or automation of what are (or used to be) normal required performed constraints like precision and note playing can potentially help take improvisation to newer heights through score-level means.

Audio buffer sequencing is MLR's forte, but some other important features that round out MLR to be a fully functioning stand-alone performance system are worth addressing:

Function row: Figure 25 is set up for an 8x8 monome, but there are only 7 audio sample rows. This is because the top row of the Monome is the function row, which takes care of MLR's extra functionality, pattern records, mute groups, octave switching, to name a few. The function row serves to enhance score-level control, which tends to add dimension and ease in complexity for a digital improvisational performance system.

Mute groups: When rows of audio are placed into the same mute group, the most recent one triggered will take priority and mute other tracks within the same group (basically a group of radio buttons).

Pattern records: MLR provides multiple event-based loopers, called pattern records. They are designed for quantized looping of button press control data, as opposed to audio looping. The looped recorded button press data persists through the switching between different audio loops, which gives an interesting rhythmic looped consistency while the audio material is changing. Data looping can be especially advantageous to improvisation in that it is decoupled from the output (from the audio), which promotes more modular user-defined creativity in looping.

Live audio record: In addition to event based pattern record looping, MLR is built to record quantized lengths of live audio, which are then immediately available as audio samples to buffer sequence on the Monome.

Rewire: a function (built into Max/Msp) that when activated allows audio routing and tempo syncing with another program such as a DAW like Ableton Live.

For being a standalone performance system, MLR is quite minimal in design. Like all of Brian Crabtree's inventions, somehow somewhat paradoxically, its stripped down aesthetic promotes substantial versatility. With MLR, depending on how a performer thinks about using audio samples is one versatile aspect, for example whether each sample is a song on its own, or a section of a song, or an instrument within a song, etc. (basically whether a performer likes to dwell in score-level or more note-level performance). Like any bit of new musical technology of its age, how to perform with MLR has tended to follow that of its initial pioneers/virtuosi, most notably Brian Crabtree and Daedelus–both highly improvisational within their sets. With a focus on the note-level with, Brian Crabtree tends to exploit MLR's live audio recording and looping functionality, creating very organic, textured, improvisationally repetitive music. Daedelus uses MLR more in a score-level approach, playing extremely live dance music. His live performance style and technology enables him to create and arrange new "songs" on-the-fly out of preexisting ones–out of his own music and otherwise–blurring the seemingly distinct lines between DJing, mashing up, and performing original music. Daedelus' unique style of live computer music performance has set a tradition for MLR performers, and an example for an evolution of what live improvisational electronic music can be.

As advantageous as MLR has the potential to be in the right electronic improviser's hands, it is not without its imperfections. It is built for blending or mashing up audio samples together (i.e. playing multiple loops from different songs simultaneously). Because samples are automatically sped up or slowed down to be quantized to a definable tempo, MLR is especially suited for rhythmic audio material. Tempo conformity is an extremely useful advantage when working with computer music, yet it also proves as the most challenging issue to overcome with MLR. This speeding up or slowing down for the sake of rhythmic conformity also raises or lowers the audio's pitch, which often causes a harmonic dissonance among songs recorded in different keys and tempos that is far too unpleasant when blended.[84] One potential alleviator to this problem might be if there existed some kind of cueing mechanism within the program, as a DJ would use to sample what to play next in headphones to have some foreknowledge as to whether the tracks work together harmonically. As cuing is not a feature in MLR, the only solution to this problem besides coping with/enduring harmonic dissonance, or abstaining from playing harmonic music, is an excessive amount of preparation: either heavily treating (carefully re-pitching) the audio material to blend in a more euphonious manner; or listening to every combination of songs with one another and making note of the ones that work well together.[85] So essentially, with the pitch slaved to rate being an unavoidable byproduct of MLR, a big (and somewhat unnecessarily tedious) part of being an MLRist requires being forced to spend a considerable amount of prep time making or finding audio to blend well together.

Another issue is that MLR is a standalone program. It does what it does very well, but it does not do very much. MLR is a fully functional time sequenced buffer manipulator, with a few other frills and necessities. It is not some plugin within a DAW; it is not a building block component for any larger performance system; it is fairly limited in scope. Of course, founding fathers, Daedelus and Brian Crabtree are a testament that MLR can be more than sufficient as a standalone performance tool, but it is one flavor of performance. It is like someone only eating mint ice cream for every single ice cream excursion. To that someone there is no reason to bother with any "pettier" flavor. But other ice cream enthusiasts might like to mix and match flavors. MLR has ways of being used in conjunction with other programs, but they tend to add further hurdles and limitations. A performance system like Ableton Live, however, already comes robust with many "flavors," and supports more blending–or added ability–by incorporating plugins.

As a technique, though MLR is extremely improvisational in some respects, it almost completely lacks any effect-level control capability on its own. Whether just EQing, or delaying or reverbing, etc., omitting effect-level control in an improvisational piece of software is a bold choice, one that again, limits the scope of an improvisational performance. To MLR's defense the issue is more than likely one of hardware limitation; there is only so much control that can

---

[84] Similar to if one were to slow down/speed up a record or tape—the pitch will follow.

[85] The author's experience is with the latter. The results of songs that blend well together (though definitely depend on style) probably average to about 1 in 15, and are still usually only within some audibly permissible tolerance, rarely perfectly in tune.

be imparted on a controller[86] before it grows to overwhelmingly difficult to control. As previously noted, and as an alleviator to this and the previous issue, MLR is built with a rewire function, which allows audio to be routed out of Max/Msp and into a DAW like Ableton Live, creating a bridge so as to harness more "improvisational flavors" like effective effect-level control. Unfortunately rewiring is not without its own set of difficulties, like an added latency between a button press and its intended result. Moreover, not all MLR data can be exchanged through rewiring, only tempo and audio. Furthermore, the incorporating of and communicating between multiple programs tends to get more confusing and extensive to setup[87] for the performer, as well as sometimes the computer.

A Bettering of Both Worlds

A digital live performance tool that predicates itself on its attempted all-inclusiveness could only benefit from incorporating increased extended performance techniques. This statement applies in reference to how improvisationally beneficial a performance DAW like Ableton Live could become with an enhanced technique like MLR operating within its own clips in its own system. Conversely, theoretically all previously described "MLR hurdles" would be overcome by this collision of performance systems. For example, the extensive user control of Ableton's clips would automatically be at the user's disposal for incorporation into MLR, such as independent pitch and tempo control to address harmonic dissonance (Figure 26 shows Ableton's various clip control parameters).

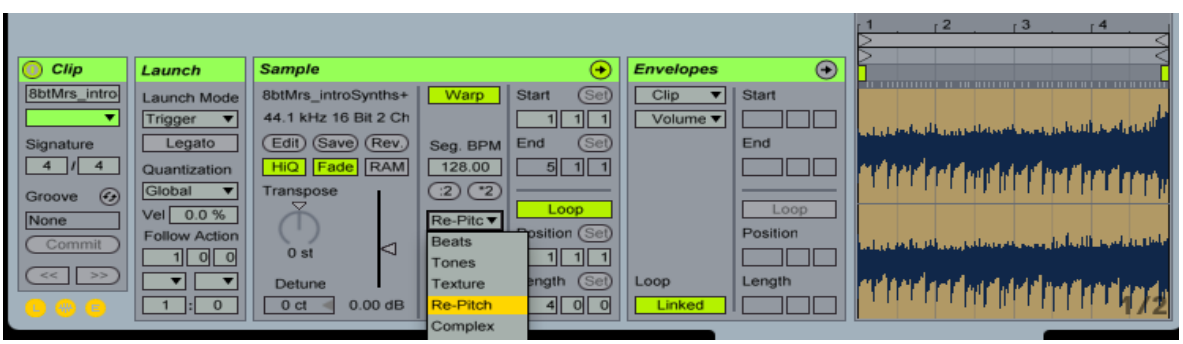


Figure 26. Ableton Live Clip Control Window

[86] A non-continuous controller at that.
[87] As will be touched upon in the next chapter, set up time is an essential factor to consider for a performance system in a live situation.

Another interesting digital performance issue is remedied by the merger between these two programs. An instrument like drums (acoustic samples, or synthesized) is generally easier and more intuitive to perform with/manipulate as a whole rather than its individual components (kick, snare, hi-hat, etc.). As one might suspect, operating on drums as a whole track however, excludes an ability to separate or switch out specific drums at will, a limitation not exactly desirable, but a sacrifice somewhat necessary with MLR alone. Although MLR-style buffer sequencing is not currently built into Ableton Live, MIDI sequencing is. With MIDI tracks in Ableton Live, individual drum components (kick, snare, etc.) can have their own track, while all being sequenced with one master MIDI track, allotting for both macro manipulation on the whole drum track, as well as on individual drum components. This provides a means for both score-level and note-level performance access, and an added freedom to switch which individual drum sound is being triggered by a MIDI note, a feature lacking within MLR. By merging the two systems, with the use of MLR on MIDI clips within Ableton, a best of both worlds collides, leaving no aforementioned improvisational limitation whatsoever! This advantage does not stop at drums; one could theoretically MLR[88] any MIDI clip to trigger sent to whatever synthesizer, instead of being stuck with an audio render of the synthesizer track. One could even use MLRed MIDI tracks to trigger other fringe or interesting MIDI outputs never before "buffer manipulated" to this fashion, such as robots, or lights, etc.

In order to merge the two programs for a heightened live computer improvisatory experience, it became necessary to understand MLR from the ground up. The first attempt came out of re-building it from scratch in Reaktor (see Figure 27). MLRReaktor has all the necessary components of MLR: audio buffer sequencing, mute groups, pattern records, button quantization/tempo synchronization, and preset switching. It functions as a VST in Ableton Live, and does its own sort of rewiring so each row becomes its own audio channel. As a replica, MLRReaktor provides little actual improvisational enhancement to either Ableton, or MLR. Additionally, Reaktor has its own quirks that ultimately render MLRReaktor more off-putting to use than MLRV in Max/Msp, most notably its preset management (i.e. the storing, setting and saving of samples and presets) and audio sample management are tedious to preset and navigate. MLRReaktor does not necessarily suffice or warrant an improved replacement over the original MLR (or MLRV). As a learning tool for understanding how to program a fully-fledged

[88] MLR will sometimes be used as a verb to describe act of utilizing MLR as a performance technique.

performance system with complex signal routing, buffer sequencing, dependable timing and real-time improvisational controllability, building MLRReaktor was a complete success.

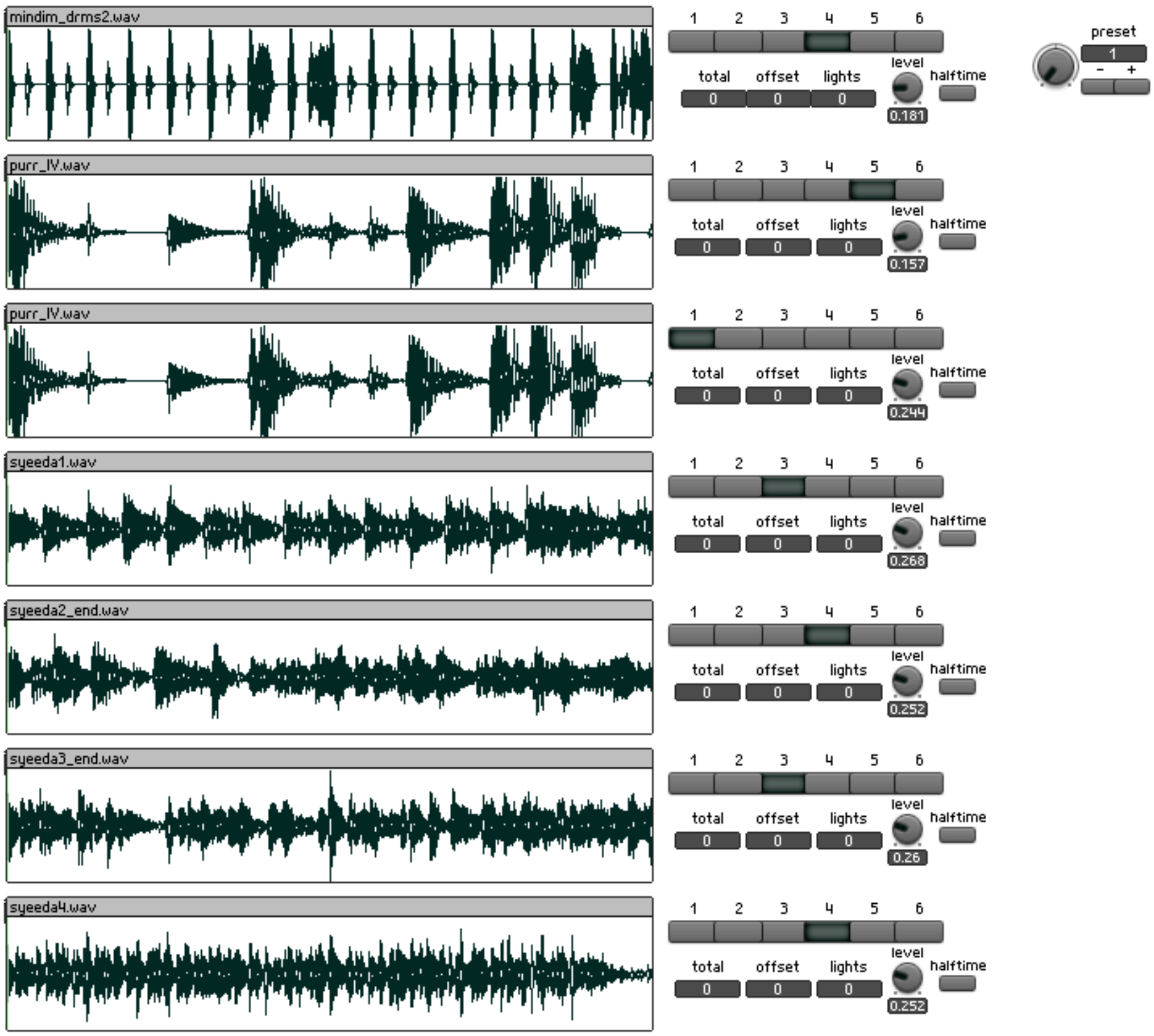


Figure 27. MLRReaktor GUI

LiveMLR explained

MLRReaktor is at best a comparable substitute to the original MLR. Such a live computer driven performance technique, however, deserves the effort to be enhanced, or improved upon. For reasons mentioned, a more imbedded marriage between MLR as a technique, and Ableton Live would stand to garner the most improvement/enhancement amongst both entities. Fortunately, Ableton's control of itself through programmatic means like remote scripting, MaxForLive, and LiveOSC provides a sort of back door into the accessing and manipulating of its clip data. In short, through these "back doors" to features (currently) unavailable out-of-the-box, buffer sequencing of Ableton's actual clips in an MLR-like manner is a realistic possibility. Thusly, this conceptual revelation began the undertaking of the creation that has become LiveMLR–a hypersequencing software add-on for Ableton Live–as a case study for enhancing improvisational software for computer performance.

The open-source programming language, Processing, was chosen for the creation of LiveMLR for multiple reasons, namely, because it is heavily documented for fast prototyping, it is text based and therefore less constrained by the rules and flow set about by graphical programming environments, and it is free and available to the public for easy distribution. LiveMLR was created to work exactly as the original MLR, the only difference being that it operates on Ableton Live's audio clips instead of stored clips within another program (like Max). This adherence to replication was decided upon for ease of transition for MLRists into a new system, for discipline in programming practice, and for a more obvious proof of concept, or, demonstration of the possibility of a bona fide exact working version of MLR inside Ableton Live. Though the difference between MLRs being little more than which program's clips are getting sequenced might seem minute, from a improvisational performance perspective, it is probably the most vast improvement that can be incorporated into MLR. All pitch, tempo, cuing, and related issues can now be replaced with added functionality that Ableton Live offers within its clips, tracks, routing, etc., most of which never made its way into the original minimalistic MLR. The performance advantages that MLR-type sequencing offer an already functional DAW as Ableton, stand alone to warrant enough of an enhancement to prove LiveMLR's worthiness, yet even more advantages are abound. One issue with Ableton Live never really tackled by its developers is the difficult-to-read small text size of its clips. Thanks to a LiveOSC command that will query Ableton's clip names, in the spirit of MLRV's extremely large readable clip text (Figure 25), LiveMLR solves Ableton's age-old small clip issue by listing largely the clip name of each clip within an Ableton scene (the comparison of each are in **Figure 28**); every scene-up or scene-down shift repopulates the LiveMLR list, which symbiotically provides the pre-built preset switching mechanism inherent in MLR.

All control within the LiveMLR GUI shown in **Figure 28**, through LiveOSC, correspond to direct control within Ableton, even though some of the functionality in LiveMLR might seem obscure as compared to Ableton Live. The volumes, mutes, stop/play, scene scroll are one-to-one relationships to the same respective control within Ableton. The loop/shot/slice and mute group radio buttons are more complex correlations that involve some algorithmic processing within LiveMLR, but help to allocate more macro level organized control over the entire Ableton-MLR performance system. The organized sample control of the loop/shot/slice and the more organized channel control of the mute groups could both be quite useful improvisational Ableton tools even on their own without MLR.

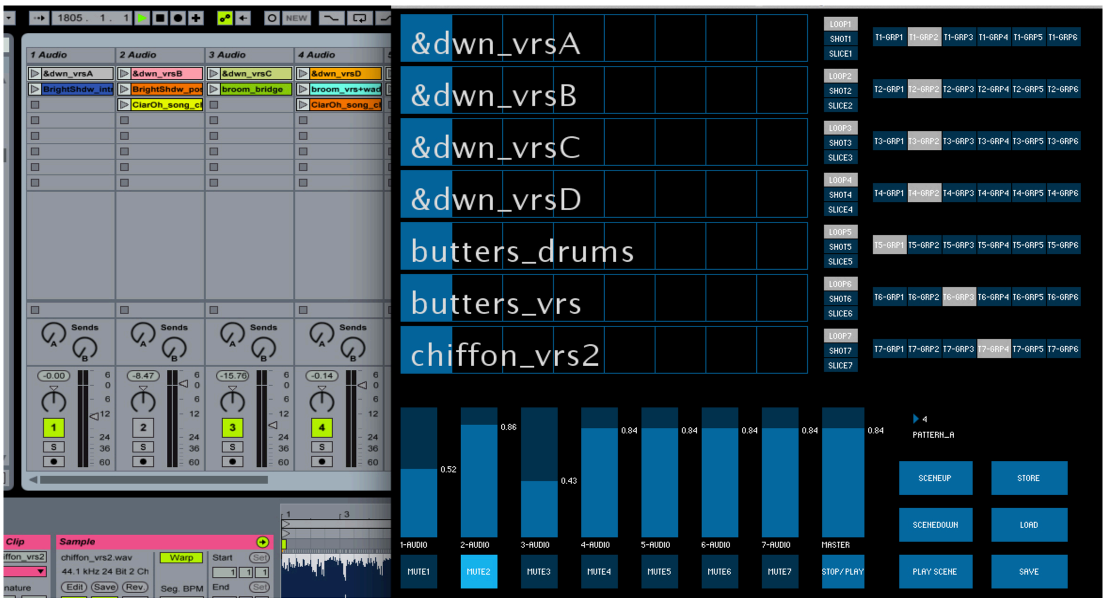


Figure 28. LiveMLR GUI (right 2/3) laid over Ableton Live Session View (left 1/3)

All of the direct control over Ableton so far described, such as MIDI mapping of mutes and volume, etc., can be done without any back doors (though much of it requires some system awareness and outside computation). Having it all wrapped up in a simple visually effective GUI such as LiveMLR is far more intuitive and user friendly than a list of mapped parameters, however. In any case, the quality that essentially characterizes MLR–its quantized buffer sequencing–is not possible in Ableton Live through any other means aside of the aforementioned back doors. With the aid of LiveOSC, one is allowed to move the "play head" or play marker indicator anywhere within the audio/MIDI clip upon command, i.e. the buffer sequencing that is MLR. A simple 'move play position' command is allocated through the back door for this purpose. Unfortunately, this command simply does not work for a properly quantized MLR. Because the current play position of a clip is usually somewhere in between its quantized divisions, a jump with this command will not quantize properly. Instead, a slew of consecutive commands (still only available through Live's back doors) are rapidly executed to get the clip to jump the exact number of beats from its starting position.

Related/Future Work

One of the essential components of MLR–its pattern records–is to date unstable in LiveMLR. The faster the tempo, the less likely the pattern record will loop exactly what was recorded. After plenty of frustrating debugging and variations, it has been assessed that the data is not transmitting and being called upon fast enough when exchanged between programs during heavy messaging. This instability in communication between programs supports the argument for having more of an all-inclusive program in a digital live performance system, rather than needing to communicate through a multitude of them. Illustrated in Figure 29 is the chain of command message data for LiveMLR during regular button press performance (process 1), and pattern record performance.

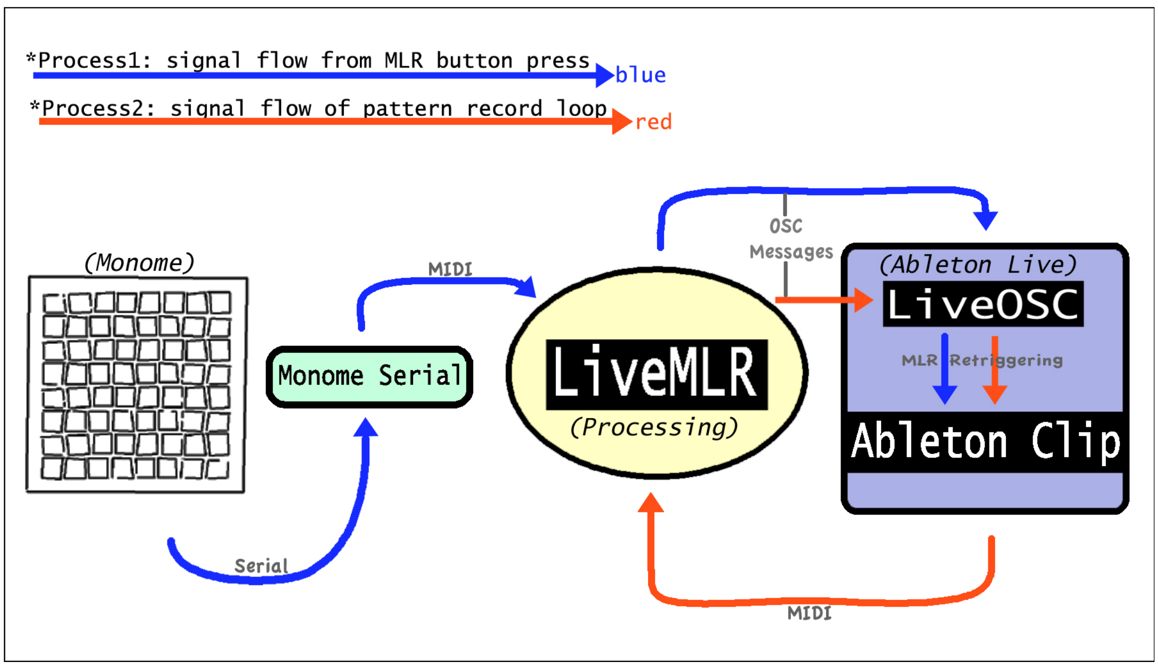


Figure 29. LiveMLR Signal Routing for regular MLR button press in blue, and pattern record loop in red

Process 1–normal performance: button presses get sent from a Monome to MonomeSerial, which converts and sends MIDI messages out to LiveMLR (Processing), which then send a slew of OSC messages to LiveOSC (Python) that combine to MLR an Ableton clip.

Process 2–pattern record process: When the pattern record is engaged, MIDI data from a Monome button press undergoes Process 1, while also getting recorded into a specially designed pattern record table. So long as Pattern Record is engaged, the table continues to loop through Process 1 with its recorded messages simulating as Monome button presses.

Process 2 requires a lot of data to be passed around multiple programs in a loop for it to work properly. Moreover, the data being transmitted for each button press consists of a bundle of actions. Ultimately, the rates at which code is being read by each program ends up being too slow for the pattern record process to actuate in time, so looped patterns often retrigger one beat early or late. Although progress is currently being made to try and make pattern records work fast enough, the eventual solution appears to be taking Processing out of the equation, and instead rebuild LiveMLR in MaxForLive or ClyphX. This option could prove faster because LiveMLR would ultimately live within the Python scripting of Ableton (the language Ableton Live also runs in), and would not need to be subject to the latency of being sent out to Processing to get sent back in.

Many styles and many artists hold an affinity for audio buffer sequencing and manipulating during live performance. MLR style buffer sequencing in particular is not extremely common,[89] but many are understanding its potential and are making it more available to the general public, either via performance, as in Daedelus, or via MLR-like software in other programs. Two versions of MLR built in Reaktor have showed themselves within the past few years aside of MLRReaktor, one from Flip Mu, and one from a programmer under the alias Kid Sputnik. Another Reaktor VST tool from Flip Mu that incorporates very MLR-like buffer manipulation is their Streaming Chopper, which retriggers a sample at properly subdivided beats, but only for a moment in time, then reverts back to the normal play position of the sample, whereas MLR will continue playing from wherever it was retriggered.

Discovered months after building LiveMLR, a program built exclusively for the Ableton Push controller called, PXT,[90] has uncanny similarities to MLR. It is a whole performance system using Ableton's "backdoor" API with different modes of performance all automatically mapped to Ableton's Push controller. One of the modes of PXT works exactly as MLR's quantized subdivided sample sequencing does, directly on Ableton Live's clips. Oddly there is no mention of MLR or Monome as an inspiration or comparison on their website or promotional/instructional videos. Though the concept behind MLR's quantized sample

[89] The most likely reasoning being that it exists in a program (Max/Msp) written for programmers, and not necessarily performers.
[90] http://www.nativekontrol.com/PXT-Live.html

sequencing seems like a fairly logical performance technique, it would be fascinating if PXT were developed completely independently of any awareness of MLR. Somewhat similar to PXT, it is the eventual personal goal to implement just the most favorable aspects of LiveMLR into a more encompassing and all-inclusive Ableton Live-based performance system.

LightBalloons

Figure 30. LightBalloons: A Visual Instrument

LightBalloons is a visual interface designed for exploration into improvisational visual control, for performance, musical accompaniment, structural digital installation art, and otherwise. Simply put, this project combines balloons and LEDs–two somewhat technologically antiquated inventions, but when combined (with newfound technology), makes for a wonderfully modern and enthralling approach to interacting with full-color light. A chain of LED/Helium-filled balloons are erected in virtually any configuration, and controlled with various input sources (anything that will output MIDI), which altogether make for an interface of extremely modular, flexible, floating, individually controllable "light bulbs," that can vary in color, brightness, size, and aesthetic configuration.

Most Music Technology aficionados tend to think of a hardware interface as a controller, or input device, and are generally correct. With technology now as completely decoupled as it has grown, *In*-terface design usually implies *in*-put, with need for little regard placed into designing what the other side, the output, will be. But there are exceptions, LightBalloons being one of them. In this sense, for a musical comparison, LightBalloons is slightly akin to a drum machine, sampler, synthesizer, or musical robotics, which represent the other end of an interface: the output result, or outcome. Not only a beautiful and fun endeavor into live performance art, Lightballoons has been designed as a case study to visually explore musical improvisational ideas, as well as an outlet to discover ways to make the output portion of interfaces more conducive to improvisation.

Motivations

Among the many, there are three direct inspirations that have helped bring LightBalloons to fruition. The most influential and reminiscent of the instrument comes out of a performance project by Robert Henke and Christopher Bauder, called, ATOM,[91] premiered at MUTEK[92] festival in 2009. In an attempt to realize a vision by Robert Henke of a 3D pixel matrix, ATOM was an audiovisual performance utilizing an 8x8 matrix of white LED/helium-filled balloons on an adjustable track, wherein each balloon's height and brightness were controlled live (as was ambient audio), in an improvisational manner. In an interview by Peter Kirn for Create Digital Music,[93] Robert Henke addresses the importance of spontaneity in not only audio, but visual performance, "I realize I am very very good if I just can do things spontaneously; a lot of the things which I found incredibly beautiful tonight, I would never ever had done [if] I had to prepare them; its just things that work spontaneously in a situation."

A little more distantly related project, but equally as inspiring to the ongoing work of LightBalloons is called, Particles,[94] by Daito Manabe and Motoi Ishibashi, wherein a surplus of Ping-Pong balls–embedded with wirelessly individually controllable white LEDs–are sent down a larger than life mechanical track by their own gravity; then back up again with a conveyor belt to continue the looped process. The blinking of the balls in various modes against an unlit black space creates an immersive real-world pixelated looking particle system. This is an installation as opposed to a performance, but the creation of a seemingly floating, real-world particle system has instilled a desire to expand on the idea for performance.

---

[91] Expose, clips of the performance, and an interview after the performance can be found at: http://createdigitalmusic.com/2009/07/video-interview-atom-by-robert-henke-christoph-bauder-musical-balloon-sculpture/

[92] http://www.mutek.org/

[93] http://createdigitalmusic.com/

[94] A video, interview, and expose of the Particles project can be watched here: https://vimeo.com/45891608

The collection of groundbreaking light art projects by the visual label, AntiVJ,[95] is the last somewhat vague, albeit direct inspiration for the idea behind LightBalloons. A major focus of the label is a utilizing of creative methods, spaces, and structures mostly for pushing the boundaries of what can be done through projection mapping. Their drive to search for creative artistic outlets to employ the latest technology was the necessary keystone to properly think outside-the-box enough to create a light-based art project that is innovative, novel, and unique, such as LightBalloons.

Figure 31. ATOM (left); Particles (middle); AntiVJ (right)

After embarking upon the creation of LightBalloons, web searches for other similar projects pointed to not an artist, but a company selling a patented automatically blinking "light-tube" specifically for insertion into balloons called Lumi-Loon.[96] [97] What separates LightBalloons to be a performable improvisational instrument rather than a fun technological toy is that each balloon's LEDs are individually controllable and full color, as opposed to Lumi-Loon's automatic blinking one-color LEDs. It is interesting to see creative minds thinking similarly, however. There are not very many examples of artists who use balloons as their artistic medium. In many cases they are hard to work with; they're fragile, and pop; and helium is in high demand, therefore expensive. However, the appeal and even practicality for working with balloons seems to outweigh their shortcomings in that they are lightweight, portable, flexible, moldable, and tend to bring about a nostalgic childlike wonder to those that view and interact with them. Additionally, the under-saturation of "balloon artists" tends to make the viewing of LightBalloons that much more impressive and memorable.

[95] http://antivj.com/

[96] http://www.windycitynovelties.com/516c/lumi-loon-balloon-lights.html

[97] Discoveries like these usually amount partly to discouragement from an independent thinker manifesting a similar concept first arguably rendering the concept less original or novel, and partly to encouragement for there being likeminded thinkers, giving a sort of affirmation that the concept is worthwhile.

LightBalloons Hardware

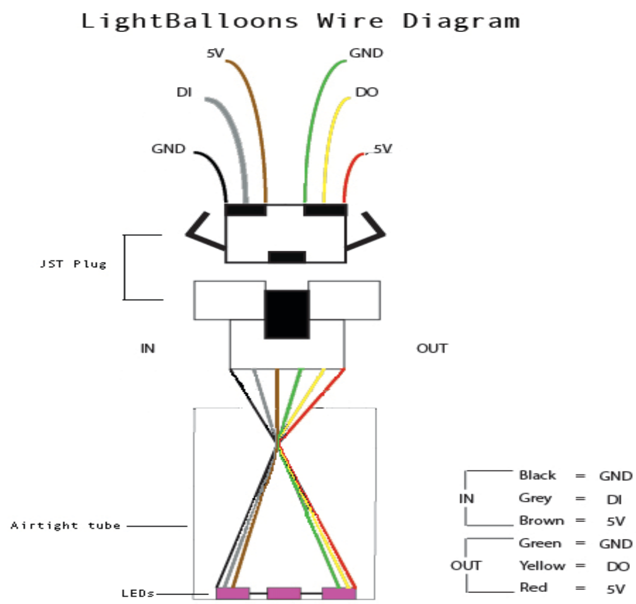


Figure 32. LightBalloons Essential Parts: LEDs, Airtight tube, locking plug

Not much is used to create LightBalloons aside of a microcontroller (Arduino), balloons, and chain of (daisy chainable) LEDS, coined, Neopixels (as in Figure 33 left) by Adafruit–their major American distributor, or WS2812s–more technically speaking. The WS2812 chip is quite easy to work with in that its operation is handled by an Integrated Circuit (WS2811) specifically designed for RGB LEDs that is embedded within the LED itself. Manufacturers have made Neopixels even easier to use by selling them complete with a fully functioning miniature PCB on a flexible strip (Figure 33 right), which only requires 3 connections to a microcontroller (power, ground, and data).

Much research is being placed into making the LightBalloons hardware more conducive to consecutive live performances. Some challenges specific to working with this particular hardware to be overcome include: (1) Trying to make balloons airtight with wires emanating out of them–eventually solved by inserting the LEDs inside a clear plastic tube which then get inserted into the balloon; (2) Being such a new, hardly documented or researched piece of technology, painstakingly trying to make the Neopixels not blow up, not flicker, and not turn to an ugly brown, has been the other major challenge. Although these issues are clearly very individualized, working out hardware kinks like these until the problem has been adequately solved plays a big role in an instrument being a performable one. It has been realized that the need to take hardware beyond a trial prototype phase, beyond unworkable lengthy setup times and maintenance that should be unnecessary, invariably helps to make the interface something desirable to perform with, and to continue to improve upon. In a live situation, if hurdles that render an interface more difficult than just being a plug-and-play device cannot be overcome, the device can quickly and increasingly grow to become an undesirable performance tool.

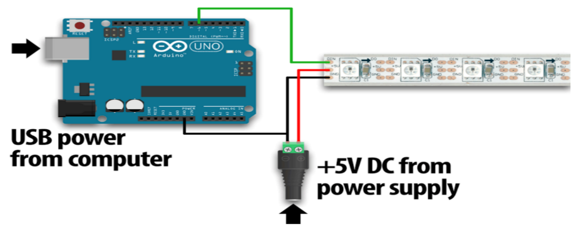


Figure 33. Neopixel RGB LED (left); Adafruit's Neopixel Circuit Diagram (right)[98]

LightBalloons Software

Whereas acoustic instruments tend to have more direct relationships between the input and output source, digital instruments can be programed to have any relationship imaginable to ease performance and aid improvisation. So as with LightBalloons, much of its capacity for modern improvisation lies in its programming.

The only control one has over each Neopixel is its variable brightness of each of its three base colors (red, green, and blue), the varying brightness levels of the three colors combine to create any and all color. Although gaining the proper color and brightness via RGB is possible, it has been converted to HSB for simplification in control.[99] But even with just the three simple continuous controls per light–hue, saturation, and brightness, a strand of 100, or even just 20 LightBalloons will surely add up to more control than most MIDI controllers have sensors for

[98] http://learn.adafruit.com/adafruit-neopixel-uberguide/arduino-library

[99] RGB (Red Green Blue) requires all three RGB values in combination to control color and overall brightness, whereas utilizing HSB (Hue Saturation Brightness) will allow a user to control color with one dimension, brightness with another, and its white/color balance with another. In essence, HSB can be thought of a more direct one-to-one relationship amongst its characteristics, whereas RGB is many-to-many. (Interestingly enough, however, as is the paradoxical nature of mapping perspectives, because the LEDs are actually red green and blue, using RGB is a more direct, one-to-one method of communicating with the Neopixel, whereas HSB requires a sort of decoupled coding middleman to make it more human operable.)

and humans have appendages for if only one-to-one control is used. Therefore, although independent brightness control has been programed to be available for LightBalloons, they are also arranged into logical modular groups (such as half and half, or every other light, etc.), all easily controlled as a group through one mapped controller parameter. Some automatic sequences (such as a color swirl, and ripple outward from a random color and location in the strand) have also been programed to be triggerable; some parameters (speed, direction, etc.) of these more automated modes are set to continuous controls. In this sense, one can improvise on more of a score level with LightBalloons if desired. For communication to an Arduino, ultimately an Ableton bridge via MaxForLive has been established that converts MIDI data to Serial–a data protocol the Arduino can read. This then provides easy light programming and enables the utilization of Ableton's already built-in robust sequencing and other MIDI effects/routing through Ableton Live (see Figure 34 for LightBalloons signal flow), lending improvisation to the lights as easy as improvising with a synth or effect.[100] New modes and groupings of lights can be catered to each show. Coming from a musical background, thinking of programming and performing LightBalloons in electronic music terms concepts and principles (such as sending LFOs of color, or rhythmic sequencing, etc.) makes improvising with the instrument easier to approach, and incites an automatic linkage with the music it is accompanying.

To date, the transmission of data from Ableton Live to the interface (the serial protocol) is not flawless. Improper messages get dropped, so with heavy messaging the occasional result is that lights occasionally will not turn on or off when supposed to; the faster the messages stream in, the more that messages get dropped. It was relatable and somewhat comforting to hear Robert Henke complaining about a similar dropped message problem during his interview after his ATOM performance,[101] and commenting how imperfections like these sort of enhance the personality of the instrument and performance to make it what it is. As important as working with and accepting imperfections can be, this is no excuse for a continued refinement. Because

[100] Whether to program complex modes inside the Arduino, or outside in external programs Like Ableton Live has been an ongoing back and fourth battle. In theory it seems more efficient and encompassing to use external programs so the microcontroller can be programmed and left alone, however, having the automation built into Arduino involves less serial communication to the Arduino, and hence makes for more perfect and stable efficient control. This question will be briefly addressed further in the conclusion chapter.

[101] http://createdigitalmusic.com/2009/07/video-interview-atom-by-robert-henke-christoph-bauder-musical-balloon-sculpture/

as close to perfection as possible is desired, a reworking of the serial protocol is in currently underway.

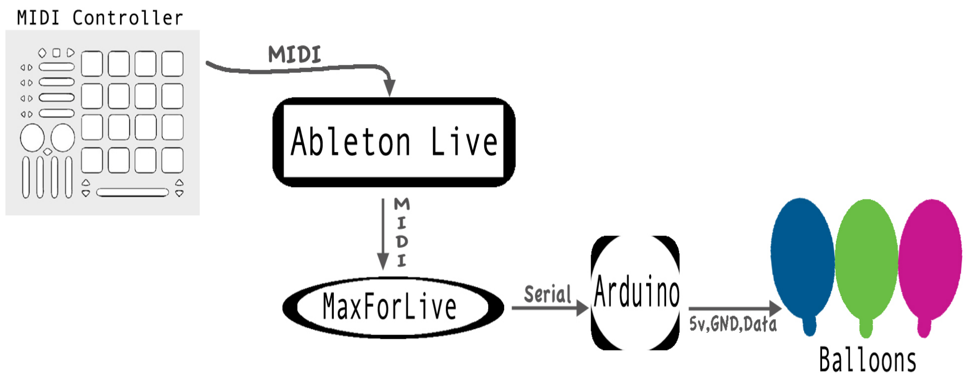


Figure 34. LightBalloons signal flow

Chapter Summary

Outside of mapping strategies, hardware and software, comprise most of the rest of a digital musical system; the two bookends on the musical shelf. Although how a musical system is improvised is for the most part dictated by how its inputs and outputs are configured or routed through mapping, what is actually improvisationally possible lies in the inputs and outputs themselves. What aspects make a piece of hardware or software more improvisational is the underlying question of this chapter. It is a personal suggestion that having them able to provide more functionality than is possible to harness at one given moment in time, or even throughout the course of one set, only serves to increase improvisational potential (assuming of course there is no physical constraint or requirement to activate all functionality at once). Human cognitive and physical constraints can slowly be overcome with time, which allows a performer to grow and develop with their system. Implementing a cap on control should not be based out of human constraint, instead, it should reach the physical and virtual constraints of the hardware/software itself.

The software for Monome, MLR, stands out not only as its own performance program, but as its own approach to electronic performance, by combining techniques not available before digital means, into one modern hyper improvisational system only made possible through the processing ability afforded by a computer. These techniques, such as quantized buffer shuffling, and complex controller-software mapping, input more abstracted conditionally which require less direct input into a system that self perpetuates to some extent. This lightens the mental and physical load by allowing performers to improvise in more macro arenas of their performance– as if they were also an improvisational conductor-composer. To this effect, performers are no longer constrained to constantly inputting energy into their musical system; the system can perpetuate on its own, with improvisational user-tweaking and modifying. MLR is a great tool for exploiting this more macro-style tweaking and modifying kind of digital performance, as it is designed to loop tracks, even more abstractly, events, without requiring energy to be placed into every sonic utterance. MLR, however, is only one flavor of improvisational computer performance, providing little control over the effect-level of performance. Where it lacks, an all-inclusive performance DAW such as Ableton Live tends to excel, things like robust effect-level manipulations, and really detailed audio clip manipulations. A merger of the two performance systems stands to create a much more improvisationally capable and complete system, throughout all levels of a musical performance spectrum. The program LiveMLR demonstrates exactly that, allowing the improvisational electronic performer a feeling of more complete control at his fingertips.

Lightballoons is an example of a hardware visual interface designed for live control along to music. It is different from most interfaces in that it focus' on the output end of an interface, consisting of individually controllable RGB LED-filled balloons; a sort of visual output equivalent to a synthesizer, or sampler, as opposed to an input-based hardware controller that normally controls outputs. Because of this, rather than exploring which types of sensors make for better improvisation (which is also an interesting subject), this section analyzes how a piece of digital hardware can be built and programmed to be more suitable for improvisational control. To this effort, LightBalloons has been programmed to be able have complete manual one-to-one control, as well as more abstracted macro score-level type control from sequences and live looping. Standing back and letting the system run it self while subjecting subtle live improvisational score-level fuel to an already fueled fire (rather than only utilizing direct manual control,) has been a digital advantage/technique at the crux of this work. To yield it in a visual domain such as LightBalloons has manifested the epitome of an opportunity to explore, study, and perform with digital media in a modern improvisational arena. Additionally, rigging it to run it through a performance DAW, Ableton Live, allows for an increased amount of functionality, and a relatively familiar and easy musical approach to improvising with visuals.

Chapter 5

Hands On Research

Ideas, concepts, research, conjectures, and the like can appear perfect in theory; it is only through their application in practice that one can properly evaluate their merit and practicality. This chapter is a performance journal reflecting and analyzing the technology techniques and ideas described throughout this thesis. From scrutinizing a series of performances, crucial problems can be ascertained and addressed, an experiential synopsis of what ideas actually do and do not work in practice can be undergone, a path for future progress can be further honed, and ultimately, more advanced and accurate concepts and conjectures can be formulated (for further testing).

Sonic Original Improvisation

Machine Orchestra

Performing with The Machine Orchestra has cultivated some of the most professional and elaborate performances I have had the pleasure to be a part of. The experience has provided me the opportunity to play alongside and work with some of the most prominent figures in Music Technology: Trimpin, Perry Cook, Ajay Kapur, Curtis Bahn, Owen Vallis and Jordan Hochenbaum (Flip Mu), Neelamjit Dhillon, Charlie Burgin (Sahy Uhns), to name a bunch. Despite an inherent form or structure within the body of work, most pieces were highly predicated upon improvisation on every level, e.g., hauntingly beautiful note-level solos of various traditional Indian instruments and hyperinstruments, effect-level improvisation on electronic drums and acoustic bass, and score-level improvisation via networking and robots. I had the opportunity during my stay at CalArts to play three shows with the Machine Orchestra, one on campus at CalArts' Modular Theatre, one at downtown LA's REDCAT (part of Disney Hall), and one in Savannah Georgia at the Tellfair Museum.[102]

Figure 35. The Machine Orchestra at REDCAT Theatre, 2012

5.1.1.1 Modular Theatre, CalArts

This Machine Orchestra event was a series of student led pieces at CalArts' impressive completely reconfigurable Modular Theatre, marking my first performance experience with robots. Most of the pieces were fantastically composed, arranged and performed. The two composed by two of my classmates, Mohammad Zareei and Colin Honigman, and myself, were probably the most improvisational. Both pieces involved me engaging in improvisational clip launching of the various loops and arrangements for robots, which were accompanying the synthesized drum melodies and chords of my (human) classmates. For one of the pieces, all three of us were using a three-dimensional arcade-style joystick controller (for X, and Y) who's handle detached and unraveled a long fishing line (for the third, Z-dimension). We mapped these controllers to parameters of digital synthesizers that we had recently built in class; because the fishing line (Z-plane) was almost impossible to see as we raised the handle up and around, our performance with these gestural controllers visually amounted to a piece with three guys improvisationally flailing about. Although it was a fun piece, this type of embodied gestural control felt more awkward the more I got into the music, which always served as a sort of

[102] A reading of the Modular Theatre and REDCAT performances are a sufficient account of the improvisation undergone for The Machine Orchestra, as Tellfair performance improvisationally was basically a combination of the two.

hindering reality check from getting too immersed. I think with better mapping strategies, the improvisational performance of the controller would have been more effective, by limiting motions of extremely jerky, embarrassingly uncomfortable spasms. In regard to Human-Computer-Interaction, watching the concert afterward was definitely a learning experience of what not to do via mapping.

5.1.1.2 April 12th & 13th 2012 – REDCAT Theatre, Los Angeles

The Machine Orchestra at REDCAT was an major production, with electronic musicians, dancers, composers, technologists, set designers, lighting designers, visual artists and of course, robots! The show was a well-rehearsed, multi-media event with strict cues, planned sets, etc., but within the pieces, there was ample room for individual improvisation amongst the large ensemble. The various augmented instrumentalists played beautifully mesmerizing solos, the electronic drums and robots were always fresh and interesting. I, being the bassist, had my Monome mapped for note-level control of various synth bass'. For this show I was also able to augment my upright bass, by running it through Ableton Live for effect-level processing. Primarily playing upright bass for this performance gave it a resemblance of my earlier days of performing less electronically driven music. It was a great opportunity to explore my previous performance methods and techniques through a more technologic filter. This performance brought about experiential lessons in improvising within a large ensemble, as in how to improvise with subtlety, and knowing how/when to blend in, and when to stand out.

Grids Beats Groups – Roy O. Disney Hall, CalArts

Grids Beats Groups is an ensemble class taught by Jordan Hochenbaum and Owen Vallis (Flip Mu) at CalArts dedicated to exposing and teaching electronic music performance techniques, usually–but definitely not always–with computers, making (usually but not always) tonal rhythmic music, in groups, as the name suggests. Each semester's class culminates with a performance given by the class' various groups. All types, styles, techniques, instruments, and ideas are welcome, so long as they tackle issues that modern audio/visual performance artists encounter.

5.1.1.3 April 25 2012 – Boma Happy

This first Grids Beats Groups show was an exploration mostly into improvisational clip launching, both for audio and visuals, performed with a classmate Tomio Ueda, together under the name Boma Happy. Tomio took the drums live, and I took everything else (audio and visual-wise). The controller mapping of Tomio's drums was especially unique, and contributed an excellent technique for improvisational electronic performance. He used Ableton Live, but had a group of audio clips mapped as a drum sequencer onto his grid controller,[103] a Novation Launchpad. This in itself was usefully unique. Live sequencing as a performance technique is a fun one, but one of its major drawbacks is that it is hard to make any macro changes over all of the sequences, as each note from each instrument has to be manually assigned. Somehow, Tomio had a page on his Launchpad to sort through some macro presets of sequenced parts/arrangements, for changes to new sections that he could improvise from there. This enabled him to produce a diverse improvisational blend of sequencing and clip launching.

My setup was more traditional for Ableton Live, with audio organized in clips and launched as groups (called scenes), utilizing whatever improvisational built-in features I could exploit within Ableton (like their beat repeat, legato mode, etc.). Ableton clips were also created to send MIDI to visuals projection mapped from Resolume, as can be seen in Figure 36. As a group we gelled well; in our 12 minute set there were some dull moments, and some really exciting ones; I definitely remember the exciting moments being spontaneously invigorating ones; the dull moments tended to crop up when feeling like just clip launching was not enough of an improvisational resource. Two overarching, somewhat opposing bits of wisdom were gained form this performance, but when combined instill a nice balance. They were (1) having more improvisational options at my potential disposal will make for a more improvisationally-driven electronic performance, for myself, and therefore hopefully for the audience as well; and (2) despite an instinctual need to always be controlling something at every moment while on stage, it is ok to have a system where musical energy perpetuates on its own, where one can just stand back and impart few and subtle user modifications if desired.

[103] As opposed to a more common built-in MIDI sequencer

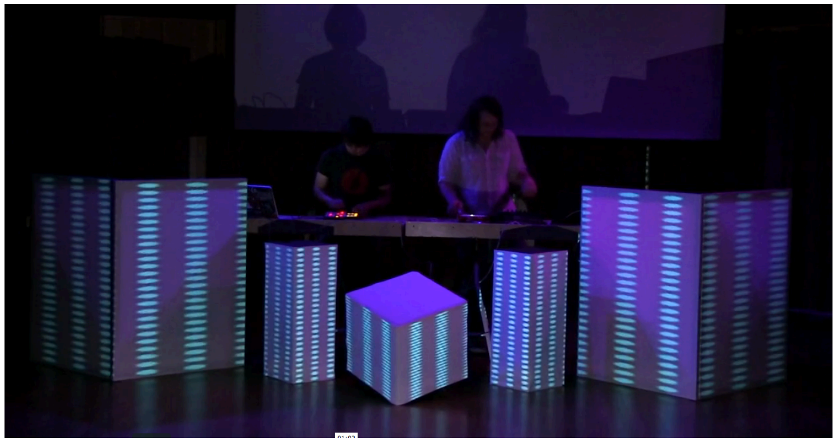


Figure 36. Boma Happy at CalArts

Figure 37. Grids Beats Groups at CalArts 2013

5.1.1.4 December 4th 2013 – Window Licker

This concert comprised three wall-to-wall tables of the most gear I have ever played alongside, let alone ever seen in my lifetime! Most of it was computer controlled, with less of it older digital gear, and a couple analog devices, such as modular synths and turntables. There were even hybrid groups that mixed computer music with non-electronic instrumentation (guitar, saxophone). A portion of the total setup can be found in Figure 37. For the most part, the night was chalked full of differently improvised electronic music, ranging in style, feel, sounds, technique, dynamics, and what not.

For this concert, I led a 7-piece computer group to recreate live the song, Windowlicker, by Aphex Twin. The intention behind this project was to see how a large group could work together to perform an elaborate production from the mind of just one person. The goal was to explore methods of improvising together while still trying to sound like one unit. Some interesting techniques were applied.

One of our seven members was the effect-level improviser of the group; he actually used no audio output from his computer to improvise with, not even effects. Instead, we created a network amongst our group to which he would map a few of his controls to the rest of the group's audio. Some controls mapped from the effect-level improviser to everyone to effect the group as a whole, and some to each member to effect one person's sound at a time. It was a unique feeling to have part of my sound controlled by someone else with no way to control or prevent it; it did create a sort of mind-like operation amongst the group–giving a sort of first-hand validity to extemporaneous aesthetic of groups like the League of Automatic Composers and The Hub. Probably because it was a new fun and interesting technique for the effect-level improviser, this particular execution felt as if there was more effect-level processing than needed; but nothing more practice could not iron out.

Another improvisational aspect worth noting was based out of the re-triggering of live-recorded vocal samples. Essentially, I routed the three vocalists (myself included) through my computer, so I could improvisationally live record small chunks of our three-part harmonies throughout the first three or four sections of the song while singing them to store for later. During the bridge of the song I then had interesting bits of random audio samples to re-launch and effect-process in different improvisational combinations. This tactic–a technique never even thought up until the need arose–went to show that there is a lot of improvisational creativity that can come out of live looping strategies.

Overall, it was hard to get seven to act as one, especially when an end goal is something so specific as the recreation of the song, Windowlicker. This experience of leading such a large digital ensemble brought about many unforeseen difficulties, even one as simple as finding enough group rehearsal time to pull the song off. Also out of working in an improvisationally rich ensemble, for better or worse, I was also forced into (not only) accepting (but encouraging) that the outcome of a large "group mind" will rarely be in accordance with my exact expectations.

Hamsterstyle – CalArts Coffeehouse Theatre

Hamsterstyle is an ongoing outlet for exploring improvisational group DJing; how can a group setting be used to make DJing more live, fresh and spontaneous? The simple manifestation to that concept was a quick rotation amongst a large group of DJs (rosters provided on flyers in Figure 38 and Figure 39), in which each performer would trade off (roughly) every other song, forcing the next performers to base their song choice in some way from the song before them, play one or two songs of their own, then trade back. The idea being akin to the way jazz musicians will trade off after specified amounts of bars, and "vibe" off of what was just played previously. There were always two or three DJs up at time (depending which show), and the sets were staggered so everyone got a chance to "vibe" off everyone else's selections. The only other (loose) rule was not to prepare too extensively, so as to encourage being free to get caught in the whim of the moment, rather than being stubbornly stuck on playing certain tunes that might not enmesh at all. Although there are those that agree DJing has some room for effusions of fancy, improvisation in DJing can be tricky and needs to be subtle, as its usual goal is to keep people dancing that are not necessarily interested in consciously evaluating and judging what type of manipulations are being controlled live. So Hamsterstyle is an iterative project of trying to strike the best balance between following traditions and creating something un-duplicatably new and fresh.

5.1.1.5 March 21st 2013 – Hamsterstyle pt. 0

This was the first installment of Hamsterstyle about a year before this writing. There were three performers up at a time, and all performers eventually routed to one multi-channel audio mixer. Methods of DJing included CDJing, Laptop DJing with Traktor, and MLRing by myself under the moniker, Merbert Moover. At this time, MLRing was the only electronic style I was comfortable enough to perform with. Probably the biggest bit of personal wisdom gained which became abundantly clear during this performance was that the heavy improv-oriented beatmashing characteristic of MLR was not ideal for basic DJing. [104] Because little improvisational beat shuffling was actually carried out, MLRing just felt like a waste of buttons and setup time for this particular type of performance.

Figure 38. Flyer for Hamsterstyle pt. 0

Because many of the performers this night were fairly new to DJing, and because all were new to this method of group DJing, and due to lack of proper audio routing, there was little accurate beatmatching successfully achieved this time around between DJs. Transitions between songs generally involved fading down one DJ on the mixer, while fading up the other. The audience was either oblivious or forgiving, so although imperfect, the night still progressed well.

Overall, there was not much improvisation within the songs being played, but the staggered rotation that comprises Hamsterstyle fostered a DJ experience that was unique, new, extremely collaborative, and arguably more fresh and impromptu than seeing one person DJ for a long time.

5.1.1.6 February 27th 2014 – Hamsterstyle pt. I ; March 6th 2014 – Hamsterstyle pt. II

The next two installments of Hamsterstyle were a series, one week apart from each other. Things that worked from the first show were kept, the rest dropped or refined. Because a consensus from the first show was that the set lengths did not seem long enough, two artists (instead of three) dueled it out on stage at a time, generally falling into a pattern of trading every two songs. This had a much better sonic flow, as well as cut down waiting time for each

[104] Of course Daedelus proves as a great counter example to this statement.

performer on stage to have his/her turn. The choice of gear for pt. I was the same as pt. 0, except I used the DJ program, Traktor with a QuNeo instead of MLR, which felt much more appropriate. The third showing of Hamsterstyle, however, a last minute decision to use CDJs as my performance tool brought about my most significant DJ performance advancement. This was my first time ever playing CDJs live, and with not more than 30 minutes of CDJ playing experience/familiarity–15 of which were during the sound check of this show. Not only do CDJs seem to provide the perfect amount of performance engagement for DJing (without leaving too much room for excessive tweaking), but it also seemed to be the easiest method with which to listen, feel and react in the moment to what was currently transpiring. Ultimately, a major lesson of having the proper tools for the job makes a vast difference, i.e., an Monome-driven MLR setup might garner great result for solo electronic improvisation, but might be far more confining than CDJs or turntables when group DJing.

In order to make for a more seamless and traditional DJ performance experience, a major revision for part II and I was a decision to place more of an emphasis on beat matching between performers. To make this possible we made sure each DJ was routed into one channel of a DJ mixer, which made cuing a feasible operation. In theory, and with a tad more planning, this would have worked perfectly; in reality the limited inputs on a small DJ mixer caused hurdles when laptop performers entered the mix needing to unplug and re-plug their own equipment. A couple mishaps occurred where audio cut out from plugging things incorrectly, but nothing so serious as to derail the events. Another major addition to the second two showings that seemed to heighten the experience was the inclusion of improvised projections/lighting, by James Hurwitz and Suzanne Kite. There was not necessarily a steady progression of improvisational ability or live-ness from each consecutive showing, partly because of the skill in balance that needs to be placed into improvisational DJing while still keeping people dancing. The experience gained did make each night a progressively more sleek and flowing performance experience, which ultimately promoted mental freedom to devote to improvising. The Hamsterstyle experiments clearly showed that there is room for effusions of fancy in a DJ setting, which will hopefully continue to be further explored.

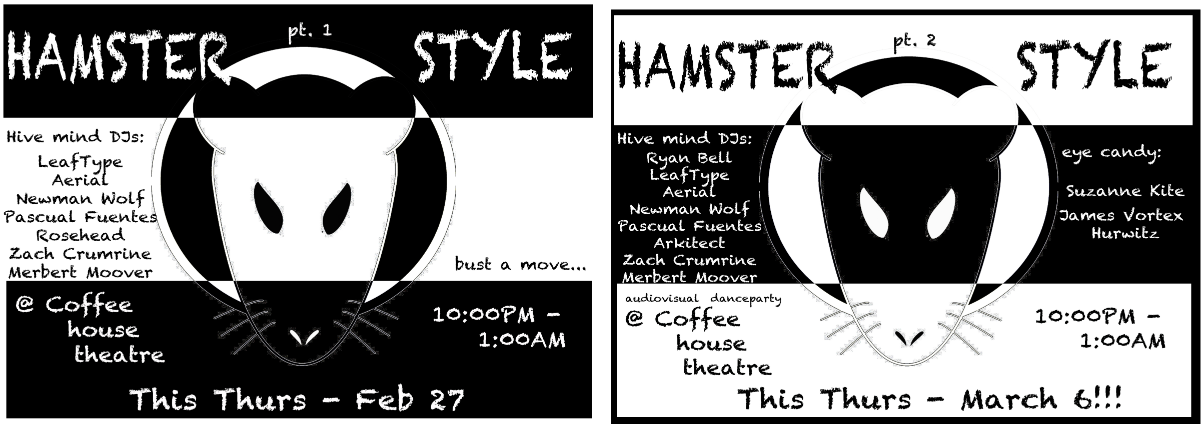


Figure 39. Flyer for Hamsterstyle pt. I (left); Flyer for Hamsterstyle pt. II (right)

March 20th, 2014 - In Relation To The Universe, We Are All Sleeping In The Same Room, Black & White Gallery - CalArts

I was asked to play this show for friend and fellow musician/student, Hill Jaeger, as a follow up after her recital, to keep people dancing and having a good time. As it seemed like an opportune outlet to test some experiments related to this work, I accepted. The benefits of this performance were manifold, (1) it provided an outlet for an up-to-date experience with using MLR as an improvisational tool for live performance (as it had been quite a while since my last MLR set); (2) it served as a control for comparison with a future performance using LiveMLR with similar audio material; and (3) it was an experiment in its own right at combining MLRing with CDJing for a more live and improvisational DJ setting.

Expanding on the 3rd benefit, my equipment for this set was fairly unfamiliar. Firstly, I doubled my Monome buttons by using a second newly self-built Monome, which I combined in software into one 8x16 Monome; 16 subdivisions of buttons per sample instead of 8 for MLR felt much more comfortable, appropriate, and free for spontaneous exploration within each sample. It also doubled my function row, enabling me some extra helpful functionality like switching presets, and changing octaves on my Monome instead of with my mouse and laptop, requiring less need to focus on my computer screen. The extra ease and intrigue I experienced with extra buttons in a sense somewhat confirms that more control can amount to more performability in improvisation. I now have no interest to revert back to a smaller 8x8 grid. The second major addition, which altered my performance completely, was the incorporation of a CDJ along with MLR. On a continuum from strictly DJ to strictly original, because this performance veered slightly more toward the DJ side, the test was to see how performable, improvisable, and enjoyable (for myself, and the audience) it would be to mix MLR and CDJing (for my second ever CDJ performance).

Figure 40. Flyer for Hill Jaeger's Graduate Recital

The MLR/CDJ experiment, for a DJ type setting was a major success in many ways. MLR is very live, but to improvise with it well requires using fairly short samples (of about 1-8 bars). Unless one is constantly button mashing, these loops can quickly grow tedious to listen to, and it was already made clear from Hamsterstyle that constant button mashing is not very stylistically compatible with DJ-type performances. Switching back and fourth between MLRing and CDJing in a sense actually proved to be more live and fresh sounding throughout the set than just MLR alone, allowing for alternate techniques, and hence, variety in sonic flow. Also, because CDJing had become a newfound hobby at the time of this performance, the task of beatmatching was one to look forward to. Beatmatching between a CDJ and MLR was slightly trickier than between two CDJs, which was not conducted perfectly every time during this performance, but it was usually easy to get back on track. Even more fun was MLRing along with the CDJ track once beatmatched. Though looking back it seems to have a connotation of performing with backing tracks, it was fun to just play the lengthy audio material through a CDJ, and improvise along with MLR. Another bonus is that CDJing allowed me to perform with audio material that would normally not be conducive to the short looping of MLR, hence it enabled me to expand my repertoire. Had this show been a setting for more original-type performance–a setting where people scrutinize a performance more closely–this setup would have probably come across as sounding boring to the audience. From a DJ perspective, however–a setting where people are forgiving as long as you make them want to dance–this setup definitely stepped up the improvisatory connotation to the set.

Incorporating CDJs was also a saving grace for a lack of enough preparation.[105] Unfortunately, to play MLR requires a lot of prep time spent building a bunch of presets. I was able to make about 15, which even when interspersing CDJing, I managed to play through just about all of them in roughly 40 minutes (out of a set length twice that). Although my set was fairly seamless and cohesive, I was improvising with MLR less unabashedly than I intended or would have preferred. This was due in major part to the other type of preparation lacked: a sufficient amount of practice time–as MLRing *is* playing an instrument, it requires practice to sound decent. A considerable performance latency from rewiring MLR into Ableton, combined with tricky-to-use longer audio samples instilled a caution toward apprehensive sloppy button mashing, something that could have been addressed with more practice time.

From an audience perspective, it is hard to say how much adding live-ness to the normally not very intensive technique of DJing really heightened the experience. It is also probably not wise to base such an in-depth question on one particular show, especially one as lightly attended as this one (probably 30 people at any given time). One guy was standing close but off to the side, half dancing wildly, half mesmerized by my MLRing, trying to understand what I was doing and how, which was gratifying enough. Similarly, a fellow student and friend that had never seen me MLR before was pleasantly impressed, and was standing close enough to ask me questions while playing. Another girl afterward told me she really enjoyed my set, and that she had not danced that hard in a long time, although I suspect she enjoyed my choice of music played more than my techniques utilized. Overall, people seemed to be in enjoyment, and it was arguably one of the more fun equipment setups I have played with. This experiment confirmed that having more of a variety of control options (in this case an expanded MLR system, and CDJ system) can lead to more of a variety of improvisational control, which brings about more intrigue for the performer, and in turn, the listening audience.

[105] This show happened to coincide just after another event, which demanded almost two weeks of working until 6:00AM to install and control a 10'x10'x10' LightBalloon Pyramid.

Visual Improvisation

July 19th-21st, 2013 – ConCon 2013: The Controller Convention,

MIA Gallery – Los Angeles

ConCon was a convention highlighting home-built audio controllers and hardware, and audio visual art performance and installation. Three days were filled with showings, lectures, panels, demonstrations, discussions, and three nights were filled with live computer musical performances and live improvisational digital visuals by myself. Daedelus, Moldover, and plenty of California music technologists came to contribute, support, network, and hangout. Though I did not have a home made controller ready to perform with at the time, I was inspired by the live visual control I had created and performed from Boma Happy's GBG performance, and wanted to start applying electronic audio techniques through musical MIDI controllers to improvisationally control projected visuals. ConCon was a great outlet for this exploration.

The equipment included Resolume as the video clip handler/manipulator, some programmed Quartz patches, found footage, and blank colors, all being controlled by QuNeo, and also to a slight degree, Ableton Live. Ableton set the tempo, sent MIDI sequences and different MIDI LFOs mapped to Resolume effects; the QuNeo took care of the rest, which was a surplus of effect control that took up the mapping of its every sensor. To remind myself what visual effects I was capable of producing, it was really easy to exhaust every control I had mapped within the first set or two, and feel as though there was nothing new to perform. Some extra effect mapping, programming, and clip retrieving was necessary each day before the next show to provide some newness to each night's visual show. Eventually, what really made things gel was to spontaneously settle on some creative "effect theme," or hook at the beginning every song, subtly modifying, changing, enhancing, improvising upon said theme throughout each song's duration. Once I came to the realization that I could use visual "hooks" and "riffs" within certain themes of effects and clips, I could better improvisationally develop these themes and maximize the provided control I allotted myself, instead of barely testing out each idea before moving on. In this sense, improvising subtly and thematically gave me more improvisatory control with which to utilize and develop throughout a piece.

These ConCon performances re-affirmed the importance of efficient mapping strategies; For example, I discovered via this performance that it is rarely advantageous to map two different effects to an X and a Y of an XY pad unless one knows he will *always* want to blend those effects. Otherwise, trying to use one effect without the other will result in playing along the edge of the XY pad (so as not to trigger the other effect), amounting to an awkward and poor utilization of an XY pad. My MIDI tool, MIDISmartToggle came in very handy for this type of visual control, the extra custom Z-toggling provided me with a lot of extra creative options for using an XY pad by toggling certain parameters at specified depths. Simplification in performance was added through complex mapping, without sacrificing any performable control whatsoever. Overall, improvising purely with visuals along to music was a gratifying experience; one that conceptually led the way to many future iterations.

Figure 41. Flyer for ConCon Convention 2013

January 27th, 2013 – LightBalloons @ Control Freqs, Ham & Eggs Tavern, Los Angeles

Figure 42. Flyer for Control Freqs

Control Freqs marked the first live controlled improvisational LightBalloons performance. Having used the interface successfully once prior in an automated and audio reactive showing, and having already experienced an improvisational visuals set for ConCon, this time I got to "play the balloons" live. For this show, a strand of 20 LightBalloons were filled with helium and configured to closely resemble a circle as possible, sort of like a halo over the performers (see Figure 43 for pictures). The output was very different from a projected visuals set like ConCon, but the equipment used to control LightBalloons was fairly identical, mostly a QuNeo running into Ableton Live, and some Ableton tricks converted to MIDI, like an arpeggiator and a robust live data-looping system (inspired by MLR) built with the aid of Clyphx. The live looping proved invaluable for this performance, as it allowed me to improvise something visually catchy once that I could then repeat indefinitely, giving me a foundation to improvise on top of further (very similar to MLR), as well as time to stand back and carefully (yet spontaneously) think of my next moves.

This being the second iteration of a LightBalloons project, many hurdles were still being overcome. The worst of which was probably the high maintenance in setup, both in time, and difficulty. With this version, tying off each balloon with a thin string took patience, dexterity, strength, and apparently blood and blisters,[106] and despite such a painful setup, little could be done to avoid balloons deflating at different rates and becoming incongruently different sizes (looking closely at the center picture in Figure 43, you can see one light blue balloon upside down and significantly deflated). Future versions of LightBalloons have alleviated this issue with updated hardware.[107] Although instrument setup has little to do with actual improvisation during a performance, it is worth noting its tedium here in that if efforts had not gone into revising the instrument's difficult setup, the desire to actually use the instrument would soon prove more of a hassle than it was worth performing, and would cease to prove useful for improvisation, research, or any other matter.

This performance felt exactly as it was: like playing an instrument for the first time. To my advantage, by this time I had already much experience in playing new systems for the first time (for better or worse, this seems to be a recurring theme for digital electronic music performance). Unlike the initial "squawks" and "squeaks" that usually come from playing more

[106] Though barely, this was acceptable, as my experience playing upright bass assured me that any good instrument requires some sweat and blood to reap its rewards!
[107] As in the clear plastic tube method mentioned previously.

traditional instruments for the first time, the digital mapping and programming of LightBalloons made it easy to obtain pleasant sights right from the beginning, which allowed me to easily grasp the potential of what the instrument could become with more familiarity, design, and effort. Nonetheless, the set was nothing but improvisational, as no rehearsal or foreknowledge of music being played left any other choice. For this show I was basically starting with a blank canvas; perfectly forced right into an improvisational setting.

"Playing the balloons" being a new experience for myself, I was eager to show off what my interface could do, and like my ConCon performance, I exhausted my capabilities early on. I attribute this merely to amateur-level performance, something that with practice (and more variety in control) will season over time. After performing for about three 45-minute sets I took a short break and let others have a try. Only one–a fellow CalArts graduate–seemed able understand my mapping and produce beautiful light patterns, which goes to show that mapping can be a very personal enterprise, and grow quite advanced when one understands what is possible. Upon returning from my break, I remembered what I learned about live visuals from ConCon–something which seems applicable to most audio improvisational experiences: picking a theme, and drawing it out with subtle ornamentation–as if I were picking a chord or scale and playing a hook within it, then improvising upon the essence of that hook–was when captivation and enthrallment set in, and creativity abound. This then allowed me to develop portions of sets over time, giving viewers themes to latch onto/relate to. Additionally, it was when more sequential automatic modes that required less control and more tasteful modifying displayed sequences physically beyond what is manually possible, that the most awe-inducing and beautiful lighting came alive. The realizations just mentioned really expanded my notions of improvisation, and solidified my awareness of how improvisation can be evolved though digital means. Most fascinating is that these improvisational concepts seem as though they can apply just as easily to the audio and to the visual spectrum of live performance.

A paradigm not exactly transferable from the digital musical performance realm to the visual one is where the performer should be looking (or not looking) while on stage. The common trend in digital audio performance is having functionality exclusively on a controller, leaving little-to-no need to gaze at the computer screen. When dealing with a visual instrument like LightBalloons, not even considering the computer screen, an interesting problem arises in that when looking down at the controller to push the correct buttons, the performer cannot see the visuals being produced; and when looking at the visuals, it cannot be assured that the intended buttons are being pressed. It seems perhaps at least for visual improvisation, that the next evolutionary trend might be a need not to even have to look at ones controller.

As an experiment on improvisation, and as an actual instrument in development, LightBalloons has been an invaluable case study/pastime.

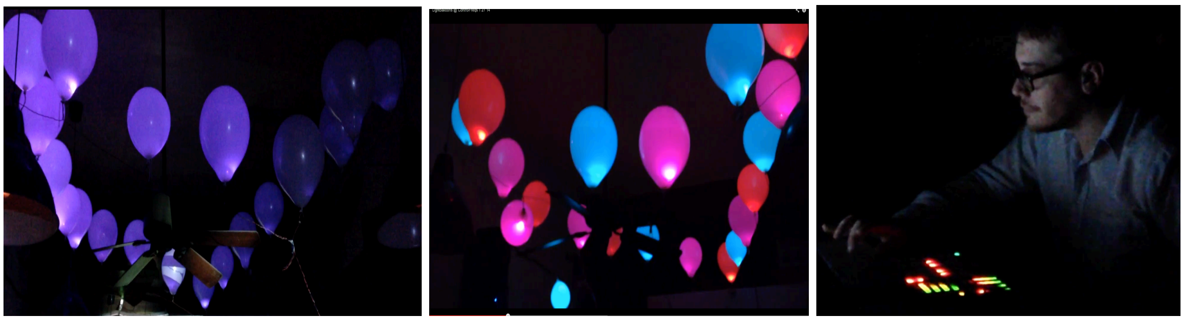
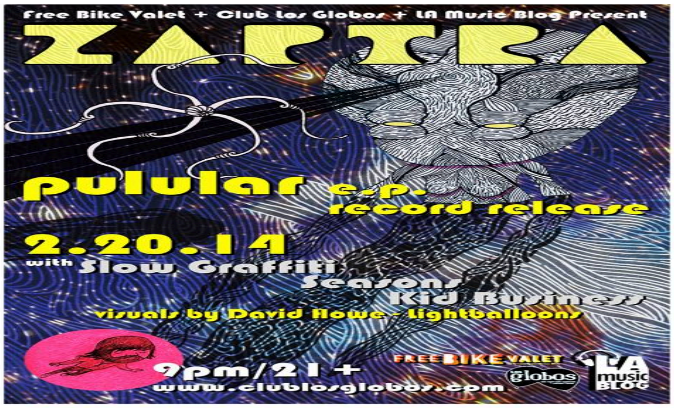


Figure 43. LightBalloons @ Control Freqs

February 20th, 2014 – Zaptra EP Release Show, Los Globos – Los Angeles

The Zaptra EP Release Show was the most recent live controlled LightBalloons performance at the time of this writing. The main goals of this show were to hone and exploit the improvisational realizations, and to refine the major issues encountered from Control Freqs and other previous improvisational performances. Because my setup was the same as before (for once)–20 LightBalloons and a QuNeo–I was able to build upon my previous work, and was prepared with more tricks. Compared to Control Freqs, this was a much larger production, at a much larger venue, with a larger drawing artist, Zaptra (Art Paz), for an event to which I was the art director. Zaptra wanted a visually astounding accompaniment to his electronic music, and asked me to play the balloons, as well as oversee some projection mapping. The visual aesthetic for the show (which can be seen in Figure 45) consisted of my LightBalloons in an attempted vertical triangle around Zaptra while he played, with me next to Zaptra just outside the balloon triangle (one audience member was certain it was a balloon *heart* for Valentines Day). Two boxes off to the side–stage right (they can sort of be seen to the left of the left most picture in Figure 45) were projection mapped by a friend of Zaptra's, Nisa Karmsomport, who was very interested in the idea but needed my tutoring to learn how to actually do it.

During the 20 minutes of balloon setup time before the set (in between bands), my biggest LightBalloons fears manifested. When first plugged in and tested, only the first 7 LightBalloons (out of 20) were working. As can be common with Neopixel LEDs, I was sure the 8th one had blown up, and the interface was done for until further repair (which has happened before). I was ready to give up and not play the balloons that night. During start time, Zaptra gave me one last bit of encouragement before he started, helped me plug into a different power source. I re-flashed the program on my Arduino, and the whole interface began to work properly. …And the show went on. This all served to remind me that revising the hardware was necessary to make this project a lasting usable instrument.



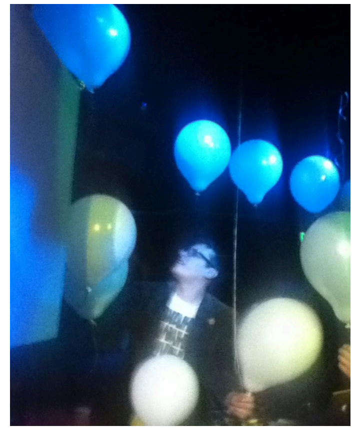


Figure 44. Zaptra EP Release Show Flyer (left); Zaptra Pre-Show Balloon Hang (right)

Thanks to weekly meetings with the artist, I was able to familiarize myself with the music I would perform along to. I was especially conscious of not giving every trick away in the first few minutes, so this time I decided to pick modes and themes to most of the songs in advance. Though it can be argued that the adding of confines allots less spontaneity, the more subtly developed improvisation within the themes I set about was more relatable, instead of all over the place. So rather than limiting total improvisational ability, it sort of refined it. Moreover it allowed me to really delve into what was possible with what I had programmed, and to stretch out what I had available to keep it new and fresh for every song. The few creative decisions I made beforehand made for a comparison between the first and second LightBalloons akin to a beginner improvising musician obliviously wandering about their instrument, versus a polished improviser, listening, reacting, and making the most out of as little as possible. In other words, the ideas adhered to and additional programming for this performance allowed for a more creatively tastefully improvised set, as intended. The concept of 'less is more' really hit home.

Due to a bug or imperfect coding, a serial issue of dropped MIDI message from Ableton to Arduino prompted more sequencing and automation hard-set right into the Arduino, as opposed to from outside, like from an Ableton MIDI clip, as intended. As has been touched upon from previous show analyses, these internal automated modes, however, worked great for improvisationally-minded performance, because there was not a constant need to continue putting energy into the system (like plucking a guitar); With this self-perpetuating type control, I was yielding one of the finest attributes of improvising within a digital system, by freeing myself to consider, contemplate, dwell in, listen and react to anything happening around me.

In addition to programming internal Arduino-run sequences, I was able for this show to build MIDI sequences as Ableton clips, which, as I suspected, is generally a far easier and more intuitive method for light sequencing than text-based coding. Using audio-related (and familiar) techniques like Ableton clips helped to feel more connected to the music, and so was a adequate compromise for incurring more dropped messages than desired (which visually amounts to a light not turning on/off or changing color when supposed to). Accepting these imperfections, I was able to very quickly amass a large collection of quite intense light sequences in Ableton, supplementing any extra needed variety for this show. This experiment confirmed that with some refinement in code, composing light clips in Ableton is the most effective and efficient compositional and improvisational performance tool thus far tested.

Chapter Summary

Many improvisational shows were performed during the course of this work. Every performance undergone during the course of this work an before has been jam-packed with learning lessons, as long as I have allowed myself open to the self-scruitiny of retrospect. This drive to constructively criticize one's own work is at the core of what helps a performer improve upon his craft. My performance has by no means reached its zenith; neither has Music Technology, nor even improvisation. It is without a doubt this attempted contribution to further perfect things/ideas/techniques/technology that makes the struggle worthwhile and entertaining. Of all the lessons learned over the course of this thesis, a brief collection of some of the more pertinent ones are summarized.

The Machine Orchestra shows were among the first that were digitally performed during this writing. Each in their own ways helped foster an understanding of what improvisational ideas transfer from an acoustic/electric realm to a computer-aided one. These shows initialized the careful attention I now pay to two major improvisational concepts discussed in this thesis: (1) That of subtly expanding upon themes instead of complete erratic impromptu improvisation; and (2) How important a role the visual aesthetic plays in musical performance (seemingly especially digital for some reason); for example, programming a controller that requires incessant flailing about to play a synthesizer could look unfittingly awkward when performed by three guys live!

The Grids Beats Groups and Hamsterstyle performances all involved different experimentations in group improvisation. Some setups worked better for improvisation than others, just as some performers were more improvisationally savvy than others. Specifically regarding group improvisation, there were not really any definitive lessons gained in summary. These shows did however serve to provide ample experimentation in a wide variety of musical systems. Collectively these group improvisational endeavors, as well as Hill Jaeger's event, forced me to continually rethink how and what I perform to best accomplish each outcome, demonstrating that there are seemingly infinite ways to approach improvisation in a digital arena. These performances have provided me with a lot to consider as far as designing my eventual all-inclusive musical system.

If the first few digital audio performances broadened my understanding of how improvisation applies and transitions to new mediums, my foray into visual improvisation magnified that scope exponentially, by applying similar and new improvisational tricks and seeing what stuck. And just like the digital audio transition, the early digital visual performances, like ConCon and Control Freqs reminded me–via trial and error–of the tasteful and pleasant outcome gained from developing themes more slowly, instead of erratically executing everything possible as soon as possible. As case studies, the visual improvisational shows also helped to crystalize a major advantage of digital improvisation by further exploiting the score-level in performance. In addition to and instead of more direct one-to-one control over note-level and effect-level parameters (amplitude/frequency/etc.), developing an automatically flowing and perpetuating system whose less direct, more macro (score-level) parameters can be tweaked and modified throughout its duration seemed to heighten digital performance. The Zaptra performance helped solidify this concept, in that I could step back and place more improvisational momentary forethought into what subtleties to develop next upon a system already perpetuating beyond human manual capabilities.

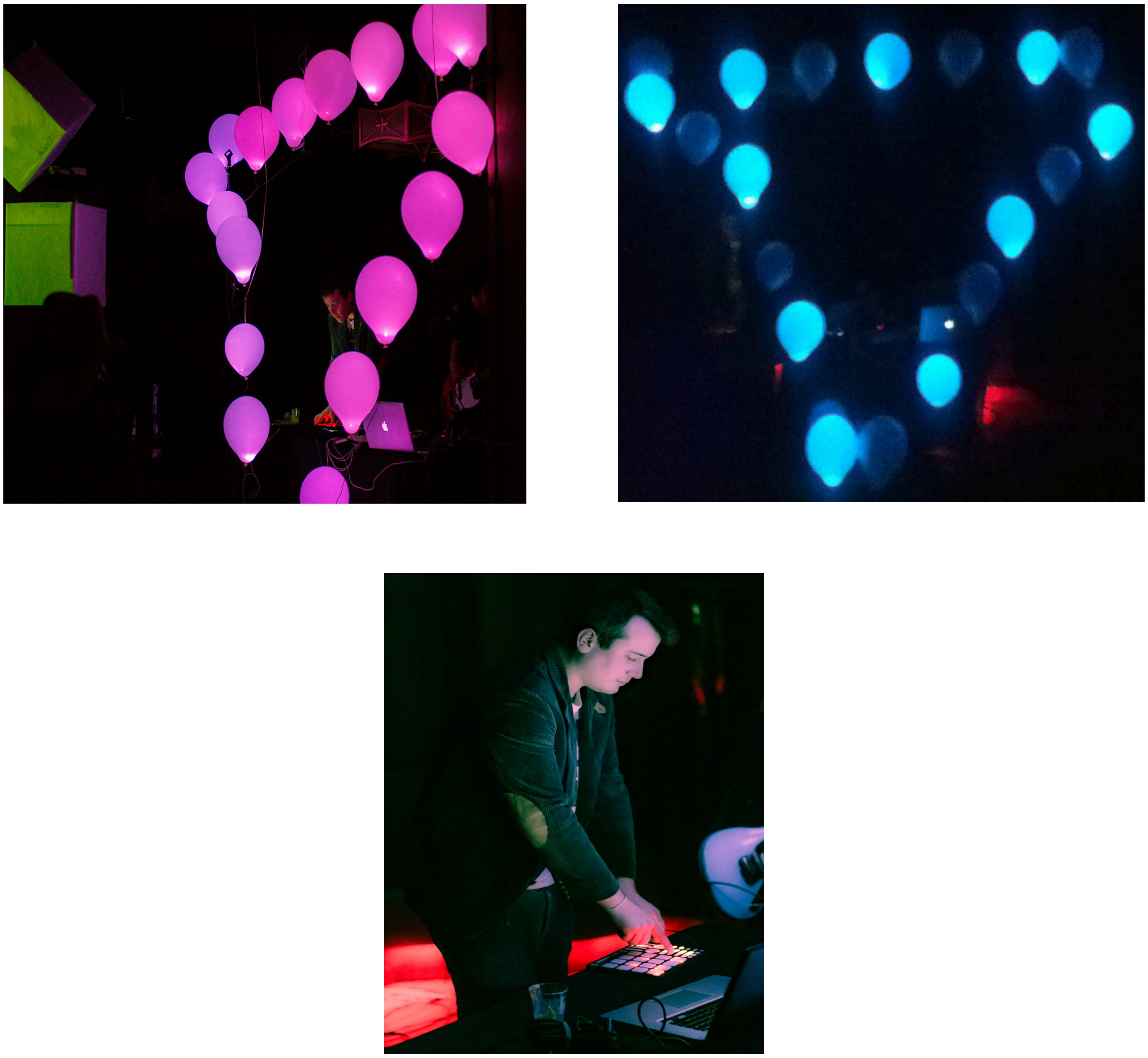


Figure 45. Zaptra Performance with LightBalloons

Chapter 6

Conclusion

> In the history of musicology, improvisation […] has played a minor role. Musicologists have been concerned in the first instance with composition, and less with the process than with the completed piece of music as set down by its creator. Affected by the research traditions of visual art and literature, they have concentrated on the finished work, analyzed the interrelationships of its components, and looked at its history, but rarely have they been concerned with the varying orders of creativity that may have led to the final product. Although it would be foolish to ignore the occasional reference to improvisation, there are few studies that go in the subject itself in detail, and thus it is reasonable to consider this form of music making as relatively neglected.

-Bruno Nettl, (Nettl and Russell 1998)

Summary

Though necessary for a most complete depiction of the wondrous pulchritude held for musical performance, detailed accounts of what where why and how improvisation has existed proves scarce. Out of the occasional references and rare improvisationally devoted works like this one, it is hopefully apparent that the act of improvisation itself is not as rare as its lack of research implies. Bouts of spontaneity are hardly documented, nor documentable; written word will never do justice to the actual experience of invigoration that improvisational performance can induce, like a sonic magic trick unveiling before our ears, giving the experiencer the notion that what was just so fleetingly executed could and would never be repeated again, at least not under the guise of improvisation.

This work was undergone to boldly analyze, categorize, quantify, and utilize improvisation, as it currently stands, entertaining questions like what it means to be an improviser in the digital age, and, how the digital age has reflected upon an such a timeworn idea as improvisation. Can modern improvisers now incorporate more musical rolls into their performance? If so, do these new roles provide an increase in musical options, or do they impose more required knowledge, experience, and restrictions in order to improvise at all? And, with all of these new options, is there such a thing as too much? Likely, these questions could and have been asked throughout the history of musical technological progress; as developments in new instrumentation have come about, performers have been afforded more improvisational freedom, yet in turn require a need for more improvisational prowess. Technologists and improvisers alike are continuously striving to increase what they are capable of musically achieving, which–especially in todays modern digital musical systems–amounts to a technological means of actualizing what was previously technically impossible. One argument of this thesis is that any amount of control that seems to surpass a mental threshold to be playable can, should, and likely eventually will be overcome though practice and thoughtful creative technological means.

A modern digital improviser can do more than freely articulate sequences of pitches, rhythms and timbres. The improviser is no longer bound by a predominantly note-level articulation; in the course of one performance, one can now dwell in any or all sonic roles, including composer, arranger, conductor, visualist, etc. (of course the list extends on). For the first time since the beginnings of Western Music the possible effusions of fancy since the inclusion of computers have been blown open, and potentially redefined. This work compares and contrasts, as well as explores these new possibilities, to give a brief account of the state of improvisation of today.

Primary Contributions

The work begins with a few terms, concepts and opinions to guide the reader throughout the document, followed in Chapter 2 by a brief, yet historically extensive account of improvisational development in Western Music throughout its history. This chapter paints a picture of what improvisation has been compared to what it has become, showing that although the technology has enhanced, improvisational approaches remained fairly stable until the inclusion of computers.

The next two technical chapters regard the electronic improviser's musical chain, and what can be done to enhance each part in it for more and easier control in improvisation.

Chapter three covers the importance of thoughtful mapping among one's musical system, and introduces some mapping strategies and tools for advanced digital improvisation. There is a strong emphasis that these tools can and should easily already exist, built-in, as a function of modern MIDI controllers. Mapping can be the key to creating a magnificently robustly improvisational musical performance, or can lead to a conveniently executed, yet severely limited hardly improvisational one. Chapter four delves into examples of how and why to design improvisationally suitable inputs and outputs. Mapping is an obvious complexity that computers have more control over than acoustic systems, and where most improvisatory control lies. Chapter 4 tackles what it even means to make hardware and software more potentially improvisationally suitable through technology. Provided as case studies are one software example: LiveMLR; and one hardware example: LightBalloons. LiveMLR constitutes a merger of two programs,–MLR and Ableton Live–as an effort to incorporate relevant and useful modern playing techniques into one all-inclusive-type system. LightBalloons is a visual instrument made for improvisational live control, focusing on approaching visual control in similar methods and concepts to musical control.

Chapter 5, then, utilizes and documents previous chapters' tools, concepts, techniques, and ideas through first hand experience, both in a audio and visual exploration, but all under the umbrella of improvisational advancement for technological musical performance. From this research and experimentation, a particular account of improvisation in the beginning of the 21st century is ascertained.

Other Discussions

Limits

One idea that tangentially pervades this work is the contemplation of virtuosity versus a jack-of-all-trades approach to musical improvisation. Of course, there is no better or worse route to take, only pros and cons to each. Virtuosity in musical improvisation is a rare skill to possess, and seemingly scarce when discussing virtuosic computer musicians. This could be in large part because "computers in music" is a relatively new paradigm for musical performance, therefore, virtuosity in computer music has not had the luxury of time to develop/define what exactly it even means. Probably equally if not more so, however, it is because of one of computer-music's most contrasting advantages (or disadvantages, depending on one's viewpoint) over acoustic:

there are so many ways to easily re-orient one's digital musical system that each desired outcome potentially requires completely different mapping of inputs and outputs and different methods of performance. Crafting a digital musical system then becomes a virtuosic art in itself, rendering the drive to become technically proficient on one particular setup less mandatory. Thirdly, the power to create/develop a digital musical system now residing within the scope of the improviser tends to be linked to the means to find digital ways to relieve physically and mentally painstaking challenging or demanding musical tasks.[108] So time spent practicing an acoustic instrument tends to get replaced with time spent programming a digital system to be easier to perform. Hence, as we evolve with Music Technology, perhaps the concept of what is virtuosic shifts gears. Of course, from experiencing traditional virtuosity in performance one understands in can lead to an emotive type of performance not otherwise attainable. This however implies the need to stick with constraints, or limitations placed upon the instrument in order for familiarity and full command to be overtaken. Personally, I prefer to explore new territories than achieve virtuosic results to limited ones. I am obviously a proponent of the All-Inclusive Approach, where the threshold of "too much" potential to create combinations of new sounds is exceedingly high, based more out of hardware and software constraints rather than mental or cognitive ones.

Final Thoughts

On Technology

Music Technology is constantly evolving, and very recently with it so is improvisation. Almost every show performed and documented in this work used different control sources, mapped extremely differently, to different outputs, which–as would be expected–had markedly different outcomes. One of the most incredible amenities afforded by computers with music is their ease in versatility. Because many of these custom musical systems are catered to particular performances they are often little used before being modified or completely changed, and because of this they can be difficult to master. One downside to this constant system redesigning is that analyzing just one of a few performances per system often left me with a continued feeling of, "I could have done better if I practiced more." Or, "I will do better next time now that I'm more familiar with the system," generally leaving a major lack of contentment or

[108] Basically through complex mappings.

gratification. An upside to this multifaceted musical approach is the ability to work in entirely new and inspiring ways, which can more easily lead an improvisation aficionado into exciting unexplored musical territories. Furthermore, the push to try different approaches and techniques, allows the digital performer to over time select the best bits and pieces to incorporate into future improvisational performance systems, continuously sparking more efficient, creative, unique, and beautiful ways of executing things, as opposed to the stagnant confines that can come from an acoustically fixed musical system. Finally, music technology is in a continuous state of flux, especially since computers. Now not only is it harder to support recently outdated technology, but newer technology often tends to exploit newer, more efficient, or different methodologies; for the improviser, seeking newness in improvisation is a useful advantage to stay on top of.

Do certain tools, approaches, or styles promote more spontaneity Or versatility? Can a piece of software or hardware be more improvisational than another? Can they evolve ways in which improvisation can be performed? Hopefully by reading up until this point it would appear plausible that the answers to all of these questions are yes. Digital mapping provides a means to improvisationally control things in newer more omnipotent ways, and so controllers and software with effective mapping functionality, or at least effective mapping middleware tools are at the core of what can further what is improvisationally accomplishable. Moreover, through digital technology, almost anything can be a sensor input, and the possibilities for interesting new virtual and physical outputs are seemingly endless, from robots to LightBalloons! This of course increases what improvisational tools are available to work and customize with.

Regarding the tools documented and used in this thesis, most serve to enhance improvisation somewhat indirectly by simplifying complex tasks and eschewing common technical confusion (and physical ability). Some of the more practical MIDI middleware tools were the 6D Push Button Knob, and MIDISnap, the first enhancing control, and the latter saving confusion. Though these tools have proven worthwhile by making improvisational digital performance easier, some were painstaking to create; and because they were middleware, some were not flawless in keeping both the controller and software in sync. If these functions were programmed into a controller's microprocessor, and already came for use as a function of the controller, there would be no issues with their functionality, a point Chapter 3 is adamantly insistent upon.

The next chapter includes the inputs and outputs of the digital musical chain. In an effort to spawn more techniques and types of control within one all-inclusive system, LiveMLR is a great start, merging the highly improvisational live software, MLR, with the exceptionally performance-friendly DAW, Ableton Live. LiveMLR is a merger offering the best of both worlds, most importantly, more sheer improvisational control over audio (and MIDI) clips. Lastly, LightBalloons has been the most frustrating, yet most gratifying contribution to this work. It has allowed me to stretch my limits of musical improvisation by creating an interface that has not yet ceased to astound both audiences and myself. Moreover, improvising within a visual realm has not only bridged a connection toward approaching improvisation amongst audio and visuals, but has completely broadened my scope of what improvisation is plausible and tasteful, whatever the medium.

On technique

As researched, documented, and experimented, there are plenty of modern performance techniques for the first time available through computer music. The evaluable differences between them do not really adhere to a hierarchy; they are equally viable approaches to improvisation. One question worth noting is, how much improvisation is too much? Is there a maximum amount of spontaneity or versatility that is aesthetically pleasing? These questions pertain to all of improvisation, but are especially pertinent in computer music because of how much more is improvisationally possible. I used to feel that when going into a solo electronic improvisational performance, doing so with any pre-planning of the flow of performance would hinder the improvisation possible. I continue to feel as though there is nothing wrong with having no preconceived notions before performing. One of the biggest realizations to come out of this work, however, is that some pre-planning into a set can actually help foster improvisation in that it provides guidance; for instance, if the improviser already knows generally what he is about to play, he can then put more spontaneous thought in to how to elaborate upon it. The performances where I was able to adhere to this idea, I managed to evoke more cohesive and catchy material, and more meaningful expression. Improvisationally speaking, this idea is nothing new; even jazz groups have their chord charts and key centers to guide them. When trying to think in new improvisational ways, it is sometimes hard to retain that old traditions often last for a reason. Although a lack of preparation can lend to being open to sometimes stumbling into those captivating intense effusions of fancy, it often can lend to seeming "all over the place" or uncoordinated. The ultimate goal seems to be finding a blend between having enough preparation and guidance, but not forcing any of it; instead, using it when appropriate, but being open enough to the moment to be able to stumble into tangents of unforeseen, un-recreatable spontaneity.

Regarding specific techniques used, all of those mentioned in Chapter 5 had an interesting uniqueness, worthy of perusing further. This adds to the argument for an all-inclusive system with multiple techniques/approaches available, then being able to call upon whatever methodologies when needed; it is all a matter of having the right task for the right job.

On Modern Improvisation

One of the bigger specific takeaways that digital systems have had on improvisation is the level of abstraction available for offering more indirect ways to interface with complex and automated systems. This ability to step back and control macro parameters allows one to be more conscious over his whole system and undergo more of a variety of tasks, increasing the capability for score-level improvisation to help round out improvisational possibilities throughout all three levels of performance. What is interesting about this is that modern improvisation actually seems to be enhanced by making control more simplistic, or less intensive on physical/mental performance (albeit more so on the programming end). As new technology develops, new outlets for improvisational control will be explored/concocted as has been the case throughout history, but unless a newer level of performance gets discovered, taxonomied, or invented beyond note, score, and effect, with computer-aided performance we seem to have now reached a zenith in the type of improvisation possible. Despite this, as the record of time suggests, the affinity for effusions of fancy will likely remain a constant throughout evolving musical performance, techniques, and technologies

Future Work

The art of spontaneity will probably always be a lingering tone throughout my own future work, be it technical, musical, visual or otherwise. Ideas for improvisational improvement span the gamut of micro to macro; from minor mapping tools, to building new visual interfaces, refining different performance techniques, and ultimately designing a complete "all-inclusive" improvisational musical system.

For speed and efficiency, LiveMLR will eventually be built in MaxforLive, or as a remote script, so no intercommunication with any other software like Processing is required. I see the future of LiveMLR less of an exact replication of MLR, but instead as a selection of the main approaches like buffer shuffling and event-driven pattern recording, for incorporation into this larger all-inclusive Ableton-driven system.

LightBalloons is in a continual state of refinement, from making its setup and operation less high maintenance, to programming more patterns, to dramatically increasing the size and scope of the project (e.g. more lights). Currently LightBalloons is being incorporated and expanded upon as an interactive Light Temple–an enterable 12'x12'x12' pyramid of LightBalloons, with hypnotic light patterns controlled by user interaction.

As music technologists, performers, improvisers, developers, and artists alike, we push the boundaries of audiovisual technology, striving to reach some ultimate end goal. For better or worse, there will likely never be any culmination of an ideal state or system of digital electronic improvisational performance; whether regarding personal exploration and development, or contributions to Music Technology as a whole, in seeking new advancements, methodologies, theories and techniques, a continual state of growth has always been and will always be inevitable. All we can do is continue through fleeting effusions of fancy to strive for that something new.

APPENDIX A

LightBalloons: A Technical Glance

This section highlights the chronology of the LightBalloons project, filling in gaps that are either too technical or apply too little to improvisation to be included in the body of this work.

A.1 Concepts

Along with the inspirations and case studies mentioned in Chapter 4, there have been some more conceptual reasons for experimenting with the project. Foremost has been a strive for procuring and controlling relatively inexpensive lighting. Primarily, they have been a feasible way to work with controlling visuals because LEDs prove very cheap compared to other light-based visual mediums (like professional DMX lighting and projections).[109]

More conceptually, visual artistic mediums have progressively grown more three-dimensional as time and technology has progressed. From hard lighting of empty spaces, to 2D projections, to fairly recent 3D projection mapping–which is really only a distortion of the 2 dimensional plane to give an illusion of 3D. What if light-based art had a three dimensional form to it? What if structure itself emanated colored light sequences? How might structural light-based art be treated differently, both artistically and perceptually? The LightBalloons project is interested in utilizing the tangibility and physicality of light as structure, or physical object, to most viscerally, intellectually, and beautifully explore the astounding relationships between light and sound.

Albeit embodying light as tangible 3D structure, LightBalloons as an instrument is initially formless. Being merely a string of balloons chained together allows the instrument to take on whatever shape desired or necessary for the venue/event, rendering LightBalloons

[109] Of course, in being cheap project utilizing somewhat raw materials, there are definitely drawbacks, like the lights are not as bright as higher powered DMX lighting, and everything from designing, to building/assembling, to programming, has to be done basically from scratch.

exceptionally improvisational in physical configuration. The instrument itself is quite modular in that it can work with just a few LightBalloons, to hundreds, depending on the occasion.

A.2 Mapping Specifics

For live performance and improvisational ease, LightBalloons is intended to be programed to offer the most functionality out of the most simplified control, without sacrificing a manual feel. One way this is accomplished is by using the same group of QuNeo buttons to control different pairings of lights (called Brightness Group Control in Figure 46), depending on which bank they were set to the bank is switched by another group of QuNeo buttons programmed as Bank Switchers (called Brightness Group Select in Figure 46). For example, Group Select B divides the lights into halves, and C divides them into thirds. “Control 1” would control the first half in group B and the first 3rd in group C, and so forth. This type of methodology implements two simultaneous complex mappings for efficiency and ease in performance: a one-to-many mapping of a button to control different groups of multiple lights at once, and a many-to-one mapping between the bank switching buttons and the ones that control the lights.

Another mapping tactic of note is the saturation control in Figure 46, consisting of two separate controls controlling the same parameter. Although this might be perceived as a waste of controller real estate, it happens to be a useful many-to-one control, allowing the slider to gradually shift between completely saturated and unsaturated, and anywhere in between, while the other button provides for a quick switch between the two extremes.

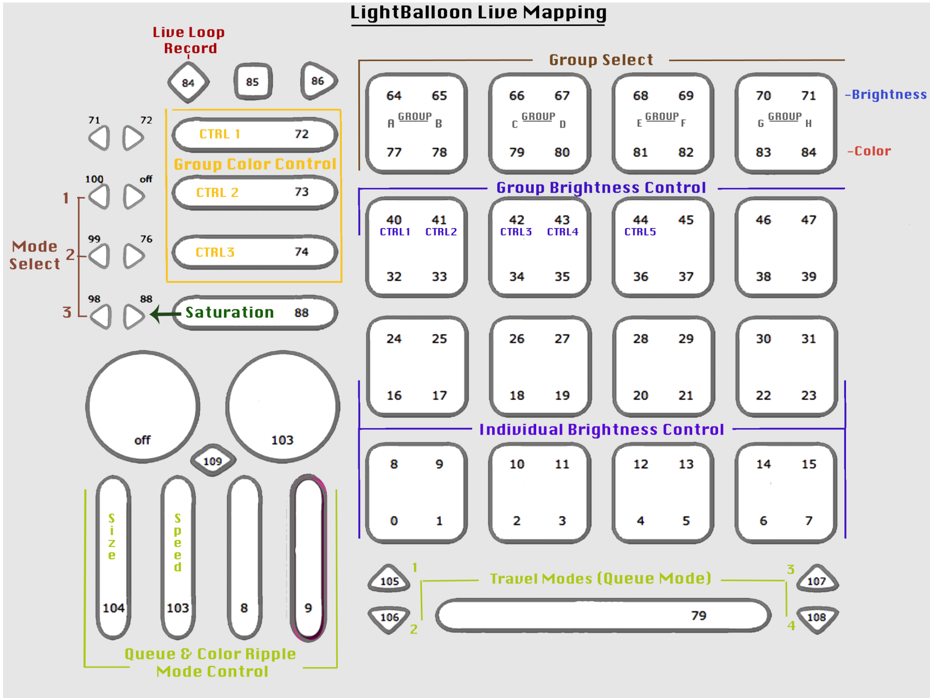


Figure 46.LightBalloons Live Mapping

A.3 Hardware Revisions

Much has been refined about the physical development of the instrument over the first handful of LightBalloons showings, most of which were out of necessity to make LightBalloons a continually usable instrument.

A.3.1 Gas Leakage

Anyone that has casually observed a helium-filled balloon for at least a day knows that it will slowly shrink and lose its ability to float. This is because helium molecules are just small enough to slowly leak out of the pores of a rubber balloon. If this was not undependable enough, the embedding of an LED with wires into a balloon implies that there will always be an opening where the wires enter/exit the balloon. The first attempted solution to alleviating this major hurdle–employed for the first three showings–was to tie off the lip of the balloon as tightly as possible multiple times with thin string against the lip with the wires hanging out. This solution required about 2 hours of "fill-up" time per 20 balloons, blisters and bloody fingers from incessant string tying, and an undependable exceedingly variable balloon life of 1 to 5 hours between balloons, meaning, after an hour some balloons would be mostly deflated and others not; by 5 hours most were mostly deflated. The next solution was to try and smooth out the wiring protruding from the balloons by spraying them with Plastidip (a liquid spray-on electrical insulation material), so as to prevent any rippling of the rubber against the wiring when again, tied off with a string. Causing more consistency in balloon lifespan, Plastidip was a major improvement, however the long setup time and sore fingers still persisted with the "tie off" method. It was soon discovered that a clear plastic tube slightly larger than the diameter of the lip of a balloon, inserted into the lip of a balloon, as Figure 32 demonstrates, will prevent any gas from leaking out of the balloon's entrance/exit whatsoever. To date this has proven to be the most effective method of keeping a balloon airtight, with a consistent balloon life span of at least (if not far more than) 7 hours, and a setup time reduced more than half.

A.3.2 Plug and Play

The simplicity in wiring of the WS2811 LEDs (Neopixels), which only require one data in and data out pin, is advantageous in that wiring LightBalloons is quicker, cheaper, and lighter (which helps them float) to assemble and leaves little room for error; it is disadvantageous in that if one LED or connection between LEDs goes bad,[110] every consecutive LED that follows after it in its daisy-chain will no longer work, then rendering that strand of the instrument defunct during performance. To work around this issue detachable locking plugs, called JSTs (Japan Solderless Terminal) have been inserted between every balloon for a quick and painless changeover if ever necessary, as well as for convenient portable assembly before each show. (as shown in Figure 32).

A.3.3 Brightness

Using only 5 volts for power, one can easily and rightfully assume that Neopixels are not all that bright, outputting roughly 0.3 watts maximum per LED (or about 9.5 watts per strip of 30). Future work will likely involve testing brighter LEDs in the same size class, but until then, a quick fix since utilizing clear plastic tubes has been to triple the brightness by tripling the amount of LEDs per balloon. Although the instrument is still fairly ineffective during the daytime, this improvement has made the LightBalloons much more professional looking and captivating at night, or in the dark.

As this project has grown in scope, tripling the amount of LEDs per balloon and daisy-chaining more balloons, it was quickly discovered that just after about 30 LEDs in the chain, the

[110] Usually caused by voltage spikes and reflections along the data line, which, with basic wiring blowing up LEDs is only an inevitable matter of time.

their color starts to turn an ugly brown. From help forums it was learned that new sources of clean voltage lines need to be tapped into the chain. This has resulted in utilizing multiple strands of no more than 10 balloons (30 LEDs) that can modularly connect to each other with fresh power and ground for each strand, or be set to output from different Arduino pins.

A.4 Software Specifics

Because the Arduino code for the LightBalloons project is so specific and long winded, it has not been included in this document, but can be made available upon request.[111] The code has been derived using Adafruit's Neopixel Library, which abstracts the low level communication. The serial receive protocol in Arduino was derived from Owen Vallis' Chronome Arduino serial protocol, modified to suit the needs of LightBalloons. In the earlier stages of the project, every so often one of the LEDs would stop working; although the reason is not for certain, it appeared to happen when messages received through serial became too fast for the updating of the LEDs. In order to protect the lights from not being updated too fast, a time regulator utilizing Arduino's millis() function was placed on the final function (stripShow_All()) that outputs data out to the LEDs. At output intervals of no less than five milliseconds, no LEDs have been harmed ever since. Here is the regulator:

```
//Arduino time regulator
unsigned long StripShowMillis = millis();

 if(StripShowMillis - previousShowMillis > stripShowinterval) {
   previousShowMillis = StripShowMillis;
//place time regulated function here:
   stripShow_All();
 }
```

Many serial transmission methods and protocols have been tested throughout the duration of LightBalloons. Although far from perfect, it has been eventually settled upon Max4Live as the most direct and efficient way to send MIDI messages from an external controller, as well as utilize the internal sequencing abilities of Ableton Live. Moreover, the serial protocol written for LightBalloons will basically render any Arduino-based controller as an automatic class compliant MIDI controller (that could be used for controlling Ableton or any other MIDI program). Because this patch can prove quite useful beyond the scope of

[111] Contact: davidhowe@alum.calarts.edu

LightBalloons, it is short enough to be shown here. The majority of the patch was derived from a Max tutorial on serial protocol. Little more has been added aside of a Most Significant Bit (MSB) scheme in the upper middle of the patch. Utilizing a MSB data scheme, any data send over the number 127 would have a MSB of 1, and therefore is received as an ID number, and any data under 128 is a value for that ID. This limits the amount of different parameters to be modified to 128, and the range of continuous value to 128 as well. Initially, the ID limit seemed like an ample amount of room for control, but has gotten used up rather quickly. A picture of the patch in plugin form in Ableton Live (Figure 47) and as the full Max4Live patch (Figure 48), as well as the actual code is provided for public use in the following figure.

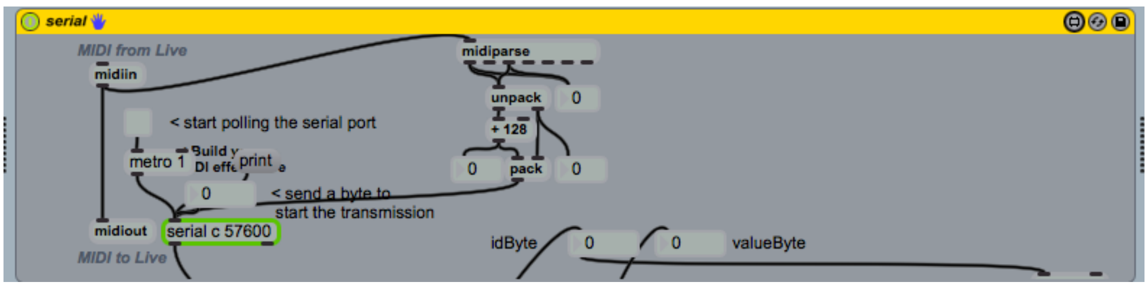


Figure 47.Ableton Live Plugin Window of Max4Live Serial Protocol for LightBalloons

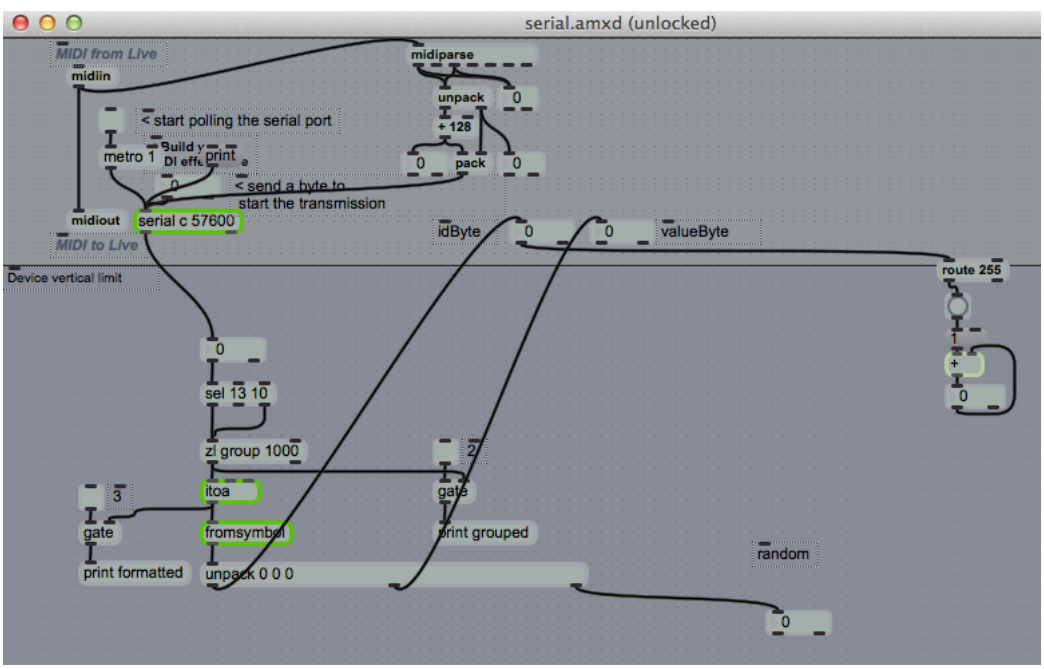


Figure 48. Full Max4Live Serial Protocol Patch (Expanded View)

Max4Live Compressed Code (copy and paste in Max4Live Patch):


```
<pre><code>
----------begin_max5_patcher----------
1739.3oc4at0aaaCE.94jeED505EHdSWF1KqnuTfseAqCExxLNpSVxPhNKoE
8+93E4aYVTzJzzJsHHVQJJRG9wygmK7juc6MAyqeh0F.9UveAt4luc6M2ntj
7B2zc9MAqxdJuLqUcaAy2v40UAyz+ppMqp2vKYb0uD1c00Y77GJpV94FVNW+
viCo2ENC.SCkGP69D72c+MEKTO954e4WPnsOd8yl+7Zl9oDLOqZYvt+Fwaun
Z2KWdsue6sxOlY4fQ7.lyZ191tuthWksR8tB98lhrxSOLQCNLQzD4ARjdL2y
vjDr6A0HdsbVymYUYyKUBP3ABUawWUWCBuilRgIw8gmhJdvrWRoG3rm340k0
M56J7tDgXljL6j+D7.g8X91cQdSQadVmPdW5XnN6eEi+sCdozot76N24ggU2
PXh7.Nz37.MXrr9zjZq9wQLOFqnLR9YDtiyi.dqXssYKY+O5Au.zCpzewn6n
B4MtG5EOJ50G5bm5Ti3cxDVhz9AC380kKNaa7nzTk5TjR2hpWWKoG5DdR5H0
G6CLy7AbVmk+OihKFzZvBiMghRh1jCaBKznyFKWbj7N.Dk3blffRlPiULIxH
SnmKSLrBzTvWHNJUN1wztEQLrDLE6GWgSu.F5fTmQSrQHgdq.oSafsoZzq5f
FzBSqkQLagAGiE1LSFZNDOqJVTHlXaYihPwFHTnlPJzDFZjPiycU22GPqYWJ
14Z6OTZrs1ejz2H1e40qVwjO4WnggOWtEZHDwDkNUma9Tiba25V22rbtNsAa
n0EiCHGxALAKAAJM0BP.uRffWubYI6rCdgFcvbr4T1ww1XabFQv3vEVW2HdM
f6qaVkw4rENbxuCP5ZZjPMM2iSB7wLcOHXYFm4v7C6F2QgGDoeei6zqc5giT
4uKthNC6Az9IScs+khrgW6Tc+swcoUBhMtvGl9Cjx+1ANJ0Bs+nqdwQbc.SP
hJfIDRMvolqrF9MdDS+Fnkm0vAqqKKEX.vefAZYpPwWW2vcn4TpJMPpNKYpQ
kJXxDKdJAkXUK.Yf4OyY.d8mp5vlDW7lrp1UEssE620.AKY40ap3Gpq4.HB0
ISCEI7nVsNdakGNIFISLL9XV4F16EH7SUNTwhjnWZVmfr40ogwSLjHzcVTux
oQrA0I5EM7pWvnIFMJVH0NbuO7NcChYcC5UhFN2CFMVoCPPwVnC7SagII3CW
2X.Jg9os7swmAkf+HT+VPn7qyEg3ACnDmpcZqOzKCCCdEbxmky89l5UsOuZd
coCy8XKq5R5xn5F0kod.O0VZioQIQps0NV7Cn3Y.s.Mxc1tGPVvqyt.HTm9l
4NzHwwMF.95QwuVpqAfbG.Bu.oCpKSRhQiWzHyFd1TndPw5MxENb0fH3oa0f
DRSSMvo8JhhKZ2fl8+QHiZ52fyMGt6+srRgWb.7B3YCQ0IajXDNzwCmY6O12
BOShJGAI5sfDZgtRzajXk5qcrTkc08tshOrni8zQVjoZGY0U4rbQFXQN0Ijd
Obg5D5RLV7LRxqI.xqXzQ8W7s2uonbA345MMf+7ie3i.182KHC3AVC6LK3lg
9KvTU2BoJUSz9cAC0G8srFB6u7EqXJef8XQNC7HqgWjKF5kEqJbYAc0S1cgn
CSL00E61hrGJVrfUcnF00mSJUJdM3OJdbnNSA7QdVYQ94WFJcP3DcEJMZ9Fe
n7cr4lz9BSjeRNtKlGrdU9ggxDBu7TTaDZLTinIMCMzcTBND330tHZRo8bPh
M1JuAWXKPCi7hJW2Xpjj88qBwXOgAcWaodHUT2gxwzK92OQIrxqeLpZE92x2
9J5RNGrWfWvZ4EUYb4tLs+dnGcO6WdcqrunnUFSoZvFdxoLWJMxdg..MHNho
500B70QDXWaCqS5LsqaZO5DcDO59vfdX7ftajI6U6AGZjDuQZJ0lI9nok7H6
cPeIOvizxLvGn+jmAmunSL9f7m7DZCev9UdPSL4YP9.mX7wixiE3AF5MwgXi
mPj+TejUCYJ49hLwbmRrQ+AE4W4A8ZkmWFIUZzgatHMUumEGelNKTcHUpyt.
ASIqA9jZ1Gai7P7l7fSsINZ+wGbrsw06I4wlUOv9KXJrM5yX+s5Axl4Kj+3C
xl4qjok3fiFWZp5NBFSn6N6Br.JxJ2U9aAKDwVEtyfnwP0+I0Hch8wctjN9L
c+KDin5twN8xPaqbO3QyIjMzN1exiM4lf7WtjPax0F4ujAf1jqDD6W4Ynb+g
H+JOCl7l+xkzpR0DNoDmwNYoq+b150OxZZ6djJAIXU1WzaIRzL0oEU5SUUSO
ng8Xw16WeCYM4OTvY47MM5xx+TRTvsx2y2u8+.fCJ+tJ
-----------end_max5_patcher-----------
</code></pre>
```

Appendix B

LiveMLR Technical Files

Similar the LightBalloons code, the Processing code f or LiveMLR is exceptionally long, and in need of revision/finalization, so it is not listed in full here, however again, it can be furnished upon request.[112] The slew of OSC messages for LiveOSC mentioned in Chapter 4 that is required to achieve MLR-like quantized jumping functionality in Ableton Live is what follows. Whether using LiveOSC, Max4Live, ClyphX or some other remote script built to access Live's "backdoor commands," using equivalent messages that follow this procedure outlined in the comments of each block of the following OSC messages will make clips jump!

```
// LiveOSC_Processing.pde
// 2012
//by: David Howe


//OSC message function for MLRing in Loop Mode
void clipJumpLoop(float butPos, float endPos, int track) {
  //turn off clip looping so that loopStart actually adjust the startPos
  OscMessage loopStateMessage = new OscMessage("/live/clip/loopstate");
  loopStateMessage.add(track);
  loopStateMessage.add(sceneSelect);
  loopStateMessage.add(0);
  oscP5.send(loopStateMessage, myRemoteLocation);

  //set loop point which is actually startPos to the jump position
  OscMessage loopStartMessage = new OscMessage("/live/clip/loopstart");
  loopStartMessage.add(track);
  loopStartMessage.add(sceneSelect);
  loopStartMessage.add(butPos);
  oscP5.send(loopStartMessage, myRemoteLocation);

  //play the clip at the current pos
  OscMessage playClipMessage = new OscMessage("/live/play/clipslot");
  playClipMessage.add(track);
  playClipMessage.add(sceneSelect);
```

[112] davidhowe@alum.calarts.edu

```
  oscP5.send(playClipMessage, myRemoteLocation);

  //set loopstate back on so we loop correctly
  loopStateMessage = new OscMessage("/live/clip/loopstate");
  loopStateMessage.add(track);
  loopStateMessage.add(sceneSelect);
  loopStateMessage.add(1);
  oscP5.send(loopStateMessage, myRemoteLocation);

  //set loop start back to 0 (aka beat 0/1)
  loopStartMessage = new OscMessage("/live/clip/loopstart");
  loopStartMessage.add(track);
  loopStartMessage.add(sceneSelect);
  loopStartMessage.add(0);
  oscP5.send(loopStartMessage, myRemoteLocation);

  //set loop point which is actually startPos to the jump position
  OscMessage loopEndMessage = new OscMessage("/live/clip/loopend");
  loopEndMessage.add(track);
  loopEndMessage.add(sceneSelect);
  loopEndMessage.add(endPos);
  oscP5.send(loopEndMessage, myRemoteLocation);
}
```

Appendix C

Some Poetry

Improvisation has been a part of my life since before I knew or could say the five-syllable word. My indoctrination started before I can remember, growing up listening to my older brother's electric guitar-driven effusions of fancy through the walls. With my family always listening to Jazz around the house or via my father's musical jazz and classic rock escapades on trumpet and keyboards, it became obvious my brother and I both caught the bug from him. Eventually I realized my father's gift of improvisation was also passed generationally from his dad, Papa Jack, who would arrange and perform Dixieland Jazz sessions while my dad was growing up. Improvisation was never forced upon me, but the awareness of the invigoration it can bring was given to me at an early age, causing me to seek it out in all of my own musical endeavors, even in completely inappropriate musical situations! Of course this document is a spin on how to take the essence of improvisation–the making of something new and fresh–and apply it to the concept of improvisation itself, e.g. how to usher improvisation into digital modernity by making it new and fresh. Regardless, the affinity that can be had for any improvisation remains consistent from generation to generation to generation. Witnessing this first-hand thanks to Papa Jack and my dad, coupled with my investigative research on the matter, I have been led to conclude that improvisatory performance has and always will be an integral component in the arts, something worth nurturing, fostering, developing, improving, and mostly, enjoying.

In my father's most musical days, for all intents and purposes, the words, *improvisation* and *jazz*, are/were synonymous. The following are a few poems from a collection written by my father about Jazz, and how it has touched him, his family, friends and surroundings. As applicable as these poems are to Jazz, I feel they overarch and begin to unveil the essence of improvisation itself. In a more creative literary approach, these poems,(if not sum up, at least) rationalize and justify the purpose of this document's endless pages of improvisational analysis that have lead up to these following ones.

C.1 Playing With JAZZ

PLAYING with JAZZ

Shout it out,
Turn about,
Play most of the right notes!
Turn about,
Play it loud,
Say most of the right quotes!
Play it loud,
Draw a crowd!
Do whatever floats
To the top.

Top it off,
Drop off and listen.
Make it glisten with hope
And position yourself,
To pose another musical threat.
Another solution!

MIX YOUR MIND AND YOUR SOUL
MAKE YOUR LIFE WHOLE
PLAY THE RIGHT NOTES
SAY THE RIGHT QUOTES

C.2 Hands

Hands

They manipulate,
    They stipulate
        They congregate
Around the right notes to play
    The right things to say
        Every new day.

They allow us to plot
    a small spot
        which means a lot,
To us, to me.
    Essentially
All there can be.

C.3 Jazz 2

Jazz 2

Defining Jazz is easy.
    It's simply the interaction of people in a creative way.
Mostly it's musicians at the core.
    Defining, and realigning
        And signing up.
They perform
    And tell you what they feel and think.
        Perhaps they are the link.
            To your feelings, and thoughts.
But YOU perform, too,
    Everyday, in every way.
        I'll bet **you** usually listen to people and their music.

The laughter and ideas of your children and your friends.
    Think of these sounds as music.
You're interaction at work, think of what you do as music.

(Don't worry, music isn't just sounds. Sound is only one of our senses. … Now Sight, we rely on it. But what do we see? We see the world around us and everything in it in our view. Visual Music! Touch? Sure, you touch stuff, but what touches you? Does She (or He)? What touches your mind, or your heart? Music? Smell? Sure how bout the smell of dinner? Music to your nose!)

All the best music is Jazz. I don't mean that "jazz" music is the best. It's simply one of them.

Jazz is simply spontaneous creation, interacting with others around you, or at least telling or showing or inviting others to share your music.

It's the symphony of sight. It's the interesting touch. It's the intriguing smell.

But mostly it's creating!

You have Jazz in you! Just listen and see, touch and smell. But mainly feel! (And think, of course.)

Simply empathize, or at least sympathize,
    And recognize,
        That what you and they do,
            Affects each other.

When you create
    YOU make Jazz.

Bibliography

Bischoff, John, R. Gold, and J. Horton. 1978. "Music For an Interactive Network of Computers." *Computer Music Journal* 2(3): 24–29.

Collins, Nicholas M. 2006. "Towards Autonomous Agents for Live Computer Music: Realtime Machine Listening and Interactive Music Systems". Citeseer. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.65.2661&rep=rep1&type=pdf.

Cook, Perry. 2001. "Principles for Designing Computer Music Controllers." In *Proceedings of the 2001 Conference on New Interfaces for Musical Expression*, 1–4. http://dl.acm.org/citation.cfm?id=1085154.

Esmonde, Peter. 2009. *Trimpin: The Sound Of Invention*. Documentary.

"Feature: Industry Interview -Roger Linn @ SonicState.com." 2013. Accessed November 18. http://www.sonicstate.com/articles/article.cfm?id=71&page=3.

Fourier, Laurent, Curtis Roads, and Jean-Jacques Perrey. 1994. "Jean-Jacques Perrey and the Ondioline." *Computer Music Journal* 18 (4): 19. doi:10.2307/3681354.

Gallin, Emmanuelle, and Marc Sirguy. 2009. "Sensor Technology and the Remaking of Instruments from the Past." In *Proc of the 8th International Conference on New Interfaces for Musical Expression (NIME)*. http://www.nime.org/proceedings/2009/nime2009_199.pdf.

Gao, Mingfei, and Craig Hanson. 2009. "LUMI: Live Performance Paradigms Utilizing Software Integrated Touch Screen and Pressure Sensitive Button Matrix." In *Proceedings of the 2009 Conference on New Interfaces for Musical Expression*. http://www.nime.org/proceedings/2009/nime2009_058.pdf.

Geiringer, Karl, and Bernard Miall. 1945. *Musical Instruments*. Oxford University Press, New York, Inc.

Gluck, Robert J. 2009. *Silver Apples, Electric Circus, Electronic Arts, and Commerce in Late 1960s New York*. Ann Arbor, MI: MPublishing, University of Michigan Library. http://www.electricsongs.com/texts/gluck_silverapples_icmc09.pdf.

Hatten, Robert S. 2009. *"Opening The Museum Window: Improvisation and Its Inscribed Values In Canonic Works By Chopin And Schumann.", Musical Improvisation: Art, Education, and Society*. Edited by Gabriel Solis and Bruno Nettl. University of Illinois Press.

Hochenbaum, Jordan Natan. 2013. "L'arte Di Interazione Musicale: New Musical Possibilities through Multimodal Techniques: A Dissertation Submitted to the Victoria University of Wellington and Massey University in Fulfillment of the Requirements for the Degree of Doctor of Philosophy in Sonic Arts, New Zealand School of Music." http://muir.massey.ac.nz/handle/10179/4525.

Howe, Jackson. Jazz. TS. 2008. Collection of Jackson Howe, Los Angeles

Hunt, Andy, and Ross Kirk. 2000. "Mapping Strategies for Musical Performance." *Trends in Gestural Control of Music* 21. http://www.music.mcgill.ca/~mwanderley/MUMT-615/Papers/Class10/P.HunKir.pdf.

Jordà, S. 2002. "Improvising with Computers A Personal Survey (1989-2001)." *Journal of New Music Research* 31.

Jordà, Sergi. 2005. "Digital Lutherie: Crafting Musical Computers for New Musics Performance and Improvisation." *PhD Diss., Universitat Pompeu Fabra, Departament de Tecnologia.* http://mtg.upf.edu/node/449.

Kapur, Ajay. 2008. *Digitizing North Indean Music: Preservation and Extension Using Multimodal SensorSystems, Machine Learning, and Robotics*. VDM Verlag Saarbücken, Germany.

Kapur, Ajay, and Michael Darling. 2010. "A Pedagogical Paradigm for Musical Robotics." In *Proceedings of the International Conference on New Interfaces for Musical Expression*, 162–65. http://mtiid.calarts.edu/sites/mtiid.calarts.edu/files/publications/2010_nime_robotics.pdf.

Kapur, Ajay, Michael Darling, Dimitri Diakopoulos, Jim W. Murphy, Jordan Hochenbaum, Owen Vallis, and Curtis Bahn. 2011. "The Machine Orchestra: An Ensemble of Human Laptop Performers and Robotic Musical Instruments." *Computer Music Journal* 35 (4): 49–63. doi:10.1162/COMJ_a_00090.

Kapur, Ajay, Michael Darling, Meason Wiley, O. Vallis, J. Hochenbaum, J. Murphy, D. Diakopoulos, C. Burgin, and T. Yamin. 2010. *The Machine Orchestra*. Ann Arbor, MI: MPublishing, University of Michigan Library. http://karmetik.com/sites/default/files/publications/2010_icmc_tmo.pdf.

Kapur, Ajay, Philip Davidson, Perry R. Cook, P. Driessen, and A. Schloss. 2004. "Digitizing North Indian Performance." In *Proceedings of the International Computer Music Conference*. Vol. 2004. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.93.7569&rep=rep1&type=pdf.

Kapur, Ajay, Georg Essl, Philip Davidson, and Perry R. Cook. 2003. "The Electronic Tabla Controller." *Journal of New Music Research* 32 (4): 351–59.

Kapur, Ajay, Ariel J. Lazier, Philip Davidson, R. Scott Wilson, and Perry R. Cook. 2004. "The Electronic Sitar Controller." In *Proceedings of the 2004 Conference on New Interfaces for Musical Expression*, 7–12. http://dl.acm.org/citation.cfm?id=1085886.

Kinderman, William. 2009. *"Imprivisation In Beethoven's Creative Process." Musical Improvisation: Art, Education, and Society*. Edited by Gabriel Solis and Bruno Nettl. University of Illinois Press.

Lawson, James, and Max V. Mathews. 1977. "Computer Program to Control a Digital Real-Time Sound Synthesiser." *Computer Music Journal*, no. 1(4): 16–21.

Levin, Robert. 2009. *"Improvising Mozart", Musical Improvisation: Art, Education, and Society*. Edited by Gabriel Solis and Bruno Nettl. University of Illinois Press.

Manning, Peter. 2013. *Electronic and Computer Music*. Oxford University Press. http://books.google.com/books?hl=en&lr=&id=75EiOeUFZMEC&oi=fnd&pg=PP2&dq=Manning,+Peter.+Electronic+and+Computer+Music&ots=z-jXg8E_5c&sig=TmMQVKNgChhoPtNkJjBGmc9bJqI.

Mathews, Max V. 1991. "The Radio Baton and the Conductor Program, or: Pitch, the Most Important and Least Expressive Part of Music." *Computer Music Journal*, no. 15(4): 37–46.

McCartney, James. 1996. "SuperCollider: A New Real Time Synthesis Language." *International Computer Music Association*.

———. 2002. "Rethinking the Computer Music Language: SuperCollider." *Computer Music Journal* 26 (4): 61–68.

Mee, C. Denis, Eric D. Daniel, and Mark H. Clark. 1999. *Magnetic Recording: The First 100 Years*. IEEE. http://www.lavoisier.fr/livre/notice.asp?ouvrage=1570135.

Merino, JC. 2003. "Lissajous Figures and Chebyshev Polynomials." *College Mathmatics Journal* 34(2) (March).

Nishibori, Yu, and Toshio Iwai. 2006. "TENORI_ON." In *Proceedings of the International Conference on New Interfaces for Musical Expression*, 172–75.

"Roger Linn On Swing, Groove & The Magic Of The MPC's Timing - Attack Magazine." 2013a. Accessed November 18. http://www.attackmagazine.com/features/roger-linn-swing-groove-magic-mpc-timing/.

———. 2013b. Accessed November 18. http://www.attackmagazine.com/features/roger-linn-swing-groove-magic-mpc-timing/.

Nettl, Bruno, and Melinda Russell, eds. 1998. *In the Course of Performance: Studies in the World of Musical Improvisation*. University of Chicago Press.

Trueman, Daniel, Perry Cook, Scott Smallwood, and Ge Wang. 2006. "PLOrk: The Princeton Laptop Orchestra, Year 1." In *Proceedings of the International Computer Music Conference*, 443–50. http://www.scott-smallwood.com/pdf/plork_icmc2006.pdf.

Vallis, Owen. 2013. *Contemporary Approaches to Live Computer Music: The Evolution of the Performer Composer* Victoria University of Wellington.

Vallis, Owen, Jordan Hochenbaum, and Ajay Kapur. 2010. "A Shift Towards Iterative and Open-Source Design for Musical Interfaces." In *Proceedings of the International Conference on New Interfaces for Musical Expression*, 1–6. http://www.karmetik.com/sites/default/files/publications/2010_nime_interfaces.pdf.

Wanderley, Marcelo, and M Battier. 2000. *Trends in Gestural Control of Music*. CD-ROM. Paris: IRCAM-Centre Pompidou.

Wanderley, Marcelo Mortensen, and Nicola Orio. 2002. "Evaluation of Input Devices for Musical Expression: Borrowing Tools from Hci." *Computer Music Journal* 26 (3): 62–76.

Wang, Ge, and Perry R. Adviser-Cook. 2008. *The Chuck Audio Programming Language. a Strongly-Timed and on-the-Fly Environ/mentality*. Princeton University. http://dl.acm.org/citation.cfm?id=1559215.