Push, Twist & Touch:
Mapping Systems for Live Performance

Cameron Sexton
Student No. 2422 6440

Submitted in partial fulfilment of the requirements for the degree
Bachelor of Music (Honours)
Unit: ATS4834

October 2017

Sir Zelman Cowen School of Music
Monash University

Statement of Responsibility

I declare that this research project contains no material that has previously been submitted for a degree or diploma at any university and to the best of my knowledge, this research project contains no material that has previously been published or written by another person, except where due reference has been made in the text of this research project.

Signed:

Date: 29/10/2017

Abstract:

This thesis documents the background and concepts underpinning the digital musical instrument, *Electric Sheep Texture Synthesizer*. Many computer-based synthesizers are compromised by a lack of adequate real-time control solutions that limits their versatility in live performance. Research into controller mapping techniques that address this issue has led to the design of this performance-focused instrument and a recorded folio of creative works demonstrating its use. Frameworks to build sophisticated mapping architectures are discussed in order to construct a model for performance-focused mapping design, and a formula for expressivity is proposed. Three mapping systems of varying complexity were developed for *Electric Sheep* and here they are compared and evaluated for expressivity in live performance.

Acknowledgements:

I would like to thank my supervisor Dr. Catherine Hope at Monash University for so generously giving her time and feedback to this project. It was a privilege to have the opportunity to work under her expert guidance, and this thesis would have been much more difficult to write (and read) without the clarity of ideas and expression that she has inspired. I could never have managed to achieve as much as I have this year without my teachers Casey Rice and Jem Savage, who have revealed programming concepts and techniques unimaginable to me a year ago. Thank you as well to Sara Sithi-Amnuai for her attention to detail in proofreading. I am grateful to the open-source community, especially contributors on the Cycling 74 and Monome online forums through which I have trawled for many hours to pick up nuggets of wisdom and advice.

I would especially like to thank my parents, Kaye and Peter, for their proofreading and who have continued to encourage me to experiment and create my own musical instruments, regardless of how “pleasant” they sound.

Table of Contents

List of Figures: v

Chapter 1 1

- Background: 1
- Aims: 1
- Delimitations: 2
- Methodology: 2
- Literature Review: 3
- Definitions: 3
- Chapter Outline: 5

Chapter 2 6

- The Digital Instrument Model 6
- Transparency, Expressivity and Efficiency 7
- Grids and Encoders 8
- Control Interfaces in Electric Sheep 10

Chapter 3 13

- Determinants, Sound Space & Constraints 13
- Mapping Techniques 14

Chapter 4 18

- Managing Complex Mappings 18
- Visual Feedback from the Synthesizer 20

Chapter 5 22

- Electric Sheep Texture Synthesizer 22
- Rotary Control – MIDI Fighter Twister 23
- Memory Matrix - Launchpad 24
- Multi-touch - iPad 25

Chapter 6 27

- Evaluation through Live Performance 27
- Discussion of Results and Conclusions: 31

References 33

List of Audio Examples 35

List of Figures:

Figure 1. Digital Instrument Model. 6
Figure 2. Monome grid. 8
Figure 3. Monome arc. 9
Figure 4. Launchpad 10
Figure 5. MIDI Fighter Twister. 11
Figure 6. iPad running Cycling 74's MIRA app. 11
Figure 7. One-to-one mapping. 14
Figure 8. One-to-many mapping. 15
Figure 9. Many-to-one mapping. 15
Figure 10. Many-to-many mapping. 16
Figure 11. Extended DMI model. 19
Figure 12. Monome grid displaying LED feedback. 21
Figure 13. The controllers arranged together on stage. 22
Figure 14. Bank of rotary encoders. 23
Figure 15. Colour coding of Memory Matrix functions. 24
Figure 16. Touch surface interaction. 26

Chapter 1

Background:

“Many devices are referred to as computer instruments but are not designed for live performance, rather for non-real-time editing operations.” (Hunt and Kirk, 2000)

The association of computers with “office tasks” and the origins of computer-based synthesis in offline sound rendering have made their adaptation to live performance challenging (Hunt and Kirk, 2000). Sound design techniques imported from the studio paradigm are largely incompatible with the real-time demands of the stage. Many hardware devices now exist to give performers hands on control over sound design, but the challenge of live performance lies with the enormous quantity of synthesis parameters afforded by modern computer music software - too many for one performer to realistically negotiate in the moment. Electronic musicians adopt various strategies to manage this difficulty, with some relying more on prepared material to be played back in performance and others embracing the intricate task of live sound design. Computer programmers have made their own attempts at tackling this issue, and musical programming environments provide a platform to develop solutions that translate a performer's actions or "control data" into sound. The pathway between control data and synthesis parameters is a major component in digital musical instruments, and the important process of constructing it is referred to as mapping.

After discovering the music programming tool *Max,* I decided to design my own instrument, *Electric Sheep Texture Synthesizer*, with the aim of developing a mapping system oriented towards performing live. *Electric Sheep* is a sample-based synthesizer for generating and manipulating textures, which is intended to be used mostly in duo and ensemble situations to accompany and improvise with acoustic instruments. I designed mapping systems to integrate three hardware controllers - a bank of endless rotary encoders, a "grid", and a multi-touch surface - with the software synthesis algorithms in *Electric Sheep*. These mapping systems are the focus of this thesis.

Aims:

This thesis explores the proposition, set out in Chapter 2, that a digital instrument's expressivity in live performance is the product of both transparency and efficiency in its mapping system. I will apply this hypothesis to my instrument, *Electric Sheep*, in order to compare and evaluate the expressivity of its three mapping systems, and determine which of

these is most useful in live performance. The accompanying recorded folio of creative works serves to supplement and support this evaluation.

Delimitations:

Hardware, software and firmware all play major roles in digital instrument design but the scope of this thesis is limited to mapping, which is a software process. Inevitably discussion of hardware and firmware will occur as these elements are directly linked to one another, and so Chapter 2 provides a comprehensive breakdown of the Digital Instrument Model in order to remove any ambiguity surrounding each element's role. Similarly, this research does not attempt to make suggestions for new technologies but rather focuses on how instrument designers might make better use of existing hardware. Accordingly, I have chosen generic hardware for my mapping systems, covering three common interface types that can be easily substituted by similar products with little change to the *Max* patches (software code).

Methodology:

Brown and Sorenson's "develop, deploy and reflect" (2009, p. 159) approach to practise-based research involving new media in the arts is a practical methodology in a field that is highly susceptible to subjective opinion. I am aware that preference for musical instrument design is extremely personal, so in order to obtain useful research outcomes I have conducted an iterative software development process following their model that has produced an analysis of the mapping systems used in *Electric Sheep*. My development process revolves around five "determinants" of sound defined by John Cage in 1957, which I use to evaluate the relative transparency and efficiency (expressivity) of a mapping system. These determinants are accepted elements of music that all common synthesis software provides controls for: "frequency or pitch, amplitude or loudness, overtone structure or timbre, duration, and morphology (how the sound begins, goes on, and dies away)" (Cage, 1957, p. 9).

The primary data collected in this study consists of my experiences undertaking the development process and these are interpreted through analytical reflections. The process is a continuous cycle of conceptualising and implementing the mappings, learning to play the instrument, performing live, identifying and solving problems to refine the mappings, performing again and so on. Reflections draw on experiences or realisations that have occurred at all points in the cycle, and I use Cage's determinants to define strict criteria to qualify them for inclusion. Integrating my research into mapping techniques with Cage's

determinants has ensured these reflections are based in musical, rather than technical, terms. There is always a risk that we can lose sight of musical goals in the pursuit of technological advances, and I intend to keep this paper strictly focused on insights that are relevant to musicians seeking to develop their own digital instruments. In the absence of a quantitative method to measure or rate a mapping system, Cage's determinants guide my reflections away from expressing personal preference and toward a qualitative analysis that is more concrete and factual.

Literature Review:

Computers had been making sound for over half a century before the popular journal *New Interfaces for Musical Expression* (*NIME*) began in 2001. Yet *NIME*, *Computer Music Journal* and *Organised Sound* are still overflowing with proposals for new digital musical instruments. Along with the significant technological advances in recent years, conceptual developments in control mapping have been necessary to keep pace with these changes. Wessel and Wright's (2002) review of the "problems and prospects" for digital instruments at the beginning of the 21st century indicates some major challenges in digital musical instrument design, many of which remain unresolved today. They identify parameter mapping, control metaphors, expressivity and interactivity as key features of digital instruments requiring improvement. More detailed papers by Hunt and Wanderley (2002); Pluta (2008); Magnusson (2010); Fels, Gadd and Mulder (2002); and Van Nort, Wanderley and Depalle (2014) delve into the specific challenges of designing digital control systems that match the expressivity of acoustic instruments. Integrating ideas from these influential authors has formed the conceptual foundations of this paper.

Definitions:

Below are some essential terms used in this thesis relating to controllers, mapping and computer-based music.

Digital Musical Instrument: These are "instruments whose sound generators are digital and separable (though not necessarily separate) from their control interfaces" (as cited in O'Modhrain, 2011, p. 1). In this thesis, the term is used to refer to an instrument in its entirety - encompassing both the control interface (hardware) and sound generator (software) components.

GUI: A Graphical User Interface (GUI) is displayed on a computer monitor (screen) and uses “windows, menus and icons” to represent programs and commands (LINFO, 2004).

DAC: A Digital-to-Analog Converter (DAC) is used to “reproduce sounds stored digitally” and connects the computer output to a speaker (Strawn and Shockley, 2014).

Max: A “graphical [programming] environment” for music invented by Miller Puckette (Strawn and Shockley, 2014).

DAW: A Digital Audio Workstation is “typically composed of the computer itself, a soundcard installed in the computer or a plug-in audio interface... and digital audio editing software.” (Strawn and Shockley, 2014). A DAW is designed for “non-real-time editing” (Hunt and Kirk, 2000) rather than for live performance.

Controller or **Control Interface**: "In electronic instruments, the device that transmits the player’s actions, via electrical connections, to relevant parts of the instrument’s sound generating and shaping circuitry" (Davies, n.d.a). A well-known controller is the MIDI Keyboard.

MIDI: The Musical Instrument Digital Interface is a “hardware and software standard established in 1983 for the communication of musical data between devices such as synthesizers, drum machines and computers" (Burnand, 2009). The mapping systems in *Electric Sheep* communicate with the control interfaces via MIDI messages over USB cables.

OSC: Open Sound Control is a modern alternative to MIDI - "a protocol for communication among computers, sound synthesizers, and other multimedia devices.... [that] provides everything needed for real-time control of sound and other media processing while remaining flexible and easy to implement" (OpenSoundControl, n.d.).

Sample: Samples are "short extracts from acoustic instruments or other suitable sources [that] are digitized, edited, and stored in a memory bank, ready for resynthesis" (Manning, n.d.). The samples used in *Electric Sheep* include percussive sounds (drums and various metallic objects), tonal material (piano, guitar) and waveforms generated by a modular synthesizer.

Endless Rotary Encoder: "In general terms, an encoder may be defined as a device which translates mechanical motion into electronic signals used for monitoring position or velocity" (Considine and McMillan, 2000, 5.1.3). More simply, these are a type of “knob” found on

control interfaces. *Endless* or “incremental” encoders are especially useful because they report directional rather than positional information (Considine and McMillan, 2000, 5.1.3), meaning that they can be dynamically reassigned (mapped to new parameters) without the need to adjust their physical position (rotationally).

LED: "A light-emitting diode: a semiconductor diode that emits light when a voltage is suitably applied" (*Oxford English Dictionary Online*, n.d.). LEDs are often built into the playing surface of hardware control devices to represent parameter values and provide other visual feedback to the performer.

Chapter Outline:

Chapter 2 provides a thorough outline of the Digital Instrument Model, giving an overview of the role of mapping and discussing the relationship between transparency, efficiency and expressivity. The history of grid and encoder interfaces is also discussed and the physical attributes of hardware controllers used with *Electric Sheep* are described. In Chapter 3, mapping techniques to improve control efficiency are discussed and Chapter 4 explores strategies to improve transparency. These concepts set the foundations for my own mapping system designs, which are then described in Chapter 5. Finally, Chapter 6 presents my reflective analyses under Cage’s five determinants, leading to an evaluation of expressivity in my mapping systems.

Chapter 2

In this chapter, I introduce the Digital Instrument Model and describe its major elements before discussing the background to the three control interfaces used in *Electric Sheep*.

The Digital Instrument Model

Digital instruments place an intermediate "mapping layer" between the control interface and the sound producing device, which is not present in acoustic instruments where they are often the same physical object (Hunt and Wanderley, 2002). The separation of components has significantly expanded the possibilities for instrument design, and nowadays instrument manufacturers produce a range of hardware controller devices and synthesizers that can be easily used together in many different configurations (Davies, n.d.b). Thanks to the widespread adoption of data protocols (MIDI) for communication between these components, musicians select their preferred methods of control and synthesis when putting together a complete instrument. Constanzo (2016) details a "collapsing of roles" culture that has emerged today, where the performing artist has absorbed the roles of instrument designer and composer, and in the following chapters I will use the words designer, composer and performer to refer to what is often the same person. As a result of the standardisation of hardware components, the mapping layer has been elevated to become the most important stage in the instrument development process for the designer/composer/performer. Magnusson writes,

> “The sound and mapping engines serve as the core of the digital musical instrument; they are its “real body.” This is the location where constraints are defined and the instrument’s functionality constructed.” (2010, p. 65)

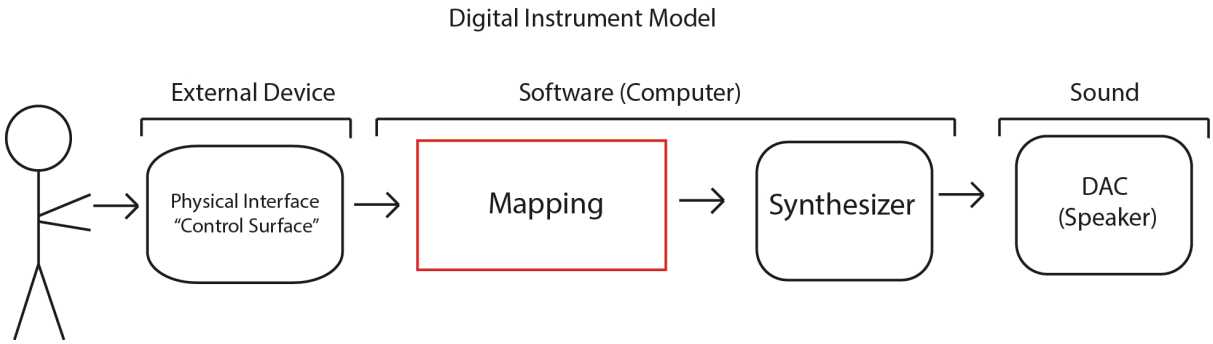

Figure 1. Digital Instrument Model.
Adapted from Wessel and Wright (2002, p. 12).

Mapping is a software process of designing the connection between hardware control interfaces and the "synthesis engine" (Hunt and Wanderley, 2002), so in figure 1 we can see

that mapping lies between these two elements. The control interface is a physical object that is touched, pushed, blown or otherwise interacted with by a performer. Electronic sensors and firmware (software embedded in the hardware) convert these interactions into usable data and transmit it to a computer where the mapping process occurs. The control interface also often includes a Graphical User Interface (GUI) displayed on the computer screen, but this is neither necessary nor always desirable as I will discuss in Chapter 4. Today, it is typical that both the mapping process and sound synthesis are handled in one software application and I use *Max* for these tasks. GUIs are necessary for the mapping process and *Max*'s programming environment relies on a visual "patcher" system to build the program logic (Strawn and Shockley, 2014). Some musicians, known as "live-coders", choose to bring the programming environment itself onto the stage as the control interface (Brown and Sorenson, 2009), but in this thesis the control interface refers only to hardware devices. Once the sound has been synthesised digitally, the audio data is passed through the Digital-to-Audio Converter (DAC in figure 1) to a speaker.

Transparency, Expressivity and Efficiency

In performance, the performer and audience need only be aware of the start and end of this chain - the control interface and the speaker - however the crucial mapping process is always occurring in between. Fels, Gadd and Mulder (2002) recognise "transparency" as a quality of mapping that contributes to the strength of the perceived link between an action and sound. A transparent mapping translates movements or gestures into sonic events in ways that a musician might describe as "expressive" or "intuitive". These subjective descriptions are admittedly problematic, but they help to establish that the goal of mapping is to intimately couple the performer's actions with the sound produced. The mapping layer must essentially be invisible to achieve this sense of interactivity.

However, expressivity can also be undermined by inefficient mapping systems that require the performer to complete too many separate steps to achieve a particular sound they desire to create. Pluta's (2008) affirmation that "maximizing information flow" is vital to live performance provides a stronger clue to expressivity. He stresses that this "does not necessarily mean maximum information flow", but that "physical action is a data stream over which large quantities of information can travel" (2008, p. 1). Control interfaces capture these streams either continuously or intermittently, but it is up to the mapping system to channel this information into the synthesis stage efficiently.

Fels, Gadd and Mulder "consider transparency as a predictor for expressivity", stating that "the more transparent the mapping is, the more expressive the device can be" (2002, p. 2).

I propose instead that expressivity in live performance is a product of both transparency and efficiency, as per the following formula:

$$\text{Expressivity} = \text{Transparency} \times \text{Efficiency}$$

An instrument is therefore only as expressive as its mapping system is transparent and efficient. Musical programming environments, such as *Max*, provide a platform to construct these mapping systems in highly customisable ways that have a major impact on the digital instrument's behaviour - hence the significance of mapping in the Digital Instrument Model.

Grids and Encoders

The 2005 launch of the Monome grid, a programmable button controller invented by Brian Crabtree and Kelli Cain, revolutionised electronic instrument design with a simple interface created principally for use with a computer. The grid layout had no precedent in any acoustic instrument in the way MIDI keyboards had, and Kirn (2014) outlines how major manufacturers of music technology have since released their own controllers derived from the grid's template. The layout consists of translucent keys/buttons arranged in an 8 x 8 or 8 x 16 "grid" pattern. Figure 2 shows LEDs illuminating keys on the grid.

Figure 2. Monome grid.

Reprinted from https://monome.org/ Copyright 2017 by Monome.

Grids are "decoupled by design" - the foundation of the Monome philosophy for user-defined control and key to its popularity (Crabtree and Cain, 2017). On its own, the grid does nothing

and the lighting system is completely separated (“decoupled”) from the buttons. Monome users define their own lighting commands in the mapping stage, customising how the visual feedback will react to button presses. Grids require software to become functional but the open-ended programming possibilities have made them extremely flexible. The grid's firmware sends and receives OSC messages over USB to communicate with software - such as *Max* - performing mapping and synthesis tasks that are determined by the user.

The simple visual feedback provided by the LEDs has allowed electronic musicians to detach themselves from the mouse, keyboard and screen and employ a more “hands on” approach - bridging the gap between digital sound processing techniques and the performability of those functions. Playing the computer more like a traditional instrument fuels the creative process for musicians working with digital sound. More recently, Crabtree and Cain released a new controller – the Monome arc – a set of two or four endless rotary encoders each surrounded by a ring of decoupled LEDs that can be seen in figure 3.

Figure 3. Monome arc.
Reprinted from https://monome.org/ Copyright 2017 by Monome.

Control Interfaces in *Electric Sheep*

While these basic tools – buttons and knobs – are nothing new to computer musicians, Monome's encouragement of an open and flexible platform for music programming has resulted in a large and passionate community of users around the world actively developing new musical and artistic applications for the Monome controllers. The flexible-use philosophy expressed in the grid and arc inspired me to develop my digital instrument using the (far cheaper) Novation Launchpad S and MIDI Fighter Twister controllers. Since these controllers share a similar form to the Monome products, it is straightforward to mimic Monome device behaviour. Indeed, my first programming efforts were to adapt Monome's online "Grid Studies" tutorials to work with the Launchpad.

Figure 4. Launchpad

Reprinted from https://global.novationmusic.com/launch/launchpad

The Launchpad differs slightly from the Monome grid in two respects that can be observed in figure 4. Physically, the Launchpad features larger keys with coloured backlighting compared to the Monome grid's single colour (but variable brightness) backlighting. Additionally, it includes two extra rows/columns of keys surrounding the main array. Instead of the sophisticated OSC messaging implemented in the Monome grid, the Launchpad communicates with the computer via MIDI over USB, which is an older but more widely used data protocol. For my purposes designing mapping software for the Launchpad, the use of MIDI rather than OSC has not presented any real limitations however.

Figure 5. MIDI Fighter Twister.
Reprinted from https://store.djtechtools.com/products/midi-fighter-twister
Copyright 2017 by DJ TechTools.

The MIDI Fighter Twister (MFT) departs significantly from the arc model, but the key characteristic they share is the use of endless rotary encoders with decoupled LED feedback. On the MFT shown in figure 5, there are sixteen low resolution encoders compared to the arc's four high resolution encoders. The resolution of an encoder refers to its sensitivity to turning - the arc is extremely sensitive and responds to very subtle adjustments whereas the MFT encoders have a much coarser, "clicking" feeling as they are rotated.

Figure 6. iPad running Cycling 74's MIRA app.
Reprinted from https://cycling74.com/products/mira Copyright 2017 by Cycling 74.

The third hardware device used with *Electric Sheep* is an Apple iPad, shown in figure 6 running Cycling 74's *MIRA* application - which is the "firmware" passing control data to the mapping software *Max*. This device was chosen because it captures that continuous, physical data stream generated by a performer's hand gestures at a much higher resolution than the grid or encoder interfaces. The precise position of multiple touch points - the player's fingertips - are transmitted over USB to a computer running *Max*. From now on I will refer to this device as the multi-touch controller.

Chapter 3

This chapter introduces Cage's “total sound-space” and four specific mapping techniques before discussing their implications for efficiency in live performance.

Determinants, Sound Space & Constraints

Cage’s five "determinants" (frequency, amplitude, timbre, duration and morphology) together make up what he defines the "total sound-space", which theoretically contains all possible sounds (Cage, 1957). Various mapping techniques give a human performer varying degrees of freedom to move about this sound space via each of the determinants, and specific synthesizer parameters generally relate to one or more of these headings. For example, filter cutoff is a timbral parameter while envelope shapes are affected by both amplitude and duration.

The first challenge of mapping design is to confront the massive array of sound parameters available to control in synthesis software. Categorising parameters by determinants is helpful but it remains difficult to design a mapping that maximises the performer's freedom in all five categories simultaneously. All instruments make some compromise and usually specialise in one or two of these areas. The degree to which a musical instrument focuses control on a subset of these determinants is what defines that instrument (Magnusson, 2010), and this is true of both acoustic and electronic instruments. This observation reveals an important insight for control mapping – the need to strike a balance between flexibility and usability. While traditional acoustic instruments are naturally limited by physics, computers are far more powerful in that they can conceivably generate any possible sound. If we place a human performer in control of the computer it is necessary to construct artificial constraints that filter the many parameters available down to a set that is controllable in real time. As Max Mathews remarked in 1989, "the 'problems' of computer music are no longer that of technology but rather our ability to control it" (as cited in Bowcott, 1996, p. 28). While Puckette (2012) laments that restricting a musician’s access to a minority of sonic possibilities almost defeats the purpose of using computers to make music, Magnusson (2010) argues that digital instruments promote creativity in performance when their mapping design emphasises limitations over affordances. In other words, by clearly defining a digital instrument’s constraints through the mapping system, we can focus the performer’s attention on decisions that substantially affect Cage's five musical determinants.

Mapping Techniques

Hunt and Wanderley (2002) outlined four mapping techniques that summarise the potential relationships between hardware controllers and synthesizer parameters as:

- One-to-one
- One-to-many
- Many-to-one
- Many-to-many

One-to-one:

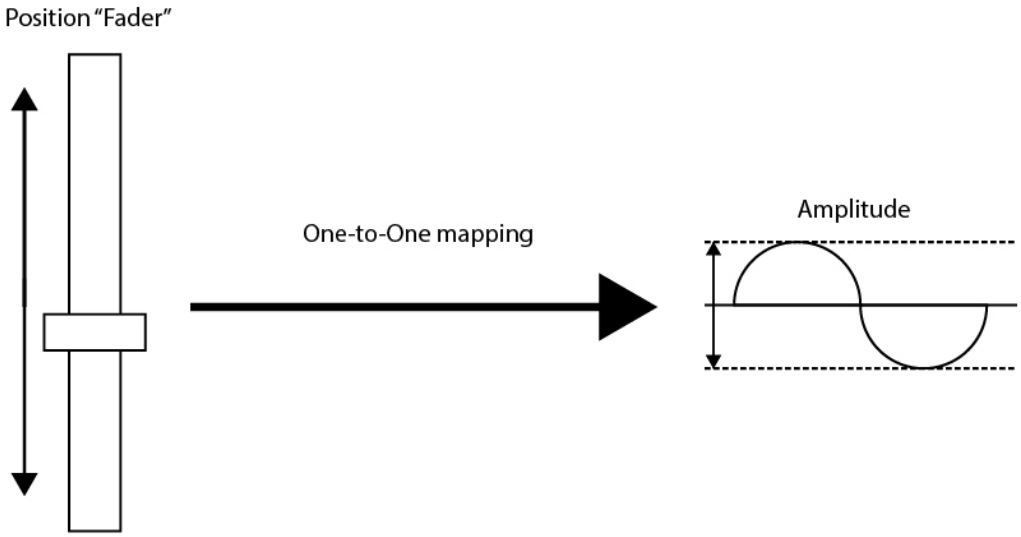

Figure 7. One-to-one mapping.
Adapted from Hunt and Kirk (2000, p. 234).

One-to-one describes the commonly implemented approach of a direct relationship between singular control components and a specific sonic parameter - for example, the physical position of a fader on a digital mixer that determines volume as shown in figure 7. Most MIDI devices and MIDI mapping systems in popular DAWs use this model as it provides detailed, independent control over every aspect of a sound. A collection of several one-to-one mappings can provide a thorough set of sound sculpting tools but which, due to the time-inefficiencies of adjusting many parameters individually, are often inappropriate for performance. For this reason I think of this category not only as "one-to-one" but also "one-by-one".

One-to-many:

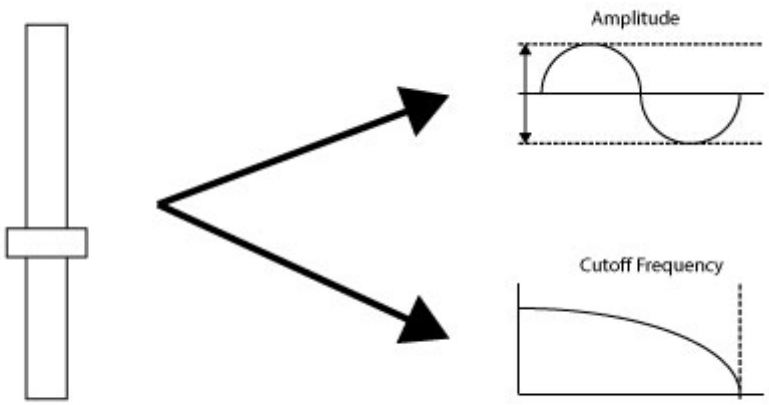

Figure 8. One-to-many mapping.
Adapted from Hunt and Kirk (2000, p. 234).

One-to-many appears to be an improvement over "one-by-one" as it describes a mapping of a single control to two or more synthesizer parameters at once - known as a "macro" in some DAWs. As we can see in figure 8, changing multiple parameters together can lead to very dramatic sonic results but this increase in the degree of movement also locks those parameters to move along a predefined pathway that cannot be altered during performance. The physical design of common hardware controllers presenting an array of MIDI faders or knobs entrenches this mode of mapping (and one-to-one) as the "engineering default" (Hunt and Wanderley, 2002) - each control represents a linear and bounded pathway.

Many-to-one:

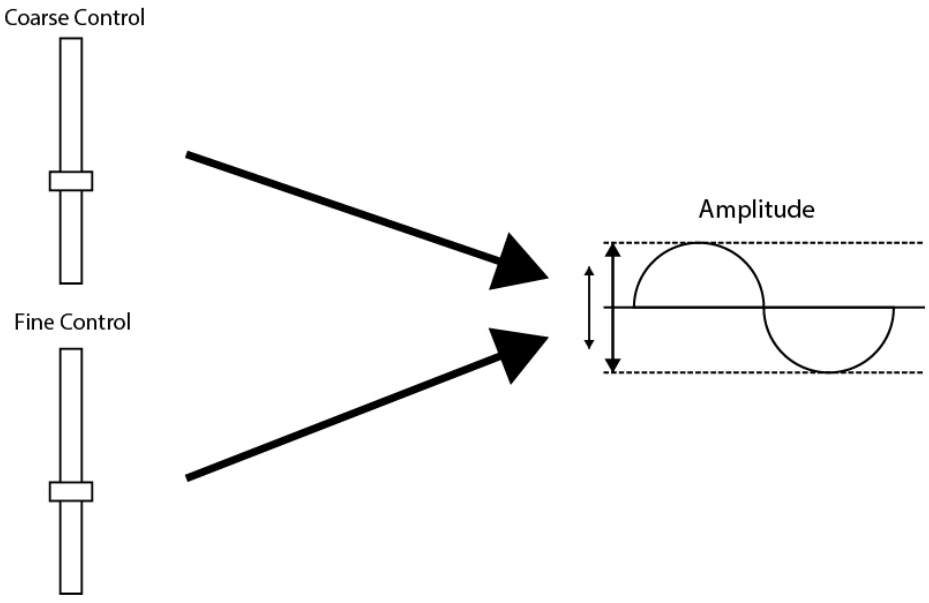

Figure 9. Many-to-one mapping.
Adapted from Hunt and Kirk (2000, p. 234).

Many-to-one mappings draw on insights from acoustic instruments where a combination of input factors affects a particular sound parameter. Hunt and Wanderley's example of reed instruments, where "loudness is a function of both lip pressure and airflow" (2002, p. 100), provides a useful reference point for how two independent variables can work together to provide nuanced control of dynamics. The equivalent approach in a DAW or synthesizer is often to provide both coarse and fine tuning controls over one parameter, such as in figure 9. Both approaches aim to improve efficiency but coordinated, intricate physical movements generate a much faster “data stream” than adjusting separate controls one-by-one (Pluta, 2008). With this technique, performers use a number of control inputs together to quickly and accurately hone in on the sounds they desire to produce .

Many-to-many:

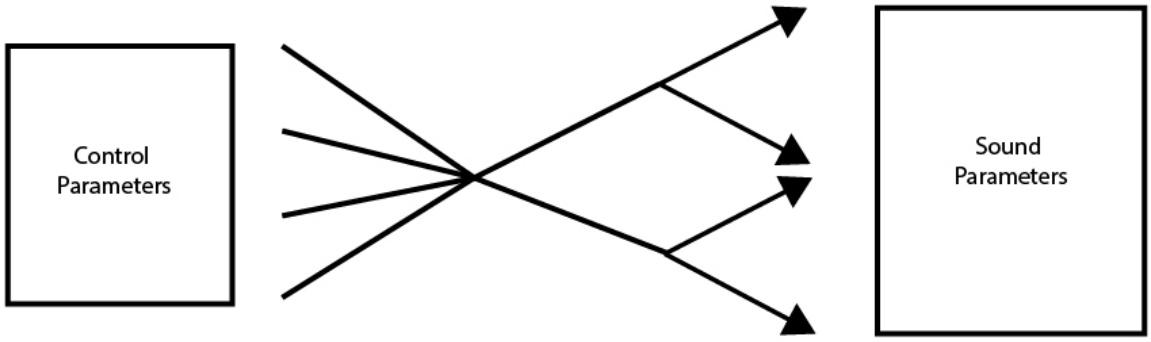

Figure 10. Many-to-many mapping.
Adapted from Hunt and Kirk (2000, p234).

The idea of a many-to-many mapping initially appears daunting and challenging. Integrating one-to-many and many-to-one mappings raises the number of possible mapping architectures exponentially, opening the door to a very wide variety of new digital instrument designs. Applying those same insights from acoustic instruments, the mapping designer should be able to build creative solutions to the difficulties posed by simple one-to-one mapping structures in performance. If we extend Hunt and Wanderley’s reed instrument example, lip pressure and airflow need not only be linked to dynamics alone. In particular, if lip pressure and airflow affect timbral or pitch parameters differently while working together to control volume, then the relationship between just a few simple input factors and sound parameters can quickly become very complex. Figure 10 shows how multiple control data streams are processed together in a mapping algorithm, rather than simply routing each of them to a sound parameter directly. One-to-one mappings are definitely easier to use at first, but their long term potential for expressive control is much lower than these sophisticated mapping techniques that take time and effort to master (Hunt and Wanderley, 2002).

Hunt and Wanderley propose that the advantages of many-to-one and many-to-many mappings makes them much more relevant to the design of digital musical instruments, while noting that the opposite has instead flourished in the controller and music software industry:

> “...complex mappings are more effective at eliciting a good performance from a human player when the performer is confronted with multiparametric tasks, more so than a series of one-to-one mappings which is often the engineering default.” (2002, p. 105)

It is important to realise that well-crafted, complex mappings actually simplify the perceived task for the performer, which might initially seem counterintuitive. Most DAWs provide facility for only one-to-one and one-to-many controller mappings, and therefore the number of steps involved in manipulating sound increases for more complex tasks - essentially pushing that complexity back onto the player. A sophisticated mapping system should instead absorb that complexity and make it equally straightforward to perform tasks of varying complexity.

Mapping designers can use Hunt and Wanderley’s techniques to improve the efficiency aspect of the expressivity equation. However, designing these kinds of mappings obviously requires much consideration and the benefits of such complexity can easily be diminished by an unintuitive or "opaque" mapping (Fels, Gadd and Mulder, 2002). Mapping designers that explore these four techniques require a framework to keep their designs (and down the line - performers) on track, focusing on how control solutions affect Cage's "total sound space".

Chapter 4

This chapter discusses strategies to enable the construction of sophisticated mapping systems that transparently channel the performer's control efforts towards desired outcomes and avoid designing complex mappings for the sake of complexity.

Managing Complex Mappings

The basic notion of mapping is a connection between interface controls and particular synthesizer parameters. Van Nort et al. (2014) propose that this perspective on mapping, concerned mainly with which controls are assigned to which parameters, be replaced with a "functional" approach. This alternative mode of thinking defines an instrument's behaviour as mathematical functions of control inputs, taking a more holistic view that mapping encompasses the overall interactive experience of playing an instrument. The one-to-one model is not consistent with this perspective. Instead of considering controls and parameters individually, Van Nort et al. suggest that "a musical instrument can be viewed as a collection of discrete and continuous control variables and their sonic effect" (2014, p. 8). These variables together become a "control space" that exists as a subset of "sound space". Figure 11 shows how these elements fit into the Digital Instrument Model. Van Nort et al. don't make any reference to Cage's "total sound space" but the link is clear and the concepts are complementary, if not identical. Synthesized sounds are generated from large lists of parameters - amplitude, frequency, filter cutoff, etc. - and the mapping designer needs to establish how the performer will manipulate this set as a whole. Viewing these as variables allows the mapping designer to see that the current state of the entire system is represented by a single point - a coordinate - in space of dimensions equal to the number of synthesizer parameters. The performer has access to the control space only, but as Van Nort et al. explain:

> "If we associate such a point in “control space” with a particular point in a corresponding “sound space” (i.e., a set of sound-synthesis variables), and we define a mapping function between spaces, then our action will in turn continuously drive the given point that exists in the sound variable space, thereby affecting the sonic output." (2014, p. 8)

Therefore the task of designing a transparent mapping system becomes a question of how to define the mapping function. The emphasis on space also places a high value on freedom of movement, consistent with the concept of transparency. The performer should be able to navigate the control space with a strong intuition of how this will affect the sound space.

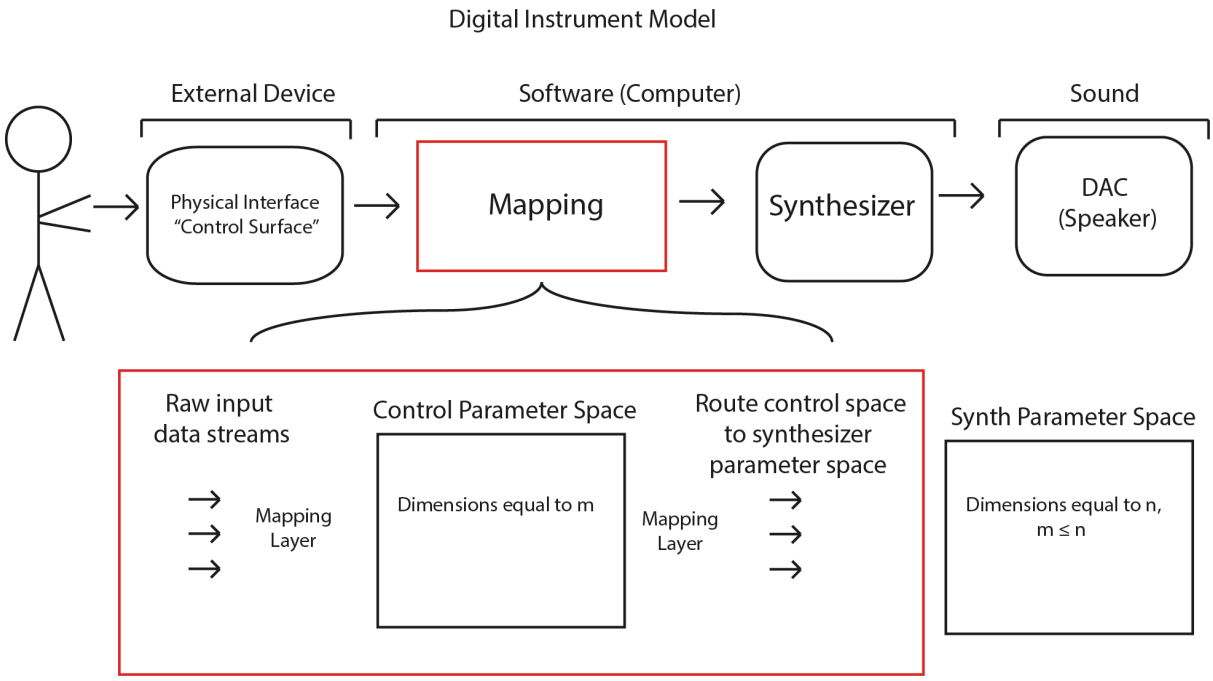

Figure 11. Extended DMI model.
Adapted from Wessel and Wright (2002, p. 12).

The common XY pad is a literal and helpful example of this space concept, at least for two variables/dimensions where a sound parameter is assigned to each of the X and Y axes. Provided with a target point on the pad surface (control space), a performer's hand can move directly to that point or just as easily take any path throughout the space. Contrasted against two encoders or faders that must be adjusted one-by-one, the XY pad simplifies the task without restricting the performer's freedom in the control space. Van Nort et al.'s concept becomes especially helpful as the number of parameters (and hence dimensions) increases, because it is much easier to understand the synthesizer's state as a point in space than memorise the entire list of parameter values.

Designing constraints in this model not only refers to the selection and elimination of dimensions that we allow the performer to occupy in the control space, but also how the performer is allowed to move around that space. A transparent mapping function should create logical associations between the performer's actions and the corresponding sonic outcomes, a relationship that Fels, Gadd and Mulder (2002) describe as control "metaphor". Employing metaphor "restricts and defines the mapping" (Fels, Gadd and Mulder, 2002, p. 3), strongly linking specific gestures to the sounds produced. Use of metaphor helps the performer to predict the outcomes of their actions.

Visual Feedback from the Synthesizer

While Hunt and Wanderley's four techniques accommodate the full set of pathways translating control inputs to sound parameters, mapping frameworks are not restricted to the one-way data flow the category titles imply. Mapping systems can generate a data stream (in addition to sound) that is routed back to the control device to be displayed visually. This visual information helps to alleviate the “large burden on the performer’s memory” (Wessel and Wright, 2002, p. 14) when the performer is required to complete a complicated sequence of actions for a desired outcome. Visual information can also enhance what Puckette (2012) terms the "feedback loop", where the performer continually analyses the results of previous actions in order to inform future decisions. But this visual information does not necessarily improve a mapping system in terms of transparency. Wessel and Wright (2002) reveal a trade-off between aiding and hindering the performer with a computer’s GUI during performance. Although the screen might display useful information, it can also disconnect the player from physical visual cues in an ensemble and distract from auditory feedback. Considering Hunt and Kirk's (2000) observation that expert traditional instrumentalists show a decreased reliance on visual feedback, to qualify for inclusion it must convey information on the state of the system that cannot be deduced from listening alone, and communicate that data as clearly and simply as possible.

In order to avoid unhelpful distractions, Wessel and Wright (2002) aimed to design mappings that remove the need for visual feedback altogether. However, in the years since their article was published, new alternatives to the computer display have become available that minimise the negative consequences of visual feedback. The arrival of the Monome grid in 2005 was a major advance because it integrated visual feedback within the control mechanism itself. Figure 12 shows the grid displaying light patterns directly on the playing surface to reveal processes occurring in the software that may not be entirely audible. While not nearly as detailed as a computer's GUI that displays text and images, the simple visual feedback system encourages the designer to create visual indicators of only the most important processes occurring in the software. The performer can read this information very quickly as a result and, by integrating it directly into the playing surface of the instrument, it is displayed literally under the performer's fingertips. The efficiency and transparency of these mappings enable the performer to access very complex states that would be otherwise unattainable. The capacity to transparently manage complex states is a powerful compositional tool that the performer uses to develop musical themes during live performance.

Figure 12. Monome grid displaying LED feedback.
Reprinted from https://monome.org/ Copyright 2017 by Monome.

Monome's success in expanding the capabilities of electronic instruments by freeing performers from their computer screens is reflected in the popularity of grid based controllers today. The Monome controllers have set the standard for state management in digital instruments, combining visual feedback with a tactile control mechanism. This design creates the feeling that players are directly manipulating these processes with their hands, which is the very definition of transparency.

Towards a complex instrument:

The above discussion has highlighted strategies to improve mapping transparency. Consolidating the many variables into a single point in "control space" and using metaphor helps the performer to predict how their actions will affect the sonic output. Mapping designers can also leverage the visual feedback possibilities of decoupled interfaces to represent the current synthesizer state. In the next chapter, I present three mapping systems, designed for *Electric Sheep*, that demonstrate my efforts to create an expressive instrument based on the techniques and strategies presented in chapters 3 and 4.

Chapter 5

In this chapter, I describe the mapping systems I have designed for the *Electric Sheep Texture Synthesizer*.

Electric Sheep Texture Synthesizer

The “synthesis engine” (Hunt and Wanderley, 2002) for *Electric Sheep* consists of a three track looping sampler. Its use involves selecting a portion of a waveform to be played back as a loop - a technique known as wavetable synthesis (Puckette, 2007). The three tracks allow for multiple samples to be layered to generate dense textural sounds. Each track contains 10 parameters that affect the sound output, these are:

- Position
- Window Size
- Playback Rate/Direction
- "Anarchy"
- "Anarchy Rate"
- Low Pass Filter Cutoff
- Low Pass Filter Resonance
- High Pass Filter Cutoff
- High Pass Filter Resonance
- Volume

There is also a filter and distortion/volume stage on the master bus. As mapping is the focus, I will not explain the functionality of these parameters further here. In total the sound parameter space contains 30 track parameters and an additional five master bus parameters. The three control interfaces that have been mapped to those parameters are intended to complement one another, and figure 13 shows how I arrange these controllers on stage to be used together as a complete instrument.

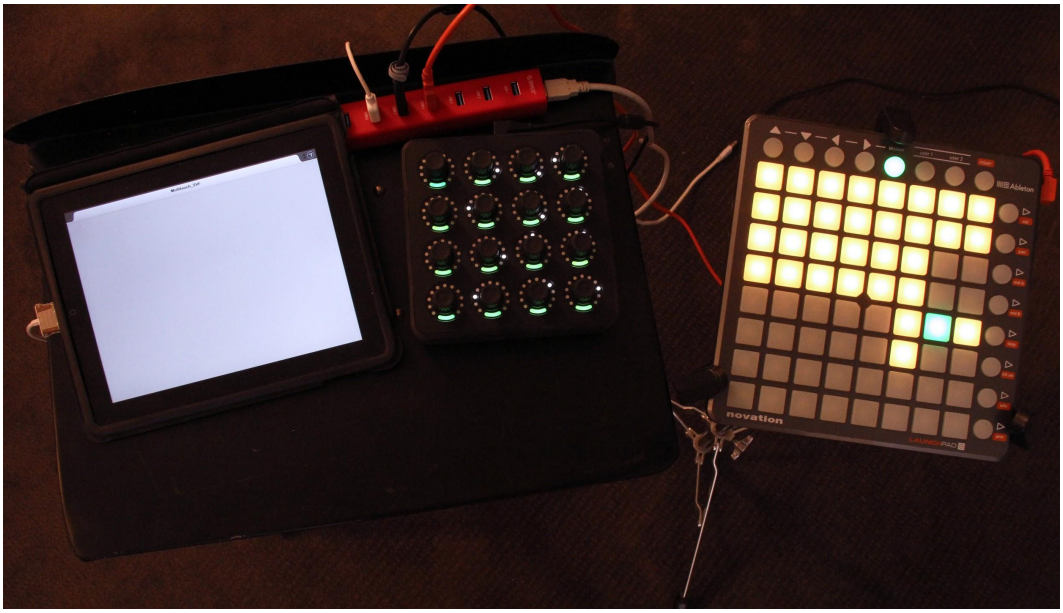

Figure 13. The controllers arranged together on stage.

Rotary Control – MIDI Fighter Twister

I chose to create a (nearly) one-to-one mapping for the MFT as a reference point. The encoders are linked directly to parameters given in the list above. The MFT's push + turn functionality actually allows two parameters to be assigned to a single encoder, and I also use multiple "banks" to fit the large parameter set onto a smaller device. Although this mapping is technically one-to-many, the process of adjusting parameters remains "one-by-one", which I consider close enough to one-to-one. Figure 14 shows encoders assigned to several sound space parameters:

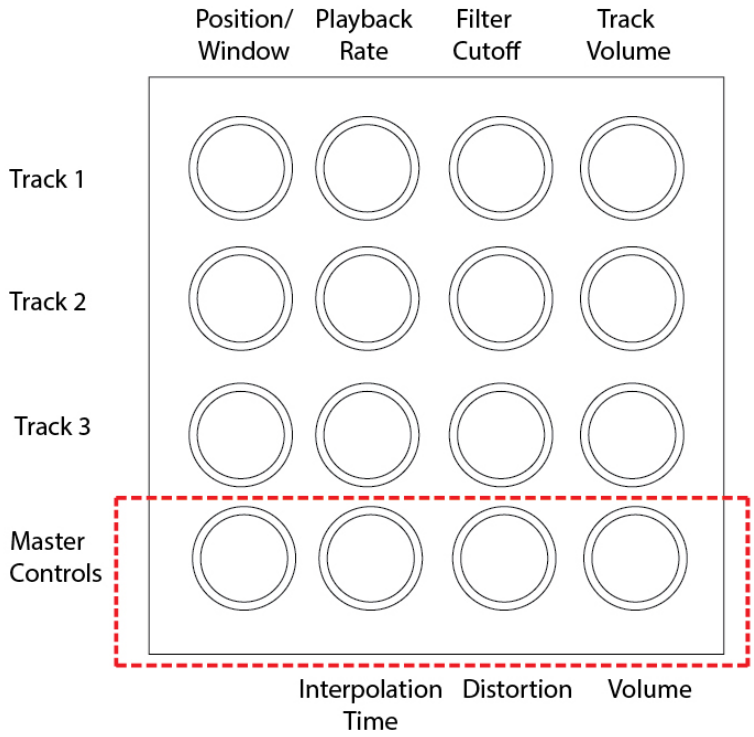

Figure 14. Bank of rotary encoders.

Controller banks are arranged so that the most commonly used controls are accessible in all states and the top three rows provide identical controls for each track. This reduces the number of individual controls a player must memorise. If one is familiar with the parameter assignments, the transparency of this mapping is reasonably high due to the visual feedback. LED rings surrounding the encoders give a clear visual indicator of each parameter value and it is easy to understand how these will change as the encoders are turned. However, the limits of simple visual feedback are encountered with the "position/window" controls at the left of figure 14. These indicate the selected region of a waveform for playback but, without viewing the waveform itself, the performer is essentially operating this control blindly.

Memory Matrix - Launchpad

The Memory Matrix stores and recalls entire sets of sound parameters. After adjusting individual controls on the MFT, the performer creates a "memory", which is a snapshot of the 30 track parameters at that moment in time. Each memory is assigned to a button on the grid as they are created, which means the actual mapping layout is built up or reconfigured during a performance. Hunt and Wanderley (2002) acknowledge that, although this kind of “dynamic” mapping is useful because it is so adaptable, it is potentially more difficult for the performer to learn - since it is constantly changing. The grid's visual feedback minimises this problem as the colour coding, shown in figure 15, helps the performer to keep track of the location of stored memories. When a free or "empty" button, which is unlit, is assigned a memory, its LED switches on to yellow to indicate the slot is now occupied and a memory is available for recall at this location. When the player selects this button and activates the parameter values stored, the LED switches to green to indicate it is the currently active memory. This functionality is identical to that of preset managers commonly found in modern synthesizers but, unlike the usual (and not at all interactive) menu provided to select presets, it is used more like a sample-pad or drum pad to "play" the presets. When snapshots are recalled, the changes in parameters are also reflected visually on the MFT as the LEDs for each encoder update accordingly.

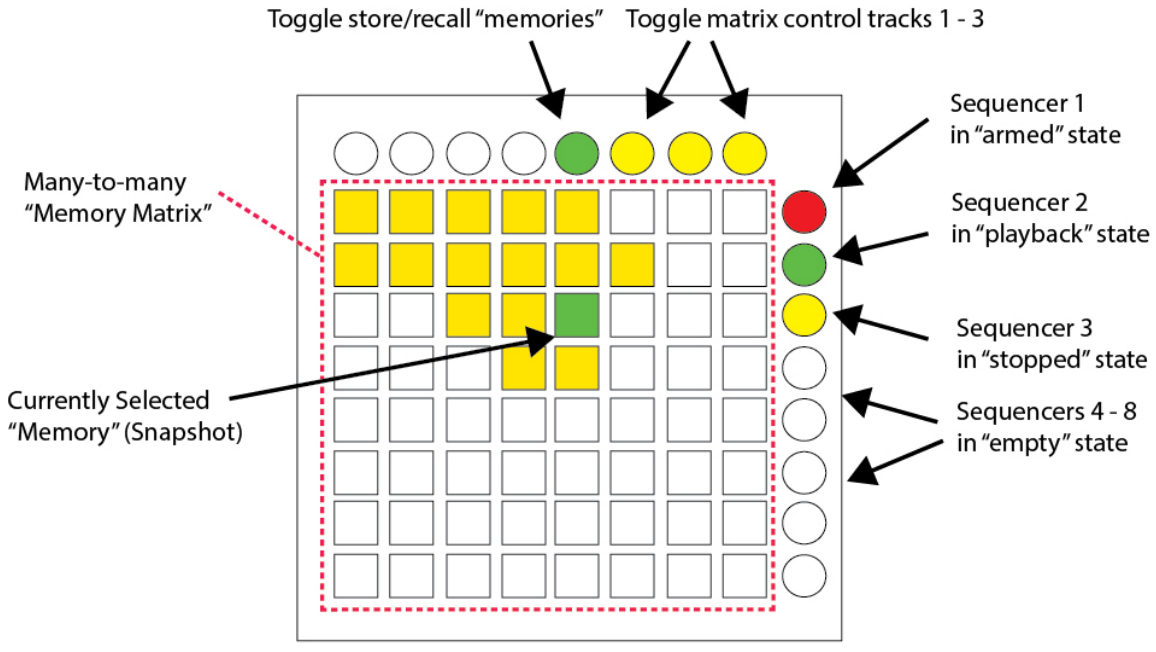

Figure 15. Colour coding of Memory Matrix functions.

This mapping is "many-to-many" because the grid presents an array of 64 buttons that each affect the same collection of 30 synthesizer parameters differently. Despite this complexity

however, the Memory Matrix is very simple to play. The key to this simplicity is the metaphor implicit in the idea of a memory. This metaphor, supported by the visual feedback that shows whether buttons are free, occupied or active, creates a strong association in the performer's mind between a button and a particular sound. Each memory also represents a single point in Van Nort et al.'s multi-dimensional "control space" and triggering various memories is analogous to jumping between these points.

The eight pattern recorders - accessible via the right-most column on the Launchpad - enable simple sequences to be played in and looped, and are especially useful when using Electric Sheep as an auto-accompanist. The mapping designed for these buttons again highlights the usefulness of visual feedback. A single button operates all the necessary sequencing commands by logically switching its function based on the current state of the sequencer, as can be seen in the colour coding of the right column in figure 15.

My goal when designing this mapping system was to distill complicated processes down to simple, colour coded functions that guide the performer to focus on sound and not on the many parameters that contribute to it - that role is left to the MFT mapping.

Multi-touch - iPad

The multi-touch mapping processes data generated by hand gestures and transfers it to the state variable filter on the master audio bus, which sculpts the timbre of the mixed output from the sampler tracks. The mapping does not detect gestures explicitly, due to difficulties with recognition and latency, but follows the suggestion of Mulder, Fels and Mase (1997, as cited in Hunt and Wanderley, 2002) to instead analyse "continuous changes" in the raw input stream and infer the state of gestural variables in real time. This gesture-driven mapping uses metaphor by creating the illusion that the performer is directly shaping the sound with their hand. The immediacy and intimacy of this connection is intended to help the performer enter Puckette’s "feedback loop” without any need for visual feedback.

As demonstrated by the XY pad example given in Chapter 4, touch controllers provide a very literal, physical translation of the space concept. Although the screen's flat surface is only a two dimensional plane, multi-touch detection enables additional dimensions to be folded down into this control space. I designed a "many-to-many" mapping that takes raw touch points - shown as circles in figure 16 - and calculates three intermediate variables:

1. stretch - the average distance between the touch points (the value is at a maximum when fingers are stretched out over the screen)
2. rotation - detecting the clockwise or counterclockwise rotation of the fingers
3. distance from centre - the distance between the centroid of the touch points and the centre of the control space

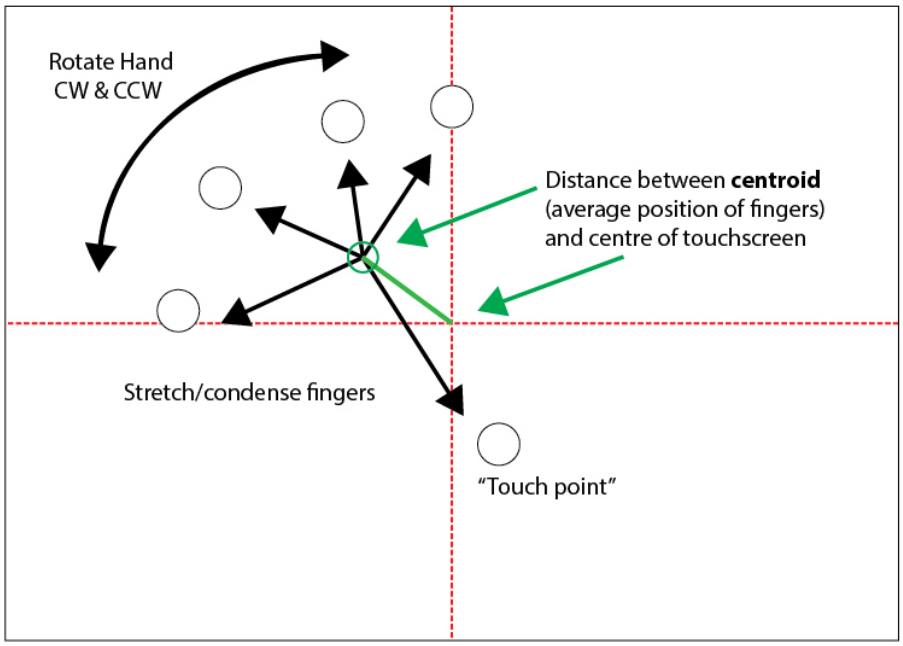

Figure 16. Touch surface interaction.

These three variables are mapped respectively to filter cutoff, crossfade between low and high pass filter outputs, and resonance. This mapping creates an implicit hierarchy of the three parameters, since there are varying degrees of physical freedom inherent to the gestures that determine these variables. "Stretch" is a clear metaphor for opening up or closing the cutoff frequency and is easily manipulated by opening and closing the fingers. Rotation of the hand is slightly unnatural when maintaining contact between the fingertips and the surface but seems adequate for crossfading either gradually or rapidly between filter types. Since there is limited screen real estate, the hand is limited in its ability to achieve high resonances. However resonance is not the kind of parameter that a performer typically needs constant, detailed control over. The fact that resonance is manipulable together with the other parameters is crucial to efficiency in performance, but by consigning it to a lower weighting the controller is focused on the parameters most commonly used. The integration and weighting of control variables also creates a high degree of nuanced timbral variation thanks to subtle changes in resonance that occur unavoidably when the player adjusts cutoff or crossfade.

Chapter 6

In this chapter I present reflective analyses on the mapping systems in *Electric Sheep*, evaluating the efficiency and transparency of controls under each of Cage's determinants. Following this are some general conclusions to the study.

Evaluation through Live Performance

Live performances and regular recording sessions have been an important part of the development cycle and evaluation process. The first performance using an early version of the software occurred on May 4, 2017 at Uptown Jazz Cafe in Melbourne, Australia and a second performance at the same venue occurred on September 7, 2017. Performances and recordings were mostly carried out in duo settings, designed to test *Electric Sheep* in its intended environment as a textural accompanist rather than for solo performance. Tracks 1, 3, 5 and 7 in my creative folio therefore feature me performing live on both *Electric Sheep* and drum set together. Tracks 2, 4 and 6 also present *Electric Sheep* in solo performance to provide raw examples of the kind of textures the instrument is capable of producing. The cycle has resulted in many revisions of the mappings and the following reflections explore difficulties encountered in performance, my attempts to address these, and remaining challenges.

Frequency

Frequency in *Electric Sheep* is primarily controlled via the one-to-one mappings between encoders and playback speed. In performance, I realised that rotary encoders, whilst useful for fine-tuning, are very limiting pitch controls. Playback speed control via the MIDI Fighter Twister is continuous and precise, so for non-tuned material (i.e. percussion samples) these controls are adequate to slow or reverse playback. The lack of an ability to jump by intervals however becomes problematic when dealing with tonal material. This has consequences for the role that *Electric Sheep* can perform in an ensemble with tuned instruments. As a texture synthesizer, *Electric Sheep* is well equipped to generate sustained, evolving sounds and can apply its techniques to tonal material to create sustained harmony. However, this harmony is difficult to shift and can become static, holding the entire ensemble back from introducing new harmonic material. I used the Memory Matrix to partially overcome this limitation in *Levering*, track 7 in my folio submission. By storing several “memories” using recorded piano chord samples, I managed to prepare a variety of harmony options for performance. Using such prescribed harmony would imply a highly composed performance, however I found that improvising within strongly defined boundaries inspired me to experiment with many different structural arrangements of the limited harmonic material. Overall however, frequency control

in *Electric Sheep* is neither efficient nor transparent due to the one-to-one frequency controls that are mainly appropriate for tuning only.

Amplitude

Wessel (2010) argues that “force to amplitude” mapping is a necessary component of expressive digital instruments. Including this mapping preserves the transparent link of physical effort with volume that is familiar to acoustic instrumentalists, making new instruments easier to comprehend and learn more quickly. Initially I naively ignored this argument, believing that I could create more expressive mappings that did not rely on differences in force to control amplitude. However, none of the three hardware controllers used with *Electric Sheep* provide any kind of velocity detection. Consequently, in performances, I quickly found dynamic expression to be generally poor. Like frequency, direct amplitude control is limited to the one-to-one encoders adjusting track volumes, distortion and master volume. I understood in the design process that these rotary controls would be unsuitable for generating amplitude envelopes in real-time, as envelope shapes evolve much more rapidly than one can rotate an encoder. Rather, they are mixing tools to adjust the overall balance and volume limit. Kirn (2014) points out that although audio samples themselves contain internal dynamics, performing solely with the kinds of on/off switches found on a grid interface creates a situation where the "aesthetic of the music is essentially flat". Track 4 of my folio, *Interlude*, makes heavy use of these internal dynamics and I actually disagree with Kirn on this point. I found that exploiting internal dynamics unlocked wide and unpredictable dynamic variation - demonstrated in *Interlude* - and by using those basic mixing tools I gained at least a loose degree of control over this. However, I do recognise now that *Electric Sheep* lacks efficient and transparent dynamic control that could be easily rectified with the addition of velocity mapping.

Duration

> "…duration is the only characteristic of sound that is measurable in terms of silence." (Cage, 1955, p. 13)

Electric Sheep reveals another major flaw when examined in terms of duration. The playing technique of triggering sustained textures essentially privileges sound over silence. It is important to realise that digital instruments do not require the performer to continually supply energy to sustain output. As with my conclusions on force to amplitude mapping, I came to appreciate how this requirement in acoustic instruments underpins not only dynamic expression but also rhythm and phrasing in music. I noticed a tendency in early recordings to fill every moment with sound, giving little room for silence. Silence is so significant because it

activates duration, which in turn enhances both amplitude and morphology. By overlooking silence I had inadvertently crippled expressivity in at least three determinant categories - amplitude, duration and morphology. Wessel and Wright explicitly confirm the importance of silence, declaring:

> "One of the most critical features of any musical control system is a silencer, a mechanism that allows the performer to gracefully shut down a musical process." (2002, p. 14)

Fortunately, the flexible Memory Matrix easily provided this capability and I quickly adopted a performance habit of ensuring the first button on the grid, at top left, always recalls the state of silence. This assignment is now my first move in any performance and I find that of any of the 64 grid keys, this is used by far the most. This mapping also demonstrates the efficiency of the Memory Matrix, as it allows the performer to access silence instantly at any point during a performance. It has been an important personal discovery that frequent use of silence actually increases the perceived impact of the sounds that do occur. Track 1 in my folio, *Switch*, illustrates this point by interspersing blocks of harsh electronic textures and intense drumming with sudden periods of silence. Compared to track 2, *Mono*, which contains very little silence, track 1 has a much more audible structure (morphology) and the extreme dynamic changes have a jarring effect on the listener. Clearly, an efficient silencer is absolutely necessary in an expressive digital instrument.

Timbre

As *Electric Sheep* is a texture synthesizer, timbral control is its strength and this is reflected in the mapping system. The two distinct filter stages in *Electric Sheep* provide a telling comparison of one-to-one and many-to-many mappings. I found that the MFT filter controls, with separate encoders to adjust each of the cut off frequencies and resonances, were cumbersome relative to the multi-touch filter controller. Of all the controllers the multi-touch is mapped to the fewest number of sound parameters, only three - cutoff, crossfade and resonance. Yet their interwoven mapping gives the performer the ability to achieve complex timbral effects very easily. It is equally straightforward to manipulate one or two parameters in isolation or even all three at once - and this is done with only one hand. The fixed hierarchy of parameters in the multi-touch mapping might at first seem to limit the performer, but in performance it definitely focused me on the highest priority parameters - cutoff and crossfade. As Magnusson (2010) reminds us, any well-conceived instrument must make compromises about which aspects of sound design to prioritise at the expense of others.

The expressivity of the multi-touch interface in controlling filters meant I ended up using it as a proxy for volume control. Although not ideal, by filtering the frequency spectrum it was possible to have reasonably fluid control over dynamics - at least by comparison to the volume encoders. I used this technique to create the gradual fade in and fade out at the start/end of folio track 7, *Chains.* This multi-touch mapping is a clear example of a complex mapping that significantly improves both efficiency and transparency over a one-to-one approach. The resulting high degree of timbral expression made this the most enjoyable and compelling control system to use with *Electric Sheep.*

Morphology

> "...how the sound begins, goes on and dies away..." (Cage, 1957, p. 9)

Morphology is the most loosely defined of Cage's determinants and hence has the widest interpretative scope. The content of samples loaded into *Electric Sheep* has the biggest influence on how the sound produced evolves over time, but aside from selecting samples the performer has little control over this aspect during performance. The samples triggered unfold on their own, with no further intervention from the performer. Yet morphology - or structure - also occurs on a larger scale and in the case of my mapping system I take it to encompass sequencing and transitional controls. These are compositional elements, affecting arrangement rather than content. Musicians have rarely demanded acoustic instruments handle musical structure as this has always been the instrumentalist's responsibility in performance (though the player-piano comes to mind as an important exception and precursor to digital sequencers). While it is debatable whether sequencing erodes the integrity of a musical performance or is simply a tool for structural manipulation, it is true that it transfers a degree of control away from the human performer and passes that to the computer. A major challenge that I did not anticipate when designing the sequencers was to maintain a balance of control between the computer and human performer. In folio tracks 1, 3, 5 and 7 where I played drums and used *Electric Sheep* as a semi-autonomous accompanist, the sequencers were useful because they allowed the synthesizer to operate hands free. However, this loss of control made it impossible to interact further with the instrument once a sequence was running.

In response I refined the mapping system in two ways to regain a balance of manual and automatic control. First of all, I created toggle switches to disengage each of the three sampler tracks from grid control. This allows the performer to initiate sequences on one or two tracks while leaving those remaining under manual control of the encoders on the MFT. For example, in *Mono* I set up a looping bass sequence on sampler track 3 and proceeded

to improvise over this, using the position and window controls for sampler track 1 to create stuttering, fragmented sounds that contrast the monotonous bass rhythm. My second refinement was to add an optional override that temporarily disengages the sequencer whenever I interact with the grid by pressing a key. These changes definitely helped to improve thematic development possibilities, as I can now interrupt ostinatos/loops to inject variation and improvise over them using disengaged tracks.

With these refinements, the Memory Matrix is both an efficient and transparent controller for structure - much more so than a one-to-one sample-pad that it so closely resembles. It is a challenging tool to learn but rewards the persistent player with the expressive power to execute dramatic and sudden shifts in texture just as easily as subtle and gradual transformations.

Discussion of Results and Conclusions:

The many criticisms outlined above might suggest that *Electric Sheep* fails to provide satisfying expressivity in Cage's "total sound space". It is true that control in the frequency and amplitude determinants is severely inhibited by the lack of transparent or efficient mappings. Still, although limited control in these areas restricts the scope of *Electric Sheep*, it alternately provides the opportunity for chance and the sample content to play a role in adding variation. The relative strength in timbral manipulation is more important however, and *Electric Sheep* has proven to be a highly expressive textural voice in performances, as is evidenced in my folio of creative works.

It is clear that complex, sophisticated mappings have the potential to greatly enhance the performer's feeling of connectedness and engagement with sound. They do this by hiding that complexity and removing barriers to fluency. Practice is still necessary to develop "control intimacy" (Hunt and Kirk, 2000, p. 232), but practise time must not be spent learning to deal with control sets that unnecessarily hinder or slow down interaction. It was clear that the control fluency I achieved playing the grid and touchscreen interfaces in *Electric Sheep* was largely thanks to the relatively efficient mappings and the grid's visual feedback. The speed of interaction on these interfaces considerably outpaced the one-to-one controls on the MIDI Fighter Twister, however I would not say those were entirely useless. The one-to-one interface was very helpful to have available as secondary controls, useful for occasionally fine tweaking parameters, though I spent the majority of time with one hand on the touchscreen and the other on the grid. It was comforting to have access to that extra level of detail when required, but not rely on it.

One important observation is that the encoder bank (MFT) and Memory Matrix (grid) are at opposite extremes of control. The rotary encoders provide very fine adjustments over individual parameters while the grid mapping uses a much blunter method to change large sets at once. The multi-touch mapping on the other hand seemed to find an excellent middle ground - efficiently allowing the performer to adjust either individual or multiple parameters continuously, and employing a strong metaphor to achieve transparency. The expressivity that results strongly supports my proposed equation:

Expressivity = Transparency x Efficiency

I believe that digital instruments designers can and should use Cage's determinants as a guide and evaluative tool because efficient and transparent mappings for each of these categories together contribute not only to expressivity, but also to a very *musical* instrument. An expressive musical instrument is the outcome of successfully designing a mapping system for live performance. Ultimately this thesis is intended to not simply propose an equation but reveal how mapping designers can apply it to popular music controller devices available today. I hope that readers of this thesis will gain insights as to how they could go about designing their own mapping systems through observation and critical analysis of my development process for the *Electric Sheep Texture Synthesizer*.

References

Bowcott, P. (1996). Interfaces for composition - What about the composer? *Contemporary Music Review, 15*(3-4), 27-37.

Brown, A., & Sorenson, A. (2009). Integrating Creative Practice and Research in the Digital Media Arts. In H. Smith & R. T. Dean (Eds.), *Practice-led Research, Research-led Practice in the Creative Arts* (pp. 153-165). Edinburgh: Edinburgh University Press.

Burnand, D. (2009). MIDI. *Grove Music Online. Oxford Music Online*: Oxford University Press.

Cage, J. (1955). Experimental Music: Doctrine. *Silence* (pp. 13-17). Great Britain: Calder and Boyars.

Cage, J. (1957). Experimental Music. *Silence* (pp. 7-12). Great Britain: Calder and Boyars.

Considine, D., & McMillan, G. (2000). 5.1.3 Rotary Motion. *Process/Industrial Instruments and Controls Handbook* (5th ed.). New York: McGraw-Hill.

Constanzo, R. (2016). *Composition, Performance, Improvisation, and Making Things, sitting in a tree : Me-Me-Me-Me-Me-Me-Me.* (PhD), University of Huddersfield. Retrieved from http://www.rodrigoconstanzo.com/thesis/

Crabtree, B., & Cain, K. (2017). Monome. Retrieved from https://monome.org/

Davies, H. (n.d.a). “Controller” *Grove Music Online. Oxford Music Online*: Oxford University Press.

Davies, H. (n.d.b). “Electronic instruments” *Grove Music Online. Oxford Music Online*: Oxford University Press.

Fels, S., Gadd, A., & Mulder, A. (2002). Mapping Transparency through Metaphor: Towards More Expressive Musical Instruments. *Organised Sound, 7*(2), 109-126.

Hunt, A., & Kirk, R. (2000). Mapping Strategies for Musical Performance. In M. Wanderley & M. Battier (Eds.), *Trends in Gestural Control of Music* (pp. 231-258). Paris: IRCAM - Centre Pompidou.

Hunt, A., & Wanderley, M. (2002). Mapping performer parameters to synthesis engines. *Organised Sound, 7*(2), 97-108.

Kirn, P. (2014). Watch the Wonders of Grids, as monome Makers Defend Minimal Design. Retrieved from http://cdm.link/2014/06/watch-wonders-grids-monome-makers-defend-minimal-design/

LINFO. (2004). GUI Definition. Retrieved from http://www.linfo.org/gui.html

Magnusson, T. (2010). Designing Constraints: Composing and Performing with Digital Musical Systems. *Computer Music Journal, 34*(4), 62-73.

Manning, P. (n.d.). “Sampling” *Oxford Music Online*: Oxford University Press.

O'Modhrain, S. (2011). A Framework for the Evaluation of Digital Musical Instruments. *Computer Music Journal, 35*(1), 28-42.

"LED, n." (n.d.) In *Oxford English Dictionary Online*: Oxford University Press.

OpenSoundControl. (n.d.). Introduction. Retrieved from http://opensoundcontrol.org/introduction-osc

Pluta, S. (2008, June). Maximize Information Flow: How To Make Successful Live Electronic Music. *New Music Box*. Retrieved from https://nmbx.newmusicusa.org/Maximize-Information-Flow-How-to-Make-Successful-Live-Electronic-Music/

Puckette, M. (2007). *The Theory and Technique of Electronic Music*. Singapore: World Scientific Publishing Co. Pte. Ltd.

Puckette, M. (2012). Design choices for computer instruments and computer compositional tools, online video. *CIRMMT Student Symposium and General Assembly 2012 External Keynote Address*. San Diego, USA: University of California. Retrieved from https://www.youtube.com/watch?v=ZLACjtOpe0Q

Strawn, J., & Shockley, A. (2014). "Computers and music" *Grove Music Online. Oxford Music Online*: Oxford University Press.

Van Nort, D., Wanderley, M., & Depalle, P. (2014). Mapping Control Structures for Sound Synthesis: Functional and Topological Perspectives. *Computer Music Journal, 38*(3), 6-22.

Wessel, D. (2010). Designing musical instruments that privilege improvisation, online video. *CIRMMT Distinguished Lectures in the Science and Technology of Music*. Berkley, USA: University of California. Retrieved from https://youtu.be/uGASpqTXz4g

Wessel, D., & Wright, M. (2002). Problems and Prospects for Intimate Musical Control of Computers. *Computer Music Journal, 26*(3), 11-22.

List of Audio Examples

Track 1. *Switch*

Track 2. *Mono*

Track 3. *Ikue*

Track 4. *Interlude*

Track 5. *Duo*

Track 6. *Chains*

Track 7. *Levering*