UNIVERSITY OF CALGARY

Sonic Matter: A Real-time Interactive Music Performance System

by

Abdullah Safa Soydan

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE

DEGREE OF MASTER OF MUSIC

GRADUATE PROGRAM IN MUSIC

CALGARY, ALBERTA

DECEMBER, 2018

Abstract

This research proposes a multi-channel interactive performance system called *Sonic Matter*. The system embodies newly developed motion tracking technology in accordance with well-accustomed MIDI technology in quest of achieving greater expression in live electroacoustic music performance. *Sonic Matter* is an open system, which offers a great deal of artistic flexibility and control due to its modular structure. The system gives its performer the ability to manipulate many qualities of sound in real-time.

This thesis presents the background for the research, discusses the design considerations, details what each module does, and presents the author's musical compositions created with *Sonic Matter*. The collection of five compositions, *Sonic Matters*, was premièred in the Sonic Arts Lab at the University of Calgary on June 14th, 2018.

Acknowledgements

Firstly, I would like to thank my supervisor, Dr. David Eagle, for his guidance, assistance, support, and most importantly, his patience throughout the entire creative process. All your advice and suggestions are welcomed and much appreciated. I would also like to thank Dr. Laurie Radford for being a source of inspiration as well as keeping me on track with my thesis.

Thank you, Anthony Reimer for being extremely understanding and flexible with my schedule in the IAML. This would be very difficult without your assistance.

I would like to thank Melike Ceylan for her compassion, kindness, encouragement, and support. Your productive feedback has enabled me to always critique my work and writing. Thank you very much for making my life easier.

Finally, thanks everyone who attended the première of *Sonic Matters* and gave me valuable feedback to question my research and creative works from different perspectives.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

Starting from a few decades ago to this day, computers have become an integral part of music production and performance. The advancements in electronic components and computer technology made some breakthroughs possible in music practice; they also laid the foundation for numerous new musical genres. Musicians have always found some ways to challenge the capabilities of computers to attain pleasing results, spending hours, even days in studios in return. As opposed to the past, testing the limits of computers is becoming harder since their computational power is growing significantly every passing day: they are now able to deliver days of work in real-time.

The research presented in this thesis encourages the creative artist to easily explore the musical and sonic potential concealed in one's recordings in real-time by harnessing the power of today’s computers. The system is called *Sonic Matter* since it grants its performer the ability to shape the sound as if one cuts rough stones into diamonds to reveal their true potential. *Sonic Matter* provides the means of spectral, rhythmic, dynamic, and spatial manipulation.

This thesis consists of five chapters: Chapter 1 introduces the research, describes the motivation towards the research, outlines the methodology, and discusses its contributions; Chapter 2 establishes the grounds for the research with historical and technical background; Chapter 3 elaborates on the system, discusses some significant considerations taken throughout the design process, and explains how each module functions; Chapter 4 describes the creative process followed in composing the musical works, explains and analyses each piece; Chapter 5 discusses the outcomes of the research along with the future work and potential for further research.

1.1 Motivation

I began the MMus Sonic Arts program at the University of Calgary in September 2016. During my first year, I concentrated on making interactive music systems and instruments that have hardware and software controllers to shape some qualities of the sound (Weale), including extending the capabilities of conventional instruments by means of computers and microprocessors (Soydan, *Delays*; Soydan, *Granulate*). My primary objective has always been to improve musical and sonic expressivity while maintaining the intrinsic playability of the instrument for the performer. I endeavoured to achieve this goal by designing simple interfaces that are easy to learn and use.

After composing two electroacoustic music pieces and experiencing the challenges and limitations of existing audio sequencing software, I realized that I needed to custom design a complete system to create and perform my music. The thesis research is an outcome of this necessity and it grows out of techniques learned in my previous works and sonic exploration.

In the design of software instruments for live situations, there is always a compromise between the use of computational power and the sound/processing quality. Based on this, I attempted to reach as close as possible to a non-real-time music system in terms of sound quality and processing capabilities. Furthermore, I set the sound processing in Trevor Wishart's *Imago* (Wishart, *Sonic Art Pieces*) as the level to reach, a piece composed with his very own non-real-time music system, *Sound Loom* (Wishart, *The Sound Loom*).

1.2 Scope

This thesis presents a new interactive music performance system called *Sonic Matter* and details its design process, both from human-computer interaction and live electronic performance perspectives. It also draws attention to some design factors affecting the musical outcome in an interactive performance system, along with the strategies to address them.

1.3 Goal

The research presented in this thesis aims to explore ways of creating original and engaging sounds as well as sonic textures with newly developed computer technologies. It also investigates how the design of an interactive performance system affects the performance and the artistic potential, both from the angles of easiness and musical expressivity. In addition, this thesis aims to demonstrate the music-making capacity of *Sonic Matter* with the accompanying creative works.

1.4 Methodology

Very often when creating an interactive composition or having a performance involving technology, the system should be composed first (Dudas 30). This strategy affects the musical possibilities directly; that's why the decisions made during the design process should be in the same direction as the performer's artistic style and musical intentions. The making of the system is also a part of the creative artistic process: it is more about the artist's approach rather than the availability of technology. As the first user to compose and perform with this system, I used my musical approach to influence the design of the system.

The methodology of the research can be broken down into four main components:

1. **Study**

At the first stage of the research, I reviewed the existing systems that my favourite electroacoustic music composers used to create acousmatic works, such as Trevor Wishart's *Sound Loom*, and Barry Truax's *POD/PODX System* (Truax, *POD*), as well as the works presented at the *NIME (New Interfaces for Musical Expression)* conference (NIME). This study in the field enabled me to understand the approaches taken thus far. It also helped me refine my ideas on the design by uncovering the strengths and limitations of each system.

2. **Design**

I programmed the framework of the system and integrated components, comprising a collection of designed effects and sound generators using the visual programming language *Max* (Cycling '74, *Max Software*). The designed effects enable the performer to control some aspects of sound processing and musical character, namely, the spatial, dynamic, spectral, and rhythmic qualities of the sound, as well as the timing. These effects are built as modules to give performers the maximum flexibility and control in signal flow and sonic creation.

3. **Control**

Having programmed the modules, I set out to implement a simple but effective way to control the performance aspects. I decided on the input devices by their convenience-flexibility-expressivity balance. Ergonomics was another critical factor in designing the controls to allow the performer to use the system intuitively and conveniently. An essential guiding principle was to empower performers to reflect their artistic and aesthetic values on the performance beyond the technicalities (Kimura 75). Another principle followed was to offer the performer a broad palette to choose from, as well as providing the facility to lead to virtuosity with continuing effort and practice. The system comprises a *KORG nanoKONTROL2* MIDI controller and *Leap Motion* as control devices.

4. **Experiment – Explore – Express**

The final stage was a period of experimentation, performance and creation, culminating in a concert of new electroacoustic works using *Sonic Matter*. I prepared audio samples consisting of close recordings of everyday objects, instruments, and female voice. I then tested the potential of the system with the samples, and explored the sonic possibilities that are usable in my compositions. These experiments proved that the decisions made in the second and third stages were somewhat arbitrary until the system can be learned, adapted and practiced to create musical results. This stage provided valuable information about the performance aspects, information that was used to review the previous stages until a greater and satisfactory musical expression was achieved (Figure 1.1). I composed five pieces and performed them in a concert only after I was comfortable in expressing my musical and artistic ideas using the system.

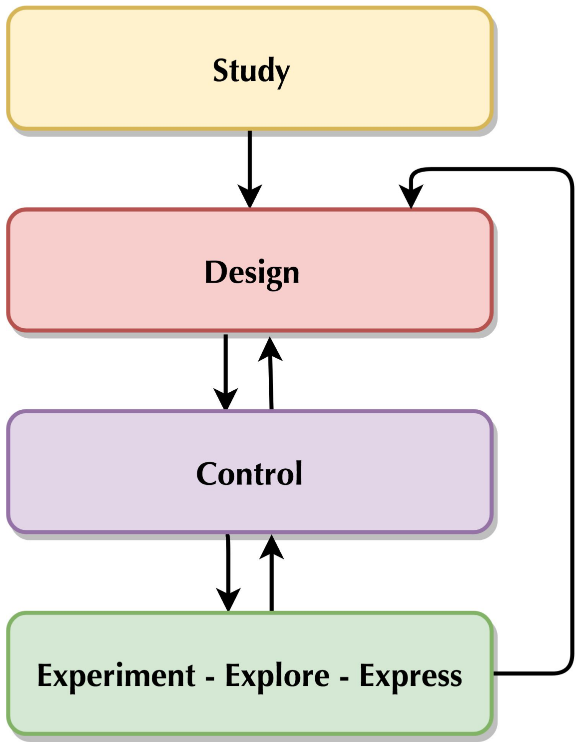

Figure 1.1: Graphical representation of the methodology used for the research

1.5 Contributions

Some significant contributions of this thesis to the field of electroacoustic music are:

- It provides a brand new real-time interactive music making system for performers, composers, and everyone who has an interest in sonic exploration. It also offers an alternative path to electroacoustic music performance practice: the system can facilitate improvisation with its potential in sonic transformation.
- *Sonic Matter* is an open system allowing users to create new modules or customize the existing modules for their convenience to develop their performance setups.
- Performers can load their own sounds into *Sonic Matter* to transform and manipulate them. This feature gives the performer the freedom to work with familiar sound material and create following their own artistic sensibility.

- Performers can spatialize sounds with ease as the system incorporates *Ambisonics*, where surround sound can be realized independent from the number or orientation of multi-channel loudspeaker systems (Rabenstein and Spors 1104). Performers can create speaker mappings of their unique speaker setup and spatialize accordingly. The system also offers a headphone option to listen to the sound output in binaural, which is a feasible and close alternative for those who wish to explore surround sound but have no access to multi-channel loudspeaker systems (Nishimura and Sonoda 4).
- This thesis contains a portfolio of creative works called *Sonic Matters*, which will be discussed in Chapter 4. All five pieces were presented by the author in the Sonic Arts Lab at the University of Calgary on June 14^{th}, 2018.

Chapter 2: Background

We have come a long way since the beginning of the 20th century: we no longer require 30 railroad flatcars to move our electronic instruments as Cahill did when moving his instrument from Holyoke, Massachusetts to New York City (Holmes 9). We are carrying smartphones in our pockets with thousands of times more capacity and computational power than the computers which made moon landing possible. The pace of evolution in technology and human anatomy differs a lot. The technology is evolving rapidly every day: for instance, five years could be enough to render a smartphone obsolete. However, the 'design' in the human body stays the same; therefore, one could spot many similarities between the current and the preceding design aspects of anything that is designed for humans.

Assessing what has been done before and building on the existing knowledge would enable us to advance, rather than repeating the same steps. For this reason, this chapter includes a brief history to review and understand the previous approaches to electronic music instrument design and practice, and aims to provide the reader with the relevant background to make the discussion in the later chapters easier to understand.

2.1 History of Sound Recording

The history of electronic music has its roots in the invention of the sound recording and reproduction technologies in the second half of the 19th century. The first successful attempt at recording sound was Édouard-Léon Scott de Martinville's *phonautograph (1857)*, which could record sound on paper, but was unable to reproduce the recorded sound (Collins 44). Thomas Edison (1847-1931) and Emile Berliner (1851-1929) both overcame the limitation of reproducing sound but arrived at two different designs: Edison's *Phonograph (1877)* employed a cylinder whereas Berliner's *Gramophone (1887)* used a disc (Millard 24, 32).

As Marshall McLuhan (1911-1980) pointed out, the medium influences any message it conveys and affects the way it is perceived (1–5). Accordingly, the sound recording medium has changed how listeners and musicians regard the sound. Analogous to the difference between theatre and movie, sound recording has created a distinction between live performance and studio composition. Musicians used the sound recording to document their creative processes, began exploring the benefits of the fixed-media, and eventually started using recording and its technology artistically. Recorded and reproducible sound had a tremendous impact on music, and composers created and performed several musical works with turntables during the first half of the 20th century. During the 1920s and 1930s, composers such as Edgard Varése and John Cage were becoming interested in using the *gramophone* as an instrument, rather than using it as a record player (Holmes 44–45; Manning 15). Several composers recognized the artistic potential of the turntable and experimented with the playback of multiple turntables at various speeds (Holmes 44–45).

In 1928, a major breakthrough occurred when Fritz Pfleumer (1881-1945) invented a new viable medium for recording sound (Hugill 6). His discovery suggested that the electronic audio signals could be stored on paper or celluloid tape that had been coated with iron oxide powder (Holmes 35). This new medium presented several advantages, such as the facility to reuse the tape, the ease of editing, and simultaneous multitrack recording on the same piece of tape (Manning 13). The advantages of this new medium paved the way for electronic music to flourish, and this period constructed the basis for several sound manipulation techniques of electronic music. These techniques are still being used in modern day electronic music pieces, and even though the utilized technology has remarkably evolved, the working principles are somewhat similar.

2.1.1 Some Fundamental Processing Techniques

With the rise of tape recorders after WWII, early electronic music pieces were composed by manipulating the tape with techniques such as changing the playback speed, reversing, splicing and reordering, and reverberating sounds. These basic techniques are still current, and they are a crucial part of the electroacoustic music aesthetic; therefore, they are explained in this section.

Mixing: The operation of combining two or more sounds. In the past, it was achieved by playing two or more tape recorders (or record players) together and re-recording on another tape (or disc). Audio sequencers and DAWs (Digital Audio Workstations) can do this by reading multiple files from the hard drive.

Looping: This technique refers to physically looping the tape by cutting a fragment of tape and joining its two ends together, thus repeating the sound on tape continuously. In modern audio programs, one can move the playback start/end markers to loop a portion of the sound easily.

Tape splicing (editing): This technique originally refers to joining ends of tapes using an adhesive material. It also includes reorganizing the sounds on tape by cutting pieces of it and juxtaposing them. Also, diagonally cutting the tape allows for transitions and crossfades between sounds. Digital computers work in a similar principle except they use digital data; hence they are more forgiving and provide a finer level of control.

Tape Echo and Delay: An echo is a sound or series of sounds reflected from a surface and heard by a listener after some delay. Echoes can be distinguished from the original sound and perceived as a different sound source. With tape recorders, this acoustic phenomenon can be imitated by feeding some signal from the playback head back to the recording head, resulting in a series of echoes. As well, consecutively playing the same sound with two or more different tape recorders can create delay effects. Delay effect in digital computers is accomplished by copying the sound to the memory and releasing it after a certain time has passed.

Reverberation: Reflection also causes this effect, but in this case, the delay between echoes is too short to be perceived as distinct sounds. Reverberation gives valuable information about the size and nature of space. This effect can be achieved by playing back a sound through a loudspeaker in a reverb chamber and recording the resulting sound. Also, this effect can be imitated electromechanically using the characteristics of metal plates and springs. Computer systems use complex mathematical algorithms to achieve reverberation. Another viable method is convolution which shapes sounds using pre-recorded impulses of space to produce reverberation.

Playback speed manipulation: Magnetic tape recorders can reproduce the original sound only if the tape is played back at the same speed as it was recorded. Changing the playback speed alters both the duration and the pitch of the sound on tape. For instance, if a recording is played back at double speed, the pitch would be an octave higher, and the duration would be half as long. However, it is possible to keep the original pitch while changing the playback speed in computers. This method is called time-stretching.

Playback direction: Magnetic tape recorders register sounds linearly on tape, in one direction. In the same manner as changing the playback speed, reversing the tape plays the sound backwards. Computers can read the recorded data in reverse to produce the same effect.

Filtering: This kind of operation is used for attenuating or boosting parts of the spectrum of a sound. There are different types of filters for specific applications, for instance, a high-pass filter attenuates frequencies below the cutoff frequency, whereas a low-pass filter attenuates frequencies above the cutoff frequency. Filtering is used in numerous electronic music compositions.

2.2 The Design Aspect

Aside from recording, some inventors were interested in using electricity to generate sound. In 1897, Thaddeus Cahill (1867-1934) invented the *Telharmonium*, in the quest to build an electric music instrument and transmit live music to remote locations through the telephone network (Holmes 8). The *Telharmonium* comprised many dynamos with varying pitch shafts and gears with corresponding inductors for electrical tone-generation, a touch-sensitive keyboard, and a speaker system (Cahill 45). Even though the sound-generating part of this invention was utterly original, the interface presented to the performer was a time-honoured one. Several early electronic music instruments incorporated keyboards in their design, such as the *Sphärophon (1927)*, the *Dynaphone (1927-28)*, the *Ondes Martenot*

(1928), and the *Trautonium (1930)* (Manning 4). Some of them provided the performer with unique controls for achieving greater musical expression. For example, Hugh Le Caine's (1914-77) *Electronic Sackbut* (1945-48) was a visionary keyboard instrument which provided performer's fingertips with many expressive controls: lateral pressure on a key affected the pitch up to an octave in either direction, allowing vibrato, portamento, and microtonality; vertical pressure affected the loudness (Collins et al. 52). This type of control inspired the touch-sensitive keyboard design in the following years, in fact, some of these control features are still absent in many commercial keyboards today, although some 70 years have passed.

Apart from keyboard instruments, some instruments introduced ingenious ways of both sound generation and control. For instance, Lev Sergeyevich Termen's (1896-1993) *Thérémin* (1924) employed two capacitor-based detectors in the form of a vertical rod and a horizontal loop translated the proximity of the performer's hand position to the antennae into corresponding pitches and amplitudes (Manning 5). The *Thérémin* provided great musical expression; however, it was a challenging instrument to play well, since there was no haptic feedback, and likewise, no physical guides for pitch and amplitude (Collins et al. 37). Regardless, it proved that the performer's gestures in space could be the source of musical sound (Dean 11–12).

No matter how much time has passed since then, problems that Cahill or Termen encountered are the same underlying technical problems that modern designers face today. Technology is continuously developing: the components are becoming smaller, faster and cheaper, but some aspects, such as the interface, the ergonomics, and the sound-shaping controls remain as concerns of a designer.

2.3 The Digital Epoch

Developments in electronics have always had a significant impact on how electronic music has advanced: the vacuum tube rendered electronic tone generation possible and transistors made the electronic instruments more compact. Another big leap in electronic music is taken as the first computers were produced and subsequently the microprocessors began to replace the transistors.

In 1957, Max Mathews (1926-2011) created the first computer synthesis programming language, *MUSIC I* (Roads and Strawn, "Introduction" 87). A series of *MUSIC* software programs followed, often referred to as *MUSICn*, with more capabilities, and which worked on newer and faster computers (Manning 70). Later, he designed the *GROOVE system* (1970) which presented a unique means of control: moving a 'magic wand' in three-dimensional space would send differentiating voltages to the voltage-controlled synthesizer; moreover, the computer could operate the motors in the wand to produce programmable haptic feedback (Park 13). Regarding the system's success in musical expression, the programmable feedback mechanism has proven itself as a brilliant design idea in establishing a meaningful interaction between the system and the performer.

Canadian composer Barry Truax's (b. 1947) *PODn systems (1972-96)* used minicomputers and processing units together and were built for real-time synthesis and interactive composition (Truax, *POD*). They were capable of processing the sounds in various ways, and moving sound in space, such as rotating the sound elements in both directions (Truax, "The POD System" 30–34). *PODn systems* allowed the performer to focus on the macro elements of the composition while the computer was adjusting the parameters to shape a complex sonic texture, thus demonstrating a great balance between human interaction and automation.

2.3.1 Digital Sampling

Computer systems require the electrical signal to be translated into digital data. The method used to represent a sound in a digital system (analog to digital), or to reproduce a digital sound through the loudspeakers (digital to analog) is called *sampling*, and the quality of the conversion depends on the *sampling rate* and *bit depth* of a system. The *sampling rate* determines how many times in a second a signal will be sampled, and the *bit depth* determines how many steps there are to represent the amplitude of each sample (Roads and Strawn, "Introduction" 26). According to the *sampling theorem*, the sample rate should be at least twice the highest frequency to be sampled for the faithful representation of audio ("Digital Audio Concepts" 30). The human hearing range is often referred to be between 20Hz and 20,000Hz (Rossing 747). To fully cover this range in the digital domain, a sampling rate of at least 40,000 Hz is required. Additionally, a low-pass filter is necessary to prevent a sampling error called *aliasing*, which would introduce distortion to the sampled audio (Roads and Strawn, "Digital Audio" 27–28). As such a filter requires a transition band, the sampling rate must be greater than 40,000Hz; therefore, the two standard sampling rates are set to 44,100Hz and 48,000Hz by *The Audio Engineering Society* in 1985 ("Digital Audio" 11).

2.3.2 MIDI Protocol

By the early-1980s, there was a large number of commercial sound synthesizers and peripheral music and sound control devices built by various makers. In 1983, prominent electronic instrument manufacturers agreed on establishing a digital protocol called *Musical Instruments Digital Interface (MIDI)* for these devices to intercommunicate (Manning 266–67). This protocol allows a device to send and receive logical data streams over sixteen separate channels through a single cable (Roads 983). The MIDI data consists of only control messages, and the device at the receiving end interprets these messages to produce sound. It also encodes the human performance and breaks down the performance aspects for the representation in the form of MIDI data (Manning 270–71). MIDI messages are composed of *status* bytes and *data* bytes: a status byte specifies the function and channel, and a data byte states the performance values (Roads 990). A data byte contains eight bits (XXXXXXXX), and the first bit denotes whether it is a status (1XXXXXX) or data (0XXXXXX) byte; the remaining 7 bits, therefore 128 (2^7) values, can be used for the representation of almost all performance parameters (The MIDI Association). *Channel Voice* messages target a specific channel, and contains information regarding the performance:

Note On message identifies the beginning of a note, and includes the *pitch* and its *velocity* information: *pitch* indicates which key is pressed, and *velocity* indicates how hard the key is hit.

Note Off message is a *Note On* message with *velocity* of zero, therefore stating the end of the note.

Pitch Bend message bends a pitch up or down, depending on how far the *Pitch Bend Wheel* is turned. Unlike most of the MIDI messages, this is a 14-bit message, which allows up to 16,384 (2^{14}) pitch divisions in total: 8192 values above and below a default centre pitch.

Control Change (CC) message informs the receiving device when the position of a particular wheel, knob, pedal, or another control is changed. Excluding the status byte, CC messages are made up of two bytes, defining the controller number and its value respectively. A few CC numbers, such as *channel volume*(CC#7), *pan*(CC#10), and *sustain pedal* (CC#64) are pre-set, and the rest are freely assignable. Universal MIDI controllers send various performance information through these unassigned CC numbers.

Aftertouch message is sent when there is a change in pressure on an individual key after the key is pressed.

2.4 Emergence of New Musical Languages

MUSICn programs were built around a unit generator concept, in which the software modules (unit generators) take sound and control input, and output the sound after processing or modifying (Roads 787). A descendant of *MUSICn,* a text-based sound and music computing system *Csound (1985)*, is still being used by many musicians worldwide (Csound, *Introduction*). *Csound* supports separate *audio rate (sampling rate)* and *control rate* to achieve computational efficiency (Csound, *Real-Time Audio*). The audio must be computed exceedingly fast (usually more than 44,100 times per second) for accurate reproduction of the sound, whereas *control rate* could be 20 times slower, and still makes no perceptual difference in many synthesis tasks (Wang 62). Another current descendant of *MUSICn*, Miller Puckette's (b.1959) *Max (originally The Patcher) (1988)* also embodied the unit generator notion, facilitated real-time control over the modules, and provided a graphical user interface (Puckette, "Combining" 68–70). With *Max*, one could connect individual *objects* using *patch cables*, and these *objects* can receive messages from computer input devices, MIDI instruments and controllers, or other *Max* objects (Winkler 49–50). In its first version, *Max* could only visualize the routing, and generate real-time control signals for external systems (Wang 63). In 1996, Puckette designed an open-source program called *Pure Data* to address *Max*'s shortcomings in sound generation and audio processing (Puckette, "Pure Data" 37). A year later, *Pure Data's* audio objects were adapted to *Max* with the name *MSP (MSP both stands for Max Signal Processing and Miller S. Puckette)*, thus renaming the program as *Max/MSP* (Puckette, "Max" 35). In 2002, *Jitter* was added to *Max* to provide the means of generating and manipulating images (Manning 441). Modern-day *Max* software supports thousands of various *objects* including signal generators, filters, mathematical operators, and interface elements (Wang 64). It also provides several widgets, namely, sliders, knobs, buttons, keyboards, and meters, making the control and GUI design easier.

2.5 Musical Input Devices

The means of controlling electricity are apparent in everything from doorbells to thermostats, game controllers to lights. Unsurprisingly, since the arrival of the *Telharmonium* to this day, numerous electronic instruments have naturally accommodated these components in their interface design. While *Ondes Martenot*'s electronic strip provided musical expression, patching cables on a *Moog Synthesizer* connected individual modules. Whether these controls addressed technical or musical concerns, they have been integrated into electronic music since its inception: one could easily see switches, buttons, pads, faders, and knobs on synthesizers, electronic instruments and controllers.

The development of electronic instruments has sparked numerous novel design ideas, which have provided great flexibility regarding control. As opposed to acoustic instruments, electronic instruments use electricity instead of physical energy for sound generation. Accordingly, the controller that mediates the performer and the computer is required to reconcile the performer's physical movement and the sound. Some electronic instrument designers have taken a traditional approach to address this matter and preferred instrument-like devices: almost all traditional instruments have been redevised as electronic instruments (Chadabe 222). However, some others invented unique gestural controls fitting to their instruments, for instance, the *Thérémin* used electronic sensors for directly translating the performer's continuous motion into seamless glissandi. A similar concept was embodied in Mathews' *Radio Baton (1987)*, which used two antennae to track the positions of two batons in three-dimensional space for sending triggers and performance information to computers or other devices via MIDI (Mathews 2). This method of control has been employed even in several gaming motion sensors, such as the *Leap Motion (2013)* and *Microsoft Kinect (2010)*, that track one's hands and body respectively (Leap Motion; Microsoft Corporation 7–8). Buchla's programmable MIDI controller, *Thunder (1990)*, comprised pressure-sensitive tactile elements that are arranged in the shape of hands (Miranda 33). Michel Waisvisz (1949-2008) was regarded as a virtuoso of his self-designed

Hands (1984), which was controlled by metal plates fitted with sensors, switches, and potentiometers comfortably resting on the hands of the performer (Torre et al. 24; Jordà 100–01).

Since the sensor input is vital in designing an electronic music instrument, the ideal programming environment should provide the support for different kinds of sensors. Music programming software, such as *Max* allows for the organization and manipulation of all kinds of computer input devices, including microphones, computer mouse and keyboard, game controllers and joysticks, MIDI devices, microprocessors, touch surface tablets, and electronic sensors (Cycling '74, *Max 7*; Cycling '74, *Mira*; Maxuino). Any combination of these devices could form the desired control interface to control performance parameters in real-time. In addition, electronic music instrument designers favour *Max* due to its versatility: its applications include processing audio, controlling MIDI data, receiving sensor input, synthesizing sound, computer-assisted and algorithmic composition, installation art, video manipulation, and computer generated visuals.

There are a few important platforms and conferences for new electronic instrument research such as *NIME*, *ICMC*, and *STEIM* (NIME; ICMA; STEIM). One finds numerous different perspectives and more recent examples of interactive music systems on these platforms. For example, the *Crackle Box* is a synthesizer that makes use of the conductivity of human skin and uses the body as a part of the circuit: the electronics react to fingers on different parts of the electronic surface, and translate touches into synthesis parameters (STEIM). The *Sound Scratcher* lets the performer play selected portions of various audio samples by aiming two handles at each other and changing the distance between them

(STEIM). The *Head Banger* utilizes headphones to convert the performer's head movement into musical control: the speed of the movement determines the tempo of the music and tilting the head to the front applies a low-pass filter to the music (STEIM). The *Finger Web* is an electronic instrument that uses tension sensors and elastic strings in the shape of a web (STEIM). The performer uses the left hand to pinch the center of the web for sound generation, and uses the right hand in the same manner to alter the sound. The *T-Stick* is an adapted PVC pipe that reacts to touch, acceleration and pressure; as well, it has a piezoelectric crystal to pick up sounds created by the performer on the instrument (Malloch and Wanderley 2–3). *The Reactable* is a tangible tabletop controller which allows for the shared control of the instrument as a way to create collaborative music performance. On a visual surface, the users can place, twist, and move pucks to generate sound, and control the interaction between the other pucks depending on their positions and proximity (Jordá 2989–92). The examples above demonstrate that there are numerous electronic sensors and methods to receive input from the performers. However, even though some of the examples utilize the same input method, the musical results are quite different depending on the translation of the sensor input into the musical output.

2.5.1 HCI Perspective in Design

Designing a system for musicians to create music comes with a lot of aspects to think about, such as ease of use, expressivity, capability, flexibility, and engagement. Without question, the number of factors taken into account goes beyond this; however, some elements outweigh others depending on the perspective of the designer. The musical interface constitutes the performance environment and transforms a complex technological system into a musical system that reconciles one's performing style with the artistic approach (Vidolin 445). Accordingly, the designer should establish the channels of communication between the performer and the system. This communication is what makes skilled musicians become one with their instruments (Wallis et al. 30), and allows them to perform their music freely. The biomechanical energy exerted by the performer makes the sound in acoustic musical instruments, whereas in the electronic systems it is the electricity that makes the sound (Leman 163). This disconnect is a substantial deficiency, and needs to be addressed in such systems; thus, some meaningful sensory feedback can help in closing the communication gap.

Musical instrument players need long-term practice of motor actions in order to automate the motor patterns so that they can concentrate on higher level artistic expression and not just the underlying gestures or movements (Leman 95). The information transferred from musical cognition to muscle memory plays a vital role in performance practice, especially in improvisation. Distinctively, interactive music systems may not be as direct, and a certain action may not necessarily yield the same musical result every time. Thus, performing with such systems cannot be learned by intuition, since they are extrinsically connected to the music (Vidolin 445). The musicians that perform with such systems still benefit from muscle memory to some extent; nevertheless, they need to be aware of the inner workings of the given system. In case this seems like a restriction, the designer can implement automation or presets for some performance aspects to set the performer free from remembering all the technical aspects and make more room for artistic expression. Automation, per se, can augment the affordances of the performance as it provides the performer with the means of studio composition such as shaping complex sonic textures but in real-time. On the other hand, knowing the working principles of a system may afford some new musical possibilities since the musicians can predict the results of their actions and aim for new possibilities.

The designer might want to include numerous sound processing and effect units in a system to expand the palette of the performer; however, there are definite practical limitations in real-time systems: it eventually comes down to what the hardware can do. For instance, if the CPU gets overwhelmed by a particular process, the designer should refrain from using it in the system unless they find a more CPU-efficient implementation. Akin to the CPU, the performer can get overwhelmed by the number of controls, and the designer must be aware of this fact throughout the design process. Constraining the performance controls could help the performer become familiar with the performance environment.

Chapter 3: *Sonic Matter*

3.1 Introduction

This chapter discusses the design aspect of *Sonic Matter* by elaborating on the considerations made and approaches taken, how the system works, along with a detailed explanation of the modules.

The System

Sonic Matter is an interactive music performance system that streamlines the creation of music, making it especially useful for electroacoustic music composers. It gives the performer control over sonic qualities; likewise, it can transform sound material into simple or complex textures. Performers can position the sound source at any point in 3D space, record trajectories, and convincingly move the sound in the listening environment.

The system consists of a computer, the *Sonic Matter* software application realized in *Max*, a *KORG nanoKONTROL2* MIDI controller (KORG Inc.), a *Leap Motion* sensor (Leap Motion), an 8-output audio interface, and optionally, a MIDI sustain pedal. This will be discussed in depth in Chapter 3.3. The typical setup of *Sonic Matter* is shown in Figure 3.1. I used the same setup for my performance at the *Music in New Technologies (MINT)* conference: I moved my left hand over *Leap Motion* to spatialize, *nanoKONTROL2* to manipulate sound, and my right foot to trigger samples (see Soydan, *MINT 2018*).

The computer software has two main components:

1. **The Framework**

The framework is the environment where users can connect and organize the chosen modules, and customize the graphical user interface for their convenience.

2. **The Modules**

Modules are the fundamental part to manipulate the sounds. The modules have a number of features including sample triggering, sound transformation, and spatialization. The system allows the user to employ various modules and multiple instances of the same module. Modules can be added one after another in any combination to give total control to the artist and provide the maximum flexibility in sonic creation. Working principles of each module will be explained in Chapter 3.6.

KORG nanoKONTROL2 controls any assigned parameter of the modules, and *Leap Motion* controls the spatialization (see Chapter 3.3.1). As an option, a MIDI sustain pedal can be connected to the input of an audio interface and used to trigger samples by foot.

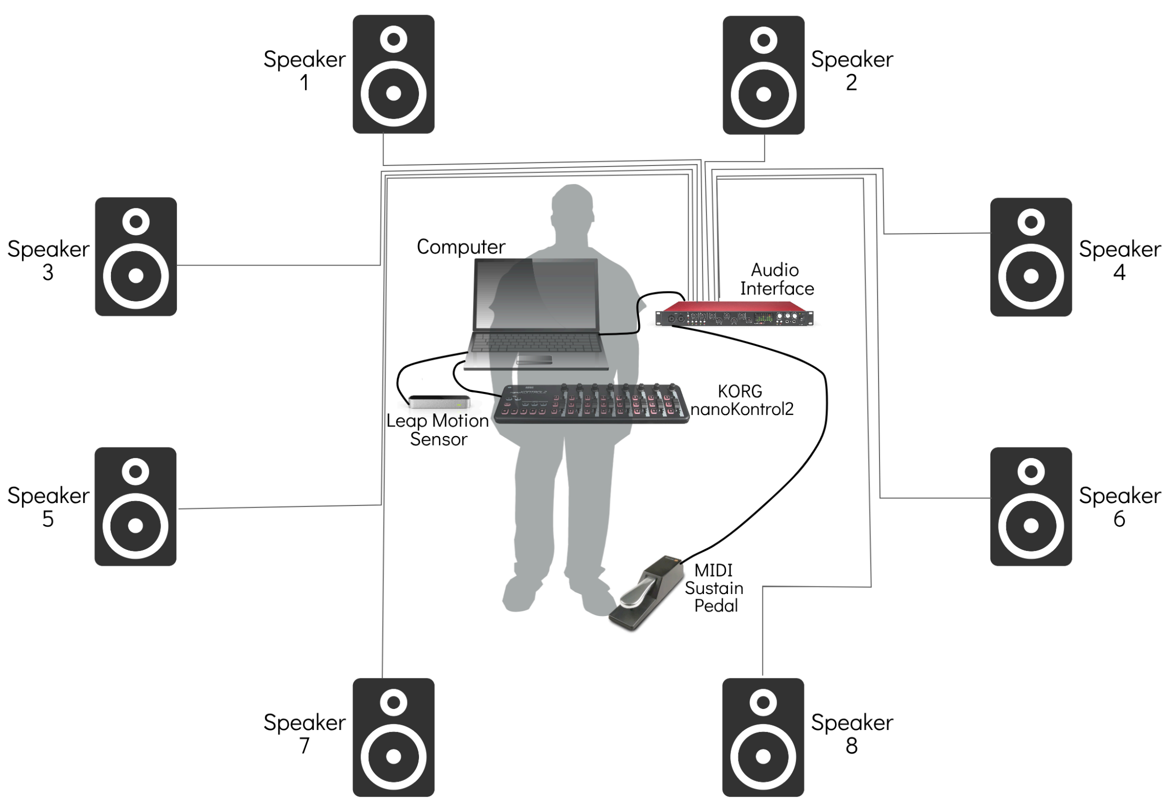

Figure 3.1: The typical setup of *Sonic Matter*

3.2 Sonic Matter and HCI

Whatever one wants to achieve with *Sonic Matter*, the result will be realized by performing with it: in consequence, the performative aspect should be exceptionally fulfilling. Also, the performer should attain the required technical competence to make more room for interactive music-making. Having studied and experienced the importance of each design aspect mentioned in this thesis, I aimed to address the concerns and find a right balance between the limitations and aesthetic choices in the designed system.

In *Sonic Matter*, my interaction with the system substantially depends on the aural feedback coming through the loudspeakers. This feedback has a significant impact on musical decision-making throughout the performance. Additionally, the kinesthetic feedback of the interface enables me to communicate with my fingers, thus making the system easily accessible. The control gestures are extremely straightforward; turning a knob and sliding a fader deliver direct musical results, thereby giving immediate control over the sound. Further, any action on the MIDI controller can be comprehensible and coherent with the sonic result since the user maps the performance controls to the buttons, knobs and faders of a given controller. The spatialization, though, is the sole control that cannot be easily programmed by the user, even so, it is intuitive: while hovering over the *Leap Motion*, the performer's hand that moves in a certain direction can move the sound respectively in the listening space.

Although ease of use and user-friendliness are vital in system design, there are more factors necessary to make a system engaging for its performer. An interactive system without an intrinsic rewarding mechanism might make the performers lose interest as the improvement of skills would have no apparent effect on the musical result (Leman 169). Also, the intrinsic motives of mastery, autonomy, and purpose need to be precisely designed into HCI to form a better long-term engagement (Wallis et al. 64). These motives played a significant role in making *Sonic Matter* desirable for myself; therefore, I incorporated them in the design. After performing with *Sonic Matter* for some time, I came to the realization that the system presents a wide range of mastery levels. In addition, its learning curve is very gradual: it is quite easy for a beginner to learn, and anyone who spends a certain time practicing can clearly discern the improvement in the musical outcome. As well, *Sonic Matter* has autonomy to some extent: controlled randomness is integrated into a few modules for the determination of some musical decisions. However, *Sonic Matter* can still be classified as an instrument paradigm system, in which the contribution of the system is used for augmenting human performance (Rowe 302). Finally, an intimate and immediate connection is attained as the performers of the system can work on their own sound samples. This feature provides the means of exploring the sound material's concealed potential, which also ensures a long-term engagement with the system.

3.3 Design Considerations

From the very beginning of the design process, I intended the system to serve artists as a catalyst in the creative process and spark their musical curiosity with the sonic possibilities it offers. Hence, I adopted a practice-based research approach in the design. As well, the human-centered design approach influenced the process: I made the system practical by focusing on the performer and considering human factors in the design. My creative artistic practice laid the foundation for the system regarding numerous features the modules employ; however, the system does not necessarily limit the musician to a specific musical direction. I was particularly concerned about creating an open environment for performers, and I am happy with the result I achieved.

Some significant decisions made throughout the design process along with their rationale are explained and detailed below.

The Hardware

KORG nanoKONTROL2

The *KORG nanoKONTROL2* has eight assignable channel strips, each having a knob, a fader, and three buttons with LEDs. On the left side are eleven buttons with standard sequencer controls such as play, stop and record (Figure 3.2). It comes with a *KORG Kontrol Editor* software by which users can control the behaviour, range, and control change (CC) number of each button and fader (Figure 3.3). Due to its technical affordances, compatibility, inexpensiveness, and small-size, I used it as my primary control interface.

Figure 3.2: The main control interface: *KORG nanoKONTROL2* (KORG Inc.)

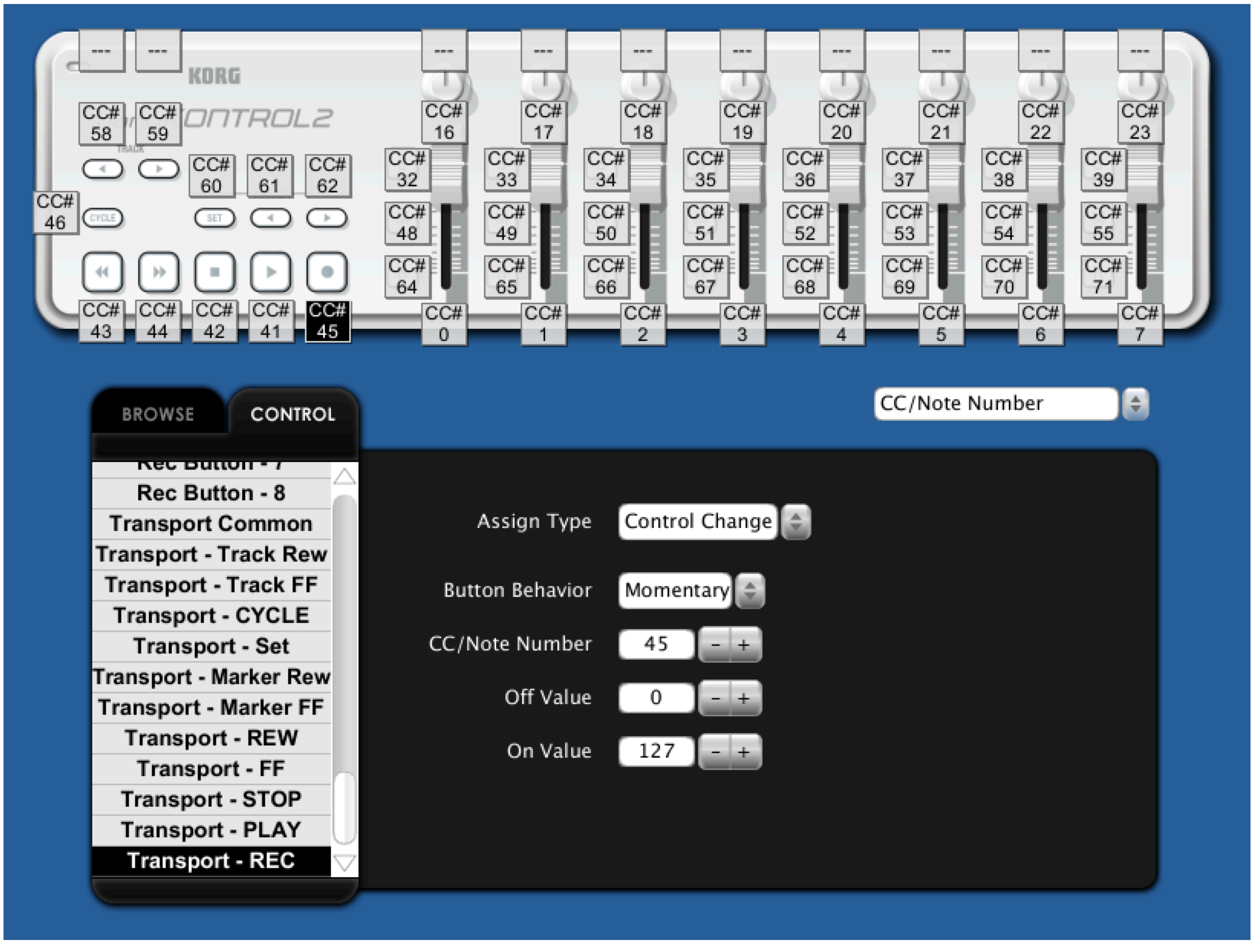

Figure 3.3: The interface of *KORG Kontrol Editor* software

Admittedly, the MIDI protocol is insufficient to accommodate total musical expression because of its low-resolution control messages (see Chapter 2.3.2). Nevertheless, the MIDI protocol offers some significant advantages outweighing its shortcomings, which are:

- It is very well-established: MIDI devices are compatible with all computer platforms, and almost all professional music software supports this protocol. Also, MIDI messages are quite easy to understand for people, and the devices are very flexible to be programmed.

- There are numerous types of MIDI interfaces and controllers on the market with varying prices and quality. Although they are diverse in design, make and model, they use the same messaging standard, which gives the user the freedom to choose their equipment depending on their preference, rather than limiting them to a specific piece of equipment.
- A well-designed MIDI controller can offer a highly intimate communication via the fingers. This feature is lacking in touch screens or photosensors since this kind of kinesthetic feedback is difficult to simulate (Manning 465).

To address the issue of low resolution, I used an algorithm to interpolate between the steps, and I mapped the control values to the performance parameters according to their musical importance (see Chapter 3.3.1).

Leap Motion

Leap Motion is a robust optical 3D sensor developed for tracking hand and finger position with precision (Weichert et al. 6383). When placed on a surface facing upward, the sensor observes and tracks a hemispherical area above it (Figure 3.4). It has numerous uses such as navigating an operating system, interaction with 3D models, playing games, and controlling several AR/VR applications (Terdiman).

I was looking for an intuitive way to position and move the sound sources in space. Considering the accuracy, adaptability, and the convenient control it offers, *Leap Motion* was an excellent choice to serve the purpose. By scaling the sensor's tracking to the speaker map, *Leap Motion* provides a seamless translation of hand movements to the spatialization.

Figure 3.4: An illustration of how *Leap Motion* works (Leap Motion)

MIDI Sustain Pedal

The MIDI sustain pedal is a simple foot-operated switch that sends an on/off control message to produce a sustain effect when pressed. In *Sonic Matter,* the sustain pedal was used to free up one of the performer's hands, which previously had been tied to triggering samples throughout the performance. With the addition of the pedal, performers can utilize their feet for this purpose, which allows them more flexibility to access essential performance controls to achieve greater musical expression. The original setup used hands for its operation, and only included the *KORG nanoKONTROL2* and *Leap Motion*. However, in practice, there were too many aspects to control and shape while triggering the samples. This issue revealed that the system was impractical to perform with; therefore, I added the MIDI sustain pedal to the setup to free up one of the hands.

The Software

The Modules

This part of the system was the core aspect to develop my musical and artistic ideas in my creative work. My primary aim was to provide means to explore interesting sonic material and to create a captivating sonic environment. I started off with a small collection of modules and subsequently added more as I conceived of more possibilities. Although the number of existing modules is not plentiful, they are equipped with exceptional transformational capabilities. With the designed modules, I can focus on the sound source, the sound transformation or the effects.

Ambisonics

Space and spatialization are two integral aspects of acousmatic music practice (Batchelor 152). As for surround sound, it provides an immediate immersion and opens a new spatial dimension for sounds to cover, whereas stereo sound limits the ideal listening position, or sweet spot, to a focused area (Moore 187). Ambisonics is integrated into *Sonic Matter* to allow its users to freely explore the capabilities that surround sound offers.

There are three main reasons for the integration of Ambisonics into *Sonic Matter*:

1. There is no single sweet spot; the surround sound image is relatively homogeneous across the concert space. The listener experience is considerably less affected by their listening position.
2. Ambisonics is flexible to work with any multi-channel speaker setup to furnish users with surround sound. It provides full surround sound with height if the speaker setup permits.
3. There is an available external library called *ICST Ambisonics Tools* for *Max* developed by the *Institute for Computer Music and Sound Technology* in Zürich, which offers a good range of control regarding spatialization, and is a result of continuing research and practice since 2000 (ICST).

3.3.1 Control Mappings

Mapping is a vital part of every interactive system as it translates the performer's input into artistic expression and the available responses from the system. A transparent layer of communication between the system and its user is unattainable unless a proper strategy is employed. Roven et al. classify mapping strategies for interactive systems in three main categories:

- *One-to-one Mapping*: Each input controls one musical parameter, usually via a MIDI control message: it is quite simple but not very expressive.
- *Divergent Mapping*: One input controls many musical parameters to achieve a musical result: it is more expressive than *one-to-one mapping*, but still limiting since the deeper level controls are unavailable to the user.

- *Convergent Mapping*: A combination of many inputs produce one musical parameter: although it has a steeper learning curve, it proves to be far more expressive compared to the previous two strategies (69).

Considering all three, one might think that *convergent mapping* strategy should dictate the way a system is designed. In *Sonic Matter*, though, all three strategies found their places. In designing the system, there was more to think about as the input device has its own constraints. There is only a limited number of input controls available and except for 16 of them, they can only send on and off messages. These factors affected how the controls are assigned regarding their musical importance. For instance, buttons are mapped to basic actions such as triggering samples or activating/deactivating modules following *one-to-one mapping* strategy. As well, there are a few *divergent* controls for global reverb time, delay feedback or spatialization. The *convergent* controls are limited by the number of continuous controls on the main interface, but they still provide the means of expression by making good use of the modules at the user's disposal.

Additionally, the main controller is a MIDI device. Therefore, an efficient strategy for the mapping was necessary to increase the 'musical resolution' of the inputs. First, I applied a slight level of smoothing using a *line* object to obtain the decimal values between each integer step. Then, I divided the input values depending on their impact on musical expression. The table in Appendix A covers all the mapped controls as well as their values.

3.4 Visual Feedback

In the earliest stages of its development, *Sonic Matter* was a difficult system to perform with. I had several clumsy performances in which I had tried turning off a module while it was already inactive. I finally came to the realization that a visual feedback regarding the status of the modules is crucial to put together a performance in which I am fully aware of what is going on in the background. After some research, I discovered that the LEDs on my MIDI controller can be programmed through *Max*, using the *ctlout* object directed at a certain CC with the on (127) and off (0) messages. I coded a few types of visual feedback with different meanings:

- **Toggle**: The LED is either on or off to indicate that a specific control is active or dormant. An example for this type is the time-shifting option of *SM_Groove* (CC#33).
- **Flash and settle**: This mode is utilized for commands that are executed over time such as fading in and out. The LED flashes until the action is completed and stays on or off depending on the status of the performance control. Fade in/out control of *SM_Polybuffer* (CC#53) is an example of this type of visual feedback.
- **Trigger**: The LED periodically flashes to display whenever the playback is activated. The blinking period is responsive to the changes in triggering frequency to inform the performer. *SM_Gestures*' on/off control (CC#46) works in this principle.
- **Flash**: The LED flashes at a fixed interval to denote the control is currently active. It is implemented in *SM_AmbiPanning*'s record trajectory control (CC#45).
- **Direction**: This mode uses two LEDs to show if the controlled parameter is either lower/higher or at a certain value. *SM_Gestures* employs this type of feedback for its playback speed control (CC#61-62). The playback speed is lower than original recording when the left LED (CC#61) is lit; it is higher when the right LED (CC#62) is lit. It is at the original speed if none of the buttons are active.

Apart from above types, all buttons with momentary behaviour stay lit while the performer holds them down to confirm that the button is pressed. Also, the LEDs in the first channel strip display if the output level of *SM_Groove* exceeds some thresholds: the bottom at -25 dBFS, the middle at -15 dBFS, and the top at -5dBFS. Finally, I created a GUI for myself to see the pertinent details of some modules (Figure 3.5). This GUI was projected on a screen for the audience to see and understand what I was controlling in real-time.

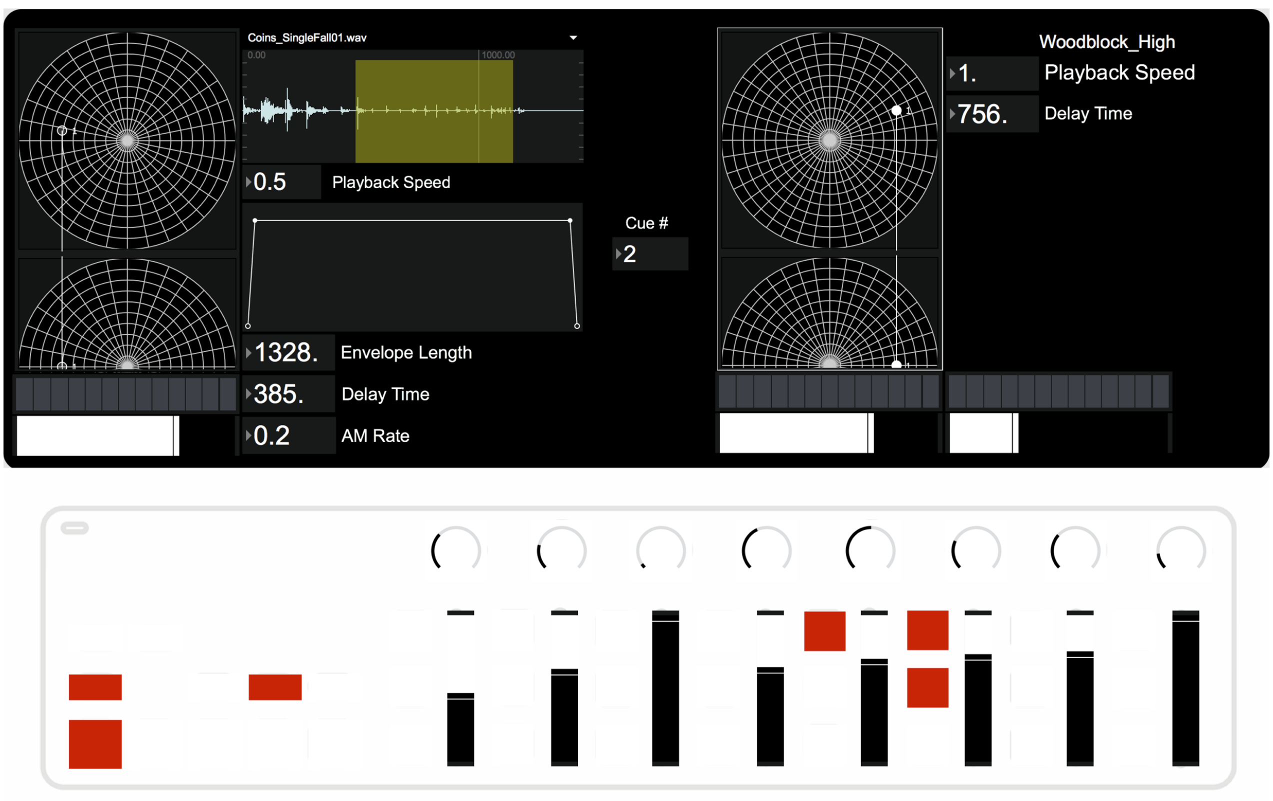

Figure 3.5: The GUI of *Sonic Matter*

3.5 Analyzing the System

Birnbaum et al. suggest that the seven aspects shown in Figure 3.6 are essential to visually analyze electronic musical instruments, installations, and other musical devices (193–94). What these axes indicate are explained below.

The ***Role of Sound*** axis denotes what the sound is used for,

The ***Required Expertise*** axis represents the level of practice necessary to interact with the system as intended,

The ***Musical Control*** axis specifies the level of control that the user has over the musical result,

The ***Feedback Modalities*** axis shows the degree of feedback provided to the user,

The ***Degrees of Freedom*** axis denotes the number of input controls available for the user,

The ***Inter-actors*** axis indicates the number of people interacting with the system,

The ***Distribution in Space*** axis represents the total physical area of the interaction.

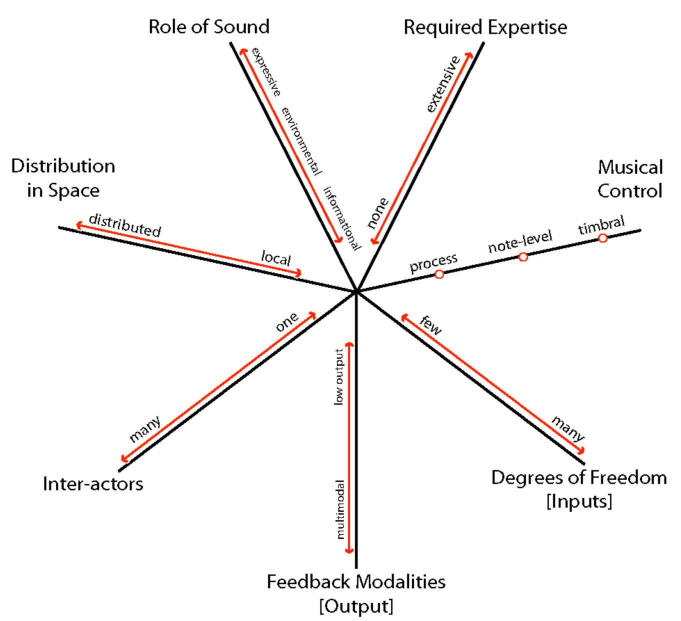

Figure 3.6: The seven aspects used to analyze electronic instruments (Birnbaum et al. 194)

Using the analysis method proposed in Birnbaum and others' research, I compared two electronic instruments that are mentioned in this thesis with *Sonic Matter*: the *Thérémin* (Figure 3.7) and *The Hands* (Figure 3.8). Reading these graphs, even though they all focus on the right side of the plot, the degree of each aspect can be quite different. A significant difference is seen in the *Distribution in Space* axis since *Sonic Matter* expands the interaction space by employing surround sound. In addition, *Sonic Matter* requires less practice than the others, provides aural, visual, and kinesthetic feedback in real-time, and allows for timbral level control.

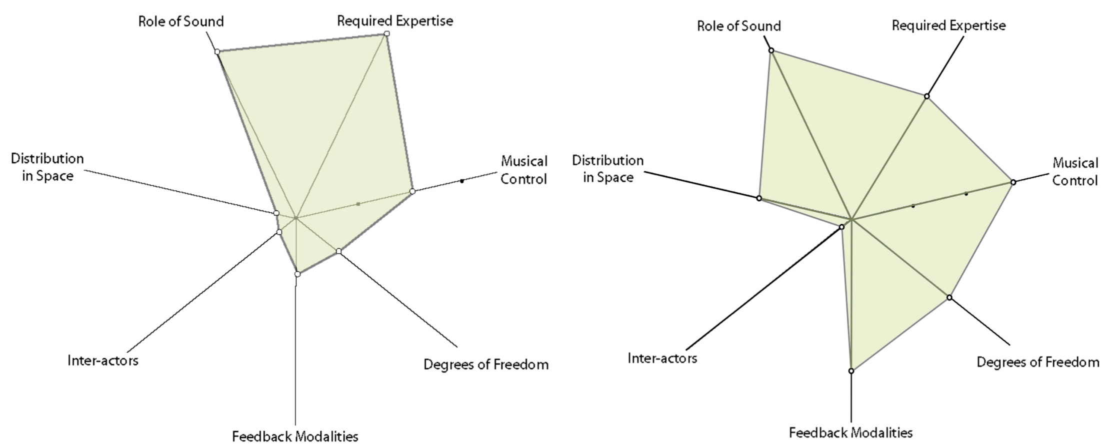

Figure 3.7: The comparison between the *Thérémin* (left) and *Sonic Matter* (right)

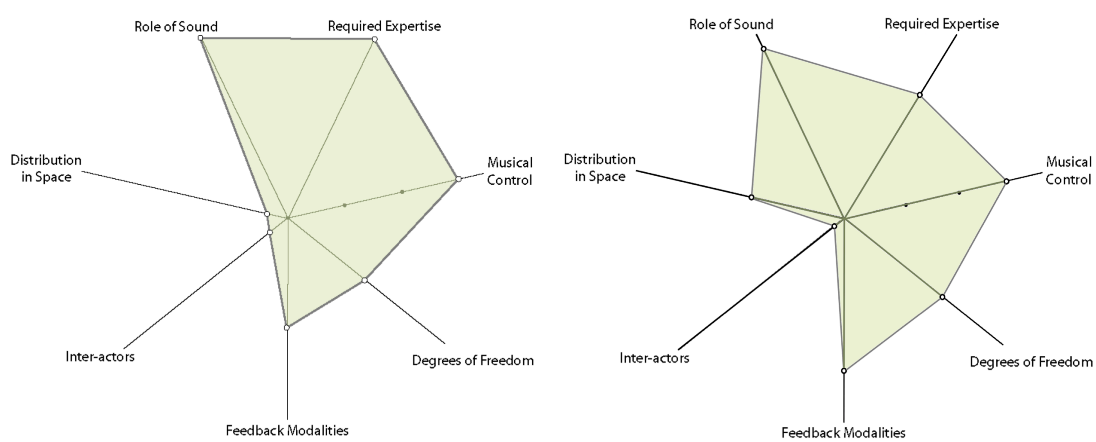

Figure 3.8: The comparison between *The Hands* (left) and *Sonic Matter* (right)

3.6 Sonic Matter Modules

3.6.1 SM_AM

SM_AM applies *amplitude modulation (AM)* to the input signal, which is the process of modulating the amplitude of a carrier signal (f_c) with a modulator signal (f_m) (Figure 3.9). AM is used for timbral synthesis as it produces two sidebands around the carrier: the sum and the difference of the carrier and modulator frequencies. This technique is also employed to create a periodic pulsation effect when the modulator frequency is below 20 Hz.

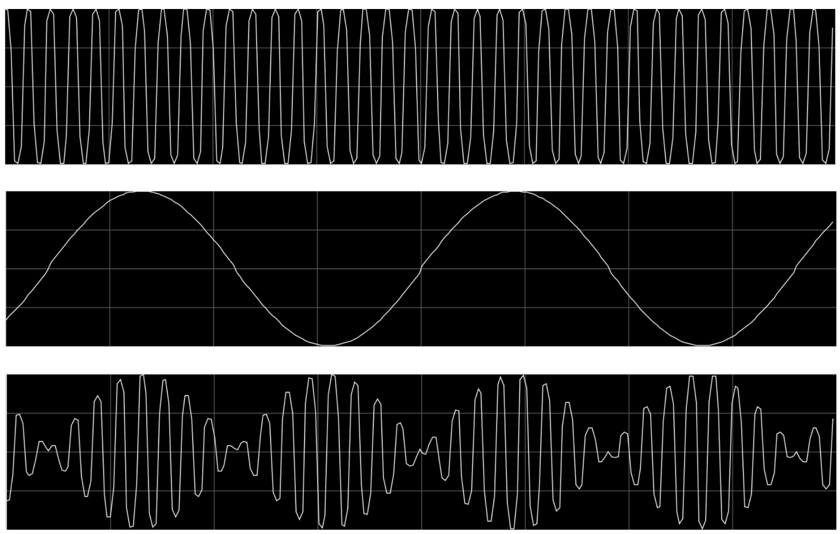

Figure 3.9: The effect of Amplitude Modulation. The carrier signal (top) is fluctuated in proportion to the modulator signal (middle) to create periodic pulsation (bottom).

The modulator frequency can be set between 0 Hz and 1000 Hz using a knob on the main controller. The performer can utilize this control to create a tremolo effect using values below 8 Hz, a change in timbre above 20 Hz, and perhaps a dissonance when pushed near the high extreme. If the controller is set to zero, the input signal bypasses the module. The depth of the effect is limited to 25 percent to ensure that sound is present with low frequency (between 0 and 1 Hz) settings.

3.6.2 SM_AmbiPanning

SM_AmbiPanning employs the *ICTS Ambisonic Tools* Max external, incorporating a GUI tool for full 3D surround spatialization (Figure 3.10). This module encodes the input signal for *Ambisonic* sound, and decodes it according to the speaker setup set by the user (Figure 3.11). It has three fundamental methods for sound movement: *random* moves the sound in all directions randomly, *rotate* spins the sound horizontally in clockwise motion, and *trajectory* lets the user define a route for the sound. In *trajectory* mode, the user activates the recording mode with a button on the main controller, and the user's hand moved over *Leap Motion* is translated into spatialization information. The sound route is then scaled to fit the speaker setup. The rate for trajectory mode can be modified by a knob on the main controller between half-speed to eight times the recorded speed.

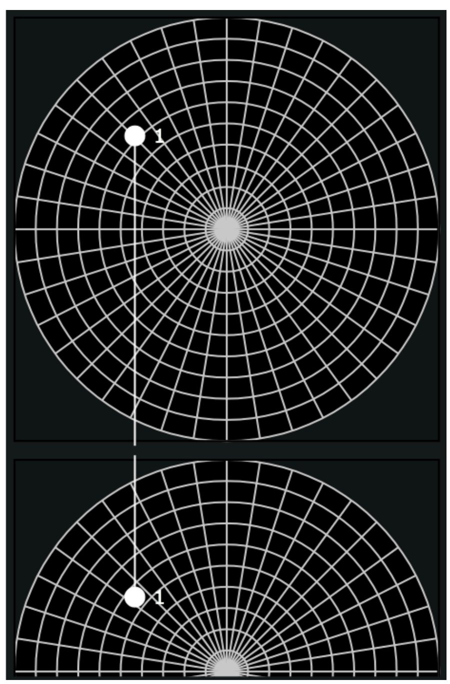

Figure 3.10: Graphical user interface of SM_Ambipanning. The full circle on top represents top-view and semicircle below is the side-view.

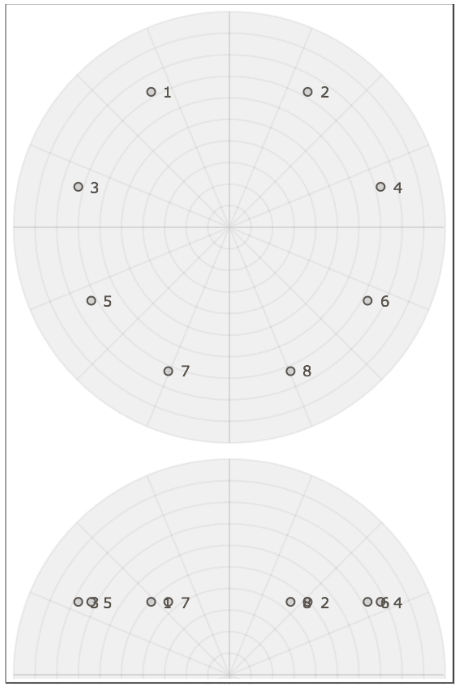

Figure 3.11: 8-channel speaker setup for the University of Calgary's Sonic Arts Lab

3.6.3 SM_Bitcrush

SM_Bitcrush produces a distortion by reducing the sampling rate and bit depth of the input signal (see Chapter 2.3.1). The word size (bit depth) takes integers between 1 and 24, and the sampling rate accepts decimal numbers range from 0 to 1, and multiplies the value with the effective sample rate. Bit depth can be controlled by a knob and sampling rate can be set by a fader on the main controller.

3.6.4 SM_Delay

SM_Delay copies the input signal into the computer memory, and outputs the signal after the specified delay time has passed. A continuously variable delay algorithm is implemented into the module: it transposes the original signal up as the delay time decreases, and transposes it down as the delay time increases. A fader on the main controller adjusts the amount of the output signal fed back to the input, and a knob sets the delay time between 0 and 2000 milliseconds. If the delay time is fixed to zero, the input signal bypasses the module. High feedback values with delay times up to around 50 milliseconds produces a perceivable pitch due to comb filtering.

3.6.5 SM_EnvelopeFunction

SM_EnvelopeFunction applies an envelope to the sounds played back by SM_Groove. A graphical interface visualizes the envelopes and let the user draw or edit them (Figure 3.12). The envelopes are stored in a preset box for the user to conveniently change during a performance.

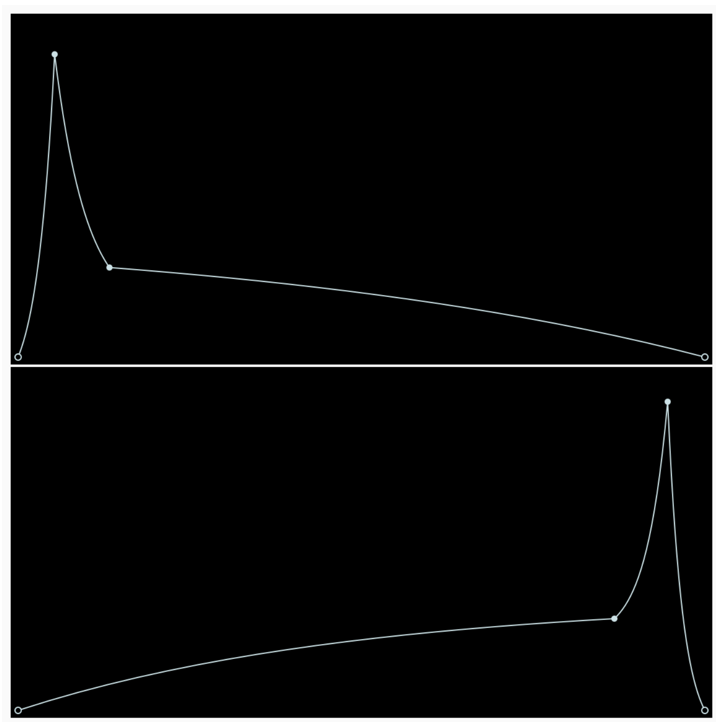

Figure 3.12: Envelopes involving curves drawn in SM_EnvelopeFunction

The user can control the playback speed and the loop length in SM_Groove, thereby changing the total playback duration. There is also a fader for the envelope length, scaling from the duration of one complete loop cycle to 50 milliseconds. The formula in Figure

3.13 is applied to reflect the changes in playback duration to the maximum envelope length. For instance, if the user sets the loop start marker to 500ms and loop end marker to 1500ms, and plays it back twice as fast, the playback duration (therefore the maximum envelope length) will be 500ms, meaning that the user can choose values between 500ms and 50ms for the envelope length.

$$PlaybackDuration = \frac{Loop_{end} - Loop_{start}}{PlaybackSpeed}$$

Figure 3.13: The formula to calculate one complete loop cycle

3.6.6 SM_Gestures

SM_Gestures randomly picks and plays audio samples from a preloaded buffer to create immersive sonic gestures. The module can handle up to 100 instances of sound playback at a time. The user can speed up or slow down the next instance using two buttons on the main controller. The user can also set steps for triggering frequency prior to performance by editing the Gesture.txt in the root folder or from inside Max following the format seen in Figure 3.14. A button on the main controller makes alternating between the steps possible during a performance. There are two triggering options: *one cycle* option triggers 8 instances and stops, whereas *toggle* option keeps triggering until the user hits the button to stop. Each instance is assigned to a different speaker using one of four speaker selection methods:

1. Clockwise: The output number is determined by counting forward for each audio sample triggered. The counter resets to 1 after reaching the last output number. To achieve clockwise and counter-clockwise movements with a speaker setup

comprising four stereo pairs, the outputs of SM_Gestures module need to be connected to audio interface following the arrow in Figure 3.15.

2. Counter-clockwise: Except from counting the output number backwards, this mode works exactly as the clockwise mode.
3. Random: The output number is selected randomly for each instance within a selected range.
4. Preset: In this mode, the user can set delays for each speaker in order to create the gestures. Unlike the three above, this mode uses the delays to create the gestures rather than depending on the triggering frequency. To this end, the module triggers all instances in rapid succession, and the sound going to each output waits until their designated delay time has passed.

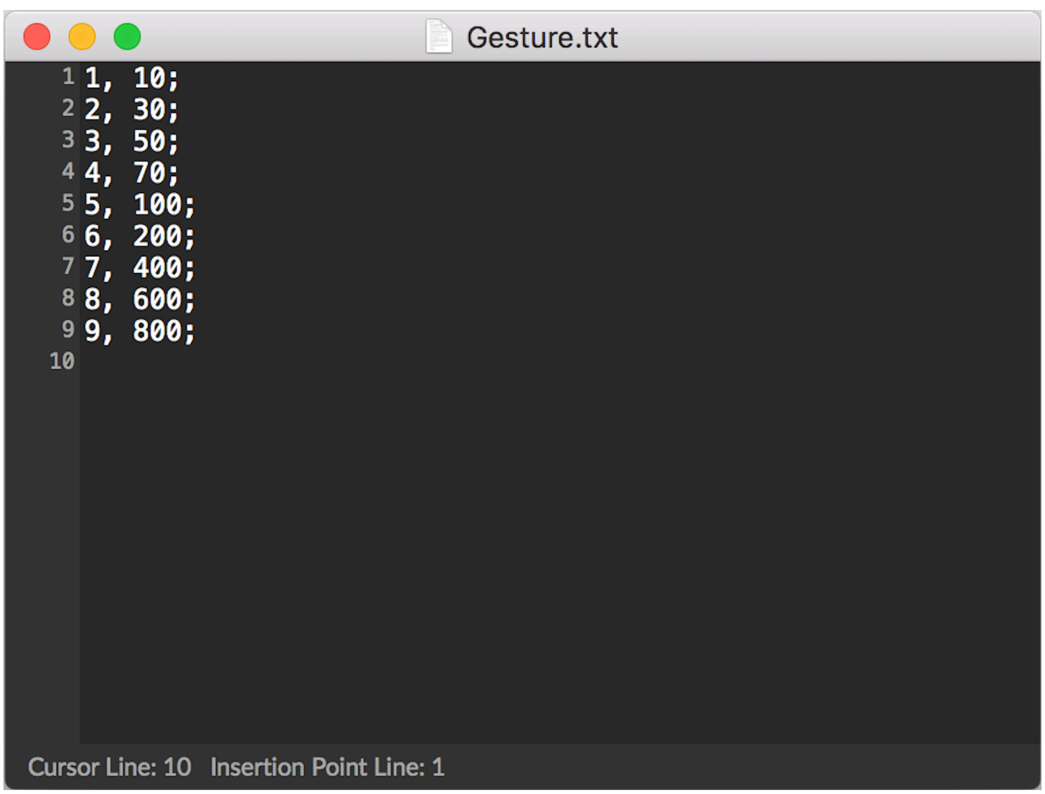

Figure 3.14: The *coll* object designated to store trigger frequency values of SM_Gestures.

Figure 3.15: Speaker configuration for four stereo pairs. Outputs from 1 to 8 should be connected to 1-2-4-6-8-7-5-3 respectively.

3.6.7 SM_Groove

SM_Groove is a sample player and the main sound source of the system as it allows for the manipulation of audio samples in a few different ways. It automatically scans all the audio files in "Audios" folder in the root directory and populates a list with their file names. Whenever the user selects an audio sample, the module first loads it into the playback buffer and then normalizes it in order to minimize level differences between distinct audio samples. The module can handle up to 100 instances of playback at a time. All instances have an envelope applied to them and there is a fader to set the release time of this envelope between 5ms and 3 seconds. The user can select the beginning and the end of the loop with a knob and a fader on the main controller, modifying the duration from 50ms up to the original sample length. If the user is triggering samples while changing either end of the loop, the module plays all the selected portions from their beginning to the end, thus creating a cascading effect. The user can exaggerate the degree of overlapping in this effect by simply turning up the release fader. There is a knob to control the playback speed between 0.1x and 8x the original speed as well as a button to toggle time-stretching on and off. There are also two buttons to navigate between audio samples. The module offers a GUI with useful information about the selected audio sample (Figure 3.16).

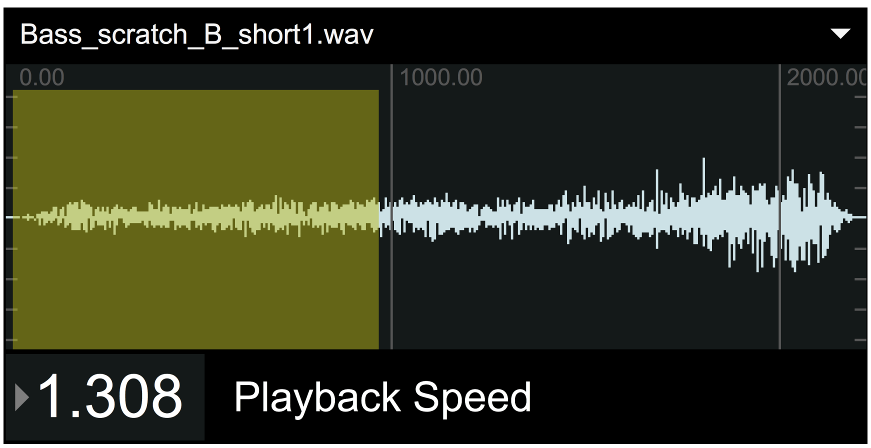

Figure 3.16: The graphical user interface of SM_Groove

3.6.8 SM_Overdrive

SM_Overdrive amplifies the input signal while limiting the signal to ±1. The distortion caused by the clipping of the signal produces several overtones that have a harmonic relationship with the original signal. A knob on the main controller is assigned to control the level of amplification. The output signal is adjusted for maintaining a stable loudness level since higher values cause an increase in the volume.

3.6.9 SM_Polybuffer

SM_Polybuffer comprises a multiple-layered sample player that reads multiple audio files in a folder, copies them into the computer memory, and randomly plays them in succession. Playback speed can be randomly selected for each instance, or it can be controlled by a fader if preferred. The user can attain weighed randomness by drawing on the customisable probability distribution table (Figure 3.17). The module can handle up to 100 layers of audio playback in both directions, as well as time-stretching the samples. Using a knob on the main control surface, the trigger frequency can be set from 5 milliseconds to 2 seconds, and the duration of each audio sample can be altered between 5 milliseconds to 8 seconds.

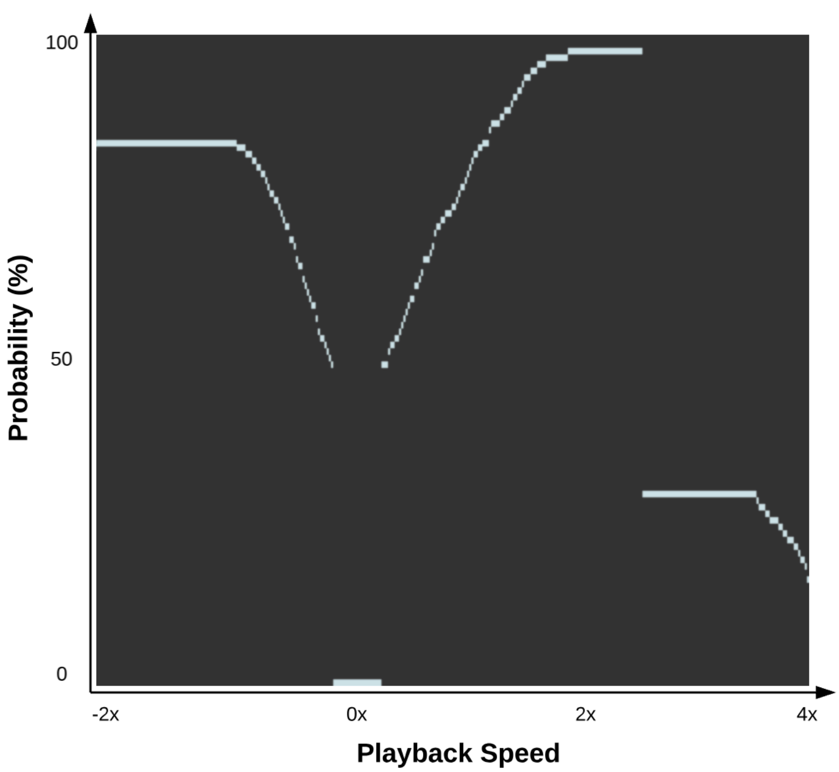

Figure 3.17 SM_Polybuffer's customizable probability distribution table

3.6.10 SM_Reverb

SM_Reverb applies a reverberation effect to the input signal, and uses *yafr2* object, which is integrated into *Max*. The *yafr2* object uses a plate-type reverberation algorithm. It has controls for room size, decay time, high-frequency damping, and diffusion. The user can change the decay time during the performance using a fader on the main controller. In case SM_Reverb is used as a parallel effect unit, it is activated by holding down a button. The user can set the portion of the original signal going into the module and assign different buttons for sending signal from individual sound sources to SM_Reverb to reverberate them separately.

3.6.11 SM_SamplePlayer

SM_SamplePlayer plays back the samples that are loaded into the buffer with a button on the main controller. The user can manipulate the playback speed between 0.1x and 8x the original speed using a fader and activate time-stretching with a button. It has a GUI providing information about the audio sample and the playback speed (Figure 3.18).

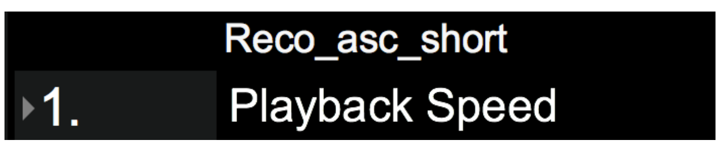

Figure 3.18: The GUI of SM_SamplePlayer

3.6.12 SM_VowelFilter

SM_VowelFilter contains four band-pass filters to generate the eight vowels of the Turkish language (Figure 3.19). Based on Bingöl and Nedim’s research, the frequencies, the bandwidths and the levels of the first four formants are set for each band-pass filter (318). A button on the main controller activates and deactivates the filter, and another button randomly selects a vowel. A *Max* object, *urn*, ensures that the successive selections are always different. When this effect is triggered, each filter gradually moves towards the closest frequency, and creates a transition between two vowels. The speed and the resolution of this movement can be set in the module to achieve the desired effect.

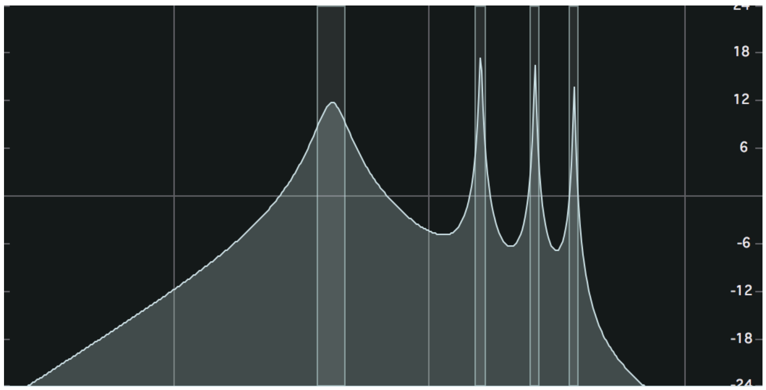

Figure 3.19 Graphical user interface of SM_VowelFilter

Chapter 4: Portfolio of Electroacoustic Music Compositions: *Sonic Matters*

Sonic Matters is a collection of five multi-channel electroacoustic pieces that are composed and performed using only *Sonic Matter* (Soydan, *Sonic Matters*). The pieces are structured improvisations, thus offering a unique listening experience every time they are performed. Each composition focuses on both a specific 'matter' of sound and a module of *Sonic Matter*, thereby demonstrating the strengths of the system in composing and performing music. The sound material used in the pieces includes everyday objects such as coins and bottles, instruments, and human voice. In this chapter, each piece is detailed based on the recordings from the first public concert, performed on June 14th, 2018. The recordings from the première can be found in Appendix B.

4.1 Artistic Approach and Creative Process

In order to compose or perform, the system requires sound samples to work before anything else. As a sonic artist, I have always been fascinated by the sonic possibilities that any sound offers after it is transformed. However, depending on the method of processing, these transformations usually degrade the sound material; therefore, maintaining a high sound quality is determined by the original recording. For that reason, I recorded all the sound sources from a very close distance to isolate them from their background and have a high signal-to-noise ratio. These close recordings allowed me to push the limits of transformation that can be done with *Sonic Matter* without having digital artifacts for the most part. I assigned the sound material relevant to the theme and focus of each piece. The first round of elimination targeted the recordings that were of insufficient sound quality. Then, I loaded them into the system and attempted to reveal their sonic potential. In the second round, I removed the ones that yielded uneventful and uninspiring results. Surprisingly, I ended up using only one fourth of the total number of recordings.

Although the sound processing makes it difficult to discern the original cause of the sound, one can identify almost all the sound material at some point throughout any piece: most of them from the very beginning, and the rest after they become a part of the sonic environment. Trevor Wishart points out that listeners require a certain minimum time to recognize the initial sound, the metamorphosis, and the final sound; thus, he calculates the time proportions in his work, describing this process as 'slow improvisation' (Vassilandonakis 10). I embraced Wishart's 'slow improvisation' concept as a performance principle to determine the duration of musical phrases whenever I perform any of the pieces.

The composition process is an ongoing cycle that makes the pieces evolve over time (Figure 4.1). I load each sound sample to an individual module one at a time and improvise for a while to discover interesting sonic results. In the meantime, I record all the performances, so that I can listen to them later and select the parts that work best for the conception of my pieces. Finally, I arrange the parts aesthetically and create the necessary cues for the modules. After loading the cues, I perform the piece once again and repeat the cycle.

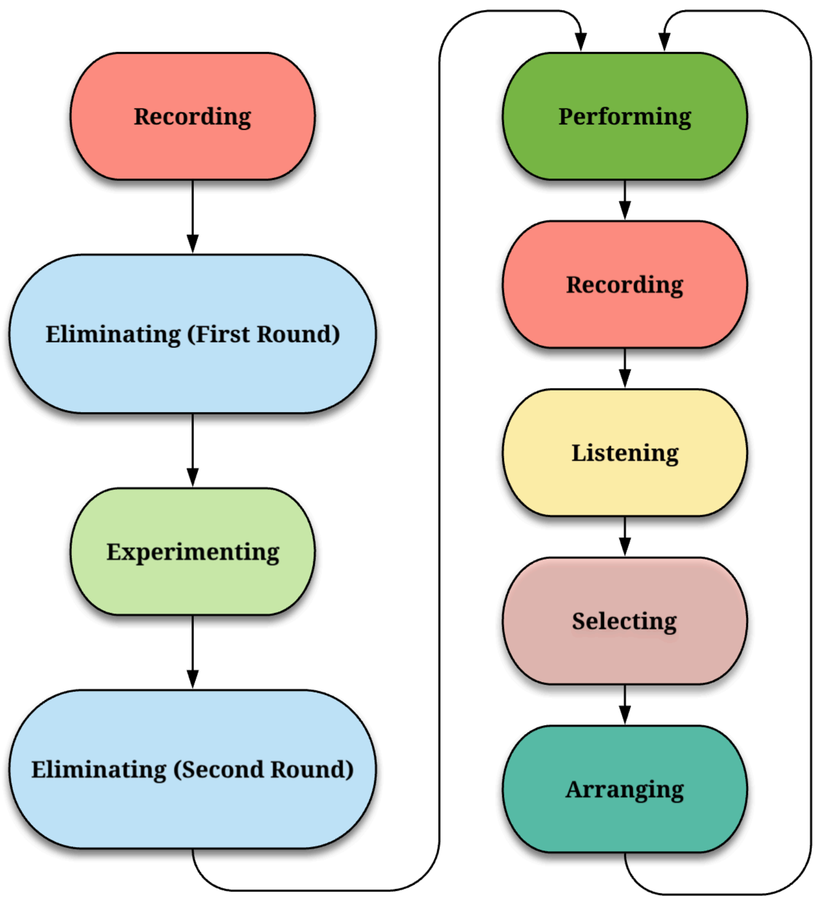

Figure 4.1: Creative process followed in *Sonic Matters*

4.2 *Composed of Metal and Wood*

Composed of Metal and Wood is a structured improvisation piece for *Sonic Matter* and multi-channel sound diffusion. It explores the sonic transformation of selected objects and instruments that are made of metal and wood (Table 4.1). The piece has four main sections that are divided by smooth transitions and each section is governed by its distinctive characteristic (Figure 4.2).

The sound material heard throughout the piece mostly consists of several percussive sounds such as falling coins and woodblock hits. The lines separating the sections are hard to discern as they melt into each other with gradual changes in texture and reverberation. Additionally, each section exhibits a change in the balance between wooden and metal sound material.

The prominent feature of the first section is sonic diffusion/immersion. The section starts with a repetitive unaltered falling coin sample that moves in different directions in the sonic space. It is followed by a rattling metal plate sound which then leads to a rapid flurry of the falling coins to increase the textural density over time (00:30-00:50). Starting around 00:50, the material starts to decrease in both the density and the pitch. At the 01:20 mark, a guiro sound is introduced and repeated a few times to demonstrate the timbral contrast between metal and wooden sound material. Also, several instances of reverberation create transitions between individual sound sources.

The second section makes use of *SM_Delay* to create rhythmic pulses. At the beginning of this section, the guiro sound is developed by delaying and transposition. Another spectrally similar sound object, the woodblock, is introduced near the beginning of the section to give a sense of familiarity with the guiro. I delay woodblock hits and feed them back into the *SM_Delay* module as I transpose the sound material up and down and change the location of stress in the rhythm (02:06-2:53). The *SM_Gesture* randomly plays rolling coin samples at low speeds through each speaker (at one tenth the original speed) hence creating a low-pitched sonic environment immersing the audience (02:29-02:40). Near the end of the section, the close-sounding popcorn pot sample provides a large movement effect (03:45-4:05) owing to its density and motion to conclude the section (Figure 4.3).

In the third section, the woodblock sample used in *SM_SamplePlayer1* is kept, albeit it is loaded into *SM_Groove* for greater expression. Retaining the sound sample bridges the previous section to this one with the performance of a similar gesture with glissandi (04:11-05:13). The section starts with sparse bursts of sounds and builds up by becoming increasingly dense and intense until around 05:35 mark. The slowed-down guiro is played a few times with interruptions to provide a gritty and percussive texture along with building a contrast with the high-pitched woodblock sound (4:58-5:56).

The final section begins with a low-pitched sustaining tone that originates from bowed crotales and cowbells (05:54-07:04). Created through time-stretching, it offers a stable foundation as a slow-developing drone until the end of the piece. Meanwhile, the percussive texture takes over at 06:11 and demonstrates some change in pitch, density and sound diffusion reminiscent of the gesture at the beginning of the second section to conclude the piece.

	Section 1	Section 2	Section 3	Section 4
SM_Groove	Coins_Fall01	Coins_Fall01	Woodblock_High	Guiro01
SM_SamplePlayer1	Guiro01	Woodblock_High	Guiro01	Crotales_Cowbell
SM_SamplePlayer2	MetalPlate_Shake	MetalPlate_Shake	MetalPlate_Shake	MetalPlate_Shake
SM_Gestures	Coins_Fall01 Coins_Fall02 Coins_Fall03 Coins_Fall04 Coins_Fall05 Coins_Fall06 Coins_Fall07 Coins_Fall08 Coins_Fall09 Coins_Fall10 Coins_Fall11 Coins_Fall12 Coins_Fall13 Coins_Fall14 Coins_Fall15	Coins_Roll01 Coins_Roll02 Coins_Roll03 Coins_Roll04 Coins_Roll05	Coins_Roll01 Coins_Roll02 Coins_Roll03 Coins_Roll04 Coins_Roll05	Coins_Fall01 Coins_Fall02 Coins_Fall03 Coins_Fall04 Coins_Fall05 Coins_Fall06 Coins_Fall07 Coins_Fall08 Coins_Fall09 Coins_Fall10 Coins_Fall11 Coins_Fall12 Coins_Fall13 Coins_Fall14 Coins_Fall15
Gesture Mode	Continuous (Toggle)	Continuous (Toggle)	Continuous (Toggle)	One cycle (8 instances)
SM_Polybuffer	Coins_Fall01 Coins_Fall02 Coins_Fall03 Coins_Fall04 Coins_Fall05 Coins_Fall06 Coins_Fall07 Coins_Fall08 Coins_Fall09 Coins_Fall10 Coins_Fall11 Coins_Fall12 Coins_Fall13 Coins_Fall14 Coins_Fall15	PopcornPot_Rotation01 PopcornPot_Rotation02	PopcornPot_Rotation01 PopcornPot_Rotation02	PopcornPot_Rotation01 PopcornPot_Rotation02

Table 4.1: The sound samples that each module uses per section in *Composed of Metal and Wood*

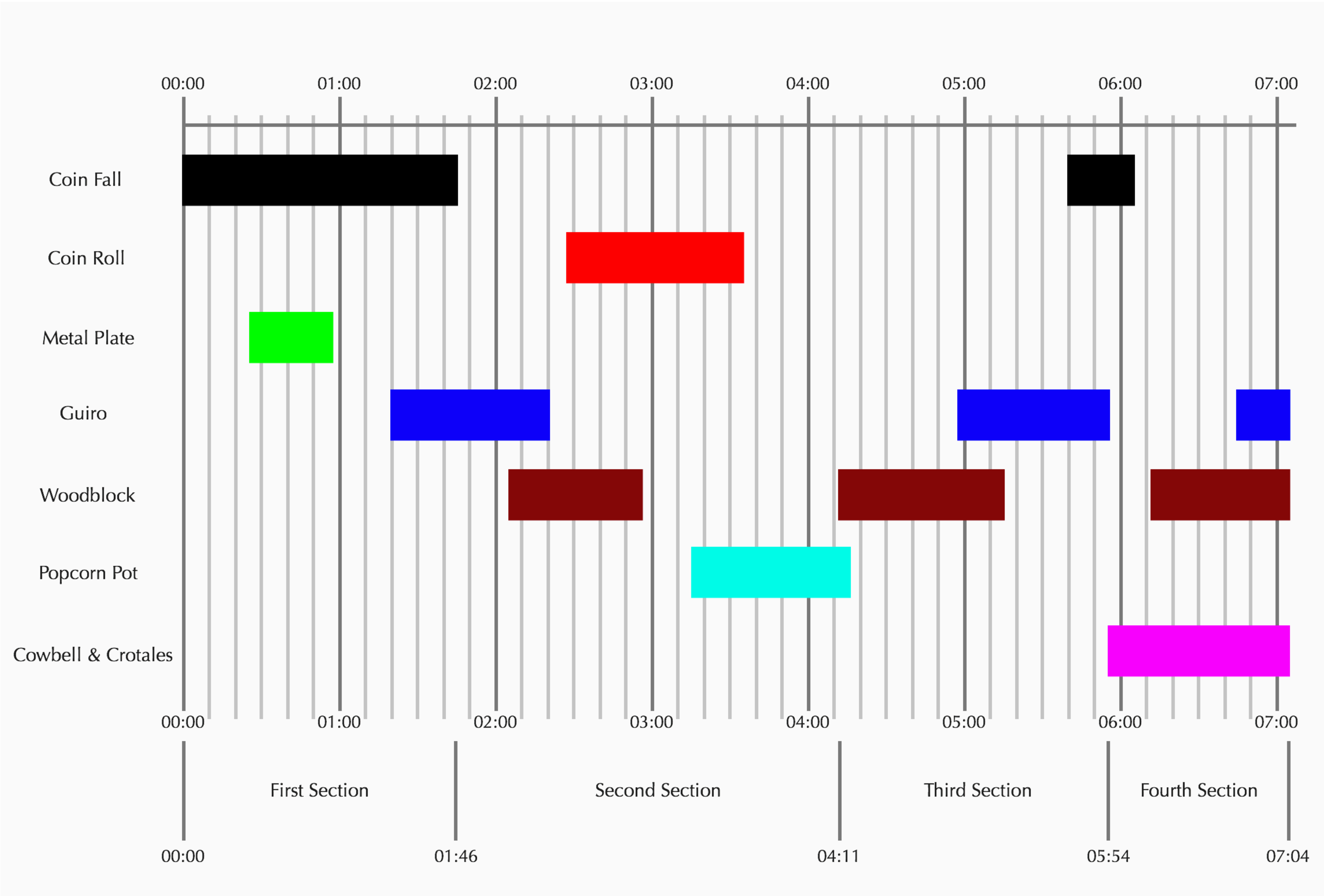

Figure 4.2: Layers of sound material in *Composed of Metal and Wood*

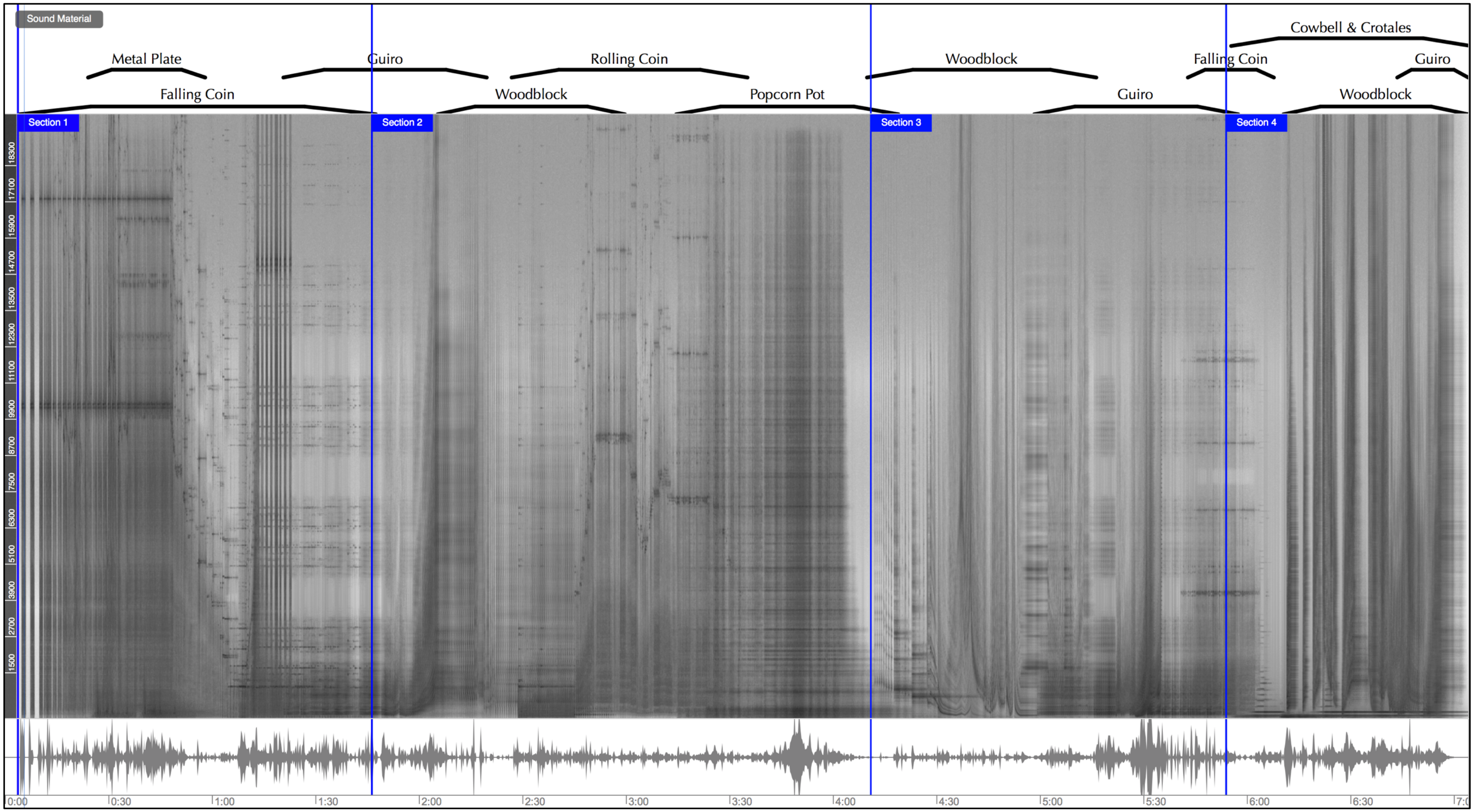

Figure 4.3: Sonogram and waveform of *Composed of Metal and Wood*

4.3 *The Passage*

The Passage is a structured improvisation piece for *Sonic Matter* and multi-channel sound diffusion. It focuses on sound spatialization and motion along with demonstrating *SM_Ambipanning*'s expressive capabilities in creating believable movements in sonic space. The piece includes only female voice recordings as its sound material including musical gestures, syllables, and vowels (Table 4.2). The piece comprises four sections with each section using a different sound material or method of sound transformation (Figure 4.5 and Figure 4.6).

The first section starts with the combination of *SM_Gesture* and *SM_VowelFilter* to create an immersive sonic environment. I preferred a spectrally rich sound sample ('ch' sound) for *SM_VowelFilter* (00:00-00:41) as the filtering process leaves out the frequency bands that create the vowels (Figure 4.7). Owing to the short attack of the sound material, *SM_Gestures*' slight timing fluctuations create an illusion of variety while the module is playing back the same sound material (00:00-00:50). After the filter is out, the same sound material forms a similar but glitchy immersive sonic texture and recedes to the background after ascending and finally descending in pitch (00:55-01:51). Transposing the purring foreground sound material while dynamically controlling the fast motion generates a *Doppler effect* for realistic movements in the space (01:51-03:00). At the end of the section, we hear the tail of the reverberation which connects the first section to the next.

In the second section, I introduce a descending musical gesture in response to the one from the previous section and manipulate it with delay and pitch shifting (03:04-04:04). The sound object follows the trajectory of the foreground element from the previous section.

Sustaining the vowel at the end of 'tra' results in a smooth crossfade between itself and the chorus of 'a' sounds (03:50-04:07). Then, I play an ascending musical gesture using *SM_Gesture*, which forms a turbulent texture and exhibits several ascending and descending glissandi rising from a very low pitch (04:42-05:05). The glissandi cover more distance than the original recording because of two main reasons: the triggering frequency is at a fast setting, therefore only the beginning of the original recording is audible; and I apply upward transposition to the already ascending material. (Figure 4.4).

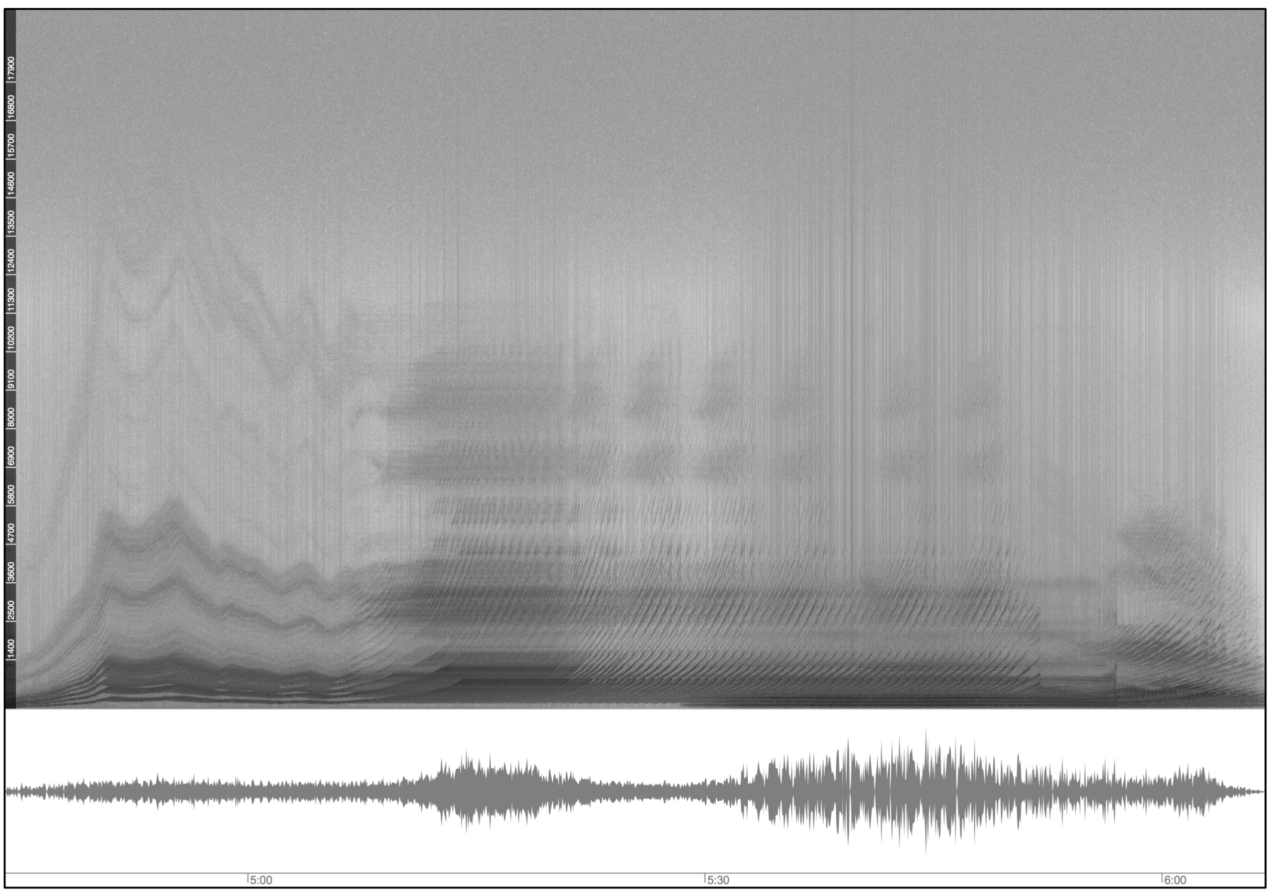

Figure 4.4: Ascending motion in the second section of *The Passage* (04:45-06:08)

The third section starts with an abrupt contrast and brings back the vowels heard at the beginning of the piece in an unfiltered form moving in the sonic space (06:06-07:20). *SM_Gesture* creates an heterorhythmic cloud of sound that comes into existence with a crescendo around 06:34, which then expresses a firmer sense of unity at 07:27.

SM_Polybuffer randomly triggers the different vowels in high playback speeds which results in a dense synth-like sound (07:51-08:55). At this point, the vowels from earlier emerge once again to create a drone (08:34) which leads to the climax of the piece around 09:23. I then attempt to match the pitch of the chorus with the repeated high-pitched voice sound and blend in harmonically towards the finale (09:40-10:11).

	Section 1	**Section 2**	**Section 3**	**Section 4**
SM_Groove	FVoc_brrra	FVoc_tra_desc	FVoc_ah_short	Fvoc_ah_short
SM_SamplePlayer1	FVoc_a_short	FVoc_a_short	FVoc_a_short	FVoc_a_short
SM_SamplePlayer2	FVoc_brrra	FVoc_brrra	FVoc_brrra	FVoc_brrra
SM_Gestures	FVoc_ch_rhythmic	FVoc_o_asc	FVoc_mannennini	FVoc_vowels FVoc_vowels_hi
Gesture Mode	Continuous (Toggle)	Continuous (Toggle)	Continuous (Toggle)	Continuous (Toggle)
SM_Polybuffer	FVoc_a_short FVoc_ah_short Fvoc_brrra	FVoc_a_short FVoc_ah_short Fvoc_brrra	FVoc_a_sus FVoc_i_sus FVoc_o_sus FVoc_u_sus FVoc_ue_sus	FVoc_a_sus FVoc_i_sus FVoc_o_sus FVoc_u_sus FVoc_ue_sus

Table 4.2: The sound samples that each module uses per section in *The Passage*

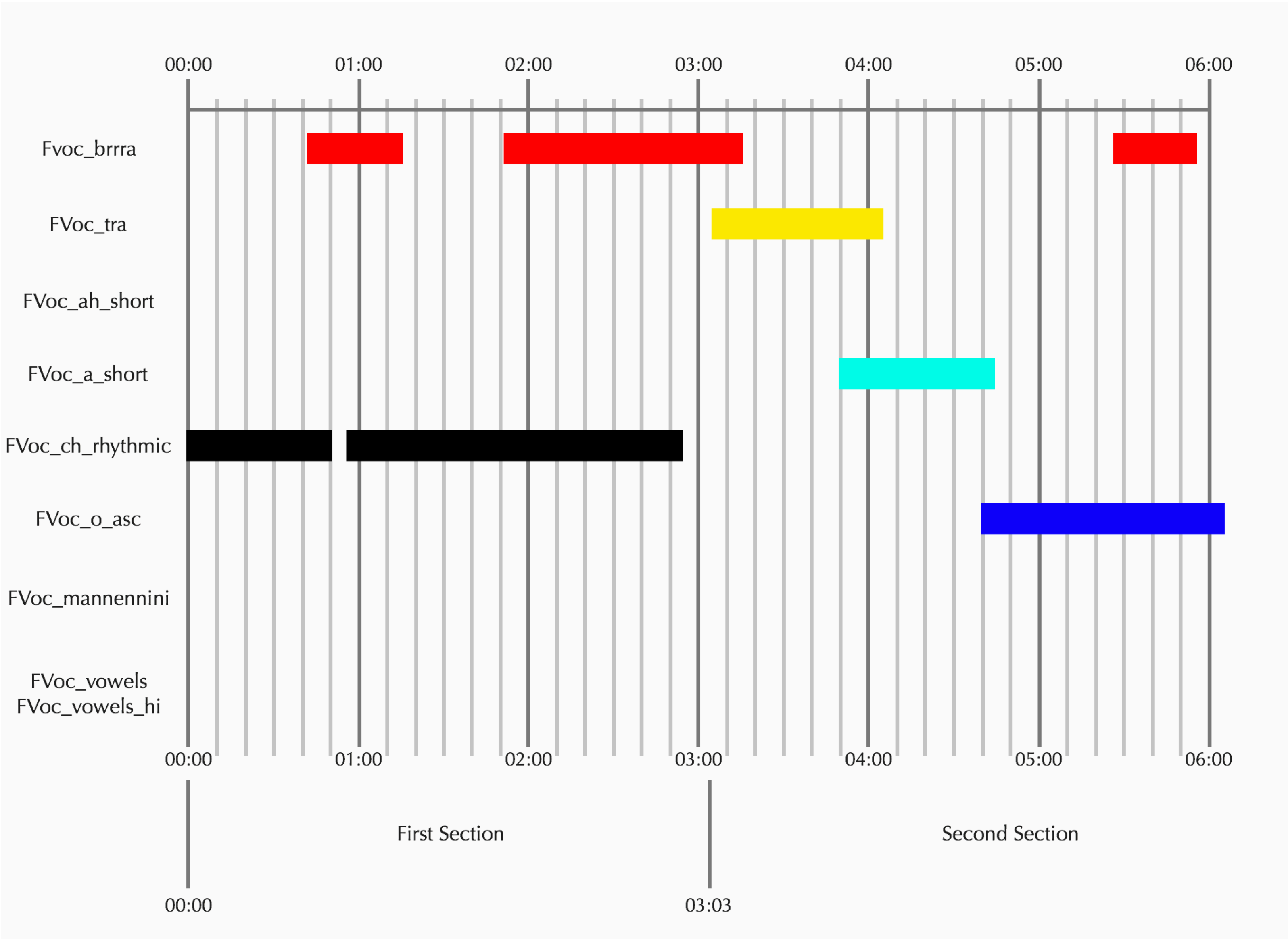

Figure 4.5: Layers of sound material in *The Passage* (00:00-06:00)

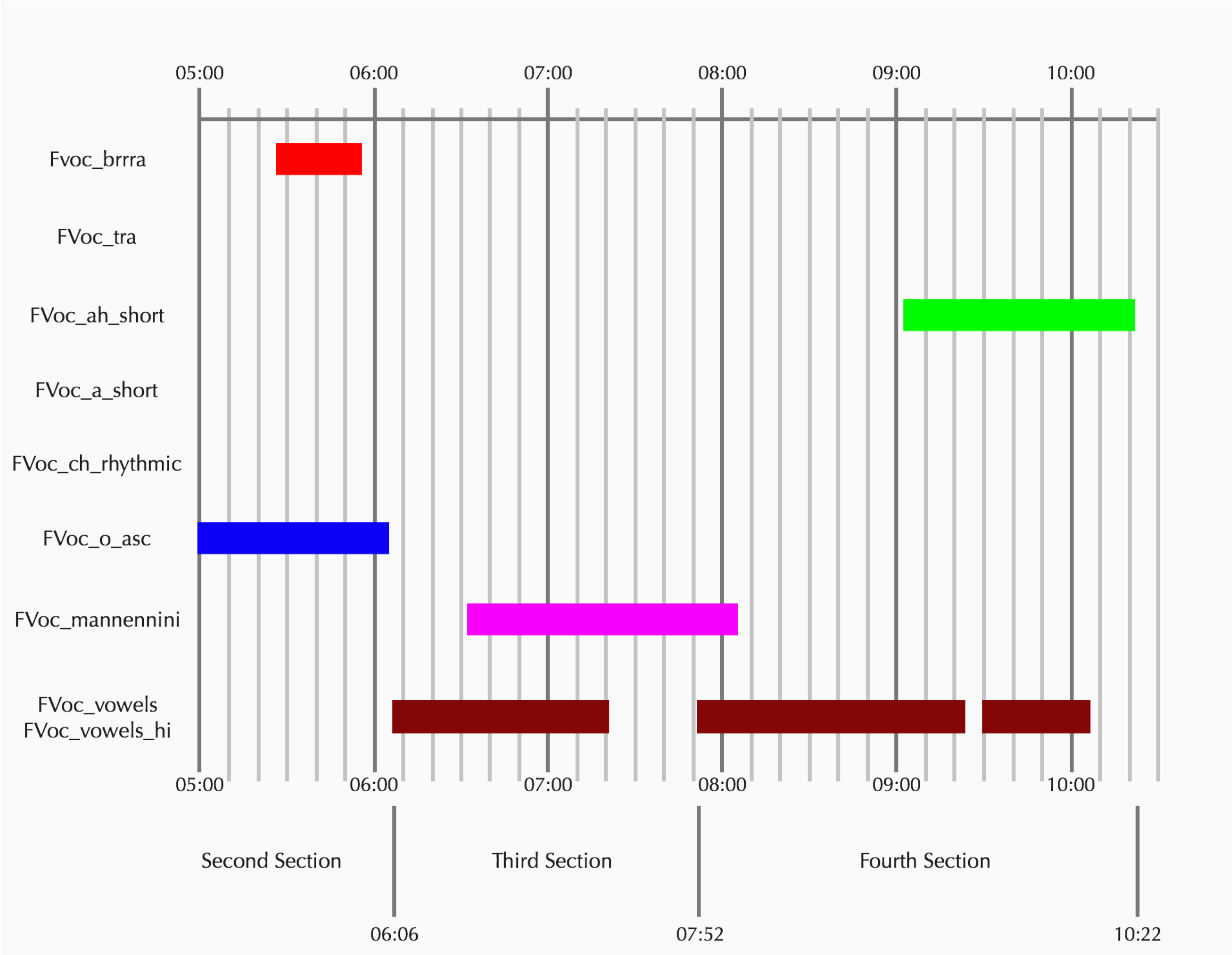

Figure 4.6: Layers of sound material in *The Passage* (06:00-10:30)

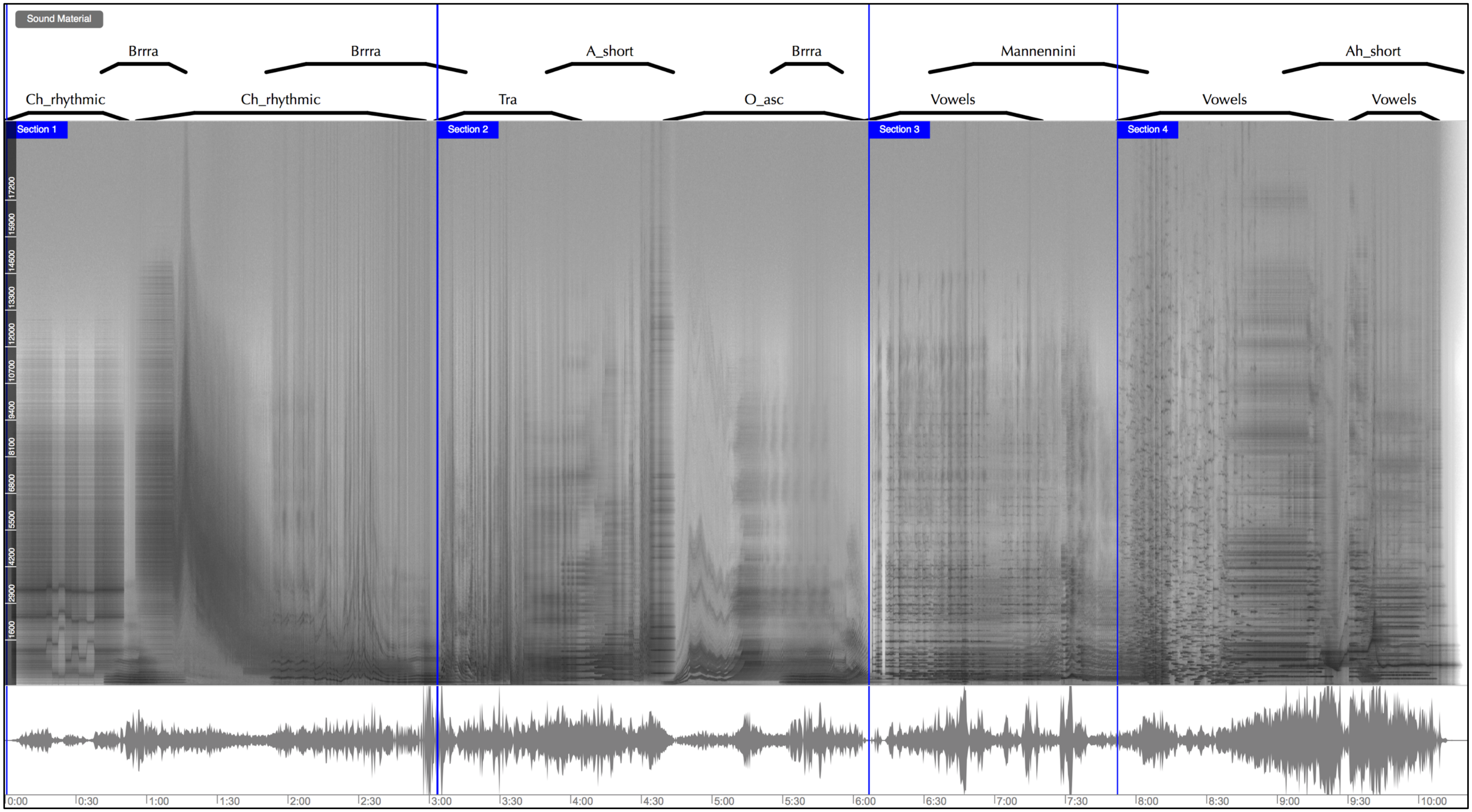

Figure 4.7: Sonogram and waveform of *The Passage*

4.4 *Up/Down*

Up/Down is a structured improvisation piece for *Sonic Matter* and multi-channel sound diffusion. The sound material comprises desk bells of various pitches and an African musical instrument called mbira: a thumb piano with staggered metal tines and bottle caps (Figure 4.8). The piece is made up of four sections and the dividing lines between them are easily defined by the long reverberation at the end of each section.

Figure 4.8: Mbira, an African thumb piano (Weeks)

The first section opens with a reversed desk bell sound. As it builds up, the original version of the desk bell and mbira coincide with the attack portion of the reversed desk bell in unison (00:00-00:29). A similar gesture accumulates until around the 00:37 mark and disperses with a long reverberation. The idea is then developed with the delay and the displacement of two sound objects' timing (00:50-01:07). Further development occurs when their pitches rise and fall to create assorted pitch clusters (01:08-01:52). Additionally, their contrasting independent spatial trajectories gradually intersperse the texture all around the listening space. The section concludes with the long reverberation that flows into the beginning of the next section (01:46-02:03).

Desk bells in different pitches are randomly played through different speakers from the beginning of the second section to 02:45 mark. Meanwhile, a continuously rising texture comes into existence and disappears (02:11-02:25). Then, a fast stream of desk bell sounds with short attack and release characteristics form a granular texture, complementing the spectral fabric of the background but contrasting the rhythmic density (02:36). Between 02:51 and 03:25, the tails of the sounds increasingly become more audible, which consequently fabricates pitch clusters using random playback speeds and reversing (Figure 4.9). The section concludes with another long reverberation that permeates the opening of the next one.

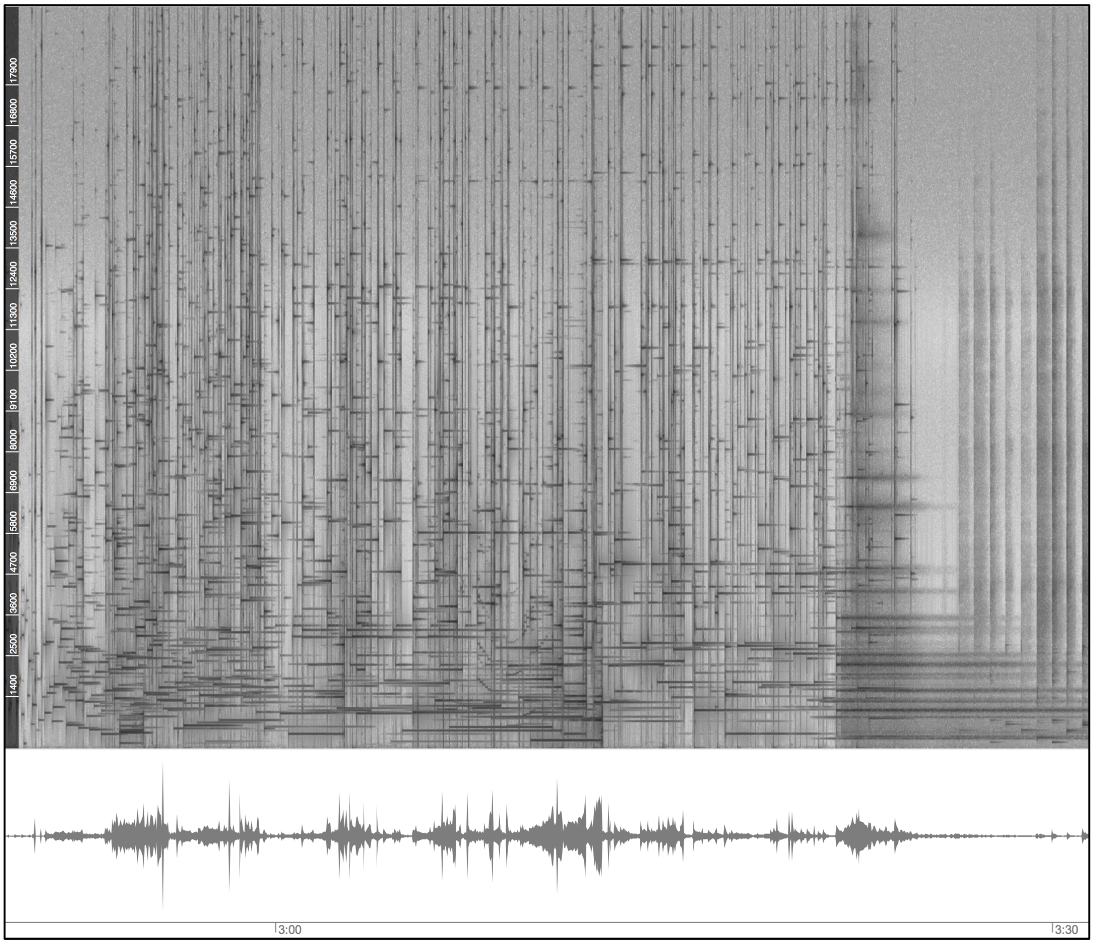

Figure 4.9: Pitch clusters created in the second section of *Up/Down* (02:50-03:32)

At the beginning of the third section, the same musical gesture from the previous section (02:11-02:25) comes into play, but this time with mbira sounds (03:25-04:13). The idea at the very beginning of the piece is also recalled with similar sound material but at a different pitch (03:48-04:13). The long reverberation delineates the end of the section as the mbira sounds are transposed down (04:03-04:13).

The last section begins with small bursts of time-stretched mbira sounds, which fluctuate in pitch up or down at each entrance (04:14-04:26). These bursts then become a streaming texture (04:27) that expands in the spectrum with random pitches and reversing, leading to the climax of the piece (04:35-04:51). The pulsating mbira sound moves around the space until the abrupt ending of the piece (04:49-05:12). Finally, the mbira sound is played in various pitches and gestures with reverberation to bring the piece to an end.

	Section 1	**Section 2**	**Section 3**		**Section 4**	
SM_Groove	Desk_Bell07	Desk_Bell09	Desk_Bell07		Desk_Bell07	
SM_SamplePlayer1	Mbira03	Mbira02	Mbira04		Mbira05	
SM_SamplePlayer2	Desk_Bell07	Desk_Bell07	Desk_Bell07		Desk_Bell07	
SM_Gestures	Desk_Bell01	Desk_Bell01				
	Desk_Bell02	Desk_Bell02	Mbira01	Mbira11	Mbira01	Mbira11
	Desk_Bell03	Desk_Bell03	Mbira02	Mbira12	Mbira02	Mbira12
	Desk_Bell04	Desk_Bell04	Mbira03	Mbira13	Mbira03	Mbira13
	Desk_Bell05	Desk_Bell05	Mbira04	Mbira14	Mbira04	Mbira14
	Desk_Bell06	Desk_Bell06	Mbira05	Mbira15	Mbira05	Mbira15
	Desk_Bell07	Desk_Bell07	Mbira06	Mbira16	Mbira06	Mbira16
	Desk_Bell08	Desk_Bell08	Mbira07	Mbira17	Mbira07	Mbira17
	Desk_Bell09	Desk_Bell09	Mbira08	Mbira18	Mbira08	Mbira18
	Desk_Bell10	Desk_Bell10	Mbira09	Mbira19	Mbira09	Mbira19
	Desk_Bell11	Desk_Bell11	Mbira10	Mbira20	Mbira10	Mbira20
	Desk_Bell12	Desk_Bell12				
Gesture Mode	Continuous (Toggle)	Continuous (Toggle)	Continuous (Toggle)		Continuous (Toggle)	
SM_Polybuffer	Desk_Bell01	Desk_Bell01				
	Desk_Bell02	Desk_Bell02	Mbira01	Mbira11	Mbira01	Mbira11
	Desk_Bell03	Desk_Bell03	Mbira02	Mbira12	Mbira02	Mbira12
	Desk_Bell04	Desk_Bell04	Mbira03	Mbira13	Mbira03	Mbira13
	Desk_Bell05	Desk_Bell05	Mbira04	Mbira14	Mbira04	Mbira14
	Desk_Bell06	Desk_Bell06	Mbira05	Mbira15	Mbira05	Mbira15
	Desk_Bell07	Desk_Bell07	Mbira06	Mbira16	Mbira06	Mbira16
	Desk_Bell08	Desk_Bell08	Mbira07	Mbira17	Mbira07	Mbira17
	Desk_Bell09	Desk_Bell09	Mbira08	Mbira18	Mbira08	Mbira18
	Desk_Bell10	Desk_Bell10	Mbira09	Mbira19	Mbira09	Mbira19
	Desk_Bell11	Desk_Bell11	Mbira10	Mbira20	Mbira10	Mbira20
	Desk_Bell12	Desk_Bell12				

Table 4.3: The sound samples that each module uses per section in *Up/Down*

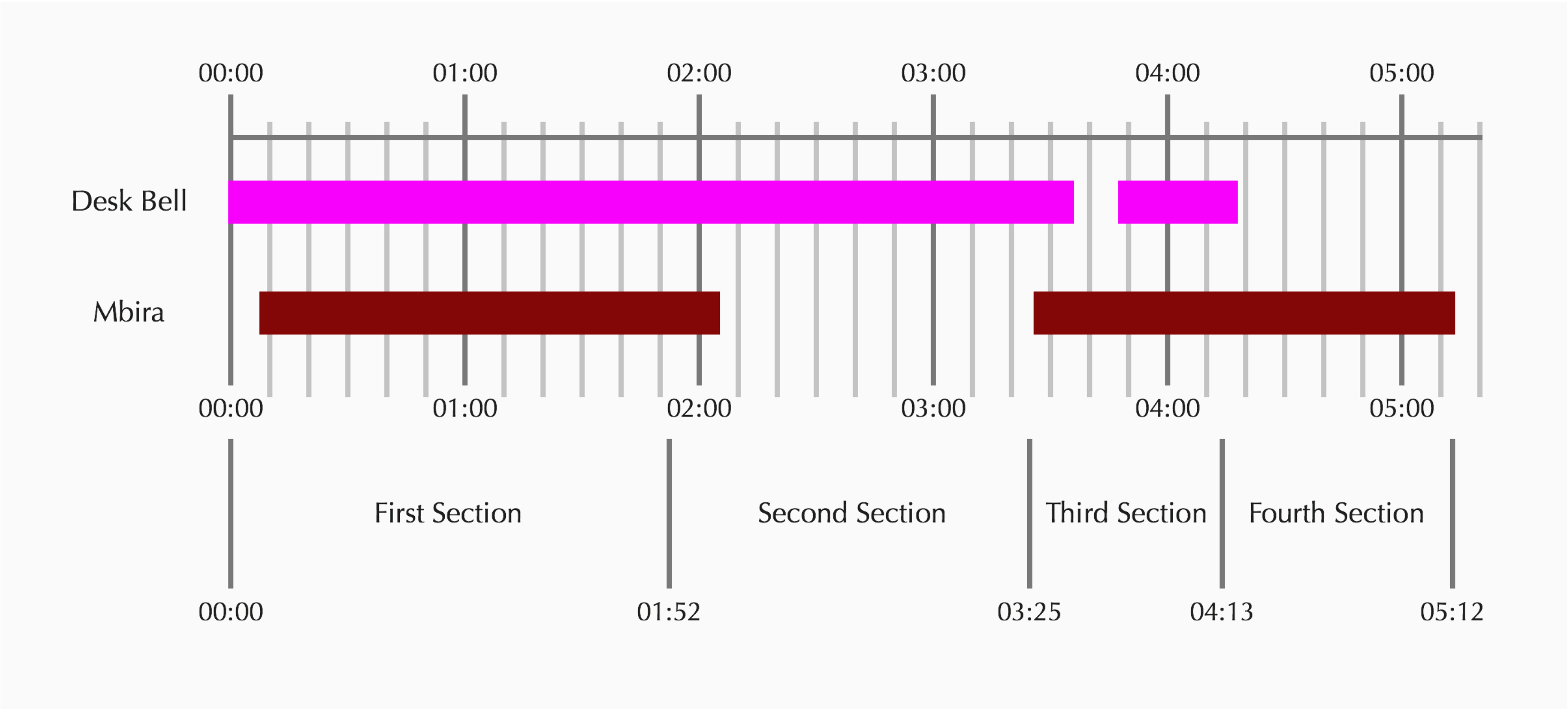

Figure 4.10: Layers of sound material in *Up/Down*

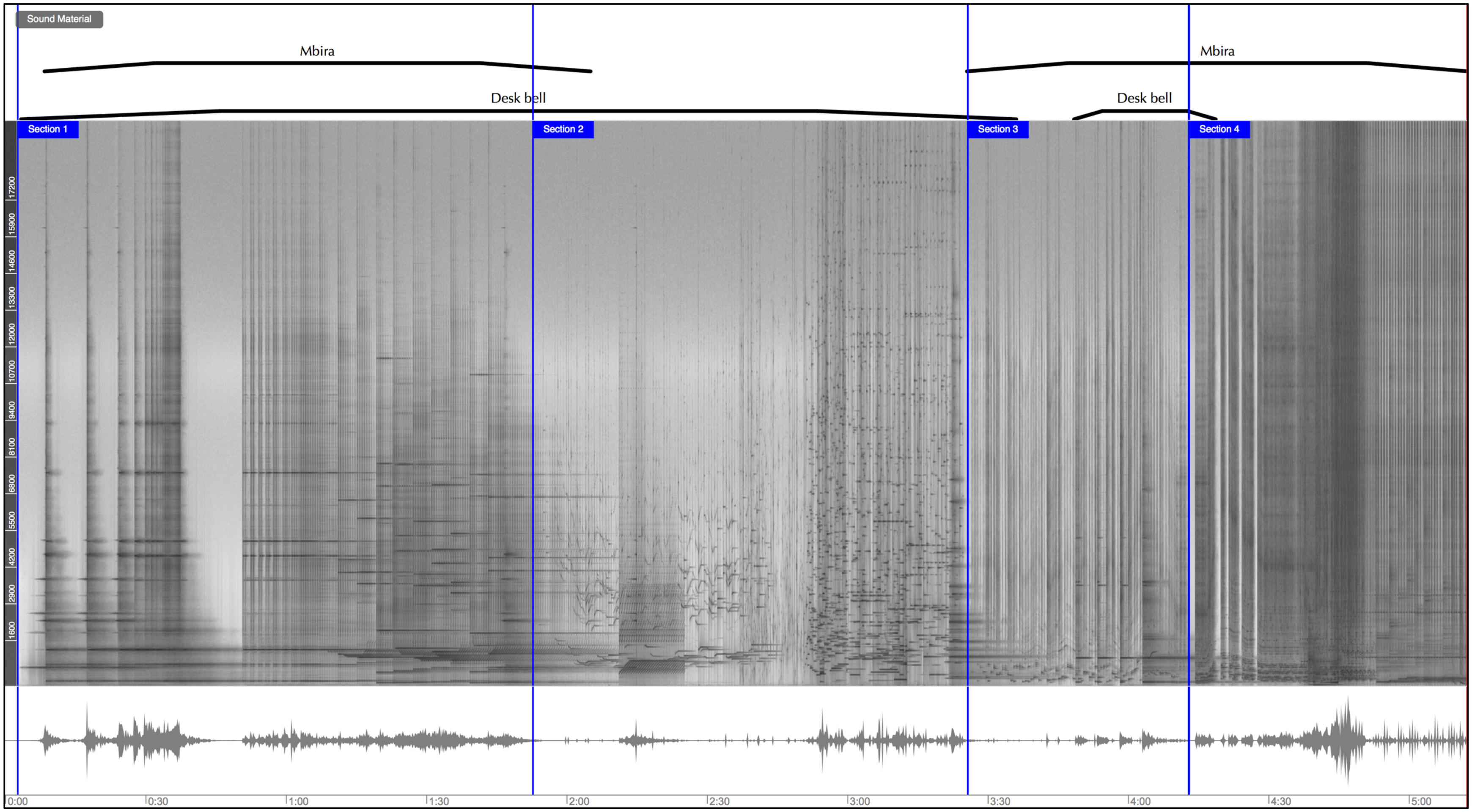

Figure 4.11: Sonogram and waveform of *Up/Down*

4.5 *Going in Circles*

Going in Circles is a structured improvisation piece for *Sonic Matter* and multi-channel sound diffusion. The piece uses assorted recordings of two everyday sound objects: bottle hits that have sharp transients and short attack/release, and bowl hits that have a similar envelope except for the long release. *Going in Circles* is a calm piece: the closely recorded sounds and their percussive nature create a sense of intimacy and closeness in the listening environment. Apart from a couple of exceptions, the overall intensity of the piece stays at a considerably low level, but the overall spectral character of the piece is dense and noise-like (Figure 4.12). The three sections of the piece have their own assigned sound material; however, the lines between them are difficult to draw since they employ uniform sound material.

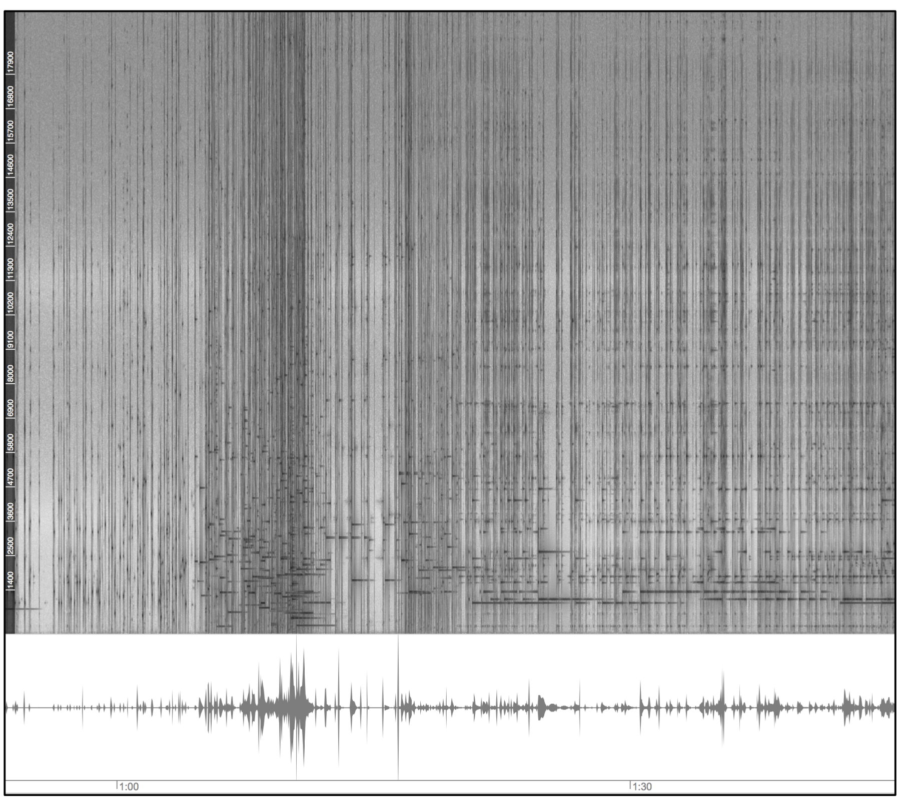

Figure 4.12: An example of the energy distribution of the sound material (00:56-01:45)

The piece opens with the juxtaposed bursts of a bowl sound with each instance coming through randomly selected speakers, sounding like a ringing old phone bell. At 00:27, the density of the sound diminishes while the bursts are transposed up and down. Then, the bowl sound settles down to its destination pitch and creates a stable sonic texture with repetitions in an even lower density (00:37-00:55). In the meantime, the fast playback of different bottle and bowl hit sounds produce a glitchy stream of sounds in the foreground, which contrasts with the background texture by bringing seamless motion across the space and variety in pitch (00:48-02:02). As their release durations increase, the tails of the samples become audible as well as the clicks caused by their sharp transients (01:05-01:13). Starting around 01:39 mark, the delayed bottle sound takes over the foreground and connects the section to the next one.

The low-pitched bottle sound from the previous section continues to develop with sporadic repetitions and delays to create various rhythmic pulses. A high-pitched bottle sound also comes into play around 02:08 and creates polyrhythms with the low one in the same way. The low bottle sound follows the spatial trajectory of the high one slightly after it to create an ongoing sense of pursuit throughout the section (02:06-02:55).

At the beginning of the last section, the gesture from the previous section is kept, but this time with bowl sounds rather than bottles (02:47-04:00). A few fluctuations in pitch and intensity are heard while the recurrences remain (03:08-03:24). The burst of bowl sounds from the beginning of the piece is reintroduced (03:31-03:38) as the piece winds up with delayed reiterations of the bowl.

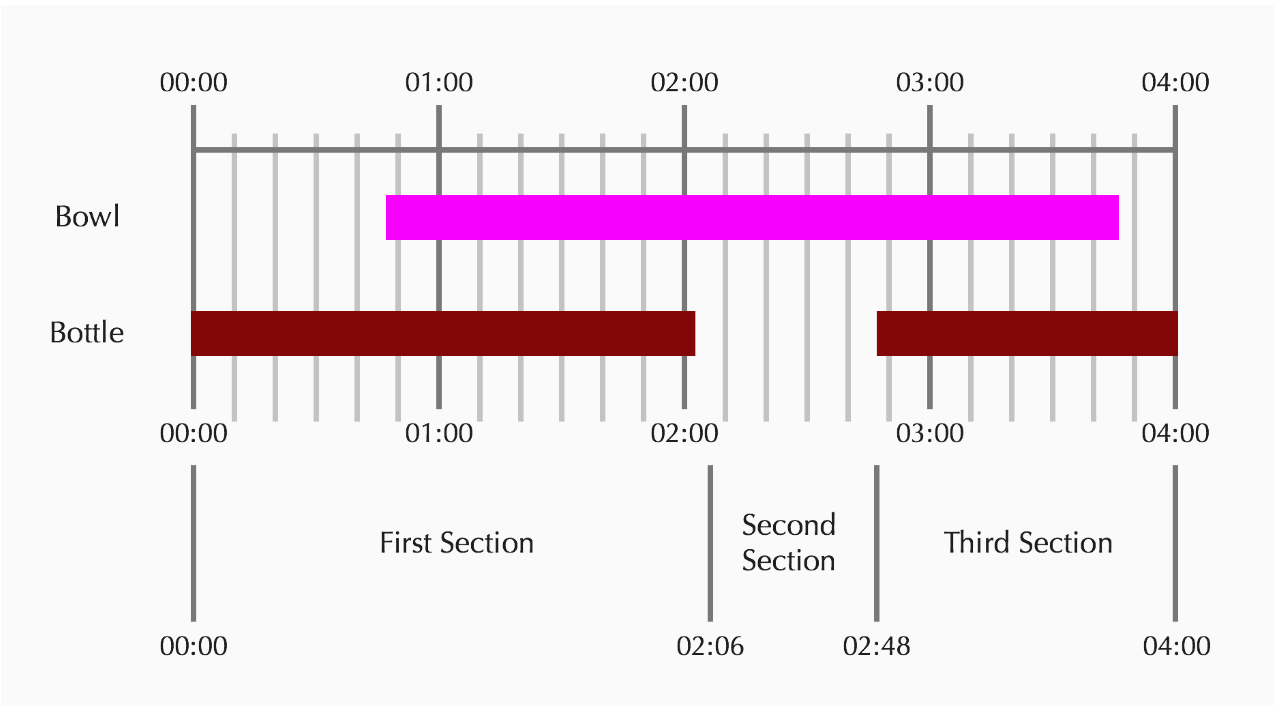

Figure 4.13: Layers of sound material in *Going in Circles*

	Section 1	Section 2	Section 3
SM_Groove	Bottle_Low1	Bottle_Low2	Bowl_Low1
SM_SamplePlayer1	Bottle_High1	Bottle_High2	Bowl_High1
SM_SamplePlayer2	Bottle_Mid	Bottle_Mid	Bottle_Mid
SM_Gestures	Bowl_Lowest1	Bowl_Highest1	Bottle_High1 Bottle_High2 Bottle_Low1 Bottle_Low2 Bottle_Mid Bowl_High1 Bowl_Highest1 Bowl_Low1 Bowl_Lowest1
Gesture Mode	Continuous (Toggle)	Continuous (Toggle)	Continuous (Toggle)
SM_Polybuffer	Bottle_High1 Bottle_High2 Bottle_Low1 Bottle_Low2 Bottle_Mid Bowl_High1 Bowl_Highest1 Bowl_Low1 Bowl_Lowest1	Bottle_High1 Bottle_High2 Bottle_Low1 Bottle_Low2 Bottle_Mid Bowl_High1 Bowl_Highest1 Bowl_Low1 Bowl_Lowest1	Bottle_High1 Bottle_High2 Bottle_Low1 Bottle_Low2 Bottle_Mid Bowl_High1 Bowl_Highest1 Bowl_Low1 Bowl_Lowest1

Table 4.4: The sound samples that each module uses per section in *Going in Circles*

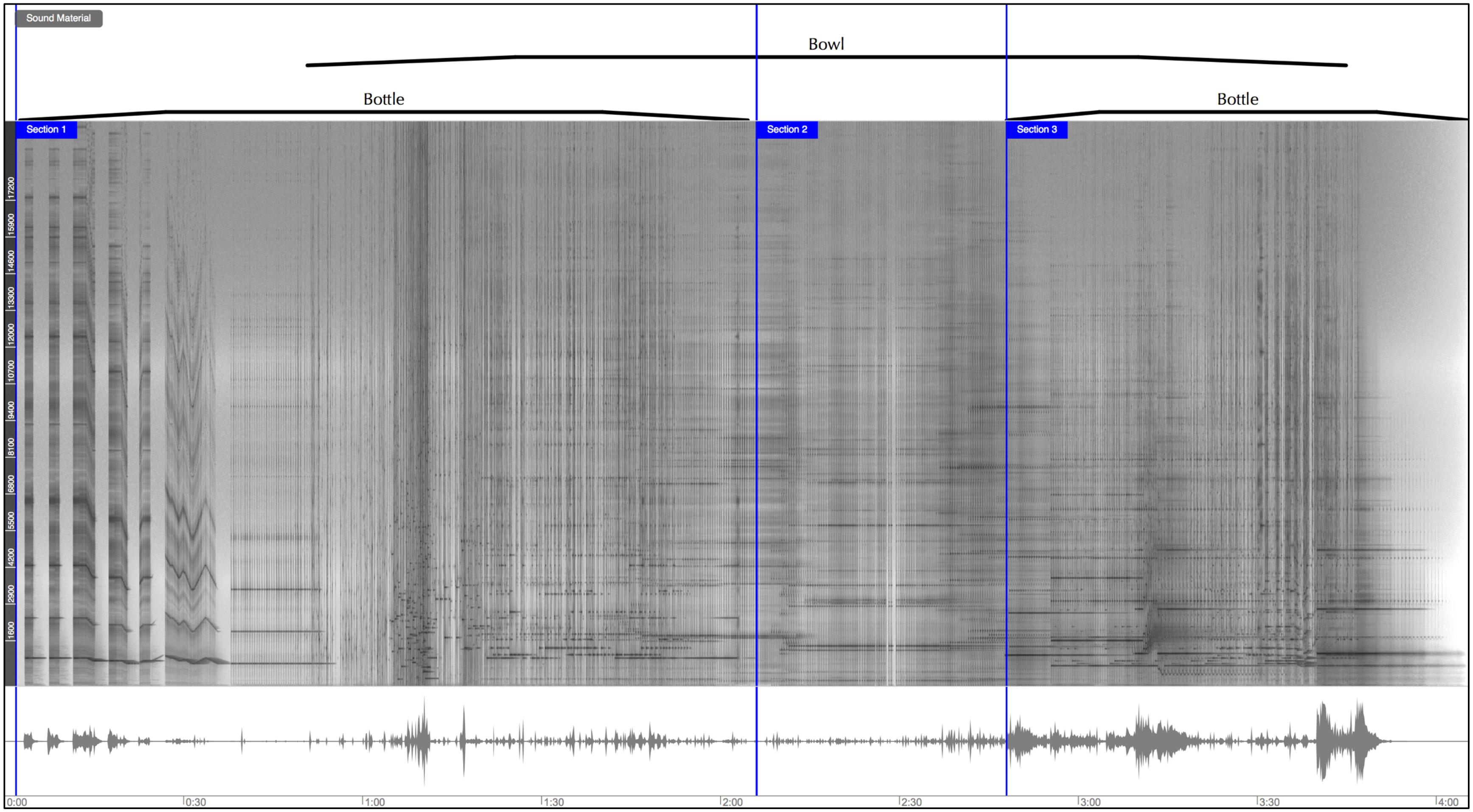

Figure 4.14: Sonogram and waveform of *Going in Circles*

4.6 *Flux*

Flux is a structured improvisation piece for *Sonic Matter* and multi-channel sound diffusion. The sound material consists of various recordings of bass guitar, recorder, and vibraphone. Recordings of the instruments exhibit several extended techniques such as scratching bass guitar strings, blowing into the recorder and bowing the vibraphone keys. The piece comprises two contrasting sections, and the separation between them is very clear since each section is characterized by different sound material.

The piece opens with a pulsating bass sound. The reverberated scratching sound creates a granular texture and moves continuously in the sonic space (00:18-01:27). At 01:15 mark, the vibraphone sound enters in response to low-pitched reverberant sound and disappears over time. While it loses its intensity, scratching sounds agglomerate and this creates a denser granular texture than the previous one (01:27-02:10). The density of the texture diminishes around the 01:44 mark, but it is developed further with a noisier texture that rises and falls (01:51-02:10). Right after that, the texture weakens once again and lingers for a while, and finally disperses at 02:40 mark. Meanwhile, a strumming sound that wanders around the space is introduced and developed with delays to form a coherent texture (02:28-02:57). Then, a higher-pitched strumming sound appears, and imitates the motion and development of the previous sound, densely filling the spectrum with a greater intensity (02:57-03:29). The granular texture from 01:44 mark, and the scratching sound from the beginning of the piece are reintroduced (03:19-03:50). The idea with the vibraphone sound is brought back and further developed with transposition to conclude the section with a sound that is harmonically coherent with the upcoming one (03:59-04:46).

The second section starts with a harmonic drone of a time-stretched recorder sound slowly roaming around the sonic space that goes on in the background until the end of the piece (04:46-06:40). A rising and falling stream of rhythmic airy/breathy sounds overlaps the harmonic texture (05:07-06:24). The vibraphone sound comes back for the last time to complement the harmonic texture and bring the piece to an end (05:58-06:30).

	Section 1	**Section 2**
SM_Groove	Bass_scratch_multi	Reco_dist_sus
SM_SamplePlayer1	Bowed_Vibes01	Bowed_Vibes01
SM_Gestures	Bass_scratch_A Bass_scratch_B Bass_scratch_B_short1 Bass_scratch_B_short2 Bass_scratch_D Bass_scratch_D_phasor Bass_scratch_E Bass_scratch_E_phasor Bass_scratch_G_asc Bass_scratch_G_short Bass_scratch_multi	Reco_blowing01 Reco_blowing02 Reco_blowing03 Reco_blowing04 Reco_blowing05 Reco_blowing06 Reco_blowing07 Reco_blowing_rhythmic
Gesture Mode	Continuous (Toggle)	Continuous (Toggle)
SM_Polybuffer	Bass_strum02 Bass_strum03 Bass_strum04	Reco_flutter_dist

Table 4.5: The sound samples that each module uses per section in *Flux*

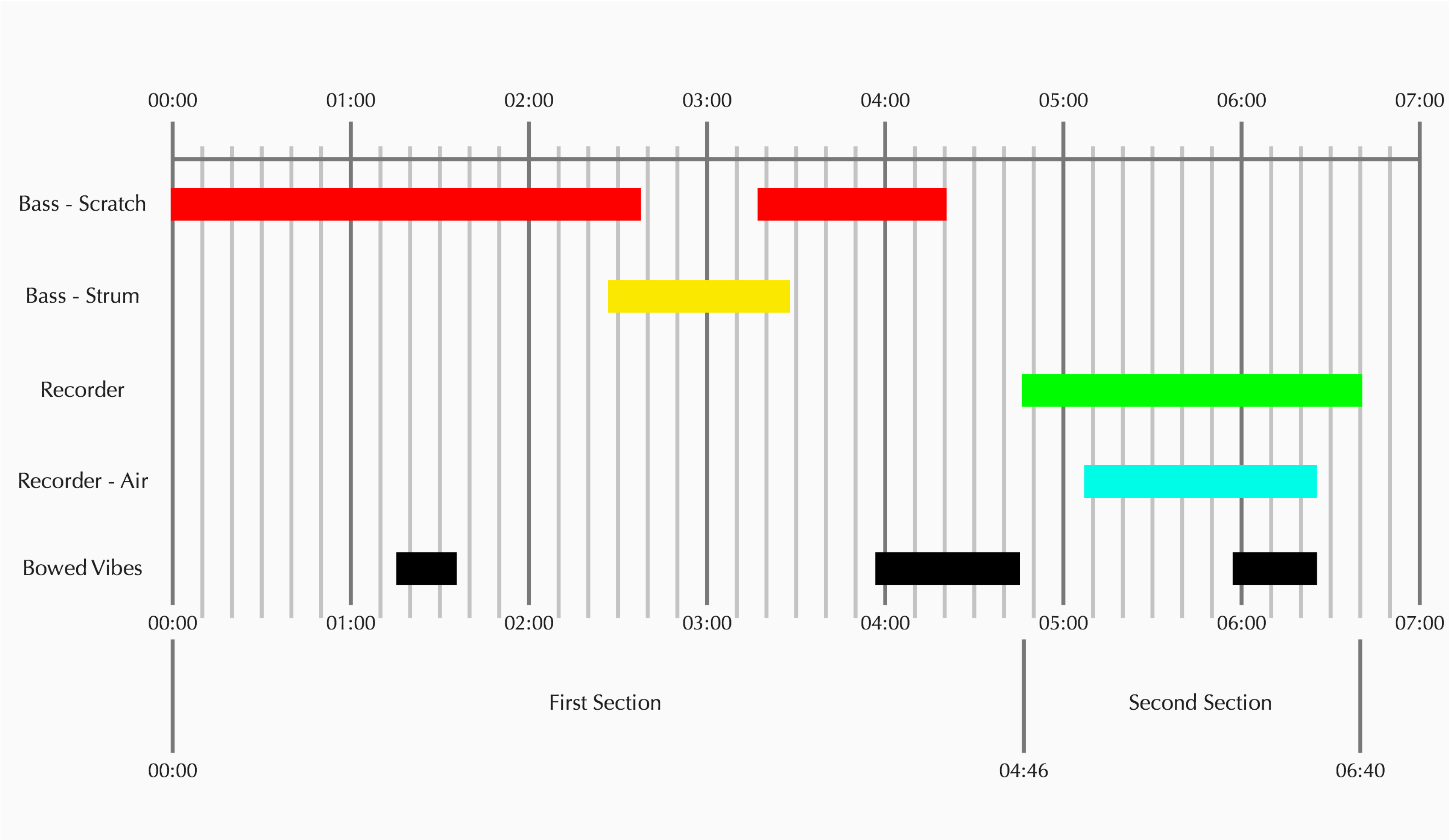

Figure 4.15: Layers of sound material in *Flux*

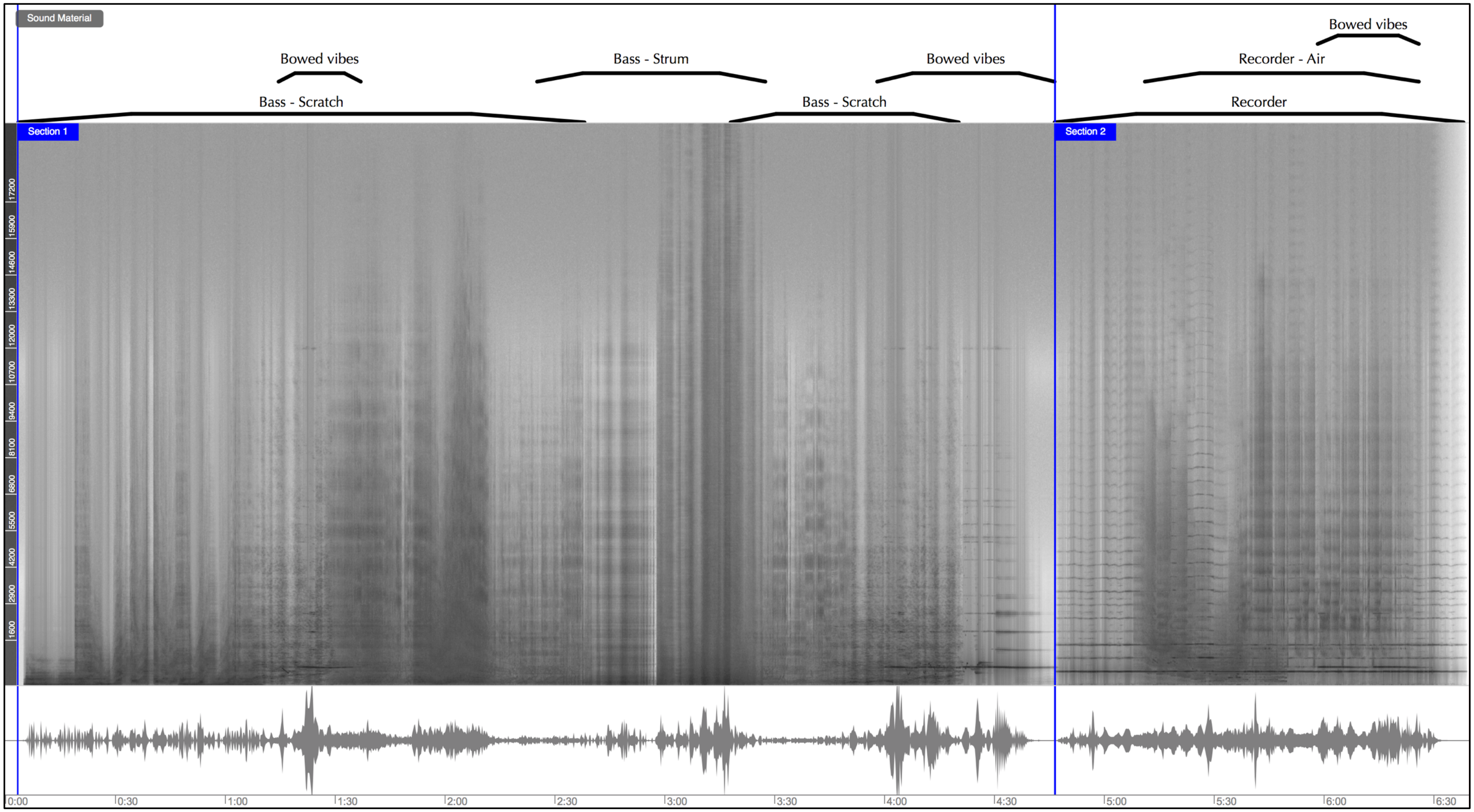

Figure 4.16: Sonogram and waveform of *Flux*

Chapter 5: Conclusion

This thesis has introduced *Sonic Matter*, its design, and the pieces that were composed and performed with it. The concept of the thesis was to create a system that would make the creative process faster and easier for myself. I hoped to get compelling sonic results without spending too much effort on technicalities, and just focus on the artistic aspect of making music. As the system came along, I realized more possibilities were opened up since *Sonic Matter* can facilitate electroacoustic music improvisation, performance and composition all at once. I believe that I arrived at a satisfying level in terms of the system's contribution in making music, so that I feel encouraged to go on stage and perform my music with it.

There are numerous trade-offs when it comes to designing an instrument. Regardless of what the decision is, it is always limiting in some way. I can certainly say that I am glad to choose *Max* as my programming environment since I shifted my path slightly during the design. *Max* has proven itself in facilitating a lot of design ideas and being as flexible as possible. However, I learned that it is extremely important to foresee the desired result sooner than later to minimize the energy spent.

While designing *Sonic Matter*, I saw how important the feedback was to make a meaningful connection with the system in a performance. For this reason, I spent most of my time thinking of ways to ameliorate this connection. In the quest to give the audience a deeper understanding of the work, I projected the 'cockpit' of the system on a screen during the première of *Sonic Matters*. I performed the first three pieces at the center of the room amongst the audience, and for the last two pieces, I asked the audience to close their eyes.

After the concert, I received various responses. Some claimed that it was engaging to see the 'guts' of the system, whereas some found the projection rather distracting. To evaluate the feedback, I performed *The Passage* at the *MINT* conference in a large hall without the visual feedback (MINT). The majority of the audience members said that they closed their eyes during the performance and felt immersed in the sound. Further performances might suggest the same or prove it wrong; regardless, this is a 'matter' to work on in the future.

I designed *Sonic Matter* because I desired to have an immediate connection with the music I create. However, I realized that I first had to change my perspective, and then bring the system to a level that I can comfortably perform music. Looking from a designer/creator point of view, the path to the artistic creation was way more complicated: I was making the tools to make music rather than just making music. Accordingly, my design decisions have had a more significant impact on my music, and the time I spent on the design paid off by accelerating the creative process. Everything I create with *Sonic Matter* is technically possible with other existing software and systems; however, *Sonic Matter* is the fastest and the most convenient for me.

5.1 Future Work

I designed *Sonic Matter* for myself as a system to perform and compose. As long as I keep using the system for making music, it will be developed further.

Firstly, I intend to make more modules to broaden my horizons in artistic creation. These modules include effects, sound processing components, and sound generation units. With the features introduced in *Max 8*, I will replace some of my modules with their more CPU-efficient versions. *Max 8* also enables the user to map controls very easily. Accordingly, displaying all the possible controls on the higher levels of the GUI might render the system more accessible and flexible for the user.

The design process is an ongoing cycle that continually updates and improves the system to maintain *Sonic Matter* as a good choice to make music. For that reason, I would like to go in new directions to extend software capabilities and discover how *Sonic Matter* can work with other input devices to keep up with the latest technologies.

Bibliography

Batchelor, Peter. "Acousmatic Approaches to the Construction of Image and Space in Sound Art." *Organised Sound*, vol. 20, no. 2, 2015, pp. 148–59, doi:10.1017/S1355771815000035.

Bingöl, A. Semih, and Nedim Karaca. *Spectral Analysis of Turkish Vowels and a Comparison of Vowel Normalization Algorithms BT - Speech Recognition and Coding*. Edited by Antonio J Rubio Ayuso and Juan M López Soler, Springer Berlin Heidelberg, 1995, pp. 317–20.

Birnbaum, David, et al. "Towards a Dimension Space for Musical Artifacts." *Proceedings of the 2005 International Conference on New Interfaces for Musical Expression (NIME05)*, 2005, pp. 192–95, doi:10.1393/ncr/i2015-10110-4.

Cahill, Thaddeus. *Art of and Apparatus for Generating and Distributing Music*. US580035A, US Patent Office, 1897.

Chadabe, Joel. *Electric Sound : The Past and Promise of Electronic Music*. Upper Saddle River, N.J. : Prentice Hall, 1997.

Collins, Nick, et al. *Electronic Music*. Cambridge : Cambridge University Press, 2013, doi:10.1017/CBO9780511820540.

Collins, Nick (Nicholas). *Introduction to Computer Music*. Hoboken : John Wiley & Sons, 2010.

Csound. *Introduction*. 2018, https://csound.com/docs/manual/Introduction.html.

---. *Real-Time Audio*. 2018, https://csound.com/docs/manual/UsingRealTime.html.

Cycling '74. *Max 7 Features | Cycling '74*. 2018, https://cycling74.com/products/max-features.

---. *Max Software Tools for Media | Cycling '74*. 2018, https://cycling74.com/products/max/.

---. *Mira IPad App for Max | Cycling '74*. 2018, https://cycling74.com/products/mira.

Dean, Roger T. *Hyperimprovisation: Computer-Interactive Sound Improvisation*. Middleton, WI : A-R Editions, 2003.

Dudas, Richard. "'Comprovisation': The Various Facets of Composed Improvisation within Interactive Performance Systems." *Leonardo Music Journal*, vol. 20, The MIT Press, 2010, pp. 29–31, http://www.jstor.org/stable/40926370.

Holmes, Thom. *Electronic and Experimental Music: Technology, Music, and Culture*. 3rd Ed., New York : Routledge, 2008.

Hugill, Andrew. *The Digital Musician*. New York : Routledge, 2008.

ICMA. *International Computer Music Association - ICMC*. 2018, http://www.computermusic.org/page/23/.

ICST. *Downloads: Ambisonics Externals for MaxMSP | ZHdK.Ch*. 2018, https://www.zhdk.ch/downloads-ambisonics-externals-for-maxmsp-5381.

Jordá, Sergi. "The Reactable : Tangible and Tabletop Music Performance." *CHI 2010: Media Showcase Session 1*, 2010, pp. 2989–94, doi:http://doi.acm.org/10.1145/1753846.1753903.

Jordà, Sergi. "Interactivity and Live Computer Music." *The Cambridge Companion to Electronic Music*, Cambridge, U.K., 2007, pp. 89–106.

Kimura, Mari. "Performance Practice in Computer Music." *Computer Music Journal*, vol. 19, no. 1, 1995, p. 64, doi:10.2307/3681300.

KORG Inc. *NanoKONTROL2 - SLIM-LINE USB CONTROLLER | KORG (Canada - EN)*. 2018, https://www.korg.com/caen/products/computergear/nanokontrol2/.

Leap Motion, Inc. *Leap Motion*. 2018, https://www.leapmotion.com/.

Leman, Marc. *Embodied Music Cognition And Mediation Technology*. Cambridge, Mass. : MIT Press, 2008.

Malloch, Joseph, and Marcelo M. Wanderley. "The T-Stick: From Musical Interface to Musical Instrument." *NIME '07 Proceedings of the 7th International Conference on New Interfaces for Musical Expression*, 2007, p. 66, doi:10.1145/1279740.1279751.

Manning, Peter. *Electronic and Computer Music*. 4th ed.., New York : Oxford University Press, 2013.

Mathews, Max. *Radio-Baton Instruction Manual*. Preliminar, MARMAX, 2000.

Maxuino. *Maxuino » About*. 2018, http://www.maxuino.org/about.

McLuhan, Marshall. "Understanding Media: The Extensions of Man." *Extensions of Man*, 2nd ed.., New York : New American Library, 1964.

Microsoft Corporation. "Human Interface Guidelines." *Microsoft Corporation*, 2012, p. 70.

Millard, Andre. "The Inventors." *America on Record: A History of Recorded Sound*, edited by Andre Millard, 2nd ed., Cambridge University Press, 2005, pp. 17–36, doi:DOI: 10.1017/CBO9780511800566.005.

MINT. *MINT – Music in New Techonologies*. 2018, http://mintconference.ca/.

Miranda, Eduardo Reck. *New Digital Musical Instruments: Control and Interaction Beyond the Keyboard*. Edited by Marcelo M Wanderley et al., Middleton, Wis. : A-R Editions, 2006.

Moore, Adrian J. “Cross-Modality in Multi-Channel Acousmatic Music: The Physical and Virtual in Music Where There Is ‘Nothing to See.’” *The International Computer Music Conference*, 2015, pp. 186–89.

NIME. *NIME | International Conference on New Interfaces for Musical Expression*. 2018, http://www.nime.org/.

Nishimura, Ryouichi, and Kotaro Sonoda. “B-Format for Binaural Listening of Higher Order Ambisonics.” *The Journal of the Acoustical Society of America*, vol. 133, 2013, doi:10.1121/1.4805446.

Park, Tae. “A Tribute to Max Mathews: An Interview with Max Mathews.” *Computer Music Journal*, vol. 33, no. 3, 2009, pp. 9–22.

Puckette, Miller. “Combining Event and Signal Processing in the MAX Graphical Programming Environment.” *Computer Music Journal*, vol. 15, no. 3, 1991, pp. 68–77, doi:10.2307/3680767.

---. “Max at Seventeen.” *Computer Music Journal*, vol. 26, no. 4, 2002, pp. 31–43, doi:10.1162/014892602320991356.

---. “Pure Data: Another Integrated Computer Music Environment.” *Proceedings, Second Intercollege Computer Music Concerts*, 1997, pp. 37–41, doi:10.1017/CBO9781107415324.004.

Rabenstein, Rudolf, and Sascha Spors. “Sound Field Reproduction.” *Springer Handbook of Speech Processing*, edited by Jacob Benesty et al., Springer Berlin Heidelberg, 2008, pp. 1095–114, doi:10.1007/978-3-540-49127-9_53.

Roads, Curtis. *The Computer Music Tutorial*. Cambridge, Mass. : MIT Press, 1996.

Roads, Curtis, and John Strawn. “Digital Audio Concepts.” *The Computer Music Tutorial*, Cambridge, Mass. : MIT Press, 1996, pp. 5–47.

---. “Introduction to Digital Sound Synthesis.” *The Computer Music Tutorial*, Cambridge, Mass. : MIT Press, 1996, pp. 85–114.

Rossing, Thomas D. “Springer Handbook of Acoustics.” *Springer Handbook of Acoustics*, edited by Thomas D Rossing, New York, NY : Springer New York, 2007.

Rovan, Joseph Butch, et al. “Instrumental Gestural Mapping Strategies as Expressivity Determinants in Computer Music Performance.” *KANSEI: The Technology of Emotion : AIMI International Workshop : Proceedings*, 1997, pp. 68–73.

Rowe, Robert. “Interactive Improvisation.” *Machine Musicianship*, MIT Press, 2001, pp. 277–315.

Soydan, Abdullah Safa. *Delays - Improvisation - YouTube*. 2018, https://www.youtube.com/watch?v=-we9TwnJ0bg.

---. *Granulate - Demonstration - YouTube*. 2018, https://www.youtube.com/watch?v=8c6Pa8yrWRM.

---. *MINT 2018 - Abdullah Soydan on Vimeo*. 2018, https://vimeo.com/303901554.

---. *Sonic Matters by Abdullah Safa Soydan*. 2018, https://soundcloud.com/abdullah-safa-soydan/sets/sonic-matters.

STEIM. *STEIM | Studio for Electro-Instrumental Music*. 2018, http://steim.org/.

Terdiman, Daniel. *Leap Motion: 3D Hands-Free Motion Control, Unbound - CNET*. 2012, https://www.cnet.com/news/leap-motion-3d-hands-free-motion-control-unbound/.

The MIDI Association. *About MIDI-Part 3: MIDI Messages*. 2018, https://www.midi.org/articles-old/about-midi-part-3-midi-messages.

Torre, Giuseppe, et al. "The Hands: The Making of a Digital Musical Instrument." *Computer Music Journal*, vol. 40, no. 2, 2016, pp. 22–34, doi:10.1162/COMJ_a_00356.

Truax, Barry. *POD*. 2018, https://www.sfu.ca/~truax/pod.html.

---. "The POD System of Interactive Composition Programs." *Computer Music Journal*, vol. 1, no. 3, 1977, p. 30, doi:10.2307/3679607.

Vassilandonakis, Yiorgos. "An Interview with Trevor Wishart." *Computer Music Journal*, vol. 33, no. 2, 2009, pp. 8–23, doi:10.1162/comj.2009.33.2.8.

Vidolin, Alvise. "Musical Interpretation and Signal Processing." *Musical Signal Processing*, edited by Curtis Roads et al., Swets&Zeitlinger, 1997, pp. 439–57.

Wallis, Isaac, et al. "Amateur Musicians, Long-Term Engagement, and HCI." *Music and Human-Computer Interaction*, edited by Simon Holland et al., Springer London, London, 2013, pp. 49–66, doi:10.1007/978-1-4471-2990-5.

Wang, Ge. "A History of Programming and Music." *The Cambridge Companion to Electronic Music*, Cambridge, U.K., 2007, pp. 55–71.

Weale, Rob. *English Definition - EARS ElectroAcoustic Resources Site*. 2005, http://ears.pierrecouprie.fr/spip.php?article270.

Weeks, Alex. *Mbira Dzavadzimu*. 2006, https://upload.wikimedia.org/wikipedia/commons/5/5b/Mbira_dzavadzimu_1.jpg.

Weichert, Frank, et al. "Analysis of the Accuracy and Robustness of the Leap Motion Controller." *Sensors (Switzerland)*, vol. 13, no. 5, 2013, pp. 6380–93, doi:10.3390/s130506380.

Winkler, Todd. *Composing Interactive Music: Techniques and Ideas Using Max*. Cambridge, Mass. : MIT Press, 1998.

Wishart, Trevor. *Sonic Art Pieces*. 2018, http://www.trevorwishart.co.uk/rbfull.html.

---. *The Sound Loom*. 2018, http://www.trevorwishart.co.uk/slfull.html.

Appendix A: Complete List of Mapped Controls

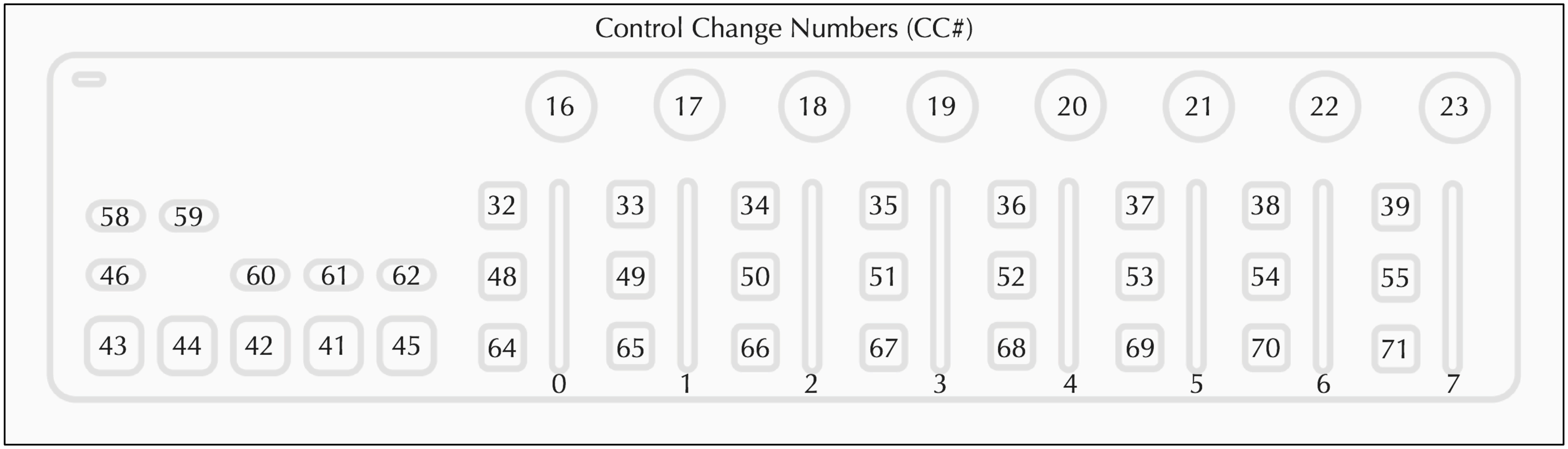

Figure A.1: KORG nanoKONTROL2 - Default CC numbers

CC#	Type	Behaviour	Module	Controlled Parameter	MIDI Value	Control Value
0	Slider	Continous control	*SM_Groove*	The location of the loop end marker	0 - 127	50ms - Maximum loop length
1	Slider	Continous control	*SM_Groove*	Envelope of each instance - release duration	0 - 127	5 – 3000 ms
2	Slider	Continous control	*SM_EnvelopeFunction*	Total envelope length	0 - 127	50ms - Selected loop length

3	Slider	Continous control	*SM_Delay*	Feedback amount	0 - 127	0 - 0.9
4	Slider	Continous control	*SM_Delay2*	Delay time	0 - 25	0 - 100 ms
					25 - 110	100 - 1000 ms
					110 - 127	1000 – 2000 ms
5	Slider	Continous control	*SM_Polybuffer*	Envelope of each instance - release duration	0 - 25	5 - 1000 ms
					25 - 110	1000 - 4000 ms
					110 - 127	4000 - 8000 ms
6	Slider	Continous control	see CC#38-54-70	Output level of the selected module	0 - 127	-inf - 0 dBFS
7	Slider	Continous control	*SM_Reverb*	Reverb decay time	0 - 127	0ms - infinite
16	Dial	Continous control	*SM_Groove*	The location of the loop start marker	0 - 127	Audio sample start - end
17	Dial	Continous control	*SM_Groove*	Playback speed	0 - 63	0.1x - 1x
					63 - 90	1x - 2x
					90 - 127	2x - 8x
18	Dial	Continous control	*SM_AM*	AM rate	0 - 25	0 - 1 Hz
					25 - 64	1 - 20 Hz
					64 - 127	20 - 1000 Hz
19	Dial	Continous control	*SM_Delay1*	Delay time	0 - 25	0 - 100 ms
					25 - 110	100 - 1000 ms
					110 - 127	1000 – 2000 ms
20	Dial	Continous control	*SM_SamplePlayer1*	Playback speed	0 - 63	0.1x - 1x
					63 - 90	1x - 2x
					90 - 127	2x - 8x
21	Dial	Continous control	*SM_Polybuffer*	Triggering frequency	0 - 110	50 - 1000 ms
					110 - 127	1000 - 2000 ms
22	Dial	Continous control	*SM_Gestures*	Output level	0 - 127	-inf - 0 dBFS

23	Dial	Continous control	*SM_Ambipanning*	Trajectory speed	0 - 127	0.5x - 8x
32	Button	Momentary	*SM_Groove*	Next audio sample	N/A	
33	Button	Toggle	*SM_Groove*	Timestretching on/off	N/A	
37	Button	Toggle	*SM_Polybuffer*	Triggering on/off	N/A	
38	Button	Momentary	CC#6	Selects *SM_Groove* to be controlled by CC#6	N/A	
41	Button	Momentary	*SM_Groove*	Triggers the sound playback	N/A	
42	Button	Momentary	*SM_SamplePlayer2*	Triggers the sound playback	N/A	
43	Button	Toggle	*SM_AmbiPanning2*	Trajectory recording for *SM_SamplePlayer1*	N/A	
44	Button	Momentary	*SM_SamplePlayer1*	Triggers the sound playback	N/A	
45	Button	Toggle	*SM_AmbiPanning1*	Trajectory recording for *SM_Groove*	N/A	
46	Button	Toggle	*SM_Gestures*	Triggering on/off	N/A	
48	Button	Momentary	*SM_Groove*	Previous audio sample	N/A	

53	Button	Momentary	*SM_Polybuffer*	Fade in / Fade out	N/A
54	Button	Momentary	CC#6	Selects *SM_SamplePlayer1* to be controlled by CC#6	N/A
58	Button	Momentary	Cues	Selects previous cue	N/A
59	Button	Momentary	Cues	Selects next cue	N/A
60	Button	Momentary	*SM_Gestures*	Triggering frequency	N/A
61	Button	Momentary	*SM_Gestures*	Playback speed - down	N/A
62	Button	Momentary	*SM_Gestures*	Playback speed - up	N/A
64	Button	Momentary	*SM_Groove*	Activates the reverb send	N/A
68	Button	Momentary	*SM_SamplePlayer1*	Activates the reverb send	N/A
69	Button	Toggle	*SM_Polybuffer*	Timestretching on/off	N/A
70	Button	Momentary	CC#6	Selects *SM_SamplePlayer2* to be controlled by CC#6	N/A

Table A.1: Complete list of mapped controls

Appendix B: Accompanying Media and Software (DVD)

The DVD includes a folder containing all five electroacoustic music compositions presented in this thesis. Three of the recordings were taken during the première performance on June 14, 2018, in the Sonic Arts Lab at the University of Calgary, and the remaining two were taken right before the concert. The pieces were first recorded in 8-channels then mixed down to stereo. As well, all the audio files used in each piece can be found in their corresponding folder.

The DVD also contains a video of the author's performance at the *MINT* conference on November 18, 2018, just to demonstrate how the system is performed. The audio heard in the video is recorded with the camcorder's embedded microphone.

Another folder contains the software modules along with the required external libraries for *Max* that constitutes *Sonic Matter*.

Appendix C: *Sonic Matter* - *Max* Patchers

C.1 SM_AM

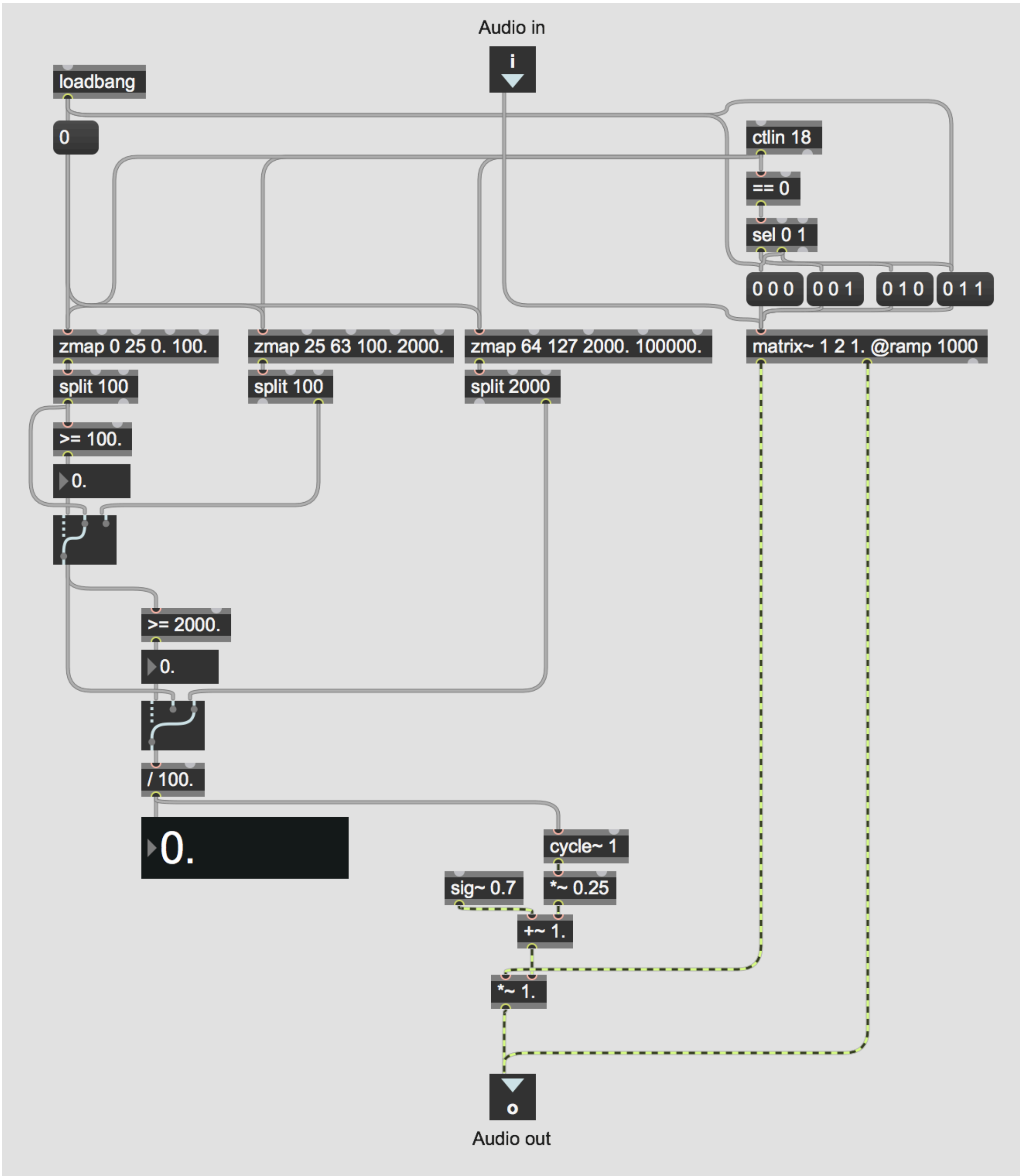

Figure C.1: SM_AM

C.2 SM_AmbiPanning

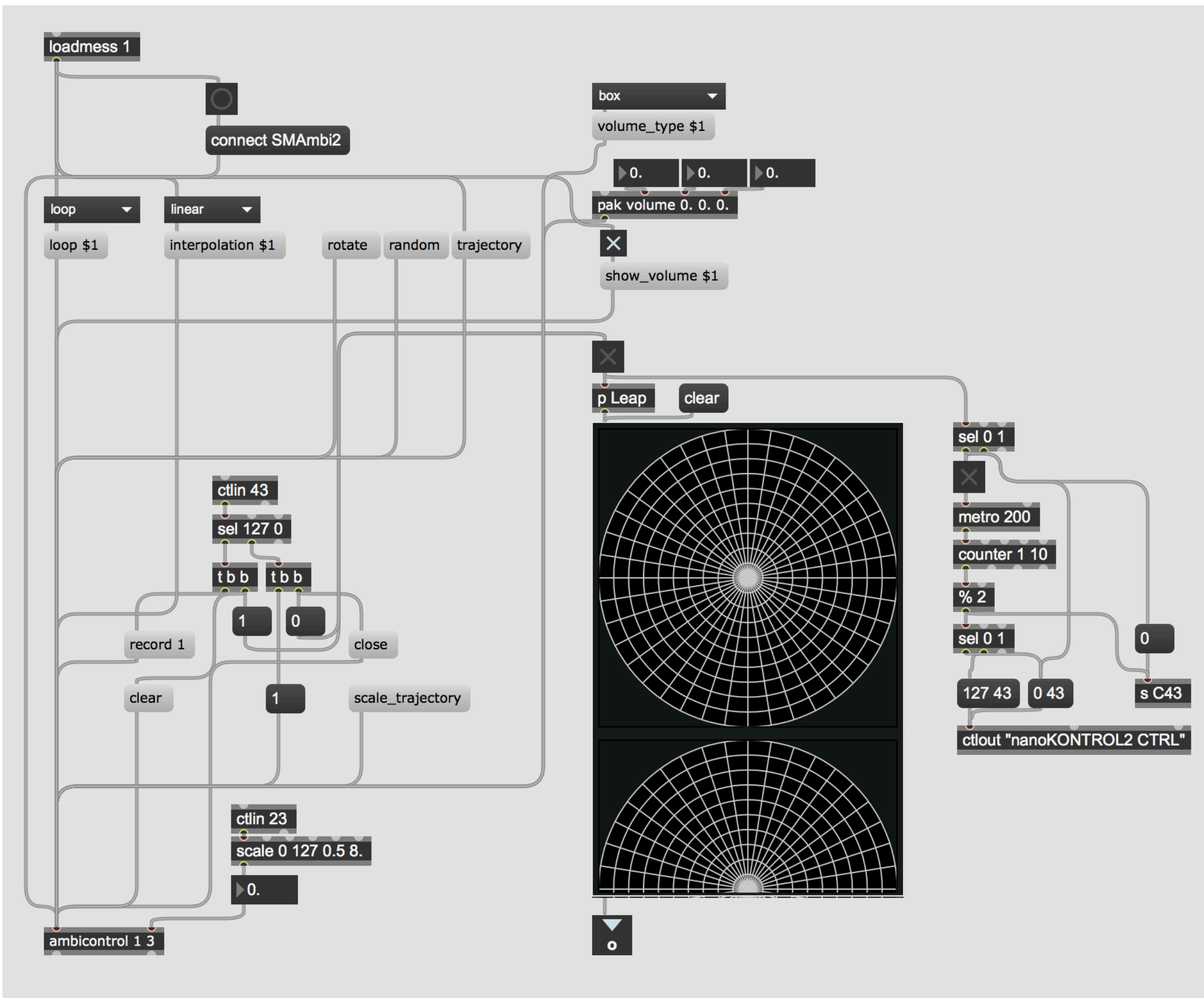

Figure C.2: SM_Ambipanning

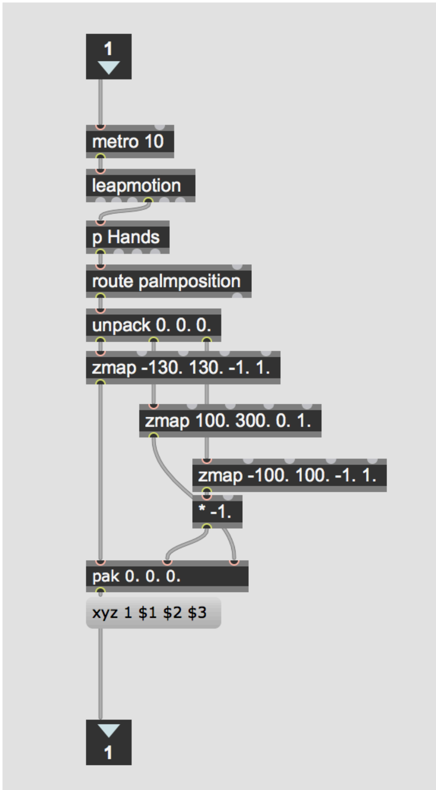

Figure C.3: SM_Ambipanning/Leap

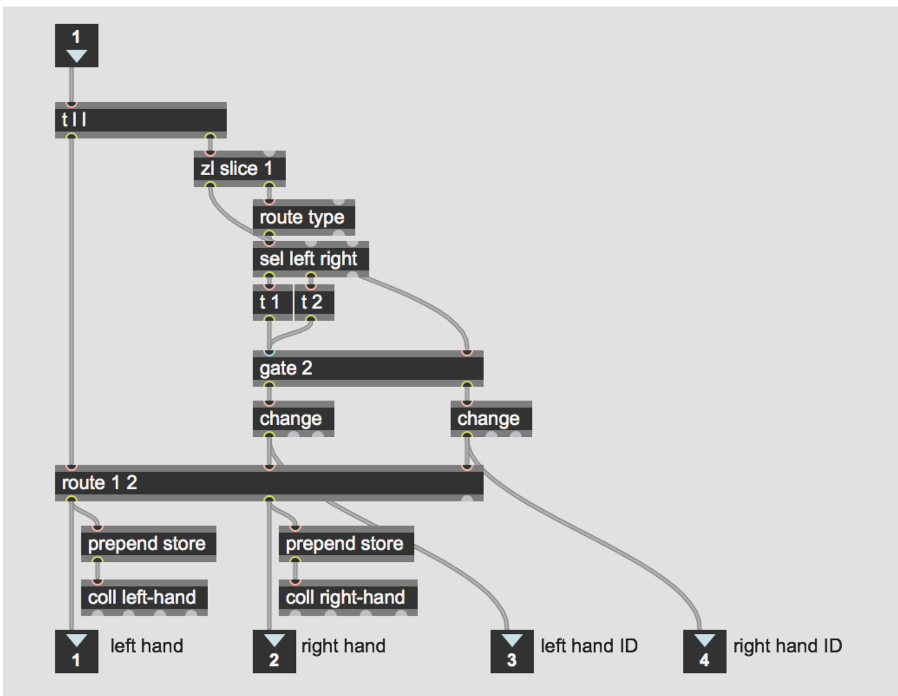

Figure C.4: SM_Ambipanning/Leap/Hands

C.3 SM_Bitcrush

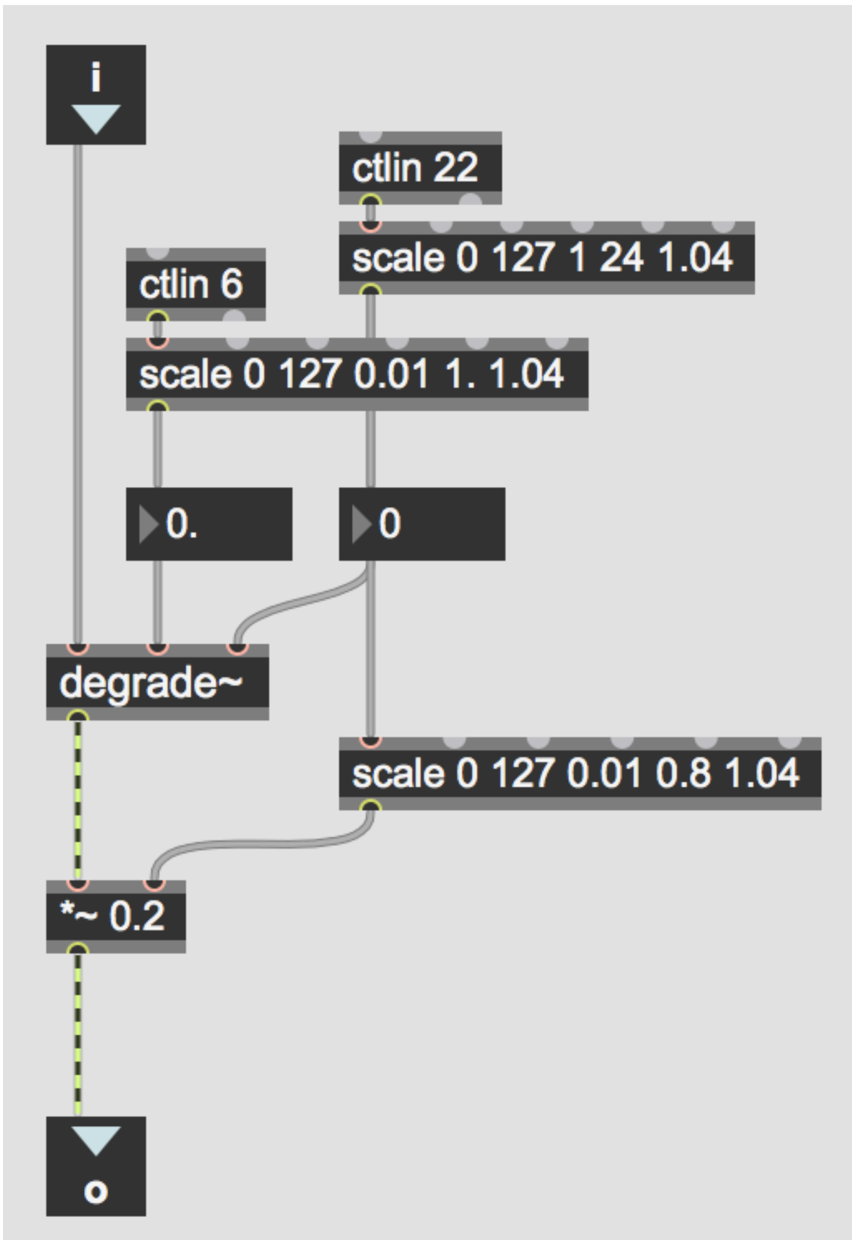

Figure C.5: SM_Bitcrush

C.4 SM_Delay

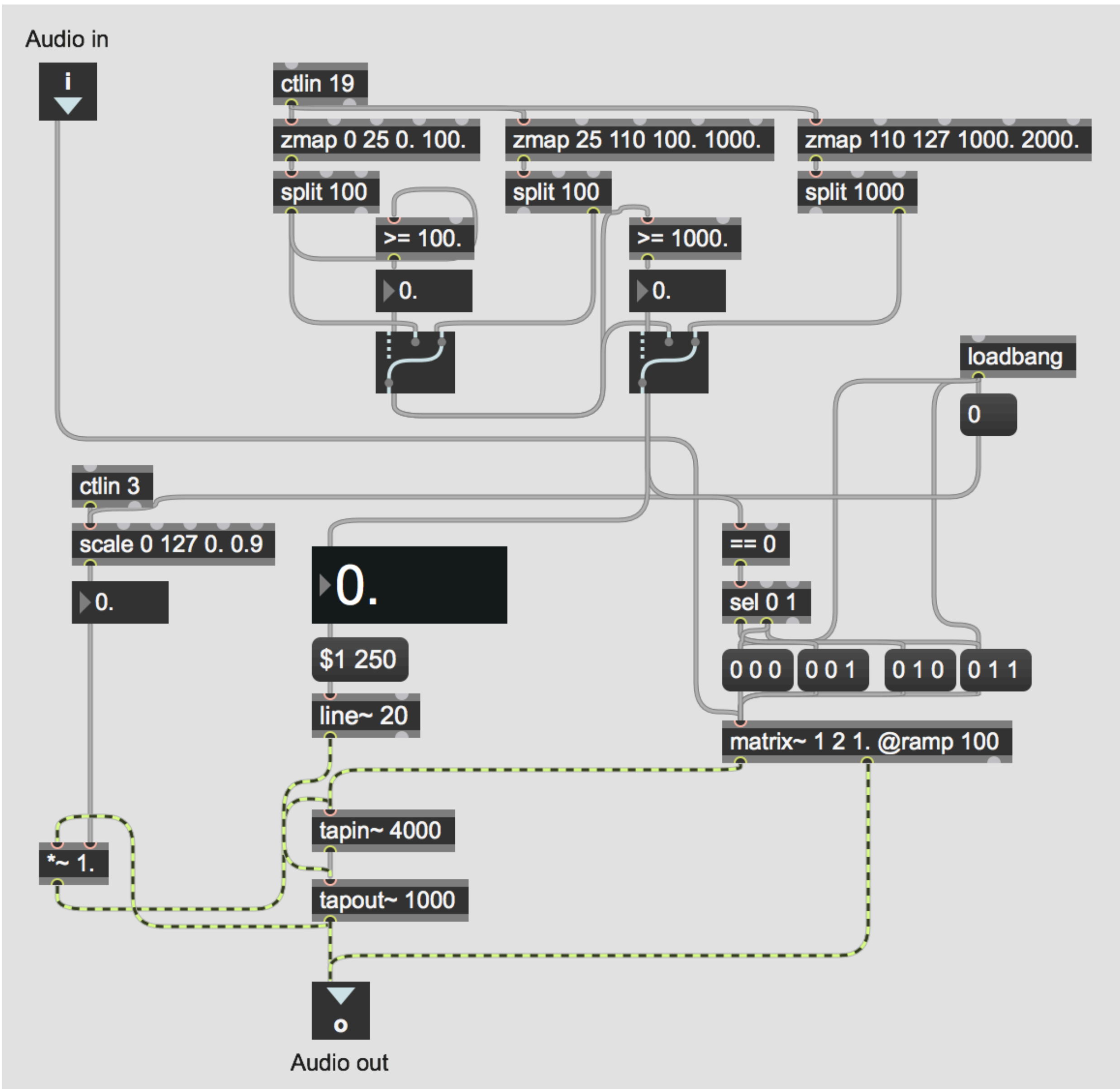

Figure C.6: SM_Delay

C.5 SM_EnvelopeFunction

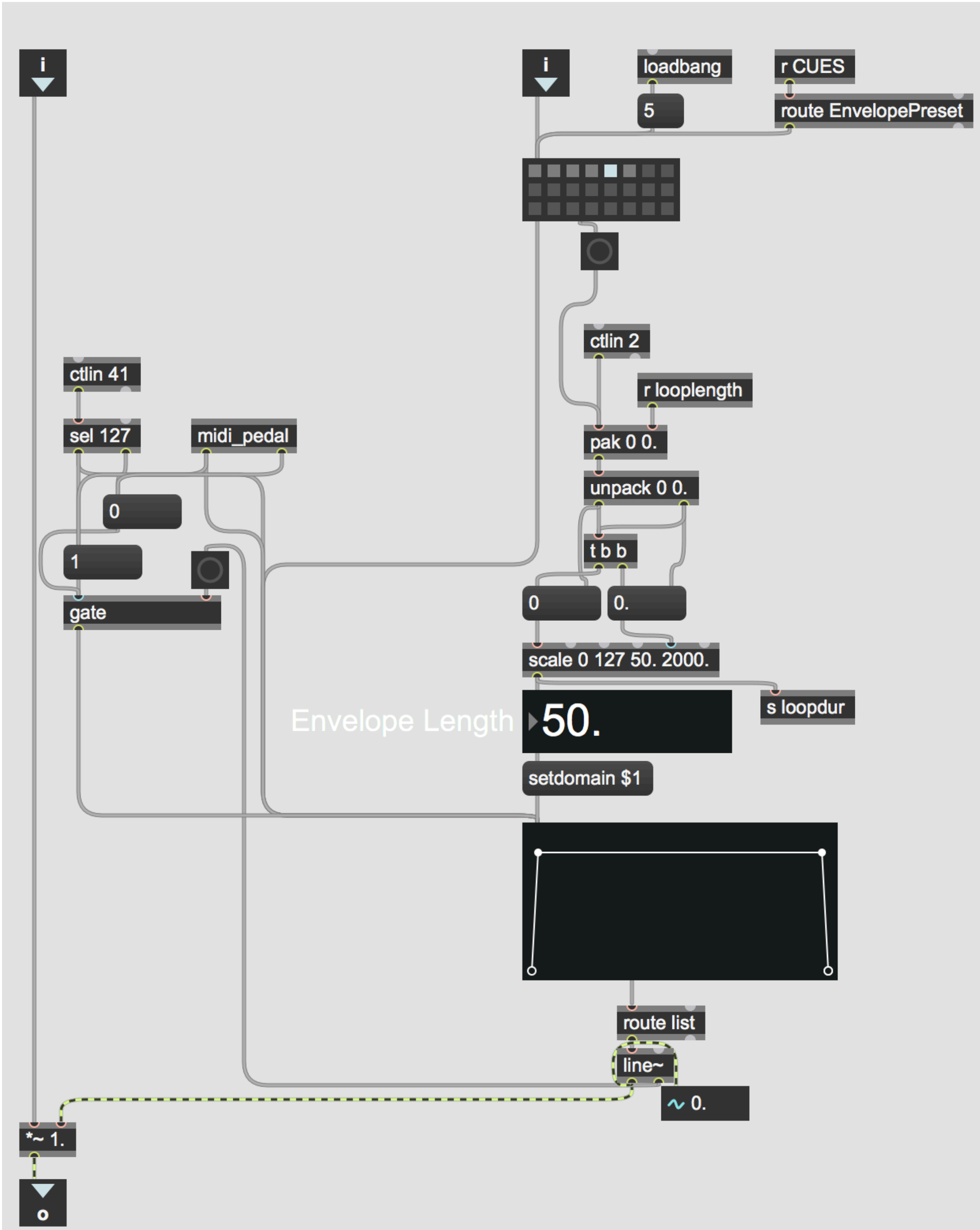

Figure C.7: SM_EnvelopeFunction

C.6 SM_Gesture

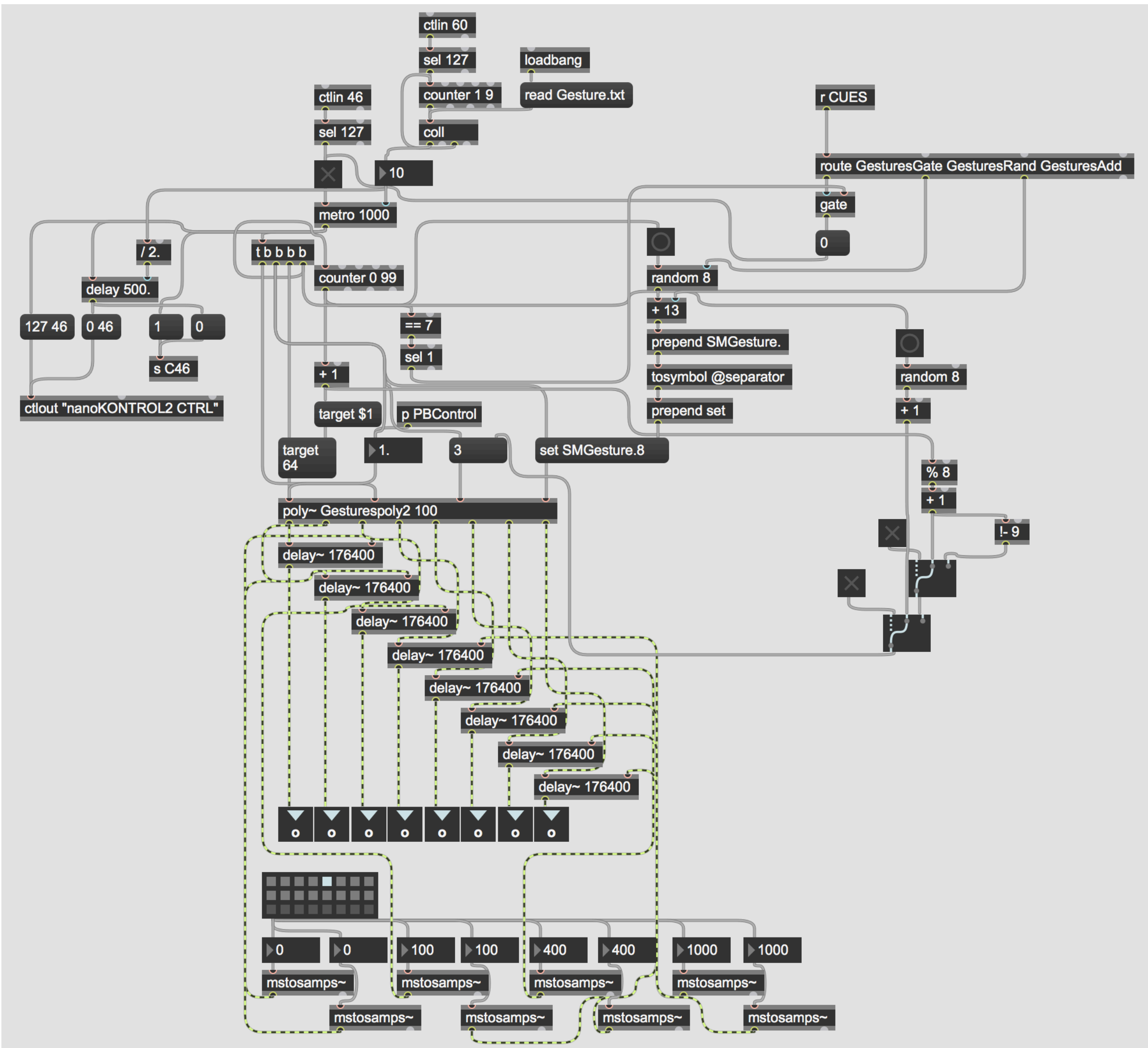

Figure C.8: SM_Gestures

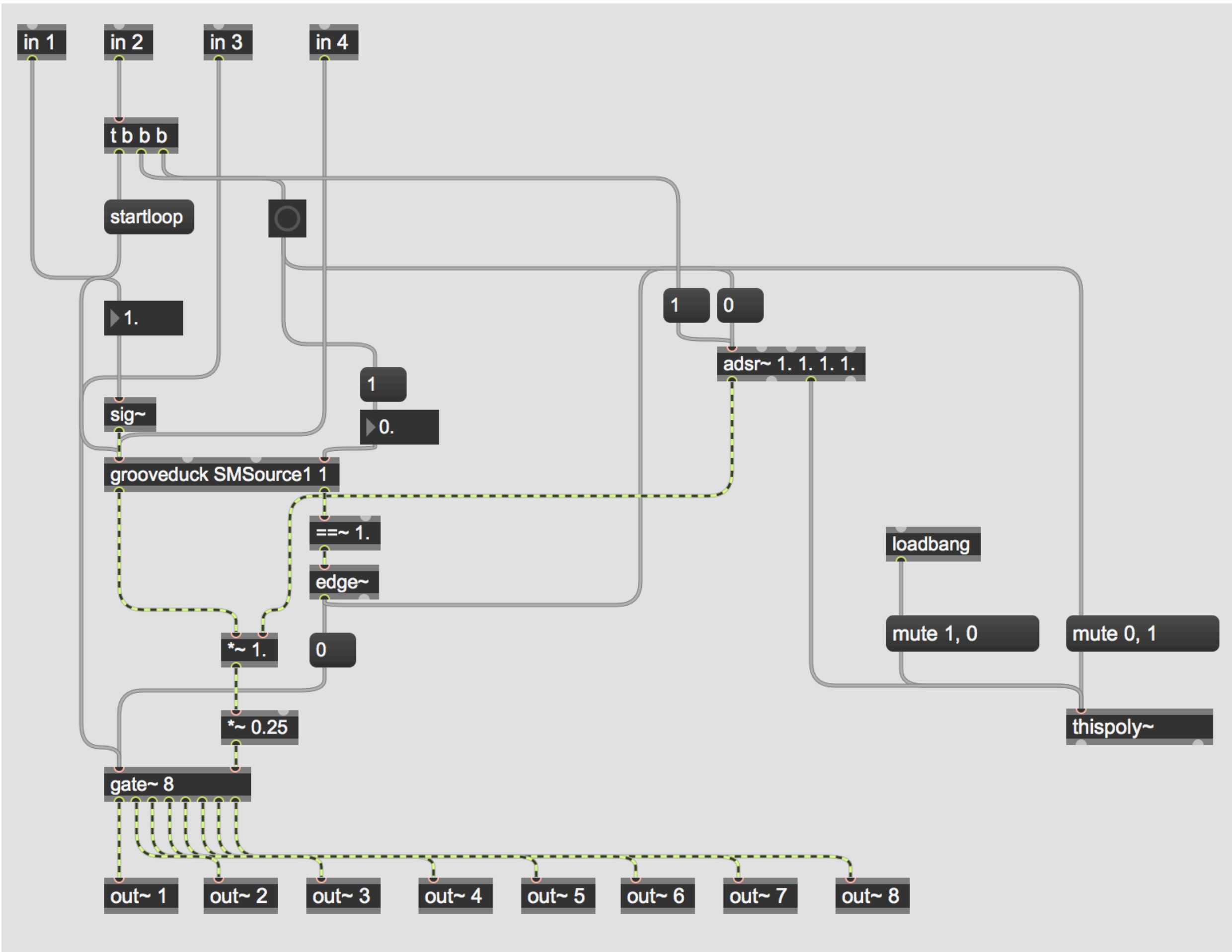

Figure C.9: SM_Gestures/Gesturespoly

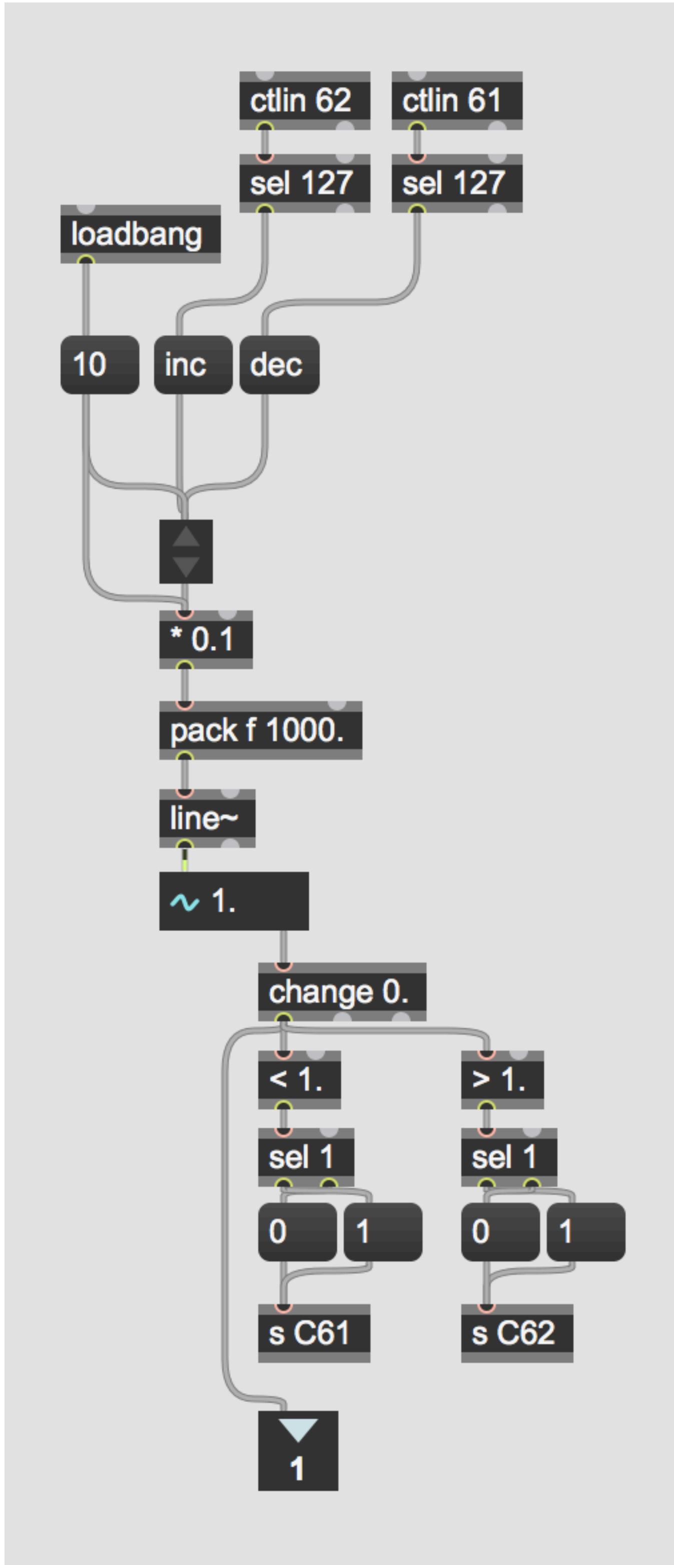

Figure C.10: SM_Gestures/PBControl

C.7 SM_Groove

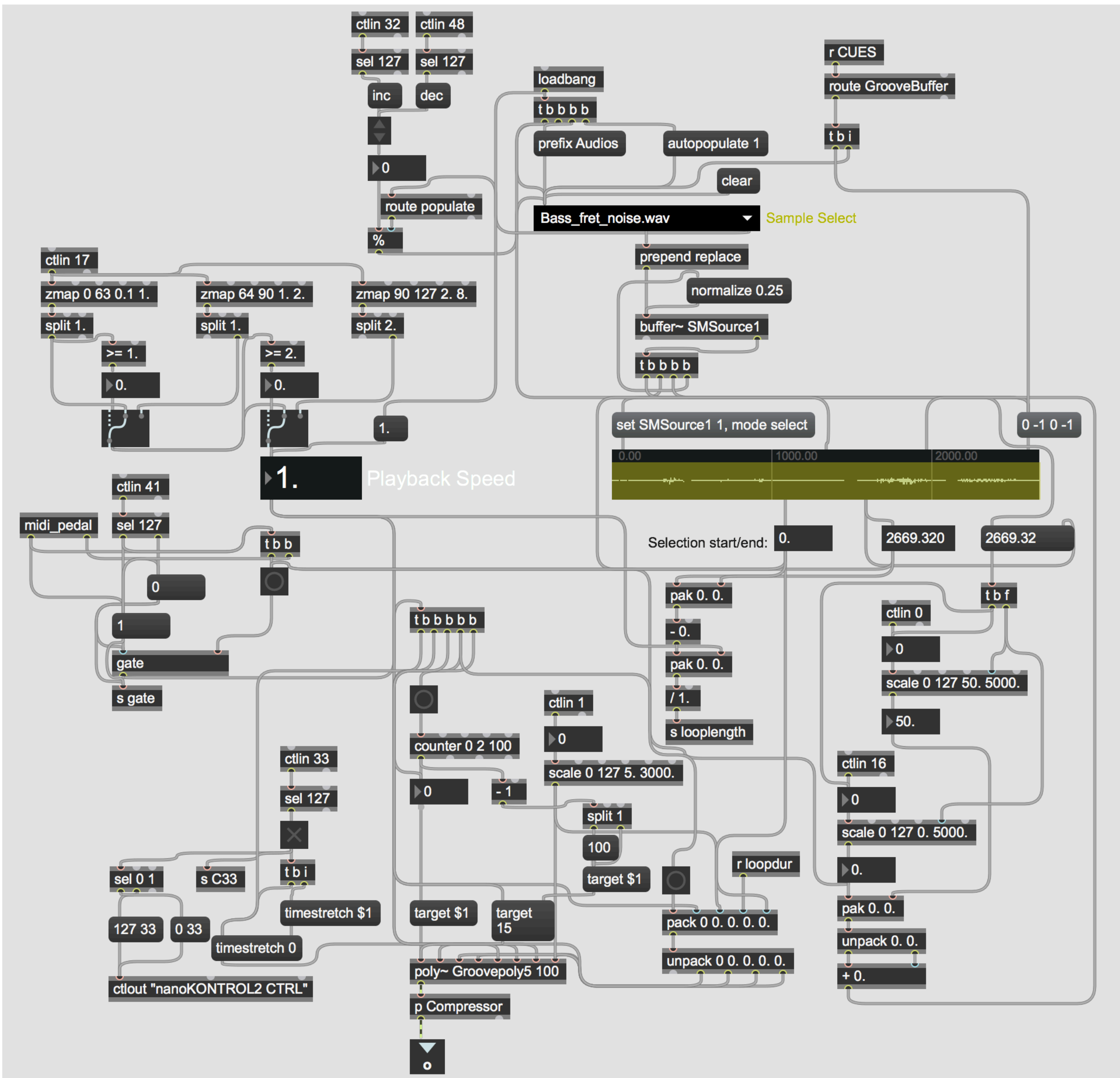

Figure C.11: SM_Groove

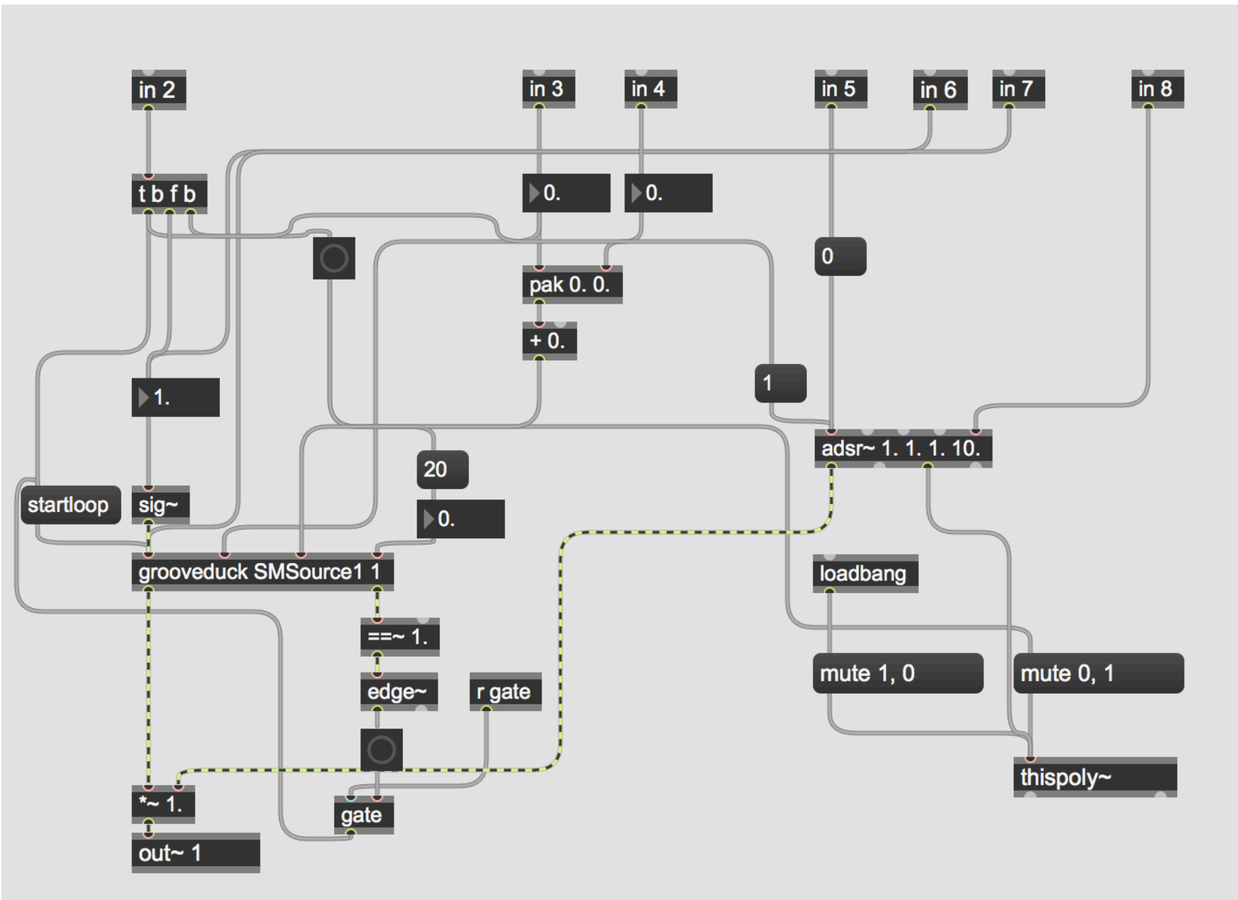

Figure C.12: SM_Groove/Groovepoly

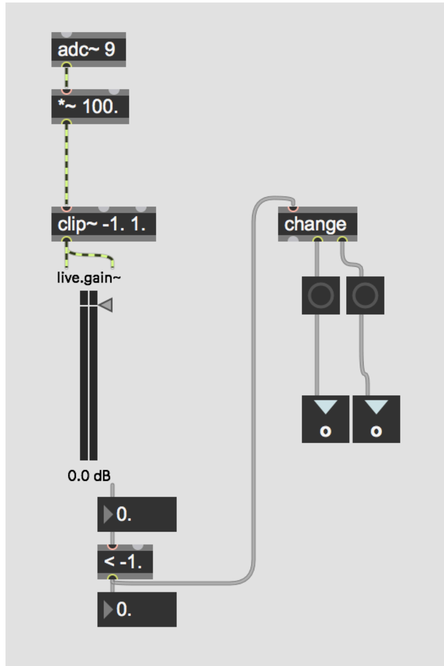

Figure C.13: SM_Groove/midi_pedal

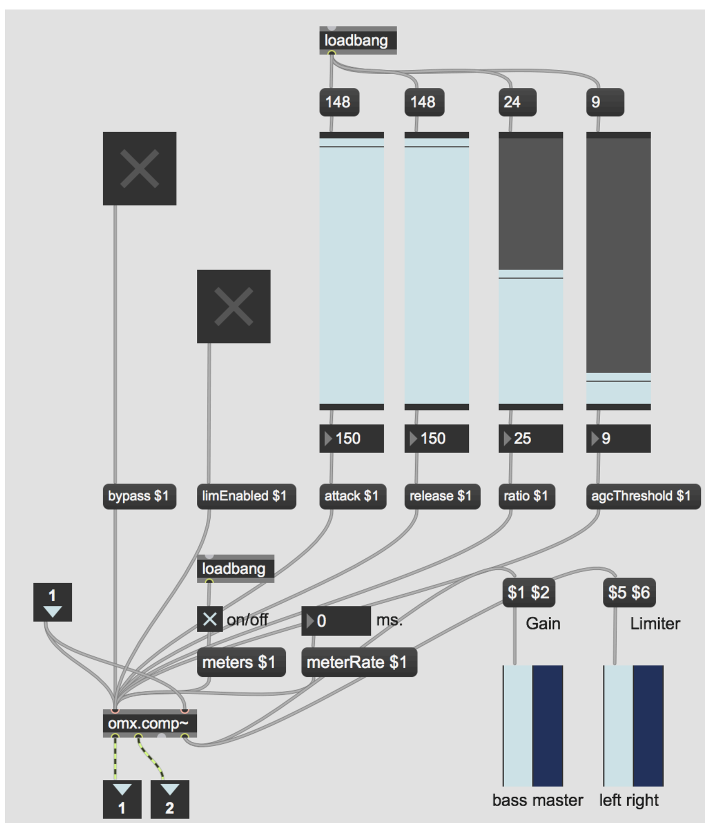

Figure C.14: SM_Groove/Compressor

C.8 SM_Overdrive

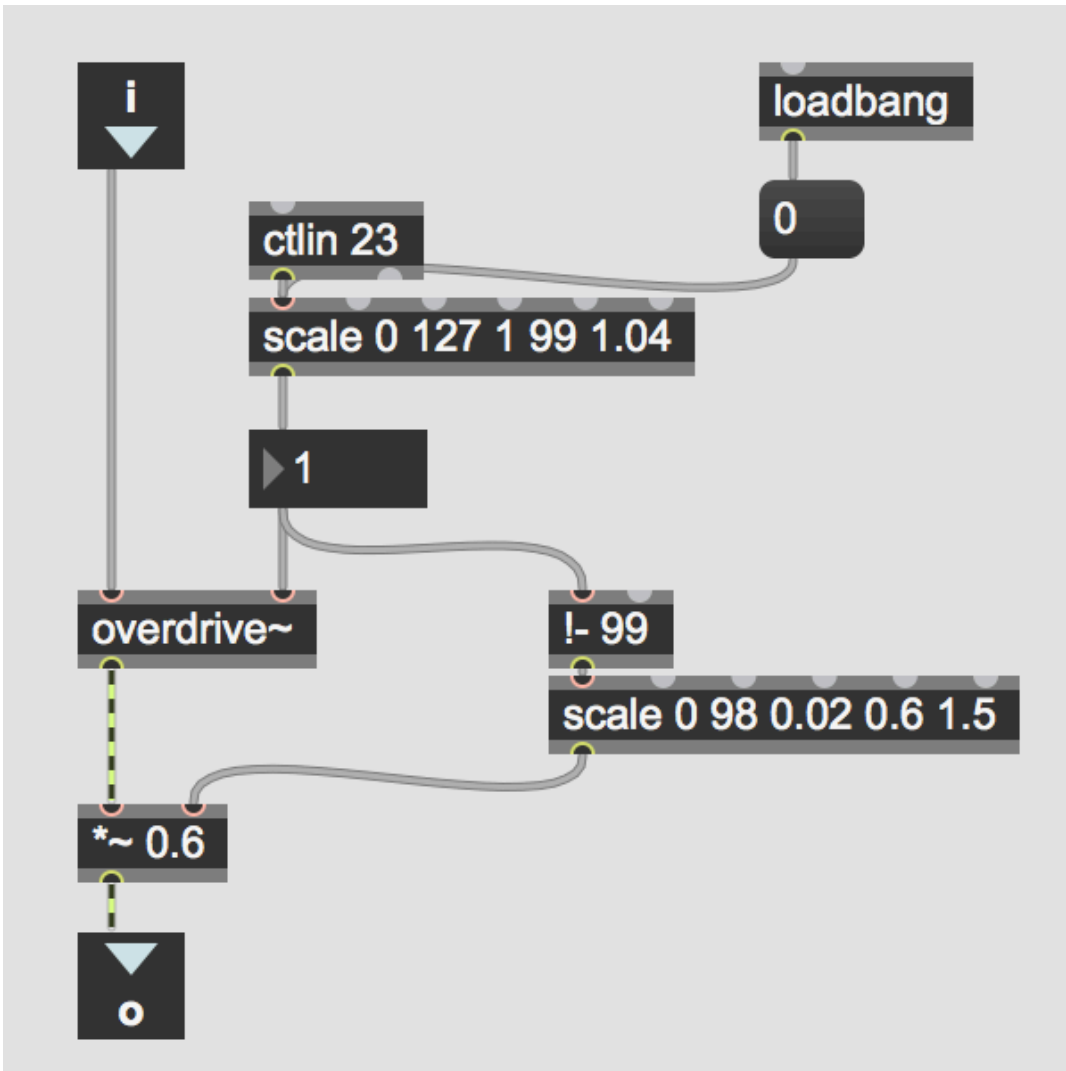

Figure C.15: SM_Overdrive

C.9 SM_Polybuffer

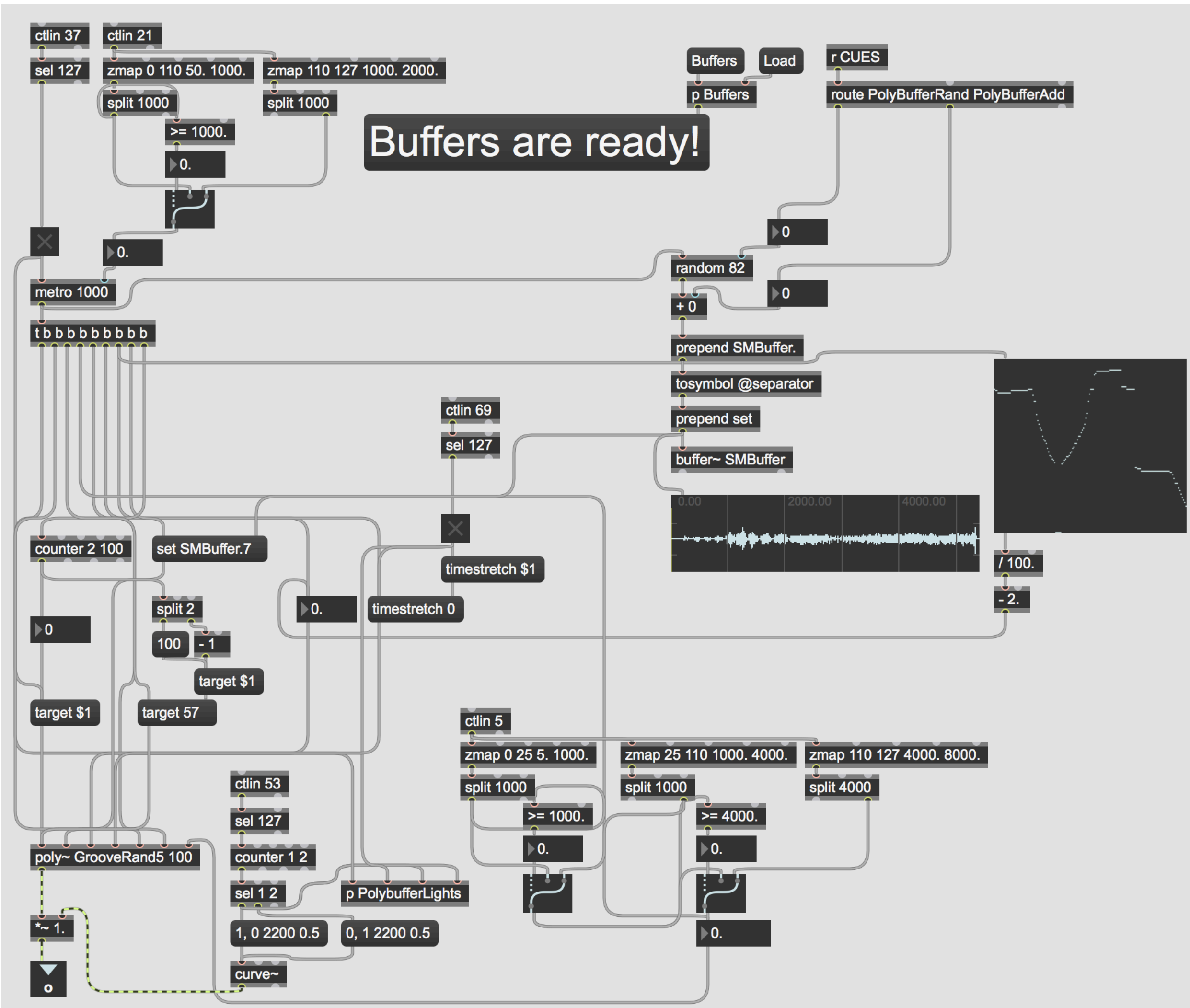

Figure C.16: SM_Polybuffer

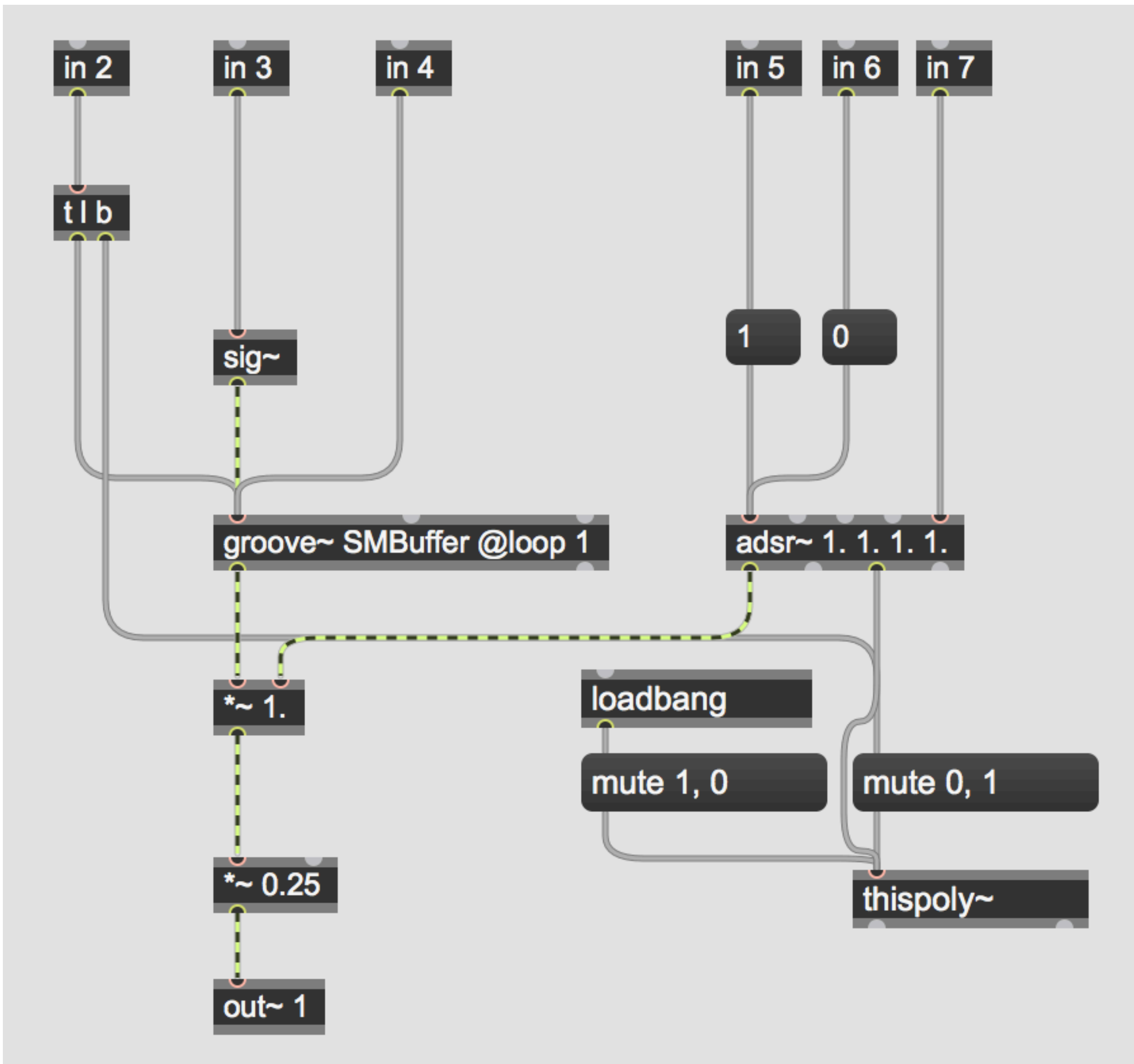

Figure C.17: SM_Polybuffer/GrooveRand

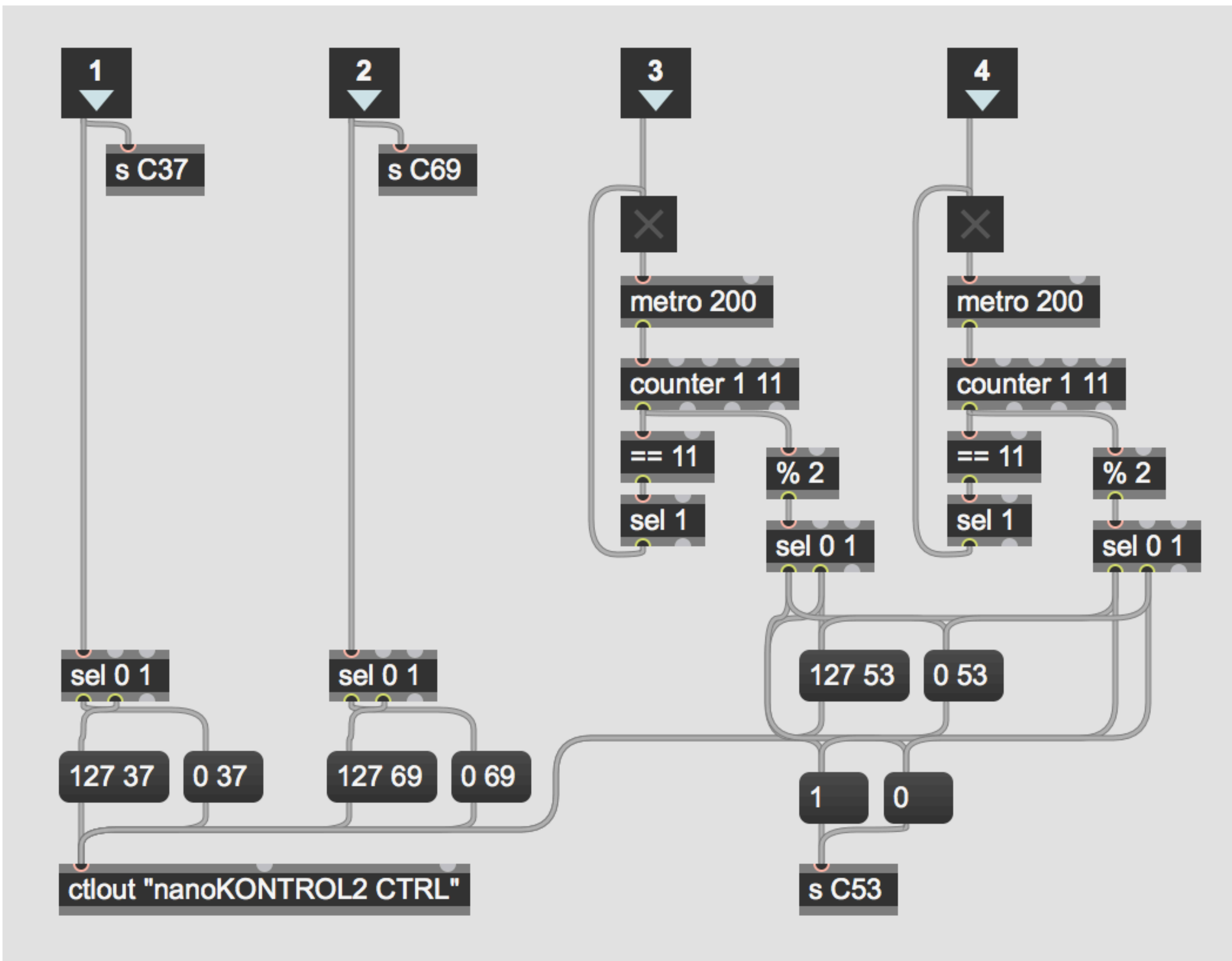

Figure C.18: SM_Polybuffer/PolybufferLights

C.10 SM_Reverb

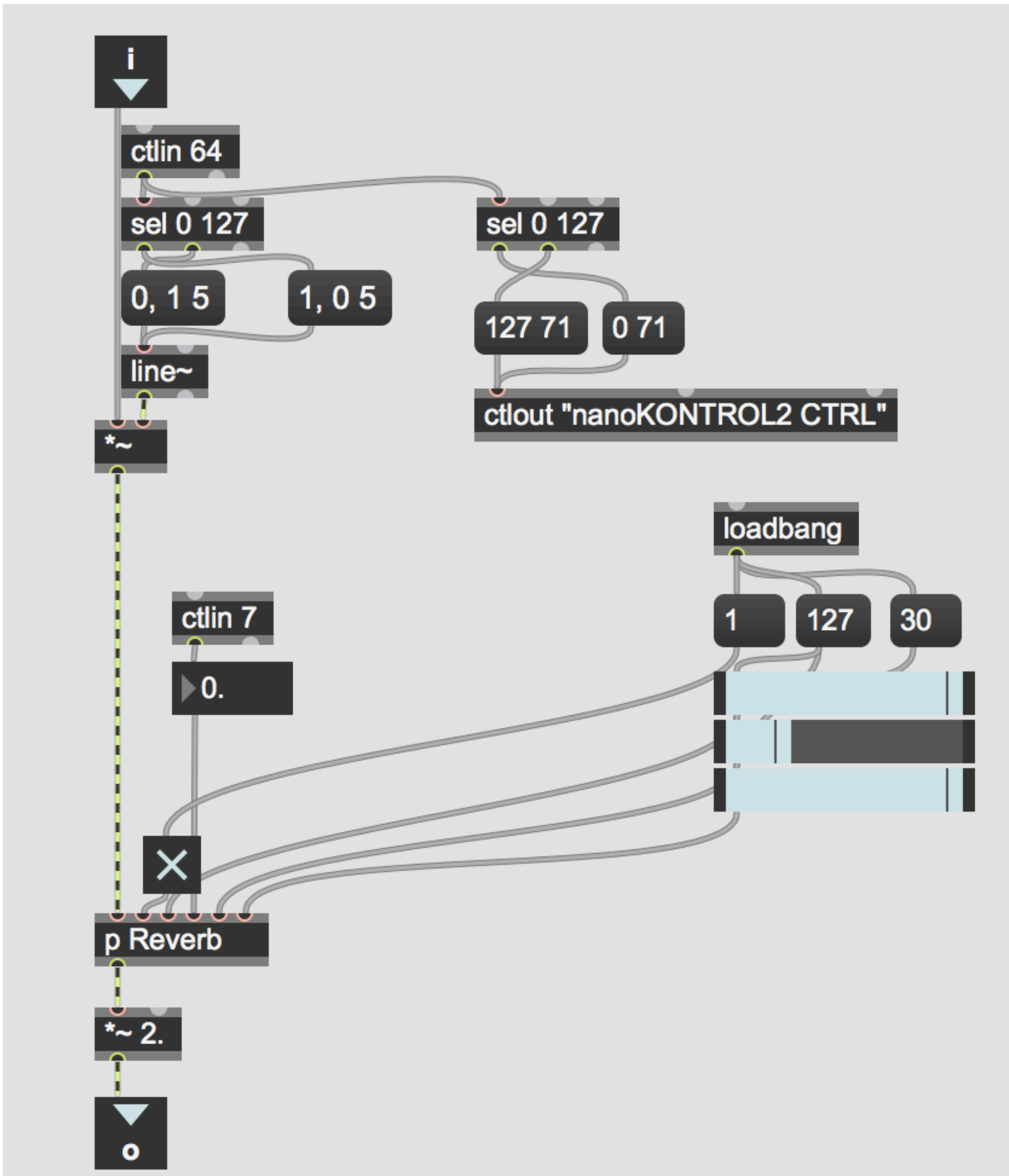

Figure C.19: SM_Reverb

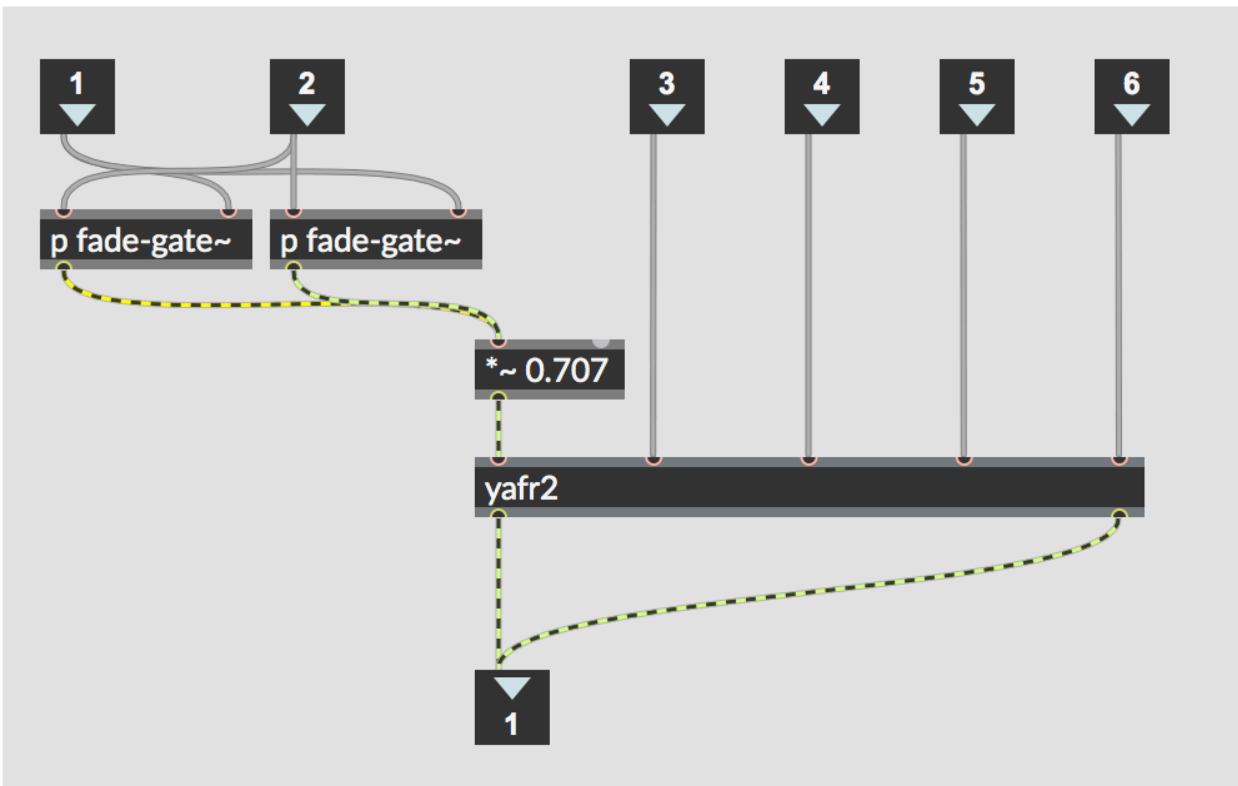

Figure C.20: SM_Reverb/Reverb

C.11 SM_SamplePlayer1

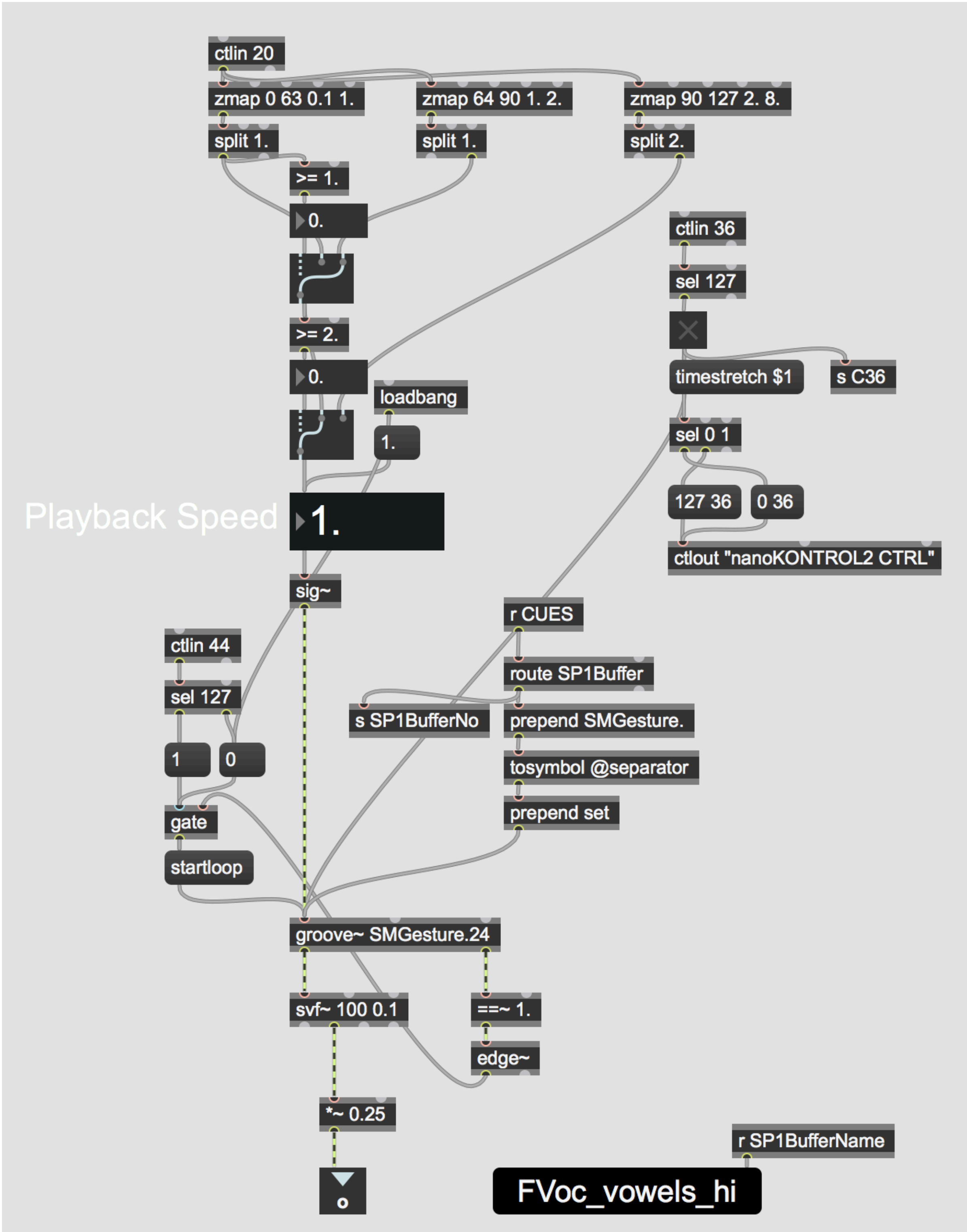

Figure C.21: SM_SamplePlayer1

C.12 SM_SamplePlayer2

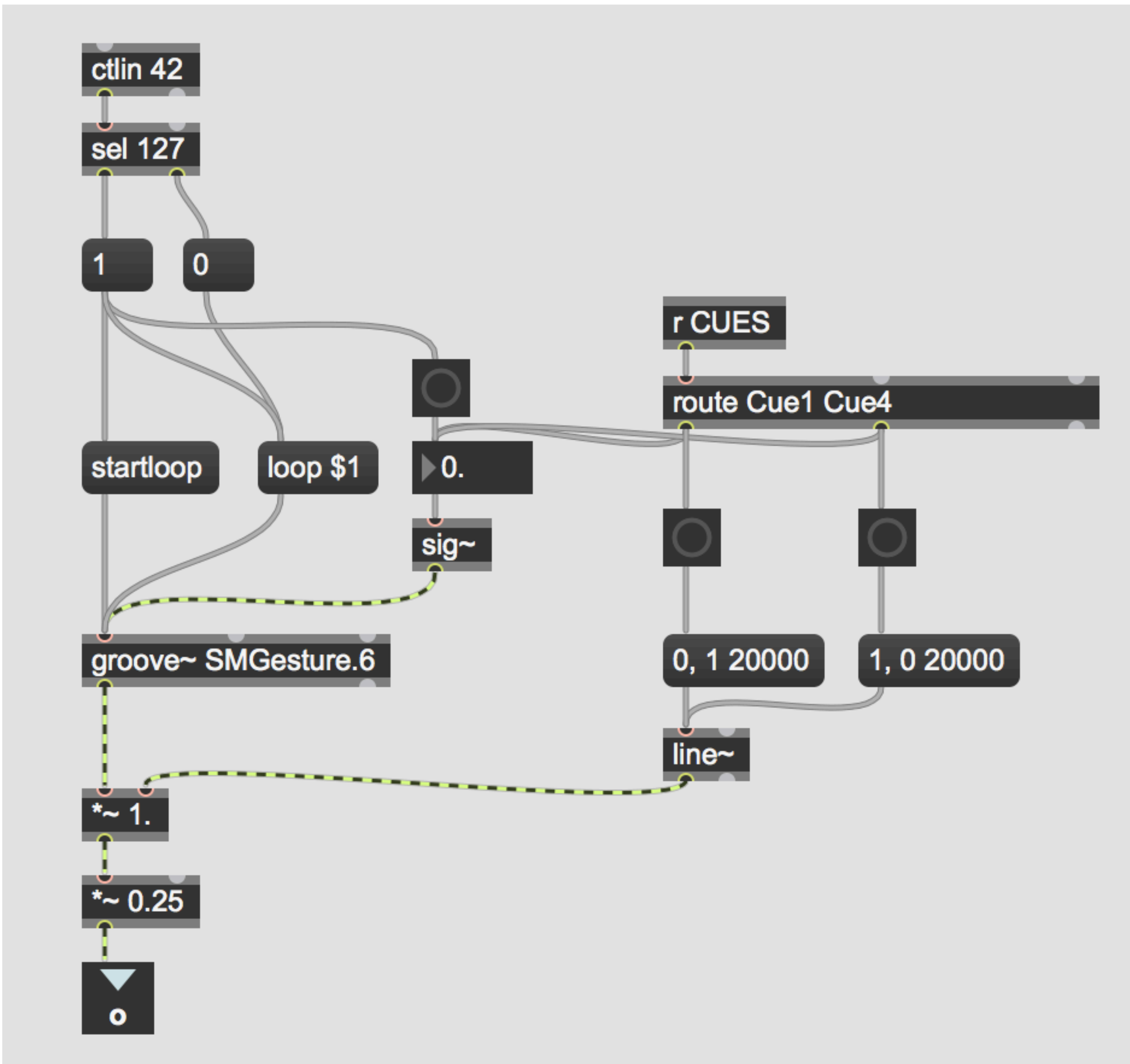

Figure C.22: SM_SamplePlayer2

C.13 SM_VowelFilter

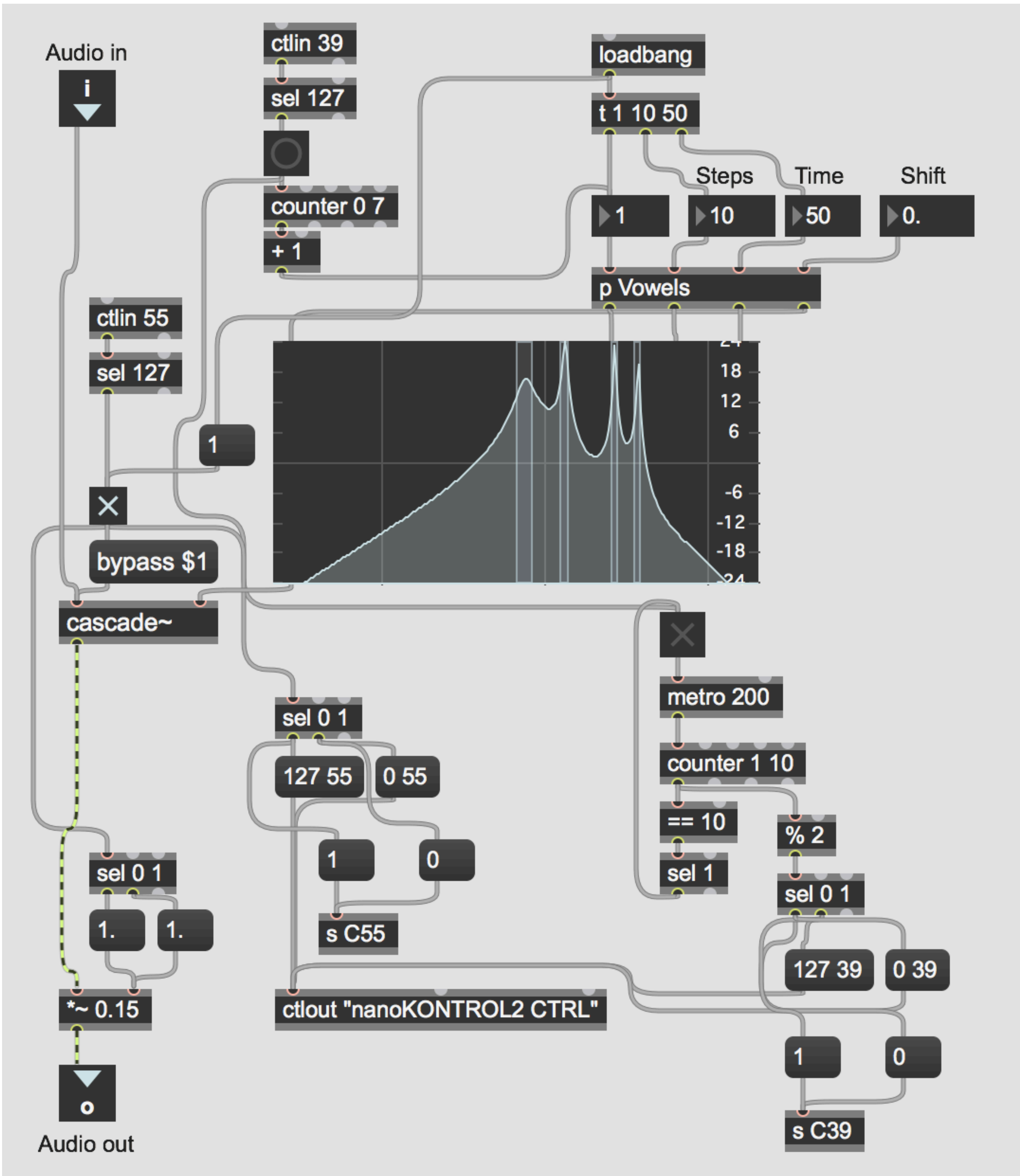

Figure C.23: SM_VowelFilter

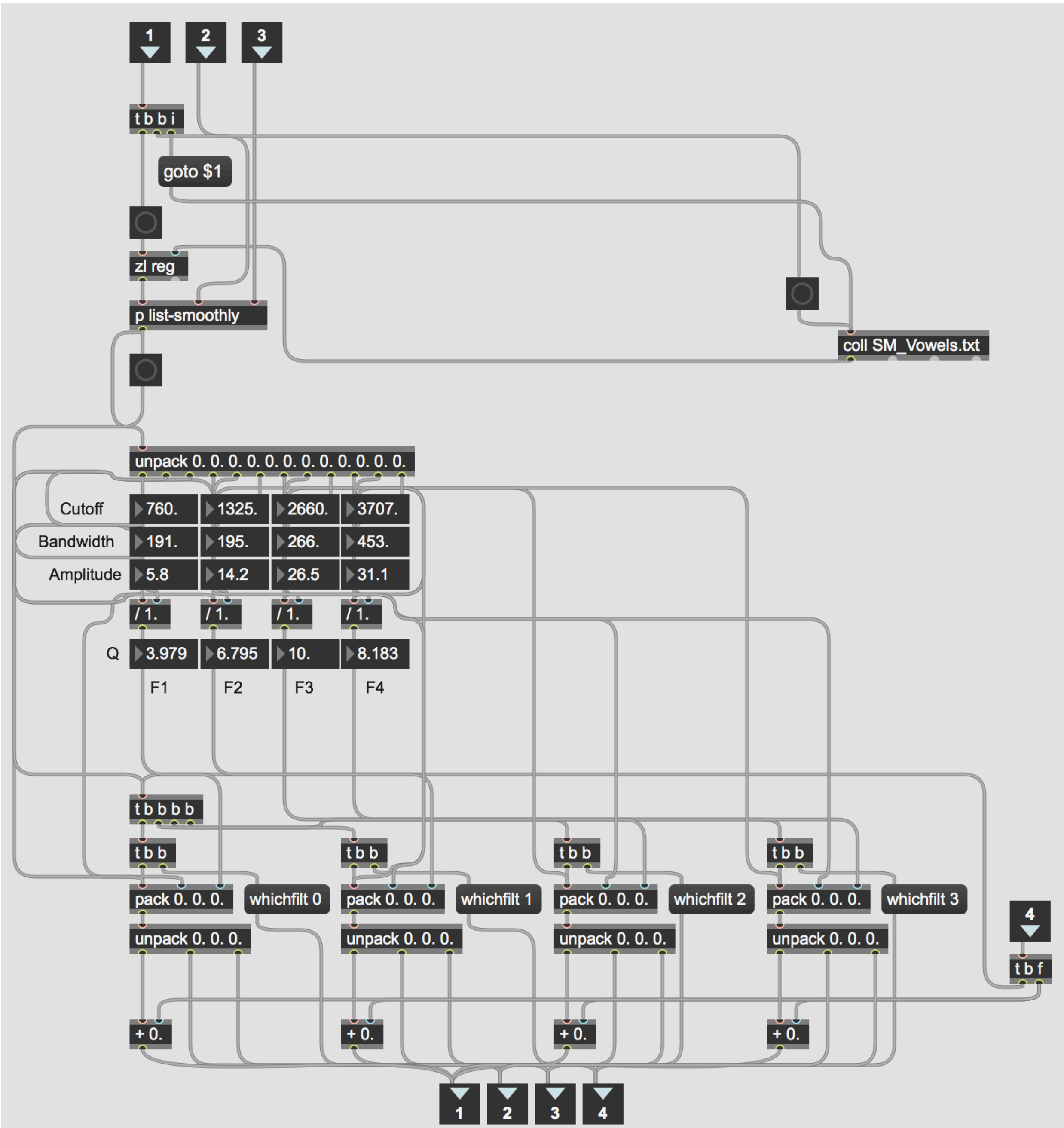

Figure C.24: SM_VowelFilter/Vowels

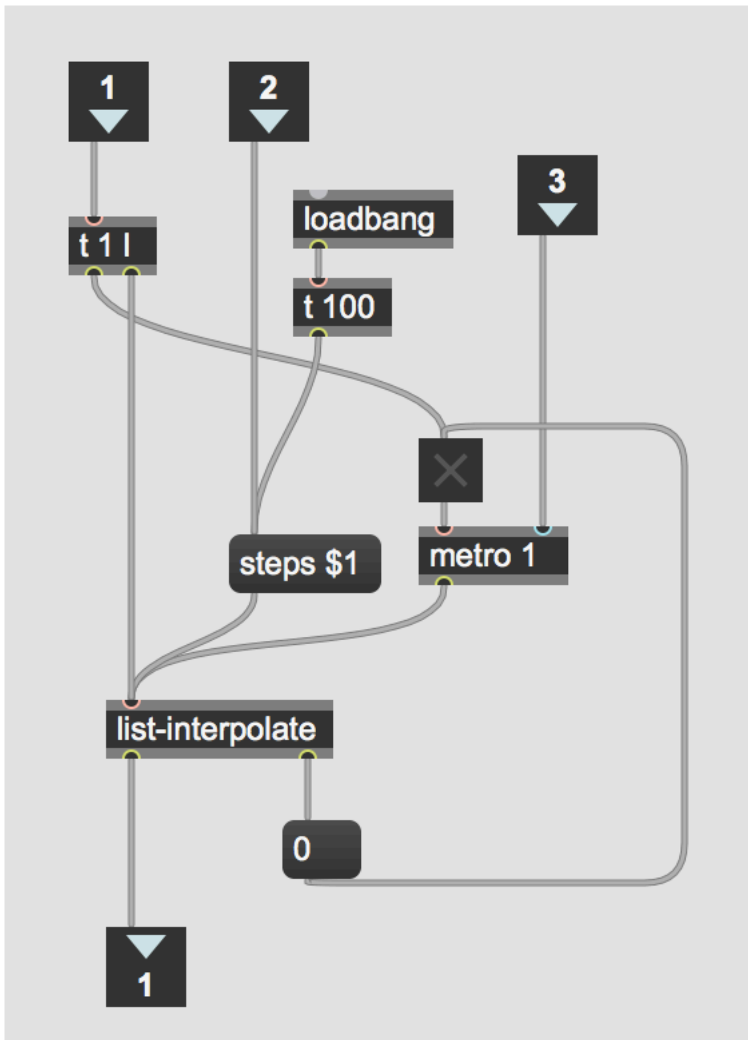

Figure C.25: SM_VowelFilter/Vowels/list-smoothly