LIVE SAMPLING IN IMPROVISED MUSICAL PERFORMANCE: THREE APPROACHES AND A DISCUSSION OF AESTHETICS

Jeffrey Martin Morris, B.A., M.M.

Dissertation Prepared for the Degree of

DOCTOR OF MUSICAL ARTS

UNIVERSITY OF NORTH TEXAS

August 2007

APPROVED:

Joseph Klein, Major Professor and Chair of the
Division of Composition Studies
David Schwarz, Committee Member
Cindy McTee, Committee Member
Jon Christopher Nelson, Committee Member
Graham Phipps, Director of Graduate Studies in
the College of Music
James Scott, Dean of the College of Music
Sandra L. Terrell, Dean of the Robert B. Toulouse
School of Graduate Studies

Morris, Jeffrey Martin. *Live Sampling in Improvised Musical Performance: Three Approaches and a Discussion of Aesthetics*. Doctor of Musical Arts (Performance), August 2007, 99 pp., 7 tables, 45 illustrations, references, 65 titles.

Three original software programs utilizing improvisation and live sampling are presented here, along with a discussion of aesthetic issues raised by each. They are entitled Zur Elektrodynamik bewegter Musiker, Motet, and Gamepad Sampler. These programs vary in the degree of required interaction and in the kind of user control. They are each studies in imitative counterpoint through live sampling, with an approach seeking elegance before solutions. Because of the improvisational nature of these works, there is no standard musical score. Instead the complete Max/MSP source code and a sound recording of performances making use of these programs in varied situations are included.

A discussion of issues raised by these works includes aesthetics, ontology, performance, and the role of the composer. Non-interactive indeterminate compositions are ontologically thin, because some composerly agency is required of the performer. An interactive work can be ontologically substantial if it makes distinct and significant contributions to performance, even though it may not make sound on its own. Although reproducibility reduces ontology and eliminates aura, live sampling within a performance can deepen the ontology of the performance by recontextualizing previous events, reframing the original event as the first reference to an abstract musical idea that lies outside the musical performance. Reproducibility also diminishes the aura or stage presence in live performance with computers. Complex feedback systems can be used to create computation instruments: musical instruments whose unique structure resonates in ways not explicit in their programs. As the human condition and the situation of the composer

change, definitions of the composer and performer must be revised. Composition is shifting away from the creation of static artifacts toward the design of dynamic systems.

ACKNOWLEDGEMENTS

Thanks to my early mentors, Jane Piper Clendenning, without whom my career may have been something mundane, and Joseph “Butch” Rovan, who has long been and remains to be an inspiration. Thanks to the members of my committee, David Schwarz, Cindy McTee, Jon Christopher Nelson, and especially Joseph Klein for his extensive advice throughout my time at the University of North Texas.

Thanks also to James Bohn, Phyllis Chen, Eric km Clark (sic), Andy McWain, Lawrence D. “Butch” Morris, Kevin Patton, Kathryn Woodard, and Mark Zanter for facilitating and joining me in performances and recordings of these works, and especially to Ulrich Maiß for, in addition to these things, inspiring me to return to improvised music.

Finally, thanks to my wife, Stephanie Lynn Morris, and the rest of my family, friends, and other colleagues who have in other ways been supportive during the research, development, and presentation of these works.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Page

CHAPTER 1

INTRODUCTION

Three original software programs utilizing improvisation and live sampling are presented here, along with a discussion of aesthetic issues raised by each. They are entitled *Zur Elektrodynamik bewegter Musiker, Motet,* and Gamepad Sampler. These programs vary in the degree of required interaction and in the kind of user control. They have been used in a variety of musical situations for over a year, which have been recorded in studio and public performances, including international audiences. The experiences and developments from these performances have raised fascinating issues and led to a deeper understanding of practical concerns surrounding this type of composition, explored and developed here. Because of the improvisational nature of these works, there is no standard musical score; instead, the complete Max/MSP source code and a sound recording of performances making use of these programs in varied situations are included.

A discussion of issues raised by these works includes aesthetics, ontology, performance, and the role of the composer. Non-interactive indeterminate compositions are described as ontologically thin, because some composerly agency is required of the performer. An interactive work can be ontologically substantial if it makes distinct and significant contributions to performance, even though it may not make sound on its own. Although reproducibility reduces ontology and eliminates aura, live sampling within a performance can deepen the ontology of the performance by recontextualizing previous events, reframing the original event as the first reference to an abstract musical idea that lies outside the musical performance. Reproducibility also diminishes the aura and stage presence in live performance with computers. Because they are based upon restructuring live improvised musical events, these compositions can be used incomplex feedback systems to create *computation instruments*: musical instruments whose unique structure resonates in ways not explicit in their programs. As the situation of the composer changes in response to new technological possibilities and the intervention of communications media in live performance, definitions of the composer and performer must be revised. Composition is shifting away from the creation of static artifacts toward the design of dynamic systems.

It is difficult to trace a lineage of works developed in a genre as direct predecessors to the works presented here. However, the following diverse works, presented in chronological order of their creation, demonstrate earlier developments in some relevant aspect.

The Studies for Player Piano by Conlon Nancarrow (fifty-one studies composed 1948–1993) have been inspiring to me for their approach to imitation and texture (Gann, 1995). The approach is more focused on natural strengths of the technology than traditional musical structure, avoiding any predominant tonality or meter with a multiplicity of tonalities and meters by imitating—or "sampling"—and manipulating the speed and pitch of an entire voice in a canon. The works highlight the musical potential of passages that are beyond human capabilities and the superposition of multiple contrapuntal voices in a kind of *meta-counterpoint*: textural interaction among groups of polyphonic musical voices. They shift focus toward time manipulated more directly as a medium of musical expression, for example through Nancarrow's tempo canon technique.

Christian Wolff's *For 1, 2, or 3 People* (1964) and his similar compositions of the time are open works that specify little content and are largely indeterminate. Like Karlheinz Stockhausen's *Klavierstück XI* (1956), passages are scattered about the page for each performer to execute in the order one's eye falls upon them. Instead of fully notated passages, each group of symbols directs a performer to play in certain timbres or registers, and to wait for certain events before starting, stopping, or changing performed events. Most notable about this work is that instead of only composing events, Wolff composed reactions to events. The composition explicitly includes interaction.

Public Supply (1966, 1968, 1970) and *Radio Net* (1977) are improvisational environments created by Max Neuhaus using radio broadcasts to create gigantic feedback loops (Neuhaus, 1994). Both works use content provided by radio listeners calling in (vocal sounds or whistling). *Public Supply* was centered in one city at a time, with Neuhaus controlling the incoming signals. For *Radio Net*, Neuhaus created automated mixing devices and sent them to the nationwide call centers. During performance, he and his assistants only intervened enough to keep the loop from getting out of control. Participants were encouraged to keep their radios on near the telephone as they called in to provide more feedback loops. These works exemplify the creation of structure apart from content, automated mixing and processing, and feedback loops for musical structure and content.

Christian Marclay was the one of the first turntablists, using a phonograph as a musical instrument from the mid-1970s. Since the phonograph was originally created as a tool for reproduction, it requires prerecorded material and allows live and dramatic warping of it, often beyond recognition of the original source. Marclay has a rich history of performances, including performing in conduction ensembles with Butch Morris.

Not a composition but rather a language for improvised performance, Lawrence D. "Butch" Morris's system called *conduction*, publicly debuted in 1984, uses hand signals to coordinate a group of improvising musicians (Morris, 1995).[1] The few content-oriented directives of conduction are limited in specificity. For example, there are directives to play long or short notes without specifying pitch or tonality; register directives can move pitches up or down by large or small amounts without being any more specific than that. The focus of conduction is dictating musical relationships while affording each performer an independent voice. The system is much more robust than it may sound. Directives like repeat, accompany, develop, and memory cues (enabling the conductor to return to previous musical material and behaviors) allow the conductor to elegantly coordinate a depth of structure and development that is otherwise very difficult to achieve in improvised music, while the music benefits fully from the unique character of each performer and the "fortunate accidents" of improvisation.

Another artist associated with Butch Morris is J. A. Deane, who performs with experimental electronics and samplers. He became active recording new music in the 1980s, and has lately focused on the use of samplers. He primarily draws on a large library of sounds he has recorded, but he also records other musicians and manipulates those recordings during

[1] Butch Morris has no family relation to Jeffrey M. Morris.

performance. Deane uses commercial software applications including LiSa™ [2] and Ableton Live.™ [3] Unlike *Motet* and Gamepad Sampler , Deane avoids pitch shifting and time compression/expansion, preferring instead to manipulate sample rates to achieve those effects (in tandem), for the unexpected discoveries yielded from within sounds.

George Lewis's *Voyager* (1985–87) is an interactive improvisation environment for electroacoustic sound or MIDI instrument that is very sophisticated in the way it interprets the contribution of the human performers (Lewis, 2000). Material is analyzed for patterns, structures, and trends in pitch, rhythm, and loudness. It is then transformed to allow the program to mimic, develop, complement, or challenge it. Robert Rowe's *Cypher* performs a similarly deep analysis and allows the user to map parameters of the input to the same or different parameters of the generated accompaniment (Rowe, 1992, 2001).

Andrew May's *The Twittering Machine* (1995) for flute and interactive electronics combines prerecorded sound with live processed sound and uses a footpedal to allow the performer to guide the accompaniment through the score. *Retake* (2001) for flute and interactive electronics, composed by May along with Elizabeth McNutt, is a vehicle for improvisation. A prerecorded improvisation played by McNutt on the flute serves as the "spine" of the piece, through which the performer can scroll with a pedal. The live flute performance and the prerecorded segment indicated by the position of the pedal inform algorithmic voices which may quote, reinforce, or contradict the human performer (McNutt, 2003).

Finally in this survey, Tristan Jehan's work (2005) includes software programs that analyze prerecorded music and create new compositions by designing and applying new structures to those recordings, after models of human listening, composing, and performing. This analysis/resynthesis routine seeks structural hierarchies in meter, rhythm, pitch, and timbre. The non-real-time process can be used for musically informed cross-synthesis between musical recordings, extrapolation of incomplete recordings, or generation of new material.

[2] LiSa, from "live sampling," is a trademark of the Studio For Electro-Instrumental Music (STEIM).

[3] Ableton Live is a trademark of Ableton AG.

My approach has been to seek elegant transformations of the performer's input instead of comprehensive or brute force approaches to machine intelligence. As an artist, my goal for a particular work is not usually to solve a particular well-defined problem for which there is an established set of viable procedures. If art is to enhance, enlighten, inform, or explore the human condition, it will benefit from exploration, discovery, experience, and play with the tools of the art form. The strengths of these tools, if they are appreciated as strengths, present elegant means to some end. The discovery process then includes evaluating whether such an avenue is worth developing and presenting as an aesthetic experience for an audience.

It has become apparent to me that much of art is like hacking—in a benevolent sense, meaning deconstructing the rules of a system in search of a way to coax from it unintended results. It is more common for programmer-hackers to consider themselves to be like artists (Wark, 2004) than for artists to see themselves as hackers, but the analogy seems appropriate. Poets short-circuit syntax in order to achieve greater depth in meaning. Alban Berg manipulated the twelve-tone technique to create tonal structures and embed secret personal allegories (Adorno et al., 1994). Circuit bending, a common musical hacker activity that involves adapting electrical circuits for rich musical results, begins with exploration—playing around in search of intriguing possibilities to pursue (Ghazala, 2005; Collins, 2006). In the explorations leading to the creation of these works, I investigated—played with—the diverse effects achievable from delay lines, the artifacts resulting from dramatically slowed samples, and the self-sustaining sounds emerging from complex feedback systems. This was of course guided by my previous experience, research, and intuition. Observations from this process have opened up multiple modes of performance, bringing up fascinating aesthetic issues including the value of audio feedback systems, the aesthetics of live sampling, stage presence, and the ontology of such works.

CHAPTER 2

ZUR ELEKTRODYNAMIK BEWEGTER MUSIKER

Elektrodynamik was designed to be an autonomous partner in improvisation, changing settings and its mode of interactivity automatically. However, controls created for use in the development stage have proven valuable for the performer to use at times. The program builds up textures and gestures from the performer's live input using delays, Doppler shifts, and changing sample rates. The user is given little control over details, but rather acts more like a conductor: starting, stopping, or triggering change, and adjusting loudness levels. In this way, the user has the ability to directly trigger an event at a given time, but the details concerning which event is triggered are determined by the program.

Zur Elektrodynamik bewegter Musiker was created for cellist Ulrich Maiß, who requested an improvisational work for a solo concert program called ZenMan Improvisations. It was first performed in Brandenburg, Germany, on September 25, 2005 and in Berlin on September 27. The work is dedicated to the centenary of Albert Einstein's special theory of relativity (1905), and the title is derived from the title of Einstein's paper presenting the theory. Einstein's title is "Zur Elektrodynamik bewegter Körper" ("On the Elektrodynamics of Moving Bodies"). The original title of the musical work specified *Cellisten* (cellists); it was later broadened to *Musiker* (musicians).

The special theory of relativity includes the conclusion that time is not real, or at least that there is no absolute or authoritative time. Because the speed of light is observed to be the same from any point of view, the appearance of synchronicity is also limited to a single point of view, and temporal measurements will differ if taken from reference frames under different amounts of acceleration. Temporality has been an interest of mine in physics, metaphysics, and the cognition and composition of music. In my musical studies, a search for a less arbitrary means of structuring the pitch continuum resulted in the conclusion that rhythm and texture—temporal elements—can be more salient and therefore expressive parameters in music (see the discussion in Morris (2004)).

The composition is a study in imitation as a technique to build musical textures and gestures, inspired by the implications of special relativity. Einstein's work on time began by musing what it would be like to ride a beam of light (Kaku, 2004). The question is often added, "What would one see in a mirror held before oneself while riding?" The mirror and the manipulation of light flowing between one's face and the mirror are devices of imitation. In the composition, delay lines and samples recorded during performance are the sonic mirrors, reflecting the sound of the soloist. These are applied in three different ways in musical voices called Player, Beat, and FX. All sounds played by the program in a performance originate from sounds played by the performer during the same performance.

The performer is invited to allow his or her performance to be influenced by one of ten statements or questions inspired by the implications of special relativity (table 2.1). These are the only instructions to the performer. Each statement is associated with one scene, which consists of settings for several parameters of the program, serving to establish formal divisions and development within each performance.

There is no designated duration for a performance. Section durations can range from ten to sixty-five seconds. Performances shorter than one minute are possible but will not allow the program to form a temporal structure in its characteristic way. While the program can continue running indefinitely, it only stores the past fifteen minutes of a performance, after which it will not replay any material played by the performer early in the performance. While this is a limitation on the imitative reappearance of musical material, it could serve to allow for variety in materials in extremely long performances. Musical ideas may still be reintroduced during a performance by the soloist after any previously recorded version of it is lost from the buffer. In this way, the performer and computer can imitate each other, prolonging and recontextualizing previous moments in the music. Although most performances to date have been around fifteen minutes in duration, I have been pleased with longer performances and installations using feedback, so I do not consider there to be any maximum performance duration.

Table 2.1. Phrases displayed to the soloist during performance.

Time is malleable, inconsistent, and might be a mere illusion.
There is no such thing as synchronicity, except from a given point of view.
Nonlinearity is natural.
Space and time are governed by the motion of light.
Motion slows the flow of time.
Two objects moving apart will see each other as the one that slows down.
Matter, energy, and motion are all related.
A moving object appears elongated.
Instantaneous interaction is impossible.
What would it be like to ride along side a beam of light?

Technical Description

Interface

The interface of *Elektrodynamik* was originally designed for user interaction at the level of head nods or hand signals between improvising performers. It displays the inspirational phrase associated with the current scene, and when it changes, it signifies to the performer that the program has entered a new mode of behavior, at least in most aspects. It also indicates the loudness setting of each voice with faders and uses a colored "light" to indicate when a voice is making sound. Finally, it acknowledges the user's commands.

Apart from initial loudness settings and starting and stopping the program, no control input from is expected the performer during performance. The performer may, however, use a set of keyboard commands to trigger changes in the behavior of the program, which are listed in table 2.2.

Scenes

Whereas the score of a traditionally notated composition contains instructions or values (in pitch, duration, loudness, etc.) to be realized by performers during performance, the Scenes module in *Elektrodynamik* sets parameters and modes of the other modules in the

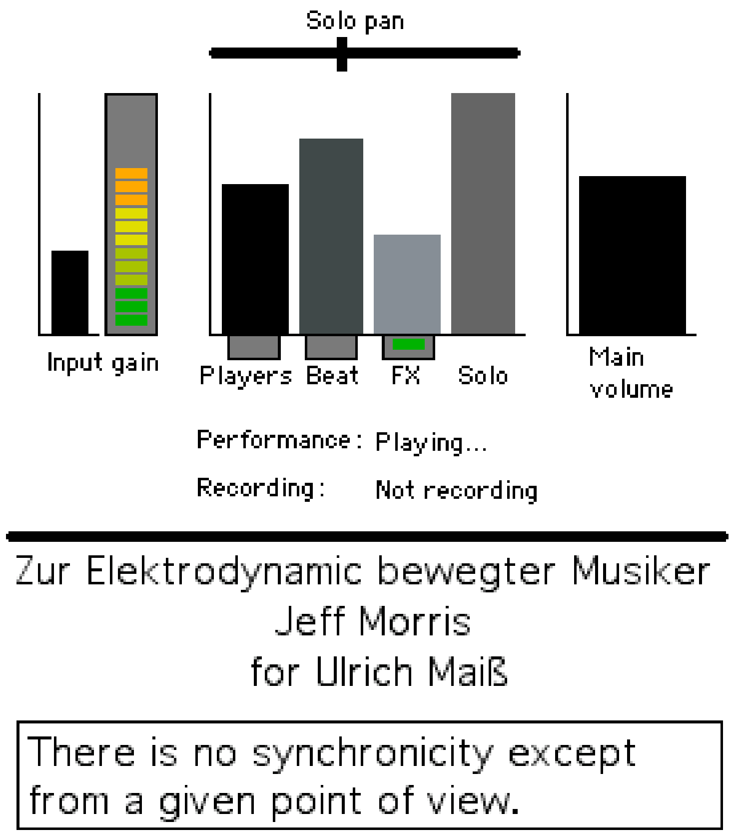

FIGURE 2.1. The user interface of *Zur Elektrodynamik bewegter Musiker.*

program (listed in table 2.3), including the module that displays inspirational text to the performer listed in table 2.1 on page 8. This is a significant contribution to the ontological substance of the composition, because it imposes structure upon the improvised performance at the level of large scale form and the character of the musical surface. This score consists of ten scenes selected through a drunk walk, allowing it to move to adjacent scenes.[1]

[1] A *drunk walk* is a randomized process that has an equal chance of moving up or down in sequence with each step, like a drunk stumbling along a path

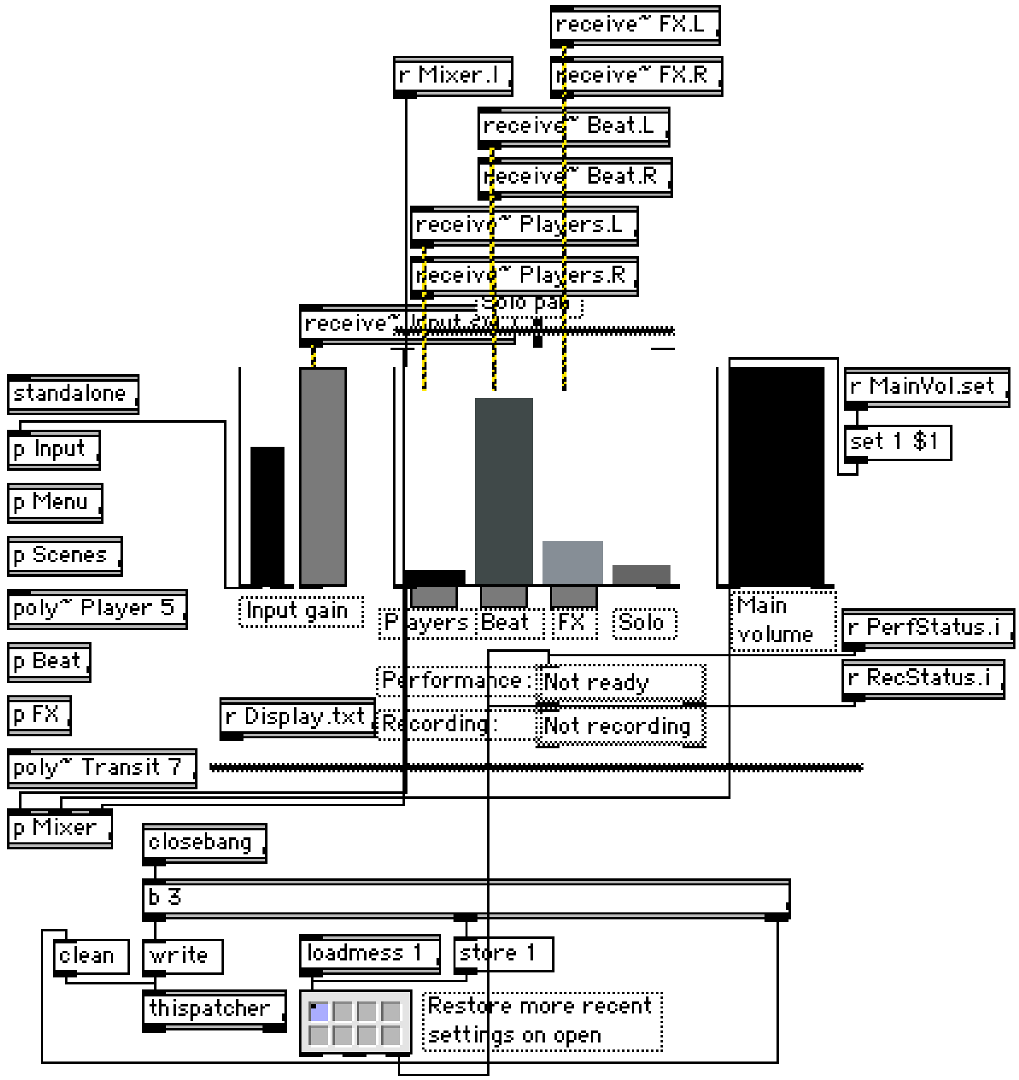

FIGURE 2.2. The main patch of *Elektrodynamik* including elements hidden from the user: subpatch boxes and objects that deal with the user interface objects. *Note:* See figure 2.4 (page 12) for the Scenes subpatch.

Table 2.2. Keyboard commands.

l: Loop playback in Player voices instead of choosing new segments to play.

L: Deactivate looped playback.

b: Activate the Beat voice.

B: Deactivate the Beat voice.

f: Randomly select a new preset in the FX module.

i: Randomly select new interaction modes for each voice.

\: Automatically fade main volume to zero over fifteen seconds (fade out).

m: Activate the Automix module.

M: Deactivate the Automix module.

0: Set interaction mode 0 for all voices, ignoring the loudness of the input signal.

1: Set interaction mode 1 for all voices, following the loudness of the input signal (faux homorhythm).

2: Set interaction mode 2 for all voices, complementing the loudness of the input signal (faux hocket).

3: Set interaction mode 3 for all voices, following the loudness of the input signal delayed by five seconds (faux rhythmic canon).

space: Mute audio input while the space bar is depressed. This is useful in controlling feedback situations, keeping unwanted noises from being sampled, or introducing rhythm into sustained sounds as they are sampled.

dirty: (When keys are pressed in sequence spelling the word *dirty.*) Activate special randomized distortion (clipping) effect (see page 27). Note: when the "i" is pressed in this word, it also triggers a change in interaction modes as described above.

pretty: (When keys are pressed in sequence spelling the word *pretty.*) Deactivate randomized distortion effect (see page 27).

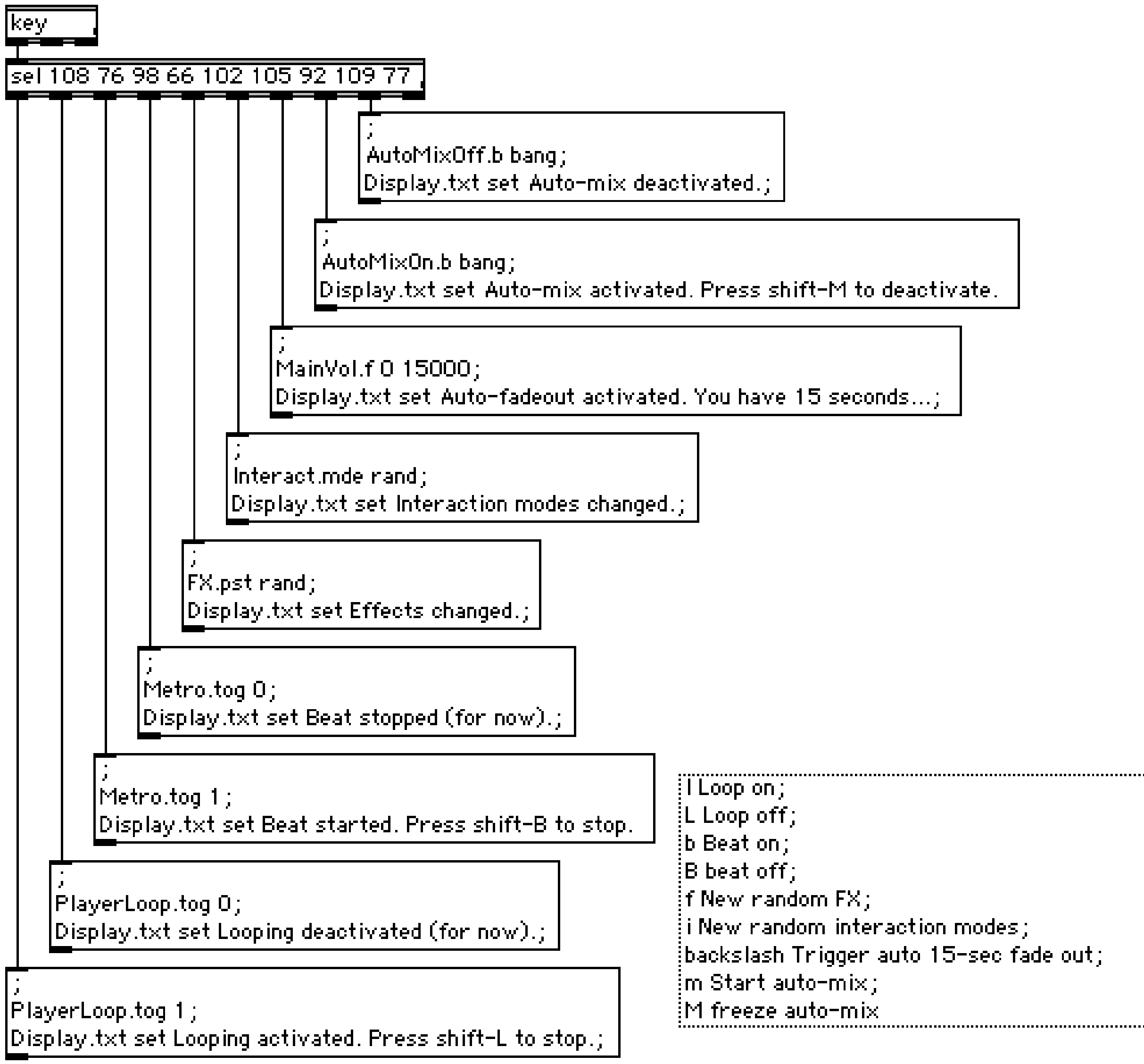

FIGURE 2.3. The Key Commands subpatch sends commands in response to corresponding keys pressed by the user. A legend is provided in the comment box (bottom right).

Table 2.3. Parameters and modes dictated by scenes.

BeatDcyMax.ms: Maximum decay time of each percussive attack in the Beat voice, in milliseconds.

Earliest.mx: Earliest point in the recorded performance to use for possible playback, described as a number between 0.0 and 1.0.

FX.pst: Delay and feedback preset indicated by an integer or rand, which triggers a random selection.

GestMax.ms: Maximum duration of a gesture played by the Player voice.

Interact.mde: Interaction mode indicated by an integer or rand, which triggers a random selection (see page 30)

Latest.mx: Earliest point in the recorded performance to use for possible playback, described as a number between 0.0 and 1.0.

Metro.tog: Activate or deactivate the metro (metronome) object that controls the Beat voice with 1 or 0, respectively.

PlayerDelMax.ms: Maximum randomly chosen latent period between gestures played by a Player voice, in milliseconds.

PlayerLoop.tog: Force Player modules to repeat the same gesture instead of choosing new ones, activated by a setting of 1 and deactivated by 0.

SectMax.ms: Maximum duration of a section, in milliseconds.

SectNewDel.max: Maximum delay between the beginning of a new section and the selection and playing of new gestures in each Player voice.

Transit.tog: Activate or deactivate the Transit module with 1 or 0, respectively (see page 23).

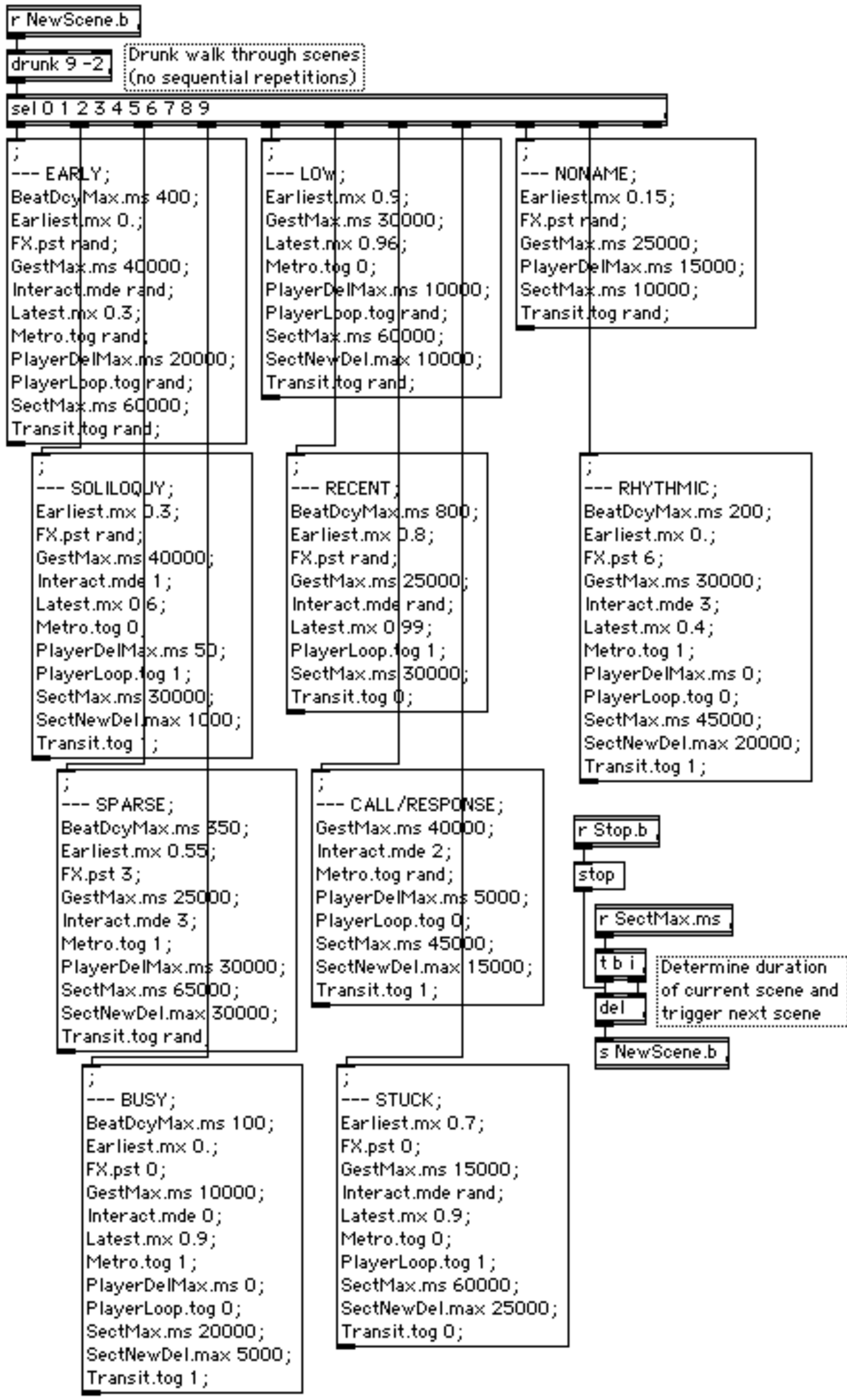

FIGURE 2.4. In the Scenes subpatch, each message box contains commands to set parameters for each scene and automatically choose the next scene within the specified parameters.

Not every scene contains settings for every parameter as described above. This is in order to allow for a variety of instances of each scene, a variety of transitions between sections, and to aid in a sense of continuity across multiple sections. For example, the scene called NONAME does not specify a value for Latest.mx, so it will inherit the value from the the last scene to set a value for Latest.mx, or it will use the default setting if no scene has set a value yet in a performance (see figure 2.4). So, if NONAME comes after the scene called RHYTHMIC, the Player voices will use material recorded earlier in the performance. If NONAME follows STUCK, the Player voices will draw on material from most of the recorded input. Similarly, the Player voices may or may not loop or not during the NONAME scene, depending on which scene preceded it.

In another example, the parameter BeatDcyMax.ms is set to 800 (milliseconds) in RECENT, but is not set in the three scenes between it and RHYTHMIC, which sets the parameter to 200. So, for a longer portion of a performance, through several section changes, the Beat voice will have longer decays after RECENT until it reaches RHYTHMIC. With a drunk walk between the two scenes, it is possible for the scene selection to wander in the area before RHYTHMIC for a very long time before reaching it. If the scene selection reaches RHYTHMIC, then the decay time will be short for a similarly long time, until the scene selection returns to RECENT. Also, if the performance begins on one of the scenes between RECENT and RHYTHMIC, the BeatDcyMax.ms parameter will be revert to the default setting of 3000 milliseconds, which is loaded when the program is started and persists until a scene sets it to another value.

Note that flexibility also results from dictating ranges instead of discrete values for many parameters. This approach to defining scenes allows the character of each scene to be defined by some parameters, but lets that character be realized with different detail settings each time it is selected. It also enables the program to establish constancy in some parameters across several sections, serving to make the overall form more coherent, but also enabling more dramatic changes when a parameter finally changes after remaining constant for a long time.

Players

If the Scenes module is the computer's score, then the Players, Beat, and FX modules are its instruments. Five sound players with randomized parameters are controlled by the Players fader in the interface. These modules play passages recorded from the soloist during a given performance. The parameters of each Player module are set with randomized processes within the specifications given. The SectNewDel.max parameter set at the beginning of each scene triggers the Player voices to start after a randomly chosen delay up to the value of SectNewDel.max. Start, end, and gesture duration values are randomly chosen, and the selected passage recorded from the soloist's input is played. After completing the gesture, the Player voice is inactive for a period up to the value of PlayerDelMax.ms, and new start, end, and duration values are generated and executed.

If PlayerLoop.tog is activated (with a value of 1), the trigger will not generate new values, but will re-trigger the previously chosen values. Unlike an ordinary loop setting, this program still allows for different latent periods between repetitions of the gesture. This gives the effect of motivic play or dynamic conflict among gestures, instead of creating a static ostinato.

Beat

The Beat module imposes a new texture on sounds recorded from the soloist during performance by dividing it into three frequency bands and sounding each band as an independent voice in a metric pattern. The sound is divided into high-, middle-, and low-frequency bands in the frequency domain (via a Fast Fourier Transform) and each band is connected to a different amplitude envelope that opens with a fast attack and slower (but still short) release when triggered, to sound like a percussion instrument.

Because the Beat voice often only sounds brief attacks carved from the input, it is usually necessary to do something to ensure that there will always (or nearly always) be some sound in the input at the time the Beat voice triggers a note. This is accomplished with a five

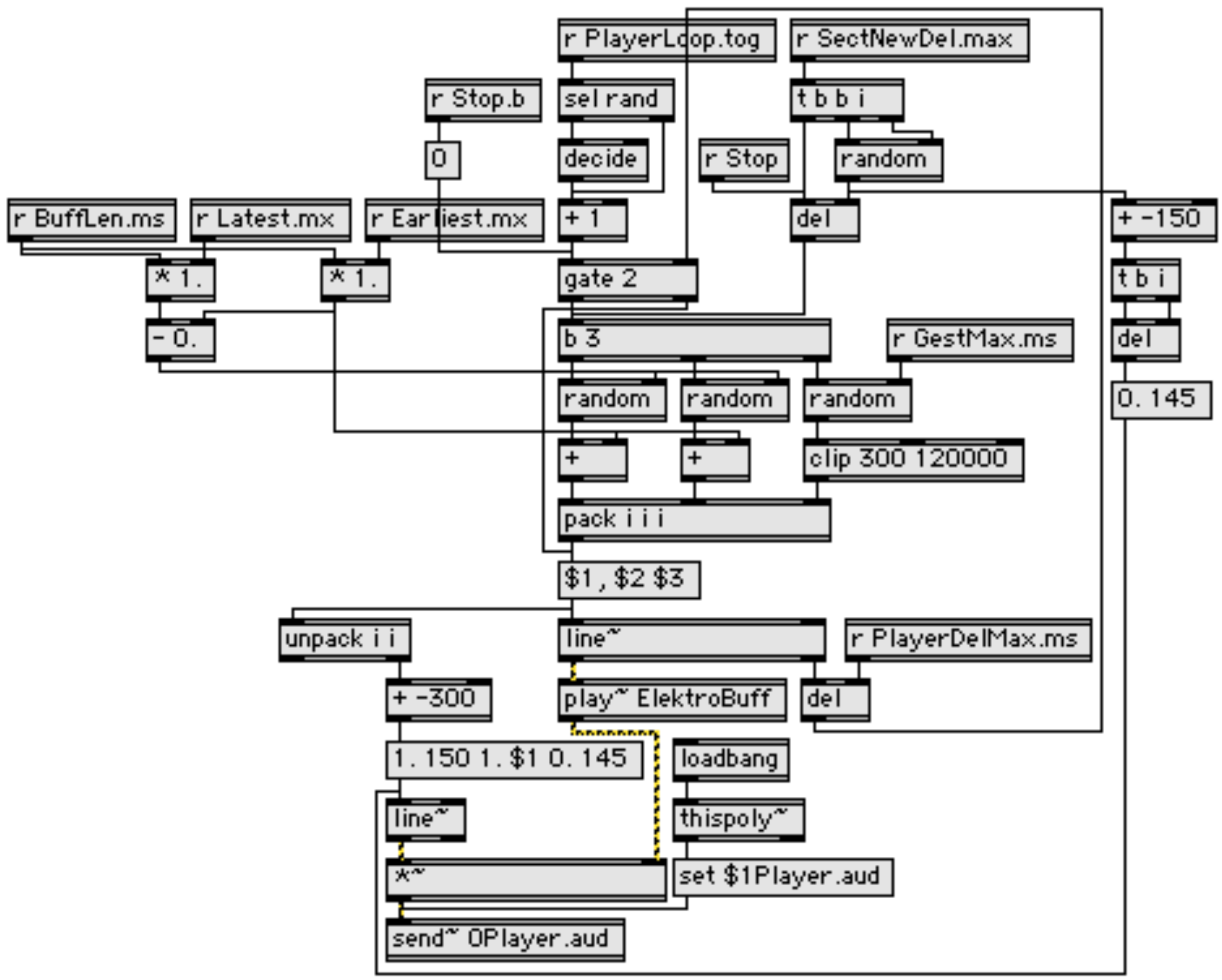

Figure 2.5. The Player abstraction plays segments of the recorded input with the start, end, and duration randomly chosen within parameters specified in the current scene.

second delay line with ninety percent feedback, which I call a *reservoir*, fed by the live input from the performer. This superimposes the player's input on itself in historical layers, with older layers growing softer in each repetition. Depending on the performer's activity, the reservoir can provide a sound source for Beat that is fairly consistent. This has proven to be a satisfying way to capture some of the character of the performer's sound while allowing it to be transformed and structured in a very different way. Because the feedback to the delay is high, the overall loudness of the material in the reservoir will fall when the performer

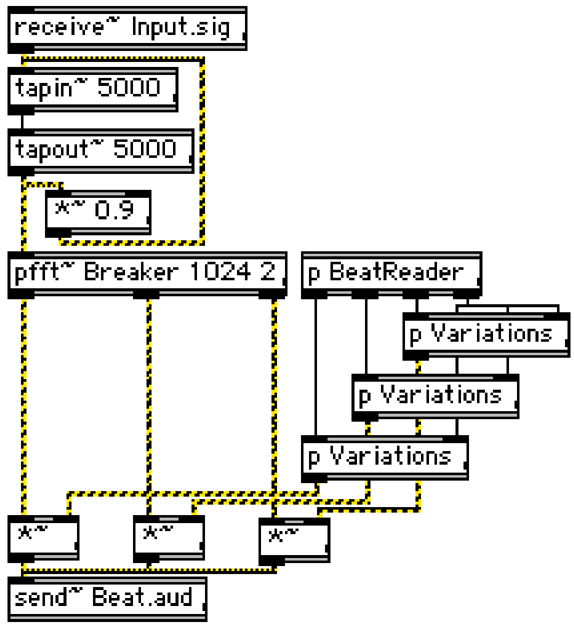

FIGURE 2.6. The Beat subpatch breaks the input signal into bass, middle, and treble voices, and imposes amplitude envelopes on each in a metric pattern.

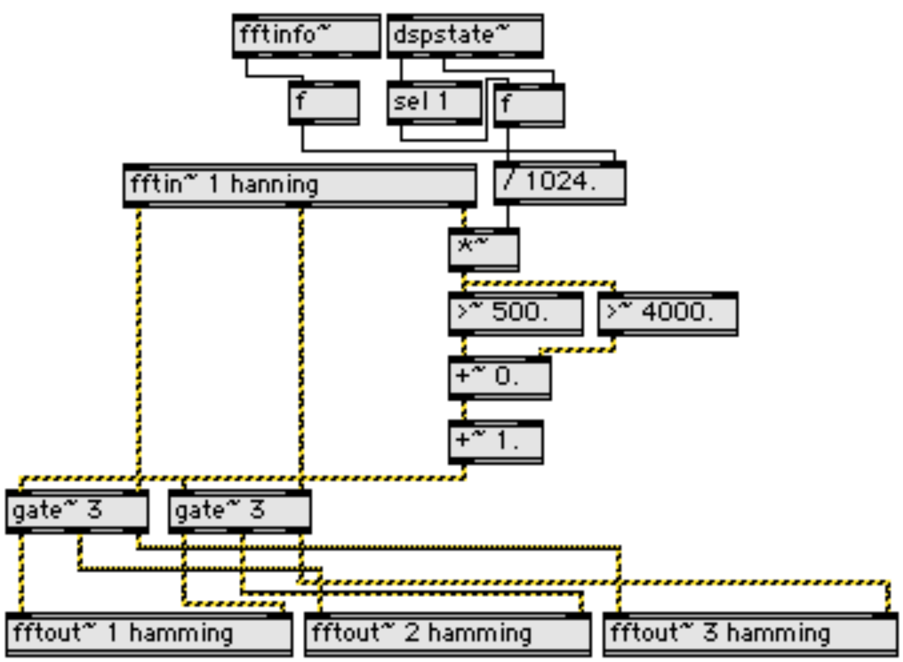

FIGURE 2.7. The Breaker abstraction divides a signal into bass, middle, and treble frequency bands.

stops playing or becomes less active, but sounds in the reservoir will persist audibly for a few minutes. It is designed to fill quickly and empty slowly, to fit the needs of the Beat voice.

The amplitude envelope on each band is triggered in a dynamic metric pattern using the method described in Morris (2004); a brief explanation of that method is provided here. The final metric pattern is created in three layers: an *accent pattern*, an *event pattern* for each voice, and variations that are imposed on each voice in performance.

The *pulse* is taken to be the basic unit of time in the metric pattern. An *accent pattern* is constructed by a randomized process that combines pulses into *pulse groups*, each consisting of two or three consecutive pulses. The first pulse of each pulse group is to be accented by metric and dynamic accent (agogic accents are also possible but are not implemented here). Note that while pulses are treated as a basic and consistent reference unit, accents are not evenly spaced.

A *sound pattern* is created for each voice (high-, middle-, and low-frequency bands) and describes the musical function of each voice within the pattern. It is defined in terms of accents, not pulses, so a given sound pattern may be used with different accent patterns for different results. The sound patterns used here are designed after a three-voice drum set model adapted from early funk drum set patterns, especially from James Brown's early solo recordings (Brown, 1996). In the early funk model, one cycle of the accent pattern consists of two measures in conventional notation. The low-frequency band sound pattern (after the bass drum) could be described as playing a few times in each cycle:

- The first accent of the pattern (i.e., the downbeat of the first measure),
- The accent on or just after the middle of the pattern (i.e., on or around the downbeat of the second measure), and
- A few randomly chosen pulses, whether accented or not.

This method provides an easy way to create a variety of rhythmic results from a single description. The definition of the second "downbeat" allows for the variety of drum set patterns that have bass drum hits on or around the expected downbeat in every other measure, which is common in early funk music. The randomly chosen pulses allow for each

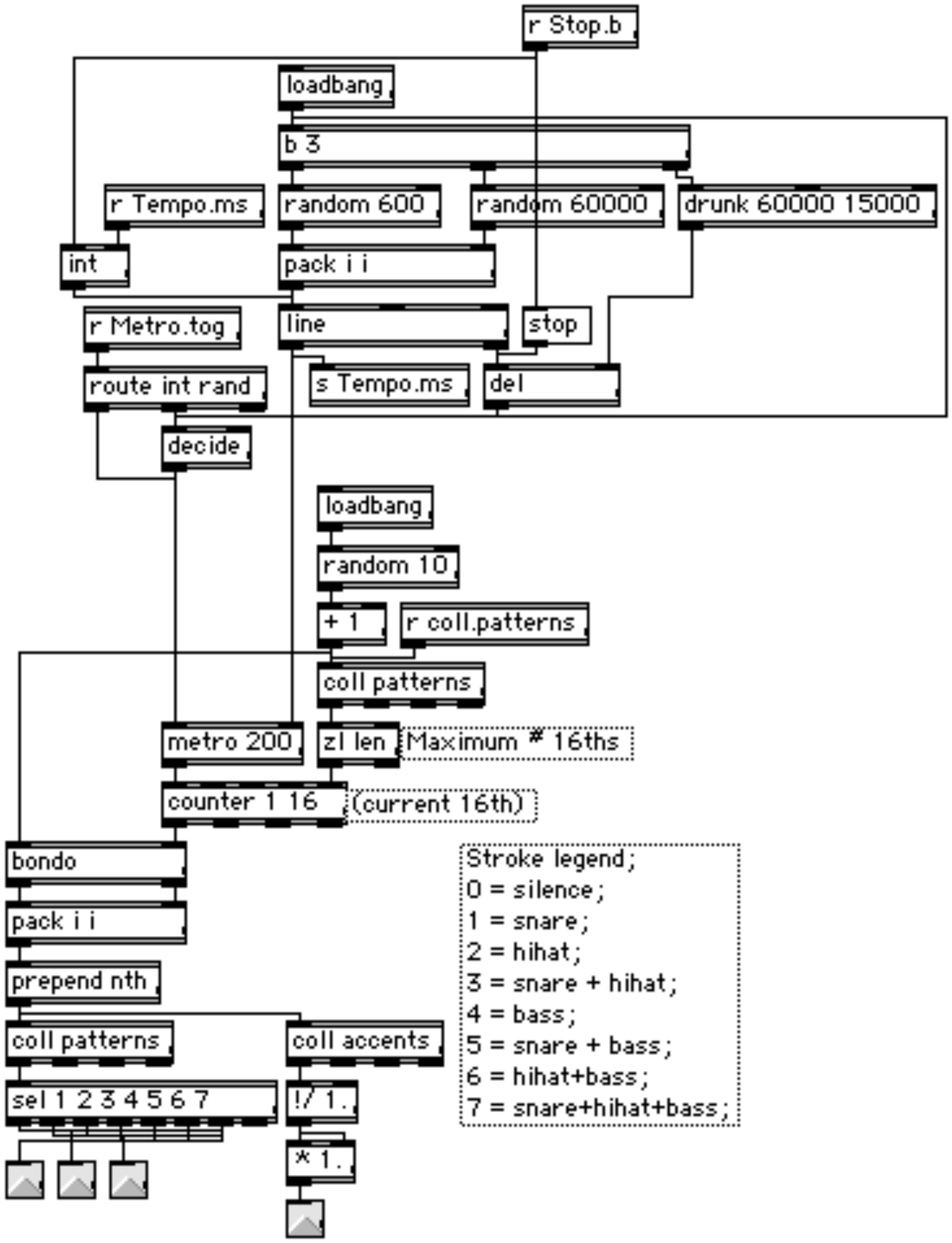

FIGURE 2.8. The Beatreader subpatch starts and stops the metronome and reads through a three-voice metric pattern for use in the Beat component of the program.

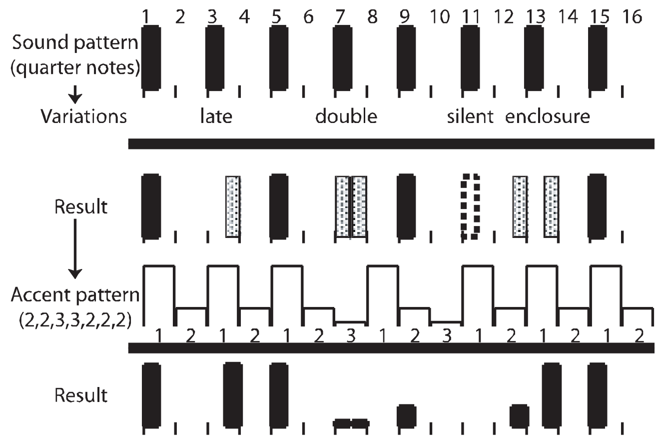

FIGURE 2.9. Metric rhythm is created in three layers: a *sound pattern* (here, it is even quarter notes), *variations* applied during performance, and finally the imposition of an *accent pattern*.

resulting metric pattern to have its own identity while preserving the same basic structural characteristics. These randomly placed events will be accented only if they fall on accented pulses; otherwise, they will fit into the overall pattern as unaccented events. The middle- and high-frequency band voices are given event patterns corresponding to the roles of the snare drum and hihat cymbal in early funk drum set patterns, respectively.

The variation layer imposes randomly selected amplitude envelopes, making each performance and each iteration of the metric pattern different while preserving the overall character of the accent pattern and event pattern. The amplitude envelopes used here are:

Single: A normal single percussive hit,

Double: A normal hit on the designated pulse followed by another hit one half pulse later (as if two consecutive sixteenth notes),

Late: A single percussive hit delayed by one half pulse, and

Enclosure: Two hits, with one a half pulse before the designated pulse and the other one half pulse after the designated pulse, leaving the designated pulse silent (this is an essential part of the funk *boogaloo* drum set pattern).

The pulse rate is constantly changing, and the Beat voice is activated and deactivated by a randomized process or when specified by the score or performer. The result does not

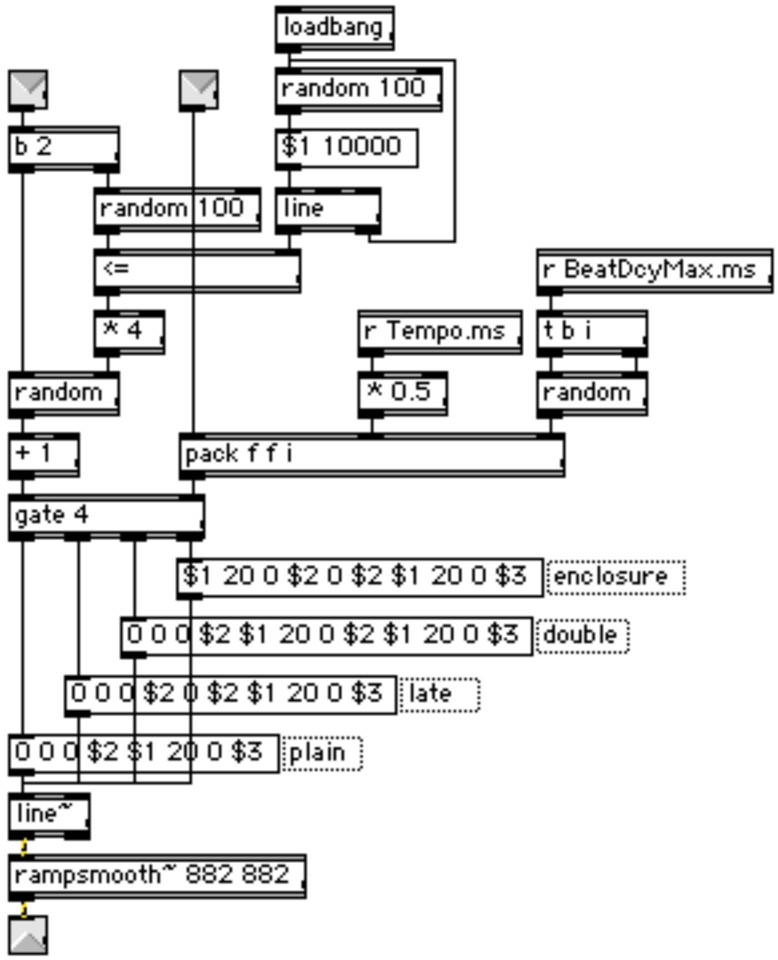

FIGURE 2.10. The Variations subpatch selects and applies one of four variations on a single event in the Beat component of the program.

sound like a James Brown recording (although it can come close with drum samples and conservative settings); it is a sonic texture with a balance of periodic and aperiodic events that evokes a stimulating effect common to funk music I have called *tickle*. Tickle is a balance between predictability and surprise just at the limit of perception: stable enough to compel a listener to anticipate future events, but elusive enough to continually defer satisfaction of those expectations.

FX

The first noticeable sound from the program in performance is the FX voice. It is simply a delay line with feedback and carefully chosen preset values. As described on pages 57–58, a

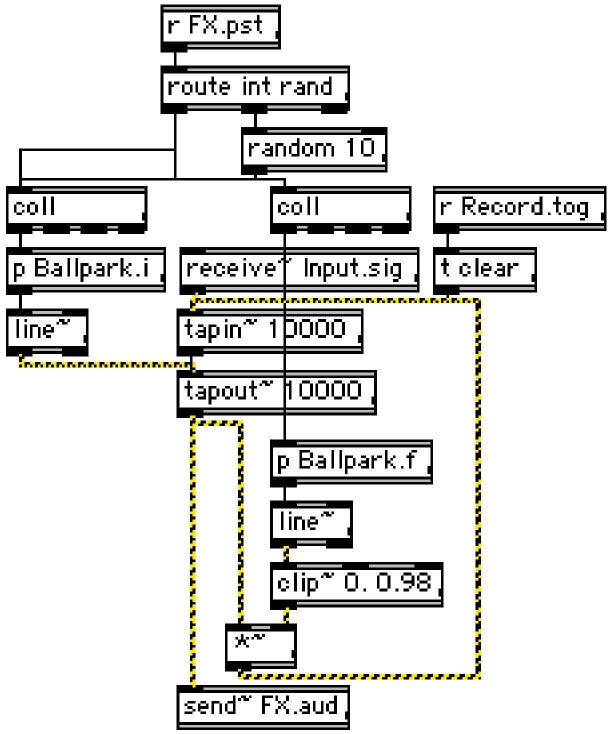

FIGURE 2.11. The FX subpatch applies delay effects to the incoming signal with gradually changing delay period and feedback settings stored in respective coll objects.

delay can be used to create many diverse effects, including not only echo and reverberation, but also chorus and flange effects, comb filters, and speed change with transposition. I designed the earliest version of this module to demonstrate these concepts to students learning to program in Max/MSP. Unlike presets in most special effect units, the presets used here include delay period and feedback values that change over time. Whereas static preset values would each yield a single static textural effect, the presets defined in this module result in textures that evolve over time or form gestures or motivic material. The FX voice is the first to be heard from the program, because the other voices take more time to capture enough material from the performer in order to begin sounding. It is often the most active, responsive, or distinct voice in the performance because it processes the streaming input from the performer instead of relying on larger buffers.

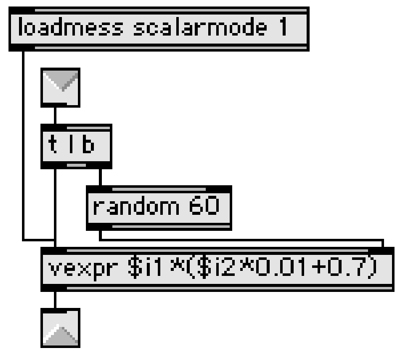

FIGURE 2.12. The Ballpark abstraction randomly adjusts the incoming values within a thirty percent margin, allowing for variety each time a scene is triggered. The .i and .f suffixes correspond to integer and floating-point calculations, respectively.

Transit

Each of the three voice types above are passed through an individual Transit module that builds a stereophonic texture by simulating effects of motion at great speeds, inspired by the effects of special relativity. This is not a straightforward panorama process that merely makes a sound louder in one channel than in the other, giving the illusion that the sound originates from a point between the left and right extremes but closer to the louder sounding channel. The Transit module is instead a textural device exploring the effects of motion among multiple reference frames. Like a panorama algorithm, the input sound is first placed and moved according to randomized procedures along a line segment in an imaginary space. The left and right outputs correspond to reference frames near the left and right ends, and the loudness of the sound in each channel varies inversely and exponentially with the distance between the imaginary position of the input source and the position of each reference frame.

The similarity to a traditional panorama algorithm, however, ends there. The left and right reference frames are not at the ends of the line segment, but are halfway between the center and an end. The loudness of the input is scaled so that it is loudest when it is at the position of a given reference frame and it is silent beyond the position of the opposite reference frame. If the imaginary position of the input is between the two reference frames,

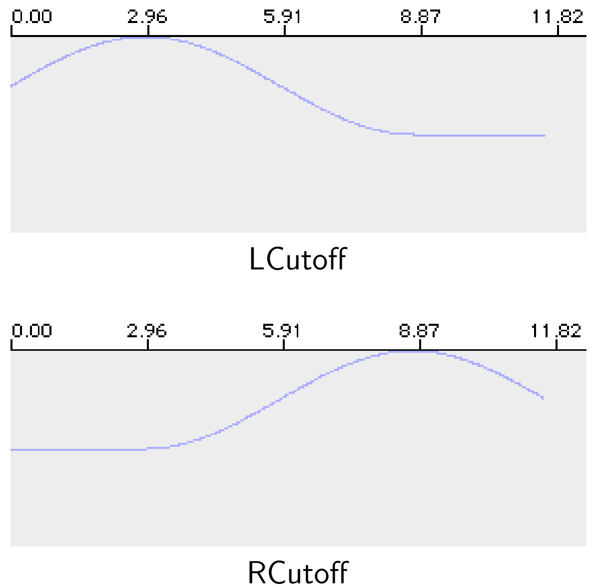

FIGURE 2.13. The LCutoff and RCutoff tables dictate the cutoff frequency of the low-pass filters in the left and right reference frames, respectively, corresponding to the position of the point source in the Transit abstraction. When the imaginary position of a sound is equal to the position of a reference frame, the full sound will be passed unaltered; as the sound is moved away from a reference point, the filter cutoff frequency falls to zero. The horizontal axes are labeled in milliseconds, but the tables can be read at any speed in either direction.

it will be heard to some extent in each channel. If it is between one reference frame and the nearest end of the line segment, however, it will only be heard in one channel.

As a sound moves away from a reference point, high frequencies are attenuated by a low-pass filter, simulating this effect of distance in similar acoustic situations. For example, a sound positioned to the left of center will be louder in the left channel, and it will also be brighter in the left channel. As the sound's imaginary position moves away from a reference frame toward one end of the line segment, it will become softer and also darker or muddier.

This allows for a variety of effects while avoiding a cluttered stereophonic texture. It is possible for voices to end up completely on one side or the other, since the reference frames are away from the left and right ends of the line segment. Panoramic extremes, especially when balanced with similar materials on each side, can yield a very vivid and intriguing overall texture and clarify complex contrapuntal textures.

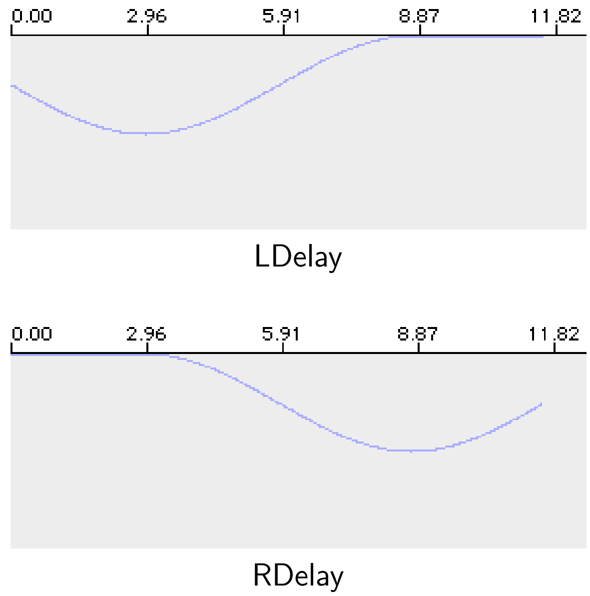

FIGURE 2.14. The LDelay and RDelay tables dictate the delay period of the left and right reference frames, respectively, corresponding to the position of the point source in the Transit abstraction. When the imaginary position of a sound is equal to the position of a reference frame, the full sound will be passed immediately; as the sound is moved away from a reference point, the delay period increases to ten seconds. The horizontal axes are labeled in milliseconds, but the tables can be read at any speed in either direction.

The distance between a sound source and a reference frame also creates a delay between the time a sound is played and the time it is heard. Building on the example given above, a sound positioned to the left of center will sound louder and brighter in the left channel and also sooner than in the right channel. Assuming an average speed of sound in air, the imaginary line segment would be over four miles long (approximately seven kilometers), the reference frames would be over two miles from each other (approximately 3.5 kilometers), and the input sound would move at speeds up to 14,400 miles per hour (approximately 23,000 kilometers per hour).

The program was not designed around these physical figures, however. These values were inferred from the possible range of delay times and durations of motion that were chosen for the resulting sonic effects. Also for the sake of the composition, the loudness of more distant sounds has been exaggerated to make the effects audible, much in the way theoretical physicists often imagine being able to see across vast distances in thought experiments on the effects of special relativity.[2]

This temporal distortion adds to the overall musical texture. Near the center, the difference in delay is short, and will thicken the texture somewhat. Farther from the center, delays can become long enough to create stereophonic ping-pong echo effects or canons on even larger scales. Large dimensions allow for extreme speeds, which yield significant delays and Doppler shifts as well: upward or downward shifts in pitch and playback rate corresponding to a sound moving toward or away from a reference frame, respectively. The Doppler shifts impose directional gestures on the recorded materials, allowing the software to contribute more actively to the performance. As the sound source moves in the area between reference frames, the sound will be heard as rising in pitch and speed in one channel and falling in the other channel, creating often fascinating contrapuntal moments.

The approach to musical signal processing described here thwarts the establishment of authenticity among the reproduced sounds. In traditional uses of panorama or delay, one instance is predominant (in loudness if not in other qualities as well), and the other instances clearly support and prolong the idea of a single sound played *now* in a particular point in space. The effects of the Transit module present multiple varied instances that refer back to a previous, perhaps quite distant, moment in the performance and integrate that moment into a new musical texture and context, sometimes with a new pitch contour imposed by the Doppler shift effect. This transformed reference may even suggest to a listener that the first instance was merely foreshadowing later musical developments.

A Special Effect

The Transit module also contains a subpatch that applies a special effect to the output of each voice when it is activated. It is a distortion effect based on randomized amplitude offsets and clipping. I discovered this technique when I fortuitously misused a filter while designing

[2] For classic examples and an excellent survey of philosophy and physics concerning the nature of time see Davies (2006).

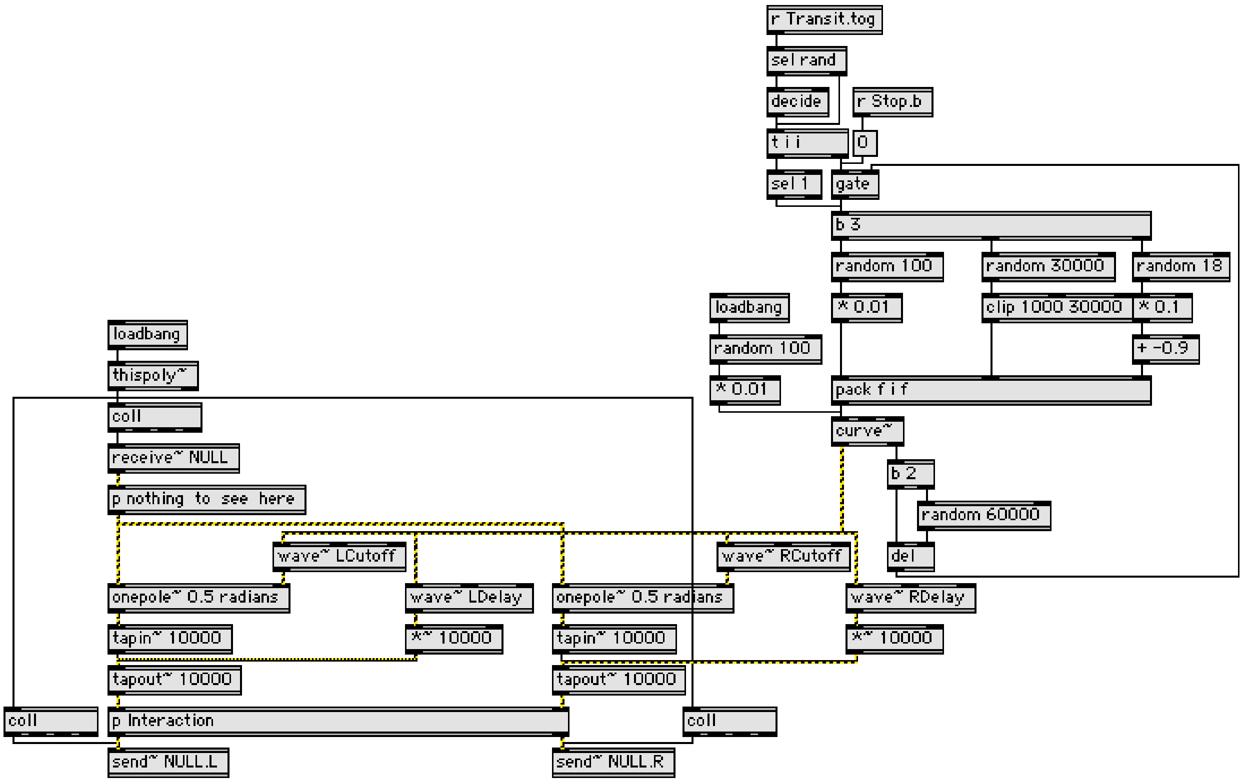

FIGURE 2.15. The Transit abstraction creates a stereophonic version of monophonic input by simulating an extreme Doppler effect. The input is a point source that is moved along a single line, and the output channels are symmetrical points of view along that line. *Note:* See figures 2.13 and 2.14 (page 24) for the functions stored in the buffers LCutoff, RCutoff, LDelay, and RDelay.

the program. For many digital filter algorithms, improper parameter settings cause a buffer to fill with absurd values which in turn feed future calculations, making the resulting sound crackle and break up before becoming terminally silent (informally called "blowing up"). However, in my flawed patch, the filter was capable of coming out of this "blown up" state and returning to normal. The sound is distinct and, although harsh, intriguing.

After finding and correcting my error, I discovered that the sound was caused by an amplitude offset that shifted the waveform of the audio signal out of the usable range (-1.0 to 1.0). Such offsets can damage audio hardware when they are converted into voltage (direct current offsets), but the program can safely handle any numerical values as long as they fit between -1.0 and 1.0 before reaching the digital-to-analog converter (DAC). I recreated the effect by adding a known offset to the audio signal, clipping it to the range -1.0 to 1.0, and then subtracting the offset. This method yields the same kind of distortion by monopolar clipping, but without any net amplitude offset in the output.

Because this effect was not an original part of the plan for the composition and it did not fit its concept, I implemented it in the style of a computer game “cheat code,” a string of characters (not officially documented) that allows a player to alter the game by typing it, for example, making the player’s character invincible. Such cheat codes are typically designed for use in the development stage so that a game need not be played seriously just to test certain parts of it, much like the keyboard commands described above in *Elektrodynamik.* Other cheat codes are designed as “Easter eggs”—(somewhat) secret features meant to enhance the user experience or add special content. The *dirty* effect is more like an Easter egg in these terms.

The effect is activated when the word *dirty* is typed. The offset changes automatically in a randomized process. It only affects the output of the sound, so that when it is deactivated (by typing *pretty*), the output returns to normal. Only the unaltered input is recorded, so passages recorded while the effect was active may be played back cleanly later, and vice versa. This effect further distorts the notion of causality or authenticity among reappearing sound material. Since the effect is applied during playback and not during recording, a distorted gesture may be heard from the computer before the unprocessed originally recorded gesture is heard. It also can dramatically transform a musical texture from dense to sparse, or from pleasant to harsh.

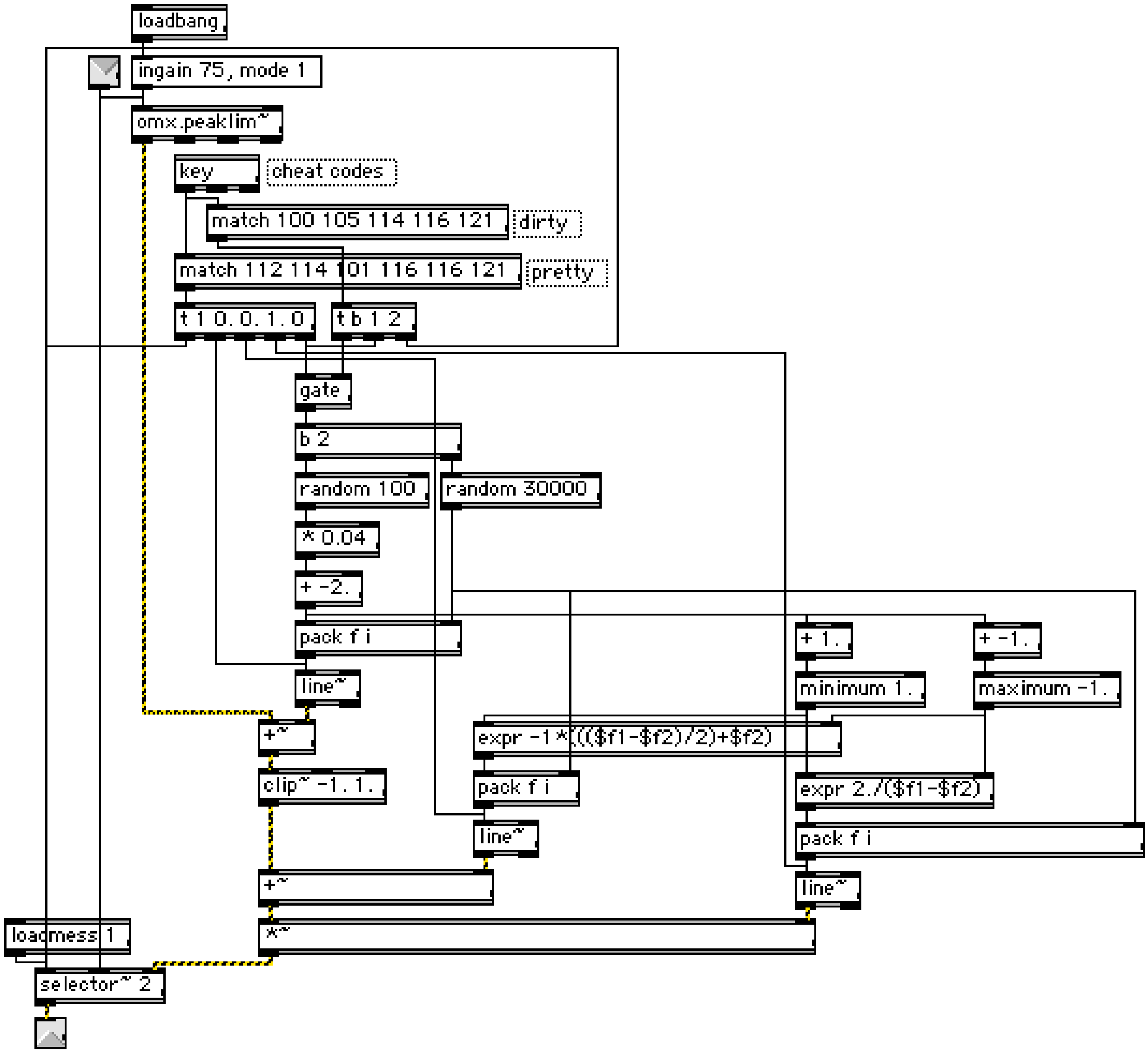

FIGURE 2.16. The subpatch nothing_to_see_here activates and deactivates the *dirty* distortion effect when keys are pressed in the right sequence.

Interaction Modes

Within the Transit module, the Interaction subpatch uses amplitude envelopes to feign one of four modes of interaction with the input signal:

Independent: Ignore the loudness of the input signal.

Similar: Match the loudness of the input signal.

Complementary: Set loudness to complement the loudness of the input signal, i.e., silent when the performer plays at full loudness, loud when the performer plays softly, etc.

Rhythmic canon: Follow the loudness of the input signal delayed by five seconds.

A scene may specify an interaction mode for all voices or specify that the mode for each voice should be selected randomly, allowing for contrapuntal behavior among the voices of the program. This section exemplifies the overall approach to machine "intelligence" in these works. While it is possible to enable the program to make more sophisticated assessments of the performer's activity, including, for example, average phrase length or seeking patterns within, the simple approach used here has proven to be sufficient and quite satisfactory for its purpose: textural variety and varied modes of response to the performer. If more sophisticated methods were used with the rest of the program as it is, much of that sophistication would go unnoticed. If it were to make better use of more sophisticated routines, it would be a different composition. In my judgement, this program only needs to "know" this much about the performer's input in order to achieve the desired result. This is in stark contrast with George Lewis's *Voyager* and other works that attempt to equip the program with a more thorough representation of the input from the performer.

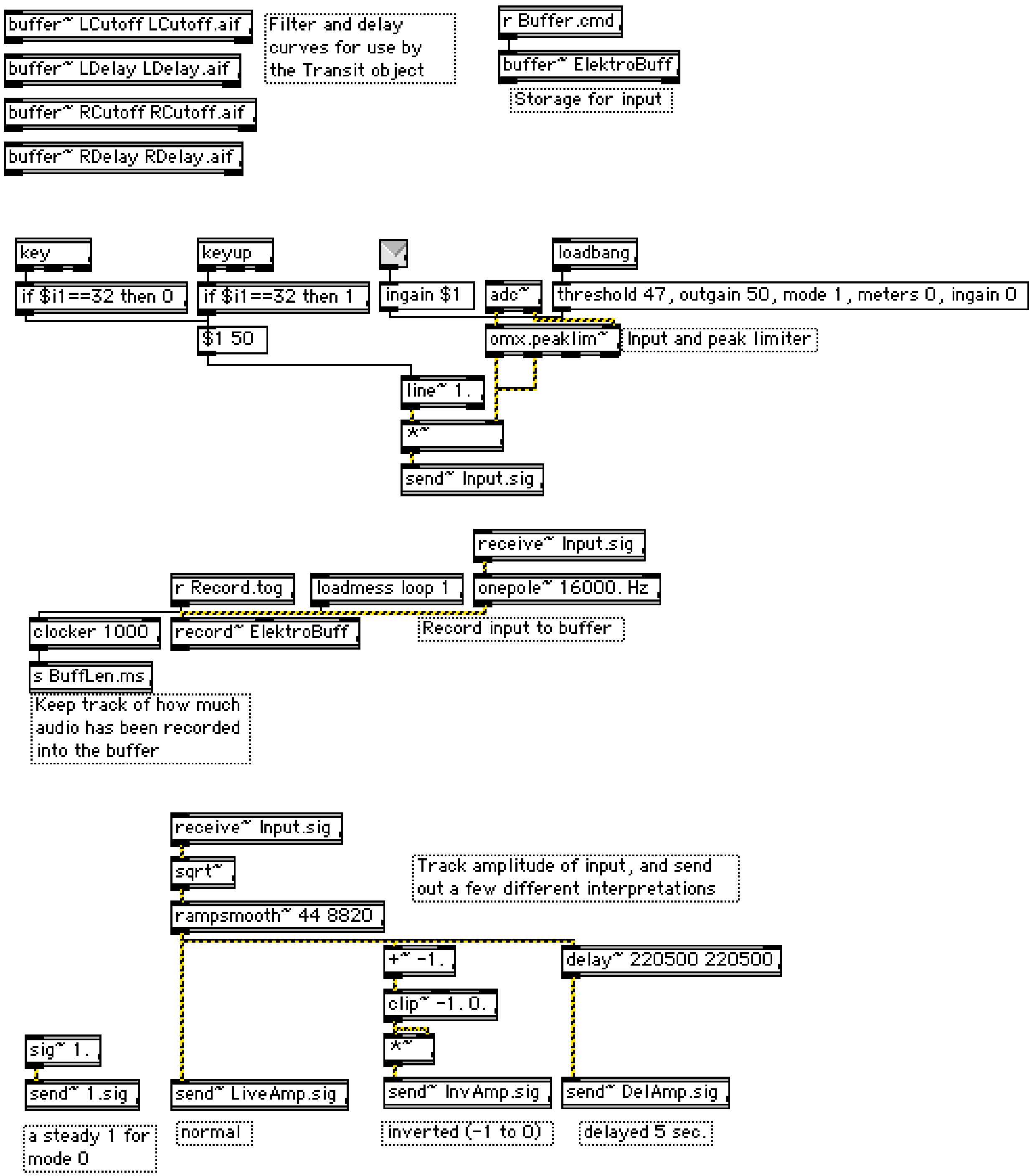

FIGURE 2.17. The Input subpatch contains several unconnected but related patches. *Note:* See figures 2.13 and 2.14 (page 24) for the functions stored in the buffers LCutoff, RCutoff, LDelay, and RDelay.

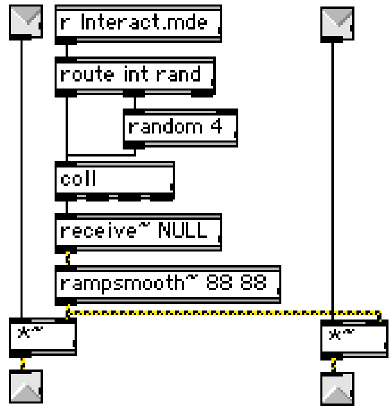

FIGURE 2.18. The Interaction subpatch applies the selected amplitude envelope to the input signal.

Musical Description

Use with a Soloist

This work has achieved the most satisfying results in performance with solo melody instruments. Playing chords or dense textures as an input to the system often yields textures that are too dense to appreciate. It is interesting how all three of these works tend to magnify the character of the performer's timbral range. For example, instruments exploring a wide range of timbres, as in Eric km Clark's (sic) violin performances, tend to exhibit a wide timbral range overall. Performances with instruments of less timbral variety, like harmonica or trumpet, allow the focus to shift to subtle variations in timbre and harmonies suggested by simultaneously sounding pitches.

Except for starting and ending, the program was designed to be largely unattended during performance. However, during development, a number of keyboard controls were added to aid in testing the program in various situations, and these controls have occasionally proven useful to the performer or to a dedicated computer attendant. In most cases so far, performers have been content to let the program operate unattended unless I was present to direct the computer.

Use in an Ensemble

Elektrodynamik has been used in a trio with piano and electric guitar, and the role of the program was different than performances with soloists. Because the program was created with one audio input, it could only sample the other two instruments together without intervention at the mixer controlling the input level from each instrument. This had the effect of acting as an extension of the piano-guitar duo, rather than being an equal contributor in a trio.

The FX module was the most useful in this performance, because it is more directly responsive. The causes and effects were more apparent to the performers, so they could learn to coax certain results from it. I found that it was best for the ensemble to begin with a loud sound in order to give the Player and Beat voices strong material to begin with. Performances that began softly yielded soft gestures from the electronics, which kept the other performers playing soft in order to hear them. After the first minute of the performance, the Beat voice became useful, but it is not sufficiently versatile to serve as a main part of the music unless it were set up to dominate the performance as an ostinato. The Players voices were difficult to fit in with the ensemble, because no controls are available to make sound happen at given times.

I have pondered a kind of "do something now" command, and may include something like it in a future version. I have used the Change command like this, since most scenes will force new gestures to be triggered soon after the scene begins. If sound is already playing, it can, of course, be shaped by taking control of the fader for one voice or the main output fader.

I do not see the observations made during this event as weaknesses, because the work is a composition and not a complete replacement for a human performer. These properties could be seen as emergent directions from an invisible score. They are behavioral tendencies that are part of the character of the composition.

Other Possible Applications

Good results have been achieved using white noise or a sine tone as the sole input, but these versions have not been presented publicly. Even though the input is not live (or rather, it does not involve live decisions), the variety and control possible through the user interface make it worthy of live performance by itself.

Elektrodynamik has not yet been used in a public installation, but it would be effective in such a setting, either using prerecorded sound or a live microphone for input. It can be left unattended indefinitely, even with a feedback loop between the speaker and microphone, because it constantly changes its state, preventing any resonance from continuing out of control and dominating the presentation. The scene (containing most parameter settings) is changed automatically after a designated time range, so the system only maintains a given state or a limited duration. The only chances for feedback to dominate the presentation are created by the FX module with short delay periods and electroacoustic feedback or with a comb filter that happens to have several intersections with the spectrum of the input. These modes could be avoided in an installation presentation, or made to occur only in situations that will not get out of control or last very long. These would be simple adjustments, and it could be easily tailored for many environments.

I used a similar technique in the live audio processing of *Platonic 5*, a three-day long sculpture-performance by Paolo Piscitelli, a resident artist of the Texas A&M University College of Architecture, September 26–28, 2006 at the Texas A&M University Riverside campus. Two pressure zone microphones (PZM, or boundary microphones) were placed in the small performance area along with seven small loudspeakers and a subwoofer. Each of the seven main speakers played a delay loop passing through a resonant bandpass filter fed by one of the two microphones. Short delay periods or high resonance settings would allow the system to be dominated by feedback resonance, but these values were periodically changed for different textural effects. To keep the sound under control in these moments, and also to prevent the processed sound from dominating the performance, an amplitude envelope was used on the input signals, acting like a camera shutter. Separate opening and closing

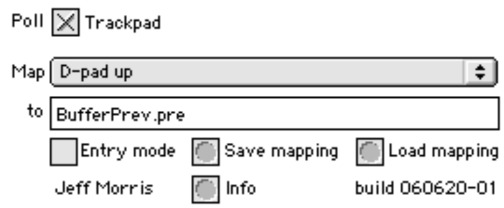

FIGURE 2.19. The Map abstraction interface from Gamepad Sampler.

speeds allowed more textural control yielding sounds with attacks and releases changing independently. In moments when other parameters might let unruly resonances appear, the overall duration of the envelope, like a shutter speed, was shortened, successfully stifling any unwanted results.

Future Work

Elektrodynamik could be easier for performers to add to their repertoire with the addition of a routine that would allow performers to specify their own controls to the interface. This would allow a performer to adapt the interface to any peripheral. The routine would be similar to the Map module used to map controls to functions in Gamepad Sampler (see figures 2.19–2.20). In this routine, the user would select a function from a menu (e.g., Change Scene, Beat On/Off) and press the desired control on a text keyboard or MIDI or USB controller to associate that control and result. A MIDI pedal board would be a convenient way for performers on most instruments to control the program.

A visual component could be a compelling addition to the work. With the Jitter library added to Max/MSP, video delay effects corresponding to the audio delay effects described above could be added to the performance. The visual and audio components could be coordinated in a few different ways. Most simply, the settings in each scene could be sent to both the audio and video processing sides, so that the behavior of the visuals would be congruent with that of the sounds. Perhaps the next most simple approach would be to allow the audio and video components to use the same set of scenes, but run and select

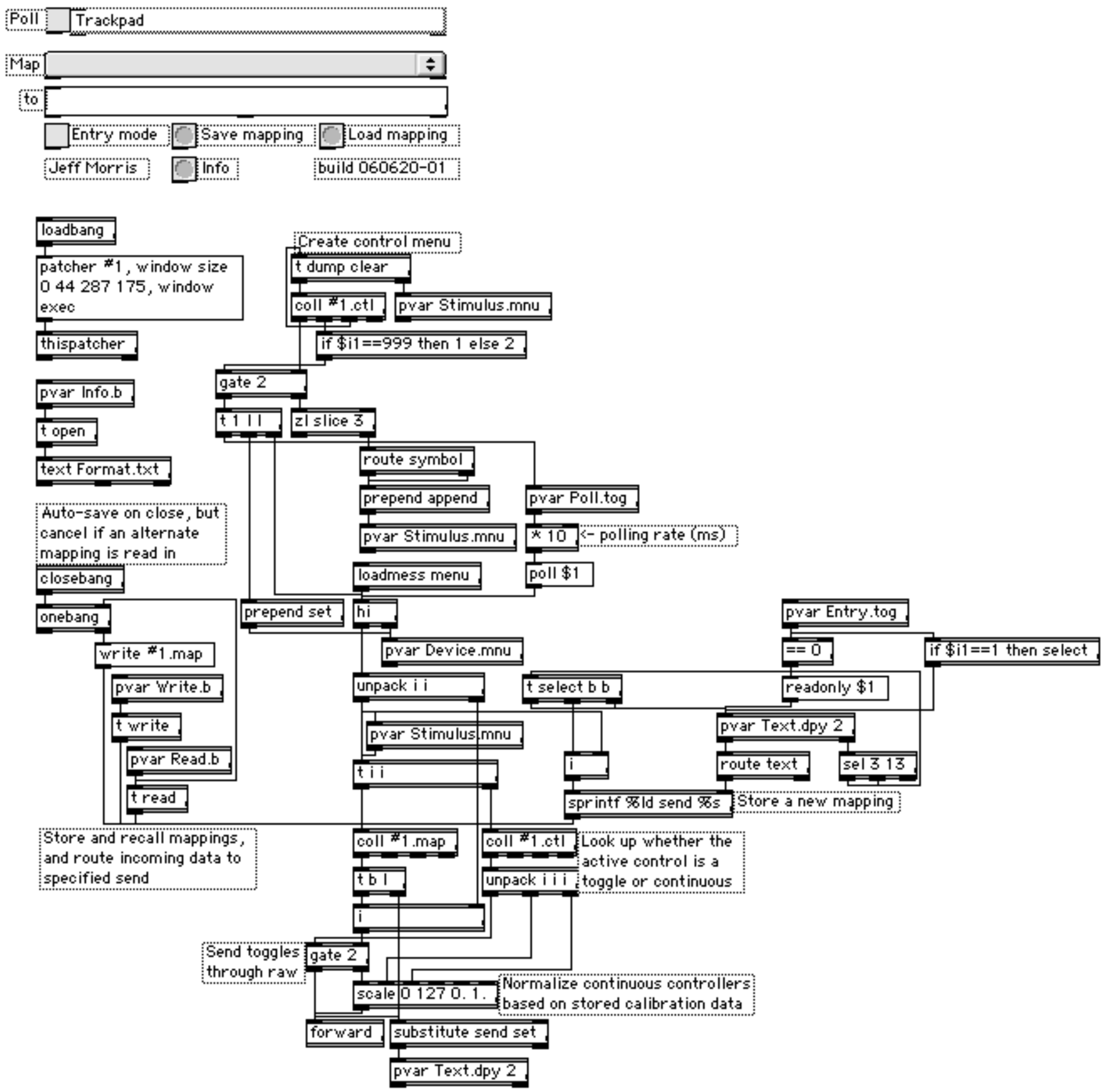

FIGURE 2.20. The main patch of the Map abstraction from Gamepad Sampler including elements hidden from the user.

them independently. This would allow for a counterpoint between visuals and sounds, but it would be rare that the two would be on the same scene simultaneously.

Expanding on the first option, separate video settings could be added to each scene, allowing the composer to establish a line of states in audiovisual counterpoint. Finally, a more advanced approach would be to allow the video and audio components to change scenes independently most of the time, but the video component could be made aware of the current scene in the audio component, and when a scene in the video component ends, it could be made to join the scene corresponding to the current audio scene. Additionally, some audio scenes could contain commands to force the video scene to change when the audio scene is activated. This would allow for more balanced and varied tension of congruence and independence between the audio and visual components.

In this development, video source material must also be addressed. It would be possible to load one or many prerecorded video clips to use as input, however, it would be more coherent to use live video sampling. A camera could be set up on the performer, or on a dance artist joining the performance. Drawing on a technique developed at CEMI,[3] a camera could be focused on a portion of the soloist. The result is a less identifiable image derived from the action on stage, with most movements associated with sonic events, but not necessarily in a straightforward way.

The addition of visuals as described above would be an effective way to extend the work into the realm of intermedia, possibly being viewed as a study in audiovisual or audiokinetic synaesthesia.

[3] The Center for Experimental Music and Intermedia at the University of North Texas. This technique was developed by CEMI director Joseph "Butch" Rovan with staff Peter McCulloch, Jon Anderson, and me for a concert entitled *Cello Machine* performed by cellist Ulrich Maiß in the Merrill Ellis Intermedia Theater at the University of North Texas on October 18, 2003. This version of the technique used a digital video camera focused on a portion of the performer fed into an LCD projector aimed at a crumpled sheet of mylar, which reflected the abstracted, fractured image onto a large screen above the performer.

CHAPTER 3

MOTET

Motet is a reflection on more straightforward imitative techniques, based on delays and pitch shifters. It operates and makes minor changes without user interaction, but depends on the user for large-scale form and tightly-coordinated moments. In this environment, the user specifies more (but not complete) detail about the delay periods of each of the computerized voices and in this way acts as a live arranger. The performer may focus the computerized voices on various relative times within the performance:

Near: Music in the past few seconds (stretto).

Early: Near the beginning.

Same: Around a randomly chosen point.

Wide: Each voice at a different randomly chosen point in the performance.

Now: Zero delay, acting, for the moment, as a harmonizer.

The voices automatically wander away from the given points some time after being cued, relieving the performer of the responsibility of triggering every change, but this ability allows the performer to plan and cue dramatic textural changes. Since the sampled sounds are only changed in pitch, the performer is challenged to create and play strong melodic lines that lend themselves well to classic imitation, as well as assuming a variety of musical roles throughout the performance.

Technical Description

Each Voice module consists of one delay line and one pitch shifter and uses a randomized procedure to select transposition level, delay period, and when to change either of these values. All time changes are quantized to a central tempo, which is by default set to 300 milliseconds (200 beats per minute). This tempo may not always be obvious, but it allows

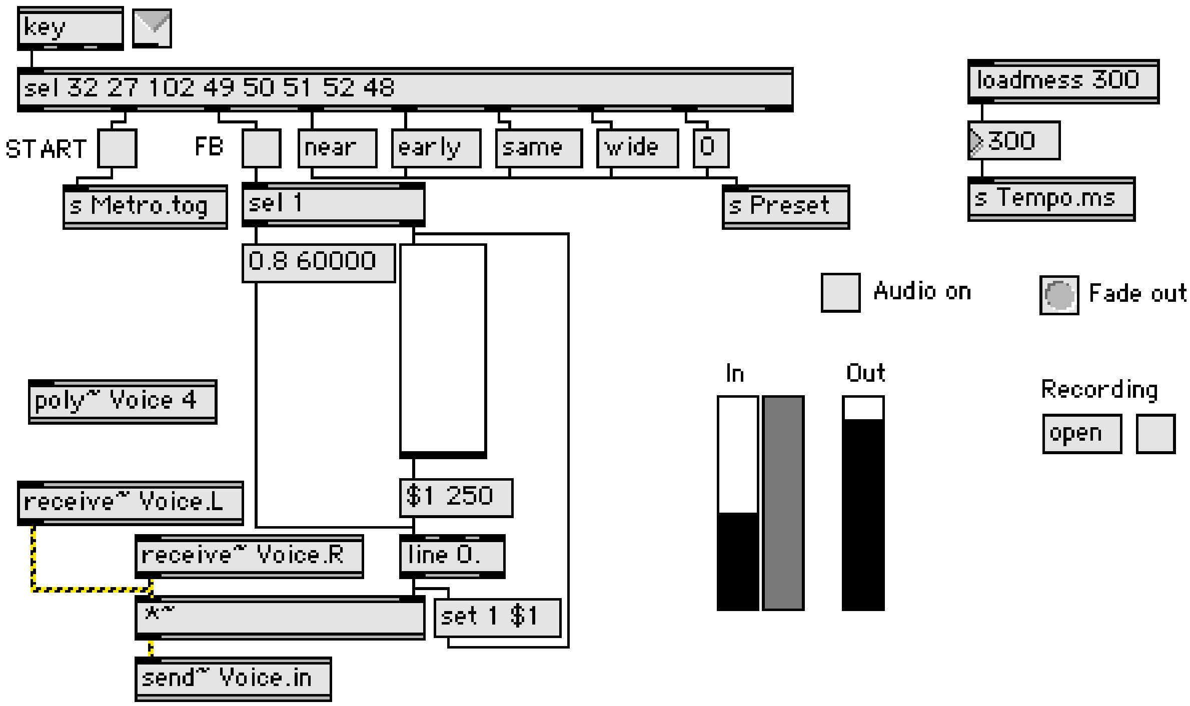

FIGURE 3.1. The user interface of *Motet* includes input and output volume controls, five delay presets, a feedback control, a button to trigger a slow fade out, a switch to activate audio processing, a switch to begin the performance, and controls to record the performance as an audio file.

for rhythmic patterns among voices and a degree of textural clarity or a metric feel when material is played in the tempo.

The pitch shifter is randomized without any user control. It also implements a spectral gate for noise reduction in the input signal. With higher values, it also causes the electronic voices to audibly deteriorate as they are copied from the soloist. When feedback is used (in any form), this degradation effect can continue progressively.

A slider is provided to fade in feedback from the output of the voices to the input of the delay line. Since this happens in the digital realm or in the domain of data, this might be called *digital feedback* or *data feedback*, as opposed to *electronic* and *electroacoustic feedback* (Meyers, 2002). The feedback amount can be controlled directly or by clicking a toggle switch triggers a gradual rise in the feedback level, quickly returning it to zero when clicked again. This feedback multiplies the number of voices and the textural and registral effects

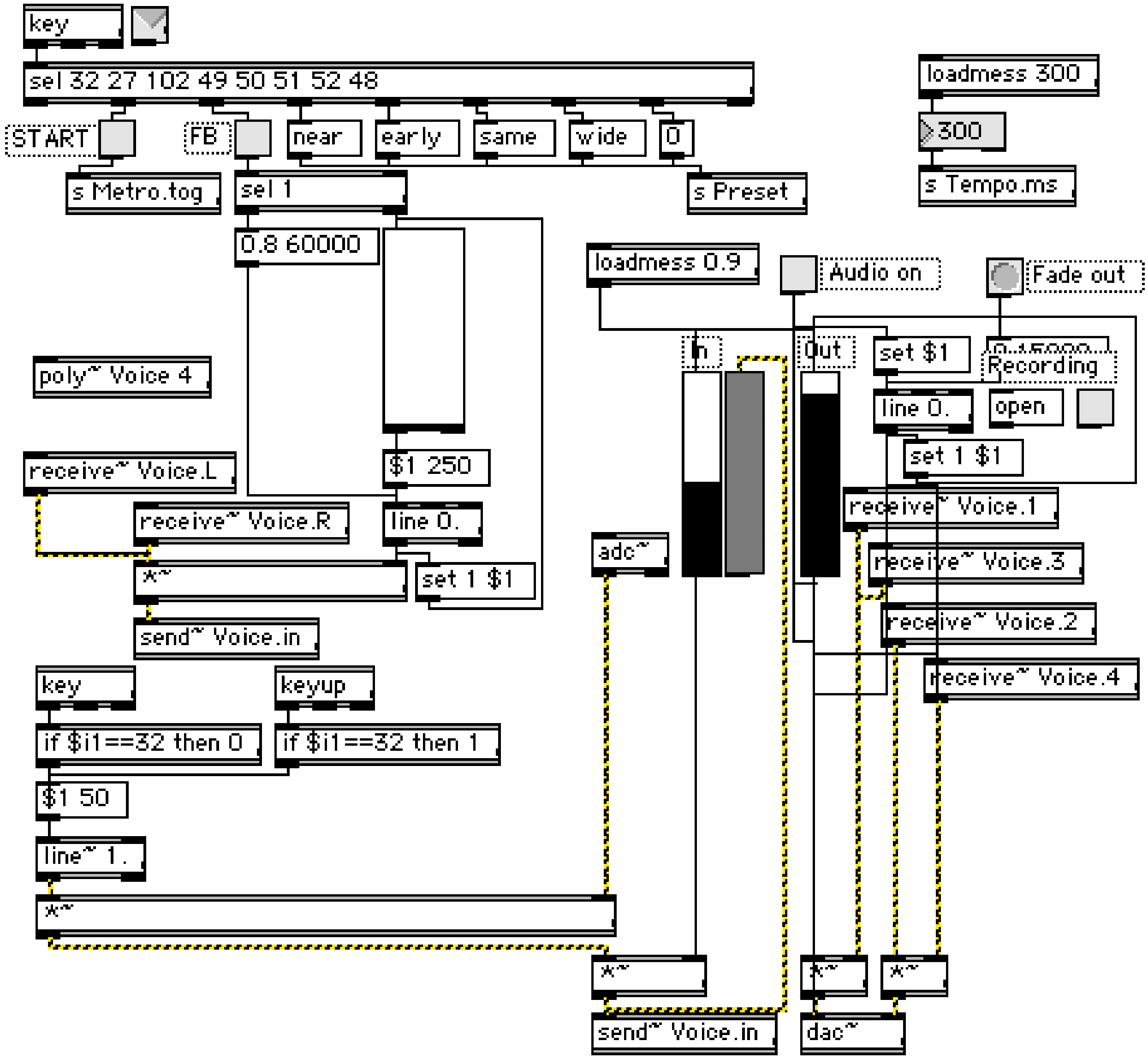

FIGURE 3.2. The main patch of *Motet* including elements hidden from the user: objects for audio input and output, mixing, and control automation.

of their delay and pitch shifter settings. It can be effective for building to climaxes, or it can help thicken sparse textures if left at a moderate level.

Musical Description

Like *Elektrodynamik, Motet* has had the best results with melody instruments, and the program may be operated by the soloist or by someone else. Unlike *Elektrodynamik, Motet* is not meant to run without occasional control by the performer. The presets offer a variety

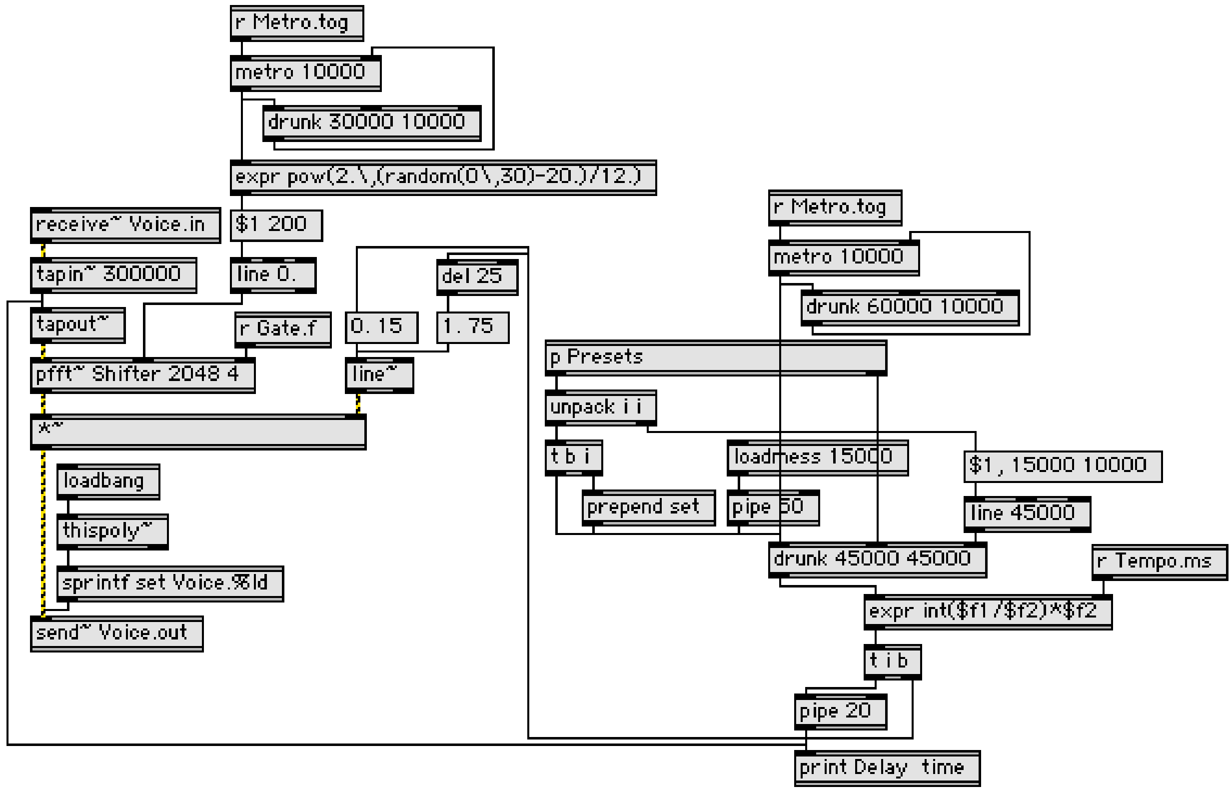

FIGURE 3.3. The Voice abstraction contains a single delay line and pitch shifter.

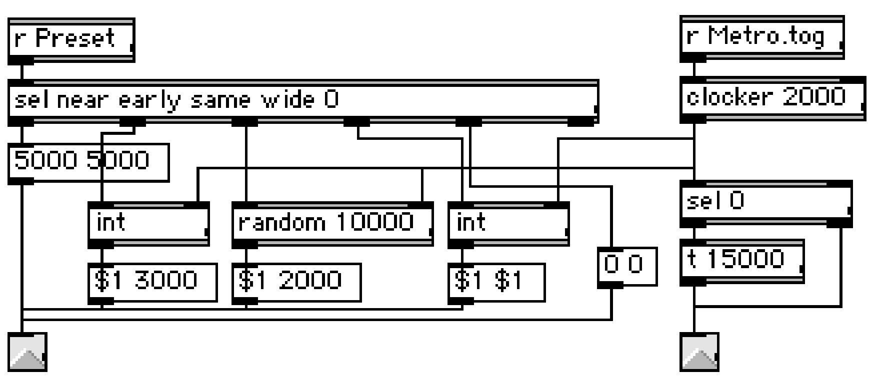

FIGURE 3.4. The Presets subpatch sends parameters associated with the user presets when triggered.

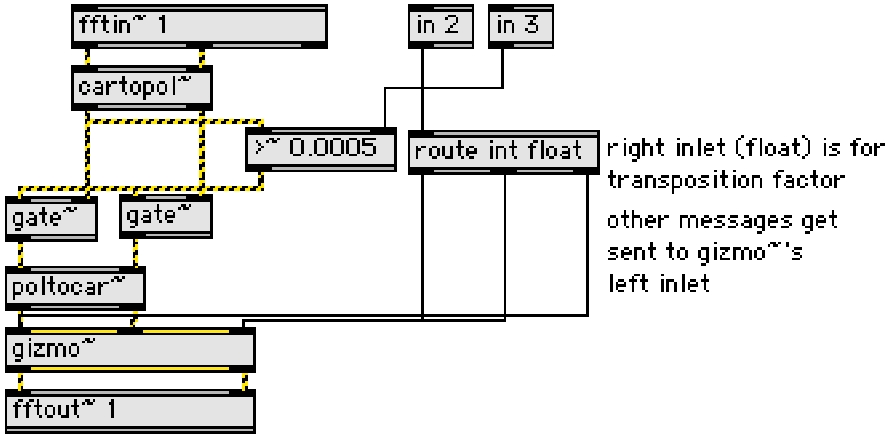

FIGURE 3.5. The Shifter abstraction transposes the input in the frequency domain with the gizmo~ object.

of textural situations. Each voice does roam on its own over time, but it is very unlikely that most of the voices would end up simulating the presets on their own. So far, only I have controlled *Motet* while performing (on trumpet, slide banjo with feedback, and harmonica accompanied by a bassist).

Because of potential contrapuntal complexity of *Motet* with a melodic soloist, it has been difficult to get good results with an entire ensemble as an input. However, it has been used well in small ensembles with one instrument as an input.

The piece begins either with long delays between voices, as in many imitative motets, or occasionally in homorhythm; this is often effective to use as an introduction, which can later be broken into individual voices. The following are techniques I have found to be musically effective.

- Triggering the Near or Now setting and playing isolated notes or very brief gestures will reveal an underlying pulse. When the tempo becomes clear, sparse gestures played in tempo can be transformed into shifting metric patterns, and driving pulses with occasional accents can create an exciting frenetic texture.
- A multilayered call-and-response texture or cascading rhythmic sequence can be established with the Near setting by alternating between two types of activity every couple of seconds; e.g., long and short notes, or playing and resting.

- The Near setting is effective in building up tension focused on a single musical idea, which can later burst into a vivid texture by triggering the Wide setting at a climax.
- If the first material played is carefully planned, the Early setting can be used to trigger a recapitulation at any time.
- The Same setting is useful for setting off the soloist from the electronic voices. For example, if the voices focus on more subtle material, this setting can serve as an accompaniment and feature the soloist. If the voices focus on an area with dramatic material, the soloist can respond to, struggle with, or extend the gestures played by the electronic voices. This could effectively be followed by the Near setting, which would allow all voices to gradually develop the previously played material (because it had also recently been played by the soloist).
- The Wide setting can allow electronic voices to become soloists, letting the human soloist rest or join the texture in a supporting role.
- The Now setting can be effective when triggered briefly on occasion, in order to emphasize certain gestures, or to build intensity.
- Playing or resting too much will cause the same to occur in the computer voices, making contrapuntal variety difficult to achieve.

Future Work

Performance without a dedicated computer attendant would be made easier by enabling the performer to associate the controls of the program with a standard control peripheral, such as a MIDI pedal board, as discussed above (page 35). It would also be useful for the performer to see a representation of the recorded sound and the portions about to be played by the Voice modules. This would enable the performer to better integrate newly played material with material repeated by the voices. Such a display could also enable the performer to visually direct Voice modules to certain portions of the recorded performance, giving the performer much more control over the average delay time and range of variations than is offered by the presets. A foot pedal could be used to good effect with a graphic display to demarcate points in time as they happen and recall them at later points in the performance.

It would be reasonable to compose a fully notated score including directions to control the program, which would “thicken” the composition ontologically (see the discussion beginning on page 66). Because each Voice module changes its settings on its own, it would be possible to set up *Motet* as an installation. However, since the Voice modules are only intended to wander around a designated point, a new agent in the program would be needed, such as the Scenes subpatch in *Elektrodynamik* that could automatically trigger presets, with drunk walks in preset selection and random duration selection that allows for the creation of more complex textural developments, as described in the previous subsection.

CHAPTER 4

GAMEPAD SAMPLER

Gamepad Sampler is more of an instrument than a composition, because the controls directly address performance events. For example, the user must specify when to start and stop playing each sample and set the loop points, volume, and other parameters. The instrument is comprised of four sample banks that allow capturing sound during performance and manipulating it expressively during playback. The primary control interface is a gamepad controller, but the mouse and a MIDI keyboard can also be used with it.

Technical Description

The loop points, loudness, pitch, speed, direction of playback, and panorama can all be controlled from the gamepad. A mouse can also be used to adjust these settings, and it can also allow the user to "scrub" through a sample, guiding playback through movement of the cursor. This also distorts the pitch and speed of the recording like manually moving a turntable or magnetic tape. A mouse or similar device (I prefer to use a thumb trackball with this program) enables the user to change settings instantly instead of gradually, and is better suited to fast or drastic changes. The strengths of the gamepad lie in its many controls in one compact space that may be accessed more quickly and without looking at them or the screen, the ability to manipulate several controls at once, and its ease of use while standing and moving about in performance.

At any moment, the user may load a bank with the past thirty seconds of sound from either of its two inputs or outputs. This way, the user need not waste energy deciding whether to record something about to happen or regretting missing an event. If the start and stop of a desired passage can be cued or predicted, the time between enabling recording and selecting a bank, if longer than one second, will also be used to set the selected playback

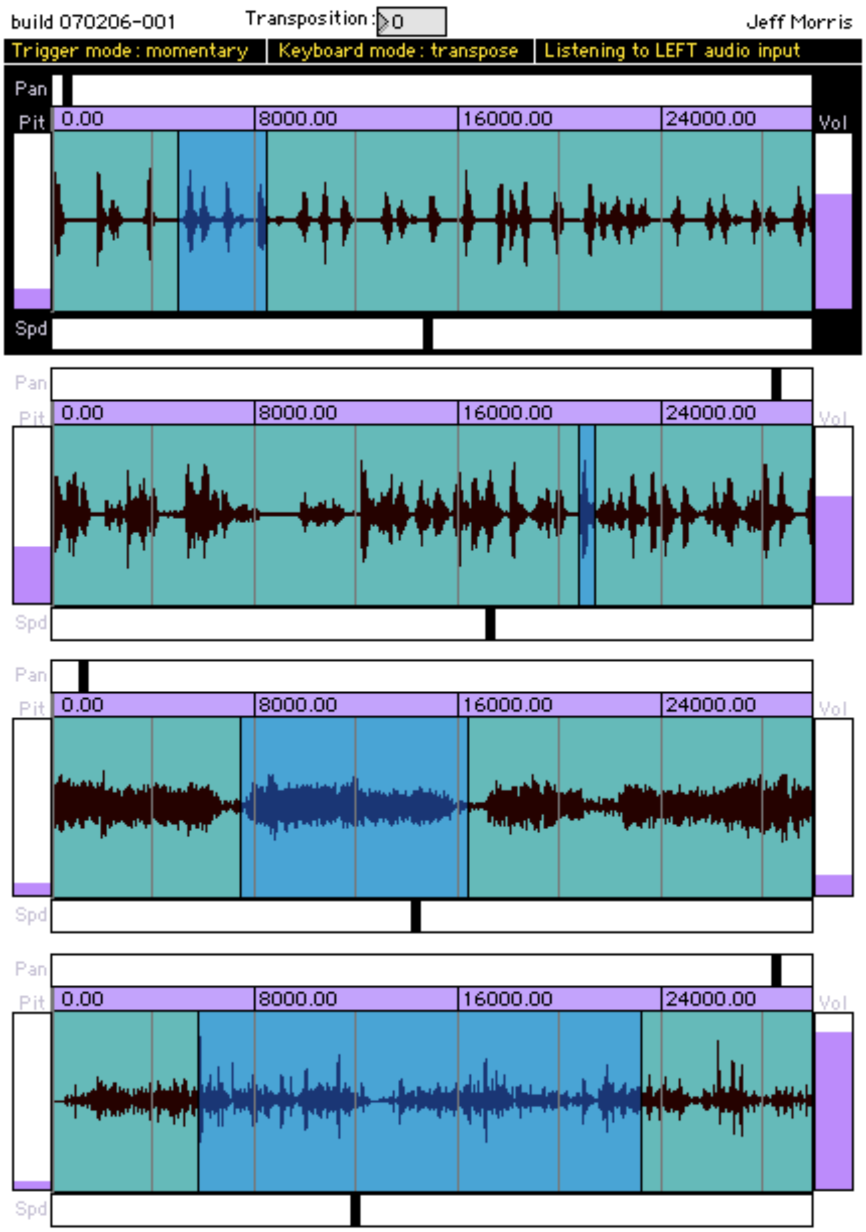

FIGURE 4.1. The user interface of Gamepad Sampler.

region. The user can then quickly begin playing the selection and adjust the start and stop times later if needed.

A MIDI keyboard can be used with Gamepad Sampler in two modes. In transpose mode, pressing a key sets the pitch transposition and loudness of the selected bank but does not start or stop playback. This is useful for "rewriting" sampled material by changing its transposition during playback. It can also allow for expressive changes in volume by pressing

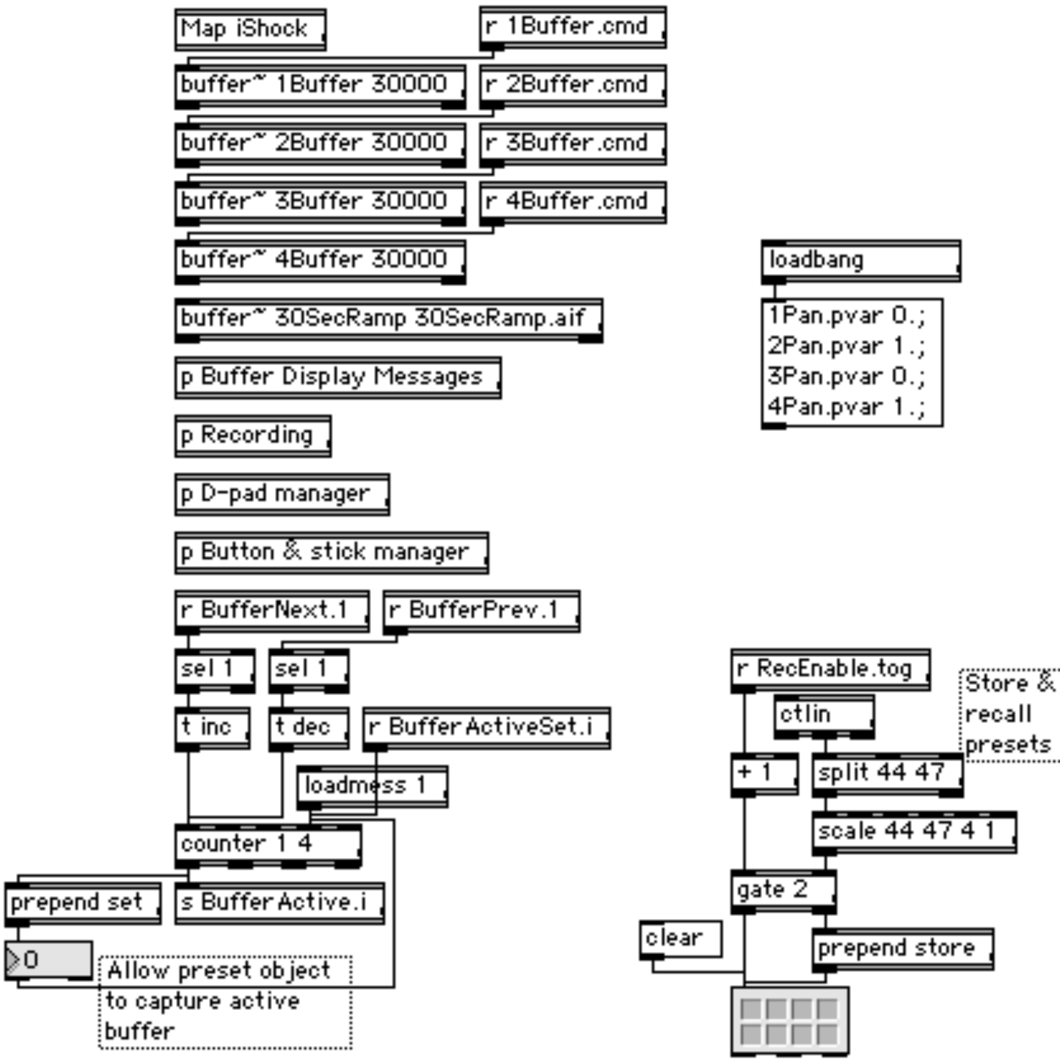

FIGURE 4.2. The main patch of Gamepad Sampler including elements hidden from the user: objects for mapping gamepad controls to program functions, buffers to store recorded audio and functions, initialize display settings, store presets, and subpatchers (1 of 3).

the same key with different velocities to change the loudness of certain notes or areas. In sampler mode, a key pressed starts playback in addition to setting the pitch transposition and loudness of the selected bank; playback is stopped when the key is released. Playback does not loop when the end of the selection is reached; it sustains the end of the selection with granular sampling instead. If another key is pressed before the first key is released, playback continues with new transposition and loudness settings. This allows time to be

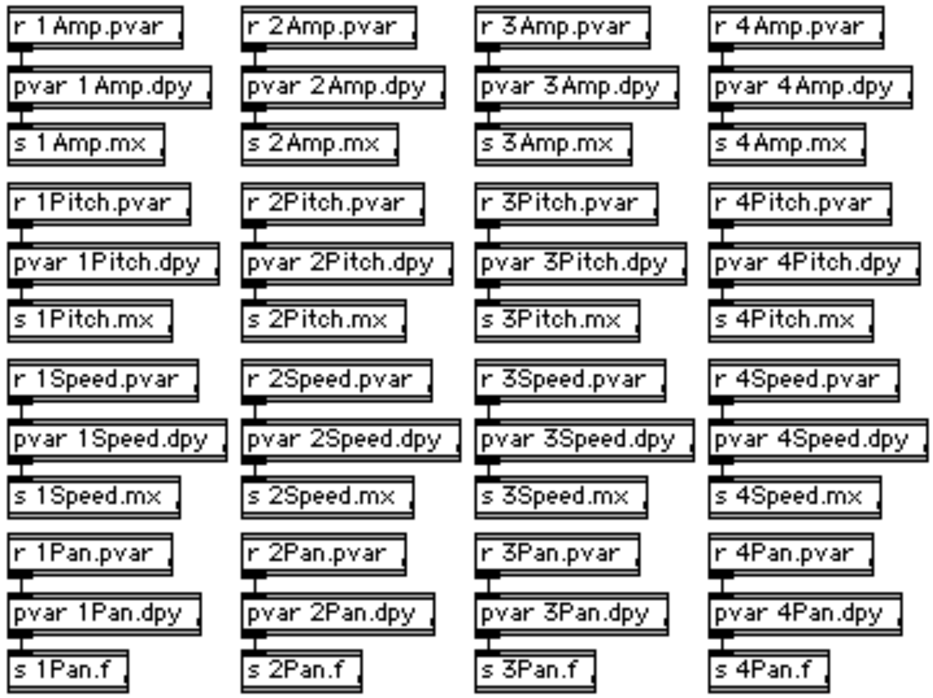

FIGURE 4.3. The main patch of Gamepad Sampler including elements hidden from the user: objects for transmitting controls to and from display elements (2 of 3).

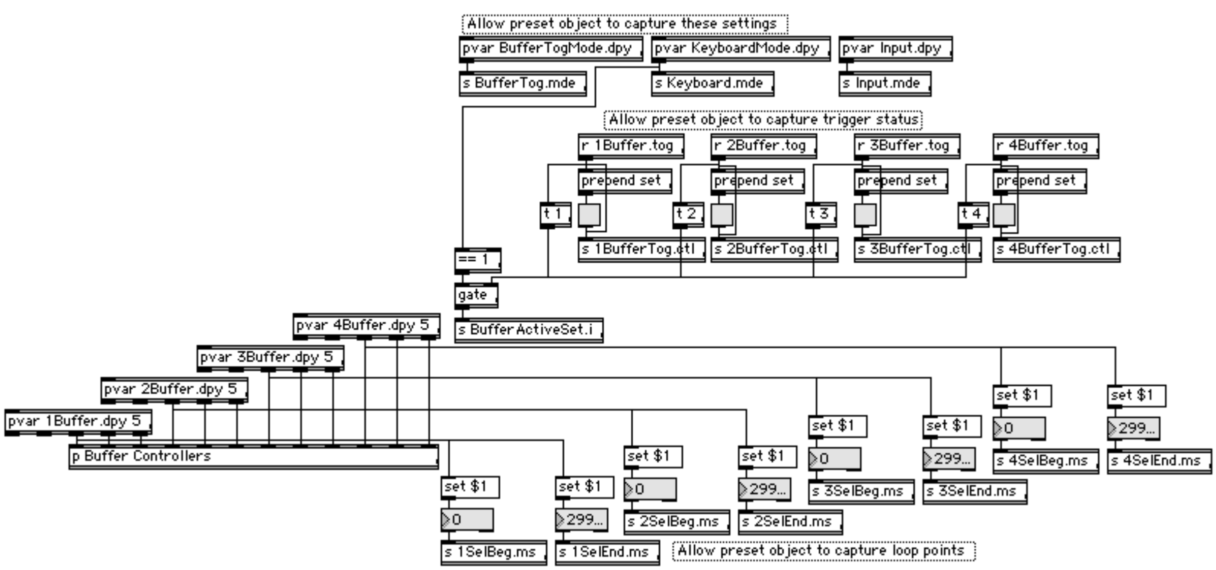

FIGURE 4.4. The main patch of Gamepad Sampler including elements hidden from the user: objects for transmitting controls to and from display elements and to allow controller states to be captured in presets (3 of 3).

TABLE 4.1. Gamepad controls.

Directional pad up/down: Select a bank to control.

Directional pad left/right: Change playback speed of selected bank.

Left analog stick up/down: Change pitch transposition of selected bank.

Left analog stick left/right: Change starting point of region selected for playback in selected bank.

Right analog stick up/down: Change loudness of selected bank.

Right analog stick left/right: Change end point of region selected for playback in selected bank.

Triggers (4): Each trigger starts or stops playback in one bank.

Left button in center section: Change behavior of triggers.
- **Momentary:** Trigger press starts playback, trigger release stops it.
- **Toggle:** Trigger press alternately starts and stops playback, and trigger release is ignored.

Center button in center section: Change mode of MIDI keyboard control (Transpose or Sampler).

Right button in center section: Select source to record from left, right, or monophonic audio input, or left, right, or monophonic sampler output.

Right thumb A button: Enable recording.

"frozen" at given moments and allows notes within a musical passage to be sustained past the length of the selected region.

Musical Description

The ability of Gamepad Sampler to independently loop four different selections as well as sampling and replaying its own output make it capable of creating very full textures on its own, given a sound source. The composition *RUhere* uses Gamepad Sampler alone with prerecorded samples. The samples were recorded in a studio conduction led by Butch Morris with Irina Botea, Lisa Cook, Maya Gurantz, and Olga Humphries reading poems they wrote based on a single starting line by poet Kelle Groom at the Atlantic Center for the Arts. The samples include whispered, sustained, detached, repeated, solo, and group versions.

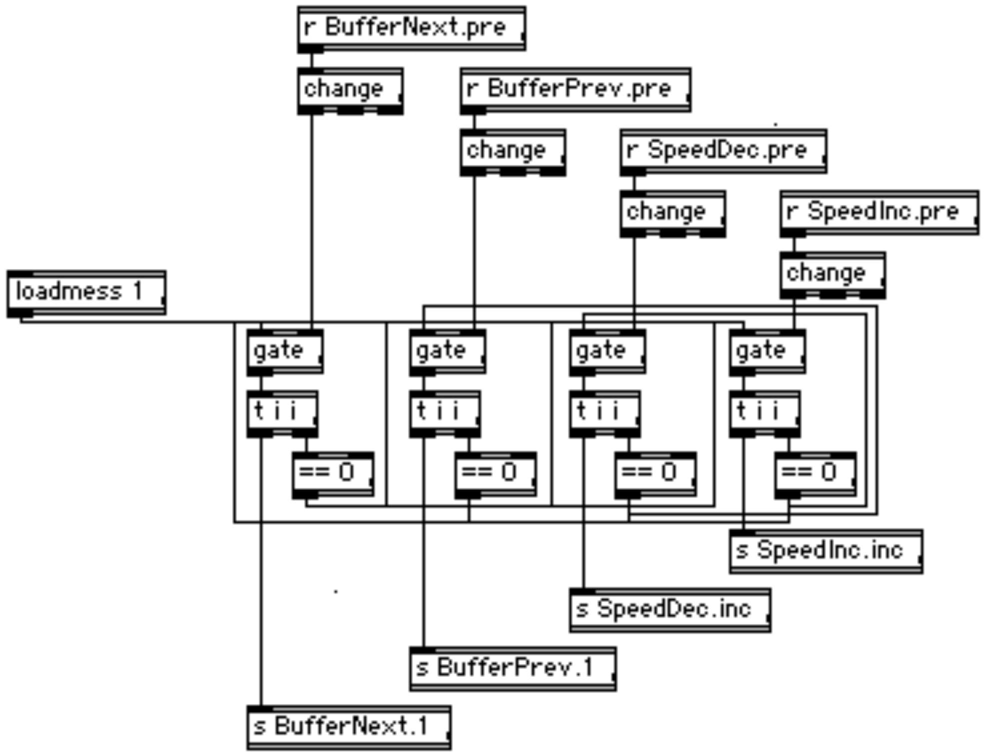

FIGURE 4.5. The D-pad manager subpatch forwards controls from the directional pad to set and manipulate the currently active bank.

Gamepad Sampler can also work very well with a solo instrument. Wide timbral variety from the solo instrument is helpful in performance. Instruments with narrow timbral ranges leave the Gamepad Sampler user to focus primarily on manipulating pitch structures, resulting in effects that tend to seem obvious and clichéd, like the "Chipmunk effect." Even unintentional brief noises can provide rich material to expand and develop in performance. During one performance, I neglected to unmute a microphone input on the mixer, but the electrical noise of the muted channel was sufficient as source material until I found a musically sensible opportunity to correct my oversight.

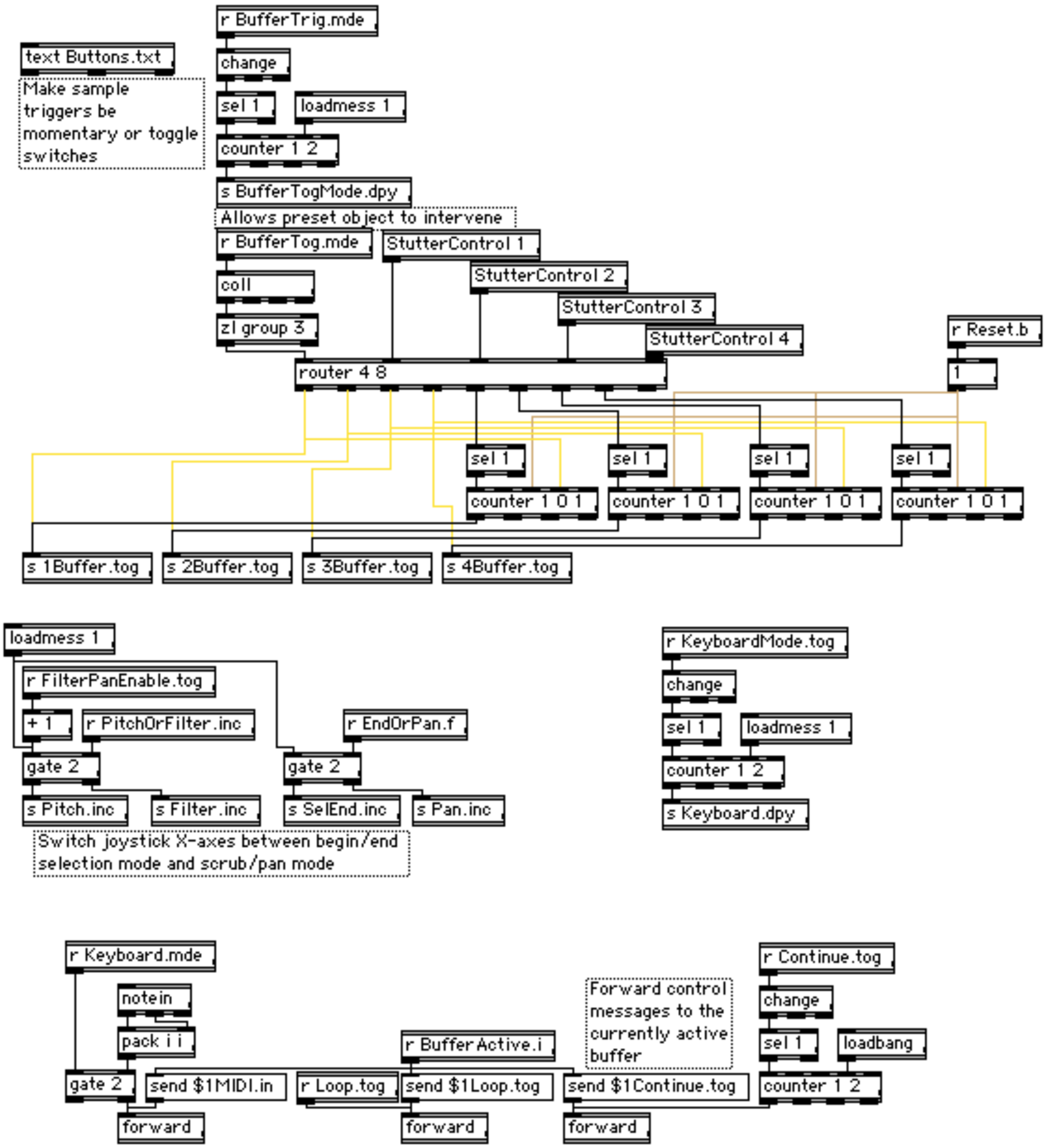

FIGURE 4.6. The Button & stick manager subpatch interprets bank trigger and stick controls, sets the keyboard mode, and forwards other button controls to the currently active bank.

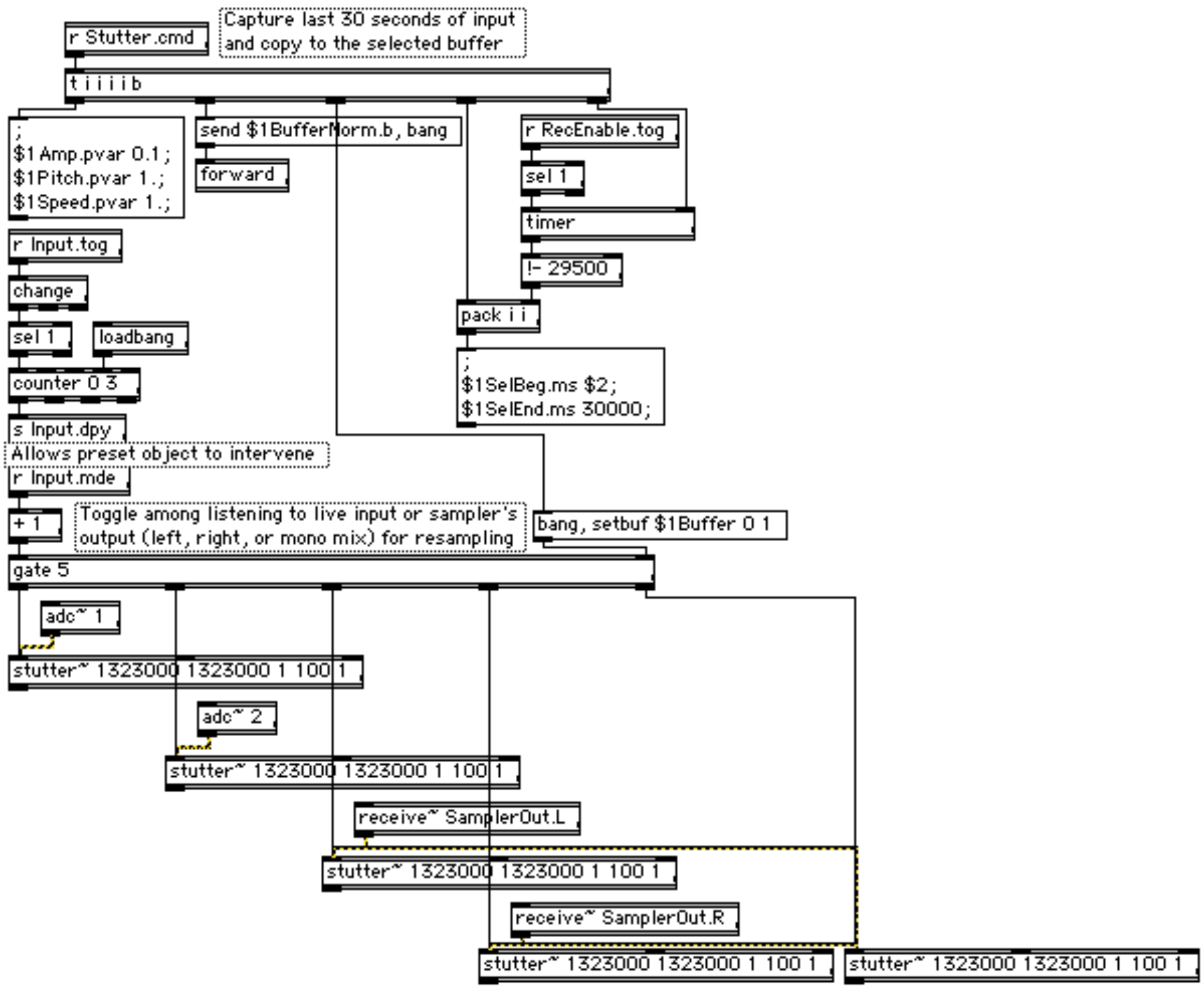

FIGURE 4.7. The Recording subpatch keeps the last thirty seconds on the left and right inputs and outputs, and loads the selected audio to the selected bank for playback.

Gamepad Sampler works better in larger ensembles than *Elektrodynamik* and *Motet,* because it is able to select a single input at a time, and it can be as responsive or automated as needed to fit in or fill space. It is currently configured for two input channels, but this feature can be easily expanded to accommodate more. In larger ensembles, it may be preferable to load prerecorded samples (as in *RUhere*) in order to give the instrument a distinct voice within the ensemble.

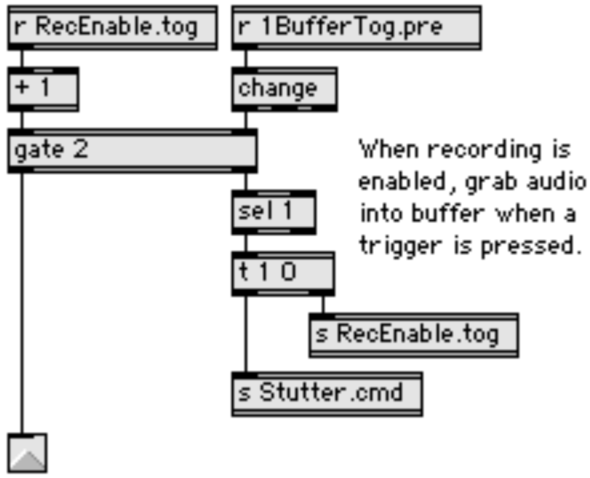

FIGURE 4.8. The Stutter control abstraction coordinates recording audio from a temporary buffer into the selected bank.

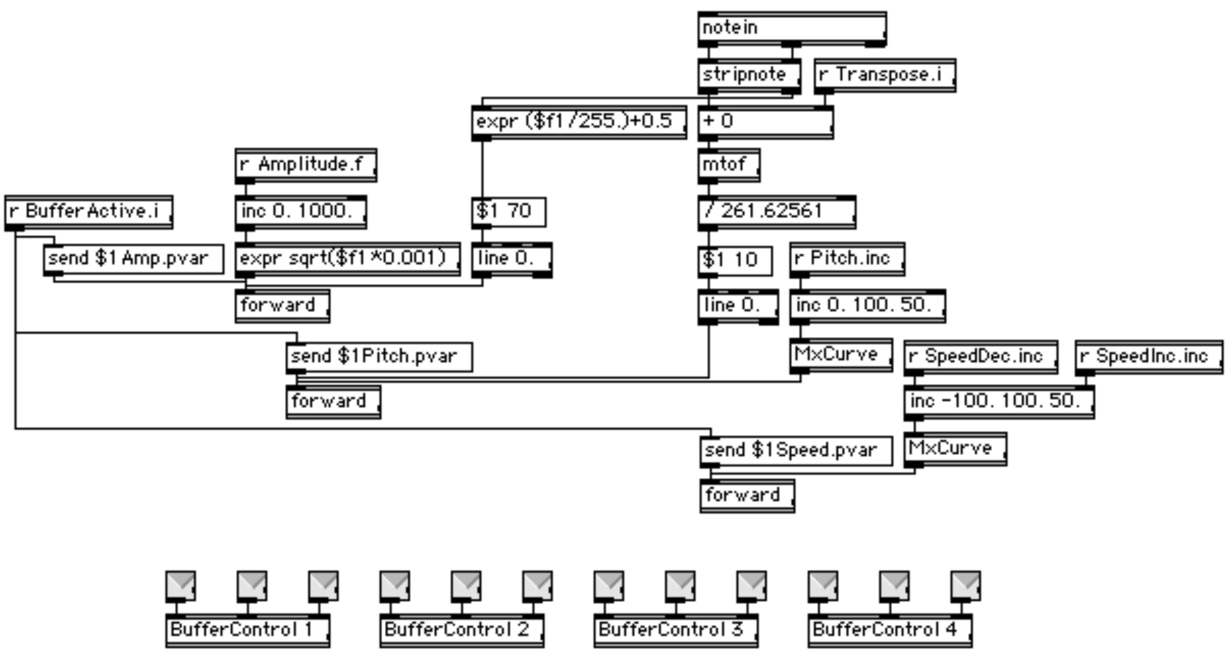

FIGURE 4.9. The Buffer controllers subpatch forwards loudness, transposition, and playback speed controls to the currently active bank.

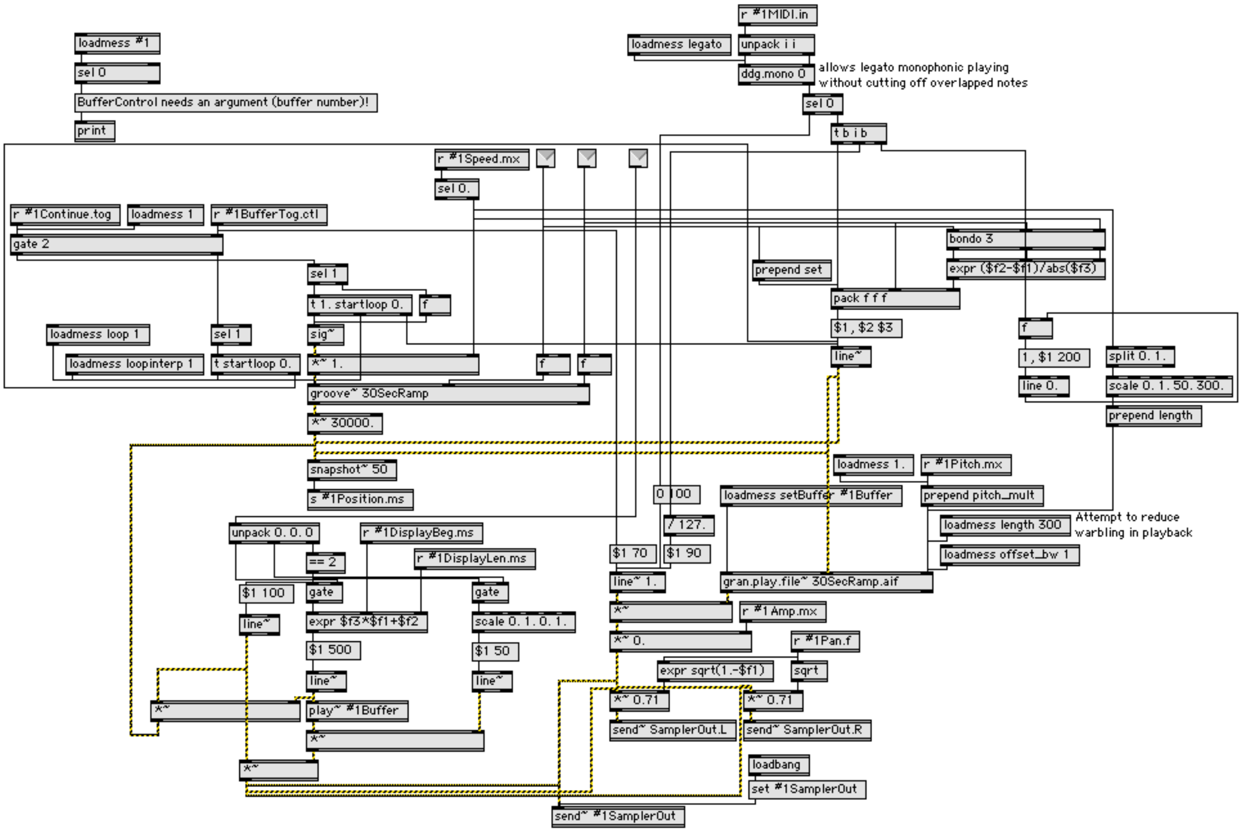

Figure 4.10. The Buffer Control abstraction reads sound recorded in the buffer for playback, as controlled by the user.

The following are performance techniques I have found to be effective:

- Listen for and try to capture a variety of gestures, textures, or musical functions.
- Try to capture a variety of timbres, even noises usually overlooked or unwanted.
- Try to capture neutral material like single sustained pitches, a single repeated pitch, or long rising or falling scale. They are easy to convincingly transform into a variety of material, giving the user more control of pitch content and contour, for example.
- Perform gestures or rhythmic patterns in Gamepad Sampler and sample the output.
- Capture multiple repetitions or variations of a repetitive pattern to allow it to be changed more during playback by altering the loop points.
- Instead of looping a distinct passage exactly, alter the loop points on each repetition to achieve a fragmented exploration of the material within.

- Vary transposition cyclically in a repetitive ostinato, so that only one or a few notes end up changing each time, and the rest remain at the same transposition level.
- Play through attacks and decays with very slow speeds, meandering toward the attack point for a climax.
- Drag a single loop end point around, quickly shortening, lengthening, or flipping the selection.
- Play a short loop with a high speed including mostly silence and just the attack or release of a note to establish a driving pulse. The random aspects of the granular sound player will create variations in each loop, sometimes skipping over the audible part of the loop. Use the loop point dragging technique described above to change the tempo of the pulsed sound or trap it in a short stuttered loop.
- Scrub slowly over very short distances for bass sounds.
- Play detached scrubbed sounds by clicking and immediately releasing in the waveform display (interpolation will play short note). Clicking at greater distances from the last cursor position results in higher speed and pitch transposition.

Future Work

The next version of this instrument will include more automation features, making it more similar to *Motet.* After a sample bank plays a certain number of iterations without being altered by the performer, the program will start varying the loop points, transposition, or volume on its own. These are designed to be fairly transparent changes: leaving dramatic or significant events to the performer, but changing the material just enough to avoid monotony while preserving the character of what it is playing. With this feature, Gamepad Sampler could be used as an installation, with occasional user interaction.

It would also be very useful to allow the sample banks to be synchronized to form multilayer patterns without collapsing it to a single track. This would facilitate more exploration of metric time and beat oriented musical styles. Automation of parameters synchronized with the loop playback would also be useful, allowing more complex control gestures, possibly including scrubbing gestures, to be captured and edited during playback. Since the graphical waveform display is already in place, it would be possible to copy and paste musical passages, within or even across different banks, offering another method of rewriting past events. With this feature, an undo/redo command would also be useful, allowing the user to switch between two versions of a bank during playback, in addition to undoing errors.

A new version of this instrument already under development replaces the gamepad with Wii Remote controllers originally designed for use with Nintendo® Wii™ video game system.[1] The Wii Remote is a small handheld device with a trigger, directional pad, and six other button switches in addition to three orthogonally-oriented accelerometers, force feedback, and an infrared camera used to determine the position and orientation of the device in relation to a "sensor bar"[2] containing two banks of infrared lights placed near the screen.

In addition to increasing the physical involvement of the performer, this development will allow the instrument to become a forum for live collaboration by allowing multiple performers to operate the same sampler instrument, possibly including the other instrumental performers.

[1] Nintendo® is a registered trademark of Nintendo of America Inc.

[2] The Wii™ sensor bar does not contain any sensors but rather provides reference points for the infrared cameras in the Wii Remotes.

CHAPTER 5

DISCUSSION

Feedback

Elektrodynamik, Motet, and Gamepad Sampler each produce profoundly different results when taking in and processing their own output, whether this feedback occurs within the software or between speakers and microphones. Especially in the cases of *Elektrodynamik* and *Motet*, in which constant streams of feedback can be established, software can quickly take on its own voice resulting from the chaotic aspects of the feedback path and computation.

Basically, any time-domain effect is based on a delay process, and many also incorporate feedback. A simple delay effect reproduces its input signal after a designated delay period. Delay effects often use feedback: the output of the delay process is also routed back to the input of the delay along with the incoming live signal. This feedback signal is usually attenuated so that progressive copies of a sound event will grow softer and approach silence with each repetition. A delay with feedback is a simple way to turn a single sound event into many. A short delay (e.g., twenty to eighty milliseconds) with twenty to fifty percent feedback coarsely simulates the effect of natural reverberation.[1] Chorus and flange effects common to popular music consist of similar short delays that are constantly changing, either randomly (as in a chorus effect) or in a simple periodic pattern (as in a flange effect, usually with a sinusoid pattern).

Digital filters use delays at the level of individual samples to accentuate or attenuate frequency bands in a signal.[2] A simple example is a low-pass filter that yields the average of the current sample and the previous sample. Each sample is delayed and added to the following sample, and the sum is halved, yielding the output. While the result of this filter

[1] More natural sounding reverberation is achieved with several delay lines with different delay periods, avoiding integer ratios among them (Roads, 1996).

[2] Here, *sample* refers to a single amplitude measurement of an audio signal, for example in compact disc quality (Red Book) audio, representing approximately 0.023 milliseconds.

may be difficult to imagine on a sample-by-sample basis, it is helpful to remember that a moving average smoothes out abrupt changes in a signal. Such abrupt changes are caused by high-frequency components, and attenuating high-frequency components is the purpose of a low-pass filter. The filter described here is a finite impulse response (FIR) equation that does not use feedback (Ballou, 2005).

Filter equations that use feedback are called infinite impulse response (IIR) filters and are useful for creating resonant filters. These filters resonate at given frequencies when the input signal contains components at those frequencies. Beyond accentuating certain frequency bands, a resonant filter can become a sound source of its own, responding to an input signal. Similar to a simple resonant filter with one resonant mode, a delay with a period of fifty milliseconds or less and feedback approaching one hundred percent will be heard as resonating at several frequencies in a harmonic series. This effect, called a comb filter because of its many evenly spaced resonant peaks, is the basis of technique described by Karplus and Strong (1983), which yields very efficient physical models of stringed instruments or drums.

This demonstrates the wide variety of effects that can be produced with feedback. In *electronic feedback*, as used in these processes, the output of the system is connected to its input while it is still an electrical signal (Meyers, 2002). These techniques were explored in the mid-twentieth century by Les Paul (Bode, 1984) and Pierre Henry (Chadabe, 1997, 31), and are now common techniques used in music production.

Electroacoustic feedback describes the more commonly known feedback situation: the output of a speaker is picked up by a microphone that is connected to that speaker through an amplifier. In this situation, if the amplification and the resonance of the room is sufficient, the signal will be dominated by a sinusoid resonance at a frequency related to the resonant modes of the room. Electroacoustic feedback can yield more complex results than electronic feedback, because the acoustical system of the room is more complex than the simple formulas described above. This simple sinusoidal electroacoustic feedback has been explored by David Tudor (Pritchett, 2004) and Alvin Lucier (2002), and more recently by Nicolas Collins (2002), a student of Lucier, and Roger Dannenberg (2006). In simple electroacoustic feedback, the only delay is the inherent latency of the whole system, which is primarily a function of the distance travelled by the sound between the speaker and the microphone and the speed of the computer audio interface.

More complex sounds can result from a feedback system whose properties are in flux. Collins used phase shifters to periodically squelch certain resonant modes and allow others to sound in *Pea Soup* (1974–76/2004; 2002). Placed in audible range of a speaker, a microphone used as an input to *Elektrodynamik* or *Motet* can yield very complex results. The shifting delay periods in both works act as macroscopic phase shifters, canceling different resonant modes and supporting others as the delay periods change. *Elektrodynamik* employs electronic feedback with digital delay lines to achieve time-domain effects related to those described above, but in most scenes, the amounts of electronic feedback to each delay are also in flux, constantly changing the system's response to the acoustic input.

In performances of *Tappatappatappa* (Morris, 2005), a 2-inch speaker cone is used as the input to the *Elektrodynamik* program, acting as a dynamic microphone. The speaker used was extracted from a headphone set. Such a low-output speaker cone has a limited range of motion and will resist introducing direct current (DC) offsets into the signal, which may damage audio equipment. The cone is excited by stroking or tapping with fingers, a pipe brush, and a frayed wire, but it also picks up sounds in the room, allowing the software to sample its own output and also establish feedback resonances. These resonances and the overall sound quality of the recaptured input can be controlled by moving the handheld speaker, pointing it in another direction, covering it, or damping it with a finger, which mutes it, as one would silence a cymbal. The implements used to excite the cone do not need to be rich in timbral possibilities. They provide enough timbral variety to influence the output of the system in a general way and allow precise temporal control to create rhythmic gestures and textures as needed for the system to build upon or to allow soloistic intervention in the performance.

The resulting resonances are particular to the orientation of the surfaces, speakers, performer, and audience within the space. This makes performance somewhat of an exploration, unique to each situation. After performing a while in one environment, the memorized association of each position and angle with a certain resonant response can become like a tactile sensation for the performer. The multiple delays with shifting properties and the beat and auto-mixing features constantly change the system's response, adding to the improvisational aspect of the performance.

Different sounds are produced from *Motet* with feedback because of its use of pitch shifters. With a short delay, or the natural latency of the system without any explicit delay, a sound will be repeated and transposed in a sequence. This sequence will continue upward or downward, depending on the degree and direction of the pitch shifter, and will continue until the output is transposed out of the sensitive frequency range of the equipment. This of course becomes more complex when multiple Voice modules are used in the program, each with a different pitch shifter and transposition setting. A single note would be reproduced as a four-note chord, which would be reiterated in sixteen voices, and so on, continuing exponentially. Unless all pitch shifters are set to transpose in the same direction, it is likely that while parts of the output would soon be transposed out of range to be reproduced, some part of the output would be audible to the system for a much longer time. Several generations of transposition sequences may continue to form, and they and the artifacts produced by the system will continue to be processed long after the initial input sound and the first sequence have faded away. Even more complex output results from the automatic shifting of delay times. Sequences will be fragmented and shifted temporally in relation to each other, yielding a texture and pattern that evolve over time.

Complex feedback results can also be achieved with longer delays or manually reiterative processing, as in Lucier's classic composition, *I am Sitting in a Room* (1969), and many of his other works. I have used a similar manual feedback approach by repeatedly applying a noise reduction process on computer-generated white noise (and consequently the most recent output of that process) in order to "creatively discover"[3] electronic voice phenomena (Enns, 2005; Morris, 2007 [forthcoming]).

Similarly, the Gamepad Sampler allows the performer to recapture the total output of the software into a sample buffer, capturing all buffers and scrubbing gestures playing at the time as a single sound, which can then be further manipulated in performance. In addition to allowing recombinant sampling of materials, this technique also captures the artifacts of the pitch and speed processing, which can let the performer explore the sampler's own voice in the same way as in the other tools with a continuously streaming feedback loop.

When used in a feedback system, *Elektrodynamik*, *Motet*, and Gamepad Sampler can be treated like distinct instruments: *computation instruments*. This echoes a sentiment in Stephen Wolfram's (2002) study of cellular automata, highlighting the beauty of computation not for the purpose of an end result, but for the sake of computation—that is, for the artifacts of the process itself. Before Wolfram, many have mused at the beauty resulting from natural processes like the formation of crystals, erosion of rock, or the growth of plants, and Benoît Mandelbrot (1982) made a connection between these natural processes and simple iterative mathematical processes. Cellular automata as explored by Wolfram exemplify such processes that can yield complex and beautiful results from simple processes. Each automaton is given the same set of rules that determine its state (usually black or white) based on the states of the automata around it. While these can be used to find a particular end result (a solution), the structure of that process can also be appreciated. For Wolfram, this is the value of computation for the sake of computation; it is an approach more common to natural structures. For example, it is more common for one to appreciate the overall form of a coastline (or its evolution over time) than to use data to determine the position of one point on a coastline. In nature, process can be appreciated as artistic. In most art, the result of a process is appreciated. In computing, the result is a solution to a problem, and

[3] I will not argue here whether electronic voice phenomena exist as such, in that work or in any other reported instance. The phrase "creatively discover" is adopted from Dodd (2004, 347), who argues whether a musical work is created (thus coming into existence at a distinct point in time) or is instead discovered. It is meant to capture both aspects of the process.

the process is usually neglected, except to see if the result could be reached faster. However, this need not be so.

Wolfram attempts to explain the artistic value in some natural or computational structures with his principle of computational equivalence:

> . . . almost all processes that are not obviously simple can be viewed as computations of equivalent sophistication. (Wolfram, 2002, 716–17)

He continues to suggest that processes humans perceive as complex or beautiful, ones that in either way make it "easy to forget that the rules are really in place," (Flake, 1998, 11), are perceived that way because the processes are as sophisticated as the humans perceiving them. Perhaps this explains why it is difficult to prescribe processes based on simple steps (like computer programs) that are capable of beautiful results. It can at least be gained from Wolfram's methodical study of cellular automata that existing fractal or chaotic systems may be instrumental in achieving beauty in computerized processes.

In *Elektrodynamik* and *Motet* with feedback, the chaotic system of room acoustics is explored during performance. While both *Elektrodynamik* and *Motet* involve streaming speaker-to-microphone feedback, the results are distinct, because the programs are distinct. Iterative feedback with Gamepad Sampler employs the randomness in granular sampling techniques and the chance of improvised performance. Feedback illuminates the natural resonances of an acoustical system as in Lucier's works, but also the natural resonances of the software system. The feedback techniques explored with *Elektrodynamik, Motet*, and Gamepad Sampler allow the unique voices of the programs to emerge, and the properties of original sonic stimuli are lost as they do.

Aesthetics of Live Sampling

While sampling or musique concrète techniques have been used in popular and art music for several decades, the pervasive incorporation of samples recorded and replayed within a single improvised performance has only become feasible more recently. The aesthetic implications of live sampling during performance deserve discussion. Peggy Phelan asserts, “Performance cannot be saved, recorded, documented, or otherwise participate in the circulation of representations *of* representations: once it does so, it becomes something other than performance” (Phelan, 1993, 146). However, the sample that is a copy of an event earlier in the same performance must lie somewhere between the “Amen break” (The Winstons, 1969; Harrison, 2004) and the return of a fugue subject: it is both the artificial reproduction of another musical event and the recurrence, prolongation, or imitation of musical material within a work. As the latter, the technique of live sampling facilitates structural phenomena, not just extramusical or intermusical (intertextual) signifiers.

As with performance art, for the audience of improvised musical performance “there is an element of consumption: there are no left-overs, the gazing spectator must try to take everything in” (Phelan, 1993, 148). It is often more fruitful for a listener to focus on the moment in an improvised performance than on cumulative or hierarchical development of recurring ideas, because Phelan’s statement is true of improvising performers as well as their audience. One must use musical ideas while they are present in the mind or risk losing them forever if there is no score, head arrangement, or well-practiced intent to reuse certain material. The live sampling techniques presented above allow for formal structures that are otherwise difficult to establish in improvised performance. While verbal directions are best at specifying material (the kernel of content) to be used in imitation, live sampling tools allow form to be created in improvisation independent of content. Any musical moment played so far could be selected and developed without dictating its use beforehand.

Further, the original signified/sampled material can also be reintroduced and developed by other performers in the ensemble, and the sampled material can be reconstructed during performance, eroding the authenticity of the first instance of the musical idea. This can be seen as either shifting authenticity toward an abstract, Platonic, and perhaps never-heard version of the musical idea or, alternatively, undermining any sense of authenticity in the performance by flooding it with intramusical references.

Phelan explains the first statement quoted in this section, “To the degree that performance attempts to enter the economy of reproduction it betrays and lessens the promise of its own ontology. Performance's being . . . becomes itself through disappearance." (Phelan, 1993, 146) Something is lost when a performance is reproduced, and the loss weakens ontology, but is it the state of the performance or that of the recording that is affected? In Walter Benjamin's terms (1936/1969), the *aura* of a work is lost in reproduction, so that the copies are degraded, and the original is unaffected by the existence of the copies. However, Phelan refers to the ontological promise of the original performance. To "enter the economy of reproduction" is to do something *reproducible*, whether or not it is reproduced. The more that would be lost in reproduction, the more substantial it is, or the greater its aura is.

When a sound is recorded, the sample does not have the aura of the original sound, but in the context of live sampling within the course of a musical performance, the sample can gain its own aura that reframes the aura of the original as an acoustic version of a musical idea that is later presented in an electronically-mediated form. Phelan writes, "Performance occurs over a time which will not be repeated. It can be performed again, but this repetition itself marks it as 'different.' The document of a performance then is only to spur a memory, an encouragement of memory to become present" (Phelan, 1993, 147). Not only does the aura of the original moment slip away, but the memory of that aura is replaced with a new one situated in a net of imitative references. Multiple copies of the musical idea, acoustic and electroacoustic, reframe each other as particular references to an abstract musical form evoked by—but outside—the performed music. The "original event" becomes merely the "first reference" to a musical idea within the performance. In these works, the sonic presence of the performer is prolonged and folded upon itself in time, taking along the gaze of the audience, to be continually revised.

Stage Presence

The fairly new possibility of using a laptop computer as a performer's primary or sole instrument has raised concern over stage presence in some audiences. When the performer on stage looks the same while doing office work, playing a video game, or giving a masterful musical performance, hunched over behind a computer display, an audience may become as detached from the musical experience as the performer appears to be.

Some laptop-oriented performance venues have expressed interest in developing a stage presence beyond that of an ordinary computer user. For example, the Laptop Deathmatch series in Dallas, Texas demands "The ability to command the attention of a theater audience by the impressiveness of one's manner or appearance."[4] Most photographs from their events, however, do not demonstrate this. Some argue that younger audiences are not discomforted by the apparent absence of effort by the performer (d'Escriván, 2006). It may also be argued that computer interfaces may become more physical, making this issue irrelevant in the future, or that a laptop performance is already similar in appearance to performances on traditional instruments such as the organ (in which the velocity of movement also has no effect on the sound, and the performer may be physically distant from the sound source). I believe that the issue of stage presence in electronic music will persist in some form as a significant element of performance. It may raise questions and concepts that inform stage presence in more traditional performances as well.

In a strongly *mediatized* culture (in which communications technology frequently intervenes in the presentation of live events), live performance may not be appreciated for its strengths, but seen as a flawed substitute for recorded media. As Walter Benjamin predicted (1936/1969), the aura of liveness and the *now* may become foreign, worthless, or imperceptible to an audience. Phelan says, "...the now is supplemented and buttressed by the documenting camera, the video archive" because it is rare in our current culture to experience live performance as such (Phelan, 1993, 147). With the exception of festivals held by universities, many electronic music venues are not friendly to live performance, because they are dominated by social activity, such as dancing.

However, even when there is an audience watching attentively, many performances do not offer them much visually. Outside universities, many developments in electronic music involve using guitar effects pedals. The pedal format is convenient for a guitarist to switch

[4] `http://www.laptopdeathmatch.com` (accessed June 10, 2007)

effects on or off while standing and keeping both hands busy. However, it is becoming more common to have electronics performers seated on the floor hovering over a collection of pedals, ignoring and hiding from the audience. This can be related to Phelan's "economy of reproduction." A lack of physical movement is associated more with the composition than with the performance, and adds no aura to the performance. A scripted process is usually a product of the composer, and the contribution of the performer is more directly triggering events or changes in ongoing events. A script is very reproducible—it was created for the purpose of repeatable automation.

Over the time I have performed with electronics, I have experimented with mannered actions, following after dramatic pianists that lunge their torso into chords and raise their hands high above the keyboard before or after a note. I understand that some of these motions have real effects on the sound, but I maintain that they are often exaggerations, and that the sonic effect is often minimal. However, the overall effect of the performance may indeed be strengthened by such visual ornaments. In performance, I began to allow my face to reflect the intensity or fulfillment with which I meant to imbue my performance. I used higher than necessary hand strokes to push buttons or sliders, which let me feel more in control of timing, like a conductor's preparatory gesture. When expressively changing playback speed with Gamepad Sampler in order to approach a climactic note attack, I began to let the gamepad tilt back and forth accordingly. I have not asked audience members their opinion of this, but I have received many positive comments. By acting as if I were playing a "real" instrument, it seems to have helped audiences process the fact that I actually was.

The computer performer may detract from the musical experience by *not* displaying mannered or obvious causal actions, especially when the computer performer is joined by acoustic instrumentalists whose instruments demand a certain level of visible causality. While it is unreasonable to insist on a one-to-one correspondence between performer actions and musical events, there can be an intriguing counterpoint between causal performance and its effects.[5]

---

[5] By *causal performance* I mean any action of a performer during a performance that has some result in the performance, either generating a sound directly or triggering events or changes in the action of the software

It appears to be helpful for the performer to act as a surrogate or model for the audience, demonstrating the emotional intensity, physiological engagement, sincerity of expression and enjoyment of the performance that a sensitive audience member might have. It can be frustrating, confusing, or insulting to the audience if such signs are not present or congruent with the musical content of the performance.

The Nature of the Works

It is clear that *Elektrodynamik* and *Motet* are computer programs, but the fact that neither includes explicit instructions for realizing particular sounds makes them different from other compositions or instruments. They are at least beyond the class of static signal processors (e.g., a reverberation or guitar effects unit). I have always considered them to be compositions, primarily because of their origin, and secondarily because of their use. *Elektrodynamik* was a response to a request by cellist Ulrich Maiß for a work involving improvisation. A performer asked a composer for something to play: in these terms, it seems plain that this thing would be called a composition. Both works have been used by other performers, sometimes with someone to operate the program and sometimes with the program running unattended. Such use befits a composition.

Composer Horacio Vaggione avoids calling such works compositions in themselves, but recognizes that "automated composition" is a part of such a system (Vaggione, 2001). Guy Garnett (2001) offers an aesthetic discussion of interactive computer music, but explicitly avoids a discussion of ontology altogether. George Lewis refers to his *Voyager* as an "interactive musical environment" that generates and performs an "automatic composition (or, of you will, improvisation)" (Lewis, 2000, 33), but after this formal introduction, he refers to it as a composition. An "automated composition" system is more of a composer than a composition. If *Voyager* is to be called a composition and not a "creator of compositions," then I would not refer to *Elektrodynamik* and *Motet* as automated composition *systems.*

program. These actions can be ornamented or mannered. Other actions may be purely ornamental, having no effect outside themselves.

While the possibilities of their output are practically infinite, they are much less versatile or intelligent than *Voyager*, and this is by design (see page 4).

Dominic Lopes likens works such as these to computer games, because they have “an indefinitely large set of possible interaction-instances with different sound structures” (2001, 79). However, he portrays such “strongly interactive” works as dependent upon performers (interactors) for their *existence* (not just their presentation, instantiation, or interpretation). What Lopes calls a “strongly interactive work,” Stephen Davies would call “ontologically thin,” because some composerly agency is afforded to (or required of) the performer. In Davies’s terms, a performance of such a work would be like a performer’s *interpretation* in origin and like a composer’s *version* (or revision or recomposition) in extent. The performance, however, is neither: Davies calls it an *interpretation version* (Davies, 2007 [forthcoming]). The “composerly agency” afforded to the performer is nothing more than what is otherwise called improvisation. This does not mean that the works are incomplete, but that they are open; improvisation is something the compositions require of the performer. The works, the products of my creation, are unchanging unless I revise the software program, and there is no necessary progression from one performance to another. Lewis has no problem with saying that a software program can have its own “sound,” something that distinguishes it from other programs and unifies its own products, even if the structure of those products differ widely.

According to Guy Rohrbaugh (2003), these forms of art have three properties:

Modal flexibility: They could be different but still be the same work.

Temporal flexibility: A work itself is "susceptible to change over time."

Temporality: A work is created and capable of going out of existence at any time.

He continues to suggest that only historical manifestations of such works are real. This would make discussing interactive works similar to discussing jazz performances. The performance of "Body and Soul" by Coleman Hawkins and His Orchestra recorded on October 11, 1939 would be a distinct entity from the version recorded on the soundtrack for the film *Body and Soul* (Polonsky et al., 1947), and the song entitled "Body and Soul" (Heyman et al., 1930) would be merely an abstract description of some harmonic properties common to both historical events. This example is perhaps not so hard to deal with, but it is different to apply it to interactive works like *Elektrodynamik* and *Motet*; each performance is at least as distinct as a jazz performance of a standard, but the compositions are hardly on par with a "lead sheet."

Rohrbaugh's view is not a predominant one. Vaggione adapts Finsler to suggest that musical ideas are real even if unperformed. Julian Dodd responds to Rohrbaugh by reasserting that a musical composition is a *type*—an abstract Platonic form—of which its performances are *tokens* (or Platonic *particulars*). He does so by exploring where Rohrbaugh (along with Jerrold Levinson) draws the line between acceptable variations on a work and substantial alterations that necessitate calling it a new work. Conventional music notation is limited in expressing nuance, and most verbal directions like tempo markings are approximate at best. Dodd leaves no reason to draw a line that makes these variations negligible. Instead, he explains that a musical composition is a *type*, but a *vague type*: a "fuzzy but unchanging" form that includes its own variability explicitly or implicitly (Dodd, 2004, 351). Rohrbaugh claims that a musical work is temporal, meaning that it is capable of being created and also destroyed. Dodd argues instead that a musical composition is an atemporal abstract Platonic form: it is more related to a mathematical proof or a patented technique than a created product. This means that the composition process is more like discovery than creation.

From a composer's perspective, this seems natural. Much of the compositional process is often seen as problem solving. I prefer to think in terms of searching elegant paths, or "hacking" rule systems in order to side-step the problem by making it irrelevant (see page 4). Alvin Lucier has commented, "Some of art is that you make connections between things that no one else would ever make" (Lucier (1995, 70), quoted in Aufermann (2002)). I agree that composition is discovery. There is very little in the program code or equipment that is truly unique, but my connections among these things have yielded distinct musical works.

I do not believe it makes sense to say that a musical work could cease to exist. Would it happen if all extant scores and recordings were destroyed? After everyone who ever performed or heard the work died? If all references to the work were erased? This is difficult to accept as a valid or worthwhile inquiry. A musical composition is an abstract atemporal Platonic form, and its performances are historically manifest particulars: *instances* of the abstract form.

Forms and Derivatives of the Works

A number of works derived from *Motet, Elektrodynamik,* and Gamepad Sampler have already been mentioned. For clarity, here is a discussion of all the related works.

Elektrodynamik is a composition for improvising performer with interactive electronics that makes no sound of its own but acts to impose structure on material recorded from the soloist. *Tappatappatappa* is a work that fully uses *Elektrodynamik* with a cone microphone as the solo instrument and sound input, which enables feedback loops that illuminate the character of *Elektrodynamik* as a *computation instrument* (as described on page 61). The composition *Tappatappatappa* is ontologically thin, because it relies heavily on the performer to control its events and progression. It relies heavily on the structure of *Elektrodynamik,* which exists as a work outside *Tappatappatappa*, for extending and developing the sound events of the performer and for form on a larger scale. In *Tappatappatappa*, *Elektrodynamik* is used as an instrument.

This may seem to be an unnecessarily complicated distinction, but it fits my intuition about the works from the perspective of a performer. Performing *Tappatappatappa* is more like performing other ontologically thin works such as *December 1952* (Brown, 1952–4). Performing *Elektrodynamik,* on the other hand, is different. It is not at all like reading a traditionally notated score, but is more like improvising with a partner that has exceptional skill in reusing material and laying out larger scale forms. Neither work tells the performer what to play; the burden of realizing of *Tappatappatappa* lies heavily on *Elektrodynamik* and the performer.

Portrait for toy piano and electronics is something different. The piano part is fully notated, only indeterminate regarding the duration of some rests. However, the program that accompanies the piano in performance is an adaptation of the FX, Player, and Transit modules from *Elektrodynamik.* The presence of the score (or the absence of improvisation in it) significantly reduces the burden of creation on the performer, so it is ontologically *thick* in Davies's terms (I would prefer *substantial*).

The term *quote* is technically applicable here, since it is the result of "copying and pasting" code from *Elektrodynamik* into *Portrait*, but the resulting effect is more like that of adopting a compositional technique from another work. This is because while portions of the code were reproduced verbatim, there is no verbatim quote of sound events from one work in the other, which would be a *musical* quote. Instead, in the musical performance resulting from the copied code, it is *behaviors* that are copied. This is perhaps like including a fragment cut out of *For 1, 2, or 3 People* (Wolff, 1964) in another score.

Motet could be considered ontologically thinner than *Elektrodynamik,* because, although it creates local temporal structure, it relies on the performer to direct it in larger forms. It can also be used in performance as a computation instrument (with feedback). Gamepad Sampler is difficult for me to call a composition, because every event is at least initiated by the performer. There are significantly more one-to-one correspondences between performer actions and sound events. *RUhere*, however, is an improvised performance that uses

Gamepad Sampler with a specific set of prerecorded samples. *RUhere* is a composition, although a thin one.

This discussion has illuminated points that suggest a definition of the word *composition* that can address ontologically provocative works such as these. Composition, from the Latin *componere*, meaning "put together," is the portion of the performance realized outside the performance. The *performance* is the live event including "compositional" acts executed by the performer during the performance, but excluding components that have an existence outside the performance (e.g., the composition, instrument, and performer).

Concluding Remarks

In the preceding sections, I have presented *Elektrodynamik* and *Motet* both as compositions and as instruments. Could an instrument be considered a composition? Do they lie on a continuum, as Stockhausen (1959) tried to establish with time unifying timbre, tone, rhythm, and form? Are they distinct classes whose definitions can overlap, along with concepts like *performance*, *version*, and *variation*? New definitions are needed.

Golo Föllmer, writing about internet-based music, calls for an adjustment of familiar terms:

> So we may have to think about a more open way to define the notions of the composer, the performer, the listener. The composer would (still) be a person who sets a scope and lays connections of ideas, concepts and atmospheres in music. The performer would be the person to mediate a primarily auditory event, albeit following different rules than in the past. Finally, the listener would be the one to put it to a final order, to receive it at a final destination and interpret his/her personal version of it. Whatever will happen, the main effect is that the range of roles for each grows. (Föllmer, 2001)

These terms began shifting with the development of open form compositions like those of Earle Brown and Stockhausen in the 1950s, and now they must be adapted to address ontologically substantial interactive music. Shifting the boundary between composer and performer does not necessarily lessen the ontology of the work, but allows it to shift away from a single well-defined performable event structure toward an infinite set of performable events derivable as behaviors in response to performed events, each member of the set possessing the same underlying structural characteristics.

Reiteration of content through live sampling does not lessen the ontology of the overall performance either. It may lessen the ontology of the sampled material as it enters "the economy of reproduction," but the moment of its re-performance, and especially the impact of its re-performance on the continuing live performance of other musicians creates new events that deepen the ontology of the performance. This even satisfies Rohrbaugh's (2003) narrow requirement of "historical manifestation." The performance develops and explores a tension between its own disappearance and prolongation.

While *Elektrodynamik* and *Motet* exist only as software programs and neither contains explicit instructions for realizing particular sounds, they are still musical compositions. Like several other open works, every performance exhibits certain qualities that are characteristic of the specific work and are resultant from the structure of its program. In these works, the act of composition includes designing algorithms to bring about certain behavioral and musical tendencies, and the object of the composition, the "work," exists in every performance. This can be seen as an additional dimension of the musical work; whereas traditionally notated two-dimensional sheet music compositions can be performed with an infinite set of variations, compositions like these involve performance as another dimension of itself. While a traditional composition is *interpreted* in performance, these compositions are *instantiated*. Inspired by mid-century musical developments, Umberto Eco's (1989) wrote about "open" works of art:

> In fact, the form of the work of art gains its aesthetic validity precisely in proportion to the number of different perspectives from which it can be viewed and understood. These give it a wealth of different resonances and echoes without impairing its original essence.

The performance is a part of the work and not merely a reference to it or trope on it. The performances of works like these are particulars that each illuminate part of the abstract form that is the real composition. An indeterminate work like *December 1952* (Brown, 1952–4) is ontologically thin, because almost every aspect of its performance is coordinated by the performer. *Elektrodynamik* and *Motet* are not thin, because they contain within them sets of distinct contributions to each possible performance—that is, they are interactive.

Gary Flake relates the situation of the computer music composer to other artists:

> The fact that good programs are logical by necessity does not diminish the beauty at all.[6] In fact, the acts of blending colors, composing a fugue, and chiseling stone are all subject to their own logical rules, but since the result of these actions seems far removed from logic, it is easy to forget that the rules are really in place. Nevertheless, I would like you to consider the computer as a medium of expression just as you would canvas or clay. (Flake, 1998, 11)

Flake says it should not be ruled out that the product of a computer can be beautiful, but it is not license to assume that any computerized result is good, either. It takes at least as much effort to get good results from a computer program as it does to compose a good fugue or create a good sculpture. Perhaps extra effort is required in computer-based art to overcome the greater distance between the creation of any part of the program and the perception of its results. While in traditional composition, any segment may be played on its own and evaluated by the composer from the perspective of an audience, one part of a computer program may require the rest of the program to be in place along with input from a performer before it can be fully evaluated. In interactive musical works like the ones presented here, William Seaman describes the composer's new challenge: "The artist need no longer seek to define a singular artefact, but instead need develop systems that enable a series of sonic artefacts to become operational and polycombinational, thus engendering an emergent relational sonic artefact during exploration" (Seaman, 1999, 234). As the human condition changes, the frontiers of art and the role of the composer are changing.

6 Here, *logic* refers to the rigid structure of a computer program, not philosophical logic.

APPENDIX A

RECORDINGS

Track 1. *Zur Elektrodynamik bewegter Musiker* Unattended with Soloist

Ulrich Maiß, violoncello, premiere at BKA am Mehringdamm in Berlin, Germany, September 27, 2005.

Track 2. *Elektrodynamik* Attended with Soloist

Studio recording session by Eric km Clark (sic), violin, at the Atlantic Center for the Arts (ACA) in New Smyrna Beach, Florida, on July 3, 2006.

Track 3. *Tappatappatappa* (*Elektrodynamik* Alone, Using Feedback)

Jeff Morris in the opening gala concert of the Bellingham Electronic Arts Festival, at the American Museum of Radio and Electricity in Bellingham, Washington, November 20, 2006. A 2-inch speaker cone was used as the input, acting as a dynamic microphone, excited by fingers, a pipe brush, a frayed wire, and sonic feedback.

Track 4. *Motet* Attended with Soloist

Studio recording by Eric km Clark, violin, at the ACA, on July 3, 2006.

Track 5. *Motet* Alone, Using Feedback

Studio recording by Jeff Morris at Texas A&M University on April 27, 2007.

Track 6. Gamepad Sampler with Soloist

Eric km Clark, violin, in a studio recording session held during the International Computer Music Conference (ICMC), New Orleans, Louisiana, November 10, 2006.

Track 7. *RUhere* (Gamepad Sampler Alone, Using Prerecorded Samples)

Studio recording by Jeff Morris with prerecorded samples at Texas A&M University, November 3, 2006.

Track 8. Gamepad Sampler in a Trio

Jeff Morris with Mark Zanter (guitar synthesizer) & Andy McWain (piano) at the ACA, July 5, 2006.

Track 9. Gamepad Sampler with a Larger Ensemble

Conduction with Butch Morris at ACA, July 7, 2006. This was the first public use of the vocal samples used with *RUhere*.

Other performers in the ensemble:

Eric km Clark, violin
Jeff Levenberg, viola
Carla Cisno, bassoon
John Bruschini, guitar
Mark Zanter, electric guitar
Andy McWain, piano
Travis Johns, electronics (non-computerized)

APPENDIX B

PERFORMANCES AND PUBLICATIONS

The following performances and publications made significant use of the programs presented in this document.

- Premiere performances of *Zur Elektrodynamik bewegter Musiker* by Ulrich Maiß, violoncello, in the ZenMan Improvisations program, September 25, 2005 at the Brandenburger Theater, Brandenburg a. d. Havel, and September 27, 2005 at BKA am Mehringdamm in Berlin, Germany.
- *Tappatappatappa* (fixed media version) in the SPECTRUM concert series in the Merrill Ellis Intermedia Theater at the University of North Texas, October 10, 2005.
- Premiere performance of *Portrait* by Phyllis Chen, opening the eXtensible Toy Piano Festival (sic) hosted at Clark University in Worcester, Massachussetts on November 4, 2005.
- *Elektrodynamik* with Kathryn Woodard, piano and Kevin Patton, guitar at the Construction Company in New York, New York as part of the Music Under Construction concert series on December 17, 2005.
- Solo performance set using *Elektrodynamik* and *Motet* with lap steel banjo and with feedback in conjunction with improvising video artist Mark Henrickson at the Spark Festival of Electronic Music and Art in Minneapolis, Minnesota on February 22, 2006.
- *Tappatappatappa* performed by Jeff Morris at Aural Tick 2006, Texas A&M University, May 1, 2006.
- Gamepad Sampler used in chamber ensemble concert in improvisation conducted by Lawrence D. "Butch" Morris in INsideOUT exhibition series at the Atlantic Center for the Arts, New Smyrna Beach, Florida on July 7, 2006.
- *Elektrodynamik, Motet,* and *RUhere* demonstrated and performed in lecture recital for MUSE series at Texas A&M University Stark Gallery on November 1, 2006.
- *Elektrodynamik* used with vocal samples in the production of the opening and incidental music for *Ojen Kaleidoscope*, a collaborative intermedia theater presentation by the Aggie Players at Texas A&M University, November 2–5 and 9–12, 2006.
- Performance of *Motet* and Gamepad Sampler with Eric km Clark, violin at the International Computer Music Conference in New Orleans, Louisiana on November 8, 2006.
- *Tappatappatappa* in the opening gala concert of the Bellingham Electronic Arts Festival at the American Museum of Radio and electricity in Bellingham, Washington on November 30, 2007.
- *Motet, Tappatappatappa*, and Gamepad Sampler at the International Society for Improvised Music conference in Ann Arbor, Michigan with Andy McWain, electric pano, and Steve Whipple, bass on December 3, 2006.

- Performances of *Motet, Tappatappatappa*, *RUhere*, and Gamepad Sampler as featured guest on a concert at the Austin Museum of Art as part of the Audio Inversions concert series, February 12, 2007.
- *Portrait* performed at the University of Albany by Robert Gluck on March 4, 2007.
- Performance with Gamepad Sampler and guest artist David Bithell, extended trumpet, at Aural Tick 2007 on March 5, 2007.
- *Portrait* performed by Phyllis Chen at Indiana University, Bloomington on March 31, 2007.
- *Portrait* prepared (but not used) by Phyllis Chen for performance in the International Gaudeamus Interpreters Competition April, 2007.
- Gamepad Sampler used by Jeff Morris with original electronic voice phenomena in *Disturbances* (Morris, 2007 [forthcoming]).
- Gamepad Sampler used by Jeff Morris in contributions to the *vanishingPoint* blind collaboration album project by Lily Maase (forthcoming).
- *RUhere* to be included on a forthcoming sound recording published by Bohn Media.

APPENDIX C

NAME SYSTEM USED IN MAX/MSP PROGRAMS

The Max/MSP graphical programming environment allows a programmer to nest portions of a patch into subpatches and also to save subpatches as individual abstractions that may be used in any patch like standard objects in the Max/MSP libraries. While static variables are not usually declared and assigned values as in most text-based programming languages, objects such as send and receive, pvar, and the buffer recording and playing objects give the programmer many opportunities to organize names and commands in a patch. When using message boxes or the qlist object to broadcast commands throughout a patch without direct connections, the naming system can become part of a scripting vocabulary unique to each patch or programmer.

In sophisticated patches, it is important to use names that are consistent, easily recalled and understood, but not excessively long. Longer names take more time to type and open more chances for typographical errors. Many programmers use their initials as a prefix followed by a period and a unique descriptive name. My practice has been to capitalize names I create, in order to distinguish proper names from names ordinary to Max/MSP, and to capitalize abbreviations when concatenated (sometimes called "camel case"). To meet the needs described above, I have developed a standard set of abbreviations and suffixes that indicate format or function. The naming system is not entirely consistent among all three programs, because it was at different stages in development when each program was created, and also because the technical needs of each program are different. Below is a list of common abbreviations and extensions used in these programs, along with a few examples from the programs.

Table C.1. Extensions used for names in Max/MSP.

.1 (or any number): An instance number.

.aud: Audio signal.

.b: *Bang*, a generic trigger without a value.

.cmd: For sending or receiving varied commands.

.ctl: Control.

.dpy: User display element.

.f: Floating point (real non-integer) number.

.i: Integer number.

.in: Input for a module, often an internal audio signal.

.L: Concerning the left audio channel.

.mde: Mode.

.mnu: Menu in a user interface.

.ms: Value in milliseconds.

.mx: Coefficient or multiplier.

.out: Output of a module, often an internal audio signal.

.pre: A value that needs to be processed before passing it on to the rest of the program.

.pst: Preset.

.pvar: Connected to a pvar object for remote connections to graphic interface objects without connecting patch cords to them.

.R: Concerning the right audio channel.

.set: Update a user interface control without it sending control information.

.sig: Signal rate data usually not meant to be heard.

.tog: Toggle, 0 or 1 only.

.txt: Text, as for display to the user.

Table C.2. Abbreviations used for names in Max/MSP.

1 (or any number): An instance number.

Amp: Amplitude.

Beg: Beginning.

DAC: Digital-to-analong converter, e.g., DAC.tog activates or deactivate all audio processing.

Dcy: Decay time.

Del: Delay.

End: Ending.

Len: Length, usually in milliseconds.

Max: Maximum.

Min: Minimum.

Sel: Selection or selected area.

Vol: Volume, usually controlling final audio output.

Table C.3. Example names from the programs. Key words related to abbreviations are shown in bold typeface.

Stop.b: Send a **bang** to trigger routines to **stop** performance.

Voice.out: Audio **output** of the **Voice** abstraction sent to a master mixer before being heard.

4Pitch.mx: **Pitch multiplier** for bank **4**.

2SelBeg.ms: **Beginning** point in **milliseconds** of the **selected** region in bank **2**.

PlayerDelMax.ms: **Maximum** allowed **delay** setting in **milliseconds** for all **Player** modules.

3BufferTog.pre: The upper left gamepad trigger controls **bank 3** and **toggles** between 0 and 1. Pressing the trigger either triggers playback or loads recorded audio into the buffer, depending on whether recording was enabled at the time, so the control first enters a **pre**processing stage to determine the appropriate result.

APPENDIX D

ADDITIONAL ELEMENTS OF THE *ELEKTRODYNAMIK* MAX/MSP PATCH

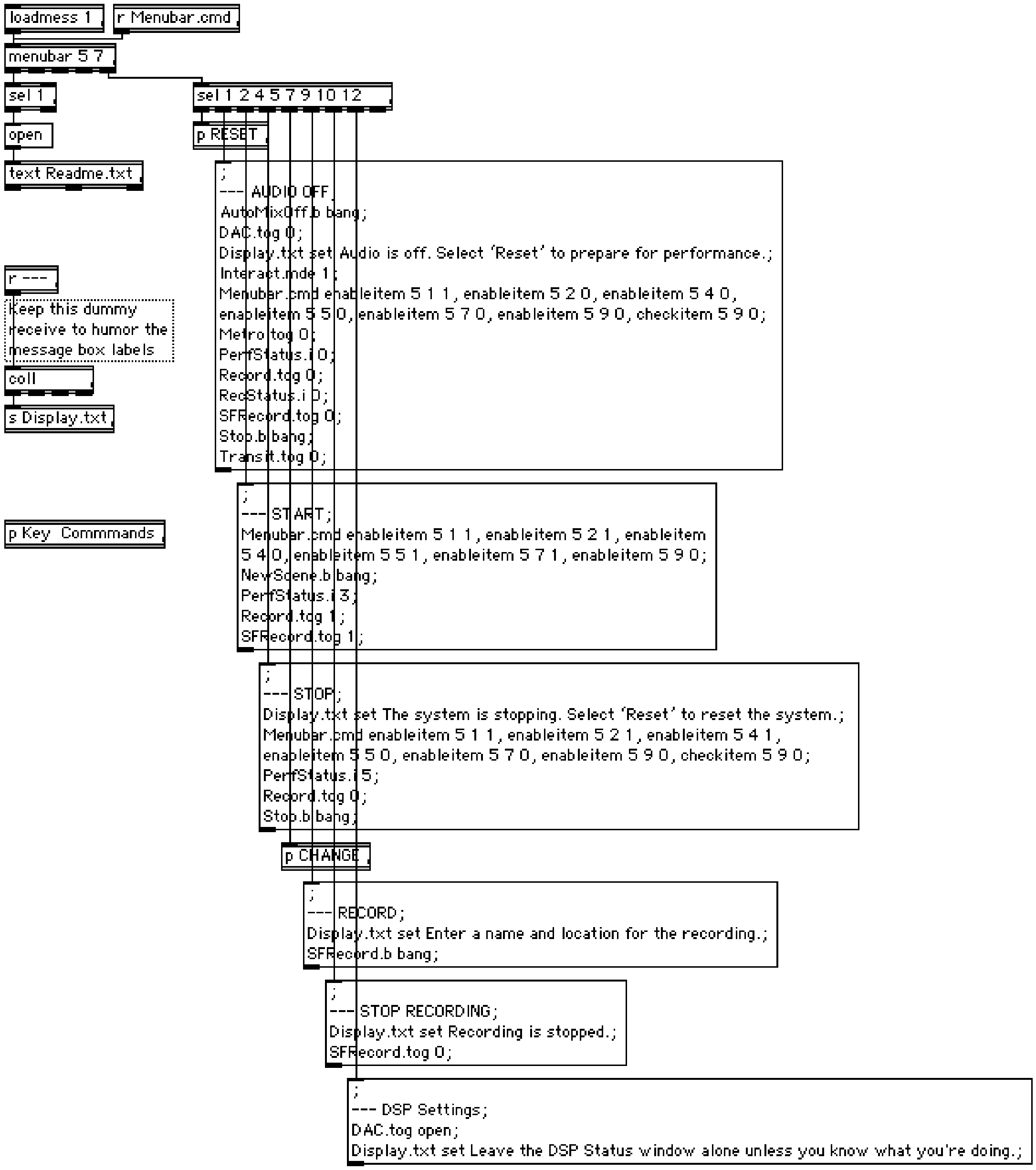

FIGURE D.1. The Menu subpatch creates a customized menubar and sends commands in response to menu selections (screen 1 of 2).

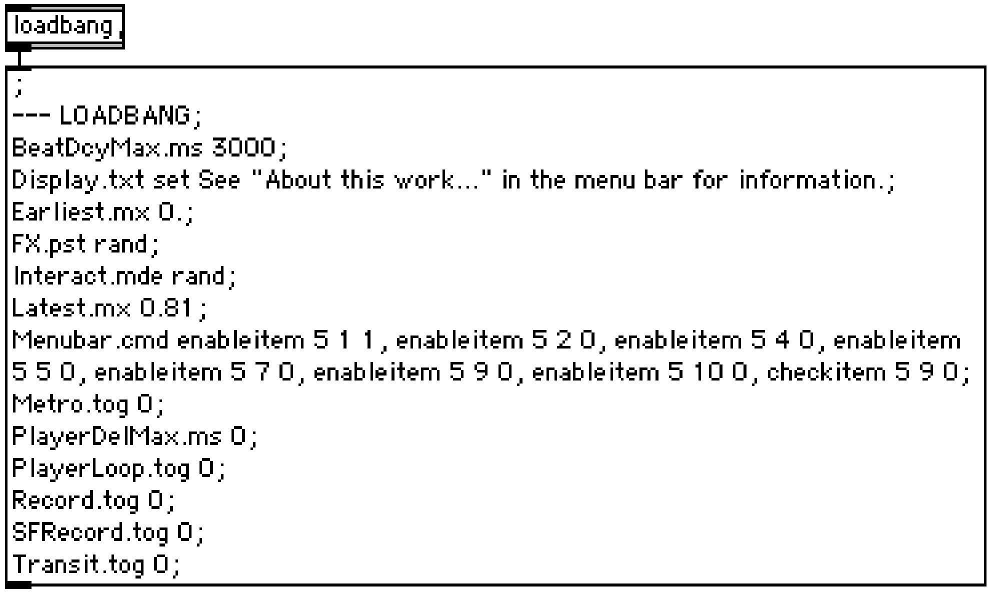

Figure D.2. The Menu subpatch (screen 2 of 2). This portion sends initialization commands when the program is first loaded.

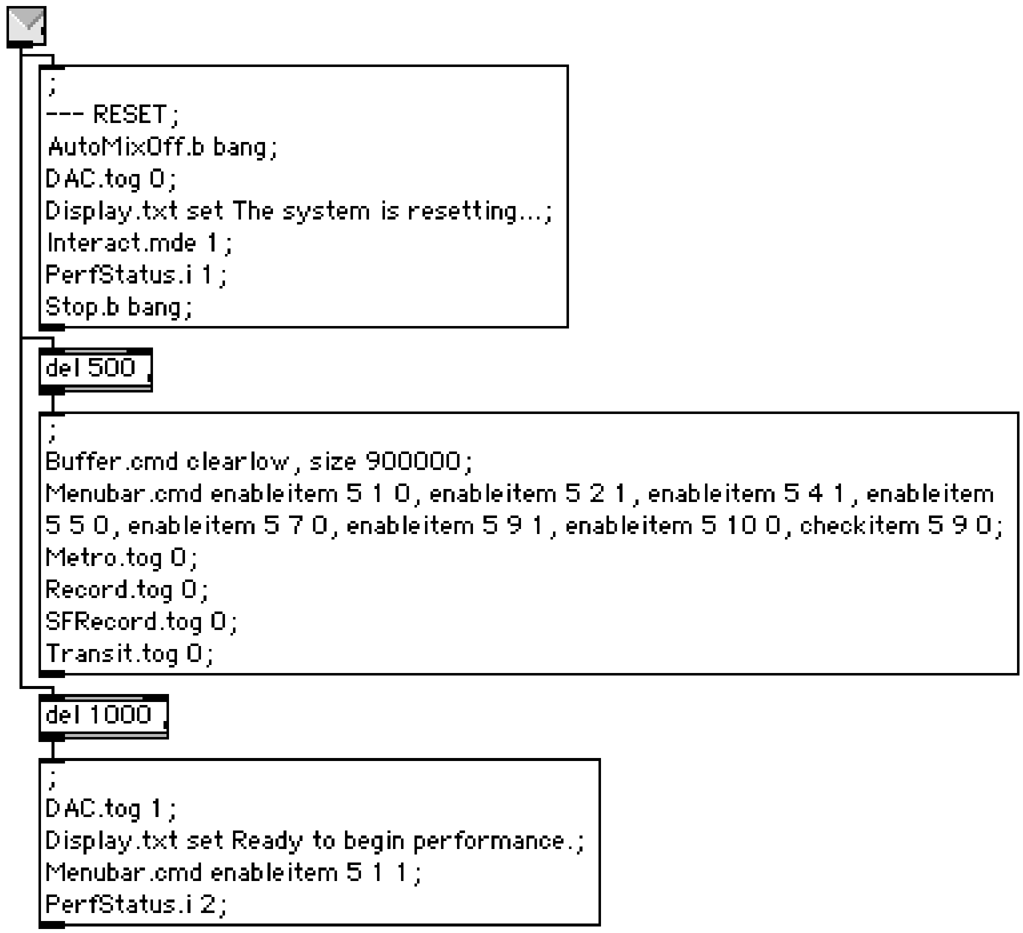

FIGURE D.3. The RESET subpatch sends commands in a three-part sequence to initialize the program.

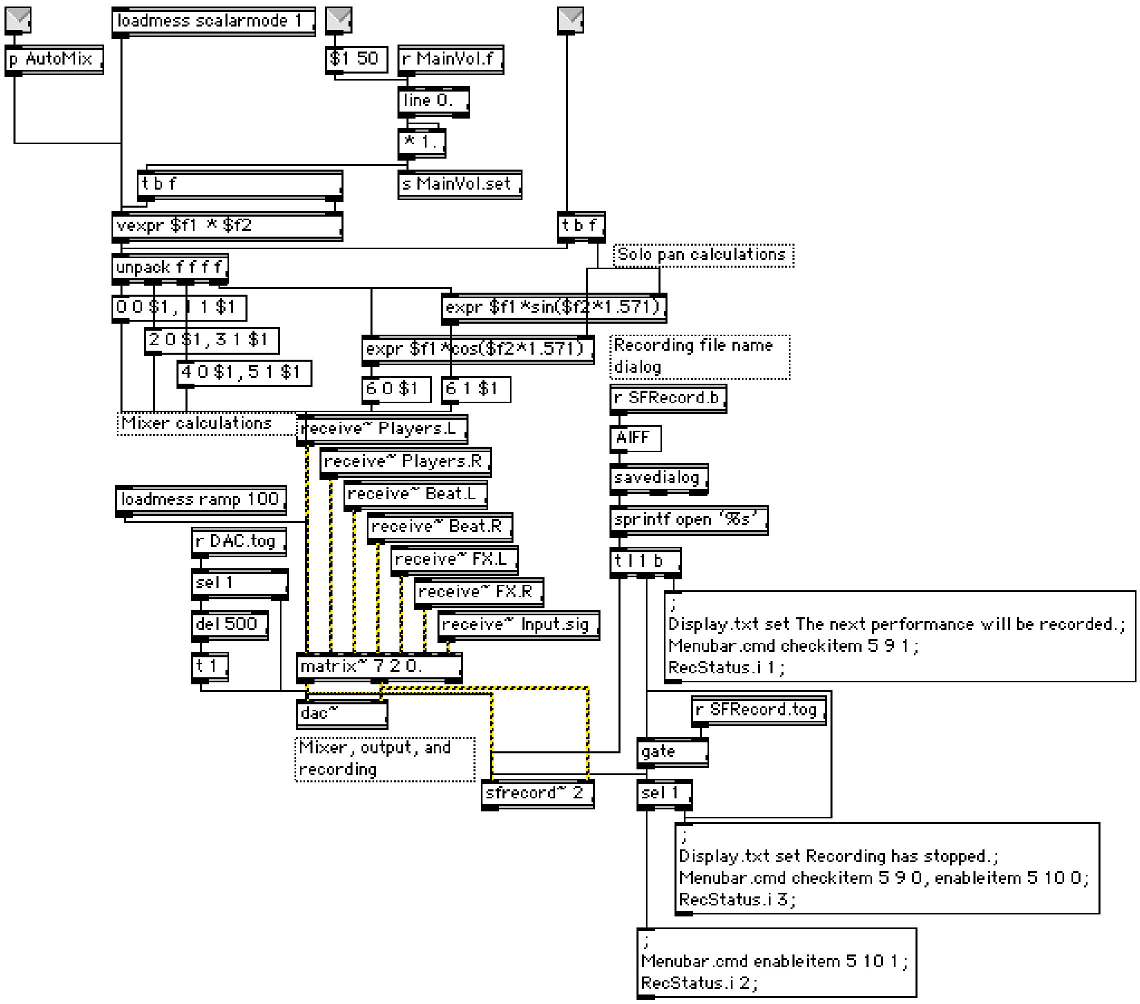

FIGURE D.4. The Mixer subpatch processes output levels.

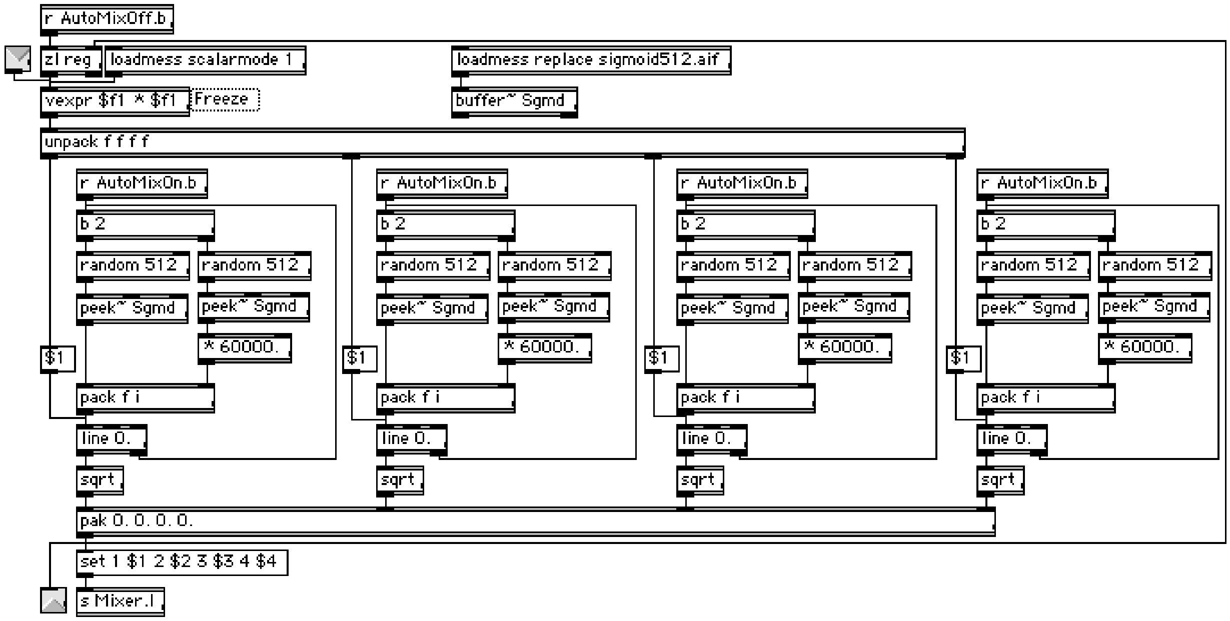

FIGURE D.5. The Automix subpatch automatically changes the output levels of each voice when activated, and stops changing values upon manual intervention.

APPENDIX E

ADDITIONAL ELEMENTS OF THE GAMEPAD SAMPLER MAX/MSP PATCH

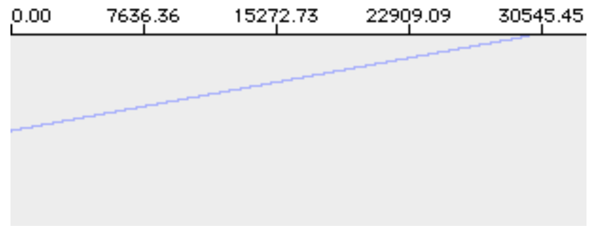

FIGURE E.1. The function 30SecRamp, a linear ramp from zero to one over thirty seconds, used to facilitate continuously changing loop points in granular playback (see figure 4.10 on page 54).

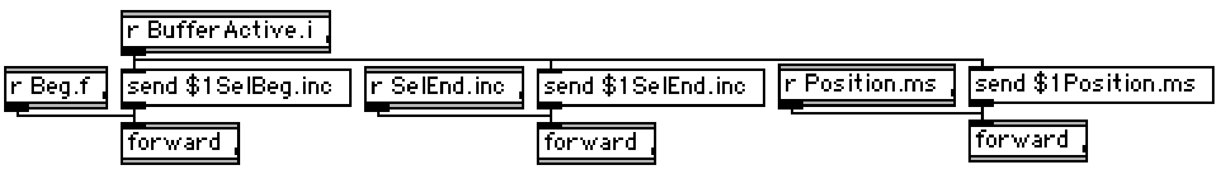

FIGURE E.2. The Buffer Display Messages subpatch forwards incoming controls for selection beginning, selection end, and scrub position to the currently active bank.

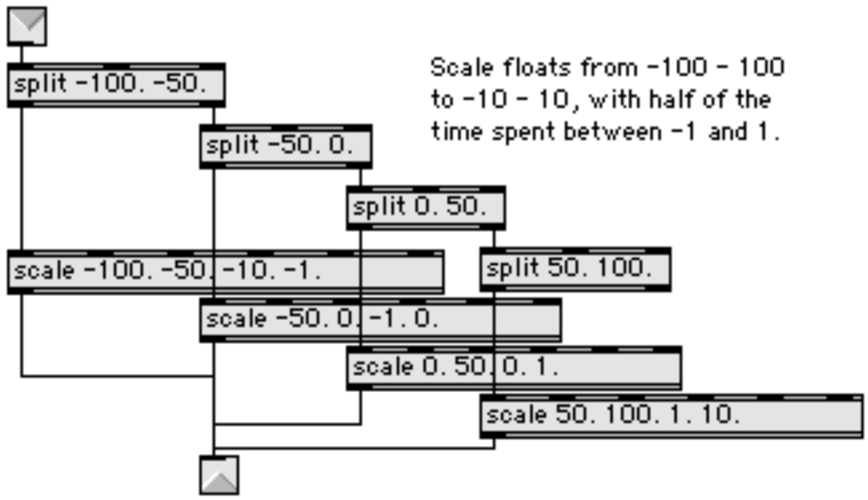

FIGURE E.3. The MxCurve abstraction maps a range of numbers to a function of three linear segments.

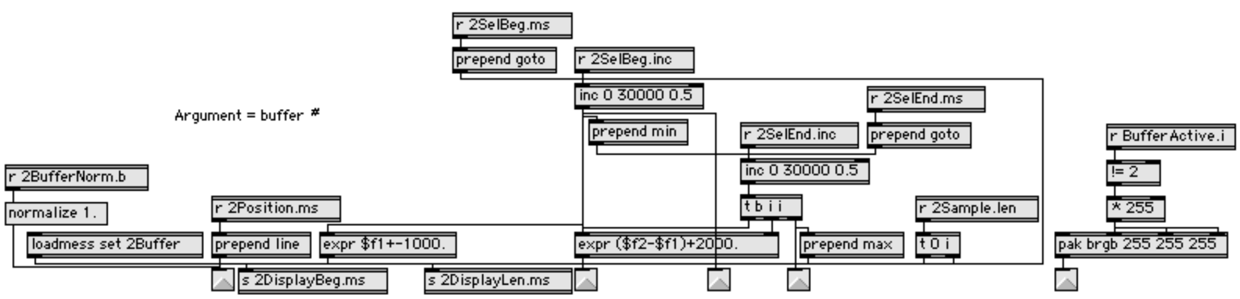

FIGURE E.4. The Waveform.mgr abstraction updates the waveform display of each bank.

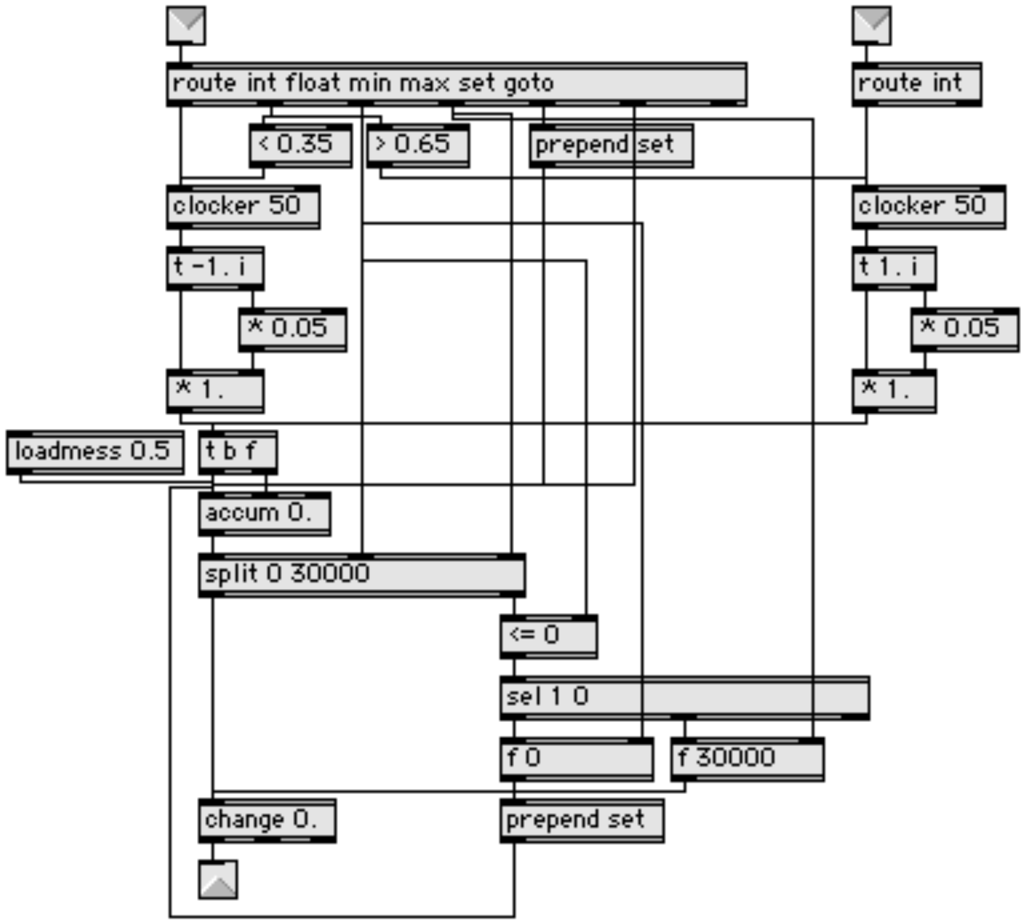

FIGURE E.5. The Inc abstraction facilitates the use of the directional pad and sticks to change parameters incrementally while held in one position.

BIBLIOGRAPHY

Adorno, Theodor W., Christopher Hailey (translator), and Juliane Brand (translator), 1994. *Alban Berg: Master of the Smallest Link*. Cambridge University Press.

Aufermann, Knut F., 2002. "Feedback Processes: An Investigation Into Non-linear Electronic Music." Master's thesis, Middlesex University. URL `http://knut.klingt.org/docs/Knut%20Aufermann%20MA%20Sonic%20Arts.pdf` (accessed August 5, 2007).

Ballou, Glen, 2005. *Handbook for Sound Engineers*. Focal Press, 3rd ed.

Benjamin, Walter, 1936/1969. "The Work of Art in the Age of Mechanical Reproduction." In *Illuminations: Essays and Reflections*, edited by Hannah Arendt. Schocken.

Bode, Harald, 1984. "History of Electronic Sound Modification." *Journal of the Audio Engineering Society* 32, no. 10: 730–739.

Brown, Earle, 1952–4. *December 1952* in *Folio and 4 Systems*. G. Shirmer.

Brown, James, 1996. *Foundations Of Funk: A Brand New Bag: 1964–1969* (sound recording boxed set). Polydor/Umgd.

Chadabe, Joel, 1997. *Electric Sound: The Past and Promise of Electronic Music*. Prentice Hall.

Collins, Nicolas, 1974–76/2004. *Pea Soup*. apestaartje. Concert piece or installation for self-stablizing feedback system performed by George Cremaschi, double bass, at the 1999 Festival Limbo 2 in the Plasy Monastery, Czech Republic.

———, 2002. "All This and Brains Too: 30 Years of Howling Round." *Resonance (London Musicians' Collective)* 9, no. 2. URL `http://www.l-m-c.org.uk/text/feedback/howling_round.pdf` (accessed August 5, 2007).

———, 2006. *Handmade Electronic Music: The Art of Hardware Hacking*. Routledge, 1st ed.

Dannenberg, Roger B., 2006. "Using Audio Feedback in Live Performance." In *Proceedings of the Fourth Annual Spark Festival of Electronic Music and Art*, 70–71. Minneapolis:

University of Minnesota School of Music.

Davies, Paul C. W., 2006. *About Time: Einstein's Unfinished Revolution.* Pengiun, revised ed.

Davies, Stephen, 2007 [forthcoming]. "Versions of Musical Works and Literary Translations." In *Philosophers on Music: Experience, Meaning and Work.* Oxford University Press.

d'Escriván, Julio, 2006. "To Sing the Body Electric: Instruments and Effort in the Performance of Electronic Music." *Contemporary Music Review* 25, no. 1–2: 183–191.

Dodd, Julian, 2004. "Types, Continuants, and the Ontology of Music." *British Journal of Aesthetics* 44, no. 4: 342–360.

Eco, Umberto, 1989. *The Open Work.* Harvard University Press.

Einstein, Albert, 1905. "Zur Elektrodynamik bewegter Körper." *Annalen der Physik und Chemie* 17: 891–921.

Enns, Anthony, 2005. "Voices of the Dead: Transmision/Translation/Transgression." *Culture, Theory, & Critique* 46, no. 1: 11–27.

Flake, Gary William, 1998. *The Computational Beauty of Nature: Computer Explorations of Fractals, Chaos, Complex Systems, and Adaptation.* MIT Press.

Föllmer, Golo, 2001. "Soft Music." URL `http://crossfade.walkerart.org/foellmer/text_print.html` (accessed August 5, 2007).

Gann, Kyle, 1995. *The Music of Conlon Nancarrow.* Cambridge University Press.

Garnett, Guy E., 2001. "The Aesthetics of Interactive Computer Music." *Computer Music Journal* 25, no. 1: 21–33.

Ghazala, Reed, 2005. *Circuit-Bending: Build Your Own Alien Instruments.* Wiley.

Harrison, Nate, 2004. "Can I Get an Amen?" art installation including the history of the famous "Amen break" exhibited February 22–26, 2006 at Frederick R. Weisman Art Museum, Minneapolis, Minnesota during the Spark Festival of Electronic Music and Art. URL `http://nkhstudio.com/pages/popup_amen.html` (accessed August 5, 2007).

Heyman, Edward, Robert Sour, Frank Eyton, and Johnny Green, 1930. "Body and Soul."

Jehan, Tristan, 2005. "Creating Music by Listening." Ph.D. thesis, Massachusetts Insitute of Technology. URL http://web.media.mit.edu/~tristan/phd (accessed August 5, 2007).

Kaku, Michio, 2004. *Einstein's Cosmos: How Albert Einstein's Vision Transformed Our Understanding of Space and Time*. W. W. Norton & Company.

Karplus, K. and A. Strong, 1983. "Digital Synthesis of Plucked-String and Drum Timbres." *Computer Music Journal* 7, no. 2: 43–55.

Lewis, George E., 2000. "Too Many Notes: Computers, Complexity and Culture in *Voyager*." *Leonardo Music Journal* 10: 33–39.

Lopes, Dominic M. McIver, 2001. "The Ontology of Interactive Art." *Journal of Aesthetic Education* 35, no. 4: 65–81.

Lucier, Alvin, 1969. "I am Sitting in a Room." Fixed-media composition.

———, 1995. *Reflections: Interviews, Scores, Writings*. Köln: MusikTexte.

———, 2002. "My Affairs with Feedback." *Resonance (London Musicians' Collective)* 9, no. 2. URL http://www.l-m-c.org.uk/text/feedback/affairs_feedback.pdf (accessed August 5, 2007).

Mandelbrot, Benoît, 1982. *The Fractal Geometry of Nature*. W. H. Freeman & Co.

McNutt, Elizabeth, 2003. "Performing Electroacoustic Music: A Wider View of Interactivity." *Organised Sound* 8: 297–304.

Meyers, David Lee, 2002. "A Personal Journey Through the Ourobouros." *Resonance (London Musicians' Collective)* 9, no. 2. URL http://www.l-m-c.org.uk/text/feedback/ourobouros.pdf (accessed August 5, 2007).

Morris, Jeffrey M., 2004. "A Dynamic Model of Metric Rhythm in Electroacoustic Music." In *Proceedings of the International Computer Music Conference*. International Computer Music Association, University of Miami.

———, 2005. *Tappatappatappa*. Improvised performance using *Zur Elektrodynamik bewegter Musiker* with live electronics and feedback.

———, 2007 [forthcoming]. "Disturbances." On *Clairaudience* (sound recording). New Bedford, MA: Bohn Media.

Morris, Lawrence D. "Butch", 1995. *Testament: A Conduction Collection* (booklet accompanying sound recording boxed set). New World Records. URL `http://www.newworldrecords.org/linernotes/80482.pdf` (accessed August 5, 2007).

Neuhaus, Max, 1994. "The Broadcast Works and Audium." *Zeitgleich* 19–31. URL `http://www.max-neuhaus.info/audio-video/Broadcast_Works_and_Audium.pdf` (accessed August 5, 2007).

Phelan, Peggy, 1993. *Unmarked: The Politics of Performance*. Routledge.

Polonsky, Abraham, Robert Rossen (director), and Bob Roberts (producer), 1947. *Body and Soul* (motion picture). United Artists.

Pritchett, James, 2004. "David Tudor as Composer/Performer in Cage's *Variations II*." *Leonardo Music Journal* 14.

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

Rohrbaugh, Guy, 2003. "Artworks as historical individuals." *European Journal of Philosophy* 11, no. 2: 177–205.

Rowe, Robert, 1992. *Interactive Music Systems: Machine Listening and Composing*. MIT Press.

———, 2001. *Machine Musicianship*. MIT Press.

Seaman, William Curtis, 1999. "Recombinant Poetics: Emerging Meaning as Examined and Explored Within a Specific Generative Virtual Environment." Ph.D. thesis, Centre for Advanced Inquiry in the Interactive Arts, University of Wales. URL `http://digitalmedia.risd.edu/billseaman/pdf/recombinantPoeticsDis.pdf` (accessed August 5, 2007).

Stockhausen, Karlheinz, 1956. *Klavierstück XI*. London: Universal Edition.

The Winstons, 1969. "Amen, Brother." On *Color Him Father* (sound recording). Metromedia MMS-117.

Vaggione, Horacio, 2001. "Some Ontological Remarks about Music Composition Processes." *Computer Music Journal* 25, no. 1: 54–61.

Wark, McKenzie, 2004. *A Hacker Manifesto*. Harvard University Press.

Wolff, Christian, 1964. *For 1, 2, or 3 People*. Glendale, NY: C. F. Peters.

Wolfram, Stephen, 2002. *A New Kind of Science*. Wolfram Media. URL http://www.wolframscience.com/nksonline (accessed August 5, 2007).