Spring 2010

Normal mode for chamber ensemble and electronics

Israel Neuman
University of Iowa

NORMAL MODE
FOR CHAMBER ENSEMBLE AND ELECTRONICS

by

Israel Neuman

An Abstract

Of a thesis submitted in partial fulfillment
of the requirements for the
Doctor of Philosophy degree in Music
in the Graduate College of
The University of Iowa

May 2010

Thesis Supervisor: Associate Professor Lawrence Fritts

ABSTRACT

Normal Mode is a composition for chamber ensemble and electronics that makes reference to the microtonality employed in Turkish music. In this composition I have made an attempt to expand the timbral palette of standard Western instruments by the use of electronic sounds, which were constructed through digital sound synthesis. The microtonal frequencies, which were used in this synthesis process, were derived from the Turkish tonal system. The ensemble material, on the other hand, was conceived within a Western-influenced serial pitch organization. These two distinct influences invite a dynamic discourse between the ensemble and the electronics. As a new instrument, which was developed specifically for this composition, the electronics initially attract more attention. Over time a new equilibrium is established and the electronics part is integrated in the ensemble.

The electronics part of *Normal Mode* was created in the object-oriented programming environment Max/MSP. It is realized in a performance of the composition with the same software. Five of the chapters of this thesis discuss the compositional process of the electronic part and the system of organization that guided this process. These chapters describe how this system was incorporated in the programming of Max/MSP patchers which generated the composition's sound library and perform the electronics part in real time. They also describe the relationships between the ensemble and the electronics. The sixth chapter presents the composition *Normal Mode*. The Max/MSP patchers that perform the electronics part are included in the supplement of this thesis.

Abstract Approved: ______________________________
Thesis Supervisor

____
Title and Department

____
Date

NORMAL MODE
FOR CHAMBER ENSEMBLE AND ELECTRONICS

by

Israel Neuman

A thesis submitted in partial fulfillment
of the requirements for the
Doctor of Philosophy degree in Music
in the Graduate College of
The University of Iowa

May 2010

Thesis Supervisor: Associate Professor Lawrence Fritts

Graduate College
The University of Iowa
Iowa City, Iowa

CERTIFICATE OF APPROVAL

PH.D. THESIS

This is to certify that the Ph.D. thesis of

Israel Neuman

has been approved by the Examining Committee for the thesis requirement for Doctor of Philosophy degree in Music at the May 2010 graduation.

Thesis Committee: ___________________________________
Lawrence Fritts, Thesis Supervisor

____
David Gompper

____
Robert Cook

____
John Rapson

____
Bryon Winn

To Yi and Amitai, and to Mina

ACKNOWLEDGMENTS

I would like to thank my mentor and thesis supervisor, Lawrence Fritts, for his guidance and support, and for sharing his inspiring knowledge with me. I would also like to thank my committee members, David Gompper, John Rapson, Robert Cook and Bryon Winn, for their contributions to my education over the years I have been a student at the University of Iowa. Finally, I would like to express my deepest gratitude to my wife, Yi, and my mother, Mina, for their encouragement, love and support, and most of all for their patience.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

In 1987, the *Computer Music Journal* devoted its spring issue to microtonality. Subsequently, in the winter of 1991, the journal *Perspectives of New Music* published an issue entitled "Forum: Microtonality Today." These publications reflected a growing interest among contemporary composers and scholars in the research and application of tuning systems and octave divisions that can serve as alternatives to the twelve-note equal-tempered Western tuning. The term microtonality often refers to any non-standard tuning or octave division whether or not it incorporates intervals smaller than the semitone.[1] Electronic music plays a prominent role in these explorations of new grounds. The computer is considered by many to be the best tool for this task as it is free of any limitations that can be imposed by the performer's technique or the instrument's mechanical and physical properties.

Composers interested in microtonality seek primarily to expand the pitch domain of contemporary music. John Eaton claims that "microtones permit a greater variety of harmonic and melodic motion."[2] Douglas Keislar maintains that microtonality is a "logical outgrowth" of the "increasingly complex pitch relationships" of Western music.[3] "Moving beyond twelve-note equal temperament," he elaborates, "can be a doorway to ever more intricate schemata in the serialist tradition."[4] On the other hand, according to Keislar, microtonality can also provide a breakaway from complexity or "a route to

---

1. Douglas Keislar, "Introduction," *Perspectives of New Music* 29, no. 1 (Winter 1991): 173.

2. Douglas Keislar et al., "Six American Composers on Nonstandard Tunings," *Perspectives of New Music* 29, no. 1 (Winter 1991): 184.

3. Keislar, "Introduction," 174.

4. Ibid.

simplification without regression."[5]

Octave divisions and non-standard tunings vary from one composer to another and from one composition to another. However, a few trends may be identified among contemporary composers. One trend is to introduce microtonal intervals in combination with the twelve-note equal-tempered octave division. This approach is exemplified in the works of Charles Ives, Alois Hába and Ivan Vïshnegradsky.[6] Another trend is associated with the composers Harry Partch, Eivind Groven, Lou Harrison, Ben Johnston, La Monte Young and James Wood. These composers have used just intonation and just interval of frequency ratios involving prime numbers.[7] Equal-tempered octave divisions in different intervals have been used by composers such as Joseph Yasser, Adriaan Fokker, Henk Badings, Hans Kox and Easley Blackwood.[8] A fourth trend includes composers who employ octave divisions derived from non-western musical cultures. Lou Harrison and Larry Polansky, among other composers, have both worked with central Javanese *slendro* tunings.[9]

While microtonality is mainly employed in the pitch organization of compositions, many composers and scholars point out its strong effect in the timbral domain. Easley Blackwood maintains that "often the timbre of an instrument is all wrapped up in the

---

5. Ibid.

6. Paul Griffiths, Mark Lindley, and Ioannis Zannos, "Microtone," in *Grove Music Online*, *Oxford Music Online*, http://www .oxfordmusiconline.com/subscriber/article/grove/music/18616 (accessed January 7, 2010).

7. Ibid.

8. Ibid.

9. Keislar et al., "Six American Composers on Nonstandard Tunings," 178, and Larry Polansky, "Paratactical Tuning: An Agenda for the Use of Computers in Experimental Intonation," *Computer Music Journal* 11, no. 1 (Spring 1987): 62-64.

tuning."[10] Ben Johnson draws attention to the change in timbre that takes place due to the utilization of different performance techniques in the execution of an alternative tuning:

> To get the intonation, the players might use alternate fingerings, lip the notes high and low, or pull out the barrel of the instrument, all of which change the timbre. You get a flute that sounds more like a shakuhachi, or instead of an oboe you get different colors of oboe.[11]

Wendy Carlos identifies mutual relationships between timbre and tuning: "Clearly the timbre of an instrument strongly affects what tuning and scale sound best on that instrument, and exactly vice versa."[12]

Microtones play an important role in the work of spectral composers such as Tristan Murail, Gérard Grisey and Claude Vivier, who have derived their influences from the spectral analysis of sound and the acoustical research of timbre.[13] The relationships between timbre and microtonality are explored by studies of tuning systems, such as the studies by Wendy Carlos and William A. Sethares.[14] These studies evaluate different tunings by examining the consonant and dissonant relations created when different timbres or harmonic spectra were applied to these tunings. They show that timbre affects the perception of consonant and dissonant relations. They provide lesser account for the effect of the tuning on timbre, which is pointed out by Carlos and Blackwood. The timbre of acoustic instruments is, after all, mainly the result of the instruments' physical

---

10. Keislar et al., "Six American Composers on Nonstandard Tunings," 185.

11. Ibid., 184-185.

12. Wendy Carlos, "Tuning: At the Crossroads," *Computer Music Journal* 11, no. 1 (Spring 1987): 35.

13. Griffiths, Lindley, Zannos, "Microtone," and Julian Anderson, "Spectral music," in *Grove Music Online*, *Oxford Music Online*, http://www.oxfordmusiconline.com/subscriber/article/grove/music/50982 (accessed February 24, 2010).

14. Carlos, "Tuning: At the Crossroads," 29-43; William A. Sethares, *Tuning, Timbre, Spectrum, Scale* (London: Springer-Verlag, 2005), http://www .springerlink.com/content/w04151 (accessed October 15, 2009).

properties and performance practices.

In electronic music, microtonality is commonly used in combination with digital sound synthesis, although composers of electronic music incorporated microtonality before the appearance of computers in that field. Stockhausen composed *Studie I* (1953) and *Studie II* (1954) based on non-standard tunings and realized them with electronic sound generators and an analog four-track tape recorder.[15] Nevertheless, the computer in general and the digital synthesis in particular offer the composer more control over the components of sound, and therefore invites the incorporation of microtonality. In regard to the capabilities of "computer-controlled synthesis," Wendy Carlos writes:

> The exponential growth in computers has finally expanded to include systems designed expressly for music production, editing, and performance at low enough cost to be as affordable as, say, a good grand piano. This is the first time instrumentation exists that is both powerful enough and convenient enough to make practical the notion: any possible timbre, in any possible tuning, with any possible timing, sort of a "three T's of music."[16]

In 1987, when these words were published, this was more a prediction of the future than a description of reality, although, in 1986, Douglas Keislar had already developed a computer program for real-time microtonal control of digital synthesizers.[17] The full realization of Carlos' prophecy only arrived with more recent developments in this field such as the suite of analysis-resynthesis programs Spectral Toolbox and the application TransFormSynth. This software enables both the control over the sound's timbral

---

15. Paul Griffiths, *Modern Music and After: Directions Since 1945* (Oxford: Oxford University Press, 1995), 45-46; Joel Chadabe, *Electronic Sound: the Past and Promise of Electronic Music* (New Jersey: Prentice-Hall Inc., 1997), 37-38.

16. Wendy Carlos, "Tuning: At the Crossroads," 31.

17. Douglas Keislar, "History and Principles of Microtonal Keyboards," *Computer Music Journal* 11, no. 1 (Spring 1987): 24-26.

characteristics and the mapping of different timbres with different tunings.[18]

Normal Mode is a composition for chamber ensemble and computer that makes reference to the microtonality employed in Turkish music. In this composition I have made an attempt to expand the timbral palette of standard Western instruments by the use of electronic sounds which were constructed through digital sound synthesis. The microtonal frequencies, which were used in this synthesis process, were derived from the Turkish tonal system. These frequencies were organized in a system that I have developed specifically for this composition. The implementation of this system within the synthesis process led to the construction of sounds which can be generalized into two types: inharmonic sounds and noise-type sounds. The library of sounds, which is utilized in the performance of this piece, includes sounds of a wide variety of mixes between these two types.

Tristan Murail writes the following regarding the influences on contemporary music and what he defines as the "revolution of the world of sounds:"[19]

> I perceive a double influence of electroacoustic music and non-western musics, which have enabled us to discover a different sense of time; they have led us to alternative methods of orienting ourselves to duration; through them, we are now attentive to phenomena previously considered secondary: microfluctuations of many kinds, sound colours, the production of sound, etc.[20]

Murail also points out the significant contribution of inharmonic sounds to this revolution. "Inharmonic spectra themselves," he says, "give rise to particularly rich and

---

18. William A. Sethares et al., "Spectral Tools for Dynamic Tonality and Audio Morphing," *Computer Music Journal* 33, no. 2 (Summer 2009): 71-84, http://muse.jhu.edu/journals/computer_music_journal/v033/33.2.sethares.pdf (accessed October 10, 2009).

19. Tristan Murail, "The Revolution of Complex Sounds," *Contemporary Music Review* 24, no. 2/3 (April/June 2005): 122, http://web.ebscohost.com/ehost/pdf?vid=3&hid=11&sid=dd148275-3368-451d-be50-747b7637e641%40sessionmgr12 (accessed October 13, 2009).

20. Ibid.

interesting spectra and can be classified under this new category of complex sounds, since they resist analysis as either harmonies or timbres."[21] Murail's description of inharmonic sounds as neither harmonies nor timbres may be explained by the following comments by Trevor Wishart: "The object [the inharmonic sound] appears to perception as an aggregate of various pitches, more fused than a typical chord in instrumental music but definitely not a singly-pitched note."[22] In other words, whereas the spectral components of harmonic sound are fused to constitute a distinct timbre, the spectral components of inharmonic sound are independent enough to be noticeable, yet fused enough not to sound like a chord or a cluster. Unlike Murail, Wishart argues that inharmonic sounds "are not radically different from normal sound-objects found in conventional musical practice."[23] Nevertheless, he describes noise-type sounds and inharmonic sounds as the two extremes of "a multidimensional array of complex possibilities."[24]

The synthesis process that generated the sound library of *Normal Mode* and the composing of the electronics part were both done in Max/MSP. The creation of this two-part programming environment is credited to two people: Miller Puckette, who initiated Max in the mid-nineteen-eighties, and David Zicarelli, who released MSP about ten years later. MSP was initially introduced by Zicarelli as a supplement to Opcode's Max 3.5 graphical programming environment, which enables signal processing.[25] Nowadays it is a

21. Ibid., 129.

22. Trevor Wishart, *On Sonic Art*, ed. Simon Emmerson, (London: Routledge 2002), 58.

23. Ibid.

24. Ibid.

25. Joseph B. Rothstein, "Products of Interest," *Computer Music Journal* 22, no. 3 (Autumn 1998): 75.

fundamental and indispensable part of the program. Nevertheless the virtue of Max/MSP may still lie in its premise or what Puckette called the Max paradigm. He describes this paradigm as "a way of combining pre-designed building blocks into configurations useful for real-time computer music performance."[26] Max/MSP real-time capability was established, according to Puckette, on the concept of parallel multi-tasking that allows the instantaneous implementation of elaborated tasks.[27] A primary influence on the Max paradigm, as stated by Puckette, was the program RTSKED.[28] The creator of RTSKED, Max Mathews, was also the inspiration for the name Max.[29]

The real-time performance capability of Max/MSP attracted many composers over the years. Miller Puckette writes about the first attempts to utilize Max as a real-time performance tool:

> The new Macintosh version of Max, which eventually grew into what is now Max/MSP, was first used on stage in a piece by Frederic Durieux in early 1988. But it was Philippe Manoury's *Pluton*, whose production started in Fall 1987 and which premiered in July 1988, that spurred Max's development into a usable musical tool. The *Pluton* patch, now existing in various forms, is in essence the first Max patch.[30]

The onstage realization of *Pluton* is described by Puckette as a complex operation that required two computers: a Macintosh as a control computer and an IRCAM 4X for audio processing.[31] Even so, the efficiency of Max/MSP has improved tremendously over the years. In 2003, the violinist composer Mari Kimura wrote about the methods she used to

---

26. Miller Puckette, "Max at Seventeen," *Computer Music Journal* 26, no. 4 (Winter 2002): 31.

27. Ibid., 31-33

28. Ibid.

29. Ibid.

30. Ibid., 34.

31. Ibid.

create "a Max/MSP 'one-touch' system." This system allows her simultaneously to play the violin and control the computer in her interactive compositions.[32] It requires a minimum of computer keystrokes and no offstage assistant.[33] Such an approach of simplifying the performance procedures has guided me in the creation of the patchers for the performance of *Normal Mode*. These patchers may appear in Chapter 4 as very complex. Nevertheless their operation is very simple. I specifically designed the computer part to be accessible for performance by practically anyone who would read the technician notes, although some previous knowledge of Max/MSP may help. The interfaces of these patchers, as shown in the technician notes, are also reduced to a few elements, which is essential for the performance of the piece.

As already reflected in this introduction, in *Normal Mode* I have drawn from multiple resources including the microtonality of Turkish music, digital sound synthesis and the real-time performance capability of Max/MSP. For the temporal organization of *Normal Mode* I have used additional resources such as Middle Eastern rhythmic patterns and the concept of the rhythm tree as it is described in the computer application OpenMusic. These resources will be discussed in Chapter 2, which describes the system of organization and the theoretical context that informed it. The compositional process of the electronics of *Normal Mode* was twofold. In the first stage I constructed a library of sounds through the synthesis process that will be described in Chapter 3. In the second stage, which will be discussed in Chapter 4, I composed the performance patchers that

---

32. Mari Kimura, "Creative process and performance practice of interactive computer music: A performer's tale," *Organised Sound: An International Journal of Music Technology* 8, no. 3 (December 2003): 289, http://iimp.chadwyck.com/articles/displayItemPDF.do?format=PAGE&PQID=1404967441&journalID=JID13557718&royaltiesid=LOUJID13557718&product=iimp&articleID=iimp00336104 (accessed January 1, 2010).

33. Ibid.

utilize this library in the performance of the piece. The fifth chapter of my thesis will discuss the compositional process of the ensemble material for which I have drawn influences from the work of serial composers. This chapter will also describe the relationships between the electronics and the ensemble material.

CHAPTER 2

THEORY AND METHODOLOGY

This chapter discusses the system of organization that guided the compositional process of the electronics of *Normal Mode* and the theoretical context that informed this system. The discussion will touch on a wide variety of topics, including the theory of Turkish music, theories of timbre, methods of sounds synthesis and Middle Eastern rhythmic structures, yet it will be confined only to aspects of these topics which were relevant to this compositional system. This information formed the foundation for a set of principle ideas and classifications that constitute the system. These seemingly unrelated topics are, therefore, woven together in this chapter by their contribution to this composition.

Turkish Music and Collections of Frequencies

Turkish music uses the Pythagorean tuning system which is based on string-length ratios between multiples of 2 and 3.[1] The Turkish tonal system employs a non-equal division of the octave into twenty-four notes. This division originates from the cycles of fifths and fourths presented in Figure 2.1.[2] The pure fifth in Pythagorean tuning is 702 cents (instead of 700 cents in the twelve-note equal-tempered system) and the pure fourth is 498 cents (instead of 500 cents). In a cycle of pure fifths the final C exceeds the perfect octave by about 23.4 cents, also known as the Pythagorean comma.[3] This comma is a fundamental element in Turkish music and it is the foundation for the six types of

---

1. Mark Lindley, "Pythagorean intonation," in *Grove Music Online*, *Oxford Music Online*, http://www.oxfordmusiconline.com/subscriber/article/opr/t114/e5424 (accessed December 9, 2008).

2. Karl L. Signell, *Makam: Modal Practice in Turkish Music* (Seattle: Asian Music Publications, 1977), 27.

3. Lalage Cochrane, "Pythagorean intonation," in *The Oxford Companion to Music*, ed. by Alison Latham, *Oxford Music Online*, http://www.oxfordmusiconline.com/subscriber/article/opr/t114 /e5424 (accessed December 9, 2008).

accidentals used in the Turkish tonal system. A Turkish accidental raises or lowers a note by a specific number of commas.[4] The six accidentals each have a notation symbol and a name in Turkish (see Table 1), however, to simplify the discussion they will be specified here by the number of commas they raise or lower.

Figure 2.1 Cycles of fifths and fourths

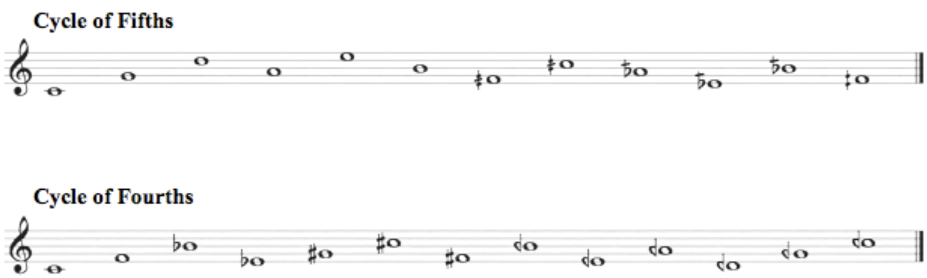


Table 1. Turkish accidentals

Turkish Name	Number of Commas	Notation Symbol
koma diyezi	raises 1 comma	
bakiye diyezi	raises 4 commas	
Küç•müc•diyezi	raises 5 commas	
Koma bemolü	lowers 1 comma	
Bakiye bemolü	lowers 4 commas	
Küç•müc•bemolü	lowers 5 commas	

The Turkish non-equal octave division has yielded a large variety of scales which are called *makams*. There are thirteen basic *makams* and many more compound and transposed *makams*. While theoretical studies maintain that all the basic *makams* can be

4. Signell, *Makam*, 24.

transposed, in practice some of the transposed *makams* receive different names, and usually *makams* are transposed to only a small number of transposition levels. The transposition of *makams* or fragments of *makams* often results in the adjustment of intervals.[5] Therefore, the concept of transposition in Turkish music differs from the Western concept of mod-12 transposition in which all interval structures are transposable to all levels of transpositions.

Figure 2.2 shows the twenty-four notes of the Turkish octave. Seven of these notes – D, E, F, G, A, B, C - and the octave D receive no accidentals. In my compositional system they are called natural notes. The rest of the notes receive comma-based accidentals and they are called microtones. For the sake of this composition the natural notes are treated as equal-tempered.[6] Each one of the microtones is defined as a deviation by a specific number of commas from a natural note and is named accordingly. For example, the second note in the Turkish octave is the microtone D 4 comma sharp as it is four commas higher then D.

The natural notes, as shown in Figure 2.2, add up to a Dorian scale. It can be observed in this figure that all of the whole-tone intervals of this Dorian scale and the semitone between B and C are divided by microtones. Hence there are six sets of microtones, each of which is framed by two natural notes. Every two framing natural notes constitute in my compositional system what I call a natural dyad. The natural notes E and F do not frame any microtones, therefore, they are not considered a natural dyad.

---

5. Ibid., 31.

6. Based on Robert Garfias, "Survivals of Turkish Characteristics in Romanian *Musica Lautareasca*," *Yearbook for Traditional Music* 13 (1981): 103, and Peter Manuel, "Modal Harmony in Andalusian, Eastern European, and Turkish Syncretic Music," *Yearbook for Traditional Music* 21 (1989): 78, in Turkish-influenced musical cultures, which were also influenced by Western music, some pitches and intervals were adjusted to the Western tempered tuning, while others maintained the Turkish tuning.

A set of microtones and its framing natural dyad constitute a segment. The six segments of the Turkish octave are specified in Table 2.

Figure 2.2 Turkish octave division

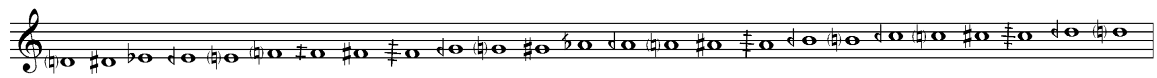


Table 2. The segments of the Turkish octave

Segment	Natural Dyad	Microtones
S1	D - E	D 4 commas sharp E 4 commas flat E 1 comma flat
S2	F - G	F 1 comma sharp F 4 commas sharp F 5 commas sharp G 1 comma flat
S3	G - A	G 4 commas sharp A 4 commas flat A 1 comma flat
S4	A - B	A 4 commas sharp A 5 commas sharp B 1 comma flat
S5	B - C	C 1 comma flat
S6	C - D	C 4 commas sharp C 5 commas sharp D 1 comma flat

In the compositional system of *Normal Mode*, each one of the notes in the Turkish octave represents a harmonic set. The latter is a set of sixteen frequencies extracted from the harmonic series of a note in the Turkish octave. The segment combines the harmonic sets of all its notes (natural dyad and microtones) into one collection, which is called a segment frequency collection (SFC). There are, therefore, six segment frequency collections which served as source collections of frequencies for the synthesis process of the sound library of *Normal Mode*.

Each time a sound was synthesized within this process an inharmonic set was selected. The inharmonic set is a set of frequencies derived by random selection from an SFC. Since the random selection of frequencies was from the entire SFC across the harmonic sets, the selected set of frequencies was not based on harmonic relations. The spectral components of sounds synthesized with such sets were not integer multiples of a fundamental frequency. Hence the synthesis process, when shaping the spectral envelopes, disregarded the role of fundamental frequency and none of the frequencies deliberately received higher amplitude. The sounds, which were created by this process, are inharmonic and undefined in pitch. These sounds may be perceived as having multiple pitches (yet not a chord) or simply as timbres.

Theories of Timbre and Methods of Sound Synthesis

The harmonic series is central to the understanding of timbre. The foundation for the modern study of timbre was laid by the mathematician Jean Baptiste Fourier and the scientist Hermann von Helmholtz.[7] Helmholtz concentrated on harmonic instrumental sounds.[8] He showed that the timbre of a sound is determined by the relative amplitudes of the spectral components. But because Helmholtz's spectral envelopes are static, they

7. Charles Dodge and Thomas A. Jerse, *Computer Music: Synthesis, Composition and Performance* (New York: Schirmer Books, 1997), 46.

8. Hermann L. F. Helmholtz, *On the Sensations of Tone as a Physiological Basis for the Theory of Music*, trans. Alexander J. Ellis (New York: Dover Publications, Inc., 1954).

describe only the steady portion of the sound. He did not examine the attack and the release, the unstable portions of the sound, nor did he incorporate the element of time in his spectral analysis.[9]

With the development of digital audio and the algorithm known as the fast Fourier transform (FFT) by Cooley and Tukey,[10] more recent studies of timbre, such as the studies by Jean-Claude Risset[11] and by James A. Moorer and John Grey,[12] were able to examine the spectrum's evolution in time. They discovered that the amplitudes and the frequencies of individual spectral components vary throughout the duration of a sound, mainly in its attack and release portions. These studies led to the introduction of time-evolving spectra, also known as the dynamic spectra, in digital sound synthesis, as reflected in the study by John Chowning.[13] The latter concentrated on frequency modulation. As Chowning maintained, the characteristics of the spectral evolution are the "signature" of the sound and the ability to reproduce them within a synthesized sound contributes to the "lively quality of a sound".[14]

Methods of sound synthesis developed hand in hand with the study of timbre. Nowadays a wide selection of methods is utilized both in the analysis and the resynthesis of sound. The synthesis process that generated the library of recorded sounds of *Normal*

---

9. Dodge and Jerse, *Computer Music*, 55.

10. Ibid.

11. Jean-Claude Risset, "Computer Music Experiments 1964 - ...," *Computer Music Journal* 9, no. 1 (Spring 1985): 11-12.

12. James A. Moorer and John Grey, "Lexicon of Analyzed Tones. Part I: A Violin Tone," *Computer Music Journal* 1, no. 2 (April 1977): 39-45, and James A. Moorer and John Grey, "Lexicon of Analyzed Tones. Part 2: Clarinet and Oboe Tones," *Computer Music Journal* 1, no. 3 (June 1977): 12-29.

13. John Chowning, "The Synthesis of Complex Audio Spectra by Means of Frequency Modulation," *Computer Music Journal* 1, no. 2 (April 1977): 46-54.

14. Ibid.

Mode utilized two methods: additive synthesis and frequency modulation. This process was based on the SFCs that were previously discussed. It also incorporated dynamic spectra. The methods of synthesis were adjusted to allow more independence and variation in time of the individual spectral components. The sounds, which were generated by this process, are inharmonic. Due to their dynamic spectra, the pitch and timbre of these sounds are constantly transformed throughout their duration.

Additive synthesis combines sine waves in various frequencies to a sum of sine waves that produces a more complex sound. The relative amplitudes of these sine waves determine the timbre of the resulting sound. If the amplitude of each sine wave is varied independently in time by an envelope, a dynamic spectrum is created. In the simulation of instrumental sound the frequencies are in harmonic relations and the envelopes duplicates the attack, decay, sustain and release that characterize acoustic sound. Yet the additive method can bear any set of frequencies and any combination of envelopes as it enables the precise control over the individual spectral components. The disadvantage of this method is that it requires a substantial amount of sine waves to generate rich and interesting sounds. In the context of digital synthesis this is translated into excessive computation requirements.

In frequency modulation, the waveform of one sine wave known as the carrier is varied by another sine wave, the modulator. As a result more components are added to the spectrum and a more complex sound is created. If the frequency ratio between the carrier and the modulator is a ratio of integers, the spectral components will be in harmonic relations. If this ratio is not a ratio of integers, then the spectral components will be in inharmonic relations. The characteristics of the spectrum are affected by the modulation index. In FM synthesis, the modulation index is related to the amplitude of the modulator. An increase in the amplitude of the modulator or an increase in the modulation index will result in a transfer of more energy from the carrier to other spectral components and in a richer spectrum. Varying the modulation index in time will result in a dynamic spectrum.

Temporal Organization

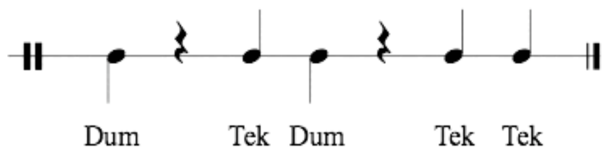
The library of sounds generated by the synthesis process was utilized in the compositional process of the electronics that will be discussed in Chapter 4. This process incorporated a temporal organization which derived influences from two sources. The first source was a collection of Middle Eastern rhythms known as the *Andalusian Muwashahat* rhythmic patterns. The second source was the concept of the rhythm tree as it is described in the computer program OpenMusic.

The *Muwashahat* rhythmic patterns originated in medieval Spain and Portugal.[15] These patterns are based on beat-cycles ranging from two to forty-eight beats. They incorporate combinations of two types of drum strikes – low sounding strike (*dum*) and high sounding strike (*tek*) – and silence. Figure 2.3 shows an example of a seven-beat rhythmic pattern. In *Normal Mode* I used patterns with up to twenty-four beats. I divided these patterns into four rhythmic groups. Group 1 included ten patterns with beat-cycles of two to five beats, Group 2 included twelve patterns with beat-cycles of five to ten beats, Group 3 included fourteen patterns with beat-cycles of ten to thirteen beats, and Group 4 included twelve patterns with beat-cycles of sixteen to twenty-four beats. These

15. Maqam World, "Arabic Rhythms," Maqam World, http://www.maqamworld.com/rhythms.html (accessed December 9, 2008).

groups were integrated with the concept of the rhythm tree in the compositional process of the electronics. The *Muwashahat* rhythmic patterns were also incorporated into the compositional process of the ensemble material, which will be described in Chapter 5.

Figure 2.3 A seven-beat rhythmic pattern



OpenMusic is a tool for computer-assisted composition, which was designed by Gerard Assayag and Carlos Agon in IRCAM, based on its predecessor PatchWork. PatchWork was designed by M. Laurson, J. Duthen, and C. Rueda, and received much attention from such European composers as Brian Ferneyhough, Claudy Malherbe, Tristan Murail and Kaija Saariaho.[16] The rhythm tree is a method for generating complex rhythmic structures. This method uses a programming procedure in which the user first defines the structure of a measure or a sequence of measures. The user can sub-divide beats in that measure by several layers of sub-divisions. The rhythmic values generated by these sub-divisions can be tied together or replaced by rests. The result is a complex rhythmic structure of nested rhythms. This method treats rhythm as a multi-layered structure in which the first layer is sub-divided in order to generate the second and the second layer is sub-divided to generate the third layer, etc. I adapted this method to be based on sub-divisions of durations instead of beats as will be described in Chapter 4.

---

16. Gerard Assayag et al., "Computer-Assisted Composition at IRCAM: From PatchWork to OpenMusic," *Computer Music Journal* 23, no. 3 (Autumn 1999): 59.

CHAPTER 3

SYNTHESIS PATCHERS

The electronics of *Normal Mode*, as already stated in the introduction, were composed in a twofold process in Max/MSP. I would like to begin this chapter with a short introduction to the principal elements of this software. This introduction will prepare the discussion of the composition's sound library in this chapter, as well as the discussion of the electronics part in Chapter 4.

A Short Introduction to Max/MSP

Max/MSP is an object-oriented programming environment. The user of this software combines virtual objects into a configuration called a patcher. Each object in this environment is programmed to perform a specific task. A patcher can be programmed to perform tasks such as the synthesizing, recording, editing and processing of sound. One patcher can be inserted within another patcher to form a subpatcher. Usually patchers in Max/MSP include several layers of subpatchers. They are also capable of accessing and utilizing external files such as sound files. This allows the user to program a complex chain of commands which is implemented instantaneously by the computer. When the programming is completed the patcher becomes a virtual performer.

The objects in Max/MSP have unique names, which will appear in this text in boldface letters. These objects can be connected by two types of virtual patch cords (Figure 3.1) – the Max patch cord (a) that transfers data and the MSP patch cord (b) that transfers digital signals. All the objects communicate by sending and/or receiving messages through the Max patch cords. Some objects in Max/MSP can send and/or receive digital signals through MSP patch cords. The messages include essential data or various types of commands. One of the most common commands in Max/MSP is the bang message that simply activates objects. An object can also receive an argument that defines parameters for its function.

Figure 3.2 displays examples of common Max/MSP objects. The first object (a) is contained within the object box, which can host most of the objects in Max/MSP. The box displays the name of the object and its argument. In this case the object is **cycle~**, which is a digital oscillator, and the argument is 1000, which specifies a frequency of 1000Hz for this oscillator. Some objects have a unique visual appearance. The **message box** (b) allows the user to type in a verbal command or a numerical value, which can be sent as a message to another object. The **button** (c) generates bang messages. The **number box** (d) receives, displays and sends integers. Similarly, the **float number box** (e) can accommodate floating point numbers. The **patcher** object (f) hosts a subpatcher and its argument will usually display the name of this subpatcher. The **inlet** (g) and **outlet** (h) objects are used within the **patcher** object to allow it to be connected to other objects.

Figure 3.1 Patch cords

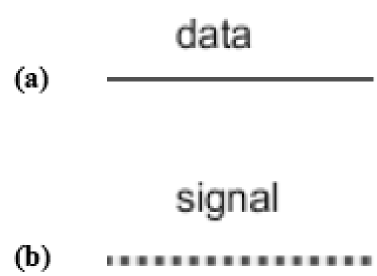
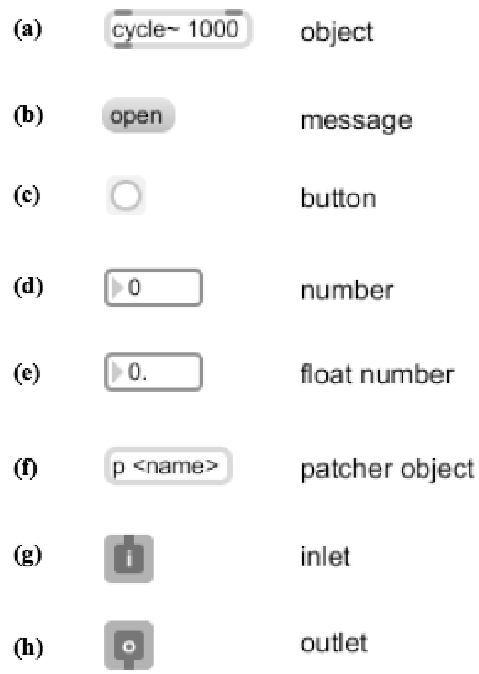


Figure 3.2 Common Max/MSP objects



Sound Synthesis in Max/MSP

The library of sounds of *Normal Mode* was generated by additive synthesis and frequency modulation in Max/MSP. The purpose of this synthesis process was the construction of complex sounds with microtonal frequencies from the segment frequency collections (SFCs). My goal was to explore new sonic possibilities by generating new sounds rather than to resynthesize preexisting recognizable timbres. This synthesis process was influenced by the incorporation of various levels of automation and random selection. As discussed in the previous chapter, the sounds, which were synthesized in this process, are inharmonic, yet not entirely gong and bell sounds. Modifications, which were made in the Max/MSP frequency modulation patchers, introduced distortion and various levels of noise into the mix of synthesized sounds. The additive synthesis patchers, as well as the frequency modulation patchers, synthesized dynamic spectra through unusual variations of amplitude envelopes and modulation indexes. Time-based transformations of pitch and timbre frequently occur within the sounds' short duration of two seconds (or 2000 milliseconds).

In the following sections I will describe the Max/MSP patchers that were constructed for this synthesis process. The discussion will focus on the incorporation of my compositional system in these patchers. It will define the methods by which frequencies were selected to be used in various types of synthesis patchers. It will also show how random selection restricted by the SFCs was incorporated within these methods, as well as in the control of other parameters of the synthesis patchers.

I began the process with the construction of a large number of additive and FM synthesis patchers. Each patcher was associated with one segment of the Turkish octave and derived frequencies from this segment's SFC. I defined a method of frequency selection for each patcher based on the type of synthesis, and the number of frequencies and harmonic sets included in the SFC. Hence, methods of frequency selection varied from patcher to patcher. I subsequently combined patchers associated with the same segment as subpatchers in a synthesis master patcher which mixed their signals into one complex sound. I had six synthesis master patchers. Each one of them generated the recorded sound files that constituted the basic sound-class of one segment.

The term sound-class refers to a group of sounds that share common characteristics. The defining characteristics can include durations, amplitude envelopes, registers and spectral envelopes. Sound-classes can also be defined by the procedure that generated the sounds or by a transformation or a set of transformations which was applied to the sounds. In *Normal Mode*, a basic sound-class is defined by the SFC that served as the source collection of frequencies for the synthesis process which generated the sounds in this class. With such classifications, sounds can be controlled by a compositional system, as can be pitch, rhythm and other musical elements.

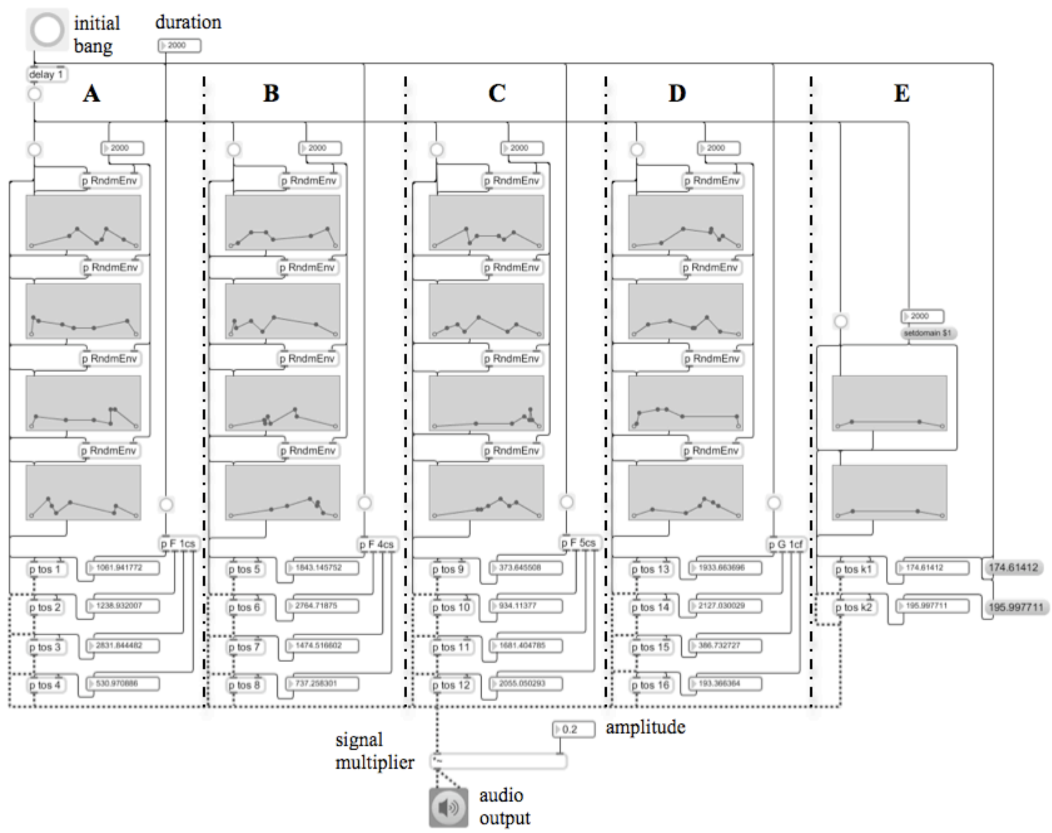
All the patchers, which are discussed in this chapter, are associated with the segment S2. These patchers served as models for the construction of patchers for the rest of the segments. Therefore, the patchers of the segment S2 exemplified the basic function of all patchers which were part of the synthesis process.

Additive Synthesis Patchers

Figure 3.3 provides an overview of an additive synthesis patcher which is related to the segment S2. This patcher combines multiple sine waves into a complex sound. The frequencies of these sine waves are derived from the SFC of this segment. A unique envelope controls the amplitude of each sine wave, yet none of these sine waves is defined as fundamental. In addition, this patcher incorporates random selection both in the selection of frequencies and the generation of envelopes.

For the purpose of illustration, the figure divides the patcher with dashed vertical lines into five columns. The columns A through D include similar objects. Each one of these columns generates four sine waves. The frequencies of these sine waves are derived from the harmonic set of one microtone of the segment S2. Thus, column A derives frequencies from the harmonic set of F 1 comma sharp, column B from the harmonic set of F 4 comma sharp, column C from the harmonic set of F 5 comma sharp, and column D from the harmonic set of G 1 comma flat. Column E looks slightly different. This column generates two sine waves. The frequencies of these sine waves are derived from the harmonic sets of the segment's natural dyad, F natural and G natural.

Figure 3.3 Additive synthesis patcher



All the envelopes generated by this patcher receive the same duration, which is entered in the **number box** at the upper-left part of the patcher. In Max/MSP, durations are specified by a whole number of milliseconds. A millisecond is a thousandth of a second. The **button** object near the duration **number box** generates the initial bang message that activates all the components of this patcher. All the signals produced by the patcher's components are transferred by the web of MSP patch cords in the lower part of the patcher to the signal multiplier. The latter mixes all the signals into one sound. The overall amplitude of this sound is determined by a floating point number entered in the **float number box**, which is connected to the right inlet of the signal multiplier.

A close view of column A is shown in Figure 3.4. The discussion of this column exemplifies the procedures that also take place in columns B, C, and D. The only difference between these columns lies in the harmonic set from which frequencies are selected. The **patcher** objects (a) in the lower part of the column contain subpatchers with **cycle~** objects such as the subpatcher shown in Figure 3.5. The **cycle~** object is a digital oscillator that generates a sine wave. The frequencies of each of these **cycle~** objects are determined by subpatcher inserted in the object **p<F_1cs>** (b) in Figure 3.6. The amplitudes of the column's **cycle~** objects are controlled by envelopes generated by the **p<RndmEnv>** objects (c) and the **function** objects (d).

The selection of frequency that takes place in the subpatcher of **p<F_1cs>** is random. This harmonic set subpatcher is shown in Figure 3.6. The bang message received in the inlet of the subpatcher activates the **uzi<4>** object (a). This object generates four instantaneous bang messages, which are sent to the **urn-jb<16>** object (b). The **urn-jb<16>** object randomly selects four integers within the range of 0 to 15,[1] which is specified in its argument.[2] These numbers are then matched with four items in the **umenu** object (c).[3] The sixteen items in this drop menu object are the sixteen frequencies of the harmonic set of F 1 comma sharp (e). The selected frequencies are distributed by the **bucket** object (d) through the four outlets of the subpatcher among the four **cycle~**

1. The unique feature of the **urn-jb<16>** object is that it does not repeat any of the integers until all the integers within the range have been selected.

2. The argument of some Max/MSP objects such as **random**, **urn**, and **urn-jb** specifies a range within a mod-*n*, in our case mod-16 or the range of integers between 0 and 15.

3. Like the **urn-jb<16>** object, the **umenu** object utilizes modular numbering of its items, therefore the first item is always item number 0.

objects of the column. For example, if the **urn-jb<16>** object selects the integers 5, 0, 7, and 12 the selected frequencies will be 1061.941772Hz, 176.99028Hz, 1415.922241Hz and 2300.873779Hz.

Figure 3.4 The microtone column A

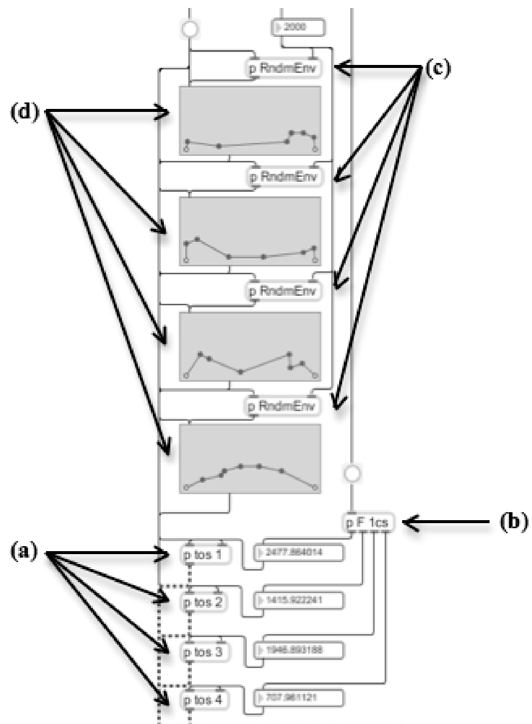


Figure 3.5 **cycle**~ object subpatcher

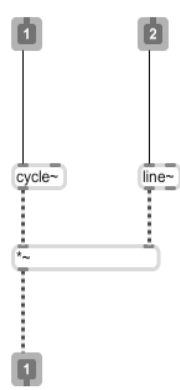


Figure 3.6 The harmonic set subpatcher **p<F_1cs>**

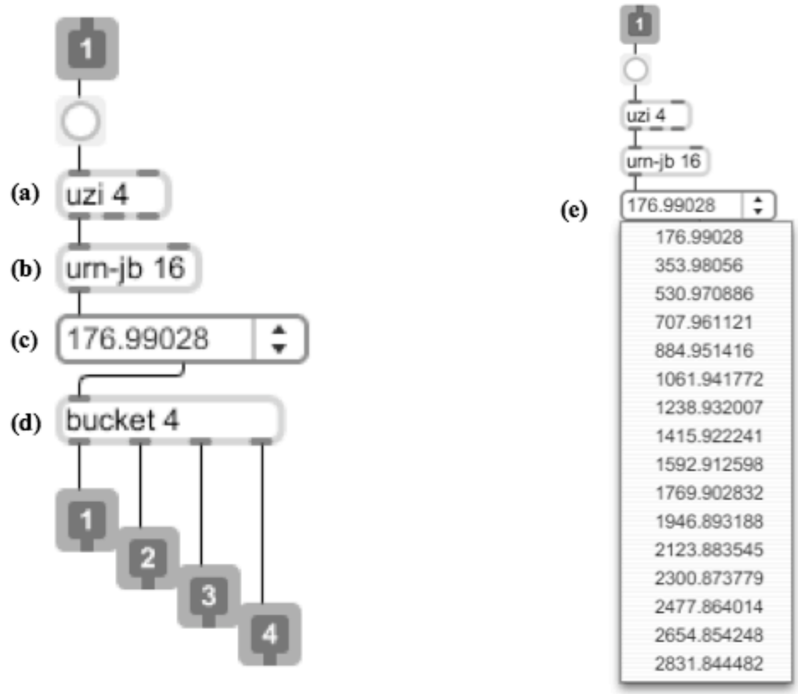
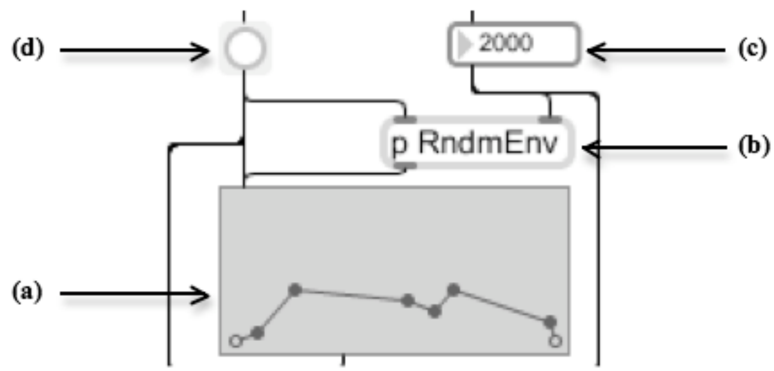


Figure 3.7 provides a close view of the objects in column A that generate the amplitude envelope of one **cycle~** object in this column. The **function** object (a) allows the user to draw and store a graph by placing breakpoints in the object itself. The object outputs a list of *x*/*y* values corresponding to the graph. This object is useful in controlling parameters that vary in time, such as amplitude within an envelope. The *x*-axis will then specify the time in milliseconds. The *y*-axis will specify the amplitude as floating point numbers between 0 and 1. The duration of the envelope is entered in the **number box** (c). In the case displayed in Figure 3.7, as well as in many other patchers of *Normal Mode*, the breakpoints are entered in the **function** object by the random envelope generator

subpatcher inserted in the **p<RndmEnv>** object (b). The **button** object (d) first activates this subpatcher and then the **function** object.

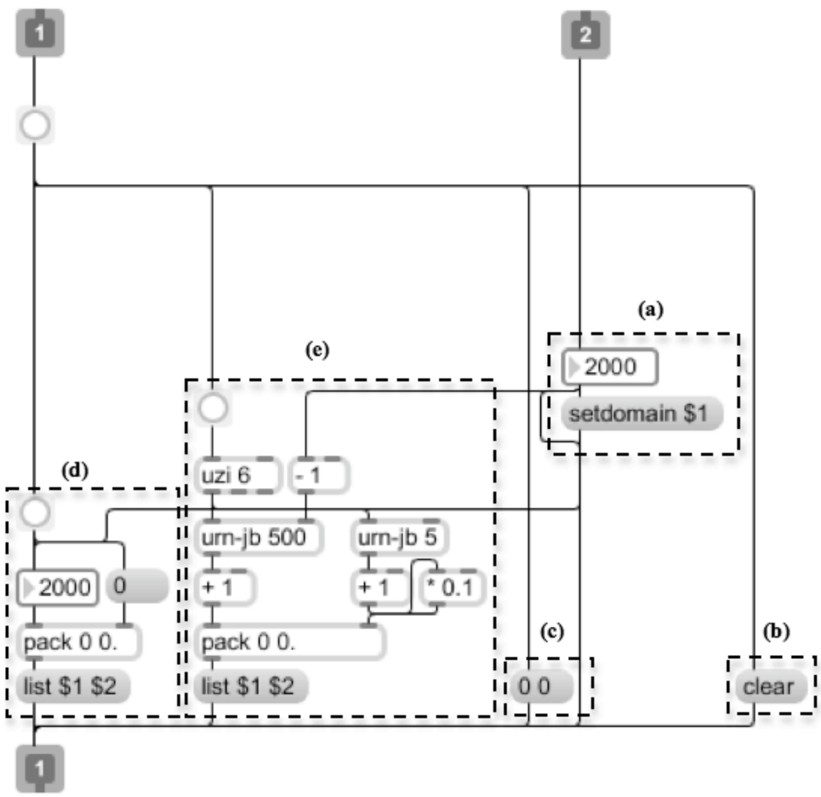
Figure 3.7 Amplitude envelope objects



The random envelope generator subpatcher randomly selects the coordinates for the breakpoints. It is displayed in Figure 3.8. This subpatcher generates eight *x*/*y* values for the eight breakpoints. While the *y* values are always in the range between 0 and 1, the *x* values can be within any duration in milliseconds that will be received through **inlet** 2 in the **number box** (a). Since the nature of the **function** object is to accumulate breakpoints, one must clear the object before generating a new graph. This is done by the **message box** "clear" (b). The **message box** "0 0" (c) sets the *x*/*y* values of the first breakpoint to 0 time 0 amplitude. Similarly, the objects in the dashed frame (d) set the *x*/*y* values of the last breakpoint. In this case, the *y* value is always 0 but the *x* value equals the duration received in the **number box** (b). The objects in the dashed frame (e) generate random *x*/*y* values for six breakpoints in between the first and last breakpoints.

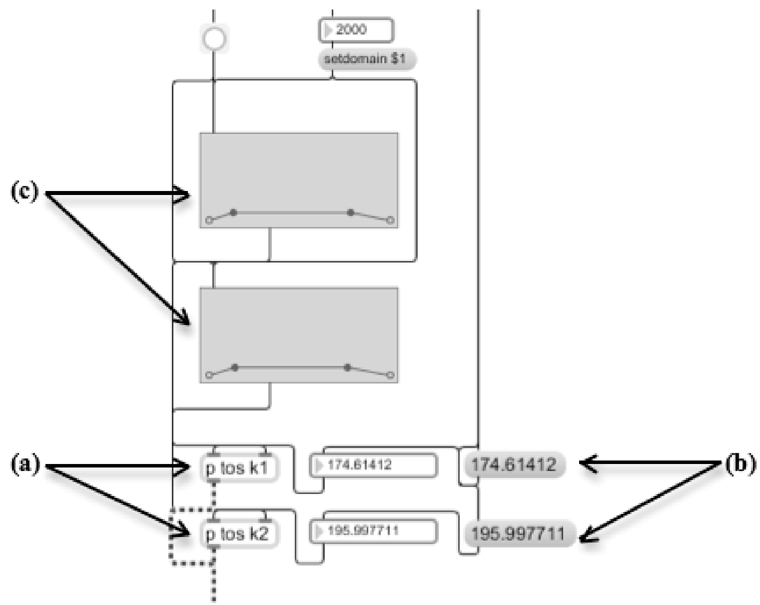
The *x* values of these six breakpoints are in the range between 1 and the duration minus 1, to prevent the overlapping with the first and last breakpoints and to insure that the envelope will always start with a fade in and end with a fade out.

Figure 3.8 Random envelope generator



Column E of the additive patcher is shown in Figure 3.9. The column's subpatchers (a) include two **cycle~** objects that generate two sine waves. The frequencies of these sine waves are fixed and are determined by the **message boxes** (b). These frequencies are each the lowest frequency (or the fundamental) in the harmonic set of one of the notes in the segment's natural dyad: F natural and G natural. The **function** objects (c) generate the envelopes of this column. These are fixed envelopes of very low amplitudes. Due to these envelopes, none of the sine waves of this column function as a fundamental frequency. Instead these sine waves create a drone that shades the rest of the patcher's components.

Figure 3.9 The natural dyad column E



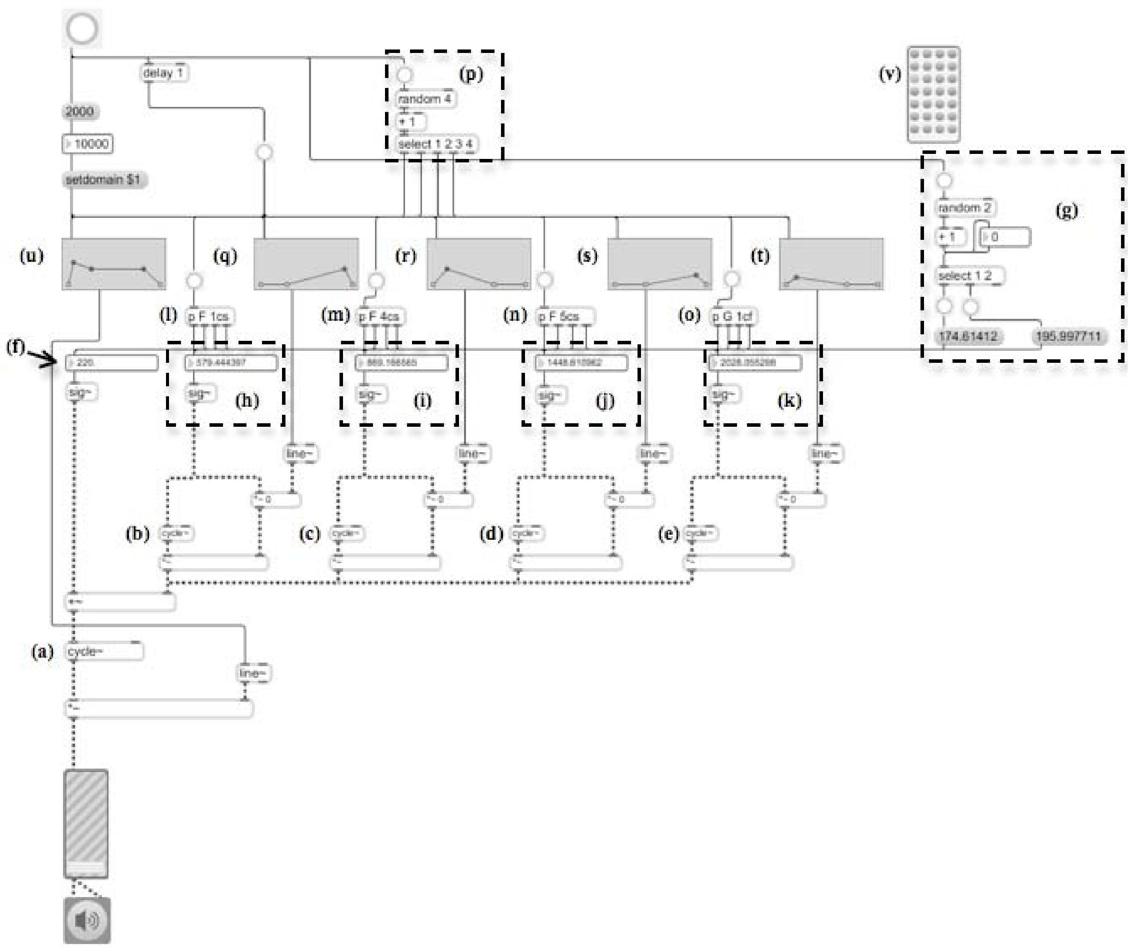
The method of frequency selection, which is demonstrated by the additive synthesis patcher in Figure 3.3, is adapted to the internal structure of the segment S2. This segment includes four microtones and a natural dyad. In each column of the columns A through D, four frequencies are randomly selected by a subpatcher, such as the one shown in Figure 3.6, out of the sixteen frequencies of the microtone's harmonic set associated with the column. Hence, sixteen frequencies are selected in columns A through D, and are joined by the two fixed frequencies of column E to a total of eighteen frequencies.

The additive synthesis patcher of the segment S2 served as the prototype for the construction of additive synthesis patchers for the rest of the segments. All of these patchers include eighteen **cycle~** objects, of which sixteen receive frequencies selected from the microtones' harmonic sets and two are associated with the natural dyad. Yet different methods of frequency selection from the microtones' harmonic sets were applied in these patchers, based on the number of microtones in each segment. The segments S1, S3, S4 and S6 include only three microtones. One method, which was applied in these segments, was the selection of two sets of eight frequencies from two harmonic sets out of the three microtones' harmonic sets of the segment. Another method combined different numbers of frequencies from all three harmonic sets that add up to sixteen. The segment S5 includes only one microtone. The method of frequency selection in this segment was order permutations of the frequencies in the microtone's harmonic set against fixed sets of envelopes.

Frequency Modulation Patchers

The FM synthesis patchers of *Normal Mode* incorporate two types of frequency modulation. The first type is parallel multiple-modulator frequency modulation. The second type is multiple-carrier frequency modulation. Figure 3.10 displays an example of a patcher using the first type of frequency modulation of the segment S2. It includes a single carrier with four parallel modulators that simultaneously modulate the carrier. In this patcher, the carrier's frequency is determined by random selection between the two lowest frequencies in the harmonic sets of the segment's natural dyad. The modulators' frequencies are selected from the harmonic sets of the segment’s microtones. As such, the modulators’ frequencies are inharmonic to the carrier’s frequency.

Figure 3.10 Parallel multiple-modulator frequency modulation patcher



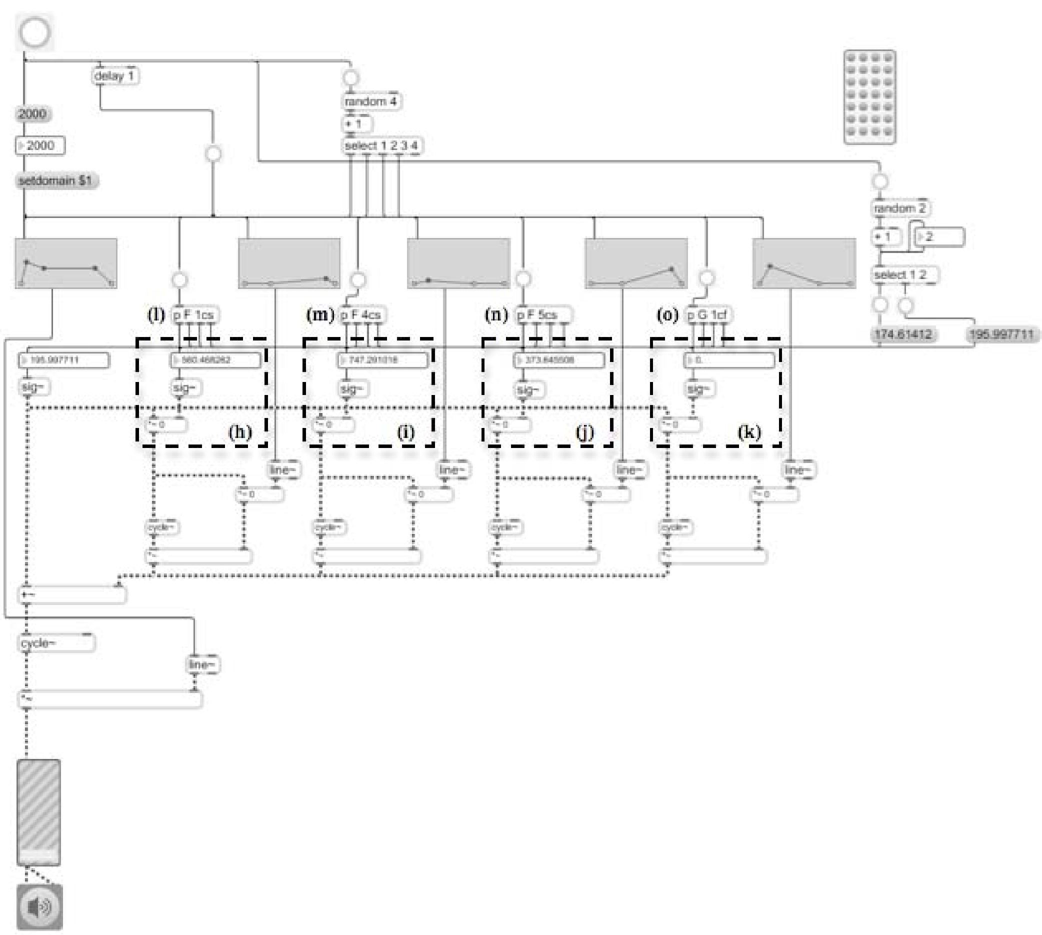
The **cycle~** object (a) generates the carrier sine wave. The **cycle~** objects (b) through (e) generate the modulator sine waves. The **number box** (f) displays the frequency of the carrier. This frequency is randomly selected by the group of objects marked by the dashed frame (g). The modulation frequencies are selected by the subpatchers inserted in the **patcher** objects (l) through (o). These are similar to the subpatcher shown in Figure 3.6, which is used in the additive patcher. As in the additive synthesis patcher each one of them is associated with the harmonic set of a microtone in the segment. In this case, the subpatcher selects four frequencies out of an eight-frequency subset of the harmonic set. With each activation, of the patcher the group of objects in the dashed frame (p) randomly selects and activates one of the subpatchers in the **patcher** objects (l) through (o). The selected subpatcher then randomly selects four frequencies and sends them to the modulator **cycle~** objects (b) through (d).

The patcher in Figure 3.10 incorporates a time-varied modulation index. The **function** objects (q) through (t) generate the envelopes that control the change in the modulation indexes over the duration of the sound. The **function** object (u) defines the overall amplitude envelope for the resulting sound. The **preset** object (v) stores and recalls different sets of coordinates for the graphs or envelopes of all **function** objects.

Figure 3.11 presents a patcher that exemplifies a modification of the patcher in Figure 3.10 within the same type of frequency modulation. These patchers differ in the objects that appear in the dashed frames (h) through (k). In the patcher in Figure 3.11, each one of the ***~** objects in these dashed frames multiplies the carrier frequency by a frequency selected by one of the subpatchers in the **patcher** objects (l) through (o). The product is a modulating frequency, which is far above the hearing range. The resulting sound is distorted and yet it is not entirely white noise. Specific modulation indexes generate various levels of noise and unique colors. This type of sound is very effective when it is mixed with the bell-type sounds, which are generated by the additive synthesis patchers.

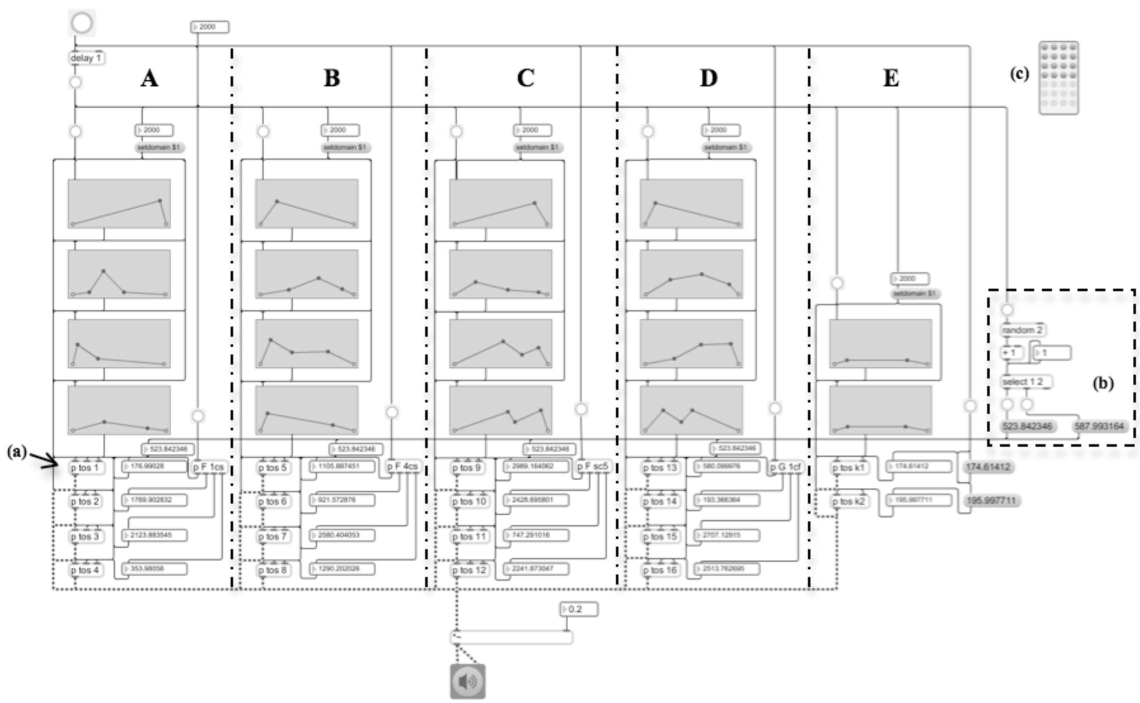
The second type of frequency modulation, the multiple-carrier frequency modulation, is demonstrated in Figure 3.12. This patcher combines multiple carriers of different frequencies in an additive meaner, each of which is modulated by one modulator. All the modulators receive the same frequency but different modulation index.

Figure 3.11 Modified parallel multiple-modulator frequency modulation patcher



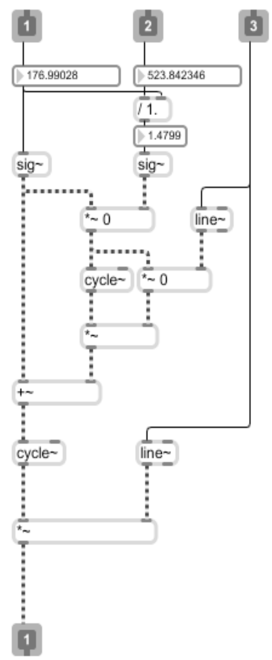
The patcher in Figure 3.12 is based on the prototype of the additive synthesis patcher shown in Figure 3.3. The modification takes place in the **cycle**~ object subpatchers which are inserted in the **patcher** objects, such as (a) in Figure 3.12, in the lower part of columns A through D. In the additive patcher, the **cycle**~ object subpatcher (Figure 3.5) includes one **cycle~** object that generates a pure (none-modulated) sine wave. In this patcher, the **cycle**~ object subpatcher (Figure 3.13) includes two **cycle~** objects: one generates the carrier sine wave and the other generates the modulating sine wave. At the same time column E, like in the additive patcher, generates a shading drone of pure sine waves.

Figure 3.12 Multiple-carrier frequency modulation patcher



The carriers' frequencies of the patcher in Figure 3.12 are randomly selected from the harmonic sets of the segment's microtones by similar subpatchers as in the additive patcher (Figure 3.6). The common modulation frequency is randomly selected from the harmonic sets of the natural dyad by the object in the dashed frame (b) in Figure 3.12. The **function** objects of columns A through D generate the envelopes that control both the modulation index and the amplitude of each pair of carrier/modulator **cycle~** objects. In this patcher these **function** objects are controlled by presets stored in the **preset** object (c).

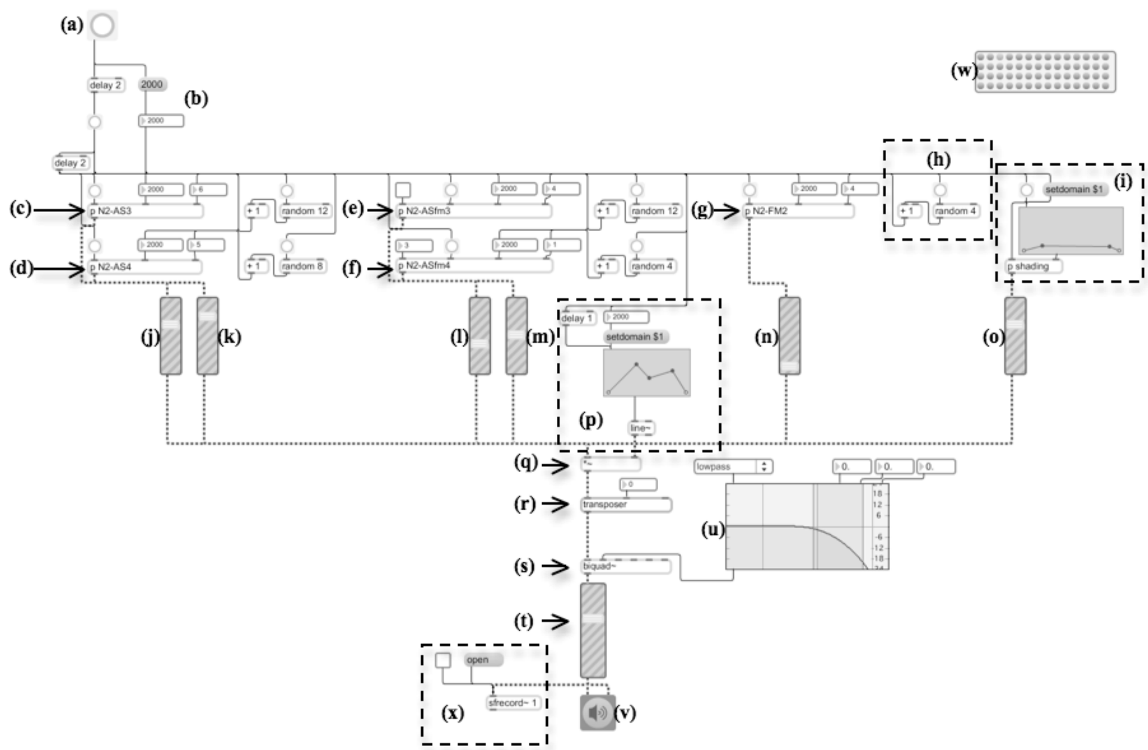
Figure 3.13 The **cycle**~ objects subpatcher of multiple-carrier frequency modulation



Synthesis Master Patcher

The discussion so far has described the prototypes of additive and FM synthesis patchers through concrete examples from the segment S2. Such patchers from each segment were combined as subpatchers in the segment's synthesis master patcher. Figure 3.14 shows the synthesis master patcher of the segment S2. This patcher mixes the signals generated by five synthesis patchers into one sound. The **button** object (a) generates the initial bang that activates the patcher. The **message box** and **number box** (b) define the duration of the sound. The **patcher** objects (c) and (d) include subpatchers of additive synthesis. The **patcher** objects (e) and (f) include subpatchers of multiple-carrier frequency modulation. The **patcher** object (g) includes a subpatcher of parallel multiple-modulator frequency modulation.

Figure 3.14 Synthesis master patcher



Some synthesis patchers, like the patchers in Figure 3.10, Figure 3.11 and Figure 3.12, utilize the **preset** object. The objects in the dashed frame (h) in Figure 3.14 randomly select presets from a **preset** object, which is included in the FM synthesis subpatcher (g). This **preset** object recalls coordinates for the amplitude and modulation index envelopes, which are generated by the **function** objects of this subpatcher. A similar selection mechanism as in the dashed frame (h) is attached to such synthesis subpatchers of the synthesis master patcher, which are not fully automated.

The objects in the dashed frame (i) add a unique component to the sounds, which are mixed in the synthesis master patcher. These objects play recorded samples of noise-type sounds in very low amplitudes. The role of these noise-type sounds is to create shadings to the sounds which are generated by the synthesis subpatchers. The term shading refers to a sound editing technique in which a primary sound object is juxtaposed with secondary sounds of very low amplitudes. These secondary sounds produce a sub-audible sound environment for the primary sound object. They are usually noise-type sounds that have no recognizable attack. Instead, they gradually fade in and fade out. As in visual arts, the shading technique enhances the depth and multi-dimensionality of the sound object.

The fader-like **gain~** objects, (j) through (o), control the levels of the output signals of the subpatchers. They function as the main tool in this patcher for mixing and fusing these signals into one complex sound. The **function** object in the dashed frame (p) applies an envelope to the mixed sound which is summed up in the *~ object (q). The **transposer** subpatcher[4] (r) applies an algorithmic transposition to the sound. The **biquad~** object (s) and the **filtergragh~** object (u) filter out some of the noise created in the patcher, mostly as a low-pass filter. The fader-like **gain~** object (t) controls the final level of the sound before the audio output object (v). The **sfrecord~** object in the dashed frame (x) records the mixed sounds to AIFF files. The **preset** object (w) stores and recalls data related to successful mixes of sounds achieved in this patcher. This data controls the

4. This patcher is taken from the Max/MSP software's library.

faders' levels (**gain~** objects (j) through (o)) as well as the general envelope (p) and the transposition levels applied by the algorithm **transposer** subpatcher.

The synthesis master patcher in Figure 3.14 is associated with the segment S2. The sound files, which were recorded by this patcher, constitute the basic sound-class of this segment. This sound-class includes sixty sound files of 2000 milliseconds each. These sixty sound files are divided into five sub-groups of twelve sound files. The sub-groups were defined by transposition levels that were applied to the sounds by the algorithm **transposer** subpatcher during the recording process. Table 3 specifies the transposition levels that defined each sub-group. Similar basic sound-classes and sub-groups were constituted for each segment using the segment's synthesis master patcher.

Algorithmic transposition or pitch shift, when applied to inharmonic sounds and sounds with significant noise component, may be perceived more as a change of timbre than a change of pitch. Not all transposition levels are very effective with such sounds. The transposition levels, which are specified in Table 3, have proved to be the most effective with the sounds generated by the synthesis master patcher.

Table 3. Sub-groups and corresponding transposition levels

Sub-group	Transposition levels
I	Untransposed sounds (T_0)
II	T_{12} T_1 T_2 T_5 T_6 T_7
III	T_{-12} T_{-1} T_{-2} T_{-5} T_{-6} T_{-7}
IV	T_{24} T_{13} T_{14} T_{17} T_{18} T_{19}
V	T_{-24} T_{-13} T_{-14} T_{-17} T_{-18} T_{-19}

The classification of basic sound-classes and their sub-groups was enhanced by additional sound classes. One sound-class was created in a synthesis master patcher that utilizes the low-band frequency collection. This collection contains the four lowest frequencies of the harmonic sets of all the notes in the Turkish octave. The resulting sound-class is made up of sounds that are easily transposed to low registers. Another classification of sounds is the classification into timbre-based sound-classes. This classification was applied to the entire library across the segments. The sound-classes were defined by a subjective evaluation of the amount of noise in the sounds. Sounds with 10 percent or less noise constitute the first sound-class. Sounds with 10-40 percent noise constitute the second sound-class. Sounds with more then 40 percent noise constitute the third sound-class. These classifications of the library of sounds into sound-classes and sub-groups have formed the foundation for the composition process of the performance patchers that will be discussed in the following chapter.

CHAPTER 4

PERFORMANCE PATCHERS

The performance of the electronics of *Normal Mode* is done with six Max/MSP patchers: one control patcher and five movement patchers. The control patcher simply enables the fast opening of the movement patchers. What I call a movement patcher is a patcher that realizes the electronics part of one movement in the composition.

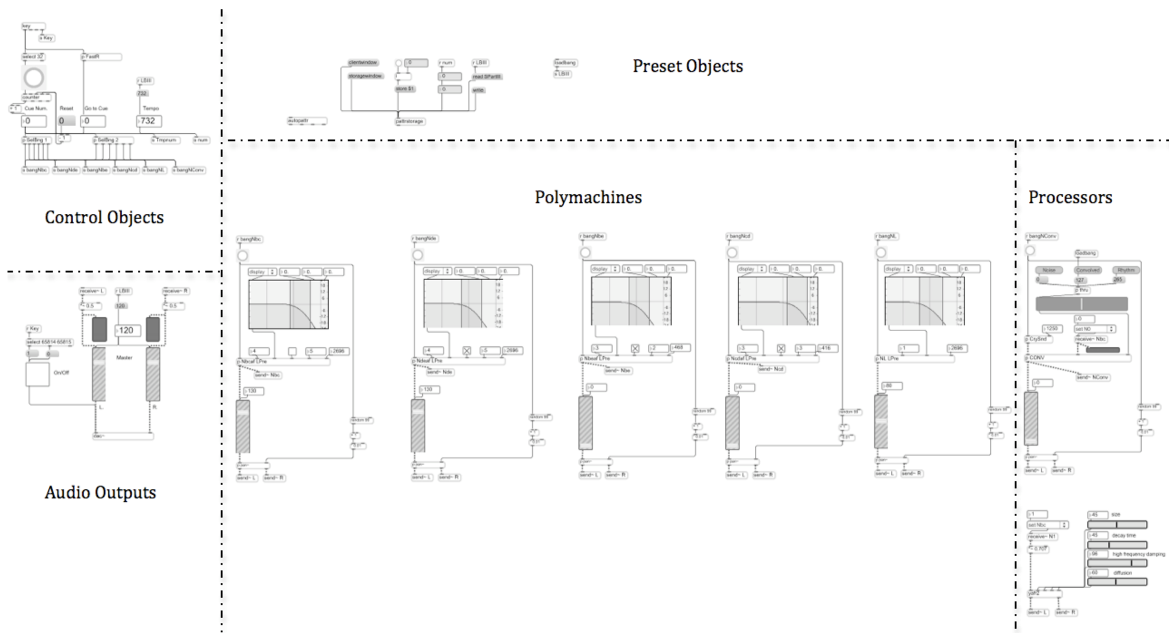
A movement patcher generates composite sound events using sound files from the composition's sound library. The components of the movement patcher can play, edit and process sound files in real time. This patcher also defines the temporal organization of sound files within the sound event. It is a multi-layer patcher that includes many subpatchers and links to external files. The programming of this patcher incorporates a complex chain of commands and data flow with both predetermined elements and random selection. The random selection is restricted to meaningful choices by classifications such as the sound-classes discussed at the end of the previous chapter. At the same time, random selection is utilized in regard to almost all aspects of the compositional process including temporal organization, selection of sounds, transpositions and envelope generation.

The sound events are the building blocks of the electronics part. They are marked in the score as cues indicating to the technician when to activate the movement patcher by striking the spacebar. The timing of a sound event within the movement is specified by the location of the corresponding cue in the score. The overall duration of a sound event is predetermined and marked in the score. The characteristics of a sound event are subject to both preprogramming and random selection in real time. Sound events may, therefore, vary with each performance.

The Movement Patcher and Its Components

Figure 4.1 provides an overview of the typical components of a movement patcher. The patcher in this figure is the movement patcher of Part III. The components of different movement patchers vary to some extent based on compositional choices unique to each movement, yet this example may serve as a prototype for understanding the function of these patchers.

Figure 4.1 Overview of a movement patcher



For illustration purposes, Figure 4.1 divides the patcher into groups of components based on their functionality. The groups of components are: control objects, preset objects, polymachines,[1] real-time sound processors and audio output objects. The control objects activate the rest of the patcher's components. The preset objects store and recall data essential for the function of the polymachines and the processors. The polymachines play, edit and process sound files from the library of sounds. The processors cross-synthesize and convolute the signals generated by selected polymachines, and they also apply reverb to some of the sounds. The audio output objects receive the signals from the polymachines and the processors and output them as 2-channel stereo.

Figure 4.2 describes the interaction between the components of the movement patcher that takes place with each sound event. This figure differentiates between three types of interaction: the flow of bang messages, the flow of data and the flow of signal. The initial bang message, which is generated by a strike of the spacebar, activates the control objects. These objects activate first the preset objects, which send data to the polymachine and the processors. The control objects then send bang messages that activate preselected polymachines and processors in predetermined time intervals. The signals generated by the polymachines are sent to the audio output objects and/or to the processors. The signals from the processors are also sent to the audio output objects.

It is important to point out that the flow of data and bang messages happens almost instantaneously, with the exception of bang messages, which are delayed for compositional purposes. Specifically the bang messages, which are sent from the control objects to selected polymachines and processors, are often delayed as part of the temporal organization of the sound event.

---

1. The term polymachine was coined by the author to describe these components of the movement patcher that utilize the Max/MSP object **poly~**.

The movement patcher acts also as a mixing board performing the following tasks: controlling the levels of individual signals generated by its components; panning these signals in the stereo field; sending signals internally from one component to the other; outputting the mixed signals as a 2-channel stereo image.

Figure 4.2 Interaction flow chart

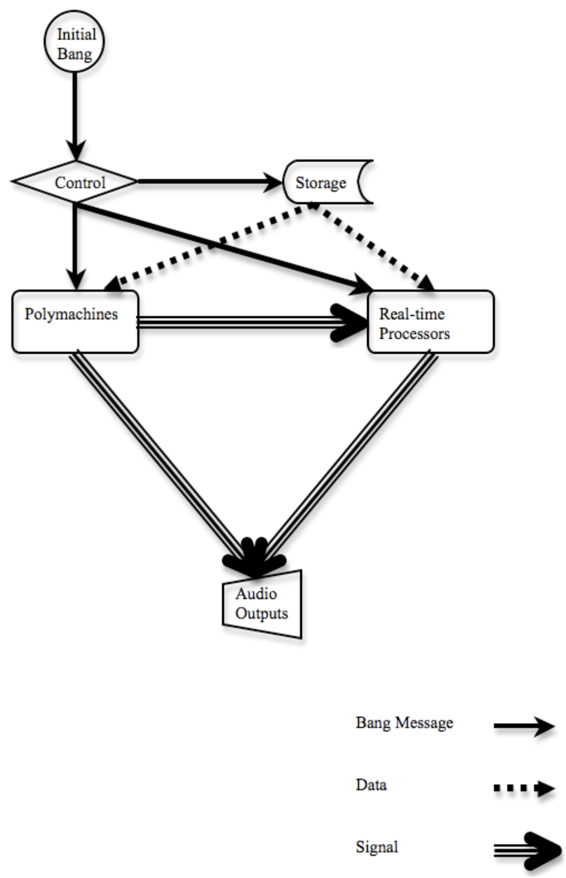


Figure 4.2 describes the surface level of interaction between the groups of the patcher's components. This level incorporates only predetermined decisions in regard to which polymachines and processors will be activated, in what time intervals and to what

duration. The predetermined decisions of this level also influence, to some extent, choices of amplitude, register and rhythm that take place in deeper levels of the patcher, mainly within the polymachines.

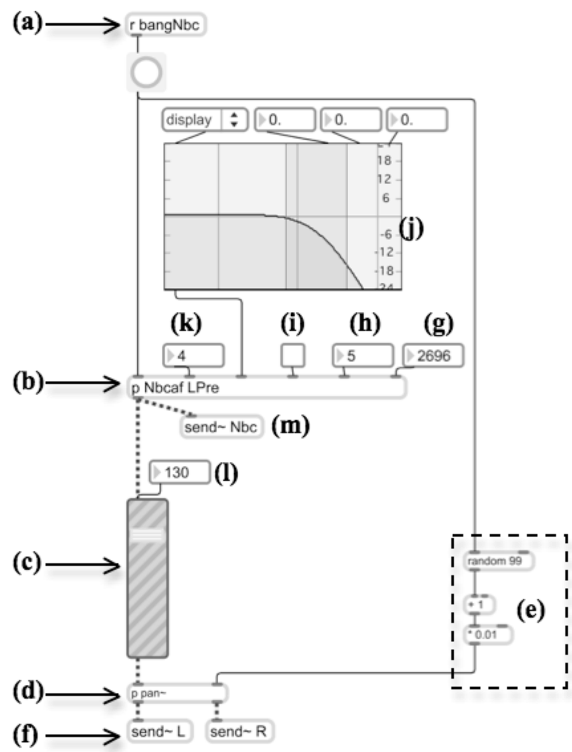
The Polymachine

The polymachines are the core of the movement patcher as they are the main sound generators of this patcher. They include several layers of subpatchers and links to external files and they perform multiple tasks, including selection and playback of sound files, editing of durations and envelopes, transposition, filtering and panning. The polymachines incorporate the **poly~** objects that facilitate efficient polyphony in Max/MSP. Each one of the polymachines is associated with a sound-class. It selects and plays sound files from this sound-class. The internal structure of the polymachine reflects the sub-groups of the sound-class.

Figure 4.3 provides a close view of the polymachine. The object **r<bangNbc>** (a) receives a bang message from the control objects. This bang message activates the polymachine subpatcher inserted in the object **p<Nbcaf_LPre>** (b). The **gain~** object (c) controls the level of the mono signal coming out of the polymachine subpatcher. The subpatcher inserted in the object **p<pan>** (d) pans this mono signal into a 2-channel stereo signal. The panning is controlled by random selection generated by the objects in the dashed frame (e). The panned 2-channel stereo signal is sent to the audio output objects by the **send~<L>** and **send~<R>** objects (f).

The **p<Nbcaf_LPre>** object (b) has several inlets that can transfer data to the polymachine subpatcher. This data controls the subpatcher's parameters. The objects, which are connected to these inlets, receive that date from the preset objects and transfer it to the subpatcher. The data includes predetermined sets of values. With each sound

Figure 4.3 Polymachine



event, or cue, a set of values unique to this cue is recalled by the preset objects and sent to the polymachine subpatcher. The **number box** (g) of Figure 4.3 receives a duration in milliseconds for the activation of the polymachine in a specific cue. The **number box** (h) and **the toggle** (i) receive values that influence the selection of register which takes place in the polymachine subpatcher. The **number box** (k) receives a value that influences the temporal organization of the files played by the polymachine. In addition the value received by the **number box** (l) determines the level of the polymachine subpatcher's output signal. The **filtergraph** object (j) controls a **biquad~** object in the polymachine subpatcher as in the synthesis master patcher (Figure 3.14(s)). The polymachine's mono signal can be sent, before it is panned, to the processors of the movement patcher by the **send~<Nbc>** object (m).

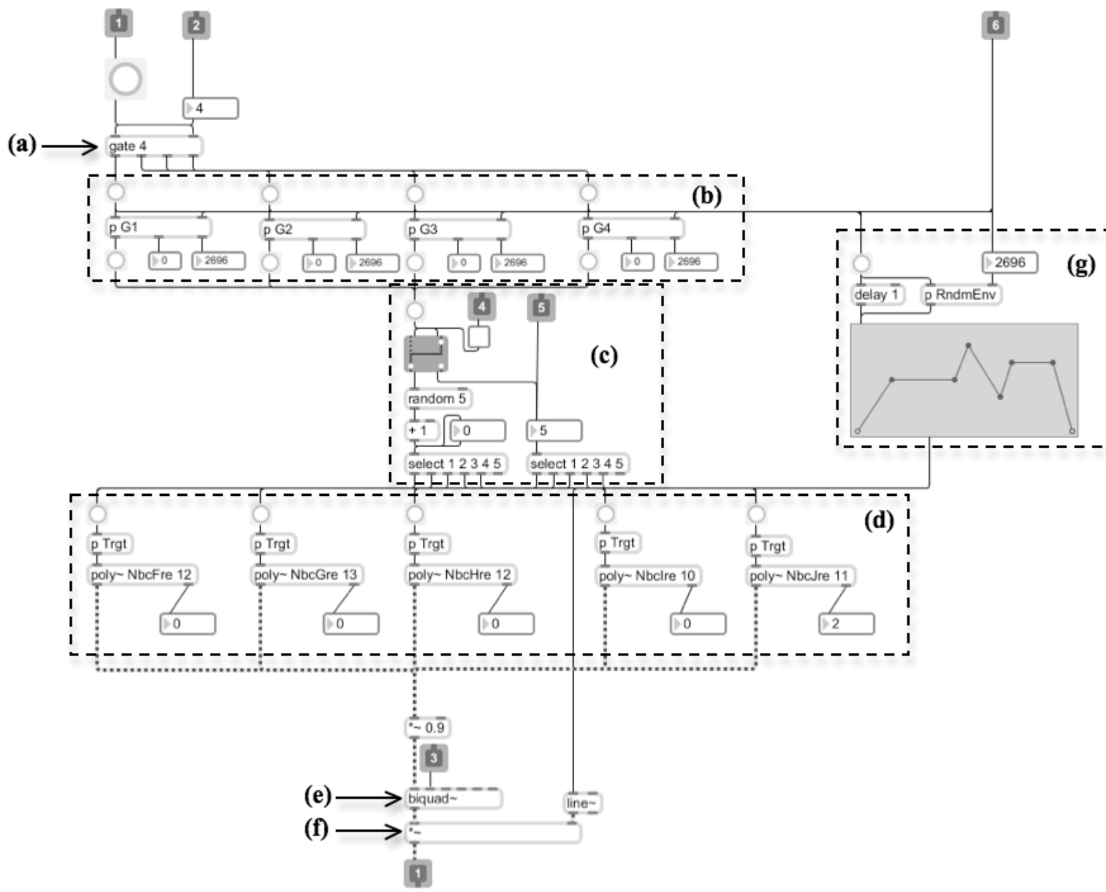
The Polymachine Subpatcher

The polymachine subpatcher is shown is Figure 4.4. The **gate** object (a) receives a bang message and sends it to one of the **patcher** objects **p<G1>**, **p<G2>**, **p<G3>**, **p<G4>**, in the dashed frame (b). The specific **patcher** object is selected by a number entered (as a preset) in the **number box** in Figure 4.3(k). These **patcher** objects host the rhythmic group subpatchers. The latter determines the temporal organization of the sound files which are played by the polymachine. They are programmed to generate sequences of bang messages. These sequences vary in the number of bang messages and the time intervals between these bangs.

The bang massages are distributed by the objects in the dashed frame (c) in Figure 4.4 among the **poly~** objects in the dashed frame (d) by two possible methods of distribution: when the **toggle** object, Figure 4.3(i), is "off," the bangs are sent to the **random<5>** object and the **select** object below it and randomly distributed among all the **poly~** objects; when the toggle object, Figure 4.3(i), is "on" and a number is entered in the **number box**, Figure 4.3(h), all the bangs are directed to one selected **poly~** object.

The **poly~** objects in the dashed frame (d) in Figure 4.4 facilitate the polyphonic playing of sound files from a sound-class in the library. Each one of them accesses one sub-group of files out of this sound-class. These sub-groups are defined by the transposition levels which were applied to the sounds when they were created in the synthesis master patcher. Additional transpositions are applied to the sounds by the **poly~** objects in real time. Hence, each **poly~** object defines a register. The methods of distribution conveyed by the objects in the dashed frame (c) in Figure 4.4 are, in fact, methods of register selection: all registers vs. one selected register.

Figure 4.4 Polymachine subpatcher



The **biquad~** object, Figure 4.4(e), along with the **filtergraph** object, Figure 4.3(j), act as a low-pass one band filter. The *~ object (f) in Figure 4.4 applies an envelope to the signal generated by the polymachine subpatcher. This envelope is randomly generated by the objects in the dashed frame (g) as in the synthesis patchers (Figure 3.7 and Figure 3.8).

The Rhythmic Group Subpatcher

The temporal organization of the sound files played by the polymachine, as previously mentioned, is defined by the rhythmic group subpatchers, which are inserted in the **patcher** objects **p<G1>**, **p<G2>**, **p<G3>** and **p<G4>** in the dashed frame (b) in Figure 4.4. This temporal organization is based on the *Muwashat* rhythmic patterns. Each rhythmic group subpatcher refers to a group of patterns of the *Muwashat* rhythms (see Chapter 2). The *Muwashat* rhythmic patterns vary in the number of attacks and the time intervals between attacks. Each rhythmic pattern was translated into a sequence of bang messages. The number of bang messages within a sequence equals the number of attacks included in the rhythmic pattern. The time intervals between bang messages is proportionate to the time intervals between attacks. While the duration of a sequence can be changed, the proportions of the time intervals will always be maintained.

The rhythmic group subpatcher is shown in Figure 4.5. It randomly selects sequences of bang messages. The subpatcher receives through **inlet** 2 the duration which is set for the activation of the polymachine. This duration is displayed in the **number box** (a). A bang message is received in **inlet** 1. It is directed by the **random** object (b) and the **select** object (c) to a sequence sub-patcher, which is inserted in one of the **patcher** objects, such as **p<seq_G2-1>** (d). The sequence subpatcher is shown in Figure 4.6. It receives the same duration through **inlet** 2 (a). The duration is multiplied by all fractions which are the arguments of the ***** objects, such as (c). The results are delay times in milliseconds, which are entered as arguments to the **delay** objects, such as (d). The bang message received in **inlet** 1 (b) is sent to all **delay** objects and delayed by time intervals which are proportionate to the rhythmic pattern.

Figure 4.5 Rhythmic group subpatcher

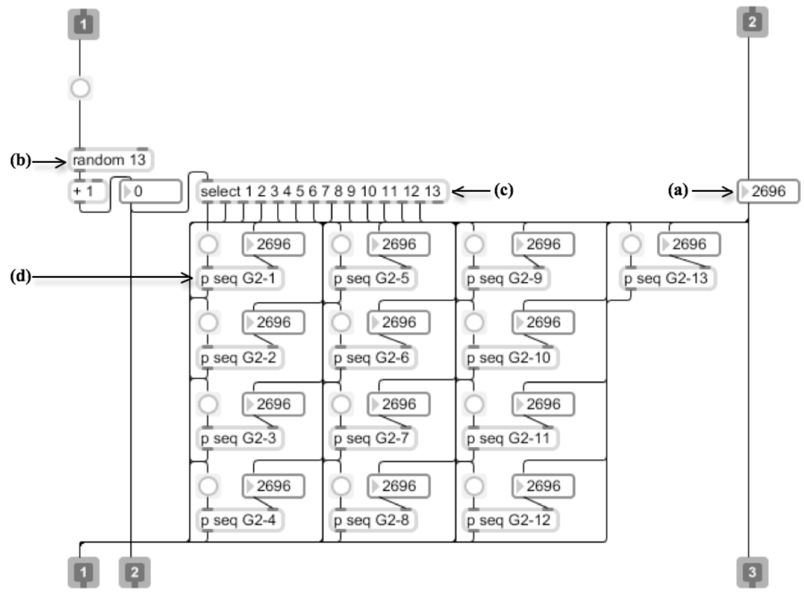
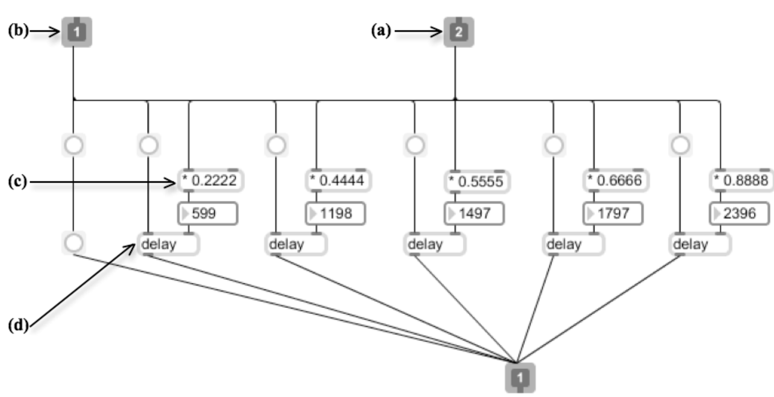


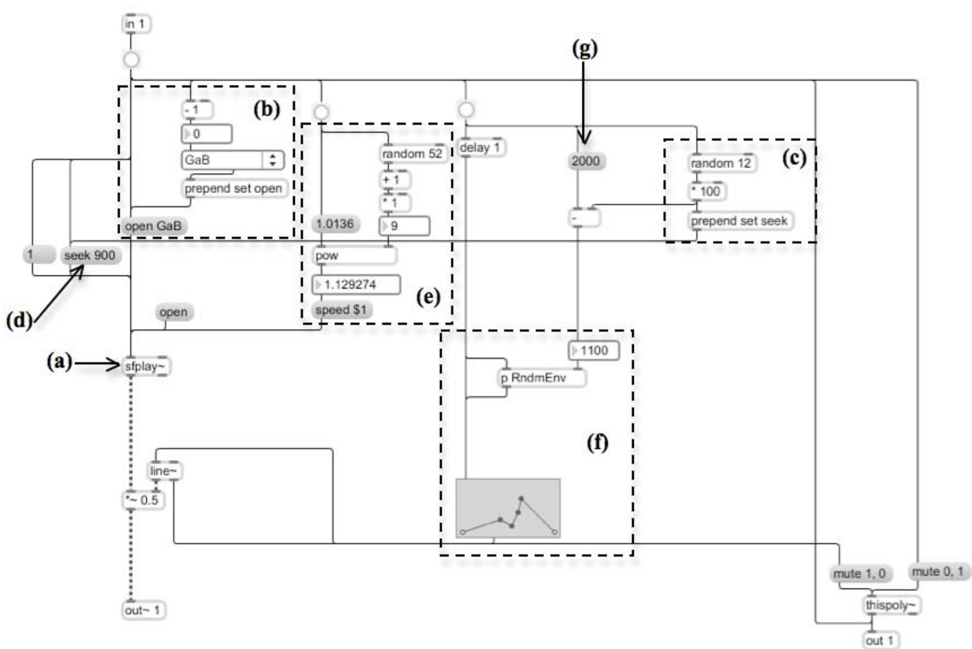
Figure 4.6 Sequence subpatcher



The poly~ Object and Its External Patcher

A **poly~** object, like the objects in the dashed frame (d) in Figure 4.4, access an external patcher. It creates copies of this patcher. The name of the external patcher and the number of copies are specified in the **poly~** object's argument. For example, the object **poly~<NbcGre_13>** accesses the file NbcGre.maxpat and creates thirteen copies of this patcher. Each copy generates a voice within a thirteen-voice polyphony. Figure 4.7 shows this external patcher.

Figure 4.7 The **poly~** object's external patcher



The core of this patcher is the **sfplay~** object (a). This object is capable of playing external sound files. It can play the sound file from and to different time points. It can also transpose the pitch by changing the playback speed of the sound file. This action on the sound file is often called speed transposition. The **sfplay~** object receives messages from other components of the external patcher that define parameters for these functions. The objects in the dashed frame (b) instruct the **sfplay~** object as to which sound file to play. Each copy of the external patcher within the **poly~** object can play a different file. The selection of files is random within the sound-class and the sub-group.

A time point, from which the **sfplay~** object will begin to play a sound file is defined by the objects in the dashed frame (c) of Figure 4.7 and the **message box** (d). This time point is randomly selected within a predetermined time interval. The **message box** (g) specifies a maximum possible duration for a single sound. In order to differentiate between legato and staccato sequences of sounds, two maximum possible durations were defined: 2000 milliseconds and 500 milliseconds. If the maximum possible duration is 2000 milliseconds, the time interval for the selection of the beginning time point is 0 to 1200. If the maximum possible duration is 500 milliseconds, the time interval for the selection of the beginning time-point is 0 to 300. The final duration of a specific sound will be the maximum possible duration minus the time interval between 0 and the new beginning time point. For example, if the maximum possible duration is 2000 milliseconds and the beginning point is set to 700, the duration of the sound will be 1300 milliseconds. In general, the durations of legato sounds range between 800 milliseconds and 2000 milliseconds and the durations of staccato sounds range between 200 milliseconds and 500 milliseconds.

An envelope is applied to the sound throughout the new duration. The objects in the dashed frame (f) in Figure 4.7 randomly generate this envelope. These objects are similar to objects shown in Figures 3.7 and 3.8.

The objects in the dashed frame (e) in Figure 4.7 define a speed transposition level based on division of the octave into 52 commas. The speed transposition is generated by sending a ratio to the inlet of the **sfplay~** object. If, for example, the ratio is 2, the **sfplay~** object will play the sound file twice as fast and the sound will be transposed an octave above. If the ratio is 0.5, the sound will be transposed an octave below. The ratio of the comma-based speed transposition is determined when the constant 1.0136 in **message box** is raised to the *n*-th power by the **pow** object. The *n* in this specific patcher is a positive number between 1 and 52 which is randomly selected by the **random** object. Such a ratio will result in a speed transposition equivalent to transposition by a specific number of commas within an octave above. For example, if the constant 1.0136 is raised to the power of 30, the ratio is 1.499 which is a transposition by a (Pythagorean) fifth, i.e. a transposition by an interval of 30 commas (30 x 23.4 = 702) is also a transposition by a (Pythagorean) fifth.

Comma-based speed transpositions are used in the **poly~** objects in registers parallel to the registers of the sound-class sub-groups. As mentioned in Chapter 3, the sub-groups are defined by the transposition levels applied to the sounds in the synthesis master patcher. The transposition levels of each sub-group are within a specific register. The parallel registers of the speed transpositions are defined by the range of numbers randomly selected for *n*. The first **poly~** object from the left in the dashed frame (d) in Figure 4.4 does not include transposition. The second **poly~** object transposes within the first octave above by *n* = 1…52. The third **poly~** object transposes within the first octave below by *n* = -1…-52. The fourth **poly~** object transposes within the second octave above by $n = 53 \ldots 104$. The fifth **poly~** object transposes within the second octave below by $n = -53 \ldots -104$.

Composing a Sound Event

The discussion so far has described the movement patcher to its deepest layers and core components. The following paragraphs will summarize the process by which sound events are composed in real time within the movement patcher. This process begins with the selection and modification of a single sound file by the **sfplay~** object and related objects within one copy of the **poly~** object's external patcher (Figure 4.7). At this stage the sound file is trimmed to a new duration, it receives a new envelope and it is transposed by a speed transposition. The modified sound file is incorporated in a polyphonic succession of similarly modified sound files, which are played by the **poly~** objects of the polymachine subpatcher (Figure 4.4). This polyphonic succession is characterized by a rhythmic structure, which is determined by a sequence of bang messages generated by one of the rhythmic groups subpatchers (Figure 4.4(b) and Figure 4.5). The registers of this polyphonic succession are defined by the methods of distribution of the bang messages among the **poly~** objects (Figure 4.4(c)). The polymachine subpatcher applies an amplitude envelope to this succession. The duration of this envelope is equal to the duration that was determined for the activation of the polymachine within a specific sound event or a cue.

The polyphonic succession of sound files generated by one polymachine is joined by successions generated by other polymachines. At the same time, sound from the polymachines may be sent to the real-time sound processors. The mixed sounds from the polymachines and the processors constitute the sound event.

All the activity that takes place within the polymachine is subject to random selection and the resulting sound is to a large extent undetermined. At the same time, this activity is restricted by predetermined parameters that specify its duration and influence the choices of register and rhythmical structure. Such predetermined parameters, which are stored and recalled by the preset objects, also define the mix between signals generated by different polymachines and processors as they control the fader levels of each component. Another set of predetermined choices is reflected in the programming of the control objects. This programming determines which polymachine and processors will be activated in each sound event. It also determines the time points within the sound event's duration from which the selected components will start to work. Thus, the programming of the control objects is an essential component in the temporal organization of the electronics part.

Rhythm Tree

The temporal organization of the electronics of *Normal Mode* in each movement is structured as a rhythm tree of three layers. The first layer determines the duration of each sound event and its timing within the movement. The second layer determines the timing and duration of the activations of the patcher components – polymachines and processors. The durations in the second layer are divided in the third layer by randomly selected sequences (or rhythmical patterns) of randomly selected sounds of randomly selected durations. All activations of sound events take place on a downbeat. The activations of polymachines and processors in the second layer are often delayed in relation to the downbeat. The durations in the second layer may be separated from each other or may overlap. Similarly the succession of the sounds in the third layer may be staccato or legato. The characteristics of a sound event may, therefore, vary anywhere between a percussive succession of short sounds and polyphonic simultaneity.

The first two layers of the temporal organization are predetermined and these layers are notated in the score. The third layer of the temporal organization is subject to random selection and, therefore, cannot be notated. The notation of the electronics is functional and it is designed to serve the technician in the activation of the patchers. (See Technician Notes and score in Chapter 6). The lower staff indicates to the technician when to activate the patchers with a strike of the spacebar. It includes numbered cues notated as quarter notes on the downbeats. The numbering in the score corresponds to the numbering of cues in the patcher. The upper staff notates the timing and duration of the activations of the patcher's components in relation to the downbeat of the cue. This staff allows the technician to anticipate the events that will follow the strike of the spacebar. The bracket between the staves marks the overall duration of the sound event.

CHAPTER 5

ENSEMBLE AND ELECTRONICS

The compositional system described in Chapter 2 establishes the timbral characteristics and the temporal organization of the electronics part. This system also maps the basic sound-classes of electronic sounds to a pitch organization which is based on the Turkish tonal system. The pitch material for the ensemble was composed in the twelve-note equal-tempered Western tuning. The common element between those two different worlds is found in the collection of natural dyads that frame the segments. Although this collection of dyads is considered in this composition as equal-tempered, it is important to acknowledge that the whole-tone dyads in this collection (D-E, F-G, G-A, A-B and C-D) are not all equal. Each one of these dyads frames a unique microtonal division. Subjecting these dyads to mod-12 transposition would cause them to lose their identity. Therefore only a specific set of transformations could be applied to these dyads.

To establish relations between the seven pitch-classes that constitute the natural dyads and the rest of the Western aggregate, I have chosen to use M5, M7 and M11 transformations. I consider these types of transformations more strongly connected to the centricity of the cycle of fifths both in Turkish music and Western music. Table 4 shows that the natural dyads can be mapped by M5, M7 and M11 transformations to include all pitch-classes with the exception of pc 6. The latter only maps to itself through these transformations and since it is not part of the original collection of dyads it is not included in this mapping. Four of the natural dyads - of the segments S1, S2, S5, and S6 - are each mapped with a specific selection of dyads. The natural dyads of the segments S3 and S4 are mapped to the same selection of dyads. Nevertheless, this mapping preserves some of the unique identity of the natural dyads within the Western aggregate.

Table 4. Mapping of the natural dyads by M transformation

Segment	Natural Dyad	M5	M7	M11
S1	{2 4}	{t 8}	{2 4}	{t 8}
S2	{5 7}	{1 e}	{e 1}	{7 5}
S3	{7 9}	{e 9}	{1 3}	{5 3}
S4	{9 e}	{9 7}	{3 5}	{3 1}
S5	{e 0}	{7 0}	{5 0}	{1 0}
S6	{0 2}	{0 t}	{0 2}	{0 t}

Several serial operations helped create the pitch organization of the ensemble material. I constructed eight different twelve-tone rows (Figure 5.1) from subsets of the Dorian scale, including the tetrachords (0235) and (0135) and the hexachord (023579). Those rows were manipulated by the multiplication system, as was used by Pierre Boulez in *Le marteau sans maitre* (1953-57),[1] to form forty domains of unordered sets. Furthermore, I applied M5, M7 and M11 transformations to the unordered domain sets for the purpose of connecting them with the segments. For example, the unordered set {8t0235} is a domain set in the first domain of the row B1. It consists of only one natural dyad {02} of the segment S6. However, the dyads {t8}, {35} and {50}, which are subsets of this domain set, can be mapped through the transformations presented in Table 4 to the segments S1, S3, S4, and S5. Similarly, M transformations of this set will include the natural dyads of these segments. M5 of this set is the set {t01234}, which includes the

1. Based on Stephen Heinemann, "Pitch-Class Sets Multiplication in Theory and Practice," *Music Theory Spectrum* 20, no. 1 (Spring 1998): 72-96, and Lev Kolbyakov, *Pierre Boulez: A World of Harmony* (New York: Harwood Academic Publishers, 1990).

natural dyad {24} of S1. M7 of this set is the set {89te02}, which includes the natural dyad {9e} of S4 and the natural dyad {e0} of S5. M11 of this set is the set {9t0247} which includes the natural dyad {97} of S3 and the natural dyad {24} of S1.

Figure 5.1 Twelve-tone rows

- A1 – 75243960e1t8
- A2 – 24756039t8e1
- B1 – 2475e9680t13
- B2 – 75240t139e86
- C1 – 24579e310t86
- C2 – 579e02t86431
- D1 – 459786e132t0
- D2 – t9576831e042

The domains were utilized in the compositional process of the ensemble material. In the first stage, two methods were defined by which domain sets were selected for compositional process. The first method associated a group of domains, which was generated from one row, with a movement or a section within a movement for the purpose of composing this section only with sets from these domains. The second method reorganized domain sets from all forty domains in classes of domain sets that have similar characteristics. The selection of domain sets for composing some sections of the ensemble material was based on these classifications.

A variety of characteristics defined the classes within the second method, and each class was divided into a number of collections. One class included collections of domain sets that had the same number of pitch-classes, for example, a collection of all the trichords, all the tetrachords, etc. Another class included collections of domain sets with specific pitch-class content, for example, a collection of all the domain sets that contain the dyad {24} (or the natural dyad of the segment S1). A third class included collections of domain sets that contain segments of the Western chromatic scale. Such domain sets were sorted out in collections based on their total number of pitch-classes, the number of these pitch-classes that constitute a Western chromatic segment, the pitch space of this segment and what other pitch-classes are included. For example, the hexachord {8te012} is a domain set in the third domain of the row C1. Five of its pitch-classes constitute a chromatic segment from pc t to pc 2. The sixth pitch-class is pc 8. This hexachord was a member of a collection of hexachords with similar chromatic segment of five pitch-classes and an additional pitch-class.

After the methods of selection were defined, domain sets were composed out in short phrases. The rhythmic structures of these phrases were derived from transformations of the *Muwashat* rhythmic patterns. The basic set of transformations included rotations of the prime, retrograde, inversion and retrograde-inversion forms of the pattern. Figure 5.2 presents the prime (a) retrograde (b), inversion (c) and retrograde-inversion (d) forms of the seven-beat pattern, which was discussed in Chapter 2 (Figure 2.3). Figure 5.3 presents the seven rotations of the prime form. Before such rhythms were combined in the rhythmic structure of a short phrase, an additional set of transformations were applied that included augmentation, diminution and metric modulation. All phrases were written in their final form in 4/4 meter. Each short phrase was then multiplied by the pitch transformations M5 M7 and M11 to generate three more phrases with the same rhythm. As a preparation for composing the ensemble material, I had several collections of such short phrases.

Figure 5.2 Transformations of a seven-beat pattern

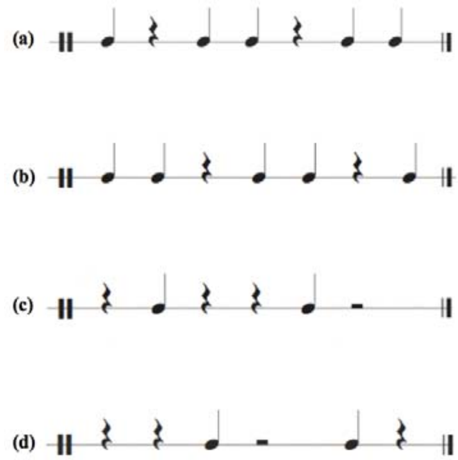
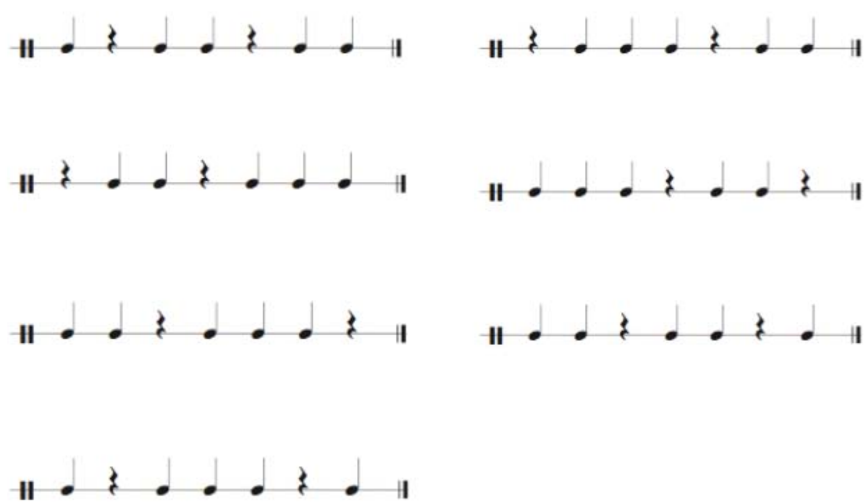


Figure 5.3 Rotations of the prime form



In the next phase of the compositional process I cut small fragments from the short phrases and pasted them together with other fragments to generate more complex gestures. These gestures were orchestrated and combined with other gestures in the score.

At this stage, motivic relations, as well as formal aspects of the piece, began to emerge. As part of the development of these motivic relations and the articulation of the form, the gestures themselves received more transformations. One type of pitch-class transformations, which is prominent in Part III and Part V, is the mapping from the numbers mod-12 to the numbers mod-*n*, for $n = 3$, 4 and 6. As in the case of the M transformations, these transformations are based on mapping between specific pitch-classes and not on manipulations of interval structures. In general, the compositional process of the ensemble material, beyond the stage of forming the domains, was avoiding the basic mod-12 operations of transposition, inversion, retrograde and retrograde-inversion. This restriction was the result of the influence of Turkish music on this composition.

As the ensemble material was shaping up, I began to compose the electronics alongside it. I analyzed the pitch content of gestures from the ensemble material based on the mapping presented in Table 4. This analysis established relationships between gestures and sound-classes for corresponding sound events. For example, the first two measures of the piece are each dominated by one dyad. (See Chapter 6 for the score). In measure one it is the dyad {35}. In measure two it is {t8}. Measure three includes the dyads {24}, {8t}, {79}, {35} and {e1}. Therefore, these three measures were associated with the segments S1, S3 and S4. In measure four the vibraphone, marimba, viola and cello emphasize a voice leading from the dyad {48} to the dyad {39}. While these dyads are not included in the mapping of Table 4, these specific pitch-classes imply only the segments S1, S3 and S4. Similarly the pitch content of measures five and six emphasize the pitch-classes 2, 5, 7, e and 0 that imply the segments S2, S5 and S6.

The rhythmic structures and locations of gestures influenced the first two predetermined layers of the temporal organization of the sound events. For example, in measures five and six of the first movement, the sound events of cues 7 and 8 are set in relation to the woodwinds and the strings. The first-layer duration of cue 7 corresponds to the activity in the woodwinds from beat three of measure five to beat two of measure six. This duration is divided by four second-layer durations. The first one begins with the clarinet on beat three of measure five. The second second-layer duration begins with the flute and tenor saxophone in the end of measure five. Another duration is interjected within the flute and saxophone gesture, and the last duration parallels the growl at the end of this gesture. Similarly, cue 8 interjects its second-layer duration within the pizzicato strings gesture of measure six.

CHAPTER 6

NORMAL MODE

FOR CHAMBER ENSEMBLE AND ELECTRONICS

Program Notes

A mechanical system is said to oscillate in a normal mode when all of its particles move simultaneously with the same frequency. Hence, a normal mode is a coordinated motion of particles. The composition *Normal Mode* derives its frequencies of oscillation from two different musical worlds. The electronics of the piece were created through a process of digital sound synthesis with reference to the microtonal octave division of Turkish music. The ensemble material was conceived within a Western-influenced serial pitch organization. These two distinct forces never cancel each other. Instead they create much tension and motion and sometimes, philosophically speaking, also find a normal mode.

Instrumentation

Flute

Clarinet in Bb

Tenor Saxophone

Horn in F

Trombone

Percussion I: vibraphone, kick bass drum, low tom, high tom, snare drum, timbales, bongos, wood block, hi-hat, suspended cymbals (ride, crash).

Percussion II: marimba, kick bass drum, low tom, high tom, snare drum, timbales, bongos, woodblock, guiro, hi-hat, suspended cymbals (ride, crash).

Piano

Violin

Viola

Cello

Computer (Max/MSP)

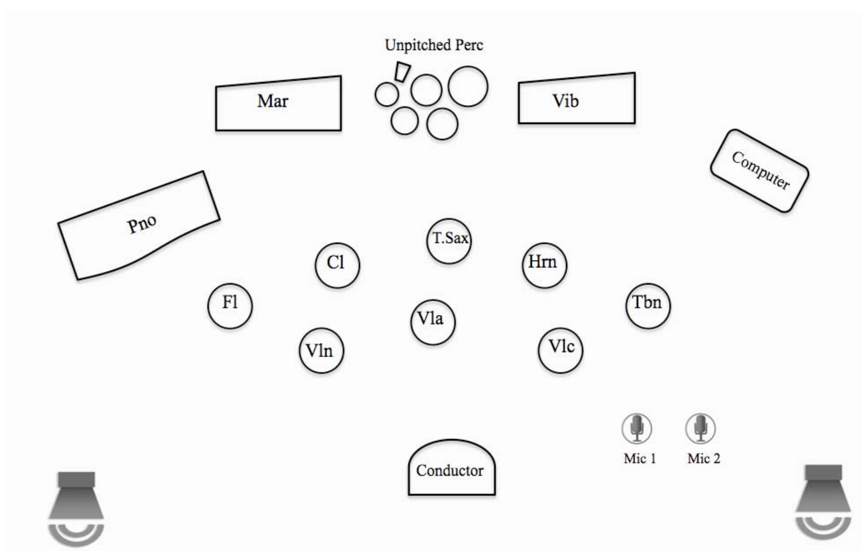
Performance Notes

The score is transposed.

The technician must operate the required equipment from a location on the stage that allows him/her to follow the conductor.

Two microphones have to be placed in the front part of the stage. They should be clearly marked as microphone 1 and microphone 2 in a way that is visible to the musicians on stage. These microphones are used in Part II and Part IV which include interaction between the electronics and soloists of the ensemble. In Part II, the interaction is with the clarinet that plays to microphone 1. In Part IV, the interaction is with the tenor saxophone that plays to microphone 1 and the trombone that plays to microphone 2. Two additional music stands should be placed near the microphones. The ensemble's seating should allow room for these transitions on stage (see Figure 6.1).

Figure 6.1 Seating diagram



Glissandi are continuous over the rests in parenthesis. Fingered glissandi should be performed as chromatically as possible. Trills are always a half step above the note. Special notation is explained in the notation key. The computer notation is explained in the technician notes.

Technician Notes

The equipment required for the performance of the electronics of *Normal Mode* includes: a computer with the software Max/MSP 5 or a later version; an audio interface device compatible with Max/MSP with two microphone input channels and two output channels; a small mixer with at least two input channels and Left/Right output channels; two microphones; and two speakers with amplification.

The technician must operate the computer, the audio interface device and the small mixer from a location on the stage that allows him/her to follow the conductor. The microphones have to be placed in the front part of the stage. They should be clearly marked as microphone 1 and microphone 2 by a marking that is visible to the musicians on stage. Microphone 1 has to be connected to input channel 1 of the audio interface device and microphone 2 to input channel 2. The two output channels of the audio interface device have to be connected to two input channels in the mixing board. The Left/Right output channels of the mixer have to be connected to the power amplification system. Figure 6.2 provides a technical diagram of this setup. To adjust the microphones' input levels, use the gain controls of the audio interface device. To adjust the system's output levels, use the mixer's faders.

Max/MSP includes a DSP status window, which allows the user to set up the communication between the software and hardware, such as the audio interface device. For this piece, set up the DSP in the following way:

1. In the options menu, select DSP. A window like the one shown in Figure 6.3 will open.
2. Select your audio interface device from the "Driver" drop menu.

3. Select the same device from the "Input Device" drop menu.

Figure 6.2 Equipment setup

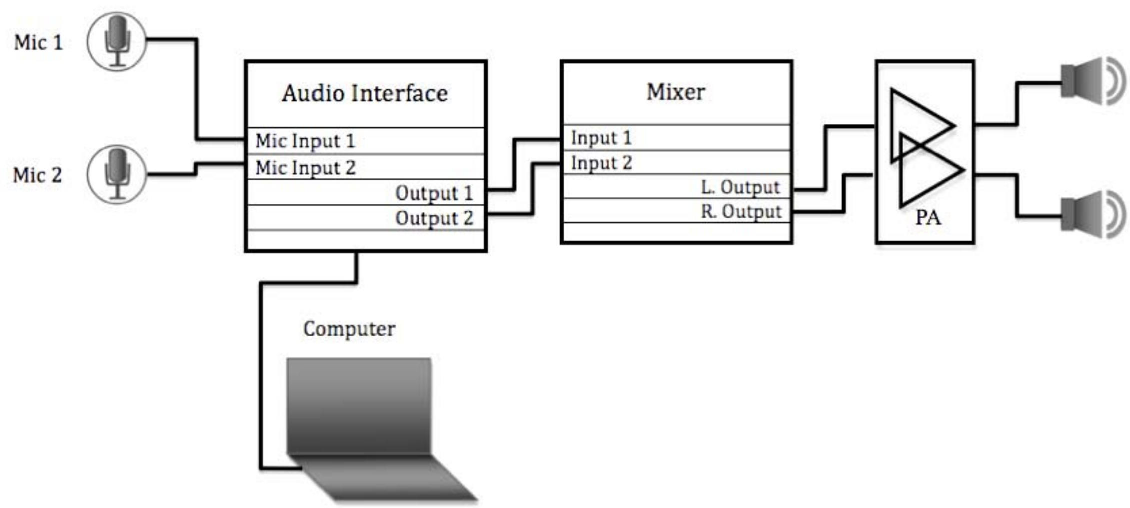
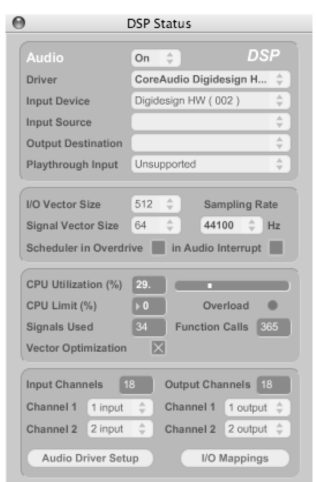


Figure 6.3 DSP status window in Max/MSP



The electronics part of *Normal Mode* was composed in Max/MSP 5.0.4 for the Macintosh operating system. All the files, which are required for the performance of this part, are located in the folder Normal Mode MaxMSP and should be kept there at all times. Moving files out of this folder may result in a malfunction of patchers.[1] The patcher Control.maxpat appears first on the alphabetic list of files in this folder. This patcher enables the fast opening and closing of each one of the five movement patchers which are required for the performance of the piece. The movement patchers PartI.maxpat, PartII.maxpat, PartIII.maxpat, PartIV.maxpat and PartV.maxpat are operated respectively in the five parts of the piece. These files should not be opened simultaneously. This will result in a CPU overload. The rest of the files in the folder are controlled and activated by these files.

The performance of the electronics part of *Normal Mode* is carried out by the following basic procedure:

1. Launch Max/MSP and open the patcher Control.maxpat. The interface window of this patcher is shown in Figure 6.4. Leave this patcher open during the entire performance.
2. The control patcher includes five **button** objects, (a) through (e), each of which opens one of the movement patchers. The **number box** (f) indicates which part is being opened.
3. Click on the **button** object " Part I" (a) to open the movement patcher of this part. The window interface of this patcher is shown in Figure 6.5. If needed, scroll to see all the objects.

---

1. The folder Normal Mode MaxMSP should be copied from a CD to the desktop or the hard drive of the computer before operating the patchers.

4. Click on the **toggle** object “Audio on/off” (a) in Figure 6.5 to turn on the audio output of the movement patcher of Part I. An X that appears in the toggle indicates that the audio is on.
5. Press the spacebar to activate the patcher. The LED lights (b) in the lower part of the window will reflect the sound generated by the patcher.
6. Continue to activate the patcher according to the notation of cues in the electronics part. The number in the **number box** titled “Cue Num” (c) indicates which cue is currently playing, and corresponds to the numbering of the cues in the score. (See below about the notation of the electronics).
7. When the performance of Part I ends, click on the **toggle** object “Audio on/off” to turn off the audio of the patcher. The X will disappear.
8. Click on the **button** object “ Part II” in the control patcher, (b) in Figure 6.4, to change to the movement patcher of part II. The control patcher will first close the current file and then will open the next file.
9. Repeat steps 3 through 8 to operate each one of the following movement patchers.

Figure 6.4 The interface window of the control patcher

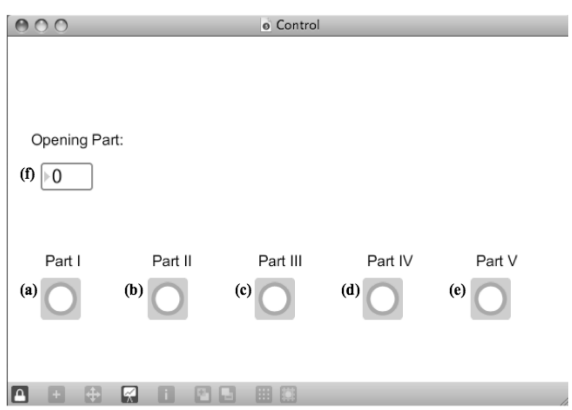
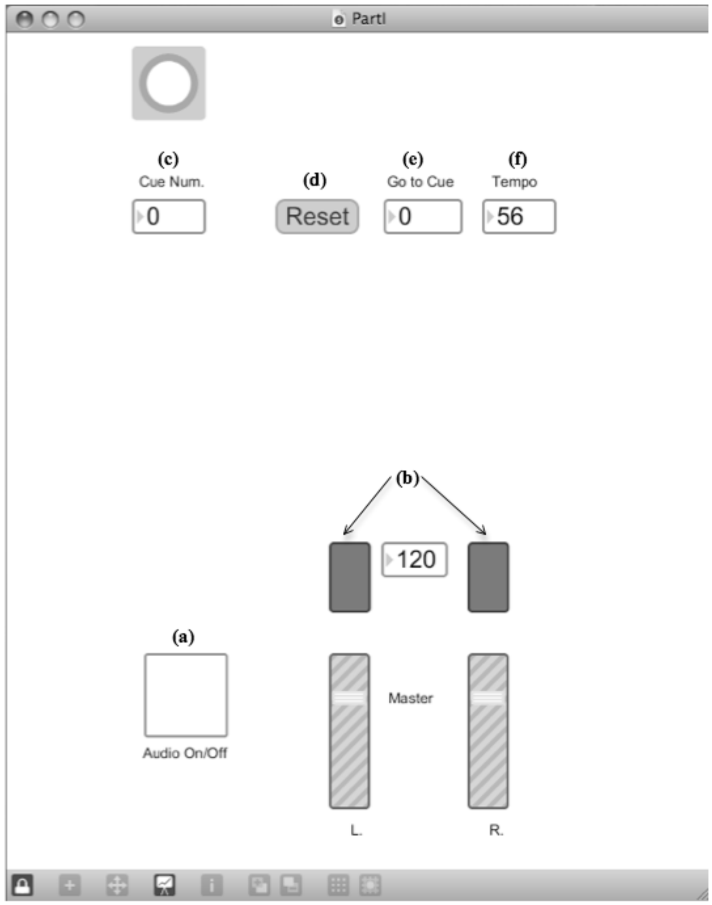


Figure 6.5 The interface window of Part I



The movement patcher has additional features that accommodate different performance and rehearsal needs. The spacebar activates cues only in ascending order. If you wish to go back to the beginning, click on the “Reset” **button**, (d) in Figure 6.5, the number 0 will appear in the “Cue Num” **number box** (c). If you now press the spacebar the patcher will play cue number 1.

In some cases you may have to quickly select a cue other than the one that will be activated next by the spacebar. Such a need can arise if you lose you place in the part and you have to jump ahead, or if, in a rehearsal, the conductor decides to repeat previous bars or sections. To select a specific cue, click and scroll the numbers in the **number box** titled “Go to Cue,” (e) in Figure 6.5, until you reach the number of the desired cue. Release the mouse and press the spacebar to activate this cue.

With each cue, the movement patcher organizes the sounds it produces in a rhythmic structure which is part of the temporal organization of the electronics (see Chapter 4). In many cues, the sound is not heard immediately with the stroke of the spacebar, but is delayed by predetermined time intervals. These time intervals are notated with standard rhythmic notation in the score. (See about the notation below). If the piece is rehearsed or performed in tempo markings other then those indicated in the score, these time intervals have to be adjusted to the new tempos. To adjust these time intervals to a new tempo marking, click and scroll the numbers in the **number box** called “Tempo,” (e) in Figure 6.5, until you reach the new tempo marking.

The movement patchers PartIV.maxpat and PartIV.maxpat are interactive and incorporate real-time processing of acoustic instrumental sound. In these parts, soloists from the ensemble will step up to the microphones. The patchers for these parts include in their interfaces objects that control the input of sound. The movement patcher PartII.maxpat is interactive with the clarinet. The clarinetist should play to microphone 1. The patcher is set to receive the clarinet sound from microphone 1 through input channel 1 of the audio interface device. The interface of this patcher is shown below in Figure 6.6. The audio input in this patcher is turned on together with the audio output by a click on the toggle “Audio on/off”. The fader (a) in the upper-right corner of the patcher controls the levels of the input signal. This fader is automated by presets and does not need to be adjusted. The "Pre" LED light (b) indicates whether or not the patcher receives incoming signals. The "Post" LED light (c) will reflect signals only when the fader is brought up by the presets.

Figure 6.6 The interface window of Part II

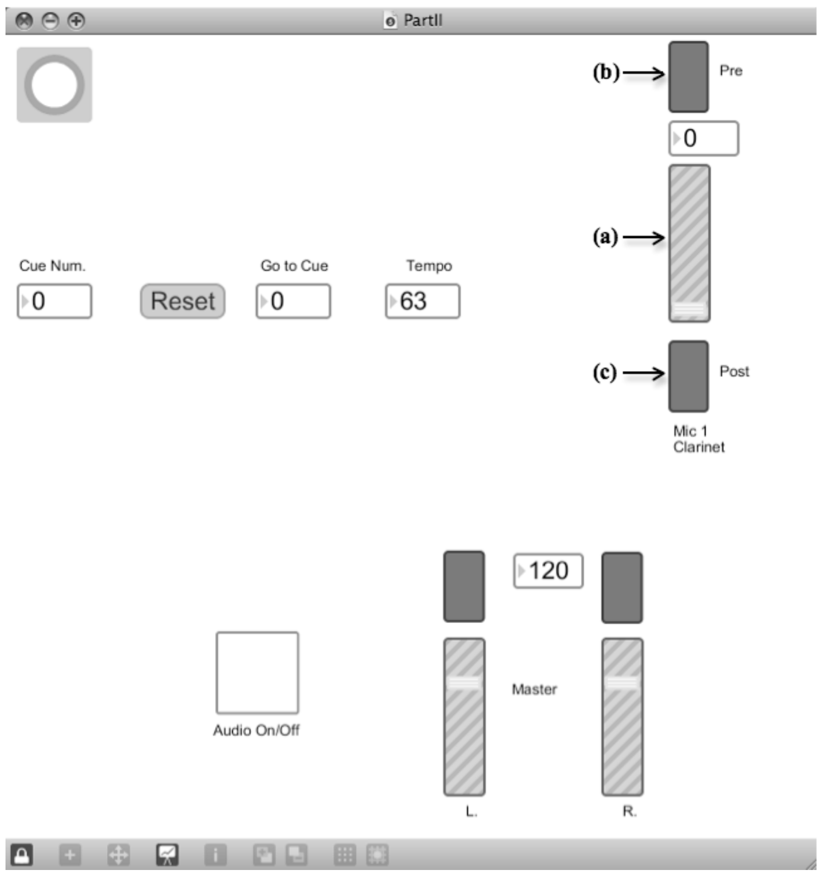
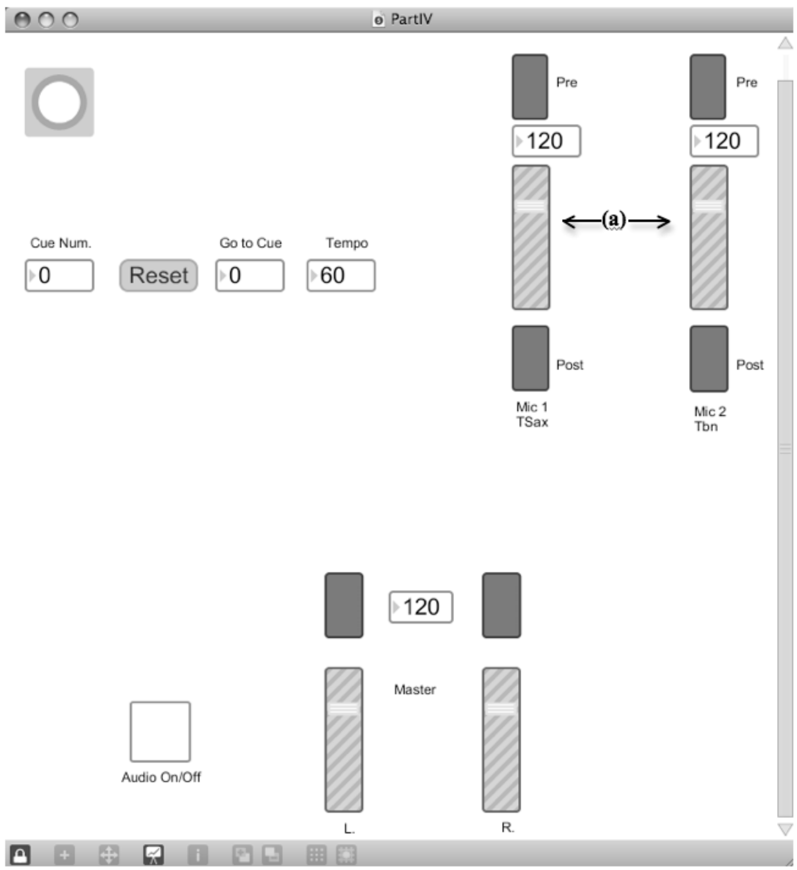


Figure 6.7 shows the interface of the movement patcher of Part IV. This patcher is interactive with the tenor saxophone and the trombone. The tenor saxophone should play to microphone 1 and the trombone to microphone 2. Similar to Part II, the audio input in this patcher will turn on together with the audio output by a click on the toggle "Audio

on/off". Unlike Part II, the faders of the input channels (a) are not automated and they should be left in the unity-gain position for the entire movement.

Note: in many of the interactive cues, the patcher will produce sound only when it receives an input signal. Therefore, the proper function of the sound system, with regard to the interactive parts, should be checked while the soloists are playing to the microphones.

Figure 6.7 The interface window of Part IV



The temporal organization of the electronics of *Normal Mode* incorporates both predetermined elements and random selection. The notation of the electronics part in the score reflects only the predetermined elements. Standard rhythmic values specify the timing of events but the notation does not indicate pitch. The notation appears on two staves combined by a bracket. The lower staff (Technician) notates the location of cues and their consecutive numbers. It simply tells the technician when to press the spacebar in order to activate the movement patcher. All spacebar strokes occur on downbeats. The upper staff notates the time intervals by which sounds will be delayed after the downbeat in each cue. It also provides the approximated durations of these sounds.

In the excerpt in Figure 6.8, the quarter note with a slashed notehead in the lower staff marks the location of cue number 19. It indicates to the technician to activate the patcher on the first beat of the bar. The cue number appears in the circle above the note. The bracket, which extends from this circle, frames the events that will follow this activation. The notation with the triangle notehead in the upper staff indicates that the first succession of sounds will begin after an eighth-note rest and will continue for an approximate duration of a whole note. The second succession of sounds will begin on the third beat of the bar after a sixteenth-note rest and will continue for an approximate duration of more than a quarter note. As mentioned in Chapter 4, such successions of sounds, which are generated by the patcher, are organized in randomly selected rhythmic structures. Therefore, these rhythmic structures cannot be notated. Nevertheless, the approximate durations, which are noted in the upper staff, give some indication as to what will follow the activation of the patcher.

In Part II and Part IV, the interactive parts, special noteheads indicate to the technician when the processing of the instruments' sounds takes place and which instrument is being processed. These noteheads are shown in Table 5. In the excerpt shown in Figure 6.9, in cue number 15, the sound of the tenor saxophone is processed on the first beat of the bar and the sound of the trombone is processed after an eighth-note rest. This notation was devised for the purpose of better performance coordination between the musicians and the electronics. It also assists the technician in checking the proper function of the sound system.

Figure 6.8 The notation of the electronics

Table 5. Special noteheads for the interactive movements

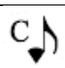
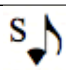
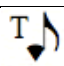
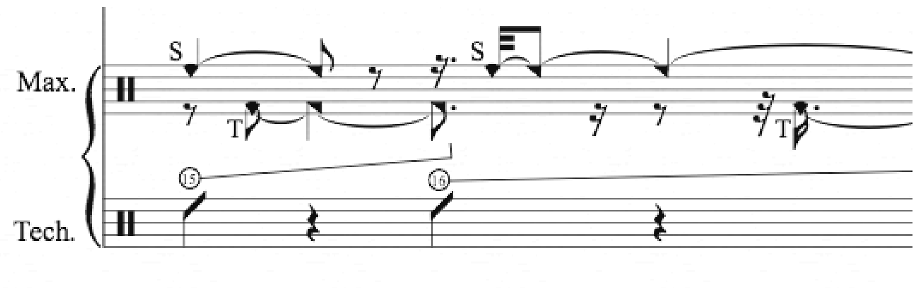
Notehead	Instrument
 C	Clarinet
 S	Tenor Saxophone
 T	Trombone

Figure 6.9 The electronics' notation in the interactive parts



Notation

gliss. (strings, trombone) lip-bend gliss. (woodwinds, horn)

fingered gliss.

[contour gliss.] fingered gliss. with intervening change of direction

scoop

repeated scoops of diminishing intervals

doink

fall

[bend] guitar like bend (cello)

[tr] trill half step above the note

shake

[s-t] slap tongue

“ghost” tongue

triple-tongue

[sing / play] sing square note, play normal note

[air only] sustained air sound, no pitch

[air only fltr] flutter tongue air sound, no pitch

[air only fltr valves] flutter valves air sound, no pitch

change from normal sound to air sound

change air sound to normal sound

inhale

key clicks, specified

key slaps, specified

random key clicks

random key slaps

tap on bell (brass)

kiss

mute indicated open strings w/ L.H.

bow on the bridge over indicated open strings

bow on tail piece

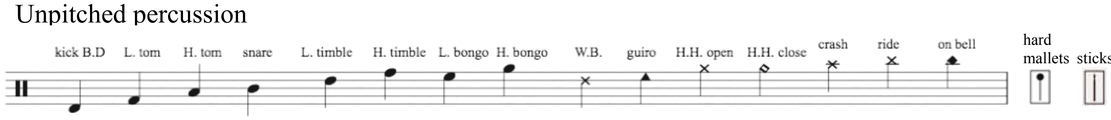
plunger

gradual change of hand mute/plunger (open to close)

no fingering change, let pitch drop (horn)

irregular plunger change

Unpitched percussion



Normal Mode

Transposed Score

Part I

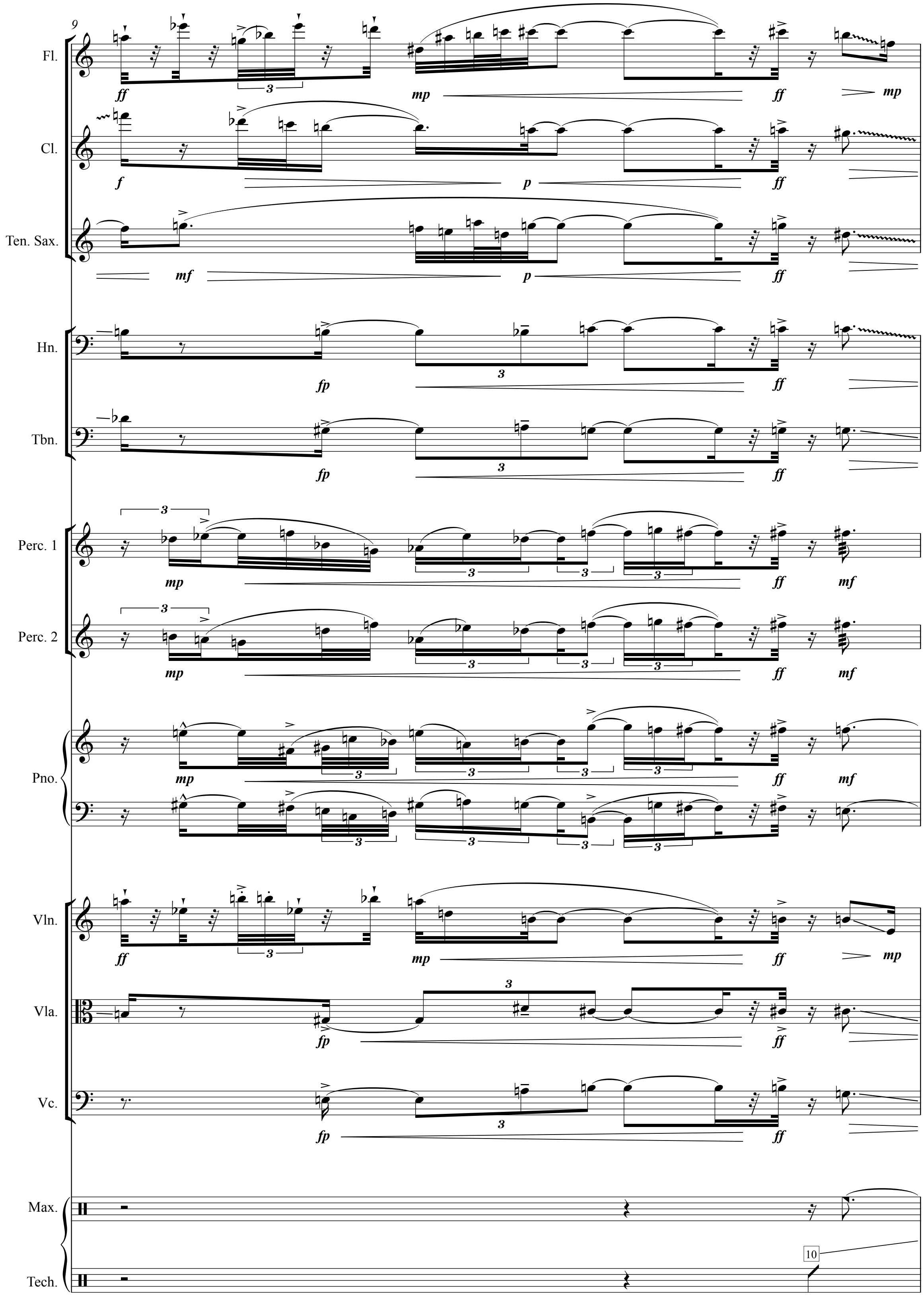
Israel Neuman

3
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.
scoops
pizz.
4
5
6


5
Fl.
pp
sffz
growl
p
mf
Cl.
pp
doink
mf
3
ff
p
Ten. Sax.
sffz
growl
p
mf
Hn.
Tbn.
Perc. 1
p
Perc. 2
3
p
Pno.
3
p
pizz.
Vln.
mf
Vla.
mf
sul G
bend
Vc.
mf
Max.
7
8
Tech.

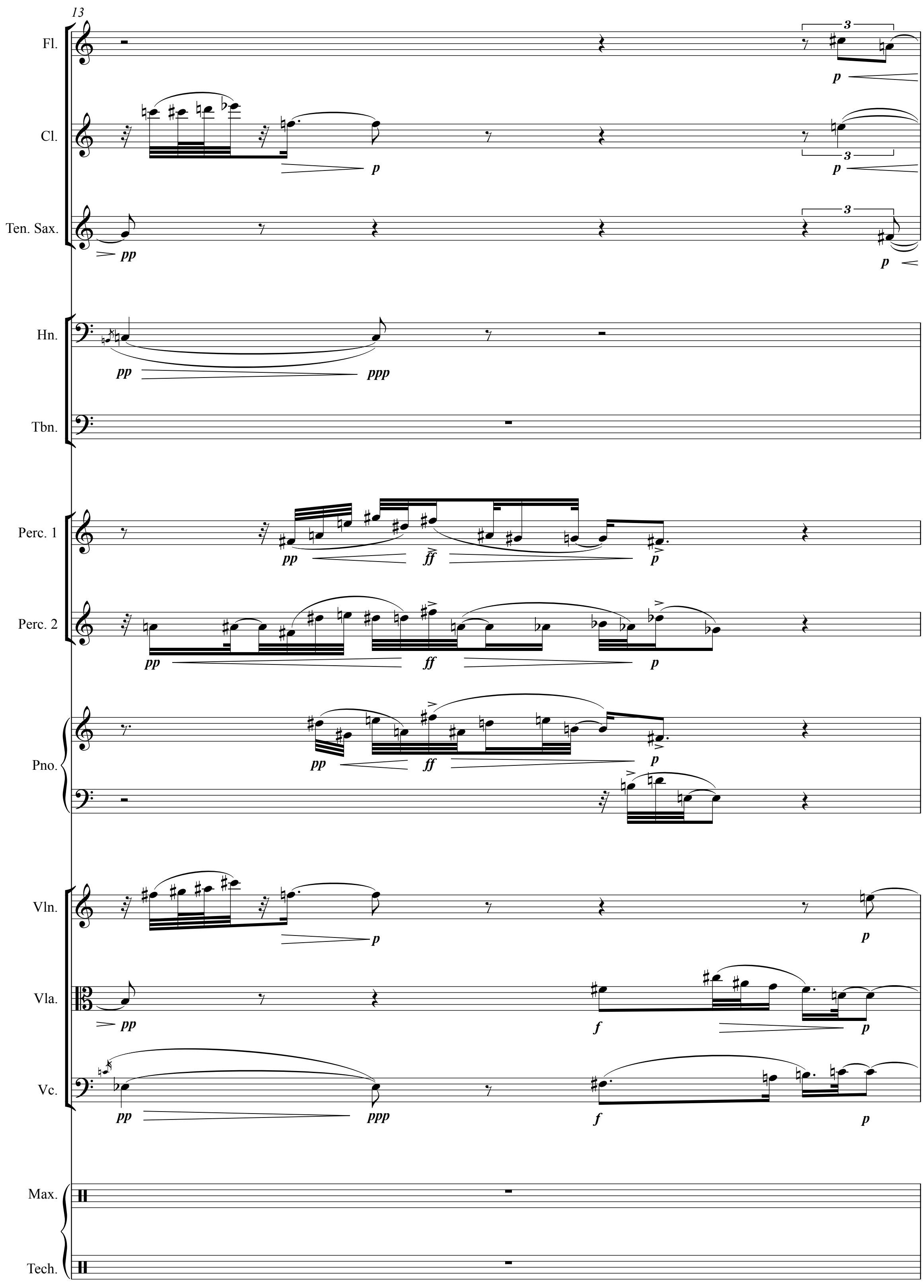

7
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.
3
f
ff
mf
mp
arco

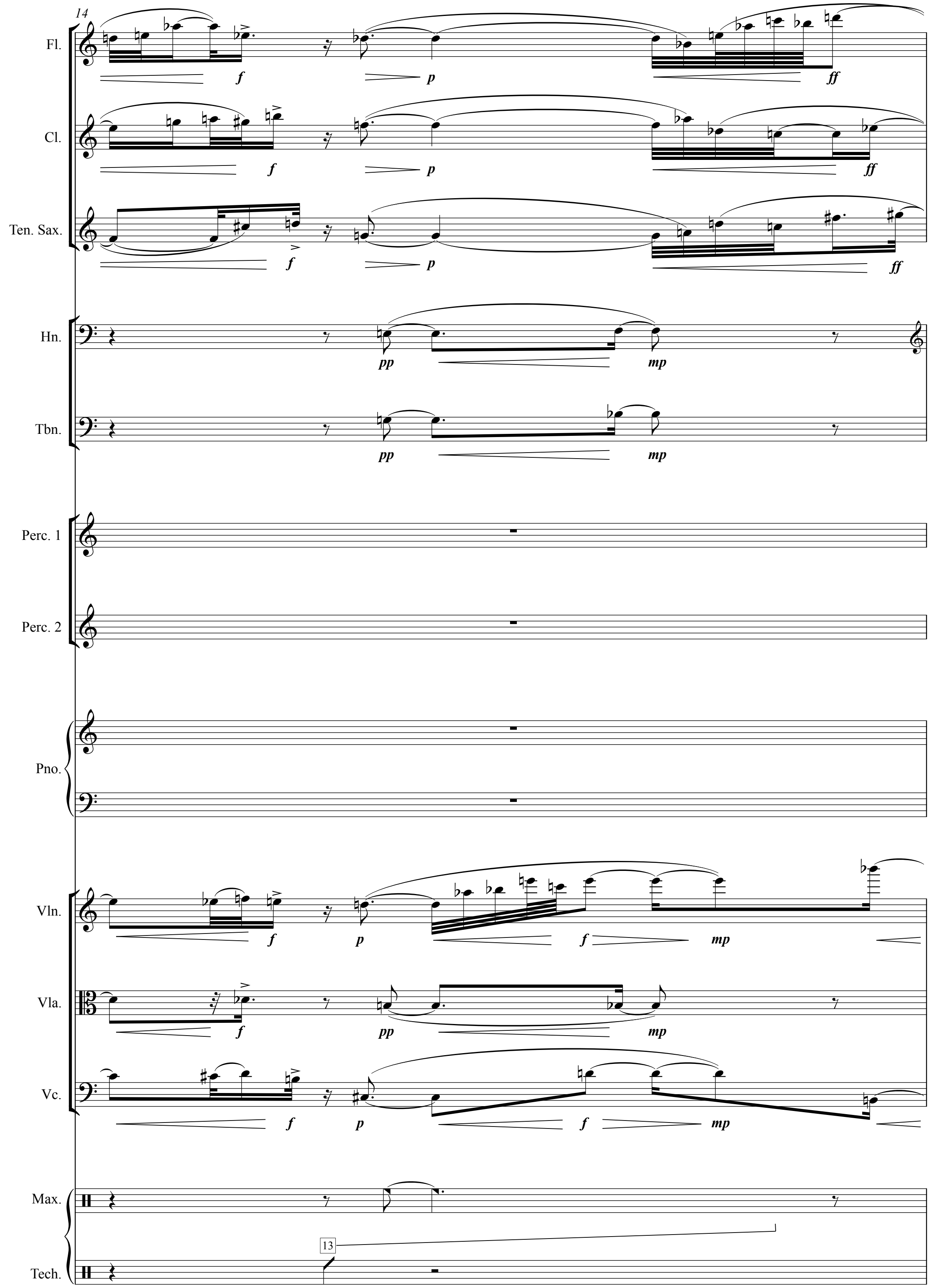

8
Fl.
Cl.
Ten. Sax.
s-t
Hn.
triple-tongue
Tbn.
triple-tongue
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
9
Tech.


9
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.
ff
mp
mf
f
p
fp
10


10
Fl.
fff
Cl.
mp
f
mf
Ten. Sax.
mp
contour gliss.
mp
f
Hn.
mp
mp
f
scoops
mp
Tbn.
mp
shake
mf
p
plunger
mf
Perc. 1
ff
mf
Perc. 2
p
mf
Pno.
l.v.
l.v.
Vln.
f
Vla.
mp
f
Vc.
mp
f
Max.
Tech.
11
12

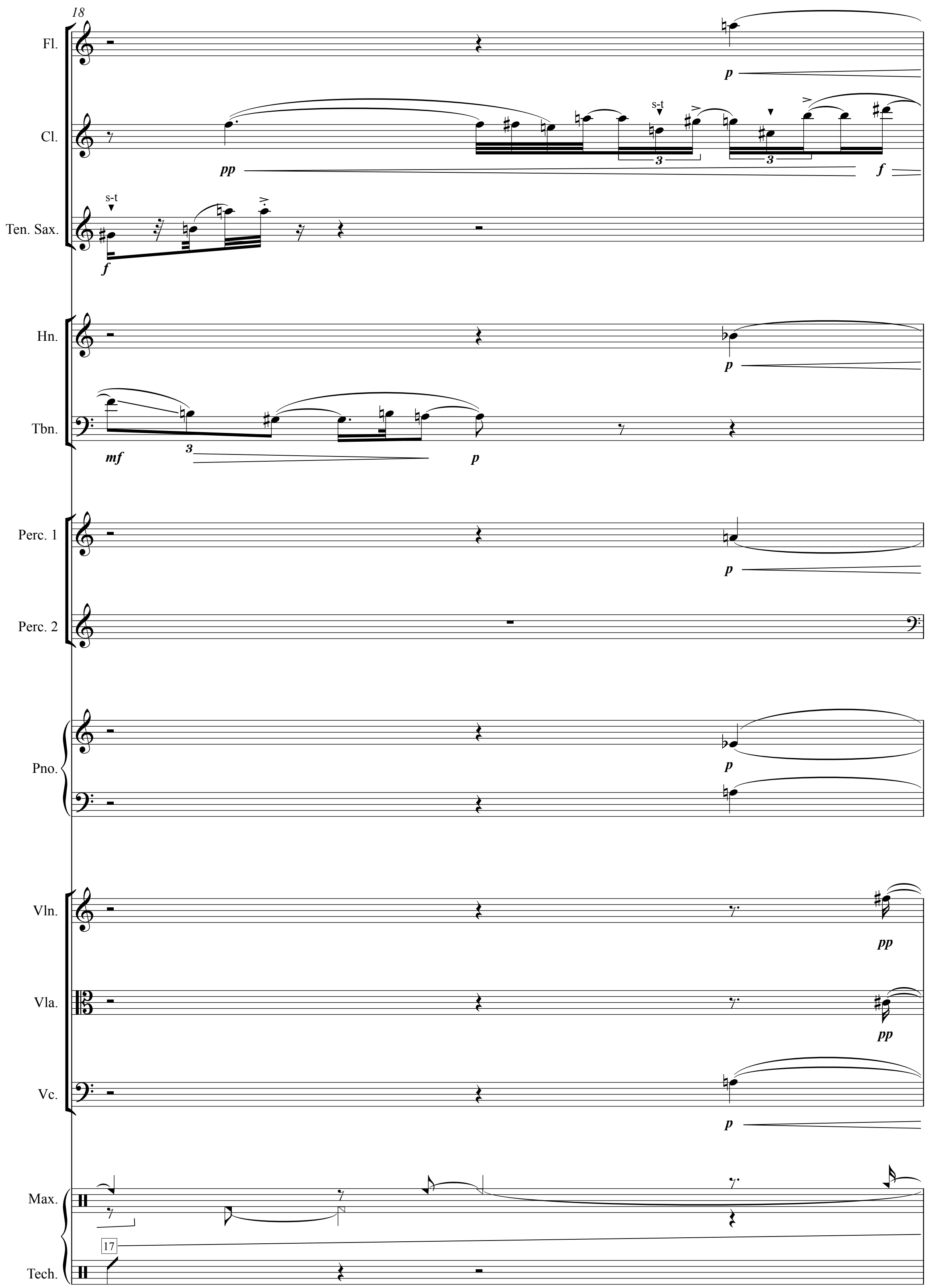

12
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.
f
ppp
mp
3
p


13
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.


14
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.
13


15
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
14
Tech.

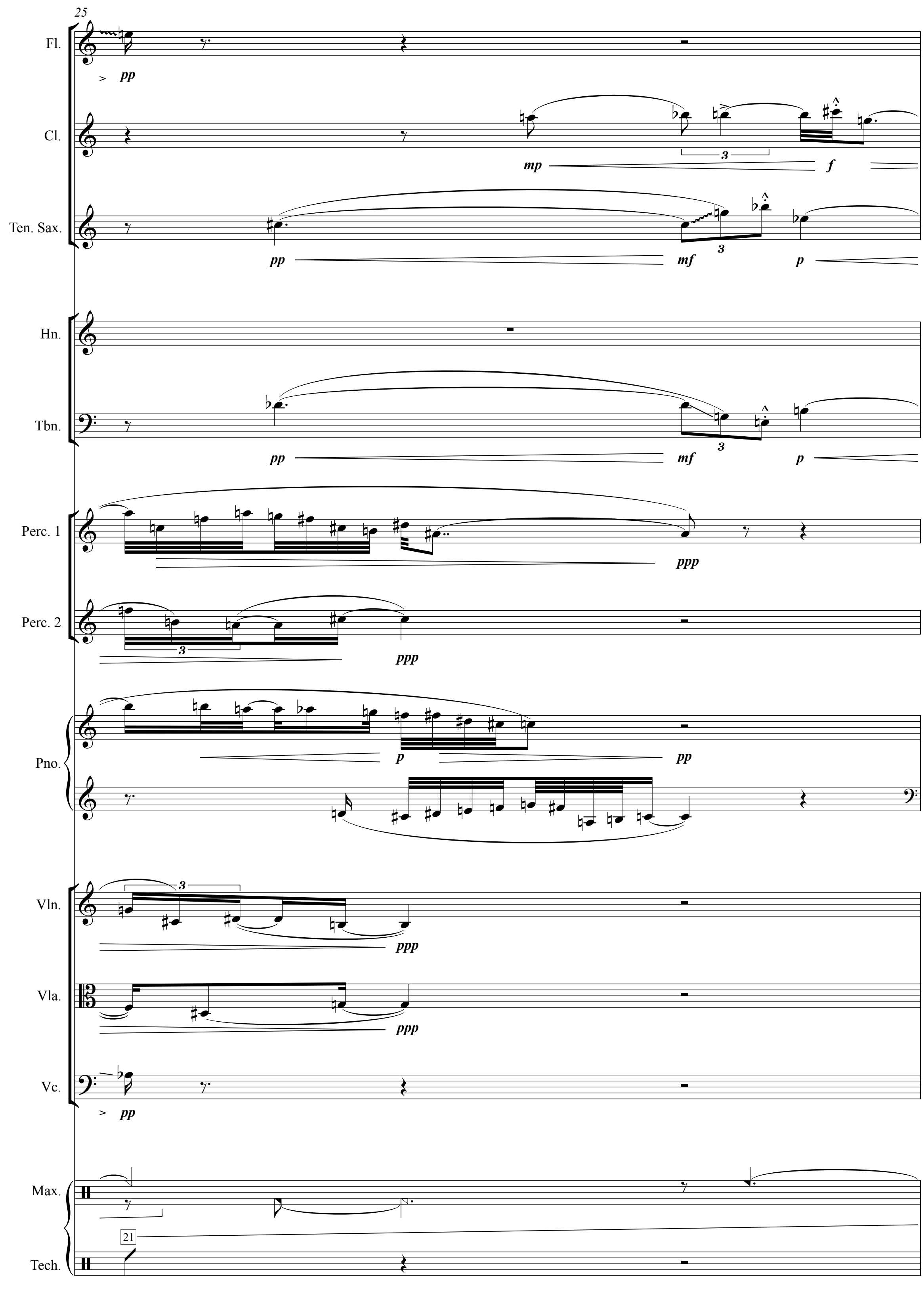

16
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.
15
16


18
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.
s-t
17


19
s-t
Fl.
mf
pp
mp
p
pp
3
3
Cl.
ppp
mf
pp
Ten. Sax.
Hn.
mf
pp
mp
mf
p
f
3
Tbn.
Perc. 1
mf
p
pp
ppp
Ped.
Ped.
Perc. 2
p
mf
p
Pno.
mf
p
pp
Ped.
Vln.
ff
pp
Vla.
ff
pp
Vc.
mf
pp
mp
mf
p
f
3
Max.
18
Tech.


21
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.
s-t
mf
p f pp
pp mf p f
p
pp mp
pp mf p f
pp
mp
pp
8va
pp
ppp
p
p
p
pp mp


23
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.
pp
mf
p
f
ppp
mp
mf
19
20


25
Fl.
pp
Cl.
mp
3
f
Ten. Sax.
pp
3
mf
p
Hn.
Tbn.
pp
3
mf
p
Perc. 1
ppp
Perc. 2
3
ppp
Pno.
p
pp
Vln.
3
ppp
Vla.
ppp
Vc.
pp
Max.
21
Tech.


26
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.
growl
pizz.
22
mf
ff
pp
p
f
mp


28
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.
air only
fltr
tap on bell
trem.
l.v.
arco
ricochet
on the bridge
mute strings w/ L.H.
23
24
25


30
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.
air only
fltr
mp
ppp
ff
air only (=𝅗𝅥)
fltr valves
tap on bell
p
mute strings w/ L.H.
mute strings w/ L.H
f
26


32
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.
random key clicks
air only
ppp
p
pp
mf
f
fltr
tap on bell
trem.
mp
pizz.
arco
27
28
29


34
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.
random key clicks
random key clicks
random key clicks
air only
fltr
air only
fltr
air only
fltr
air only
fltr
tap on bell
trem.
tap on bell
30
31


36
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.
ff
mf
pp
mp
Ped.
32


39
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.
doink
shake
l.v.
8va
33
mf
ff
fp
p
f
mp
pp
sf

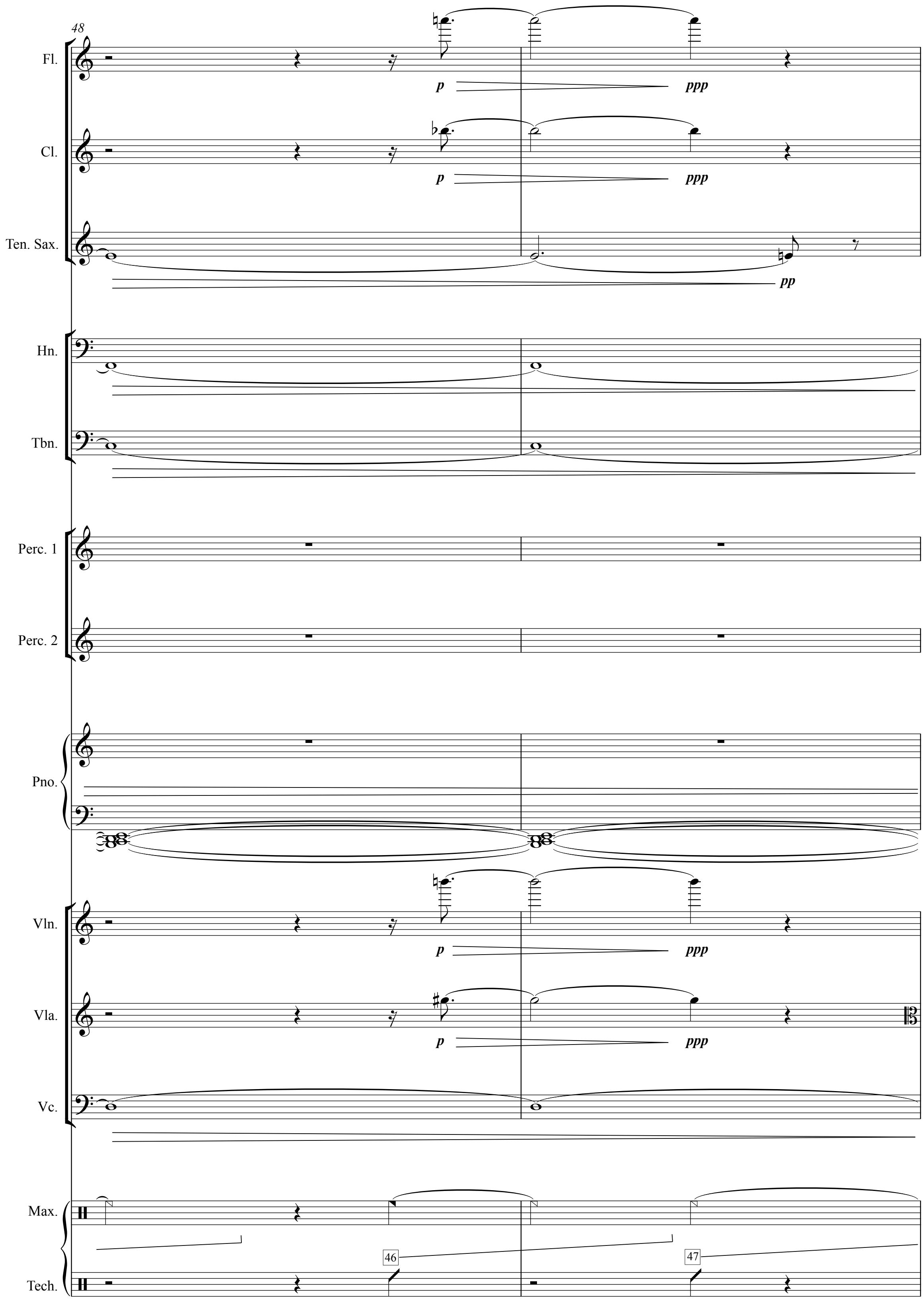

41
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.
scoop
ff
p
fall
f
l.v.
mf
3
Ped.
sfz
34
35


42
Fl.
ff
Cl.
ff
ff
Ten. Sax.
ff
ff
Hn.
f
fall
fz
Tbn.
f
fall
fz
Perc. 1
ff
Perc. 2
ff
Pno.
ff
Vln.
Vla.
Vc.
ff
Max.
36
Tech.


43
Fl.
mp
Cl.
mp
Ten. Sax.
Hn.
Tbn.
Perc. 1
mp
Perc. 2
mp
Pno.
mp
p
3
3
Vln.
f
Vla.
f
Vc.
f
Max.
37
38
39
Tech.


44
Fl.
contour gliss.
Cl.
contour gliss.
Ten. Sax.
contour gliss.
Hn.
contour gliss.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
40
41
42
Tech.


46
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.
contour gliss.
contour gliss.
contour gliss.
contour gliss.
43
44
45


48
Fl.
p
ppp
Cl.
p
ppp
Ten. Sax.
pp
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
p
ppp
Vla.
p
ppp
Vc.
Max.
46
47
Tech.


50
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.
mf
ff
pp
mp
f
p
3
48
49

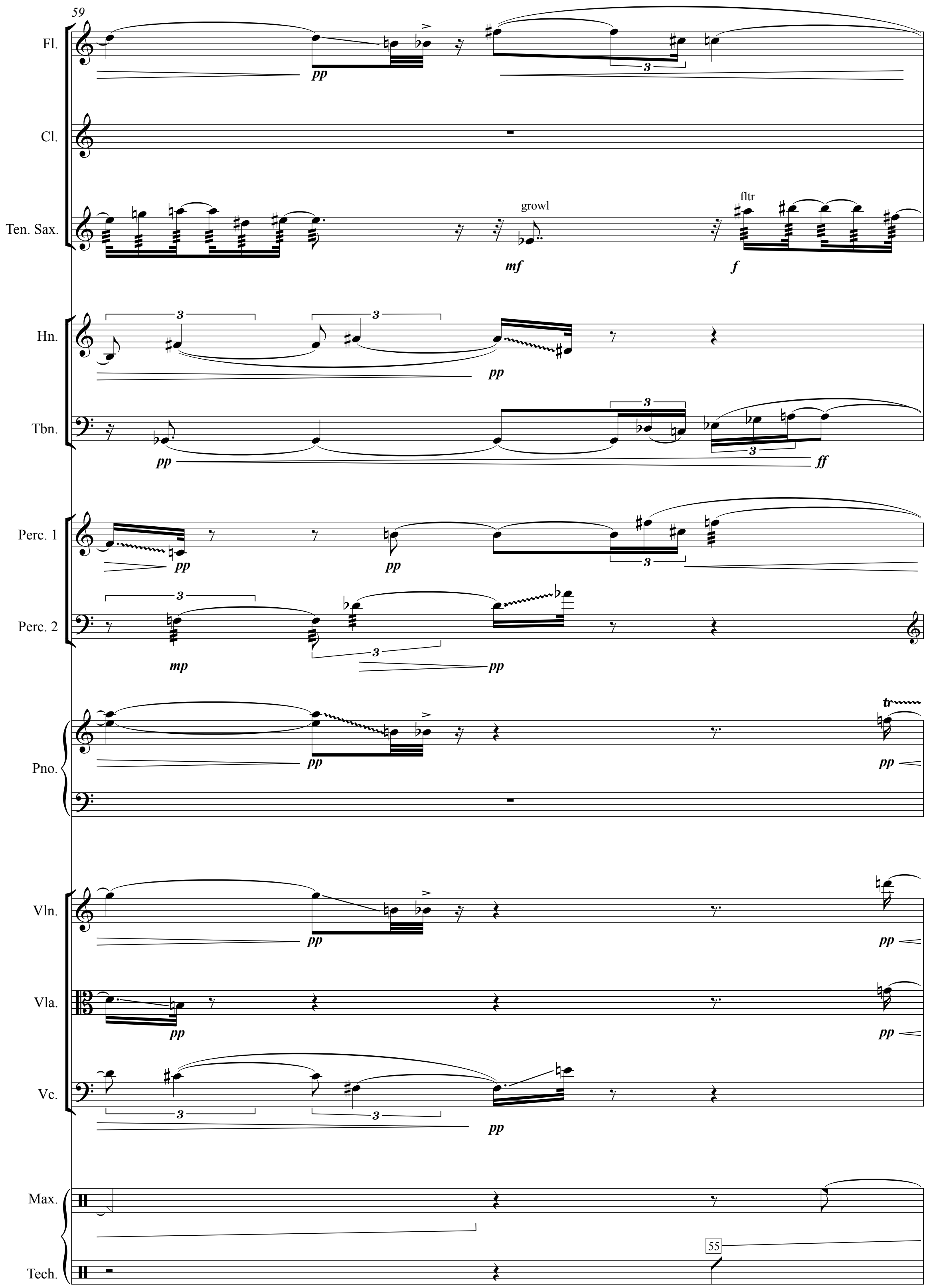

51
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.
50
mp
mf
p
ff
f


52
Fl.
p
Cl.
3
mp
f
Ten. Sax.
mp
Hn.
mp
Tbn.
mp
Perc. 1
Perc. 2
p
Pno.
mp
mf
f
Vln.
mf
ff
mp
Vla.
mf
ff
mp
Vc.
mf
Max.
Tech.
51

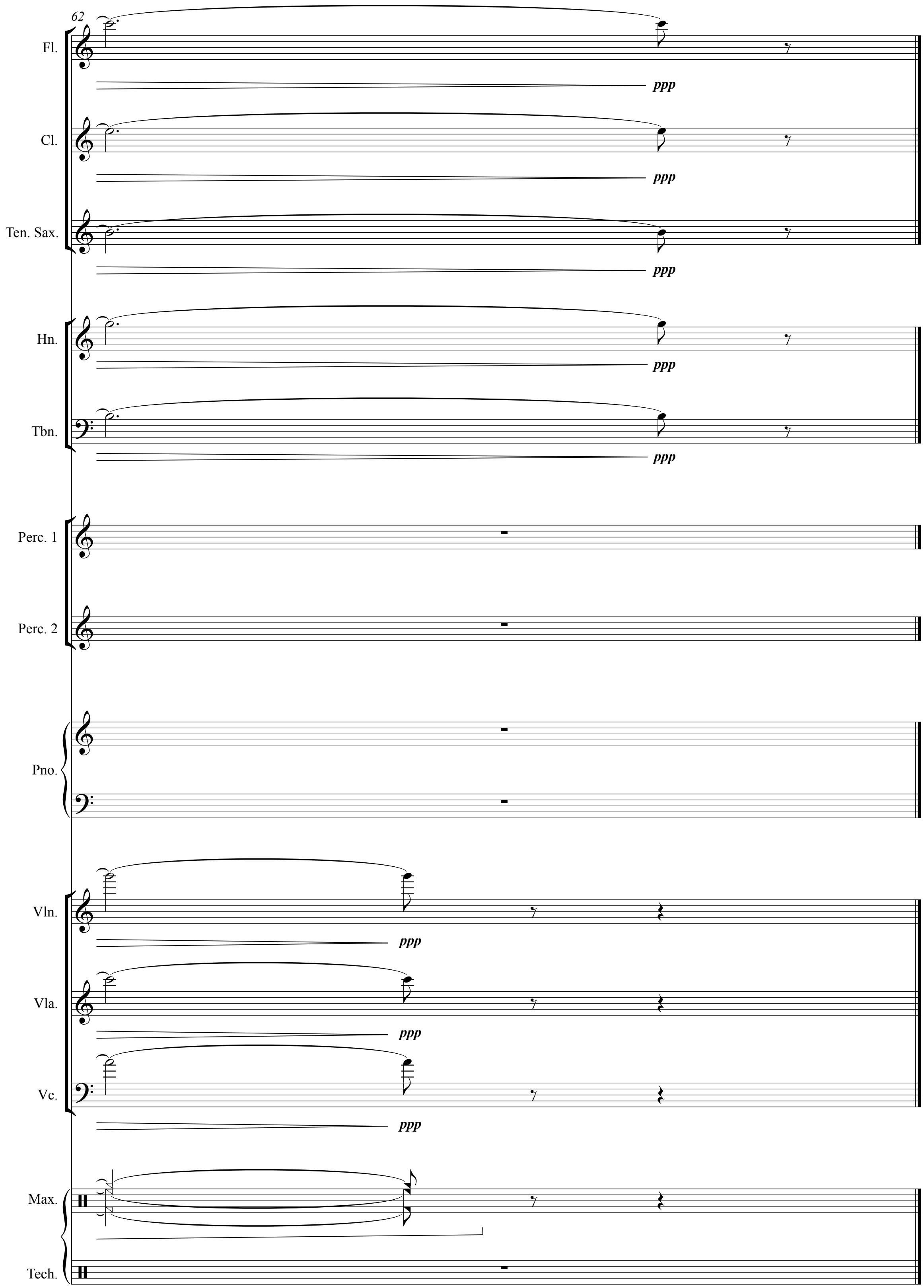

53
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.
mf
p
f
pp
ff
mp
s-t
Ped.


55
Fl.
p
ppp
pp
Cl.
p
ppp
pp
Ten. Sax.
3
s-t
p
ff
sffz
mf
3
3
s-t
Hn.
pp
p
pp
Tbn.
ff
p
3
f
p
f
Perc. 1
Perc. 2
Pno.
pp
p
pp
Vln.
p
ppp
pp
Vla.
p
ppp
pp
Vc.
p
ppp
Max.
52
53
Tech.


57
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.
s-t
fltr
3
sffz
mp
ff
f
p
pp
54


59
Fl.
pp
Cl.
Ten. Sax.
growl
mf
fltr
f
Hn.
pp
Tbn.
pp
ff
Perc. 1
pp
pp
Perc. 2
mp
pp
Pno.
pp
tr
pp
Vln.
pp
pp
Vla.
pp
pp
Vc.
pp
Max.
Tech.
55


60
Fl.
Cl.
Ten. Sax.
growl
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
pizz.
arco
Max.
Tech.
56
mf
p
fff
f
mp
ff
pp


62
Fl.
ppp
Cl.
ppp
Ten. Sax.
ppp
Hn.
ppp
Tbn.
ppp
Perc. 1
Perc. 2
Pno.
Vln.
ppp
Vla.
ppp
Vc.
ppp
Max.
Tech.

Part II

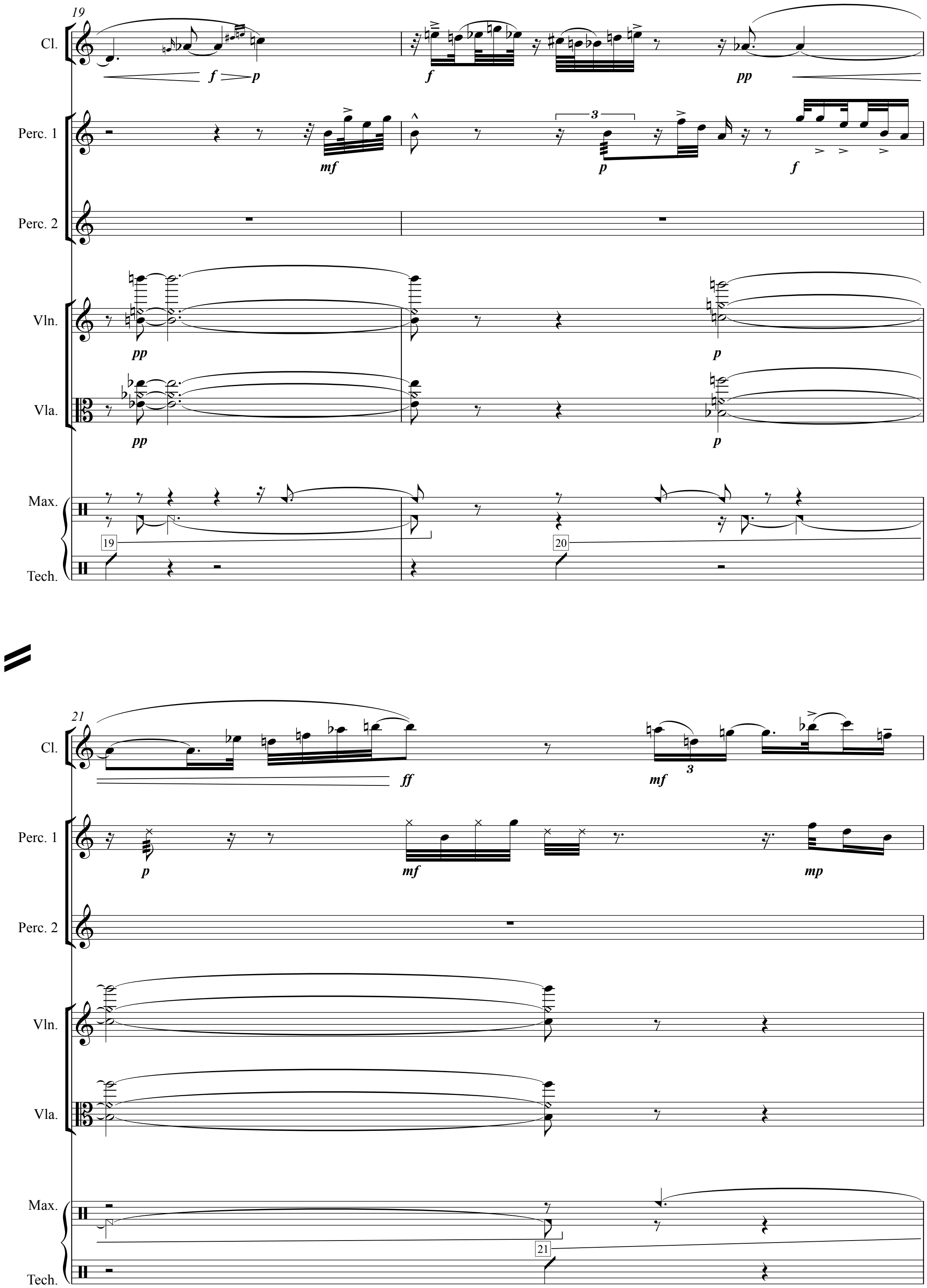

1
= 63
Fl.
to mic 1
Cl.
sffz
p
f
p
mf
Ten. Sax.
Hn.
Tbn.
Perc.
Perc. 1
pp
mf
p
Mar.
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
C
C
C
1
2
3
4
Tech.


4
Cl.
Perc. 1
Max.
Tech.
p
f
mp
ff
p
f
mp
mf
p
mf
p
C
C
C
5
6
7
6
growl
fltr
f
ff
pp
pp
p
mf
pp
mp
f
C
C
8
9
10
8
mf
mp
p
f
p
3


9
Cl.
f
mp
mf
Perc. 1
mf
p
Max.
Tech.
10
Cl.
p
f
pp
Perc. 1
p
mf
p
f
Max.
C
11
Tech.
11
Cl.
mf
growl
mp
3
Perc. 1
Max.
C
12
13
Tech.


12
Cl.
growl
ff
mp
f
Perc. 1
Max.
C
14
Tech.
13
Cl.
p
mf
p
f
fff
mf
Perc. 1
mf
f
ff
Perc. 2
mf
Vln.
pizz.
mf
Vla.
pizz.
mf
Max.
C
15
16
17
18
Tech.

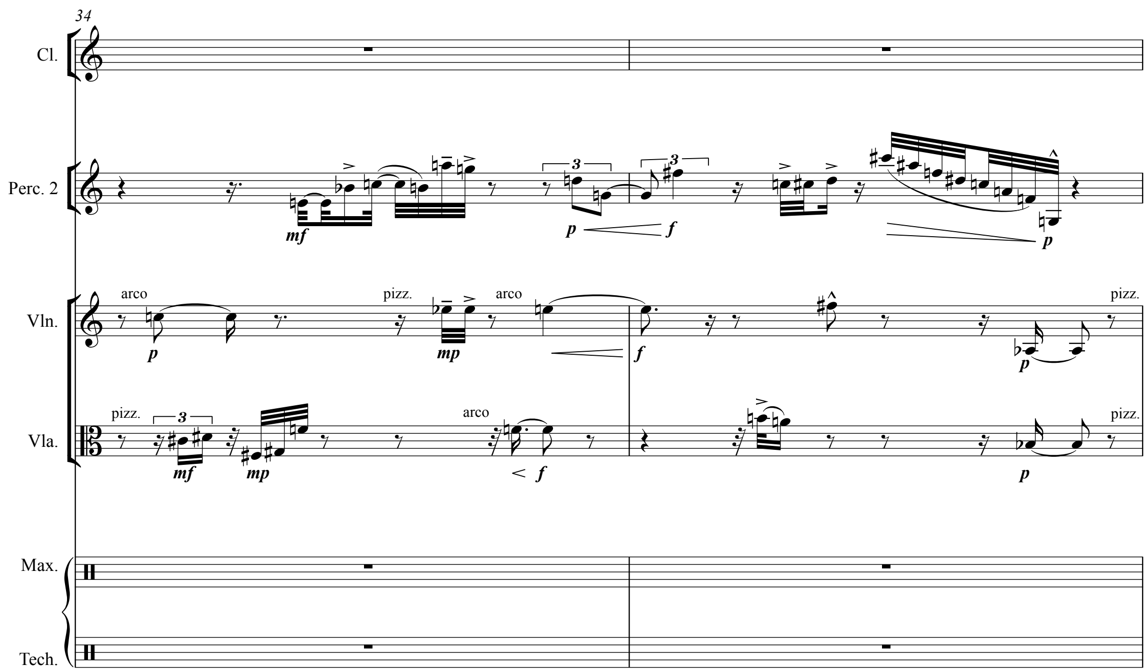

15
Cl.
Perc. 1
Perc. 2
Vln.
Vla.
Max.
Tech.
arco
pizz.
3
f
p
mp
mf
17
ppp


19
Cl.
Perc. 1
Perc. 2
Vln.
Vla.
Max.
Tech.
f
p
f
pp
mf
p
f
3
pp
p
19
20
21
Cl.
Perc. 1
Perc. 2
Vln.
Vla.
Max.
Tech.
ff
mf
3
p
mf
mp
21


22
Cl.
Perc. 1
Perc. 2
Vln.
Vla.
Max.
Tech.
fp
f
pp
ff
mf
mp
p
C
22
23
24
Cl.
Perc. 1
Perc. 2
Vln.
Vla.
Max.
Tech.
p
f
mf
ff
mf
mp
p
24

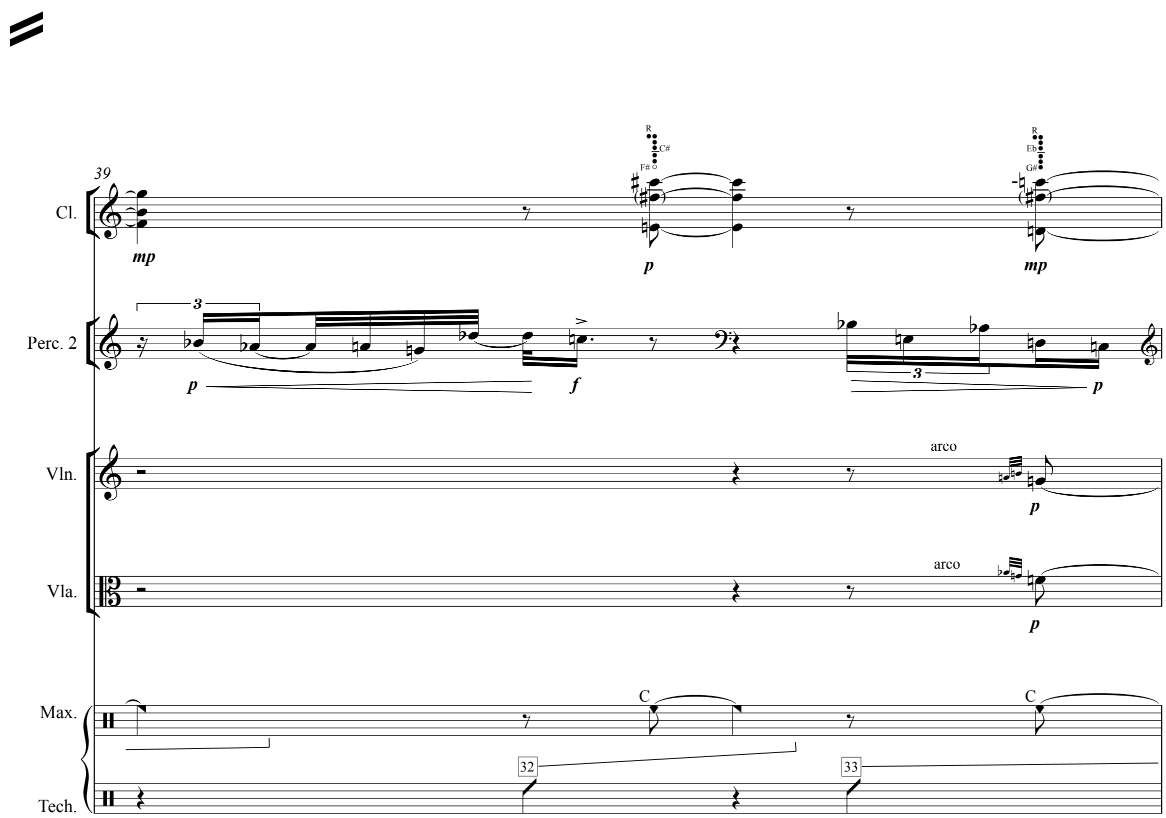

26
Cl.
scoops
Perc. 1
Perc. 2
Vln.
Vla.
Max.
C
25
Tech.
28
Cl.
s-t
Perc. 1
Perc. 2
Vln.
Vla.
Max.
C
26
Tech.

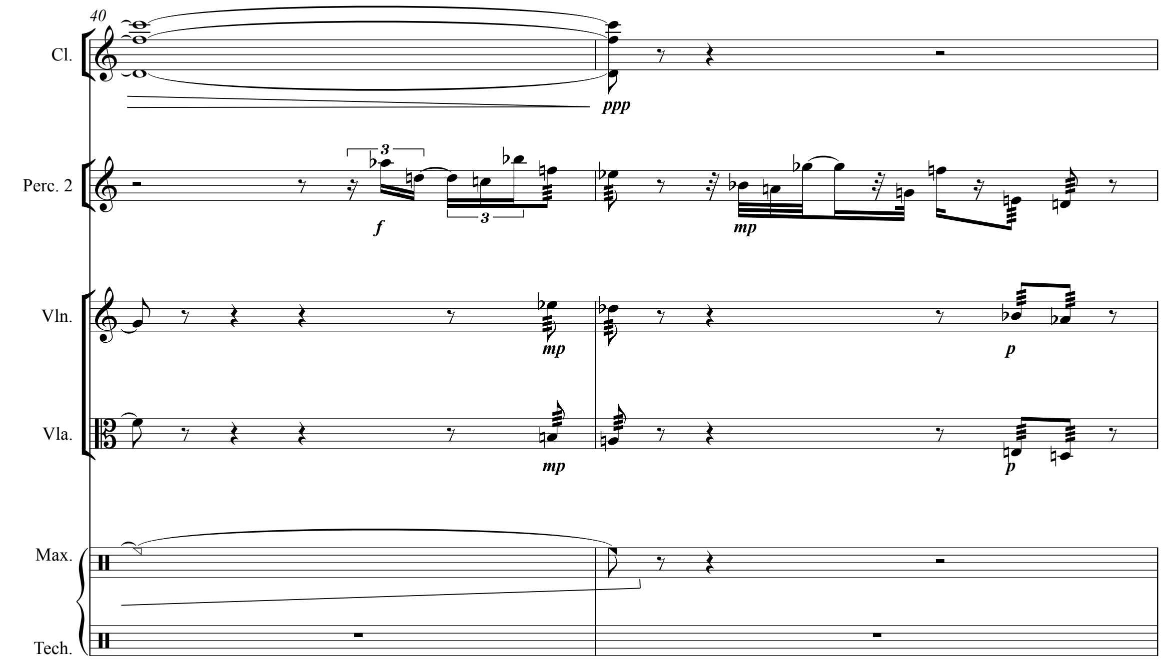

30
Cl.
s-t
Perc. 1
Perc. 2
Vln.
Vla.
Max.
Tech.
C
27
28
32
change to Vib.
pizz.
29


34
Cl.
Perc. 2
mf
p
f
p
Vln.
arco
pizz.
arco
p
mp
f
p
pizz.
Vla.
pizz.
arco
mf
mp
f
p
pizz.
Max.
Tech.


36
Cl.
p
mf
Perc. 2
p
mf
f
Vln.
mf
mp
arco
pp
pizz.
Vla.
mf
mp
arco
pp
pizz.
Max.
C
30
Tech.


38
Cl.
Perc. 2
Vln.
Vla.
Max.
Tech.
R
p
pp
mf
mf
mf
C
31


39
Cl.
Perc. 2
Vln.
Vla.
Max.
Tech.
R
C#
F#
Eb
G#
mp
p
mp
3
p
f
3
p
arco
p
arco
p
C
C
32
33


40
Cl.
ppp
Perc. 2
3
3
f
mp
Vln.
mp
p
Vla.
mp
p
Max.
Tech.


42
Cl.
ff
sfz
f
tr
Perc. 2
f
p
Vln.
mf
p
Vla.
mf
p
Max.
Tech.


44
Cl.
Max.
Tech.
fltr.
p sf mp ff mf mp mf
46
mp ff mp f p mf mp f p
C
34
35
48
growl
f sffz mp f p
36
37
38
50
mf ff ppp p pp
39
40
41


53
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.
Vib.
p
pp
pppp
mp
3


55
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
fltr
p
ppp
Perc. 1
ppp
mp
Perc. 2
ppp
mp
Pno.
tr
ppp
mp
Vln.
Vla.
Vc.
Max.
Tech.


57
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.
p
ppp
pp
pppp


59
Fl.
pppp
Cl.
pppp
Ten. Sax.
pppp
Hn.
Tbn.
Perc. 1
p
3
pp
Perc. 2
p
3
Pno.
Vln.
Vla.
Vc.
Max.
Tech.

Part III


= 82
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Vib.
Perc. 1
Mar.
Perc. 2
8va
Ped.
mp
7
Pno.
p
Vln.
Vla.
pizz.
arco
Vc.
mf
3
ff
f
Max.
1
2
3
4
Tech.

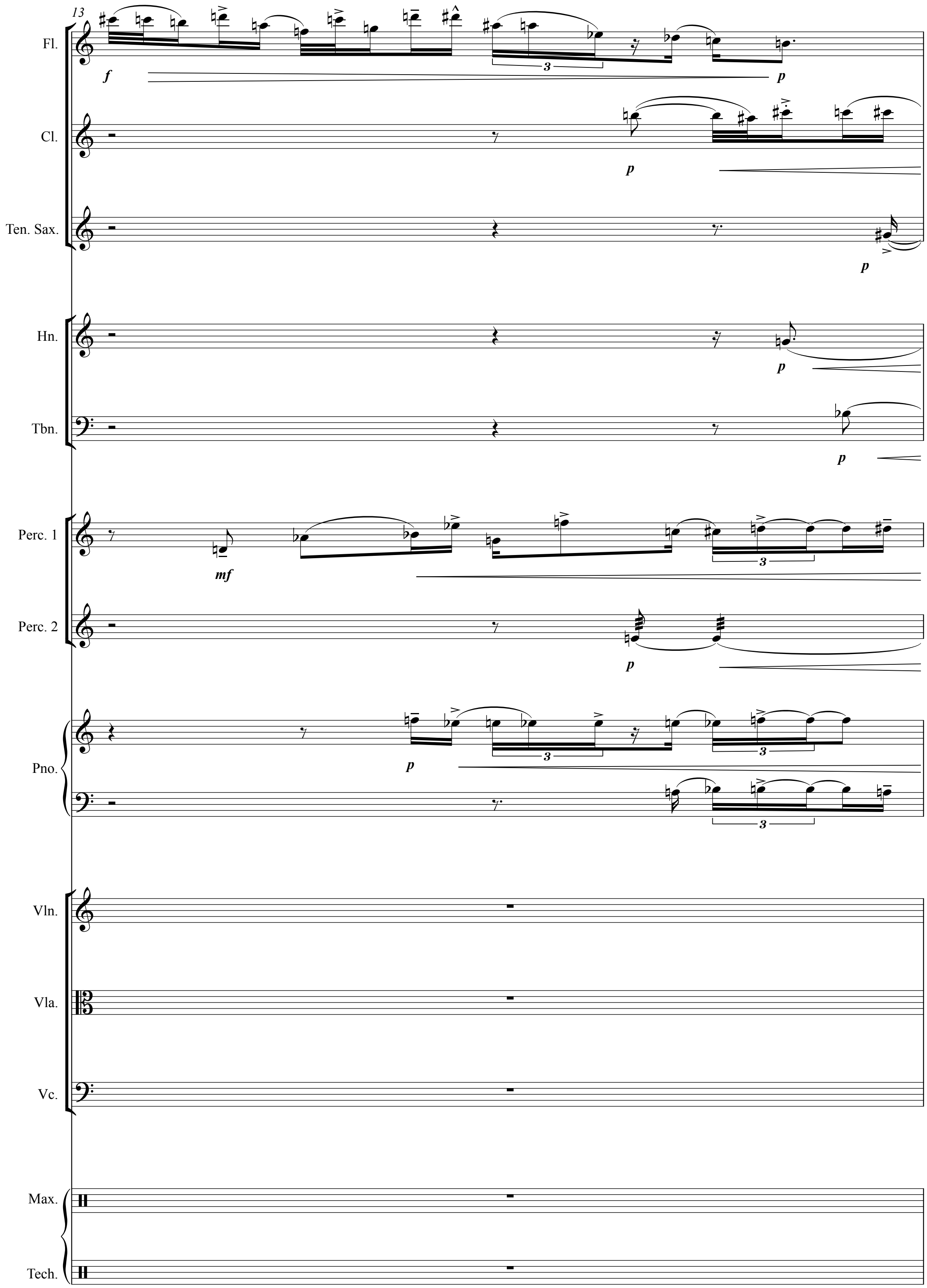

Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.
random key clicks
air only
fltr
tap on bell
trem.
Ricochet
on the bridge
mute strings w/ L.H.
15ma
Ped.
p
sffz
mf
pp
f
5
6

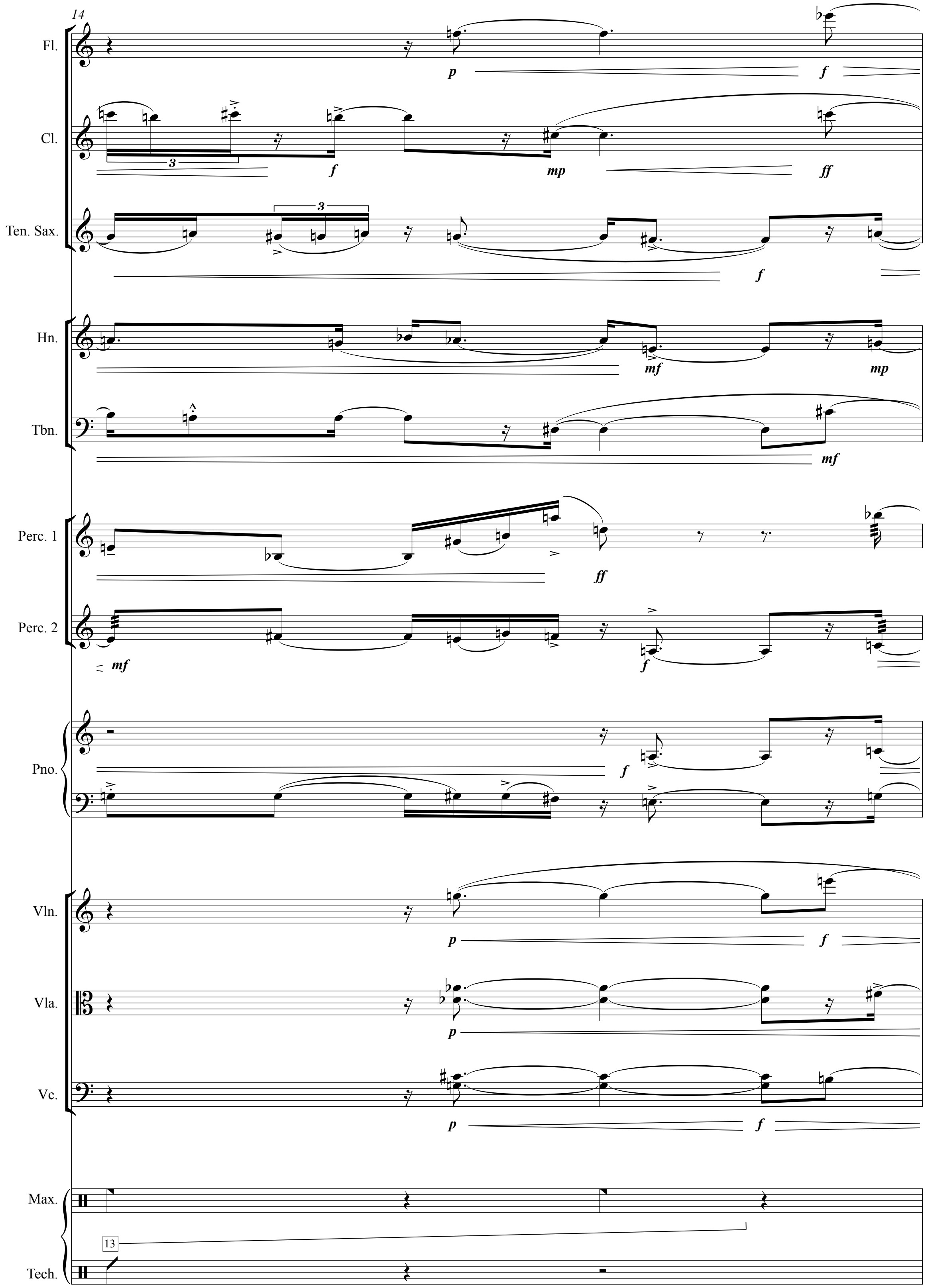

5
Fl.
ff
Cl.
Ten. Sax.
Hn.
f
3
mf
Tbn.
Perc. 1
mf
6
ff
mf
3
Perc. 2
8va
3
Pno.
mp
ff
mf
Vln.
Vla.
Vc.
3
mf
3
3
3
mp
3
Max.
Tech.


7
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.
ff
p
f
fff
mp
3
7
8


9
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.
f
p
mp
ff
3
8va
Ped.
9
10
11

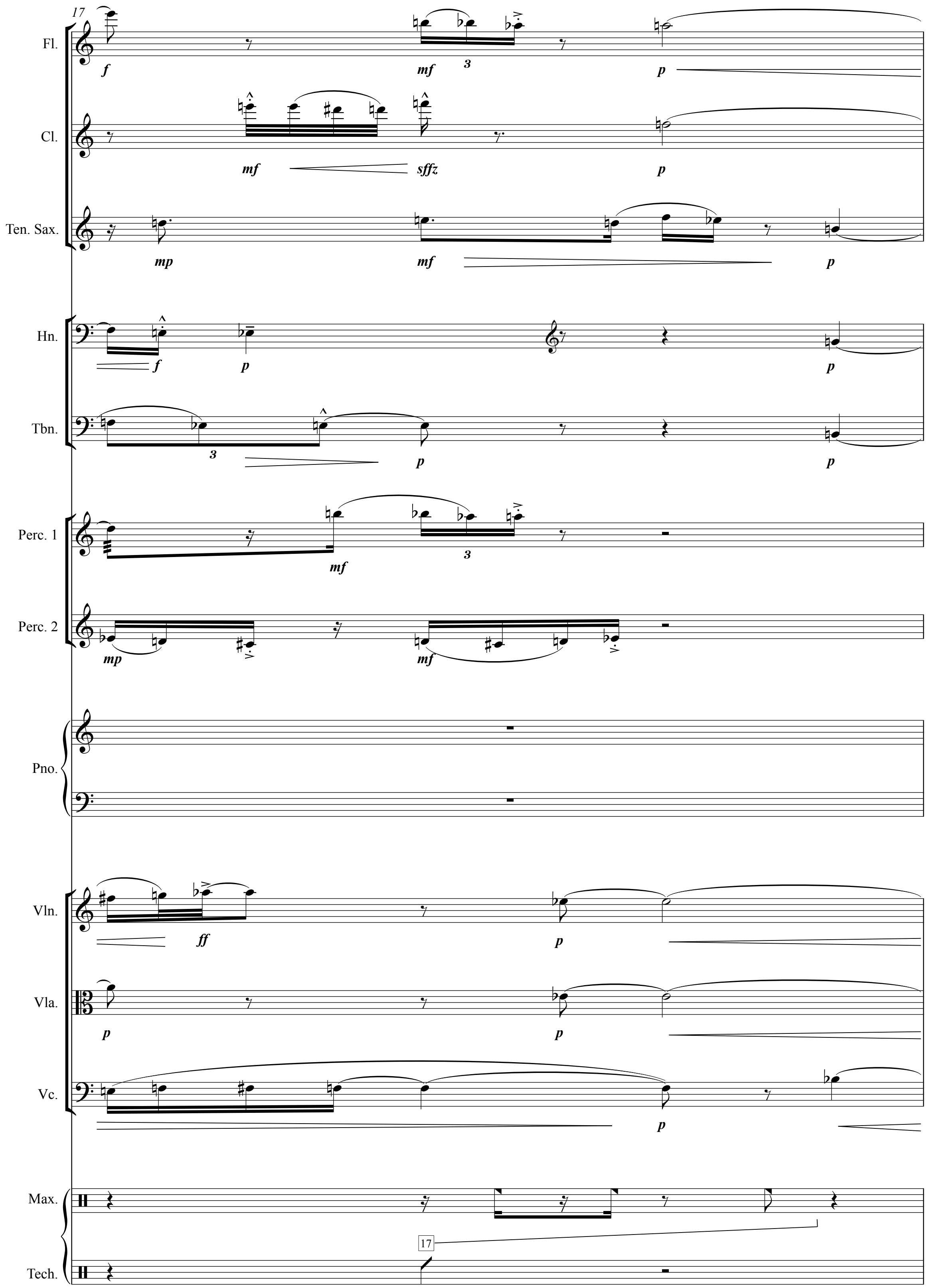

11
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.
random key clicks
air only
fltr
Ricochet
on the bridge
mute strings w/ L.H.
bow on tailpiece
mf
p
pp
12

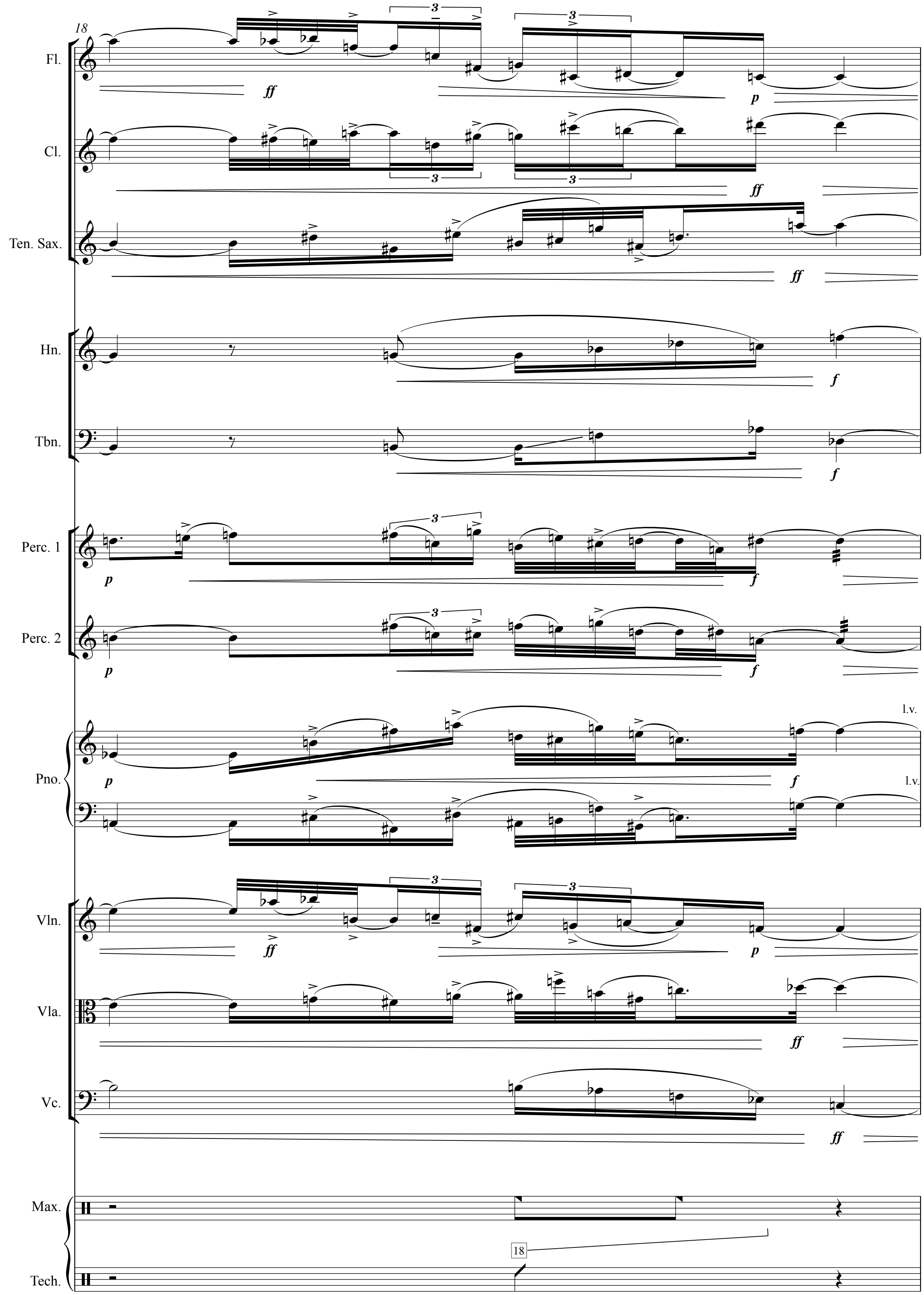

13
Fl.
f
p
Cl.
p
Ten. Sax.
p
Hn.
p
Tbn.
p
Perc. 1
mf
Perc. 2
p
Pno.
p
Vln.
Vla.
Vc.
Max.
Tech.

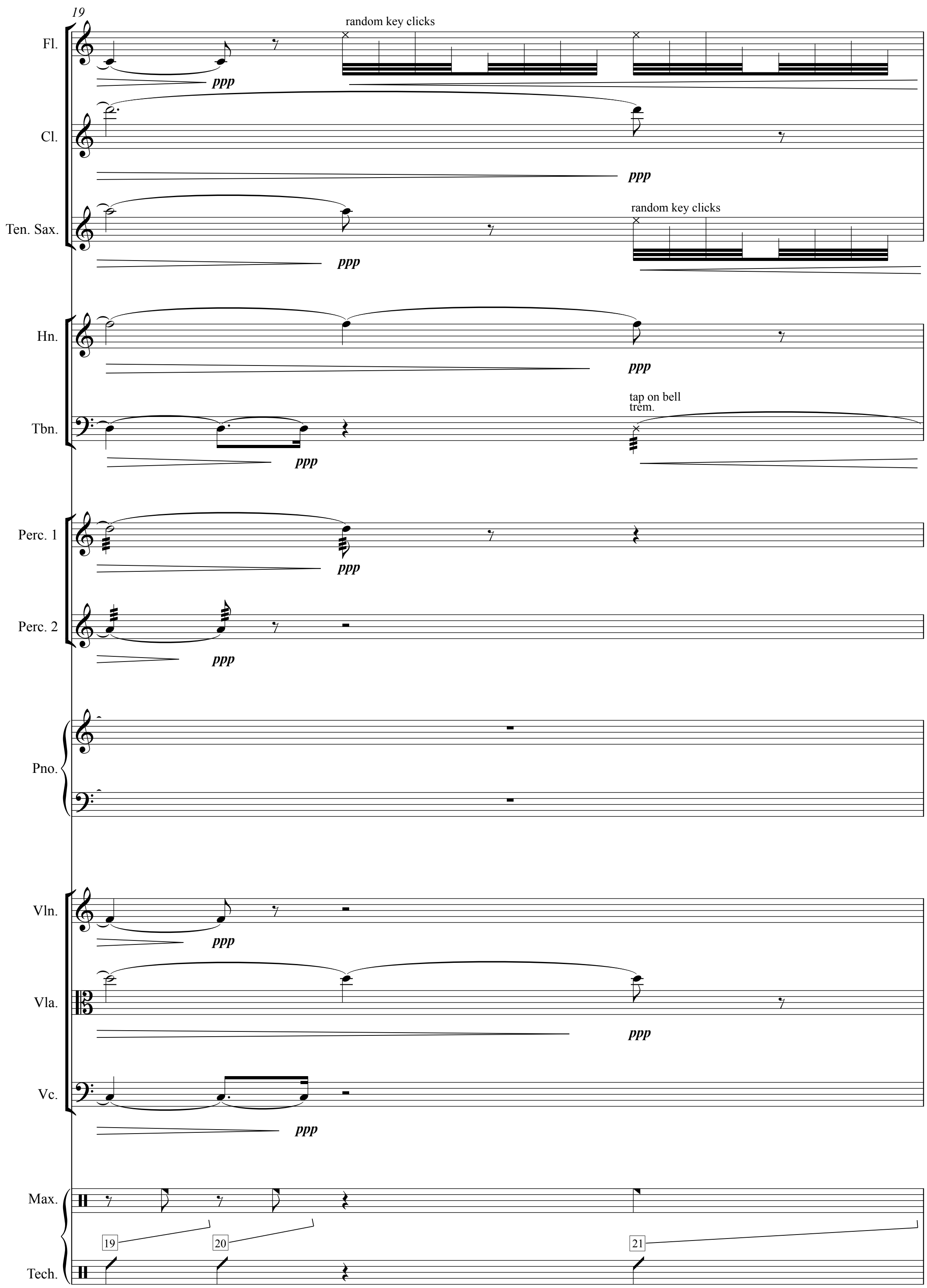

14
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
13
Tech.
p
f
mp
ff
mf
3


15
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.
14


16
Fl.
ff
fp
Cl.
3
ff
pp
Ten. Sax.
p
f
mf
Hn.
3
p
mp
Tbn.
pp
mf
Perc. 1
3
p
Perc. 2
f
Pno.
Vln.
ff
mf
Vla.
Vc.
Max.
15
16
Tech.


17
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.
f
mf
p
mf
sffz
mp
ff


18
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
l.v.
l.v.
Vln.
Vla.
Vc.
Max.
18
Tech.


19
Fl.
random key clicks
ppp
Cl.
ppp
Ten. Sax.
random key clicks
ppp
Hn.
ppp
Tbn.
tap on bell
trem.
ppp
Perc. 1
ppp
Perc. 2
ppp
Pno.
Vln.
ppp
Vla.
ppp
Vc.
ppp
Max.
19
20
21
Tech.


20
Fl.
Cl.
random key clicks
Ten. Sax.
Hn.
tap on bell
trem.
air only
fltr valves
fltr
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
on the bridge
mute strings w/ L.H.
Vla.
Vc.
Max.
22
23
Tech.


22
Fl.
f
3
Cl.
Ten. Sax.
Hn.
ppp
3
mf
Tbn.
ppp
Perc. 1
Perc. 2
Pno.
Vln.
pp
Vla.
Vc.
mf
mp
Max.
Tech.


24
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.
p
mf
f
8va
Ped.

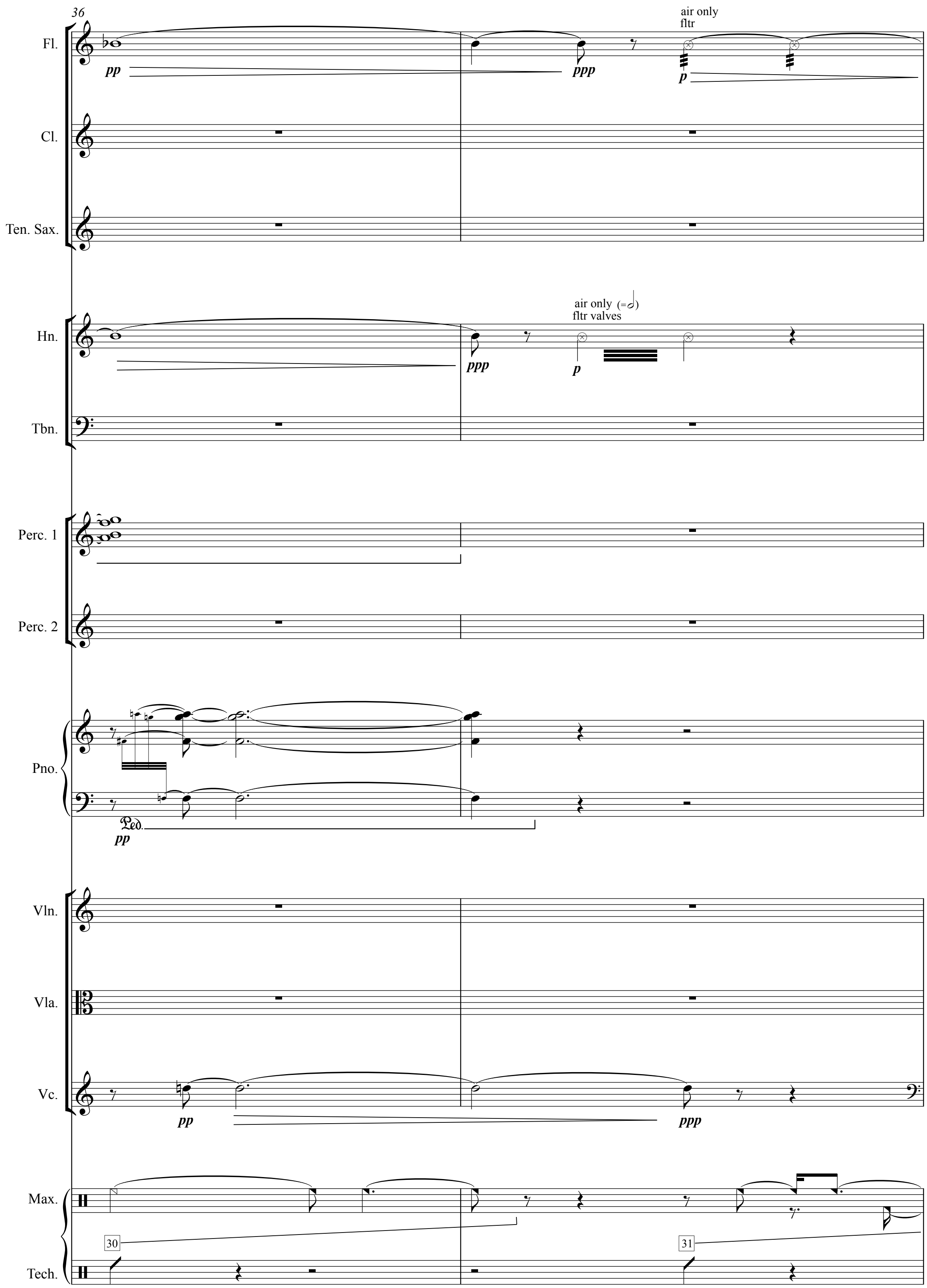

26
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.
f
mp
p
pp
ff
l.v.
mf
8va
Ped.
ppp


28
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.
3
mp
pp
p
ppp
mp
8va
mf
p
mp
f
24


30
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.
f
p
mf
pp
l.v.
25
26


32
Fl.
f
p
3
p
Cl.
Ten. Sax.
Hn.
+
mp
pp
Tbn.
Perc. 1
Ped.
mp
mf
Perc. 2
8va
Pno.
Ped.
mp
3
3
Vln.
Vla.
Vc.
3
mp
pp
Max.
27
Tech.

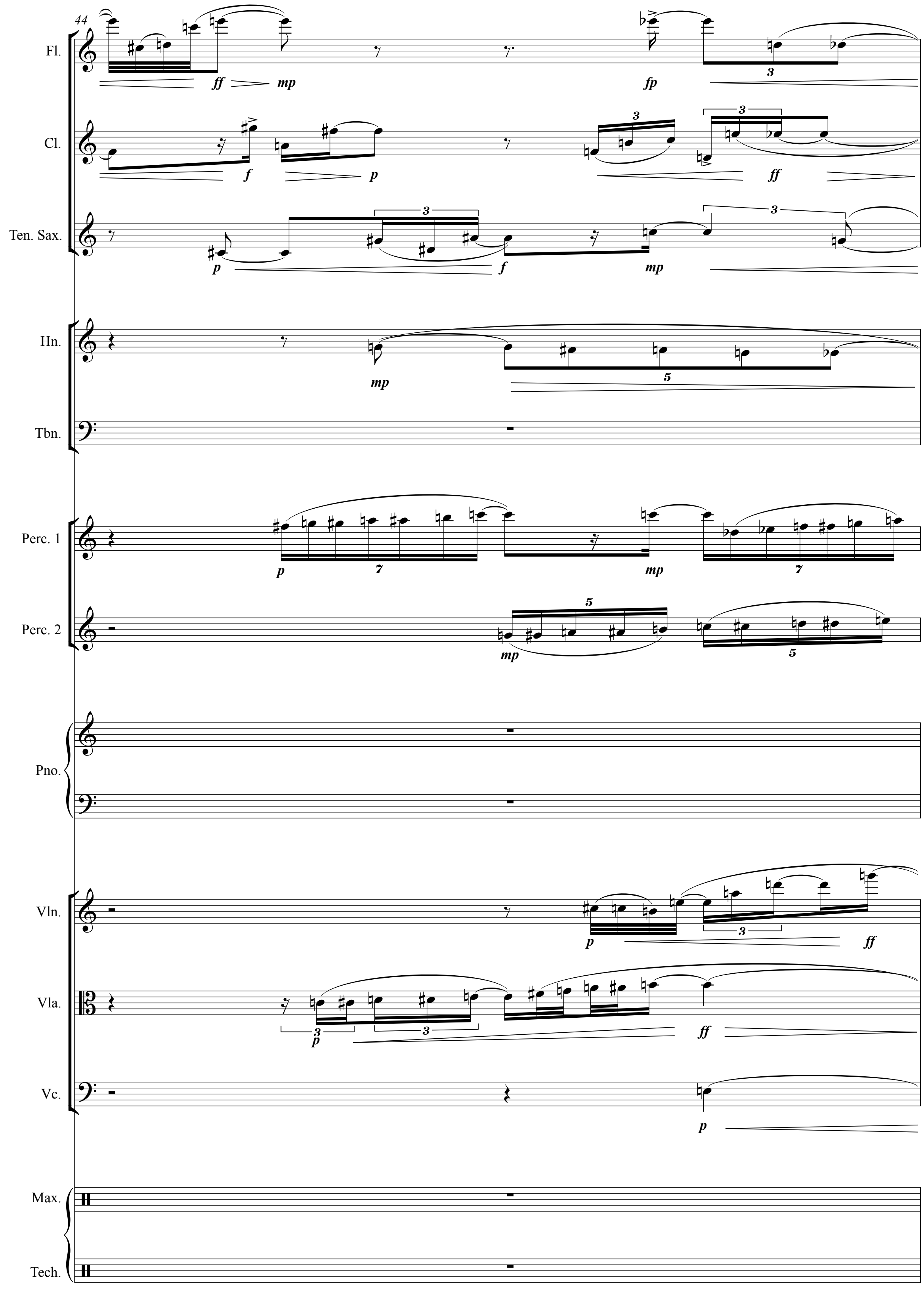

34
Fl.
mf
p
mf
pp
Cl.
Ten. Sax.
Hn.
pp
mf
p
pp
Tbn.
Perc. 1
p
Ped.
Perc. 2
Pno.
mf
p
Vln.
Vla.
Vc.
mf
p
Max.
28
29
Tech.

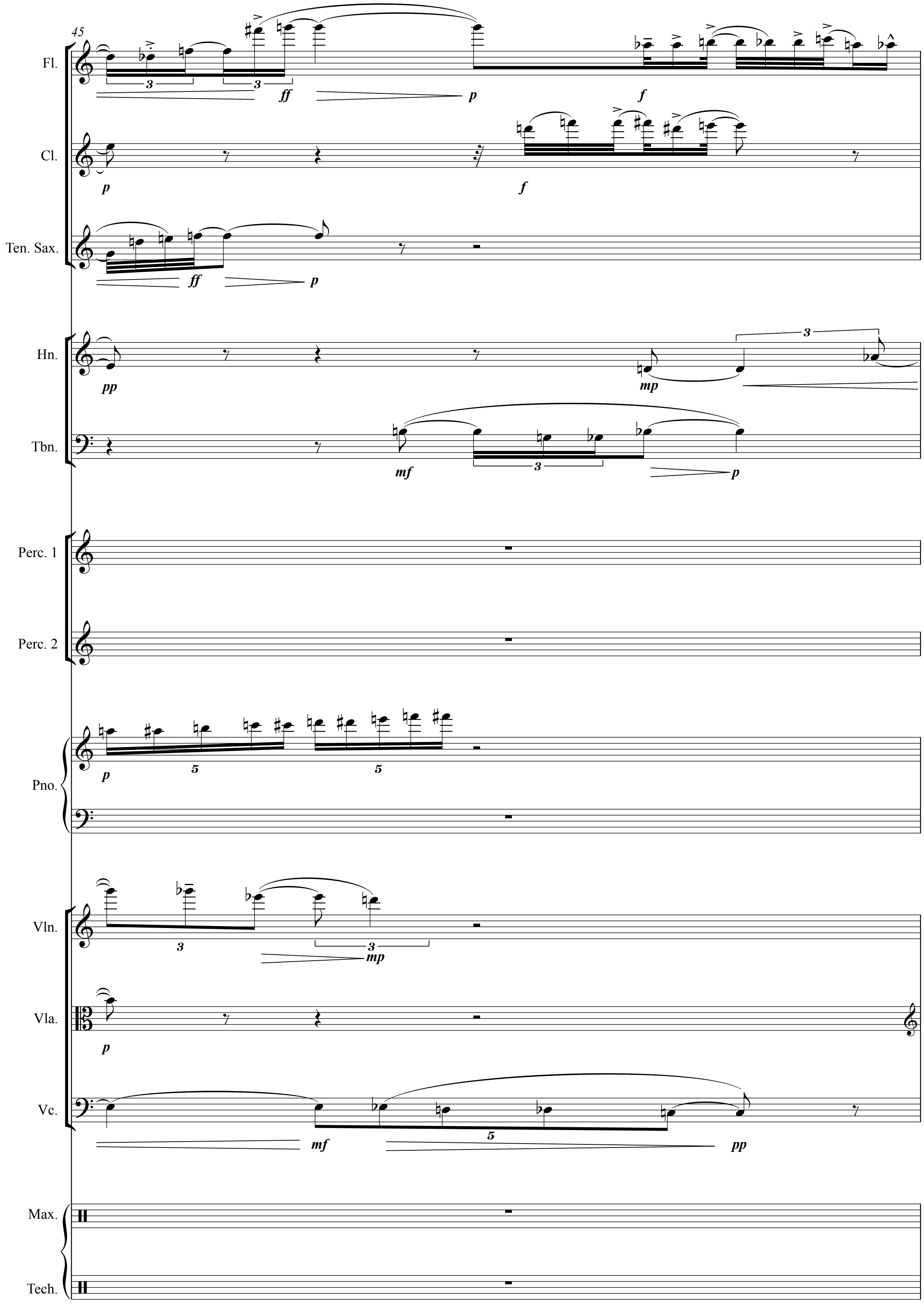

36
air only
fltr
Fl.
pp
ppp
p
Cl.
Ten. Sax.
air only
fltr valves
Hn.
ppp
p
Tbn.
Perc. 1
Perc. 2
Pno.
Ped.
pp
Vln.
Vla.
Vc.
pp
ppp
Max.
30
31
Tech.

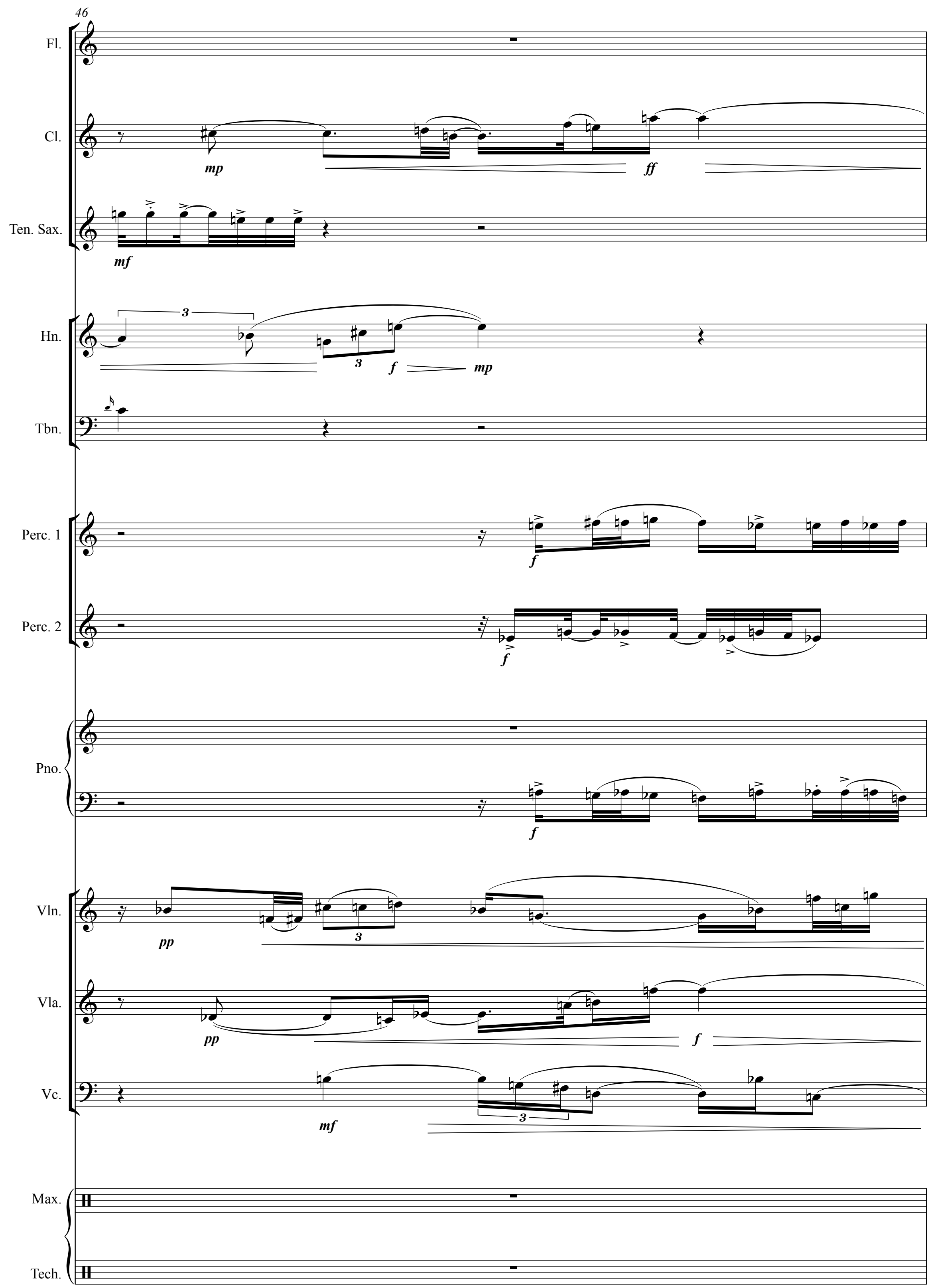

38
Fl.
random key clicks
ppp
p
random key clicks
air only
pp
p
Cl.
Ten. Sax.
tap on bell
trem.
Hn.
p
pp
air only
fltr
p
ppp
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Ricochet
on the bridge
mute strings w/ L.H.
Vc.
pp
on the bridge
mute strings w/ L.H.
pp
p
pp
Max.
3
3
32
33
34
35
Tech.


40
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.


42
Fl.
mp
Cl.
pp
Ten. Sax.
Hn.
mp
3
f
p
Tbn.
Perc. 1
Perc. 2
8va
8va
Pno.
p
p
f
mf
Vln.
Vla.
mp
pp
Vc.
3
p
Max.
Tech.


44
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.
ff
mp
fp
f
p
mp
mp
p

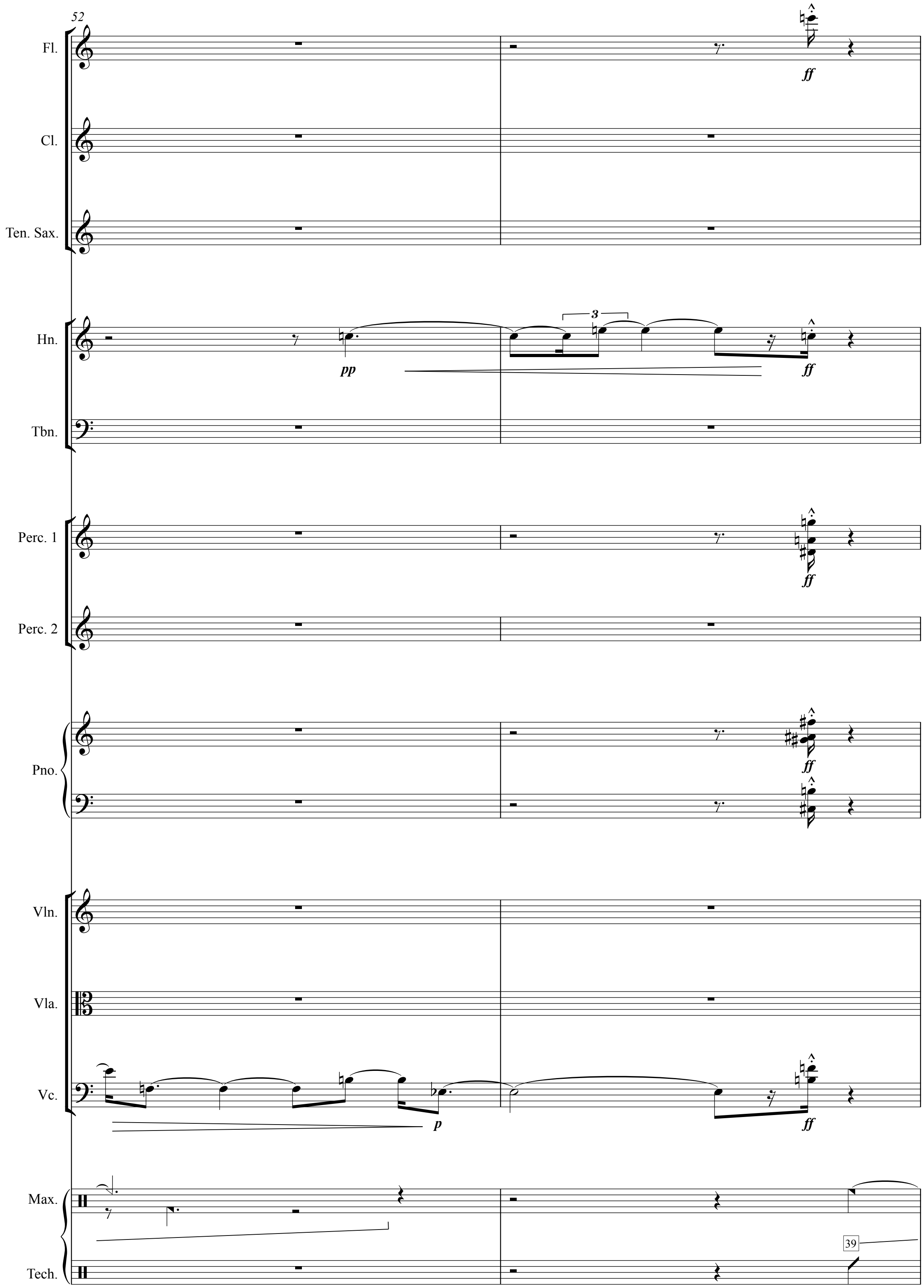

45
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.


46
Fl.
Cl.
mp
ff
Ten. Sax.
mf
Hn.
3
3
f
mp
Tbn.
Perc. 1
f
Perc. 2
f
Pno.
f
Vln.
pp
3
Vla.
pp
f
Vc.
mf
3
Max.
Tech.


47
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.
f
p
ff
mf
pp
3


Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.

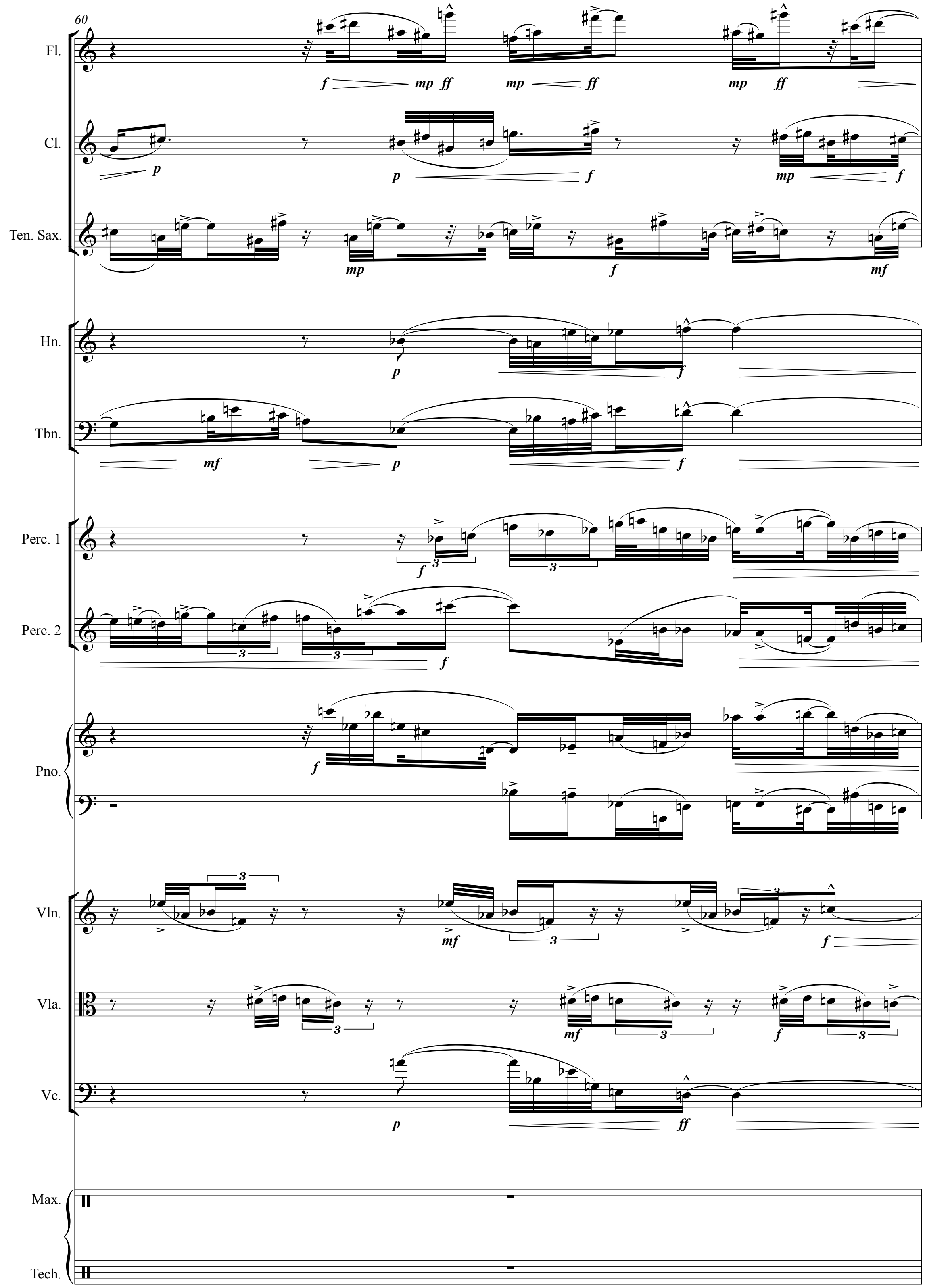

50
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.
ff
mp
p
mf
f
37
38

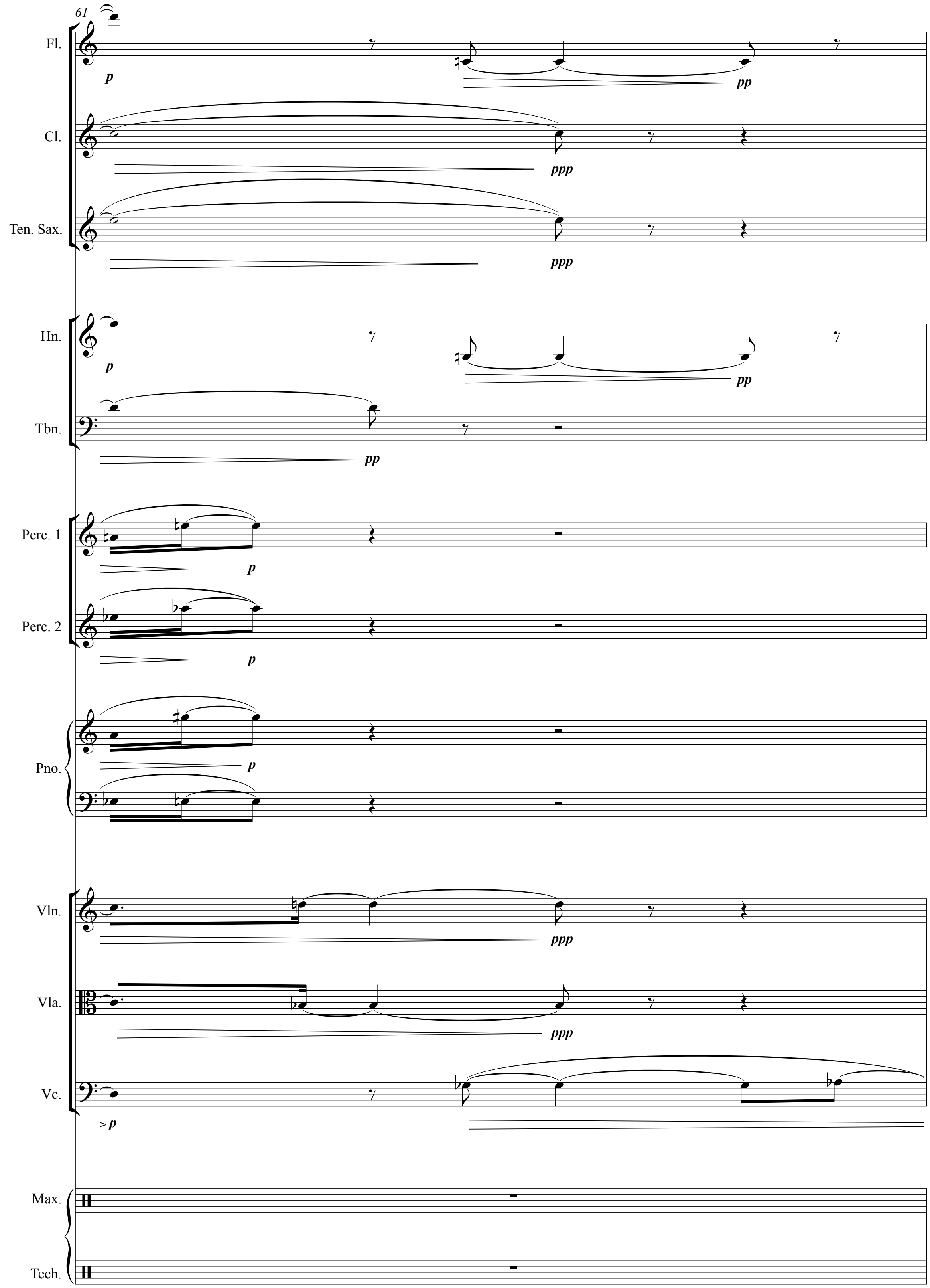

52
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.
ff
pp
3
p
39


54
Fl.
random key clicks
air only
fltr
Cl.
Ten. Sax.
Hn.
tap on bell
trem.
air only
fltr
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Ricochet
on the bridge
mute strings w/ L.H.
bow on tailpiece
Max.
40
Tech.


56
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.
p
fp
f
p
f
mf
p
41


58
Fl.
ff
3
p
Cl.
3
mp
f
Ten. Sax.
f
Hn.
Tbn.
p
Perc. 1
Perc. 2
p
Pno.
mp
3
ff
Vln.
3
f
sffz
mp
Vla.
f
3
sffz
mp
Vc.
Max.
Tech.


60
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.


61
Fl.
p
pp
Cl.
ppp
Ten. Sax.
ppp
Hn.
p
pp
Tbn.
pp
Perc. 1
p
Perc. 2
p
Pno.
p
Vln.
ppp
Vla.
ppp
Vc.
p
Max.
Tech.


62
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.
tr
f
mp
p
mf
pp
8va
pizz.
3
42


64
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
arco
Vln.
arco
Vla.
arco
Vc.
Max.
43
Tech.


66
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.
44


68
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.
fff
mp
p
f
ff
pp
pizz.
arco
45
46


70
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.
ff
ppp
mp
p
8va
47

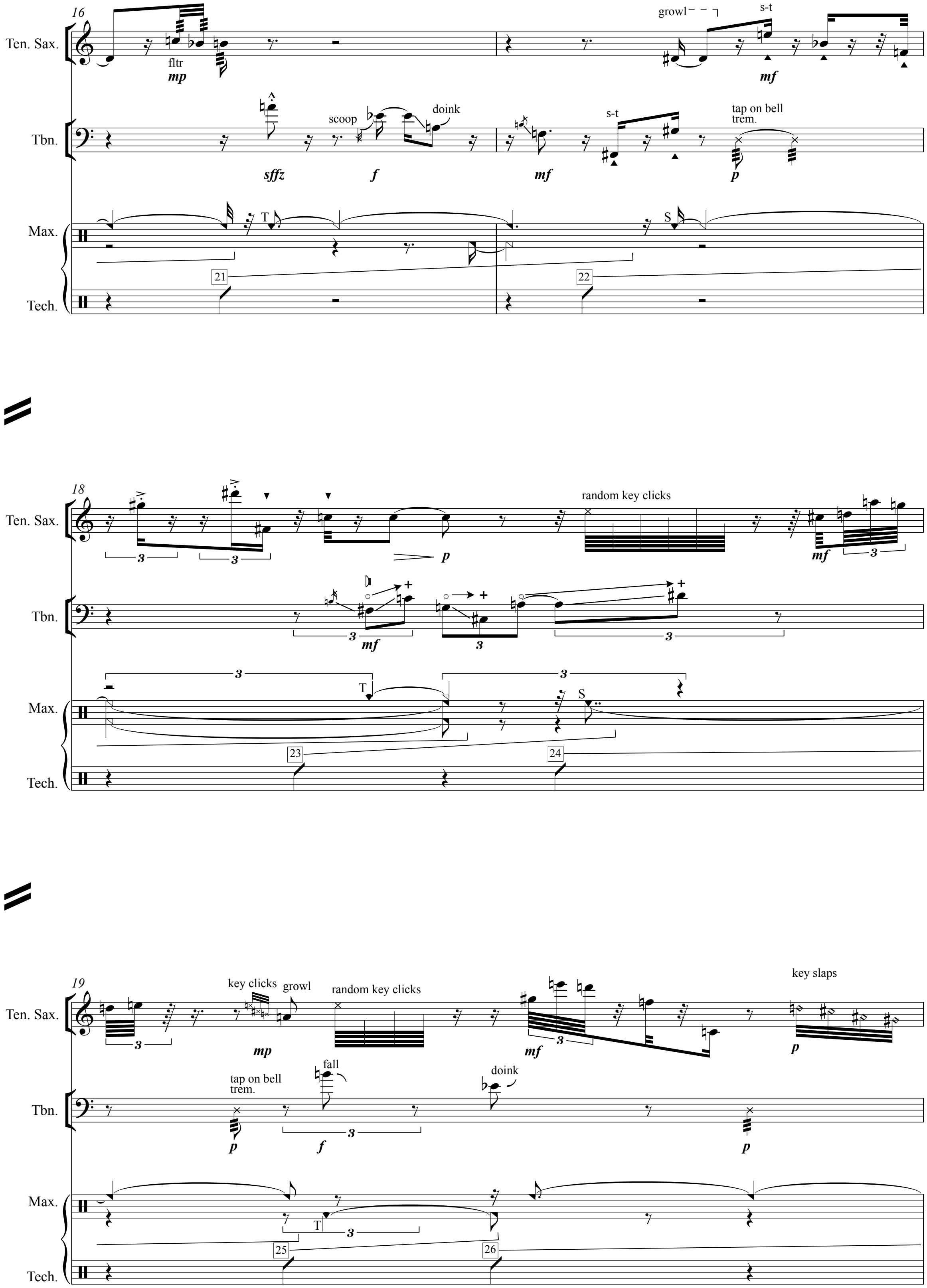

72
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
(8)
Pno.
Vln.
Vla.
Vc.
Max.
Tech.

Part IV


= 60
Fl.
Cl.
to mic 1
air only
fltr
random key clicks
air only
inhale
random key clicks
Ten. Sax.
Hn.
to mic 2
air only
fltr
tap on bell
kiss
tap on bell
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.


Ten. Sax.
Tbn.
Max.
Tech.
random key clicks
air only
air only
fltr
change to sound
tap on bell
trem.
play
sing
shake
fltr
sing
play
fltr
change to air
fltr
change to air
fltr
S
T
6
7
8
9
10
11
12
13
14


10
growl
fltr
"ghost" tongue
fltr
Ten. Sax.
Tbn.
s-t
growl
p
mf
Max.
S
T
15
16
17
Tech.
12
s-t
growl
p
play
sing
random key slaps
"ghost" tongue
air only
fltr
18
19
14
fltr
f
mf
20


16
Ten. Sax.
fltr
mp
growl
s-t
mf
Tbn.
scoop
doink
sffz
f
mf
s-t
tap on bell
trem.
p
Max.
T
S
21
22
Tech.
18
random key clicks
p
mf
3
T
S
23
24
19
key clicks
growl
random key clicks
key slaps
fall
doink
tap on bell
trem.
25
26


20
Ten. Sax.
mf
sffz
Tbn.
3
change to air
f
p
Max.
T
Tech.
21
random key slaps
key clicks
mp
pp
> ppp
mf
S
27
28
22
ff
p
fff
mp
(=♩)
29

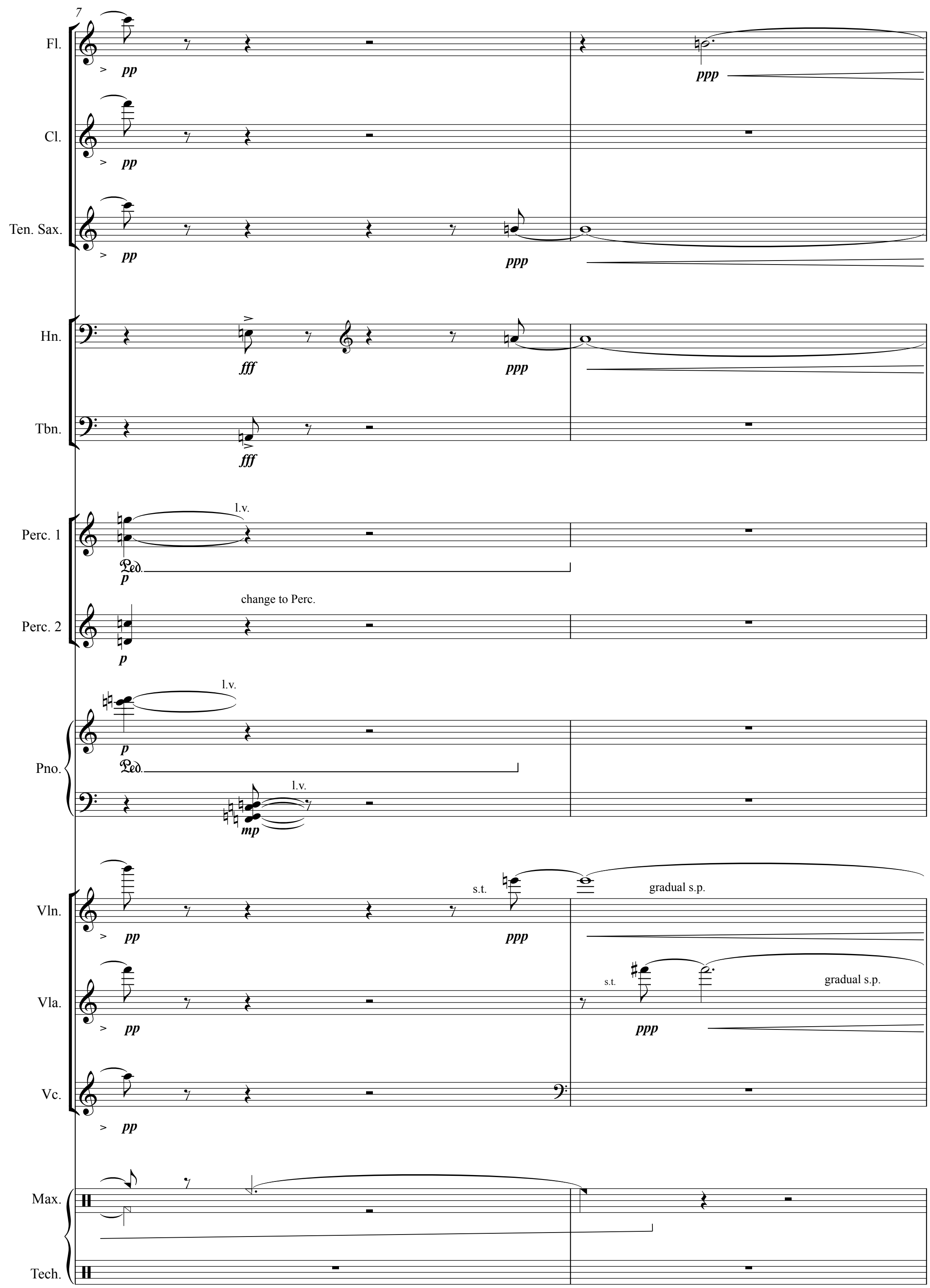

23
Ten. Sax.
growl
f
growl
p
Tbn.
irregular plunger change
p
mf
p
f
Max.
T
Tech.
30
24
Ten. Sax.
f
p
Tbn.
sfz
mp
pp
f
Max.
S
T
Tech.
31
25
Ten. Sax.
mp
change to air
p
ppp
Tbn.
change to air
p
ppp
Max.
T
S
32
Tech.


27
air only
fltr
random key clicks
air only
Ten. Sax.
p
air only
fltr
(=♩)
tap on bell
kiss
Tbn.
p
Max.
Tech.
29
inhale
random key clicks
random key clicks
air only
Ten. Sax.
tap on bell
tap on bell
trem.
Tbn.
Max.
Tech.
31
Ten. Sax.
pp
sing
play
air only
fltr
Tbn.
pp
S
T
S
T
33
34
Max.
Tech.


Part V
♩ = 89
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.
play
sing
fltr
s-t
Vib. + W.B.
Mar.
sul ponticello (s.p.)
normal


Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
sul tasto (s.t.)
no pressure
normal
add pressure
Max.
Tech.


5
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.
pizz.
arco
3


7
Fl.
pp
ppp
Cl.
pp
Ten. Sax.
pp
ppp
Hn.
fff
ppp
Tbn.
fff
Perc. 1
l.v.
p
Perc. 2
change to Perc.
p
Pno.
l.v.
p
l.v.
mp
Vln.
s.t.
gradual s.p.
pp
ppp
Vla.
s.t.
gradual s.p.
pp
ppp
Vc.
pp
Max.
Tech.


9
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
plunger
Perc. 1
l.v.
Perc. 2
Pno.
l.v.
Vln.
s.p.
normal
Vla.
s.p.
Vc.
Max.
Tech.


11
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.
s-t
Perc.
normal
8va
4
f
mp
ff
mf
pp
p


13
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.
f
mp
ff
fp
ppp
p
mf
fff
l.v.
Ped.
3
5
(8)


15
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.
random key clicks
random key clicks
random key clicks
tap on bell
tap on bell
W.B.
w/ fingertips on music stand
on the bridge
mute strings w/ L.H.
on the bridge
mute strings w/ L.H.
on the bridge
mute strings w/ L.H.
mf
pp
mp
ppp
p
5


17
Fl.
Cl.
Ten. Sax.
random key clicks
air only
fltr
mp
pp
p
Hn.
Tbn.
tap on bell
Perc. 1
Perc. 2
7
f
mp
mf
Pno.
3
Vln.
Vla.
Vc.
ricochet
on the bridge
mute strings w/ L.H.
6
Max.
Tech.


19
Fl.
Cl.
Ten. Sax.
Hn.
tap on bell
trem.
Tbn.
tap on bell
trem.
Perc. 1
Vib.
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.


21
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
6
Tech.

23
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.


25
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.
air only
fltr
tap on bell
W.B.
w/ fingertips on music stand
bow on tail piece
sustain
7


27
Fl.
"ghost" tongue
ppp
pp
f
Cl.
"ghost" tongue
mp
Ten. Sax.
"ghost" tongue
p
Hn.
air only
fltr
Tbn.
air only
fltr
Perc. 1
Vib.
Perc. 2
mf
Pno.
Vln.
Vla.
on the bridge
mute strings w/ L.H.
Vc.
on the bridge
mute strings w/ L.H.
Max.
Tech.


29
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.
fltr
tr
p
f
mf
mp
ff
pp
3
7


31
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.
random key clicks
tap on bell
W.B.
Vib.
w/ fingertips on music stand
ricochet
on the bridge
mute strings w/ L.H.


33
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.


35
Fl.
shake
random key slaps
Cl.
shake
random key slaps
Ten. Sax.
random key slaps
Hn.
shake
tap on bell
Tbn.
shake
tap on bell
Perc. 1
W.B.
Perc. 2
Pno.
tap on music stand
w/ fingertips
Vln.
ricochet
on the bridge
mute strings w/ L.H.
Vla.
ricochet
on the bridge
mute strings w/ L.H.
Vc.
ricochet
on the bridge
mute strings w/ L.H.
Max.
Tech.
9


37
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Vib.
Perc. 1
W.B.
7
Perc. 2
Pno.
tap on music stand
w/ fingertips
Vln.
Vla.
Vc.
Max.
Tech.


39
air only (=o)
Fl.
pp
f
air only (=o)
Cl.
pp
Ten. Sax.
f
p
f
air only (=o)
Hn.
pp
Tbn.
3
3
f
p
mf
pp
7
7
Perc. 1
Perc. 2
mp
mf
f
p
change to Mar.
Pno.
ppp
on the bridge
mute strings w/ L.H.
Vln.
pp
p
on the bridge
mute strings w/ L.H.
Vla.
pp
on the bridge
mute strings w/ L.H.
Vc.
pp
Max.
10
Tech.


41
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.


43
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Vib.
Perc. 1
Mar.
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
11
Tech.


45
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.
fp
mp
f
p
pp
l.v.
s.t.
3


47
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.
fltr
fltr
tr
gradual s.p.
gradual s.p.
normal
s.p.
gradual s.t.
normal
f
p
mf
f
pp
pp
mp
ff


49
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.
normal
s.p.
(tr)


51
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.
mp
ff
p
mf
ppp
s-t
fff
f
change to Perc.
pp
l.v.
pizz.
arco
s.p.
normal


53
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.
s-t
s-t
change to air
change to air
normal
mp
mp
p
mf
p
mf
pp
f
ppp
f
ppp
ff
ff
f
ppp
f
ppp
p
mf
p
mf


55
Fl.
random key clicks
p
Cl.
random key clicks
Ten. Sax.
random key clicks
Hn.
air only
fltr
ppp
mp
Tbn.
air only
fltr
W.B.
Perc. 1
7
Parc.
Perc. 2
mf
f
Pno.
ricochet
on the bridge
mute strings w/ L.H.
Vln.
6
Vla.
Vc.
Max.
12
Tech.


57
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.
random key slaps
"ghost" tongue
tap on bell
bow on tailpiece
sustain
on the bridge
mute strings w/ L.H.
mp
p
mf
f
ppp


59
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.
mp
fff
air only
fltr
pp
7
3
f
mf
change to Mar.


61
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.
play
sing


63
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Vib.
Perc. 1
Mar.
Perc. 2
Pno.
pizz.
arco
Vln.
Vla.
Vc.
Max.
Tech.


65
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.


67
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.
s-t
s.p.
normal
13
mp
ff
mf
p
f
fff


69
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.
s-t
normal
normal
ff
mp
mf
p
pp
f


71
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.
mf
ff
sffz
p
f
sfz
pp
5
3


73
Fl.
ff
p
Cl.
ff
p
Ten. Sax.
ff
p
Hn.
Tbn.
Perc. 1
Perc. 2
tr
Pno.
pp
3
ff
mf
Vln.
pp
3
ff
mf
Vla.
pp
3
ff
mf
Vc.
Max.
Tech.


75
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.
tr
pizz.
p
f
mp
mf
3
5


77
Fl.
fff
mp
Cl.
fff
mp
Ten. Sax.
fff
mp
Hn.
Tbn.
Perc. 1
p
f
Perc. 2
3
pp
Pno.
p
f
Ped.
arco
Vln.
pp
ff
f
arco
Vla.
pp
ff
f
arco
Vc.
pp
ff
f
Max.
14
Tech.


79
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.
fff
mf
ff
mp
pp
f
p
tr
pizz.
15
16


81
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.
(tr)
tr
ff
p
f
arco
mp
8va


83
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.
p
ff
pp
fff
sul E
sul D
sul G
sul A
8va
17
18


86
Fl.
Cl.
Ten. Sax.
Hn.
Tbn.
Perc. 1
Perc. 2
Pno.
Vln.
Vla.
Vc.
Max.
Tech.
fltr
19
20


89
Fl.
ppp
Cl.
ppp
Ten. Sax.
ppp
Hn.
ppp
Tbn.
ppp
Perc. 1
ppp
Perc. 2
ppp
Pno.
ppp
Vln.
pizz.
ppp
Vla.
pizz.
ppp
Vc.
pizz.
ppp
bend
pull-off
sul C
l.v.
Max.
Tech.
21

CHAPTER 7

CONCLUSION

Composers of electronic music commonly use in the subtitle of their compositions terms like fixed media, computer-generated sound and interactive electronics. These terms denote the manner in which electronics are used in the composition. At the same time, a subtitle such as “for chamber ensemble and electronics” can imply a duality of two equal entities: the ensemble vs. the electronics. In the subtitle of *Normal Mode* I do not suggest such a duality. Instead, I perceive the electronics of this composition as one-twelfth of the ensemble. Nevertheless, whereas the rest of the instruments in the ensemble are standard and familiar instruments, the electronics of *Normal Mode* constitute a new instrument. This instrument was developed specifically for this composition and its existence may not extend beyond this composition. A discussion of the properties of this instrument and its relationships with the rest of the ensemble is, therefore, valuable for the appreciation of this composition. Like the “new kid on the block,” the electronics initially attract more attention. Over time a new equilibrium is established and the electronics part is integrated in the ensemble.

In compositions for acoustic instruments and electronics, the relationships between these two elements are often defined in the timbral domain. While the timbres of acoustic instruments are generally fixed, the electronics may replicate these timbres, process and alter these timbres, or suggest an entirely different set of timbres. In addition, these relationships can be defined by the gestures, the textures, and the temporal and pitch organizations that characterize the electronics and the instruments. In *Normal Mode*, the relationships between the ensemble and the electronics evolve throughout the movements of composition. Each movement suggests a different perspective of these relationships.

The interactive movements, Part II and Part IV, lean towards a fusion between the instrumental and the electronic sounds, although, this fusion involves only a few instruments from the ensemble. Part II serves as a stepping stone to Part IV. In Part II, the clarinet is only partially affected by the electronics. In Part IV, the acoustic sounds of the tenor saxophone and the trombone are almost completely absorbed in the electronics. In the other movements, the relationships between the electronics and the ensemble vary on a scale between conflict and agreement.

The electronics in Part I interject abrupt statements within the ensemble discourse. The harsh and distorted electronic sounds stand in contrast to the timbres of the instruments and introduce tension and instability; however, this somewhat chaotic exchange supports the energetic flow of the movement. The tension is released, for a brief moment, towards two-thirds of the movement, as the altercation becomes more organized. In Part III, the electronics are softened and smoothed out to match the light sound and fast-pace motion of the quintet that dominates this movement, and includes the flute, the horn, the vibraphone, the piano and the cello. Aggressive interruptions are introduced in this movement by *tutti* sections of agitated rhythms. Part V displays a more organized discourse between the ensemble and the electronics. At times, the ensemble complements the timbre of the electronics with the use of extended techniques. The electronics articulate in this movement their most composed elongated statements in the piece.

The electronics of *Normal Mode* are almost entirely self-contained within the digital domain. Their primary sound sources are digital oscillators. They incorporate a sound library which was created through sound synthesis with these digital oscillators.

They are realized through real-time sound editing and processing, also within a digital environment. These electronics become real acoustic sounds only in the final stage of this process when the digital signal is converted to an analog signal and, in turn, to an air pressure wave. Nevertheless, they have strong ties with the acoustic world of sound. To begin with, the methods of sound synthesis, which generated these new timbres, are based on the understanding of the properties of acoustic sound. In addition, the compositional system of *Normal Mode*, as described in Chapter 5, ties the electronics to the pitch organization of the ensemble material. And finally, the interactive movements bring acoustic instrumental sounds into the electronics.

My choice of using digital oscillators as a primary sound source and to synthesize sounds with these oscillators is rooted in the compositional system and its reference to the microtonality of Turkish music. Methods of sound synthesis, such as additive synthesis and frequency modulation, enable the conversion of a numerical list of frequencies into an audible sound. The additive synthesis, in particular, also allows the precise control over each one of the spectral components. My additive synthesis patchers, which are discussed in Chapter 3, initially served as a research tool that helped me to explore the possibilities of combining microtonality in an electronic composition and consequently to establish my compositional system. The musical problem of additive synthesis is that it tends to generate sounds which are not rich enough to stand on their own.

In *Normal Mode* I have compensated for this problem using several means. I first expanded my selection of synthesis patchers to also include patchers of frequency modulation and modified frequency modulation, as discussed in Chapter 3. I combined these patchers in the synthesis master patchers and mixed their signals into more complex sounds. I enhanced these sounds by shading them with samples of acoustic noise-type sounds. I then used the synthesis master patchers to generate the sound library of the composition. In addition, I developed an intricate process through which the performance patchers recompose the electronics part in real time. These steps, which were controlled by my compositional system, increased the depth of my electronic sounds and the intensity of the electronics part.

My interest in non-standard tunings and octave divisions, which began with *Normal Mode*, extends beyond this composition. As mentioned in the introduction, sound synthesis is an efficient tool for composing electronics that incorporate microtonaliy. In future compositions, I will attempt to expand the sound synthesis to include methods that can more efficiently generate complex sounds. One such method is the technique of granular synthesis that allows the construction of composite sounds from a large amount of very short sonic grains. I will also consider using, in future compositions, applications, such as IRCAM's AudioSculpt and Kyma X, that provide the tools for spectral analysis and modification of acoustic sounds. These applications allow the extraction of the spectral components of these sounds. If the spectral components are of periodic instrumental sounds, they can be retuned to fit a microtonal system of organization similar to the one that I have used in *Normal Mode*. At the same time, such components include non-periodic or noise elements that do not exist in electronically generated pure sine waves, and, therefore, they are richer building blocks for the construction of new sounds.

The limitations of the synthesis process have led me, as previously mentioned, to develop the process through which the electronics part is realized in real-time by the performance patchers during a performance of the piece. This complex and multilayered process is tied through the classifications of sounds to the compositional system. It redefines in real time every small detail that constitutes the sound events which are generated by the performance patchers. And it incorporates random selection in many of its layers. As such, this process turns the electronics of *Normal Mode* into a dynamic entity that evolves with each performance of the piece.

Real-time sound processing, randomness and the interaction between electronics and live performers will most likely continue to interest me in future compositions. I perceive computer applications, such as Max/MSP, which incorporate these capabilities, as excellent tools for composing dynamic and progressive electronics. In *Normal Mode* only the electronics incorporate randomness and indeterminism. The ensemble material is fixed. It is my intention to experiment in future compositions with the incorporation of indeterminism and improvisation also in the material of the live performers. I believe that such elements, when combined with a detailed compositional system, can enhance the composition and invite more involved performances.

BIBLIOGRAPHY

Anderson, Julian. "Spectral music." In *Grove Music Online*. *Oxford Music Online*, http://www.oxfordmusiconline.com/subscriber/article/grove/music/50982 (accessed February 24, 2010).

Assayag, Gerard, Camilo Rueda, Mikael Laurson, Carlos Agon, and Olivier Delerue. "Computer-Assisted Composition at IRCAM: From PatchWork to OpenMusic." *Computer Music Journal* 23, no. 3 (Autumn 1999): 59-72.

Carlos, Wendy. "Tuning: At the Crossroads." *Computer Music Journal* 11, no. 1 (Spring 1987): 29-43.

Chadabe, Joel. *Electronic Sound: the Past and Promise of Electronic Music*. New Jersey: Prentice-Hall Inc., 1997.

Chowning, John. "The Synthesis of Complex Audio Spectra by Means of Frequency Modulation." *Computer Music Journal* 1, no. 2 (April 1977): 46-54.

Cochrane, Lalage. "Pythagorean intonation." In *The Oxford Companion to Music*. Edited by Alison Latham. *Oxford Music Online*, http://www.oxfordmusiconline.com/ subscriber/article/opr/t114 /e5424 (accessed December 9, 2008).

Dodge, Charles, and Thomas A. Jerse. *Computer Music: Synthesis, Composition and Performance*. New York: Schirmer Books, 1997.

Garfias, Robert. "Survivals of Turkish Characteristics in Romanian *Musica Lautareasca*." *Yearbook for Traditional Music* 13 (1981): 97-107.

Griffiths, Paul. *Modern Music and After: Directions Since 1945*. Oxford: Oxford University Press, 1995.

Griffiths, Paul, Mark Lindley, and Ioannis Zannos. "Microtone." In *Grove Music Online*. *Oxford Music Online*, http://www.oxfordmusiconline.com/subscriber/article/grove/ music/18616 (accessed January 7, 2010).

Heinemann, Stephen. "Pitch-Class Sets Multiplication in Theory and Practice." *Music Theory Spectrum* 20, no. 1 (Spring 1998): 72-96.

Helmholtz, Hermann L. F. *On the Sensations of Tone as a Physiological Basis for the Theory of Music*. Translated by Alexander J. Ellis. New York: Dover Publications, Inc., 1954.

Keislar, Douglas. "History and Principles of Microtonal Keyboards." *Computer Music Journal* 11, no. 1 (Spring 1987): 18-28.

———. “Introduction.” *Perspectives of New Music* 29, No. 1 (Winter 1991): 173-175.

Keislar, Douglas, Easley Blackwood, John Eaton, Lou Harrison, Ben Johnston, Joel Mandelbaum, and William Schottstaedt. “Six American Composers on Nonstandard Tunings.” *Perspectives of New Music* 29, no. 1 (Winter 1991): 176-211.

Kimura, Mari. “Creative process and performance practice of interactive computer music: A performer's tale.” *Organised Sound: An International Journal of Music Technology* 8, no. 3 (December 2003): 289-296. http://iimp.chadwyck.com/articles/displayItemPDF.do?format=PAGE&PQID=1404967441&journalID=JID13557718&royaltiesid=LOUJID13557718&product=iimp&articleID=iimp00336104 (accessed January 1, 2010).

Kolbyakov, Lev. *Pierre Boulez: A World of Harmony*. New York: Harwood Academic Publishers, 1990.

Lindley, Mark. "Pythagorean intonation." In *Grove Music Online*. *Oxford Music Online*, http://www.oxfordmusiconline.com/subscriber/article/opr/t114/e5424 (accessed December 9, 2008).

Manuel, Peter. “Modal Harmony in Andalusian, Eastern European, and Turkish Syncretic Music.” *Yearbook for Traditional Music* 21 (1989): 70-94.

Maqam World. “Arabic Rhythms.” Maqam World. http://www.maqamworld.com/rhythms.html (accessed December 9, 2008).

Moorer, James A., and John Grey. “Lexicon of Analyzed Tones. Part I: A Violin Tone.” *Computer Music Journal* 1, no. 2 (April 1977): 39-45.

Moorer, James A., and John Grey. “Lexicon of Analyzed Tones. Part 2: Clarinet and Oboe Tones.” *Computer Music Journal* 1, no. 3 (June 1977): 12-29.

Murail, Tristan. “The Revolution of Complex Sounds.” *Contemporary Music Review* 24, no. 2/3 (April/June, 2005): 121-135. http://web.ebscohost.com/ehost/pdf?vid=3&hid=11&sid=dd148275-3368-451d-be50-747b7637e641%40sessionmgr12 (accessed October 13, 2009).

Polansky, Larry. “Paratactical Tuning: An Agenda for the Use of Computers in Experimental Intonation.” *Computer Music Journal* 11, no. 1 (Spring 1987): 61-68.

Puckette, Miller. “Max at Seventeen.” *Computer Music Journal* 26, no. 4 (Winter 2002): 31-43.

Risset, Jean-Claude. “Computer Music Experiments 1964 - ...” *Computer Music Journal* 9, no. 1 (Spring 1985): 11-18.

Rothstein, Joseph B. "Products of Interest." *Computer Music Journal* 22, no. 3 (Autumn 1998): 75-81.

Sethares, William A. *Tuning, Timbre, Spectrum, Scale*. London: Springer-Verlag, 2005. http://www.springerlink.com/content/w04151 (accessed October 15, 2009).

Sethares, William A., Andrew J. Milne, Stefan Tiedje, Anthony Prechtl, and James Plamondon. "Spectral Tools for Dynamic Tonality and Audio Morphing." *Computer Music Journal* 33, no. 2 (Summer, 2009): 71-84. http://muse.jhu.edu/journals/ computer_music_journal/ v033/33.2.sethares.pdf (accessed October 10, 2009).

Signell, Karl L. *Makam: Modal Practice in Turkish Music*. Seattle: Asian Music Publications, 1977.

Wishart, Trevor. *On Sonic Art*. Edited by Simon Emmerson. London: Routledge, 2002.