SNOW SPELL: AN INTERACTIVE COMPOSITION FOR ERHU, FLUTE, PIANO, CELLO AND MAX/MSP

Chien-Wen Cheng, B.A., M.M.

Dissertation Prepared for the Degree of

DOCTOR OF MUSICAL ARTS

UNIVERSITY OF NORTH TEXAS

August 2007

APPROVED:

Andrew May, Major Professor
Stephen Slottow, Minor Professor
Cindy McTee, Committee Member
Joseph Klein, Chair, Division of Composition Studies, College of Music
Graham Phipps, Director of Graduate Studies
James C. Scott, Dean of the College of Music
Sandra L. Terrell, Dean of the Robert B. Toulouse School of Graduate Studies

Cheng, Chien-Wen, *Snow Spell*: An Interactive Composition for Erhu, Flute, Piano, Cello, and Max/MSP. Doctor of Musical Arts (Composition), August 2007, 134 pp., 22 illustrations, bibliography, 33 titles.

Snow Spell is an interactive composition for erhu, flute, cello, piano, and Max/MSP interactive computer music system. This one-movement piece, *Snow Spell*, is intended to depict the beauty of a snow scene by presenting four different impressions of snow envisioned by the composer through music. The definition, history, and significance of interactive music are explored. Various modes of interactivity to control signal processing modules, and technical considerations for signal routing and level control in the interactive computer music system are also explored. Chinese music elements in *Snow Spell* including pentatonic scales, glissandi, and quotations from the Chinese folk tune *River of Sorrow* are investigated.

Copyright 2007

by

Chien-Wen Cheng

TABLE OF CONTENTS

Page

LIST OF ILLUSTRATIONS iv

PART I: CRITICAL ESSAY 1

Chapter

1. INTRODUCTION 2
2. CONCEPT AND HISTORY OF INTERACTIVE MUSIC 4
    - Definition
    - History
    - Significance
3. INTERACTIVITY BETWEEN PERFORMERS AND COMPUTER IN *SNOW SPELL* 14
    - Programming Structure
    - Interactivity and Compositional Algorithms
4. OTHTER TECHNICAL CONSIDERATIONS 37
    - Audio Signal Routing and Mixing
    - Spatialization in Stereo Environment
    - Efficiency Consideration in CPU Consumption
5. COMPOSITIONAL CONSIDERATIONS 53
    - Formal Structure of *Snow Spell*
    - Elements of Chinese Music in *Snow Spell*
6. CONCLUSION 66

BIBLIOGRAPHY 68

PART II: COMPOSITION 71

PERFORMANCE INSTRUCTIONS 72

COMPOSITION: *SNOW SPELL* 79

ILLUSTRATIONS

Figure Page

1. Programming Structure 16
2. Phrase Length Detector to Control Multi-voice Harmonizer 21
3. Harmonizer in *Snow Spell* 22
4. Pitch Detection Combined with Amplitude Analysis 23
5. Speed Limit Device 24
6. Pitch Range Sieve to Control Signal Processing Modules 27
7. Pitch Sieve to Trigger Virtual Instruments 30
8. Pitch Stabilizer to Control DSP Modules 32
9. Pitch Tracking to Trigger Pitch Shifter for Sample Playback 34
10. Note Density Analyzer 36
11. Mute Control Subpatch 39
12. Mute Control 40
13. Audio Signal Routing and Mixing 42
14. Crossbar and Mixing 45
15. Random Panning in Linear Crossfade Mode 46
16. Windowed Random Panning 48
17. Equal-Distance Random Panning 50
18. Mute Control and Input Selector 51
19. D as Pitch Center (mm.163-164) 55
20. D as Pitch Center (mm.1-2) 55
21. Comparison between Different Versions of *River of Sorrow* 62
22. Allusions and Quotations from River of Sorrow in *Snow Spell* 63

PART I

CRITICAL ESSAY

CHAPTER 1

INTRODUCTION

Snow Spell is a real-time interactive work for erhu, flute, cello, piano, and computer. This work is designed to extend the possibilities of music expression through the use of contemporary technology. It is also an attempt to broaden the scope of real-time interactive pieces, most of which are limited to solo instruments either due to the limitations of technology or the lack of former examples.

The total duration of *Snow Spell* is about 18 minutes. In addition to exploring the possibilities of composition with new media and technology, this one-movement piece is also intended to depict the beauty of a snow scene by presenting four different impressions of snow through music, *The First Snowflake*, *Footprints Vanishing in the Storm, A Soliloquy in the Snow*, and *The Past Melts from the Window Panes*. The four impressions of the snow scene are presented in music through different textures, timbres, and temporalities. Transitions sometimes obscure the demarcation between the four different sections. Pentatonic scales and references to the Chinese folk tune *River of Sorrow* are incorporated to evoke Chinese folklore as a context for the images of snow. Idiomatic embellishment and glissandi from traditional Chinese erhu music are also employed. A novel glissando notation system is used in order to incorporate various glissando techniques from traditional erhu music into western notation. The harmonies focus on the interval of a third, used both diatonically and chromatically in (0 1 4) or (0 1 4 5) pitch sets.

The computer part of the piece is programmed in Max/MSP[1], a graphical and object-oriented programming environment for music, audio, and multimedia. Max/MSP and other similar software systems allow the user to create a "patch", [2] a collection of graphical objects interconnected by lines to serve as instructions obeyed by the computer. The functional content of each object is hidden from the user to simplify the programming process. Additional third-party libraries were used to enhance Max/MSP: Jimmies 1.1 libraries,[3] Eric Lyon's Potpourri[4] libraries, and Miller Puckette's external "fiddle~".[5] These tools and others are used to create the patch that runs the computer part. Prerecorded samples of flute and erhu are edited and transformed using sample manipulation software. To advance the cues in the patch, this piece may be controlled either by the computer's keyboard, or by a MIDI footswitch connected to the computer.

---

[1] Cycling'74, *Max/MSP*, http://www.cycling74.com (accessed March 1, 2007).

Among the four instruments used in this piece, the erhu takes the leading role in the music (especially in *A Soliloquy in the Snow*). Even though each of the four instruments is amplified, the computer does not always process all four instrumental sounds at the same time, in order to maintain a different sense of spatialization between the processed instruments and the non-processed acoustic sounds. Extensive solo passages and instrumental silence are used to leave more space for the computer to do precise pitch tracking and to bring its potential into full play.

---

[2] Miller Puckette, "Max at Seventeen," *Computer Music Journal* 26, no. 4 (2002): 35.

[3] IRCAM (Institut de Recherche et Coordination Acoustique/Musique), *Real-Time Software Pack*, http://www.ircam.com (accessed March 1, 2007).

[4] Eric Lyon, *Potpourr*, http://eamusic.dartmouth.edu/~eric/MAX (accessed March 1, 2007).

[5] Miller Puckette, *fiddle~*, http://www.akustische-kunst.org/maxmsp/other.html (accessed March 1, 2007).

CHAPTER 2

CONCEPT AND HISTORY OF INTERACTIVE MUSIC

The history of music shows that the development of technology has affected the development of musical instruments and the diversification of musical media, inspiring new musical ideas and ways of expression. Valved horns and trumpets facilitated chromatic playing in the first decades of the 19th century, sparking Wagner's interest to raise the role of the brass section within the orchestra.[6] The phonograph record and magnetic tape make possible Pierre Schaeffer's experiments in Musique Concrete in the late 1940s.[7] More recently, new developments in computer science started a new trend of interactive composition and performance, and began to attract more and more composers and performers to this new area. This helps expand the possibility of seamlessly combining traditional instrumental performance and the sound world of electronic media.

Definition

In a broader sense, everything involving people working together or having influence on each other is interactive. The term "interactive" may be confusing when applied to music since interactivity has always been there among performers: between the conductor and the performers, and between the performers and the audience in any live performance. If we narrow down the definition of "interactive music" to "interactive computer music," emphasizing the relationship between human and machine, it can still mean various things. A piece created through algorithms whose computation is initiated by the composer with keys, buttons, or mouse, can be interactive music since it is accomplished through the collaboration between composer (or programmer) and computer. The pre-composed music made non-linear and adaptive through the storyline of a computer game can be interactive music. A piece of web art intended for non-musician users to create their own music using the mouse to “drag and drop” musical elements or parameters can also be interactive music.[8] Todd Winkler has defined interactive music as “a music composition or improvisation where software interprets a live performance to affect music generated or modified by computer.”[9] Robert Rowe, instead of defining interactive music, defines interactive computer music systems as ones whose behaviors change “in response to musical input” and whose responsiveness allows them to “participate in live performances, of both notated and improvised music.”[10] These definitions may not be able to cover various categories of interactive music, but they stress the importance of combining live performers with computers.

---

[6] "Orchestration," in *The New Harvard Dictionary of Music*, ed. Don Michael Randel (Cambridge: Belknap Harvard, 1986), 576.

[7] Robert P. Morgan, *Twentieth-Century Music: A History of Music Style in Modern Europe and America*, 1st ed. (New York: W.W. Norton, 1991), 463.

Guy E. Garnett looks at “interactive computer music” as “a sub-genre of performance-oriented computer music – that is, any computer music that includes a strong performance component.”[11] Under this broader category, there is at least one live performer joined with computer-generated or electronically produced or modified music. Therefore this genre “incorporates both the tape-plus-instrument medium and more recent interactive computer music – whether pre-composed, improvised, randomly generated, or a mixture of these."[12] However, even if we confine the meaning of "interactive music" to the concert performance context, it still carries too broad a meaning concerning various forms of music creation and presentation. It raises the issues of the distinction between music and sound work, instruments and audio controllers, and composition and improvisation. Even the roles of computer and performer are called into question since a person operating various controllers to generate sounds can be viewed as either a performer or a composer in a broader sense. An instrument player triggering musical response from a computer by inputting instrument sound through microphone or by inputting various parameters of physical gestures through sensors can be viewed as a composer in a broader sense, too.

---

[8] Various examples of music related interactive web art can be found on this website (http://vispo.com/misc/ia.htm) which presents different on-line user-friendly interface for people to enjoy the easy fun of creating their own music (accessed March 1, 2007).

[9] Todd Winkler, *Composing Interactive Music: Techniques and Ideas Using Max* (Cambridge, Ma.: MIT Press, 1998), 4.

[10] Robert Rowe, *Interactive Music Systems* (Cambridge, Mass., London: MIT Press, 1994), 1.

[11] Guy E. Garnett, "The Aesthetics of Interactive Computer Music," *Computer Music Journal* 25, no. 1 (2001): 21.

In the following discussion of interactive music, the author will follow Garnett's definition but confine the performance element to traditional acoustic instrument players, excluding those that manipulate electronic audio devices or computer music software to generate sounds or music in a concert performance. This definition will serve as a better focal point for the documentation of the interactive piece, *Snow Spell*.

History

In interactive music, the interaction happens in two aspects: "either the performer's actions affect the computer's output, or the computer's actions affect the performer's output."[13] These two aspects may also combine together in various ways and at different levels. Interactive music, as a sub-genre of performance-oriented computer music, can be traced back to its forerunner, the combination of tape music with live performance. In this case, the tape part tends to dictate the performer's tempo. However, such tape-plus-instrument combinations still provide some valuable elements that the absolutely fixed tape piece does not: spontaneity, the immediate expressive contact, visual stimuli, and the subtleties of musical interpretation.[14] The earliest example of such tape-plus-instrument piece was the *Musica su due dimensioni* (*Music of Two Dimensions*) for flute, percussion, and tape, composed by Bruno Maderna in the Cologne studio in 1952.[15] The early works of such type have the tendency to emphasize timbral contrasts between the taped and performed music but later also to exploit their similarities as well as differences.[16] In order to involve more temporal interaction between tape and live instrument, Davidovsky's *Synchronisms* No.1 for flute and tape (1962) employed cues to allow the flutist to affect the start and stop points of the pre-recorded material. Later on, composers started to include "live" electronic parts, which present a closer model to modern interactive music, rather than just pre-recorded tape music. These might be as simple as manipulating some effects added to the acoustic instruments. For example, Stockhausen's *Mantra* (1970) uses real-time electronic ring modulators to transform the timbre of two pianos during the performance.[17]

---

[12] Ibid.

[13] Ibid.

By 1980, instantaneous interaction and response had been realized in some studios. For example, at the Colgate University, Dexter Morill experimented with a system that allows one to

---

[14] Morgan, *Twentieth-Century Music: A History of Music Style in Modern Europe and America,* 470.

[15] Ibid.

[16] Ibid., 471.

[17] John A. Maurer, *Interactive Electronics in Computer Music*, 1999, http://ccrma.stanford.edu/~blackrse/interactive_electronics.html (accessed March 1, 2007).

"play" the computer on a drum-like surface.[18] The introduction of the Musical Instrument Digital Interface (MIDI) and the development of MIDI-related controllers, software, and instruments greatly affected the later development of real-time interactive computer music with live acoustic instruments. The introduction of MIDI facilitated the digital and computerized communication with MIDI-compatible musical instruments in concert performances. Examples include the use of a MIDI trumpet by Dexter Morrill in his *Sketches for Invisible Man* (1989) to trigger spontaneous electronic events, and Gary Nelson'**s** creation of a MIDI horn for *Warps of Time* (1987) to trigger spontaneous effects and algorithmically generated music output to various synthesizers.[19] However, even today, acoustic instruments still can not be replaced by digital MIDI instruments because of their complicated timbral characters and processes of sound generation and control. Early pioneers in MIDI-based interactive music systems are the M and Jam Factory software packages released in 1986.[20] These programs can generate musical patterns based on a performer's actions using MIDI instruments. Opcode's version of Max, and Mark Coniglio's Interactor software[21] opened the field further. A more recent direction to realize the combination of live acoustic instruments and controllable computerized sound worlds is through real-time audio analysis, synthesis and sound processing.

---

[18] Elliott and Daniel Godfrey Schwartz, *Music since 1945: Issues, Materials, and Literature* (New York: Schirmer Books, 1993), 354.

[19] Ibid., 362-63.

[20] Winkler, *Composing Interactive Music: Techniques and Ideas Using Max*, 15-16.

[21] Robert Rowe, "Interactor 4.0.8," *Leonardo Music Journal* vol. 2, no. No. 1 (1992).

By 1990, various programmable interactive MIDI systems had proved their success in concert situations, among which the Kyma[22] system (starting from 1986) once stood out for its capability of dealing with audio synthesis and sound processing. Another important programming environment, Max, grew from the Patcher created by Miller Puckette at IRCAM in the mid-1980s. It gives composers easier access to an authoring system for interactive computer music through its graphical user interface and efficient way of "combining pre-designed building blocks into configurations useful for real-time computer music performance."[23] The earliest example of realizing the combination of live instruments and interactive computer music system with Max is Philippe Manoury's piece *Jupiter* (1987) for flute and computer which "broke new ground in using real-time pitch tracking and score following to correlate computer-generated sound with a live performer."[24]

With the technology of pitch tracking, the pitch content of an audio signal from the live performance can be detected to allow closer communication between live acoustic instruments and the computer. The detailed synchronization between live instruments and electronics can be achieved when such data is used in score following, a process of automatically "listening" to a live performance and tracking the position in the score in order to trigger or automate different presets and accompaniments in the interactive computer music system.[25] Nowadays in addition to Kyma and Max/MSP mentioned above, interactive music systems such as SuperCollider,[26] jMax,[27] PureData and CsoundAV,[28] RTcmix,[29] RTCSound are also available to users. Several of these systems support pitch tracking and score following techniques.

---

[22] Symbolic Sound Corporation, *Kyma*, http://www.symbolicsound.com (accessed March 1, 2007).

[23] Puckette, "Max at Seventeen," 31.

[24] Andrew May, "Philippe Manoury's Jupiter," in *Analytical Methods of Electroacoustic Music* ed. Mary Simoni (New York: Routledge, 2006), 145.

[25] M. and C. Lippe Puckette, "Score Following in Practice," in *Proceedings of the International Computer Music Conference* (San Francisco: 1992), 182.

Significance

"Pure" electroacousitc music played back from fixed media creates a very different performance environment from other music: the stage is empty, and the human elements of interpretation and communication in performance are absent. Adding a live performer to the performance enables computer music, like traditional repertoire, to incorporate "gestural nuance such as rubato, subtleties of phrasing and articulation, and dynamics."[30] Fixed media works are usually played in a darkened environment due to their lack of visual elements, whereas visual stimuli from performers help communicate the music to listeners. Facial expressions and physical gestures, synchronized with musical gestures, help reinforce the mood and expression of the music. This creates the "acoustic intimacy" and "the feeling of closeness to the performer."[31] It also prevents the audience from becoming “detached, isolated, and emotionally removed from the scene”[32] and helps the listener to concentrate.

---

[26] James McCartney, *SuperCollider*, http://www.audiosynth.com (accessed March 1, 2007).

[27] IRCAM (Institut de Recherche et Coordination Acoustique/Musique), *jMax*, http://freesoftware.ircam.fr (accessed March 1, 2007).

[28] Barry Vercoe, *Csound*, http://www.csounds.com (accessed March 1, 2007).

[29] Brad Garton and Dave Topper, *RTcmix*, http://music.columbia.edu/cmc/RTcmix (accessed March 1, 2007).

[30] Garnett, "The Aesthetics of Interactive Computer Music," 25.

[31] Jerald R. Hyde, "The Concert Hall Experience: How Visual Stimuli Can Affect What We Hear," in *Acoustical Society of America 147th Meeting* (New York: 2004).

From the performer’s perspective, playing with fixed media severely limits timing, tempo, and phrasing. This has been described as “like working with the worst human accompanist imaginable: inconsiderate, inflexible, unresponsive and utterly deaf.”[33] Interactive computer music can give the performer better control of time and dynamics. With more flexibility for interpretation in interactive music, “each performance is a unique representation of the work, each performance can change details and change emphasis, and, for better or worse, each performance becomes a version of the work and not the work itself.”[34] Just like the repertoire of traditional classical music which focuses so much on performer interpretation, the fact that interactive music is open to new interpretations creates opportunity for “the growth of the work over time or across cultural boundaries.” [35] Because of this, such music may be able to “maintain a longer life and have a broader impact culturally, because it is able to change to meet changing aesthetic values.”[36]

Real-time signal processing in interactive music can greatly expand the performers’ instrumental capability as well. For example, through real-time transposing techniques, the pitch range of an acoustic instrument can be expanded beyond the physical limits of the instrument.

---

[32] Garnett, "The Aesthetics of Interactive Computer Music," 32.

[33] Elizabeth McNutt, "Performing Electroacoustic Music: A Wider View of Interactivity," *Organized Sound* 8, no. 3 (2003): 299.

[34] Garnett, "The Aesthetics of Interactive Computer Music," 26.

[35] Ibid., 27.

[36] Ibid.

Through phase vocoding, looping, reverberation, and granulation, an instrument with no sustaining capability can produce sustained tone based on its unique timbre. Through time-stretching and delay effects, the contrapuntal textures of imitation, diminution, augmentation and even retrograde can be achieved with one monophonic instrument. While still retaining the recognizable features of the instrumental source, real-time interactive music systems can realize an “unhearable ideal” of instrumental timbre. In this way, a clear relationship between live performance and computer sound is possible: one which can be made “cognitively graspable, without confining it to what is already cognitively grasped.” This “brings about a more realistic compositional attitude which in turn leads to more successful works,”[37] in the view of some musicians.

In interactive music, the computer is humanized as a performer, extending and amplifying the capability of the live instrument to bring new surprise to the listener. At the same time, the interactive relationship between the performer and the computer in particular systems using a minimum of hardware on stage creates a magical presence that is intrinsically mysterious and surprising. This presence itself becomes a compositional element that requires great care and consideration:

> Live interactive music contains an element of magic, since the computer music responds “invisibly” to a performer. The drama is heightened when the roles of the computer and performer are clearly defined, and when the actions of one has an observable impact on the actions of another, although an overly simplistic approach will quickly wear thin. On the other hand, complex responses that are more indirectly

---

[37] Ibid., 26.

> influenced by a performer may produce highly successful musical results, but without some observable connection the dramatic relationship will be lost to the audience.[38]

To sum up, interactive music is a new form of musical expression which leaves more space for interpretation (compared to other types of computer music) and thus has greater potential to grow over time. It provides visual stimuli and human interactions to the audience as in traditional instrumental performance. It takes advantages of computer technology to transform, extend, and amplify instrumental sounds, creating beauty out of the mixture of the familiar and the unfamiliar. The magic element of its disembodied responses also creates surprise and mystery. All these features endow interactive music with a strong potential to become a uniquely powerful medium for art music expression in the twenty-first century.

---

[38] Winkler, *Composing Interactive Music: Techniques and Ideas Using Max*, 9.

CHAPTER 3

INTERACTIVITY BETWEEN PERFORMERS AND COMPUTER IN SNOW SPELL

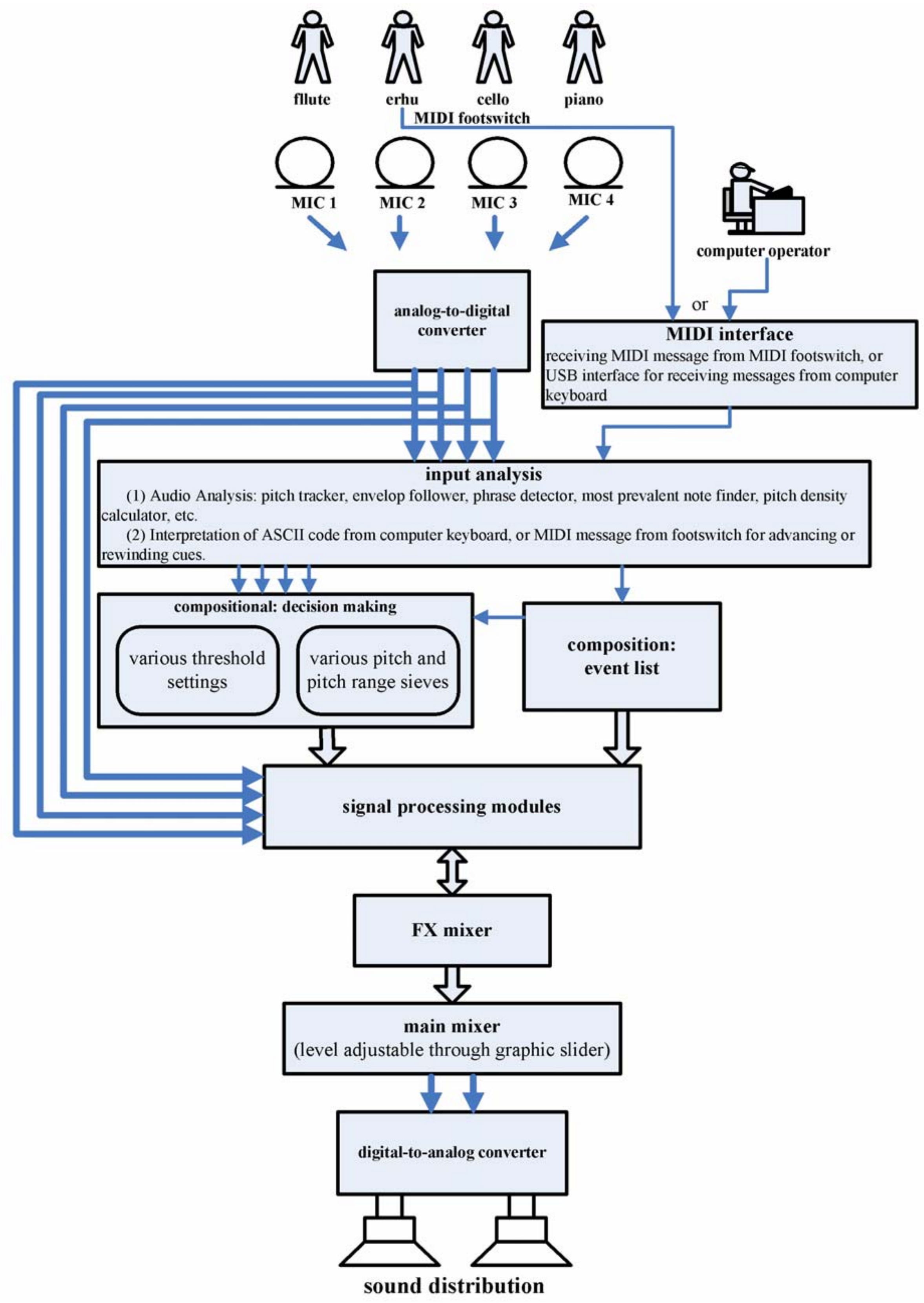
Programming Structure

The programming structure for *Snow Spell* is mainly composed of five components:

1. Input data analysis includes the audio analysis of live instruments and the analysis of input data from MIDI footswitch or computer keyboard. The former includes the interpretation of basic musical elements of the live instrumental sound such as pitch, amplitude, phrase starting and ending points, phrase length, pitch density, and the most prevalent pitches. The latter includes interpretation of MIDI messages or ASCII codes to advance or rewind the cues in the music.
2. Compositional and decision-making systems use the results of input data analysis to control signal processing modules. Input data are interpreted by threshold settings (for density of pitch activity, pitch range, amplitude, and pitch prevalence) and various pitch sieves. They send out parameters to the signal processing modules or decide when to turn on/off these modules. The mapping of input-based decisions to output parameters is controlled by the event list, which serves as the music score triggering signal processing modules or sending out parameters to them.
3. Signal processing modules which transform or generate musical materials in accordance with the controls described above.
4. Matrix mixers which handle the routing of input and output signals from the live instruments and the various signal processing modules

The diagram of the programming structure for *Snow Spell* is illustrated below:

Figure 1. Programming Structure


fllute
erhu
cello
piano
MIDI footswitch
MIC 1
MIC 2
MIC 3
MIC 4
computer operator
or
analog-to-digital converter
MIDI interface
input analysis
compositional: decision making
various threshold settings
various pitch and pitch range sieves
composition: event list
signal processing modules
FX mixer
main mixer
(level adjustable through graphic slider)
digital-to-analog converter
sound distribution

receiving MIDI message from MIDI footswitch, or USB interface for receiving messages from computer keyboard

(1) Audio Analysis: pitch tracker, envelop follower, phrase detector, most prevalent note finder, pitch density calculator, etc.

(2) Interpretation of ASCII code from computer keyboard, or MIDI message from footswitch for advancing or rewinding cues.

The connection between live instruments and the computer is built through four microphones picking up signals and sending them respectively to the computer through analog-to-digital converter. The four different incoming signals from the A/D converter are analyzed by the computer separately through pitch and amplitude detection functions offered by Miller Puckette's fiddle~[39] and other signal processing objects in Max/MSP. The pitch tracker outputs MIDI-style pitches and the amplitude detector outputs amplitude scaled as MIDI numbers (0-127). Since there are four different groups of pitch and amplitude data respectively from flute, erhu, cello, and piano, they are used selectively for decision making and control, as determined by the event list.

The signal processing modules can be viewed as virtual musical instruments (in a broader sense) which serve to produce or process sounds in the interactive music system. To identify characteristics of different interactive music works, Robert Rowe classifies interactive systems as transformative, generative and sequenced; score-driven or performance-driven; and instrument or player paradigm.[40] Similar to Rowe's first way of classification, Cort Lippe exemplifies these differentiations:

> A DSP module which outputs altered version of an incoming signal can be considered transformative, while a module that does not take an incoming signal, but only outputs a signal, can be considered generative. Thus, reverberation is transformative and synthesis via oscillators is generative….Sample playback falls, somewhat arbitrarily, in a grey area between transformative and generative. Depending on the application, sampling could be considered highly transformational (real-time

---

[39] Puckette, *fiddle~*.

[40] Rowe, *Interactive Music Systems*.

> recordings of input which are played back in non-standard ways), while normal playback of pre-recorded sounds might be considered generative.[41]

According to these definitions, the interactive system for *Snow Spell* includes both transformative and generative functions plus playback of pre-recorded sound files which is viewed as generative method by Lippe, or as sequenced method by Rowe. The FM virtual instrument is generative. Other signal processing modules in *Snow Spell*, such as delay effects, harmonizer, FFT Freezer, granular synthesis are transformative. These signal processing modules are controlled by compositional algorithms which can be viewed as virtual performers making musical decisions based on the result of input analysis mechanism, the virtual listeners.

Snow Spell is primarily a score-driven work in which most events are predetermined and organized in a linear way. Compared to a performance-driven work which is basically improvised and does not expect the realization of any particular score, a primarily score-driven work allows the composer to determine the musical materials, the overall formal structure, the timing and the style of the piece. In order to construct a work with such a pre-determined linear structure, it is necessary to design a scheme for encoding and successively triggering all of the events and parameter changes that will articulate the form, as well as a system for correlating these with data from the performers. In this work, the encoding of events and their correlation with input data are accomplished using the technique of score following, which was designed to permit the computer to accompany live performance. A score following technique based on tempo detecting was first demonstrated at the 1984 International Computer Music Conference, independently, by Barry Vercoe[42] and Roger Dannenberg.[43] For the purpose of better accommodating performers' timing interpretation, a new score following system based on pitch tracking was developed later at IRCAM.[44] However, the pitch-tracking based score follower has proved not to be perfectly stable, and usually requires an assistant to click through the major cues of the piece.[45] In view of this, instead of relying on pitch tracking, the major cues for large sections in *Snow Spell* and other cues that demand precise synchronization are cued via a MIDI footswitch or by an assistant at the computer keyboard. In addition to different cues which trigger different settings for continuous and automated signal processing modules, control of digital effects, synthesis, and sound file playback is provided by threshold settings, different mechanisms of pitch sieves and their combinations. These compositional devices become the nucleus of the interactivity in *Snow Spell* and will be elaborated in the following section.

---

[41] Cort Lippe, "Music for Piano and Computer: A Description," *Information Processing Society of Japan SIG Notes* 1997, no. 122 (1997).

Interactivity and Compositional Algorithms

Phrase length detector to control harmonizer DSP effect:

The phrase length detector used in *Snow Spell* is based on the assumption that a musical phrase is an audio event whose amplitude has to be above a certain threshold. In order to exclude unexpected noise from being counted as a new audio event, an audio signal with its level above the specified amplitude threshold needs to be long enough to be counted as the start of a new event. Similarly, for the purpose of ignoring short break points in musical articulation in exchange for recognizing a larger phrase, a duration threshold for silence to be counted as the ending of an event is also required. These three threshold settings for amplitude, duration of silence, and duration of sound are indispensable for more meaningful analysis results that can suit compositional purposes.

---

[42] Barry Vercoe, "The Synthetic Performer in the Context of Live Musical Performance," in *The Proceedings of International Computer Music Conference* (Paris, France: 1984), 185.

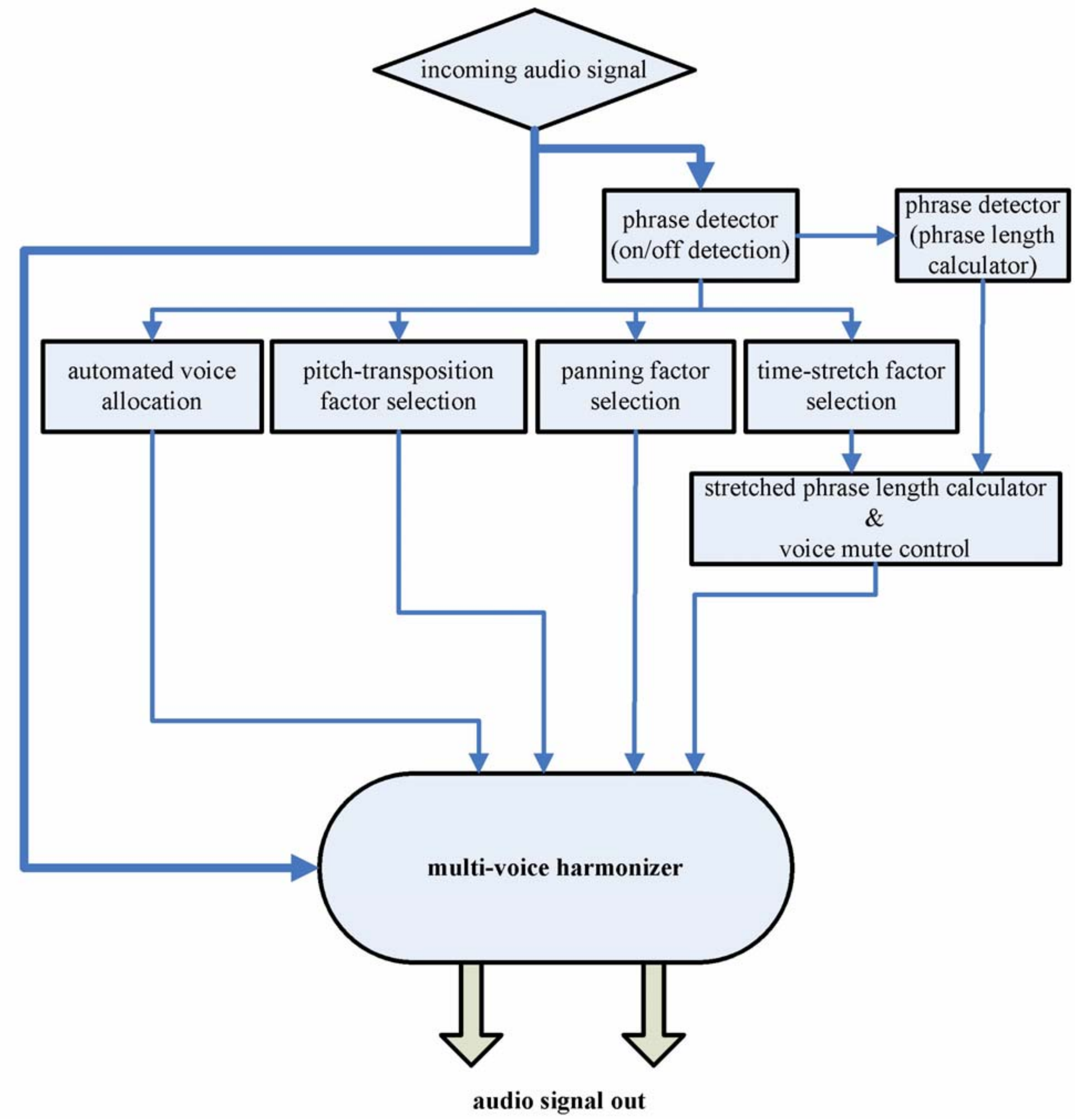
[43] Roger Dannenburg, "An on-Line Algorithm for Real-Time Accompaniment," in *The Proceedings of International Computer Music Conference* (Paris, France: 1984), 187.

[44] Miller Puckette, "Score Following Using the Sung Voice," in *The Proceedings of International Computer Music Conference* (Canada: 1995), 199.

[45] McNutt, "Performing Electroacoustic Music: A Wider View of Interactivity," 301.

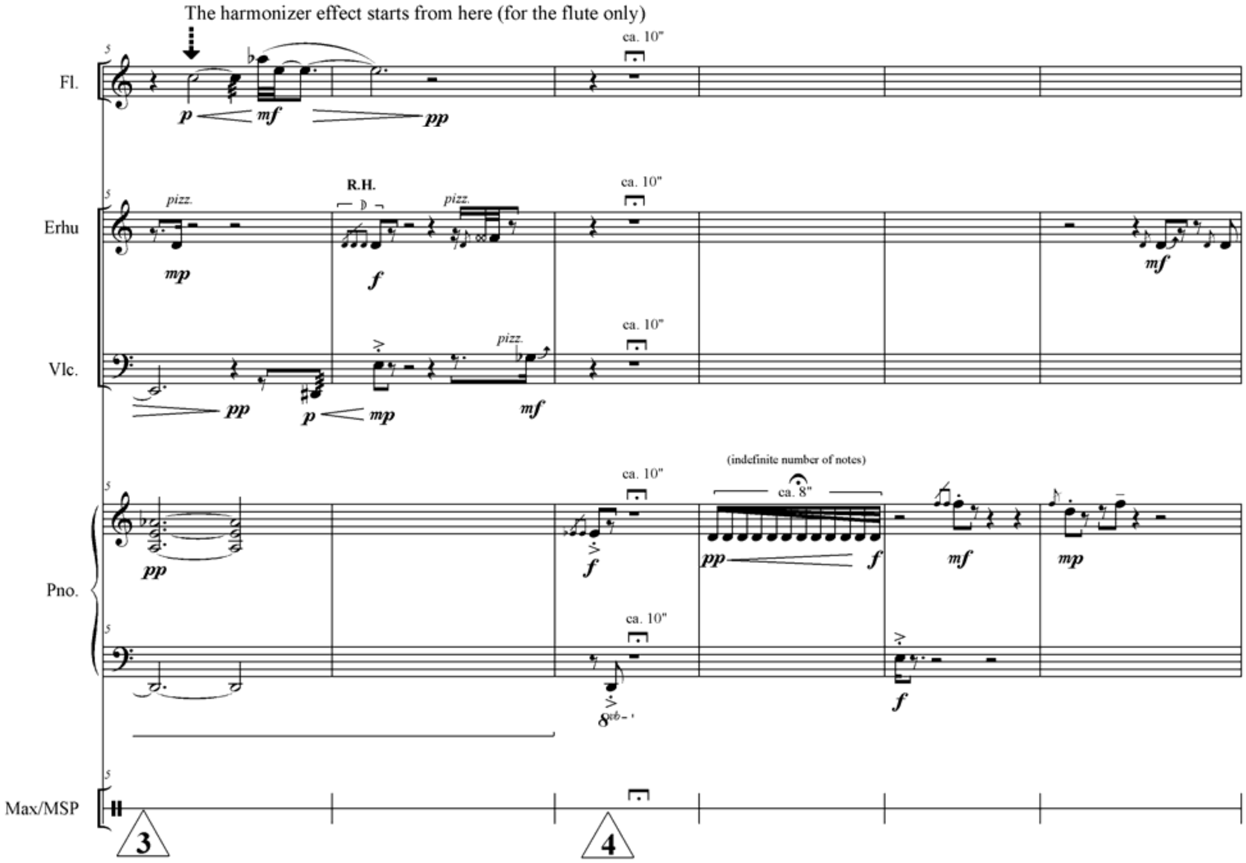
In *Snow Spell*, the phrase detector is used to control the multi-voice harmonizer DSP effect which transposes the incoming signal and stretches its duration. Whenever a new phrase is detected, a pitch transposition factor, a time-stretch factor, and a panning factor are chosen and sent to one of the harmonizer voices.   At the same time, the time-stretch factor is memorized in another algorithm. Thus whenever the incoming current event is detected as ended, the algorithm will be able to calculate the total length of the stretched signal and to send out a command later to that voice when the stretched audio signals ends in order to free it for further use by other detected audio events. These procedures can be illustrated in the following diagram.

Figure 2. Phrase Length Detector to Control Multi-voice Harmonizer


incoming audio signal
phrase detector (on/off detection)
phrase detector (phrase length calculator)
automated voice allocation
pitch-transposition factor selection
panning factor selection
time-stretch factor selection
stretched phrase length calculator & voice mute control
multi-voice harmonizer
audio signal out

This application of a multi-voice harmonizer is intended to create a kind of strict counterpoint in augmentation. Voices are added into the music one by one following the onset of every newly detected phrase, and may also overlap each other according to their time-stretch factors and the original phrase length. Since this technique is used in a multiple-instrument piece, the pitch transposition factors, even though randomly chosen, are still confined only to different octaves to ensure the clarity of melodic lines and harmony. The imitation relationship is most recognizable when the texture is thin. This happens with the flute line in measures 5-6, introducing the device to the audience. At this point, the two string instruments are doing pizzicato, and the pitch activity of erhu and piano is limited to only two pitches (D and F), which further differentiates the flute melody and its duplicated versions from the other instruments.

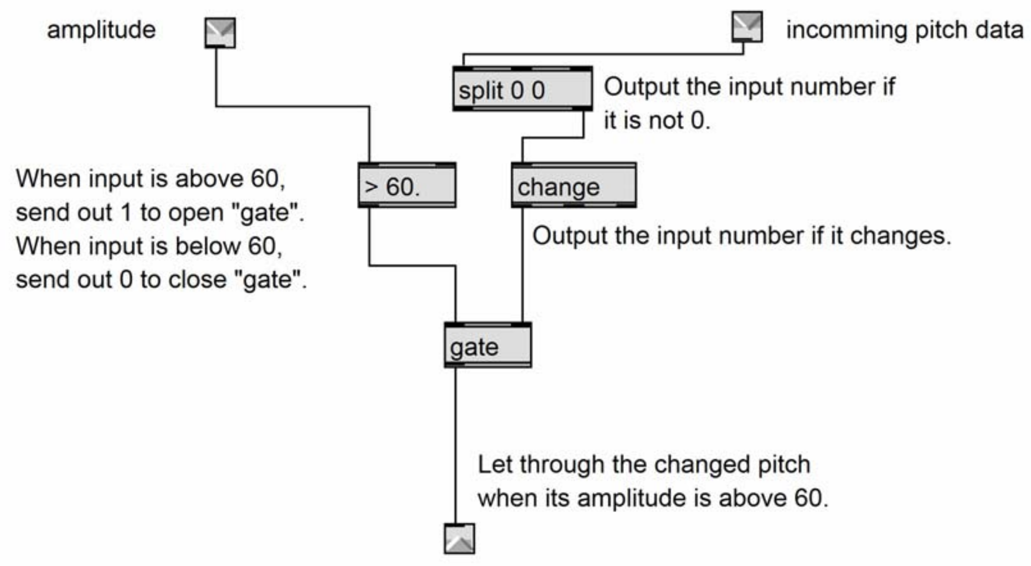
Figure 3. Harmonizer in *Snow Spell*


The harmonizer effect starts from here (for the flute only)
Fl.
Erhu
Vlc.
Pno.
Max/MSP
ca. 10"
pizz.
R.H.
(indefinite number of notes)
ca. 8"
p
mf
pp
mp
f
3
4

New pitch detection combined with amplitude analysis result to trigger FFT Freezer:

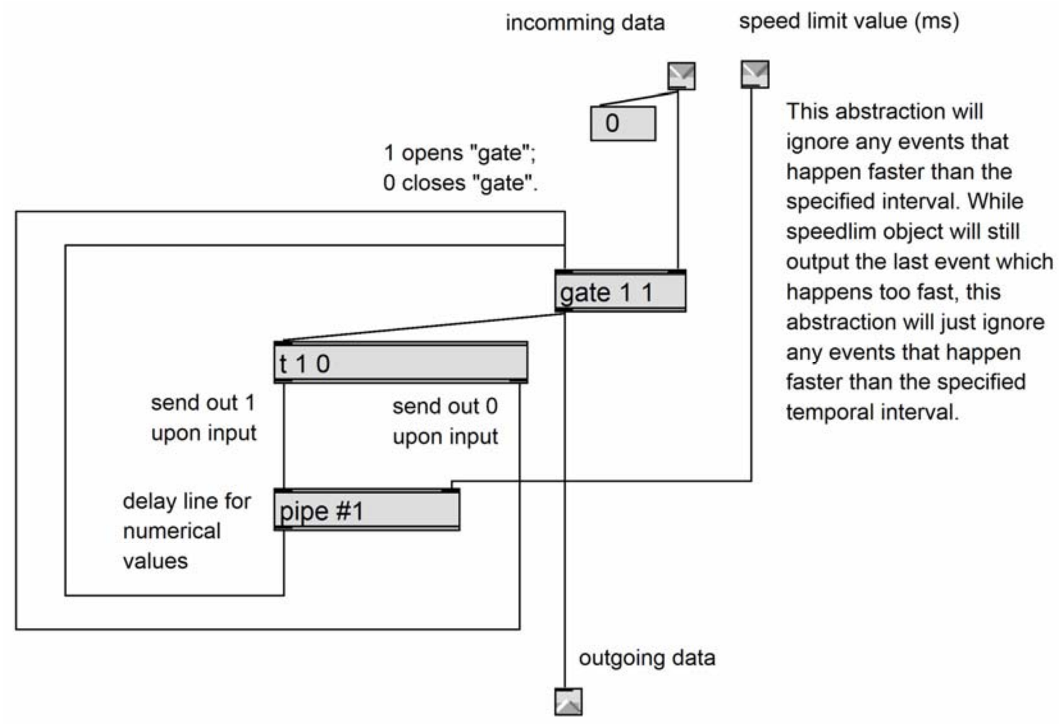
The effectiveness of signal processing modules depends greatly upon the input signal and the controls applied to the modules. This is true of the FFT Freezer, a sampling and re-synthesis technique based on the Fast Fourier Transformation. This technique analyzes the spectrum of a frame of samples, and re-synthesizes that spectrum repeatedly to create a sustained timbre. The timbre produced by the FFT Freezer does not vary over time. It does not change much when the effect is re-triggered by the same pitch on the same instrument. When several FFT Freezers are triggered at nearly the same time, they tend to fuse into a single timbre. In *Snow Spell*, individual notes are frozen and panned to different locations to create a magical sense of momentary events frozen in space and time. To accomplish this and to overcome the limitation described above, the control system excludes consecutively repeated pitches, as well as events of very low amplitude.

Figure 4. Pitch Detection Combined with Amplitude Analysis


amplitude
incomming pitch data
split 0 0
Output the input number if
it is not 0.
When input is above 60,
send out 1 to open "gate".
When input is below 60,
send out 0 to close "gate".
> 60.
change
Output the input number if it changes.
gate
Let through the changed pitch
when its amplitude is above 60.

It also limits the rate at which new “frozen notes” can be triggered. Max/MSP provides an object named “speedlim” that limits the speed at which data passes through. However, it does not ensure the synchronization of incoming and outgoing data since it will hold and delay a message. A better solution to limit the speed of data and to ensure better synchronization of incoming and outgoing data is shown below.

Figure 5. Speed Limit Device


incomming data
speed limit value (ms)
0
This abstraction will ignore any events that happen faster than the specified interval. While speedlim object will still output the last event which happens too fast, this abstraction will just ignore any events that happen faster than the specified temporal interval.
1 opens "gate";
0 closes "gate".
gate 1 1
t 1 0
send out 1 upon input
send out 0 upon input
delay line for numerical values
pipe #1
outgoing data

This device prevents output of a new message until a certain amount of time has passed since the last message was output. Events in between are ignored (the “gate” is closed to them). This mechanism is also used for other signal processing modules such as FM synthesis and sound file triggering in *Snow Spell*.

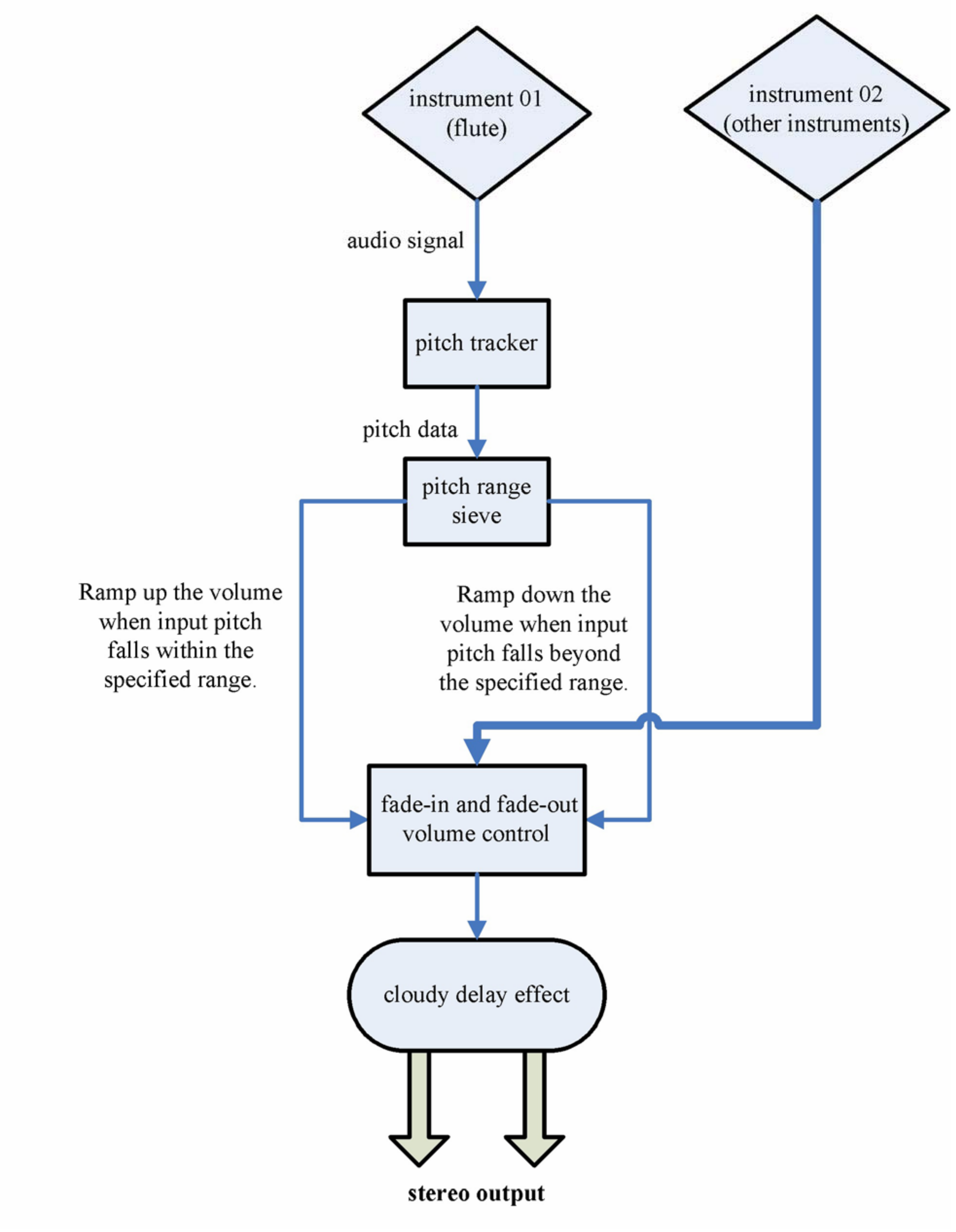
With these algorithms, the re-synthesized signal of FFT Freezer and the attack of the note that triggers it combine together to form a new instrumental timbre which extends the original. Since complicated texture of the original live instruments will obscure the clarity of the interesting effect, FFT Freezer is mainly applied in music of thin texture (e.g. m. 5, for piano without loud pedal) or solo dominated passages (e.g. m.135, for erhu solo).

Pitch range to control signal processing modules:

In *Snow Spell*, the pitch range sieve is built based on the assumption that pitches of extreme range usually indicate higher tension in music. A pitch range sieve algorithm can specify the pitch range within which the incoming pitch data can trigger a certain signal processing module to further intensify or amplify the original music in addition to its transformative or generative purposes. In *Snow Spell*, instead of directly controlling the parameters of signal processing modules, the pitch range sieve is mainly used to control the input level sent into one particular signal processing module, cloudy delay. This module splits the incoming signal into fragments of irregular length, which are then distributed randomly through a stereo panning system, and combined with delay effects to create a cloudy texture. Whenever the pitch analysis result of the incoming signal falls within the specified pitch range of the sieve, the sieve algorithm will send out the message “1” to trigger a ramp-up process for the audio signal level. Conversely, whenever the pitch falls beyond the specified range, the sieve algorithm will send out the message “0” to trigger a ramp-down process for the audio signal level.

Since the pitch range of the flute melody in *Snow Spell* mostly coincides with the overall rise and release of tension in this piece, analysis of the flute’s pitch range, is used to control the level of cloudy delay applied to the other instruments (not the flute itself). In this way, the clarity of the original melody is retained while the overall musical shape is enforced and the tension of the music is selectively intensified by the processed sound. Examples of such usage in *Snow Spell* can be found in m.45, where sound from the piano is processed by the cloudy delay effect according to the pitch range analysis result of the flute. A similar example, with processing applied to the cello sound can also be found in m.107. A diagram demonstrating this method is shown below:

Figure 6. Pitch Range Sieve to Control Signal Processing Modules


instrument 01
(flute)
instrument 02
(other instruments)
audio signal
pitch tracker
pitch data
pitch range
sieve
Ramp up the volume
when input pitch
falls within the
specified range.
Ramp down the
volume when input
pitch falls beyond
the specified range.
fade-in and fade-out
volume control
cloudy delay effect
stereo output

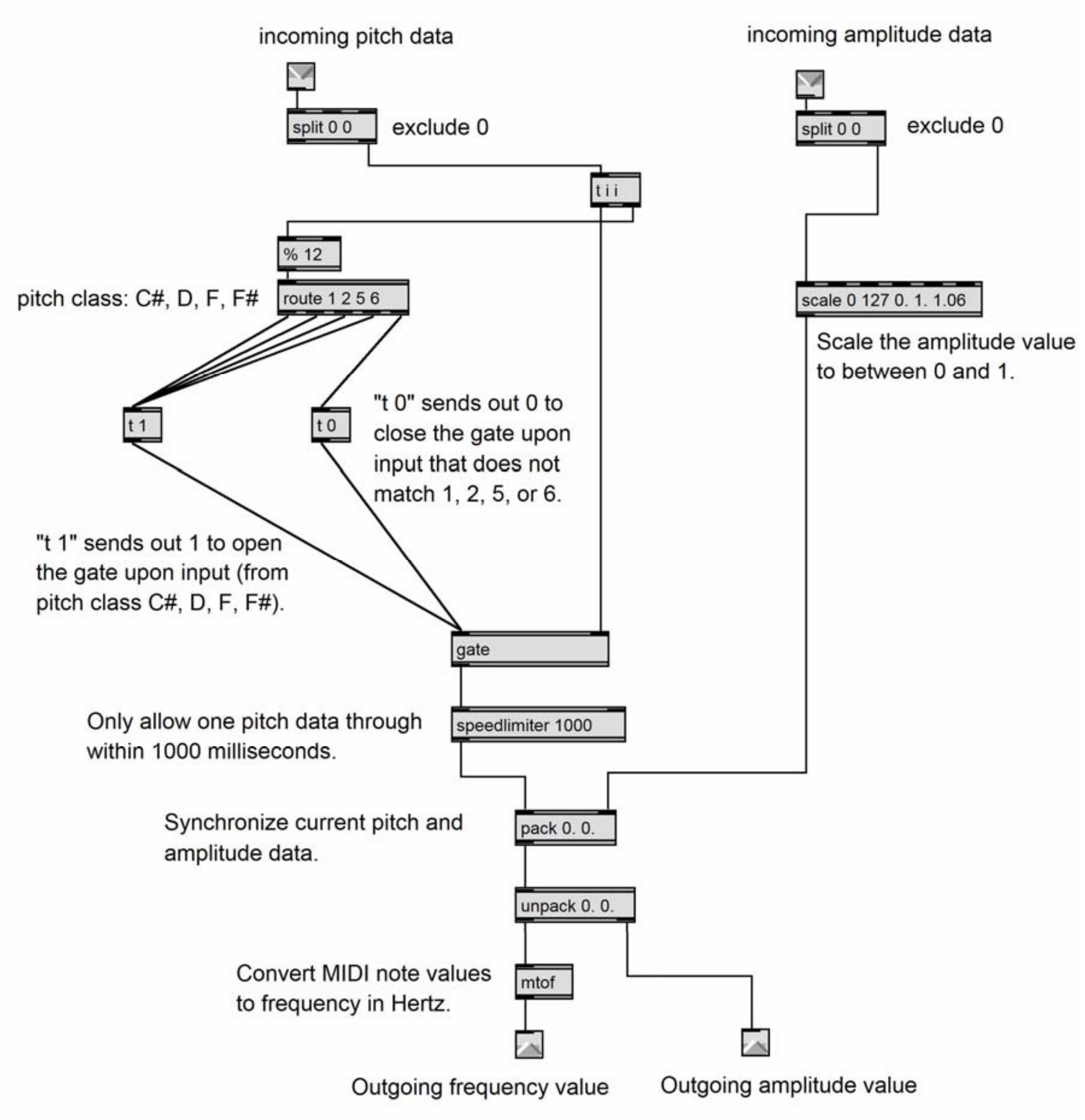
Pitch sieves are also used for arbitrary parameter selection for the granular transposition module. The granular transposer streams incoming pitch data into seven ranges; each is assigned a different factor for pitch transposition. Whenever the mean average of the recent pitch data falls within a particular range, it will send the corresponding factor to the granular transposition modules. In order to keep the processed signal at the same pitch level for enough time to form a recognizable fragment of imitation of the original live instrument melody before a new transposition level is selected, the speed limiter previously described is once again used, with 6000 milliseconds as the minimum interval between each transposition.

Pitch sieve to trigger virtual instruments:

In *Snow Spell*, a pitch sieve is used to control a virtual instrument based on FM synthesis. This software-based synthesizer mainly produces longer sounds which serve to produce pedal-tone effect parallel to the FFT Freezer mentioned above. The main difference between the two is that the FM synthesizer does not take any audio signal from the live instruments as the source for re-synthesis. Another feature of the FM synthesizer is that the timbre changes each time it receives a trigger. Randomly assigned envelopes, stretched proportionally over time according to randomly selected factors, are used to control amplitude and modulation index, creating timbral variety.

In *Snow Spell*, the FM synthesizer not only provides pedal effects, but also plays a structural role, appearing near the beginning (from m.16 to around m.136) and the end (from around m.145 to the end) of the piece to reinforce the mirror form of the work. The pitch sieve algorithm that controls the FM synthesizer only lets through the pitch classes C#, D, F, and F# from the pitch tracking result of the piano. Whenever the piano plays one of these pitches, a new FM note is triggered. This forms a melodic line based on pitch class of C#, D, F, and F#, with an irregular rhythm. The FM synthesizer thus emphasizes the harmonic center of the piece: the harmony of the work focuses on pitch classes C#, D, F, and to a lesser degree F#, at the beginning and returns to them at the end of the piece. In addition to these pitch and rhythmic connections, the amplitude of the FM synthesizer is also controlled by the amplitude of each note detected in the piano part, creating a further correlation between the two. Since the total number of active voices in the FM synthesizer is limited to eight (to save CPU power) and no note can be triggered if all eight voices are occupied, the aforementioned speed limiter algorithm is once again used to ensure a more even distribution of notes over time. The pitch sieve patch that controls the triggering of the FM synthesizer is illustrated below.

Figure 7. Pitch Sieve to Trigger Virtual Instrument

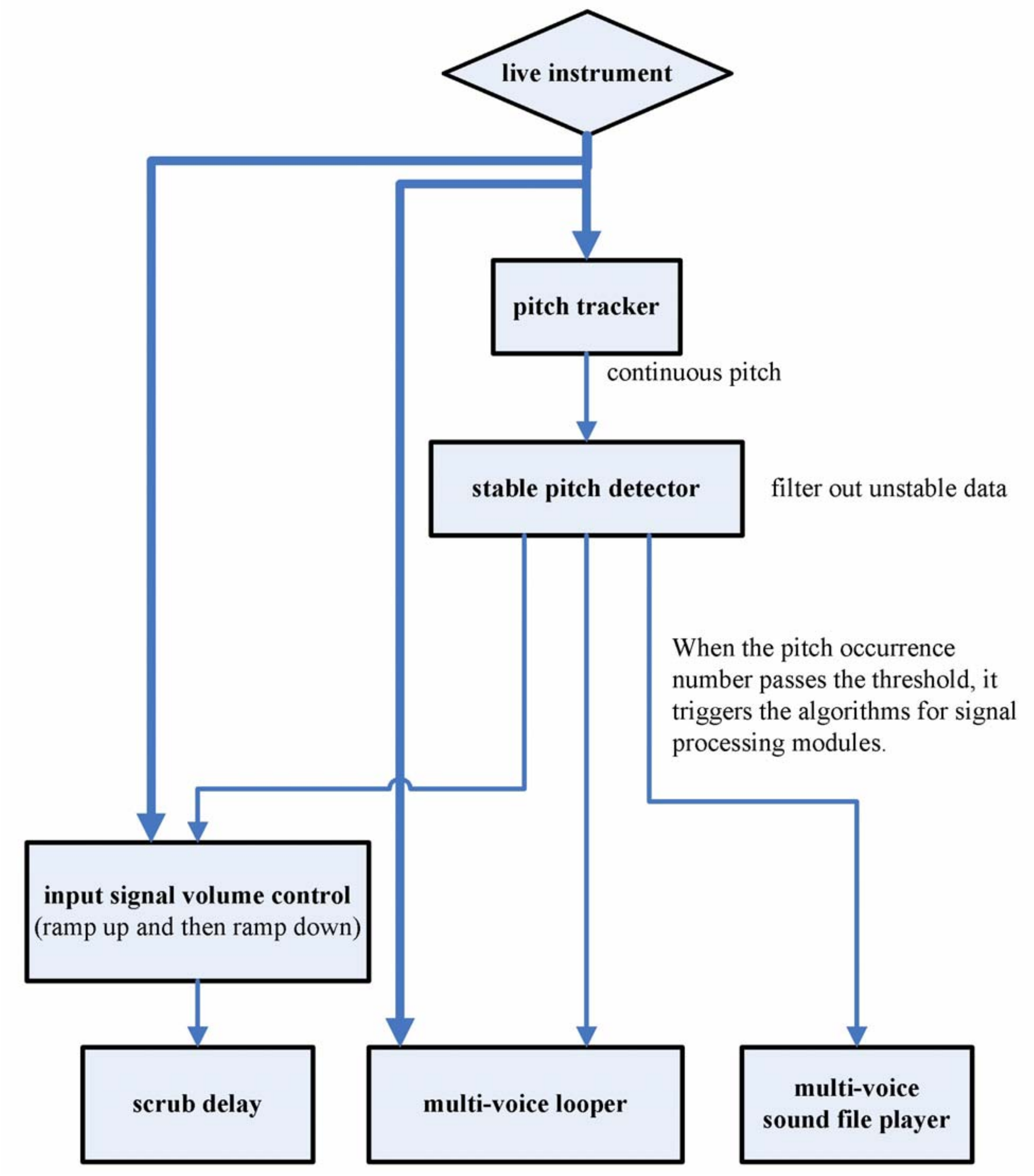


incoming pitch data
split 0 0
exclude 0
incoming amplitude data
split 0 0
exclude 0
t i i
% 12
pitch class: C#, D, F, F#
route 1 2 5 6
scale 0 127 0. 1. 1.06
Scale the amplitude value to between 0 and 1.
t 1
t 0
"t 0" sends out 0 to close the gate upon input that does not match 1, 2, 5, or 6.
"t 1" sends out 1 to open the gate upon input (from pitch class C#, D, F, F#).
gate
Only allow one pitch data through within 1000 milliseconds.
speedlimiter 1000
Synchronize current pitch and amplitude data.
pack 0. 0.
unpack 0. 0.
Convert MIDI note values to frequency in Hertz.
mtof
Outgoing frequency value
Outgoing amplitude value

Pitch stabilizer to trigger delay-based looper:

A technique for detecting a stable pitch, developed by Andrew May, is used to trigger looper. In this technique, continuous pitch data are saved in a circular buffer, and the most prevalent values are detected. The buffer for pitch data collection is updated at the rate of the fiddle~ window size – for flute and erhu, about every 13 ms, slower for cello and piano. When a value occurs enough times within the buffer, this most prevalent pitch value is output. This will trigger signal processing modules in the *Snow Spell* patch; it is desirable because it ensures a highly accurate and stable detector of a note’s pitch. It is also used to improve duration detection by removing extraneous “bounces,” or rapid outputs of inaccurate pitches. A diagram that demonstrates the use of note prevalence algorithm is illustrated below.

Figure 8. Pitch Stabilizer to Control DSP Modules


live instrument
pitch tracker
continuous pitch
stable pitch detector
filter out unstable data
When the pitch occurrence number passes the threshold, it triggers the algorithms for signal processing modules.
input signal volume control (ramp up and then ramp down)
scrub delay
multi-voice looper
multi-voice sound file player

The looping algorithm that is triggered by the pitch stabilizer in *Snow Spell* captures a small segment of the current audio signal in a buffer, plays looped audio from the buffer, and gradually fades the loop down to silence. It is basically another method of making a pedal effect, with a very different timbre from the FFT Freezer and the FM instrument. FFT Freezer sounds when applied to shorter notes of abrupt and strong attack (e.g. piano or string pizzicato); in contrast, the delay-based looper is only suitable for longer notes of smooth envelope (e.g. woodwind or string). Thus, these two algorithms address a similar musical goal in very different situation. Because the delay-based looper relies on sampled sound from the live instrument, it is less flexible but more timbrally consistent than the FM instrument. In *Snow Spell*, the delay-based looping device is only applied to long flute notes at the beginning (mm.1-16) and at the end (mm.157-171) of the piece, which also serves to clarify the formal structure of music.

The pitch stabilizer is also used to trigger drone-like sound files of varying duration, pitch and frequency bandwidth. By waiting for a stable and sustained pitch, these drones are introduced in a way that appears to grow out of the instrumental sound.

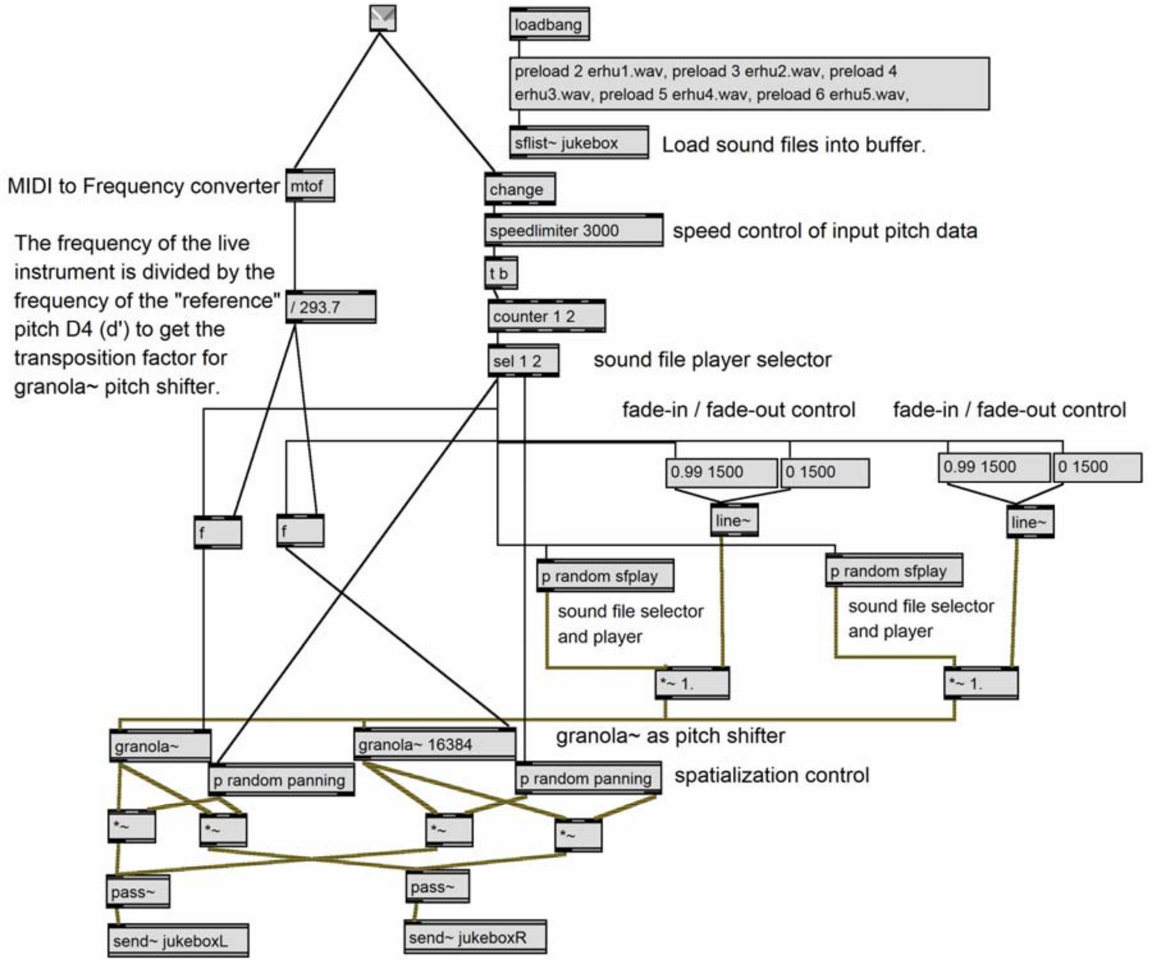
Pitch tracking to trigger pitch shifter for sample playback:

In mm.135-171 of *Snow Spell*, pre-recorded erhu music is triggered by the live erhu. These sound files are randomly selected in such a way that none is repeated until all five have been heard. In order to closely relate the pre-recorded sound files[46] to the notes that trigger them, each sound file begins at the same "reference" pitch level and is transposed to the triggering note. Once the pitch of the instrumental sound is detected, the sound file is transposed to begin on the same pitch, using the granular pitch shifter "granola~," taken from Eric Lyon's Potpourri library. Two sound file player modules play the sound files in alternation. In order to retain melodic clarity and prevent the erhu-dominated polyphonic textures from becoming too thick, a crossfade effect is applied when the next sound file is triggered before the previous one ends. Each triggered sound file is also assigned a randomly chosen panning factor to create spatial variety. The patch is shown below:

---

[46] See also Chapter 5, the section of "Elements of Chinese Music in *Snow Spell*" for further description about the musical purpose of using these sound files.

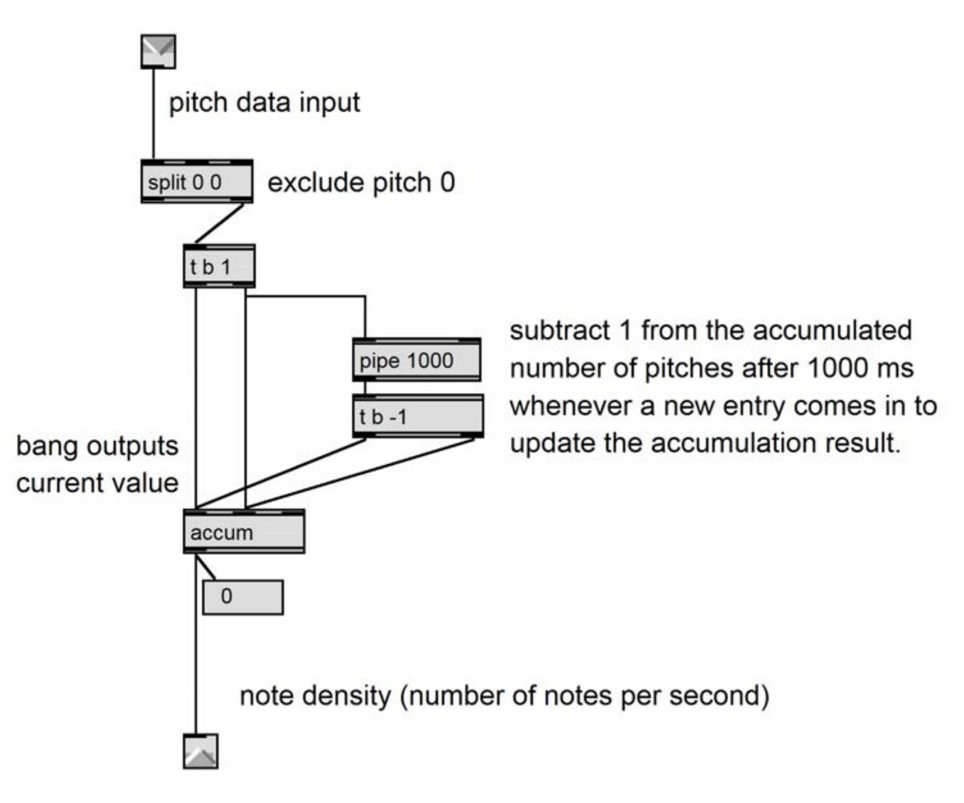
Figure 9. Pitch Tracking to Trigger Pitch Shifter for Sample Playback


loadbang
preload 2 erhu1.wav, preload 3 erhu2.wav, preload 4 erhu3.wav, preload 5 erhu4.wav, preload 6 erhu5.wav,
sflist~ jukebox
Load sound files into buffer.
MIDI to Frequency converter
mtof
change
speedlimiter 3000
speed control of input pitch data
The frequency of the live instrument is divided by the frequency of the "reference" pitch D4 (d') to get the transposition factor for granola~ pitch shifter.
/ 293.7
t b
counter 1 2
sel 1 2
sound file player selector
fade-in / fade-out control
fade-in / fade-out control
0.99 1500
0 1500
0.99 1500
0 1500
line~
line~
f
f
p random sfplay
sound file selector and player
p random sfplay
sound file selector and player
*~ 1.
*~ 1.
granola~
granola~ 16384
granola~ as pitch shifter
p random panning
p random panning
spatialization control
*~
*~
*~
*~
pass~
pass~
send~ jukeboxL
send~ jukeboxR

Multiple comparison operators of different musical elements to control spatialization:

Since the same expressive purpose may be achieved by different musical elements together or separately, several comparison operators for different musical elements may be used at the same time to trigger a signal processing module. For example, a sense of tension in instrumental music can be achieved by higher pitch range, stronger dynamics, or greater density of notes. Based on these assumptions of tension, an interactive computer music system can detect moments of tension according to the musical elements mentioned above, and further intensify them by applying DSP effects. When these real-time DSP effects are used, the combined amplitude of the processed signal and the original instrumental sound, in addition to the special texture generated by DSP effects, actually also serves to enlarge the dynamic range and heighten musical contrasts. In the *Snow Spell* patch, different thresholds are set respectively for pitch range, dynamics, and note density. Note density here refers to the number of notes per second. The note density data is obtained by counting note events in an accumulation, and then “forgetting” them (subtracting each counted item) after 1000 milliseconds. Thus, at any given time, the accumulated value is the number of note events over the last second. The note-density calculation patch is shown below:

Figure 10. Note Density Analyzer


pitch data input
split 0 0
exclude pitch 0
t b 1
pipe 1000
t b -1
subtract 1 from the accumulated number of pitches after 1000 ms whenever a new entry comes in to update the accumulation result.
bang outputs current value
accum
0
note density (number of notes per second)

When the pitch range, dynamic, or note density detected in the live instrument's signal exceeds a given threshold, it triggers a "crescendo and decrescendo" of random panning. In this Grain Panning module, the signal is divided in short grains, which are panned alternately to left and right. The distance of panning from the center is randomly chosen. The speed of granulation accelerates and then decelerates again after each trigger, heightening the sense of growing and diminishing tension.

CHAPTER 4

OTHER TECHNICAL CONSIDERATIONS

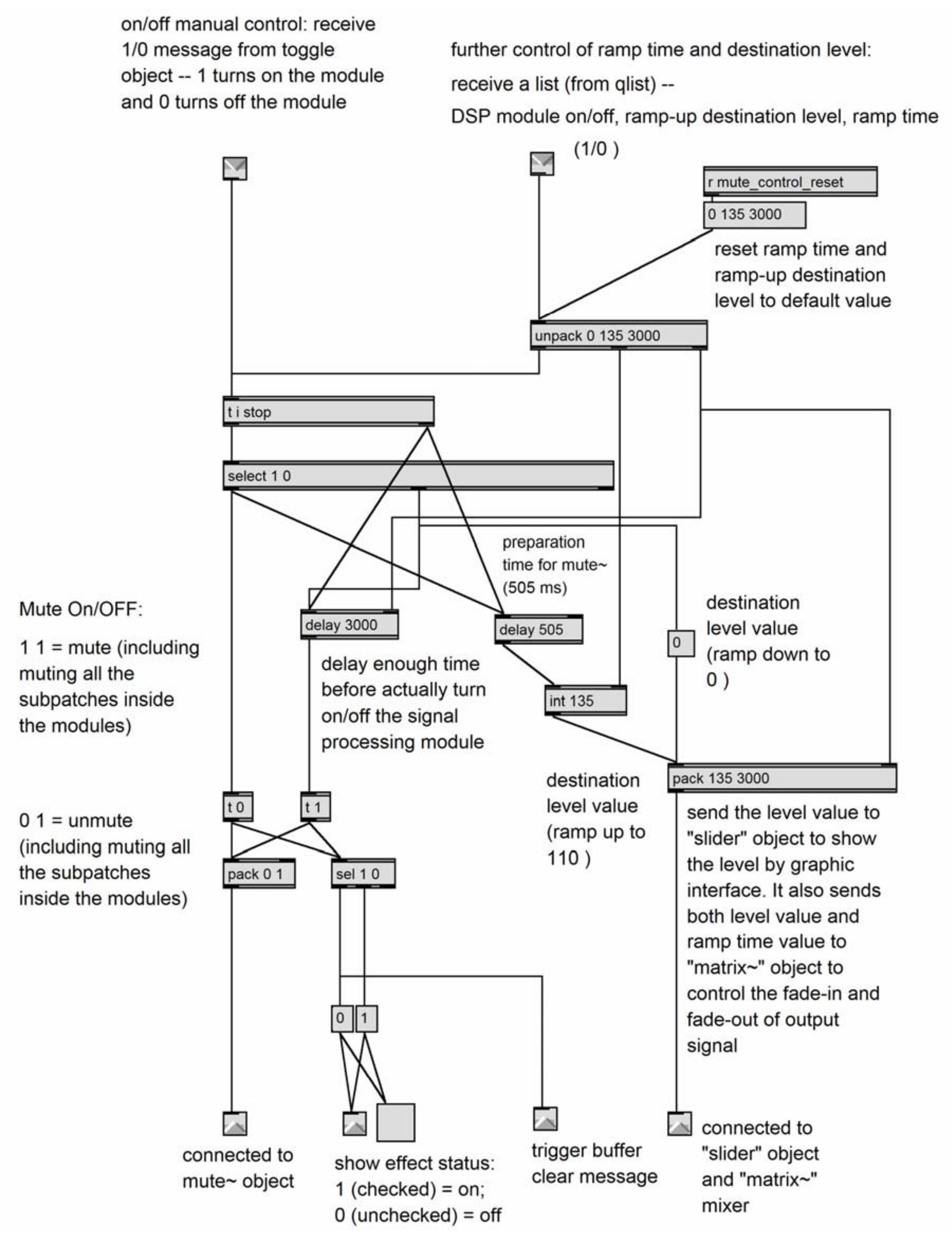
Audio Signal Routing and Mixing

A flexible real-time interactive system will necessarily have a complex set of real-time controls which demand great caution. Among various parameter controls in real-time music, volume adjustment can be problematic if the programming is not well structured. There are at least four aspects that need to be considered:

1. Clicks might happen upon turning on/off signal processing modules.
2. Clicks might happen upon switching source signals for signal processing modules.
3. A way of routing the output from one signal processing module to another may be required for musical purposes and for more efficient use of existing modules.
4. A handy volume control interface may be required for immediate balance adjustment during the live performance.

In order to reduce CPU consumption during performance, it is helpful to turn off the audio function in unused signal processing modules. However, this may cause unwanted clicks as a signal is abruptly cut to zero. Moreover, such an abrupt change could be musically disruptive. Therefore, fade-in and fade-out controls are needed. This may be achieved by including fade-in and fade-out messages in the event list whenever necessary. However, this solution may complicate the task of programming the event list by repeating the same messages again and again. A more efficient way is to program a device that automatically applies the fade-in and fade-out control to the output level of the modules whenever they receive the on or off command from the event list. This device makes the output signal fade in after the signal processing module is turned on and fade out before the module is turned off. The “mute control” subpatch shown below serves to perform this function:

Figure 11. Mute Control Subpatch



on/off manual control: receive 1/0 message from toggle object -- 1 turns on the module and 0 turns off the module

further control of ramp time and destination level:

receive a list (from qlist) --

DSP module on/off, ramp-up destination level, ramp time

(1/0)

reset ramp time and ramp-up destination level to default value

preparation time for mute~ (505 ms)

destination level value (ramp down to 0)

Mute On/OFF:

1 1 = mute (including muting all the subpatches inside the modules)

delay enough time before actually turn on/off the signal processing module

0 1 = unmute (including muting all the subpatches inside the modules)

destination level value (ramp up to 110)

send the level value to "slider" object to show the level by graphic interface. It also sends both level value and ramp time value to "matrix~" object to control the fade-in and fade-out of output signal

connected to mute~ object

show effect status: 1 (checked) = on; 0 (unchecked) = off

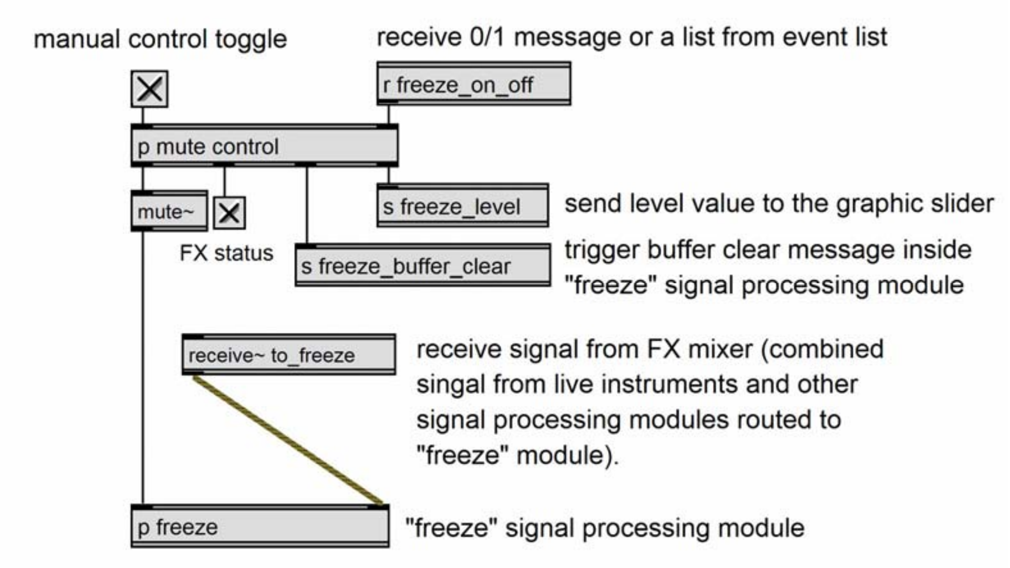
trigger buffer clear message

connected to "slider" object and "matrix~" mixer

r mute_control_reset
0 135 3000
unpack 0 135 3000
t i stop
select 1 0
delay 3000
delay 505
int 135
0
pack 135 3000
t 0
t 1
pack 0 1
sel 1 0
0
1

An example of the mute control subpatch positioned in a higher-level patch (in the “freeze” signal processing module) is shown below:

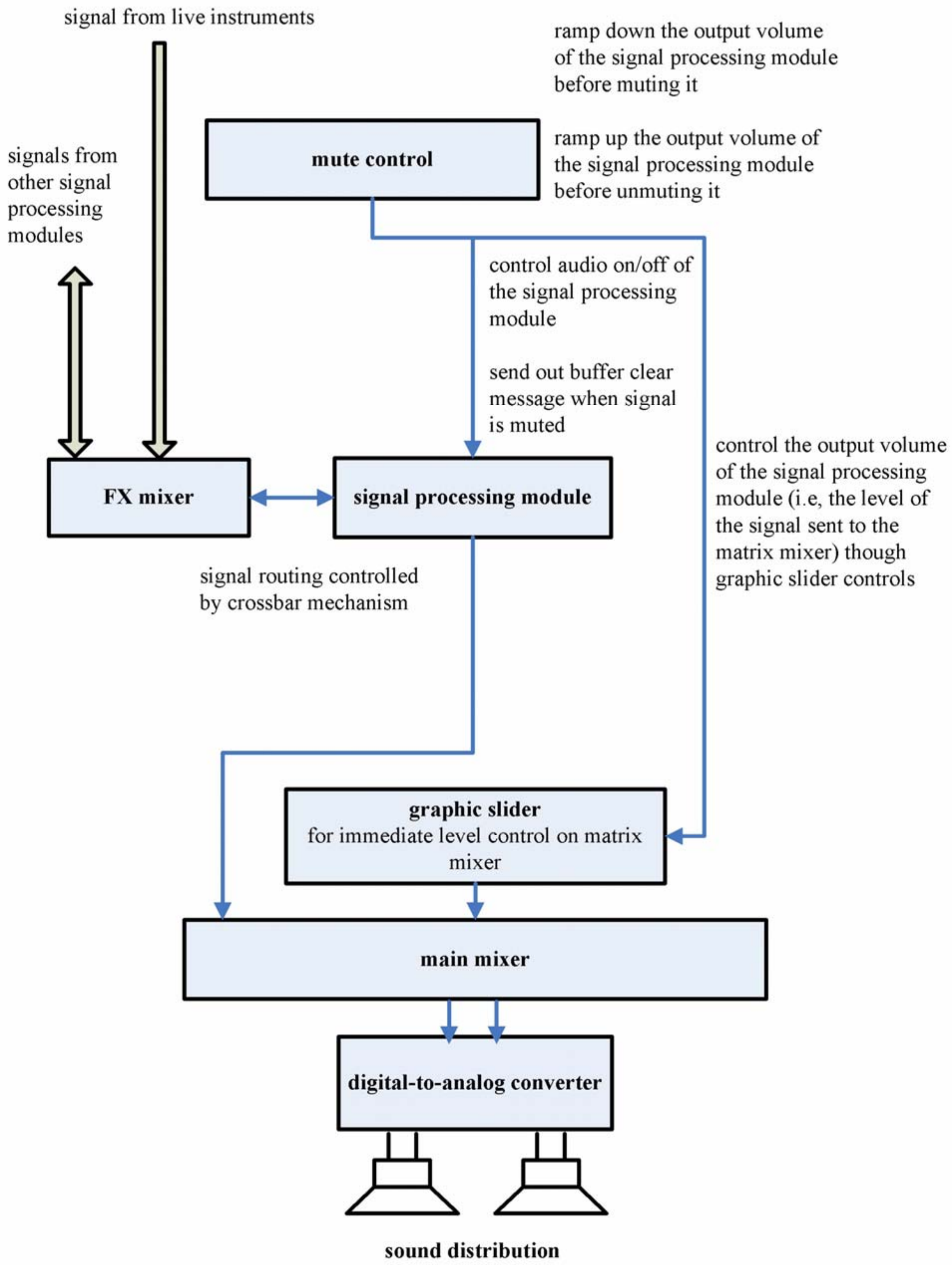
Figure 12. Mute Control


manual control toggle
receive 0/1 message or a list from event list
r freeze_on_off
p mute control
mute~
s freeze_level
send level value to the graphic slider
FX status
s freeze_buffer_clear
trigger buffer clear message inside "freeze" signal processing module
receive~ to_freeze
receive signal from FX mixer (combined singal from live instruments and other signal processing modules routed to "freeze" module).
p freeze
"freeze" signal processing module

This “mute control” subpatch can take 1 or 0 message from its left inlet to switch “mute” and “unmute” status of the connected signal processing modules. The “toggle” object connected to the left inlet is designed for direct manual control of turning on/off the signal processing modules with default or fixed ramp-up destination level and the time needed. This helps simplify the testing procedures in the composition process. The right inlet of the subpatch, on the other hand, allows control of the ramp time and the destination level in addition to “mute” and “unmute” status. All level control commands are sent to graphic sliders, to provide visual representation and manual control of current levels. These values are also sent to the matrix mixer, which executes signal switching and mixing according to the specified ramp time and level. When patches are muted and unmated, a delayed response is necessary: the “mute~” object requires about 500 ms to restore audio, so when a patch is unmuted, the ramp-up process is delayed accordingly. This avoids clicks and other artifacts.

One further protection against click artifacts is necessary: audio in delay lines, which might otherwise start abruptly, must be cleaned. This is accomplished by sending a “clear” message to “tapin~” object.

Figure 13. Audio Signal Routing and Mixing


signal from live instruments
ramp down the output volume of the signal processing module before muting it
signals from other signal processing modules
mute control
ramp up the output volume of the signal processing module before unmuting it
control audio on/off of the signal processing module
send out buffer clear message when signal is muted
control the output volume of the signal processing module (i.e, the level of the signal sent to the matrix mixer) though graphic slider controls
FX mixer
signal processing module
signal routing controlled by crossbar mechanism
graphic slider
for immediate level control on matrix mixer
main mixer
digital-to-analog converter
sound distribution

In the *Snow Spell* patch, two separate matrix mixers are used, both relying on the “matrix~” object. The master mixer receives data from the “mute control” patch to adjust the levels of outputs from the signal processing modules to the master output. Control messages from the event list can be sent anytime when necessary to control these levels in addition to the fade-in/out process upon switching mute on/off. The main concerns of designing this kind of routing is to ensure the clickless on /off switch of each signal processing module, to create a user friendly interface for immediate level control through the graphic slider during the performance when it is needed, and also to facilitate manually testing various signal processing modules in the composition process.

The other matrix mixer handles the internal routing among different signal processing modules and live instrument inputs: This mixer is controlled exclusively by event list messages, which drive Crossbar menus based on Cort Lippe’s model for controlling “matrix~”:

> Finally, using an automated signal crossbar (similar to a studio patch-bay) to connect modules to each other, signals can be sent from the output of practically every module to the input of every other module. This signal crossbar maximizes the number of possible signal paths and allows for greater flexibility when using a limited number of signal processing modules.[47]

These crossbar menus serve to facilitate the event list programming by automatically generating parameters in “matrix~” acceptable format whenever the user supplies the names of modules for connection or disconnection, the volume value, and the ramp time value. The following examples show the “matrix~” routing for various signal processing modules (and instrument inputs) in *Snow Spell*, and the crossbar menu for FFT Freezer to connect to other signal processing modules. The text in the numbered boxes connected to the inlet of “matrix~” represent signal sources connected to signal destinations (e.g. “receive~ freeze_from” represents signal source from FFT Freezer). “Crossbar” is an abstraction, a self-created patch that functions as an object in Max/MSP,[48] which takes three arguments: the name of the control message for the crossbar, the input source number, and the output destination number. For example, the crossbar marked “xbar freeze_to_rand 5 0” deals with the connection between input 5 (FFT Freezer) and output 0 (a random panning module). A crossbar receives a message consisting of the destination level value and ramp time value; it formats this as a message to matrix~ that directs these data to control the appropriate signal routing. Thus, when a “freeze_to_rand” message is sent, the matrix mixer will follow the order to connect FFT Freezer to the random panning module (or if the level is 0, disconnect it), and ramp the level to the value specified within the specified ramp time.

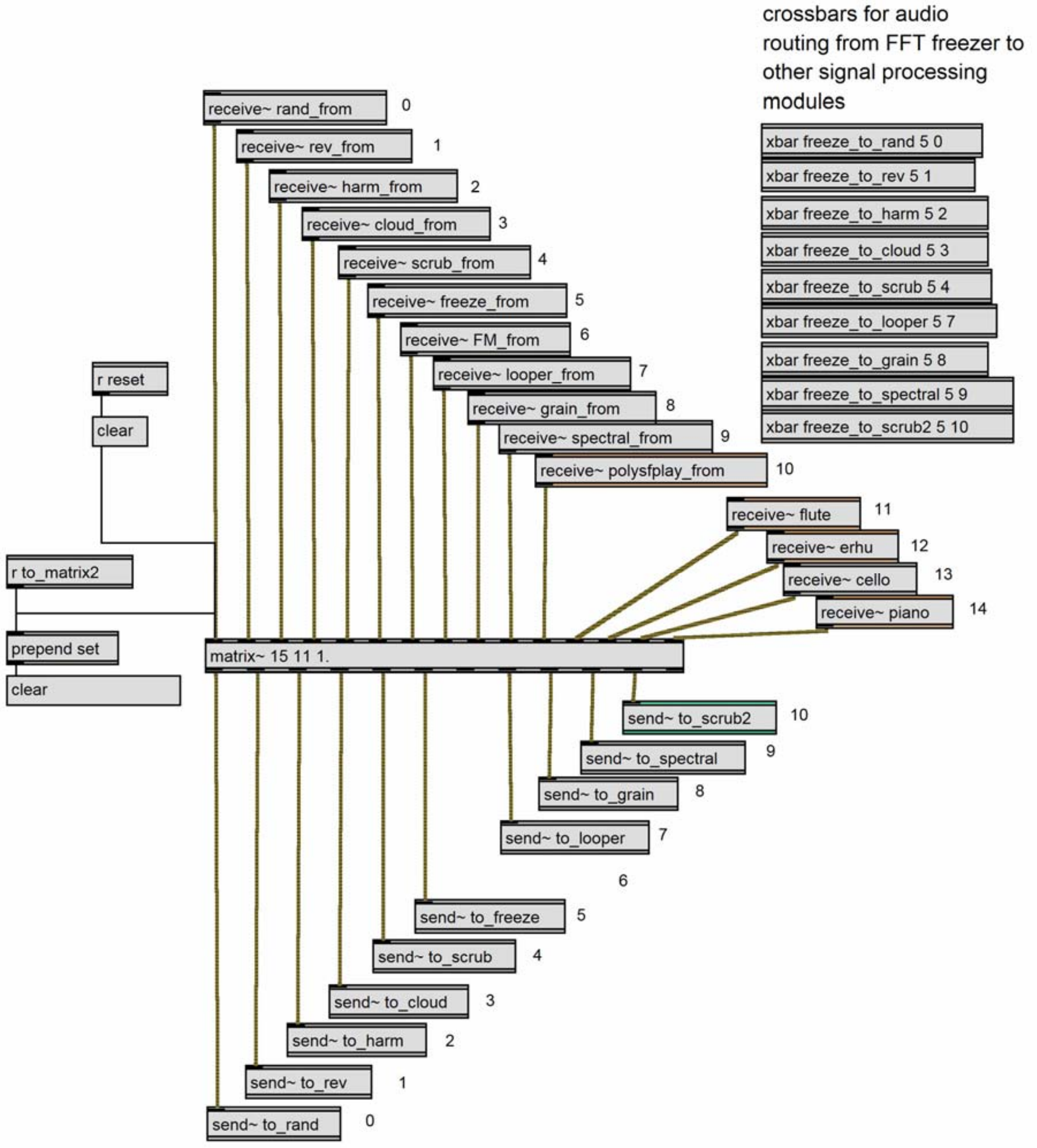
---

[47] Lippe, "Music for Piano and Computer: A Description."

---

[48] Christopher Dobrian, *Max Tutorials and Topics*, ed. David Zicarelli (San Francisco: Cycling '74, 2005), 126.

Figure 14. Crossbar and Mixing

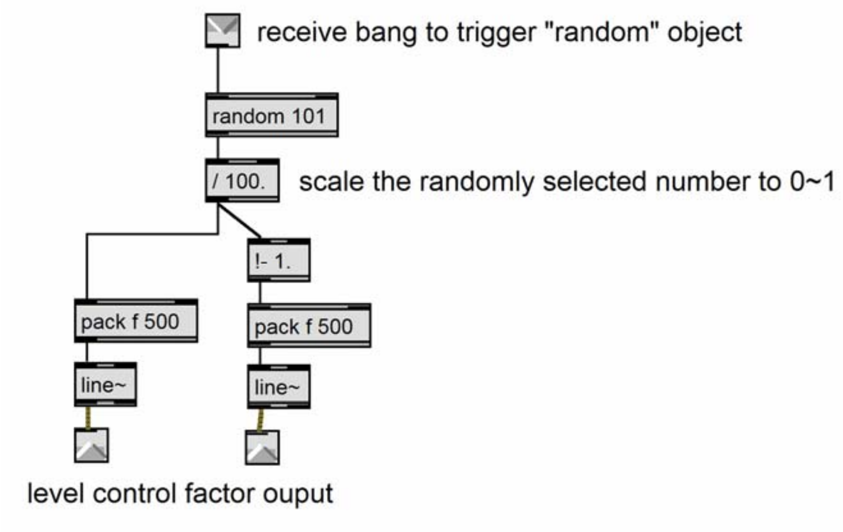

crossbars for audio routing from FFT freezer to other signal processing modules
xbar freeze_to_rand 5 0
xbar freeze_to_rev 5 1
xbar freeze_to_harm 5 2
xbar freeze_to_cloud 5 3
xbar freeze_to_scrub 5 4
xbar freeze_to_looper 5 7
xbar freeze_to_grain 5 8
xbar freeze_to_spectral 5 9
xbar freeze_to_scrub2 5 10
receive~ rand_from
0
receive~ rev_from
1
receive~ harm_from
2
receive~ cloud_from
3
receive~ scrub_from
4
receive~ freeze_from
5
receive~ FM_from
6
receive~ looper_from
7
receive~ grain_from
8
receive~ spectral_from
9
receive~ polysfplay_from
10
receive~ flute
11
receive~ erhu
12
receive~ cello
13
receive~ piano
14
r reset
clear
r to_matrix2
prepend set
clear
matrix~ 15 11 1.
send~ to_scrub2
10
send~ to_spectral
9
send~ to_grain
8
send~ to_looper
7
6
send~ to_freeze
5
send~ to_scrub
4
send~ to_cloud
3
send~ to_harm
2
send~ to_rev
1
send~ to_rand
0

The Crossbar strategy simplifies the process of event list programming. In combination with matrix~, it also helps to create a smooth transition when signal modules are connected or disconnected, and makes smooth the process of switching among different instrument inputs for the same signal processing modules. However, it does not facilitate immediate control of levels by using graphic sliders, and cannot turn off audio to reduce CPU consumption. This is the reason for the “two-mixer” solution, using two models of level control: graphical slider control for the final outputs from signal processing modules, plus mute control for each subpatch of these modules, and the crossbar abstractions for internal routing among different processing modules and instrument inputs.

Spatialization in Stereo Environment

In order to enrich the soundscape of a stereo interactive piece, a sense of physical space is of great value. In *Snow Spell*, this is accomplished by assigning different reverberation depths for different instruments and processed signals, and by adjusting panning settings between the two speakers. Panning modes used in *Snow Spell* mainly include equal-distance panning and linear panning, and are combined with randomly chosen panning factors integrated in the design of signal processing modules. The linear-crossfade random panning control is used in the FM virtual instrument and is illustrated below.

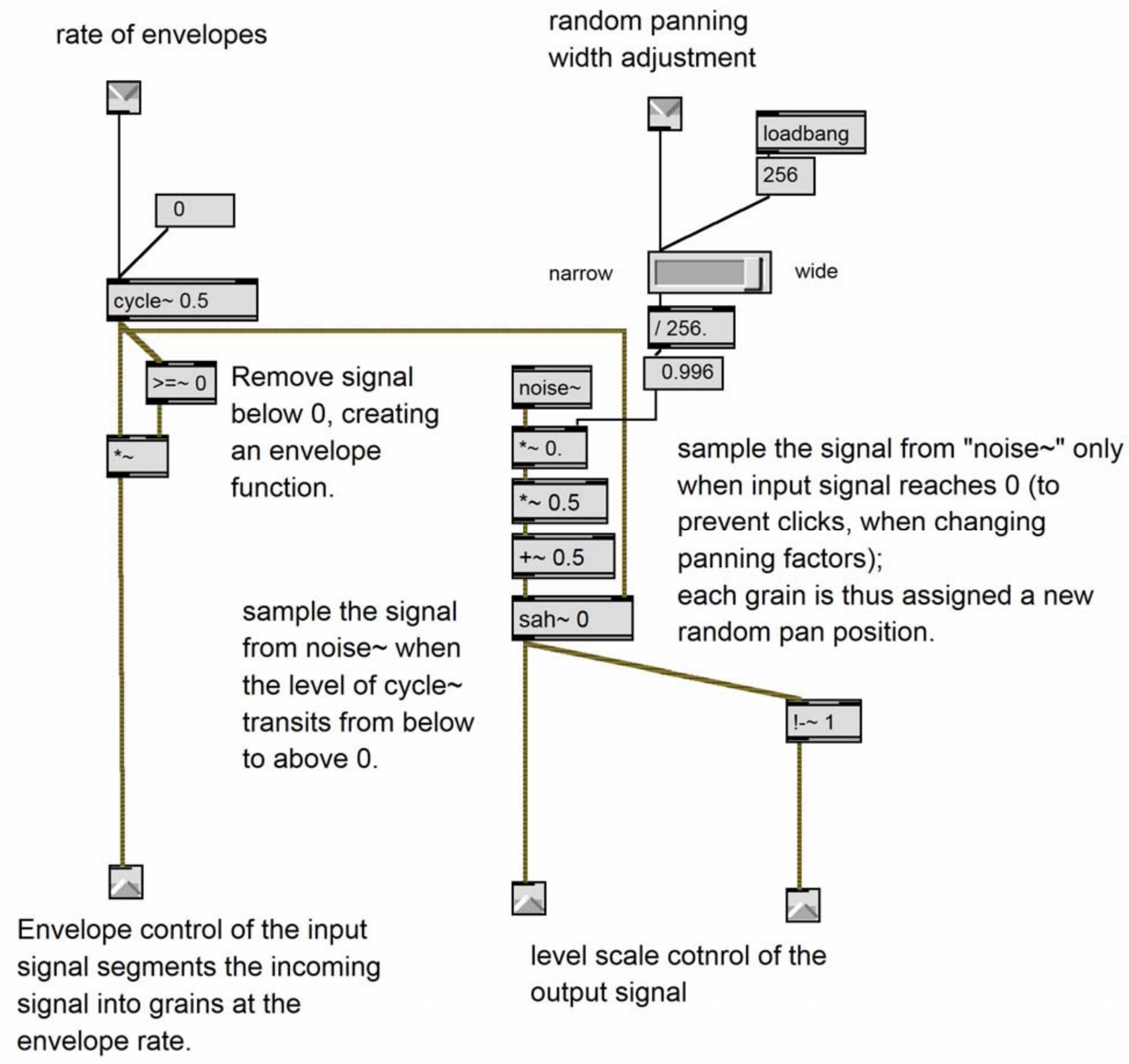
Figure 15. Random Panning in Linear Crossfade Mode


receive bang to trigger "random" object
random 101
/ 100.
scale the randomly selected number to 0~1
!- 1.
pack f 500
pack f 500
line~
line~
level control factor ouput

In this case, whenever the FM virtual instrument receives a note-on message from its compositional algorithm, it sends a “bang” message to the “random” object which generates a number as a panning factor; this number is scaled to lie between 0 and 1. This result is used as the level of one of the audio signal output in FM virtual instrument; its complement (the result subtracted from 1) will be the level of the other audio signal output. In this way, each long note generated from FM virtual instrument is assigned a different panning factor, creating notes of ever-changing timbre floating all around in a virtual space.

Random panning factors may also be triggered along with a windowing function, as used in the Grain Panning module. The subpatch that controls this process is shown below:

Figure 16. Windowed Random Panning


rate of envelopes
random panning
width adjustment
loadbang
256
0
narrow
wide
cycle~ 0.5
/ 256.
>=~ 0
Remove signal
below 0, creating
an envelope
function.
noise~
0.996
*~
*~ 0.
sample the signal from "noise~" only
when input signal reaches 0 (to
prevent clicks, when changing
panning factors);
each grain is thus assigned a new
random pan position.
*~ 0.5
+~ 0.5
sah~ 0
sample the signal
from noise~ when
the level of cycle~
transits from below
to above 0.
!-~ 1
Envelope control of the input
signal segments the incoming
signal into grains at the
envelope rate.
level scale cotnrol of the
output signal

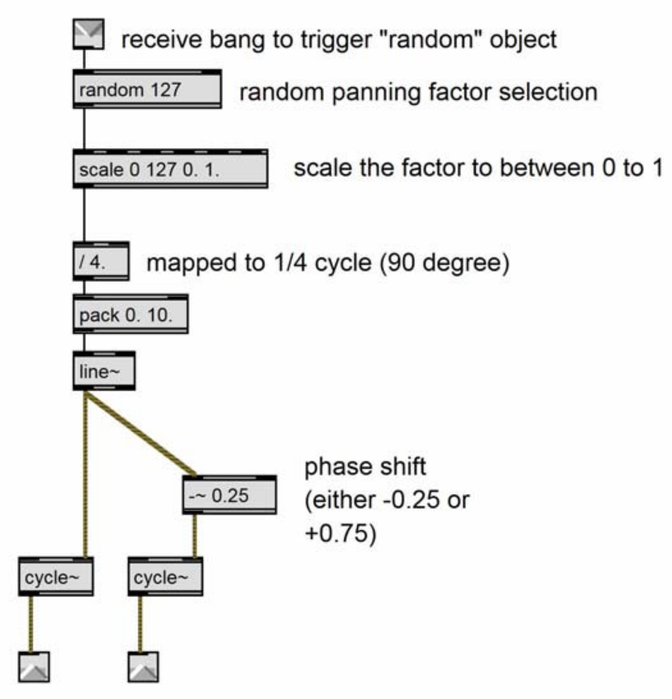
In this case, the input signal is windowed by a sinusoidal oscillator (the “cycle~” object) with its negative values removed to create a signal audible for only half the cycle. A noise generator (the “noise~” object) is used as the source of random values and is sampled when the input signal is crosses zero (i.e., when the window function ends and the signal is silenced). The sampled value from the noise generator is used as a panning factor in a linear crossfade between channels.

Because the panning factor is assigned at the precise moment when the input signal level reaches zero, no undesirable clicks are generated. This granulator patch sends window of the input signal to random positions, at a rate and intensity corresponding to the intensity of the input signal, creating a kind of modulation that adds activity and complexity to the sound.

A random panning treatment based on equal distance crossfade technique is also used in *Snow Spell*, as for example in the FFT Freezer module. It can also be combined with delay effects. In one module in *Snow Spell*, a delay is generated using the tapin~ and tapout~ objects, and the delay time is varied at random intervals. Whenever a new random delay time is assigned, a window is applied to prevent clicks, using the adsr~ object. As the new delay time is implemented, the equal-distance panning control generates levels for each audio output.

The equal-distance panning level control is based on the assumption that the two speakers are located 90 degree apart with the listener at the center. Using the Pythagorean Theorem, the relative amplitudes are set to simulate a constant distance between sound source and listener. The subpatch of the equal-distance random panning control is illustrated below:

Figure 17. Equal-Distance Random Panning



receive bang to trigger "random" object
random 127
random panning factor selection
scale 0 127 0. 1.
scale the factor to between 0 to 1
/ 4.
mapped to 1/4 cycle (90 degree)
pack 0. 10.
line~
phase shift (either -0.25 or +0.75)
-~ 0.25
cycle~
cycle~

Through these techniques, the limited opportunity for spatialization in stereo environment is used to maximal effect. The *Snow Spell* patch combines different panning trajectories and controls of panning speed to achieve a variety of imaginary spatial relationships.

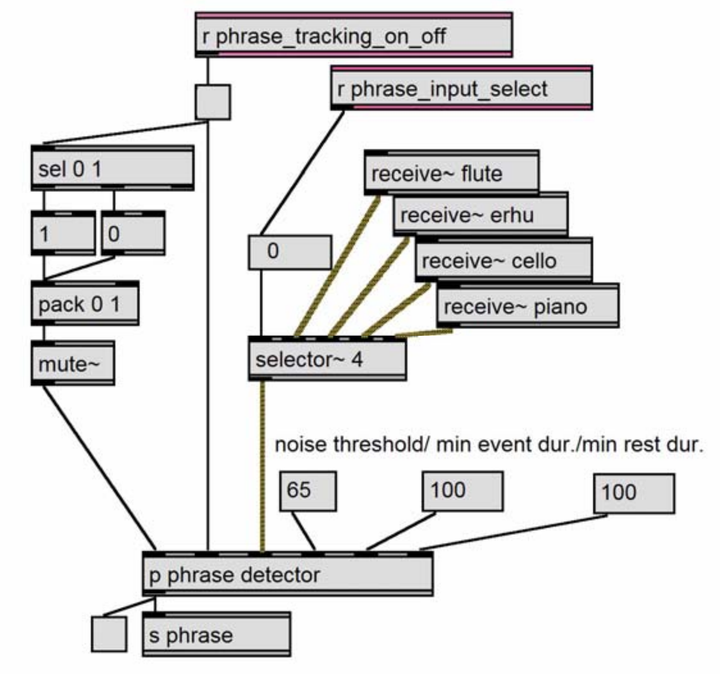
Efficiency (poly~, matrix~, mute~, pcontrol, etc.) consideration in CPU consumption

While real-time signal processing effects contribute to timbral variety, they also increase the burden on the computer’s CPU which can make the interactive music system unstable. Balance between CPU conservation and the number of real-time signal processing effects used is necessary, as well as a strategy to make the best of CPU resources.

In *Snow Spell* patch, there are three ways to reduce CPU consumption: (1) turn off MIDI or audio function in unused subpatches (using the mute-control system previously described); (2) make the best use of signal processing modules by combining them in series (using the matrix mixer previously described); and (3) remove unnecessary graphical user interfaces, or replace them with simple ones that consume less CPU.

Another approach to CPU consumption involves management of input data. For example, the fiddle~ object generates a great deal of data based on the input signal. To reduce CPU consumption, one may close the gate to data generated by fiddle~, preventing further analysis of these data. The following example shows a combination of "mute~" control and input signal selector, which is meant to turn off the audio function and to prevent any input data flow from entering the module for further CPU-consuming calculation or analysis.

Figure 18. Mute Control and Input Selector


r phrase_tracking_on_off
r phrase_input_select
sel 0 1
1
0
pack 0 1
mute~
0
receive~ flute
receive~ erhu
receive~ cello
receive~ piano
selector~ 4
noise threshold/ min event dur./min rest dur.
65
100
100
p phrase detector
s phrase

For multi-voice signal processing modules such as the sound file player, the poly-looper, and the FM virtual instrument, a special object "poly~" is used to automatically assign muted (free-for-use) subpatches for further use and to allow the audio function of unused subpatches to be independently turned off. In this way, the audio function of each subpatch can be more efficiently used.

The matrix mixer previously described also helps improve efficiency: parallel routing of DSP effects would be more CPU-consuming. In a quartet piece like *Snow Spell*, in order to clearly differentiate transformed versions of the four different instrument sounds, different instruments are usually routed to different processing networks. The four instruments can be left unprocessed, or they can be processed by different effects simultaneously, or some can be processed while others not. However, except in the case of reverberation, it will happen only rarely that more than two instruments are routed in parallel to the same effects module; such blending of signals could sound muddy and make each instrument lose its identity. A multitude of distinct processing types for each instrument would consume a great of CPU power. A similar variety and complexity can be obtained much more efficiently through serial routing of just a few signal processing modules. For example, with three modules, one can obtain seven distinct combinations – or fifteen, if the order of routing is significant.

CHAPTER 5

COMPOSITIONAL CONSIDERATIONS

Structure of *Snow Spell*

The structure of *Snow Spell* depicts four impressions of snow as envisioned by the composer. Rather than saying that each of these impressions creates a distinct section within the structure, they should instead be viewed as the origin of the composer's inspirations, upon which different musical ideas are based. In fact, the most apparent sectional divisions in the piece, based on shifts of texture, dynamic, and melodic activity, as seen at m.63 and m.99, do not coincide with the point where a new impression starts.

In the first impression of snow, the pitch D serves as the main pitch center, and is emphasized from the outset. This pitch is either elaborated by neighboring tones or combined with the pitch F to form another important motive. The most apparent example can be found in mm.6-13 where the pitch center D is elaborated with its neighbor tone E*b* or E repeated with different rhythmic figuration, and alternately presented by ehru and piano. Later in the same impression (*The First Snowflake*), the pitch center is gradually obscured even though the pitch D does come back once in a wile and is usually recognizable in the figuration that emphasizes the pitches D and F (e.g. mm.41-42 in flute).

In *Footprints Vanishing in the Storm*, the music is designed to move toward the climactic point of the piece (mainly from m.118 to m.127) with thick texture, fast pacing, and active pitch movement combined with repetitive patterns in both instrumental parts and pre-recorded sound files. One secondary climax in this passage appears around mm.59-63, reinforced by pre-recorded sound files that present a version of the piano part in mm.19-22, time-stretched through granular resynthesis. The other secondary climax appears in mm.157-162, created by an upward motion in the erhu melody combined with transformed versions of the same erhu melody in pre-recorded sound files and from the real-time granular effect.

The beginning point of the third impression of snow, *A Soliloquy in the Snow* in fact coincides with a very evident sectional division, where the tutti texture suddenly shifts to a passage dominated by the erhu, which leads to the climax in mm.159-162. This is immediately followed by the fourth impression of snow, *The Past Melts from the Window Panes*, which gradually diminishes the energy of the piece as if brings it to an end.

The pacing of *The Past Melts from the Window Panes*, compared to previous passages, is relatively static and slow, featuring repeated figures in the piano and limited pitch activity in the other instruments. An implicit return to the pitch center D can be found in the repeated piano figures and the recurrent bass pedal.

Figure 19. D as Pitch Center (mm.163-164)

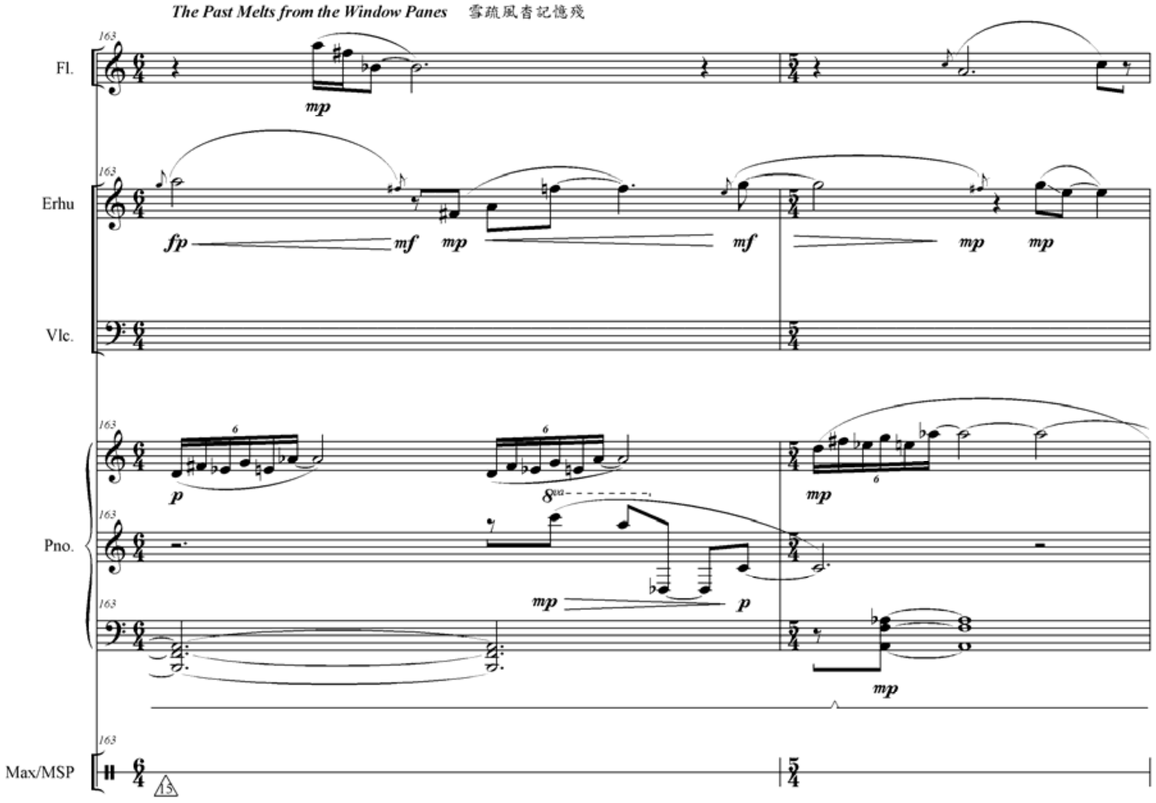
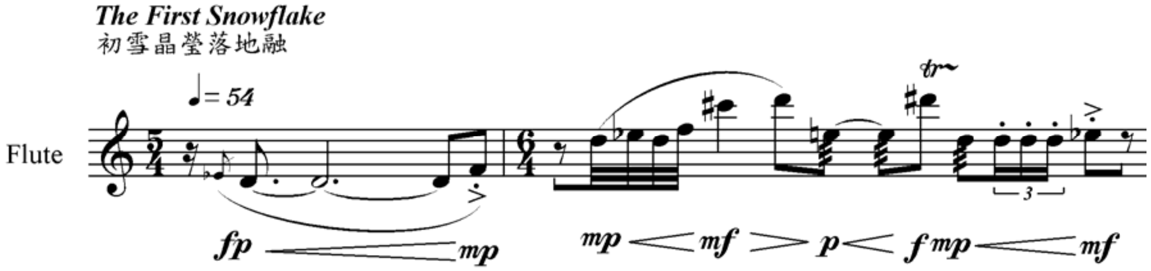

The Past Melts from the Window Panes 雪疏風杳記憶殘
Fl.
Erhu
Vlc.
Pno.
Max/MSP

Figure 20. D as Pitch Center (mm.1-2)


The First Snowflake
初雪晶瑩落地融
♩= 54
Flute

The upward melodic motive formed with the prominent pitch set (0 1 4) is emphasized again at the beginning of this passage (m.163) by the erhu as the final recall of the flute’s figure at the opening of the piece, shown in the excerpt above, here, too, the main notes, one based on the (0 1 4) set. The melodic lines of the erhu and cello later also return to the pitch center D (mm.170-171), though this is obscured intentionally by other pitches in flute and piano to avoid an excessively evident tonal center.

Elements of Chinese Music in *Snow Spell*

The elements of Chinese music incorporated in *Snow Spell* mainly include the use of idiomatic expressions from traditional Chinese erhu music, and references to the Chinese folk tune, *River of Sorrow*.

The erhu, known as the “Chinese violin”, is a two-stringed fiddle with a small resonator body covered with python skin on the front (playing) end. It does not have a fingerboard, which makes its performance demanding but also allows more flexibility through finger pressure control. The bowhair passes between the two strings. The bow is removed from the instrument only when replacing the strings, bowhair or bow. It belongs to the huqin family of Chinese bowed string instruments. Huqin came into China through the Mongolian conquest of the thirteenth century and has become the most popular stringed instrument in China.[49] It is mainly used in Chinese folk music. Chinese folk music, known as Su Yueh, is very different from Ya Yueh, which was the music of royal ritual. While Ya Yueh is more “elegant, polished, refined, and polite,” Su Yueh is full of “turns, grace notes, glissandos, portamentos, and excessive rhythmic variation and vitality.”[50] Since erhu is an instrument used mainly for Chinese folk music, it bears the same feature as other Chinese folk instruments, serving as an extension of vocal music. The following is quoted from Bliss Wiant:

> Both vocal and instrumental music have had important functions in China. Instrumental melody does not differ fundamentally from vocal music except, of

---

[49] Bliss Wiant, *The Music of China* (Hong Kong: Chung Chi Publications, 1965), 96.

[50] Ibid., 2-3.

> course, in range, but even in this respect many instruments are limited to the approximate range of the human voice, say three octaves or less. The greatest single factor in instrumental music is that of timbre.[51]

Regarding the tradition of Chinese folk songs, Bliss also wrote:

> Such songs are common property of the folk and are so full of grace notes, portamentos, and embellishments that no notation has ever been invented competent to represent all the nuances which greet the ear. The texts of many of these songs have been assembled……But no musical scores have been made.[52]

The erhu and other melodic instruments for Chinese folk music have developed various techniques of imitating human singing, and the use of these techniques also varies according to the music style in different geographical areas. One of the most important factors is a wide range of glissandi techniques. However, in the old Chinese music notation system, Gongchepu, these nuances are not notated and only the nuclear notes (primary notes) are notated. Even in the newly and widely used notation system for traditional Chinese music, Jianpu, introduced from Japan at the end of the Ching Dynasty, [53] these nuances are still not fully notated. On one hand, there is no way to notate all the nuances; on the other, it is rooted in this tradition to give the performer a great deal of freedom and flexibility for personal interpretation.

According to the most popular and authoritative erhu pedagogical book Shao Zheng's *Ehru Broadcasting Instruction*, glissando techniques for ehru can be mainly divided into seventeen kinds belonging to five different categories. The following is the author's translation and summary of Zhang's categorization:[54]

---

[51] Ibid., 5.

[52] Ibid., 97.

[53] Shao Zhang, *Erhu Broadcasting Instruction* (Shanghai: Shanghai Music Publication, 1980), 6.

- I. Widely used glissando techniques:
    - a. Small glissando:
        - i. Upward glissando (approximately at the interval of third)
        - ii. Downward glissando (approximately at the interval of third)
        - iii. Glissando after the main note
        - iv. Continuous glissando (over two or more main notes)
        - v. "Finger-carrying" glissando upon changing fingering position: This effect is similar to continuous glissando but more subtle. It is used at performer's discretion and is not notated in a music score.
        - vi. Half-step glissando (usually in downward motion)
        - vii. "Round tone" glissando: glissando of pitch interval between one half to one third of a half step
    - b. Large glissando:
        - i Upward glissando
        - ii Downward glissando
    - c. glissando turn
        - i. Upward glissando turn: Quickly move from the main note to an upper note and then back to the main note.

---

[54] Ibid., 87-94.

        ii. Downward glissando turn: Quickly move from the main note to a lower note and then back to the main note.

II. Special glissando techniques:

  a. multiple-finger glissando:

    i. Regular multiple-finger glissando: Put two or three fingers on the string and slide the higher-pitch finger towards the lower-pitch finger, releasing them gradually to let the lower pitch be heard.

    ii. Multiple-finger glissando turn: Combine the techniques of double-finger glissando and glissando turn.

    iii. Pressed-string glissando (by pressing the string harder to change the pitch)

  b. Glissando in combination:

    i. Trill glissando: Perform the trill and glissando simultaneously

    ii. Glissando with pressed vibrato (vibrato produced by rapidly changing finger pressure on the string)

    iii. Tremolo glissando: Perform the glissando and bowed tremolo simultaneously.

The glissando techniques in Zhang's modern edition of erhu music do not always specify a definite pitch interval; sometimes they are notated approximately with a slanted line or a curve plus an arrowhead. As mentioned before, it is intended to allow more flexibility in performance. In this case, it also implies that a clear attack of the pitch starting or ending the glissando is not preferred (unless these pitches are specified in the score).

In *Snow Spell*, the composer not only incorporates these glissandi nuances to evoke Chinese cultural resonances, but also to make the erhu music sound different from traditional Chinese music by putting it in a relatively atonal context, mixing different cultural elements into a unified composition. In addition to some glissando notation ideas drawn from *Erhu Broadcasting Instructions*, the composer also take ideas from Kurt Stone's notation book[55] to differentiate "into-a-note" glissandi from "out-of-a-note" glissandi by using different notation symbols.[56] However, just as in the Chinese tradition in which many nuances can not and should not be fully notated, the composer also expects the erhu player to add his or her own glissando or portamento in addition to those of particular importance, which are notated in the score.

Besides glissandi nuances, the music *Snow Spell* also refers to a Chinese folk tune *River of Sorrow*. *River of Sorrow* was originally a solo piece drawn from an old Chinese opera score. It was written for guanzi, a Chinese double-reed wind instrument, and became popular as a folk tune in the northeast of mainland China. It was rearranged by Chang-An Zhu and Xin-Shan Gu in 1950s and was later adapted as an erhu piece by Hai-Huai Huang.[57] Before the rearrangement by Zhu and Gu, the tune of *River of Sorrow* was used both in rituals of celebration and of mourning. However, after it was arranged, it became a very sad piece describing the sorrow and resentment of a widow against the governing class, crying over the river bank for her husband who was conscribed and killed.[58] The introductory passage of *River of Sorrow* was added when it was rearranged, and serves as a summary of the materials and the emotion of the whole piece.[59] Therefore, it is referred to in *Snow Spell* both to evoke sorrowful feelings and to embody a multitude of tragic episodes in Chinese history.

---

[55] Kurt Stone, *Music Notation in the Twentieth Century* (New York: W. W. Norton & Company, 1980), 20.

[56] Kurt Stone describes these short "into-a-note" and "out-of-a-note" glissandi as quick, short slides or portamento; in this he differs from Samuel Adler's definition for portamento as "a natural, expressive way of connecting melody notes which are a great distance apart and [which] is rarely indicated in the score."

[57] Jiu-Sheng Yang, "The Origin and Development of the Music 'River of Sorrow'," in *Anthology of Muisc Research Paper*, ed. Yong-Qi Peng (Shanghai: Shanghai Musicology Publication, 2005), 266-67.

Reference techniques in music composition may be divided into two categories, allusion and quotation. Quotation is regarded as stronger than allusion. A quotation is often explicitly identified and is intended to be recognized by its audience, while allusion is reference of a more covert kind.[60] In *Snow Spell*, the composer refers to *River of Sorrow* by allusion, segmenting the original introductory melody of *River of Sorrow*, shifting the segments to different pitch levels, making use of the most recognizable materials in *River of Sorrow* (in particular the first four-note figuration), and distributing the derived materials (e.g., grace-note figures and the interval of a third) throughout the piece. The most explicit reference to *River of Sorrow* can be found in mm.135-144.

The following is a comparison between the introductory excerpt from the original version of *River of Sorrow* notated in Jianpu; the western notation for the original Jianpu version of *River of Sorrow*; the transcribed version from erhu master Hui-Fen Min's performance[61] recording of *River of Sorrow*; and the references to *River of Sorrow* in *Snow Spell* in mm.135-144.

---

[58] Ibid.

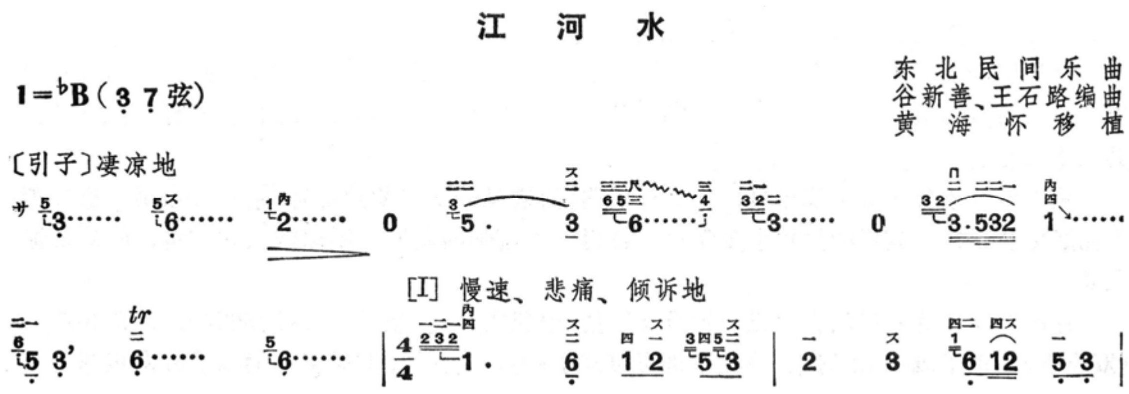
[59] Ibid.

[60] Nicholas Marston, *Schumann Fantasie Op.17*, Cambridge Music Handbook (Cambridge: Cambridge University Press, 1992), 35.

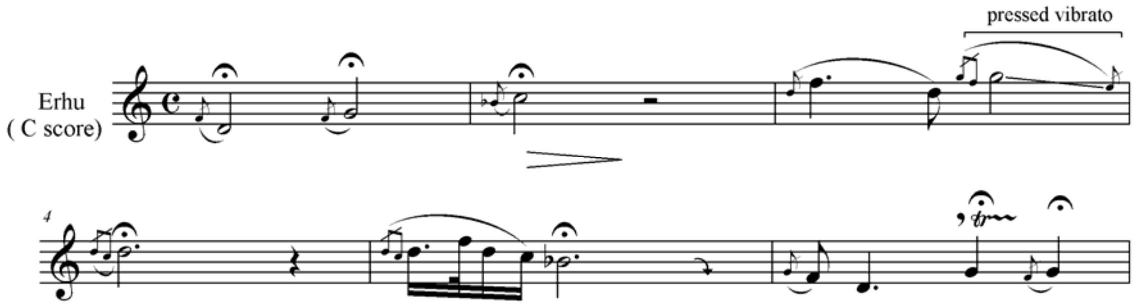
[61] Hui-Fen Min, *River of Sorrow*, Immortal Chinese Instrumentals Album (Washington: Lasting Impression Music, 2002).

Figure 21. Comparison between different versions of *River of Sorrow*

The Introduction of *River of Sorrow* Notated in Jianpu System


江 河 水
1=♭B（3 7弦）
东北民间乐曲
谷新善、王石路编曲
黄海怀移植
〔引子〕凄凉地
[I] 慢速、悲痛、倾诉地
tr

The Author’s Translation of the Jianpu Notation above into Western Notation


pressed vibrato
Erhu
(C score)
4
tr

The Author’s Transcription from Erhu Master Hui-Fen Min’s Performance Recording

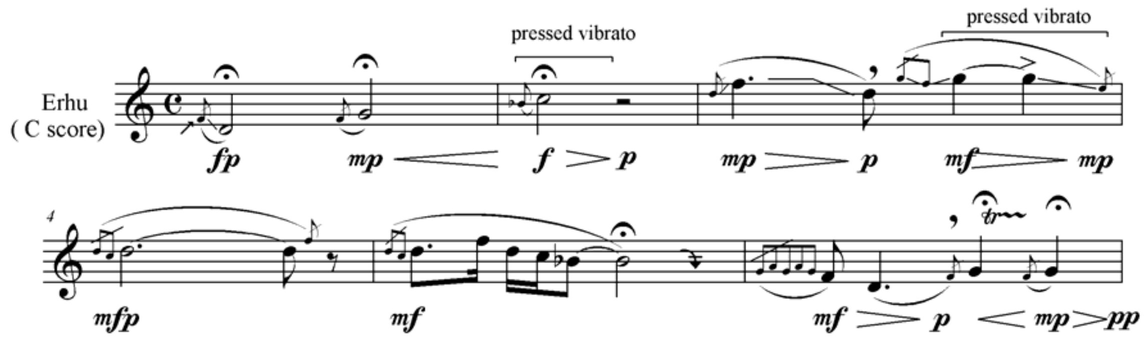
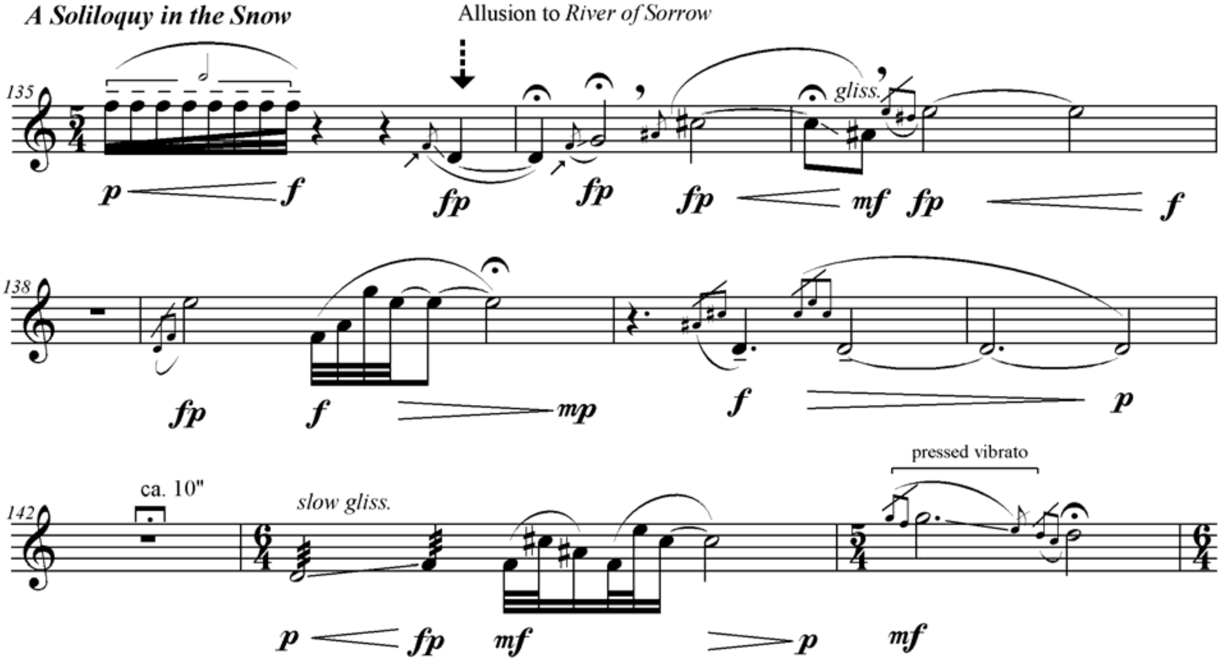

pressed vibrato
pressed vibrato
Erhu
(C score)
fp
mp
f
p
mp
p
mf
mp
4
tr
mfp
mf
mf
p
mp
pp

Figure 22. Allusions and Quotations of River of Sorrow in *Snow Spell* (quotations are found in mm.135-144):


A Soliloquy in the Snow
Allusion to River of Sorrow
135
gliss.
p
f
fp
fp
fp
mf
fp
f
138
fp
f
mp
f
p
142
ca. 10"
slow gliss.
pressed vibrato
p
fp
mf
p
mf

As mentioned above, traditional Chinese music scores generally notate only the nuclear notes and leave the grace notes, glissandi, and portamento for the performer's interpretation. The newly edited score from *Erhu Broadcasting Instructions*, even though adding some grace note and special glissando still does not and cannot notate all these nuances. As shown in the excerpts above, Min's interpretation includes extra glissandi in m.1 and m.3, an extra pause in m.3, and extra "out-of-a-note" grace-notes in m.4 and m.6. This tradition of decoration is stated by Stephen Jones as follows:

> Having established the nuclear notes of a melody, written in the score and embedded in the musician's head, it remains to give flesh and blood to the skeleton. Early in the learning process of chanting the score, musicians begin to add simple decoration to the basic framework. This obligatory practice is known by names such as jiahua ('adding decoration': cf. the standard translation 'adding flowers'), akou ('ah-

> mouthing'), henghe ('humming and ha-ing'), or caiqiang ('embellishing the melody').[62]

Therefore, a piece may sound very different from one performer to another and from one geographical area to another. The performer's taste and the geographical style have a great deal of impact on the way the same piece is played. The erhu master, Hui-Fen Min's recording of *River of Sorrow* has been widely regarded as authoritative since 1970s; her interpretation of this piece can almost be viewed as the "standard" version. The composer therefore adopts ideas from her version of the ornaments and glissandi nuances as the model for his references to *River of Sorrow* in *Snow Spell*. In addition to Hui-Fen Min's personal glissando interpretation, the composer also adopts her special "out-of-a-note" grace-note figure in his piece. This special grace-note figure at the end of the main note with the interval of upward third is used not only as a decoration in *Snow Spell* but also as an emphasized motive appearing in m.1 (flute and piano), mm.3-4 (flute), m.17 (erhu), m.26 (cello), m.41 (flute), and etc. This kind of decoration can be traced back to the idiomatic playing from Chinese wind instrument. Since *River of Sorrow* is originally a piece for guanzi, such figuration is used by Hui-Fen Min to imitate the effect of guanzi. In my piece, it serves as a central motive. When it is used as a grace note (which is expected to be lighter and shorter), it serves both as the reiteration of this motive and as a coloristic effect intended to evoke Chinese resonances.

In addition to the use of references to Chinese folk music in the acoustic instrumental part, the melody of *River of Sorrow* is also segmented and loaded into the computer as samples to be triggered by the live erhu performance in the fourth section of the piece, *A Soliloquy in the Snow*. These pre-recorded melodic segments are transposed in real-time and are triggered in a random

---

[62] Stephen Jones, *Folk Music of China - Living Instrumental Traditions* (New York: Oxford University Press Inc., 1995), 126-27.

order according to the pitch change in the live erhu performance. They are spaced apart by at least three seconds, and transposed to the same pitch level as the live erhu sound that triggers them. If the next sound file is triggered before the previous one ends, a crossfade effect is applied. Each triggered sound file is assigned a randomly chosen panning factor in order to create spatial variety. In this way, the traditional Chinese folk melody is obscured through its polytonal expression against a written-out allusion to the same folk music played live by erhu. The quickly changing segments of the pre-recorded sound serve to imply a quick flash of memory while the live erhu allusions to *River of Sorrow* serve to imply the current self, absorbing and internalizing the past to become a new self.

Through these extra-musical associations, the music goes beyond the immediate passage of time and unites the past and present moment into one. However, the use of cultural references in music requires the listeners' familiarity with the culture that is invoked. Some of the Chinese music elements mentioned above may only be effective for those who are familiar with Chinese musical culture. Even so, the way these materials are organized within the domain of the work still makes them coherent within the structure of the piece. The work is more emotionally resonant to a listener familiar with the references to Chinese tradition, but the flow of impressions and emotions in the work does not rely on such familiarity.

CHAPTER 6

CONCLUSION

Most interactive pieces are solo works. *Snow Spell*, a quartet with interactive computer music system, thus presents an unusual challenge. There are few models and resources for such a project. During the processes of design and programming, the composer attained much greater control and sophistication in composing interactive music. This led to a unique personal model for structuring an interactive piece in Max/MSP for multiple instruments. The experiments on further control of DSP effects, and the attempt to optimize audio signal routing among multiple signal processing modules, contributed to new possibilities for further interactive work.

In addition to the technical issues, there is a fundamental difference between purely instrumental composition and interactive computer music composition: the latter requires a great deal of time for experimenting with various DSP effects, their results with various instrumental sounds, and their opportunities for algorithmic controls in order to find an ideal direction for the music. While instrumentation and orchestration have been studied by traditional composers in relatively systematic ways, there is no organized documentation of the ways instrument sounds can be processed effectively and efficiently by real-time DSP effects to achieve particular results. However, the variety of timbre and texture in electroacoustic music can be seemly unlimited and the results can be fresh and surprising; it is a fertile field for further research, study, and experimentation. Experiments in combining different DSP effects and controls for interactive music are not unlike the study of instrumentation and orchestration for acoustic music composers. Nevertheless, the former might be even more challenging because of the way they can mingle musical, technical, and scientific tasks as the composer explores new audio technologies. This challenge affects the opportunity for expression in such music: great care must be taken to go beyond the technological to the artistic domain.

As an electro-acoustic music composer focusing on interactive music, the author believes there will be endless challenges in the future in searching for new expressions through technology and tradition at the same time. These challenges have kept many contemporary composers away from interactive technology. Nevertheless, the opportunities for imagination, and the surprising results that can be obtained, justify the effort.

BIBLIOGRPHY

Cycling’74. *Max/MSP*. http://www.cycling74.com (accessed March 1, 2007).

Dannenberg, Roger. "An on-Line Algorithm for Real-Time Accompaniment." In *The Proceedings of International Computer Music Conference*, P. 187. Paris, France, 1984.

Dobrian, Christopher. *Max Tutorials and Topics*. Edited by David Zicarelli. San Francisco: Cycling '74, 2005.

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

Garton, Brad, and Dave Topper. *RTcmix*. http://music.columbia.edu/cmc/RTcmix (accessed March 1, 2007).

Hyde, Jerald R. "The Concert Hall Experience: How Visual Stimuli Can Affect What We Hear " In *Acoustical Society of America 147th Meeting*. New York, 2004.

IRCAM (Institut de Recherche et Coordination Acoustique/Musique). *Real-Time Software Pack*. http://www.ircam.com (accessed March 1, 2007).

———. *Jmax*. http://freesoftware.ircam.fr. (accessed March 1, 2007)

Jones, Stephen. *Folk Music of China - Living Instrumental Traditions*. New York: Oxford University Press Inc., 1995.

Symbolic Sound Corporation. *Kyma*. http://www.symbolicsound.com (accessed March 1, 2007).

Lippe, Cort. "Music for Piano and Computer: A Description." *Information Processing Society of Japan SIG Notes* 1997, no. 122 (1997): 33-38.

Lyon, Eric. *Potpourri*. http://eamusic.dartmouth.edu/~eric/MAX (accessed March 1, 2007)

Marston, Nicholas. *Schumann Fantasie Op.17*, Cambridge Music Handbook. Cambridge: Cambridge University Press, 1992.

Maurer, John A. *Interactive Electronics in Computer Music*, 1999. http://ccrma.stanford.edu/~blackrse/interactive_electronics.html (accessed March 1, 2007)

May, Andrew. "Philippe Manoury's Jupiter." In *Analytical Methods of Electroacoustic Music* edited by Mary Simoni, 145-85. New York: Routledge, 2006.

McCartney, James. Supercollider. http://www.audiosynth.com (accessed March 1, 2007).

McNutt, Elizabeth. "Performing Electroacoustic Music: A Wider View of Interactivity." *Organized Sound* 8, no. 3 (2003): 297-304.

Min, Hui-Fen. *River of Sorrow*, Immortal Chinese Instrumentals Album. Washington: Lasting Impression Music, 2002.

Morgan, Robert P. *Twentieth-Century Music: A History of Music Style in Modern Europe and America*. 1st ed. New York: W.W. Norton, 1991.

"Orchestration." In *The New Harvard Dictionary of Music*, edited by Don Michael Randel. Cambridge: Belknap Harvard, 1986.

Puckette, M. and C. Lippe. "Score Following in Practice." In *Proceedings of the International Computer Music Conference*, 182-85. San Francisco, 1992.

Puckette, Miller. *fiddle~*. http://www.akustische-kunst.org/maxmsp/other.html (accessed March 1, 2007)

———. "Max at Seventeen." *Computer Music Journal* 26, no. 4 (2002): 31-43.

———. "Score Following Using the Sung Voice." In *The Proceedings of International Computer Music Conference*, 199-200. Canada, 1995.

Rowe, Robert. *Interactive Music Systems*. Cambridge, Mass., London: MIT Press, 1994.

———. "Interactor 4.0.8." *Leonardo Music Journal* vol. 2, no. No. 1 (1992): 122-23.

Schwartz, Elliott and Daniel Godfrey *Music since 1945: Issues, Materials, and Literature*. New York: Schirmer Books, 1993.

Stone, Kurt. *Music Notation in the Twentieth Century*. New York: W. W. Norton & Company, 1980.

Vercoe, Barry. *Csound*. http://www.csounds.com (accessed March 1, 2007).

———. "The Synthetic Performer in the Context of Live Musical Performance." In *The Proceedings of International Computer Music Conference*, P. 185. Paris, France, 1984.

Wiant, Bliss. *The Music of China*. Hong Kong: Chung Chi Publications, 1965.

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

Yang, Jiu-Sheng. "The Origin and Development of the Music 'River of Sorrow'." In *Anthology of Music Research Paper*, edited by Yong-Qi Peng. Shanghai: Shanghai Musicology Publication, 2005.

Zhang, Shao. *Erhu Broadcasting Instruction*. Shanghai: Shanghai Music Publication, 1980.

PART II

COMPOSITION

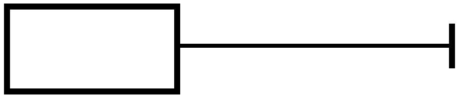
- **Title: *Snow Spell*** (Quartet for Erhu, Flute, Cello, and Piano)
- **Duration:** ca. 18 min.
- **Finished Date:** March 15, 2007
- **Program Notes:**

This one-movement piece, *Snow Spell* for erhu, flute, cello and piano, is intended to depict the beauty of a snow scene mixed with eastern cultural association and related personal fantasy by presenting four different impressions of snow through music, *The First Snowflake*, *Footprints Vanishing in the Storm*, *A Soliloquy in the Snow*, and *The Past Melts from the Window Panes*. The four impressions of the snow scene will be presented in order in music through different textures, timbres, and temporalities. Transitions will obscure the demarcation between the four different sections. Pentatonic scales, traditional erhu glissandi, embellishment, and pizzicato techniques, and quotations from Chinese folk tune "River of Sorrow" are incorporated to evoke Chinese folklore as a context for the images of snow. The pitch structure focuses on the interval of a third, used both diatonically and chromatically in (0 1 4) or (0 1 4 5) pitch sets in order to combine the elements of eastern pentatonic materials, and western atonal techniques as a whole.

- **Performance Instructions:**

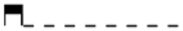
General:

1.



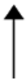
Repeat the pattern in the box until the end of the extender.

2.



Let the bow bounce on the strings elastically, as in ricochet bowing.

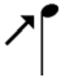
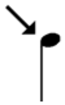
3.



Highest possible pitch on a string of erhu or cello

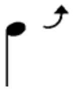
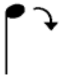
Glissando for erhu and cello:

1.

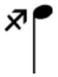
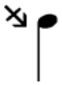
 or 

A short glissando into the main note at an approximate interval of a third with the initial attack intentionally obscured by softer dynamics

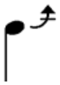
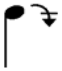
2.

 or  A short glissando out of the main note at an approximate interval of a third

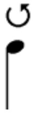
3.

 or  A short half-step glissando into the main note with the initial attack intentionally obscured by softer dynamics

4.

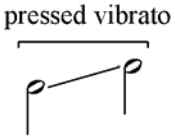
 **or**  A short half-step glissando out of the main note

5.



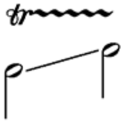
A fast “glissando turn” from the notated pitch to a whole step downward and then back to its original pitch

6.



Glissando with pressed vibrato (vibrato produced by rapidly changing finger pressure on the string)

7.

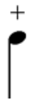


Trill glissando: Perform the trill and glissando simultaneously

Pizzicato for erhu

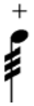
Even though Erhu pizzicato is produced by two strings together, the pizzicato in the score only notates the pitch of the inner string.

1.



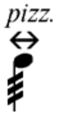
Left-hand pizzicato.

2.



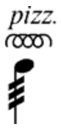
Left-hand pizzicato roll achieved by making a backwards and forwards motion with the middle finger.

3.



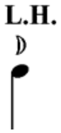
Right-hand single-finger pizzicato roll achieved by plucking erhu strings back and forth with the index finger. For better result of this performing technique, the performer may let the hand pivot on the rim of the snake skin on erhu when plucking the strings.

4



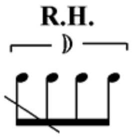
Right-hand multiple-finger pizzicato roll achieved by plucking erhu strings with four fingers (except the thumb) alternatively

5.

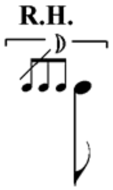


Left-hand single-finger fingernail flick achieved by flicking the middle finger against the string.

6.

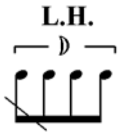


or

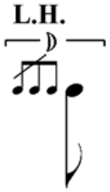


Right-hand four-finger fingernail flick achieved by flicking the little finger, ring finger, middle finger, index finger in order (with the thumb) against the string.

7.

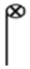


or



Left-hand four-finger fingernail flick achieved by flicking the little finger, ring finger, middle finger, index finger in order against the string.

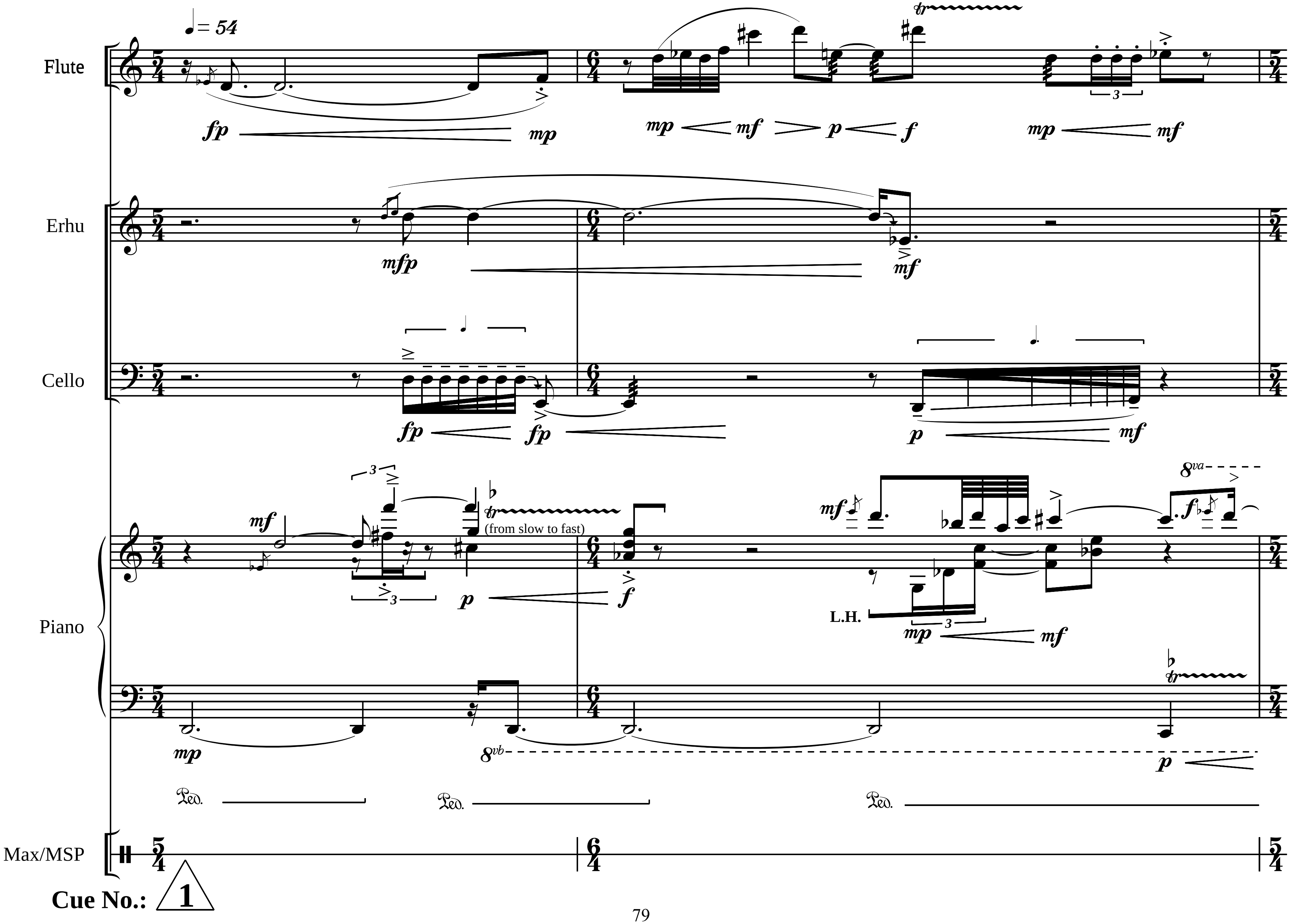
8.

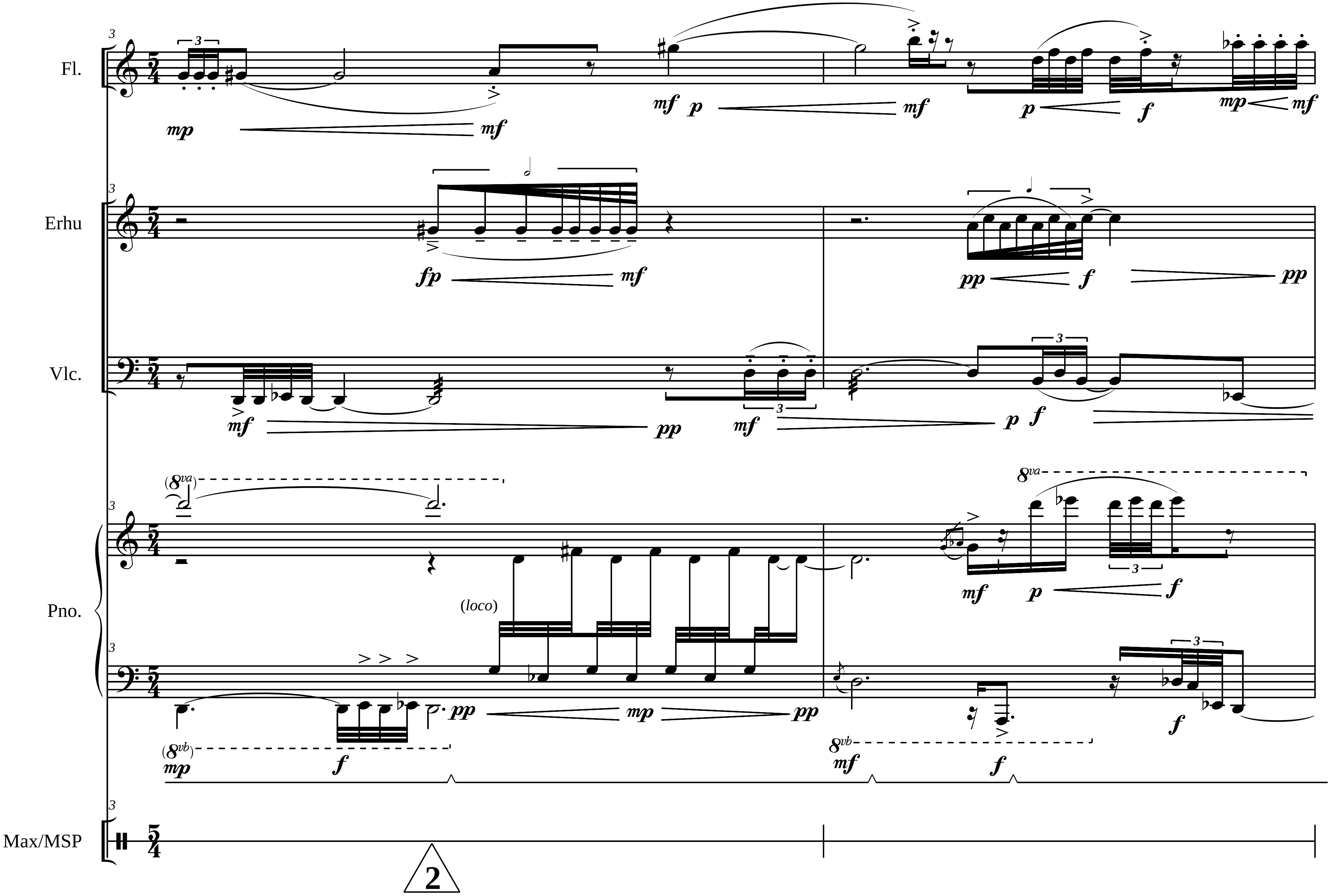


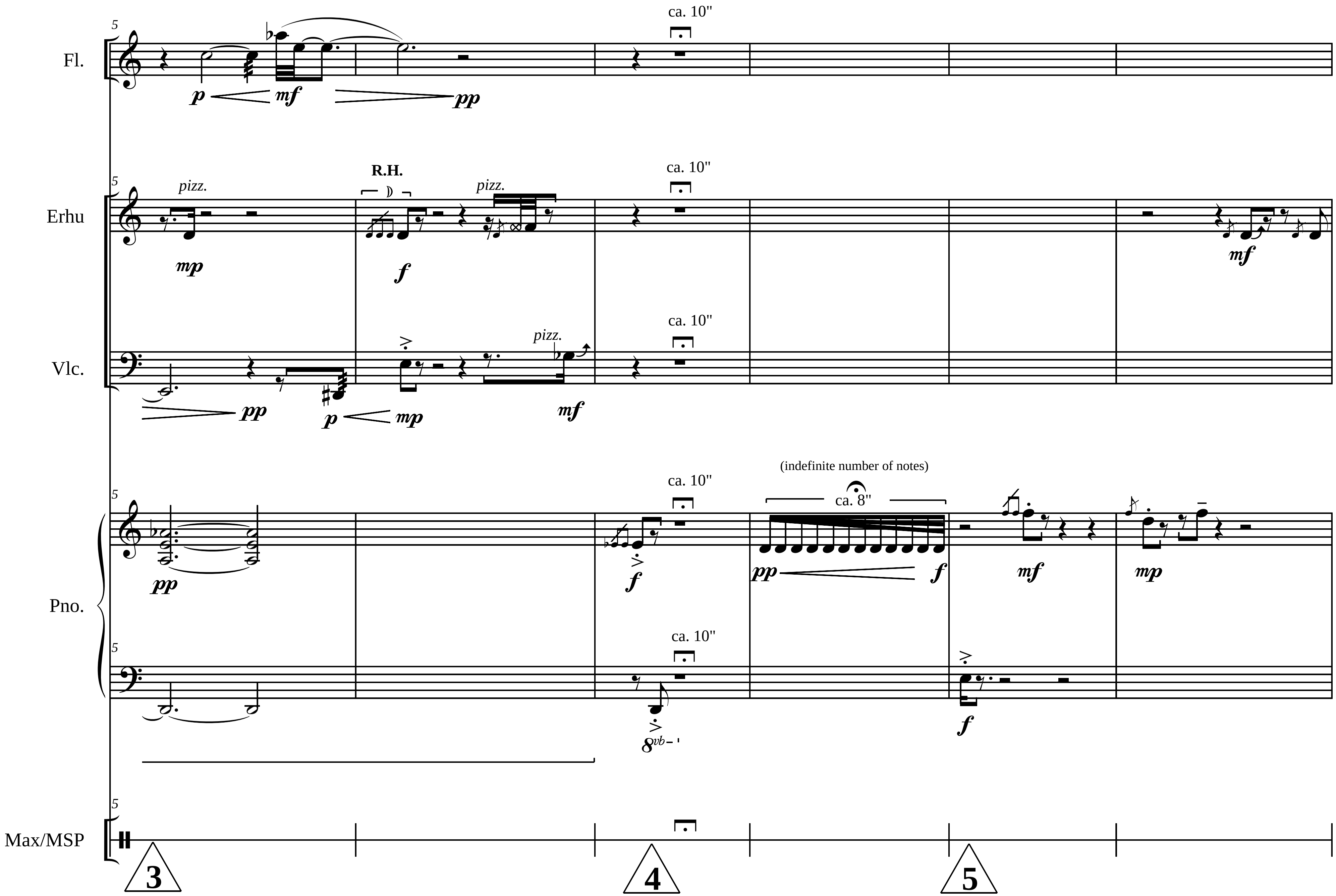
Stop the resonance of the previous pizzicato sound by touching the specified pitch point without plucking the string.

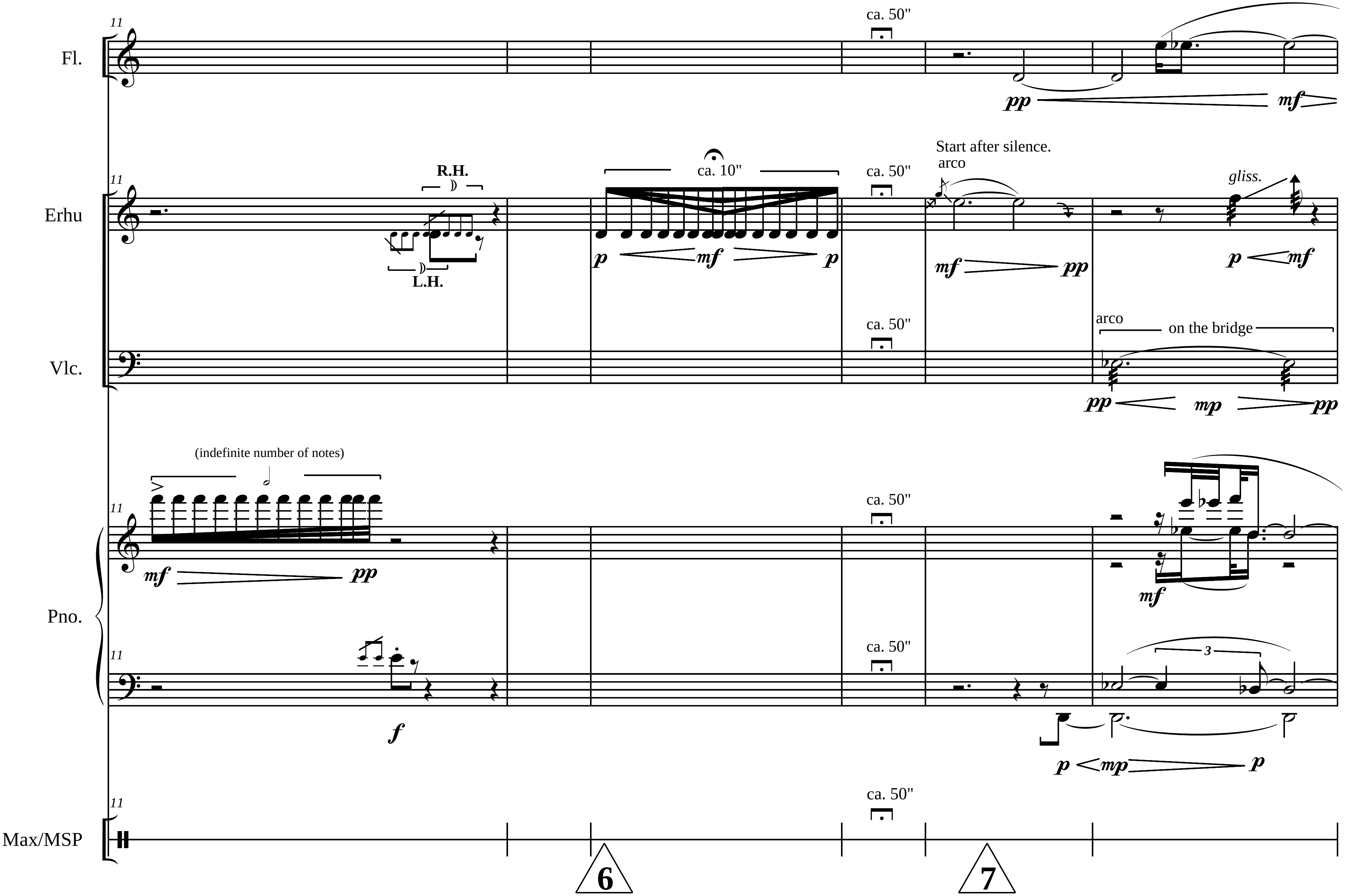
Snow Spell

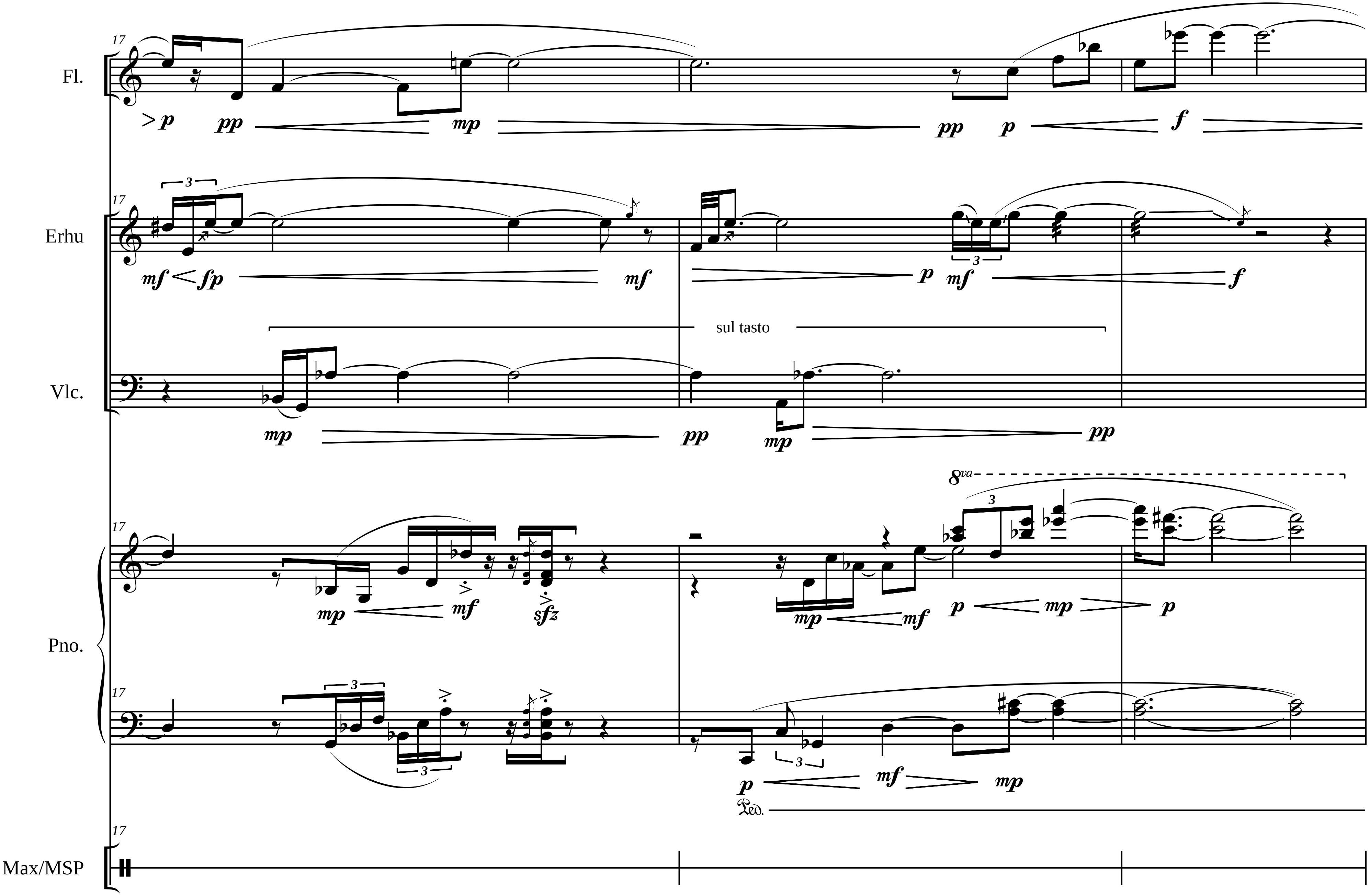
The First Snowflake
初雪晶瑩落地融

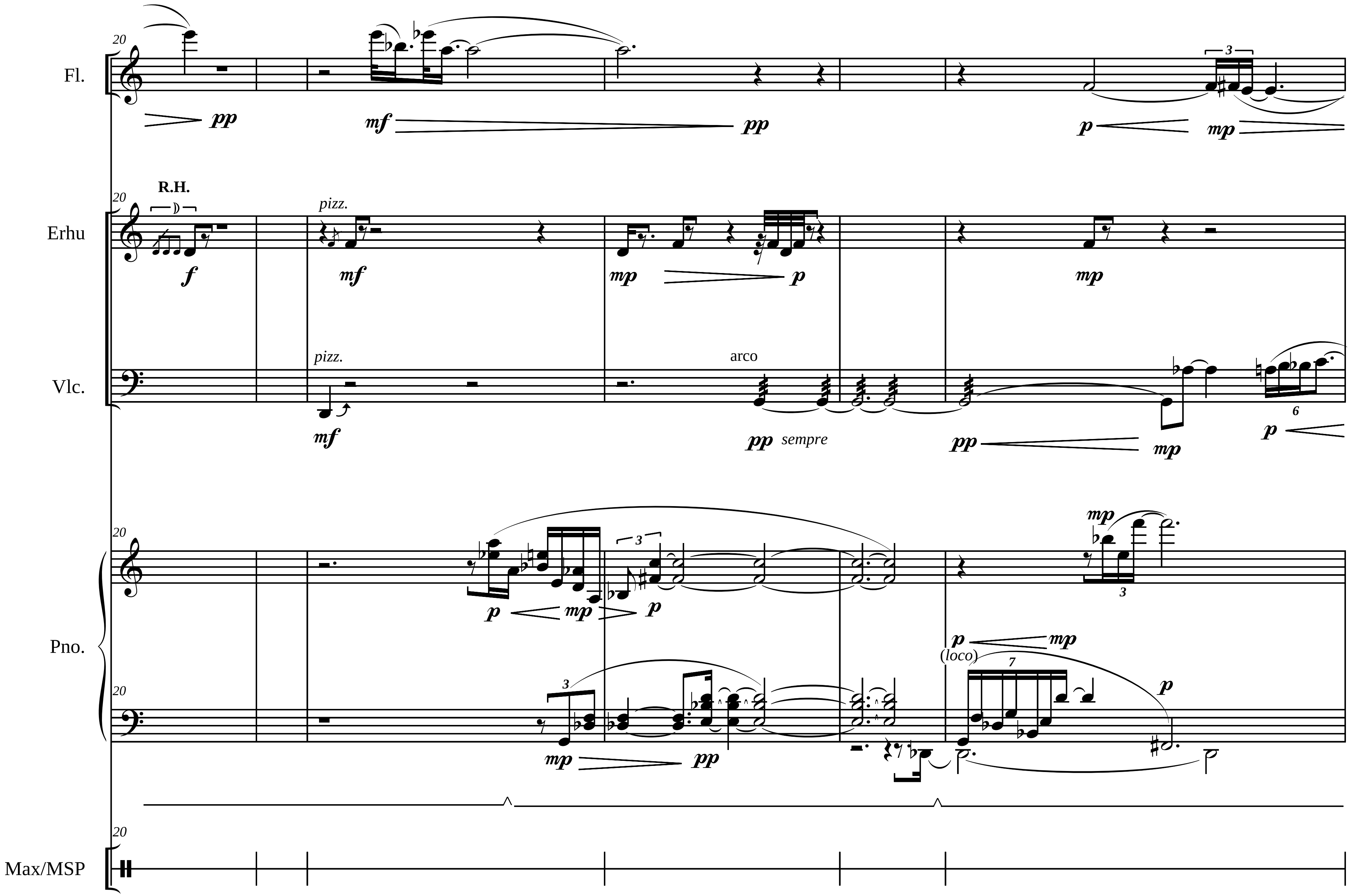

♩= 54
Flute
Erhu
Cello
Piano
Max/MSP
fp
mp
mf
p
f
mfp
(from slow to fast)
L.H.
8va
8vb
Ped.
Cue No.:
1


3
Fl.
Erhu
Vlc.
Pno.
Max/MSP
mp
mf
p
f
fp
pp
(8va)
(loco)
8va
(8vb)
8vb
2

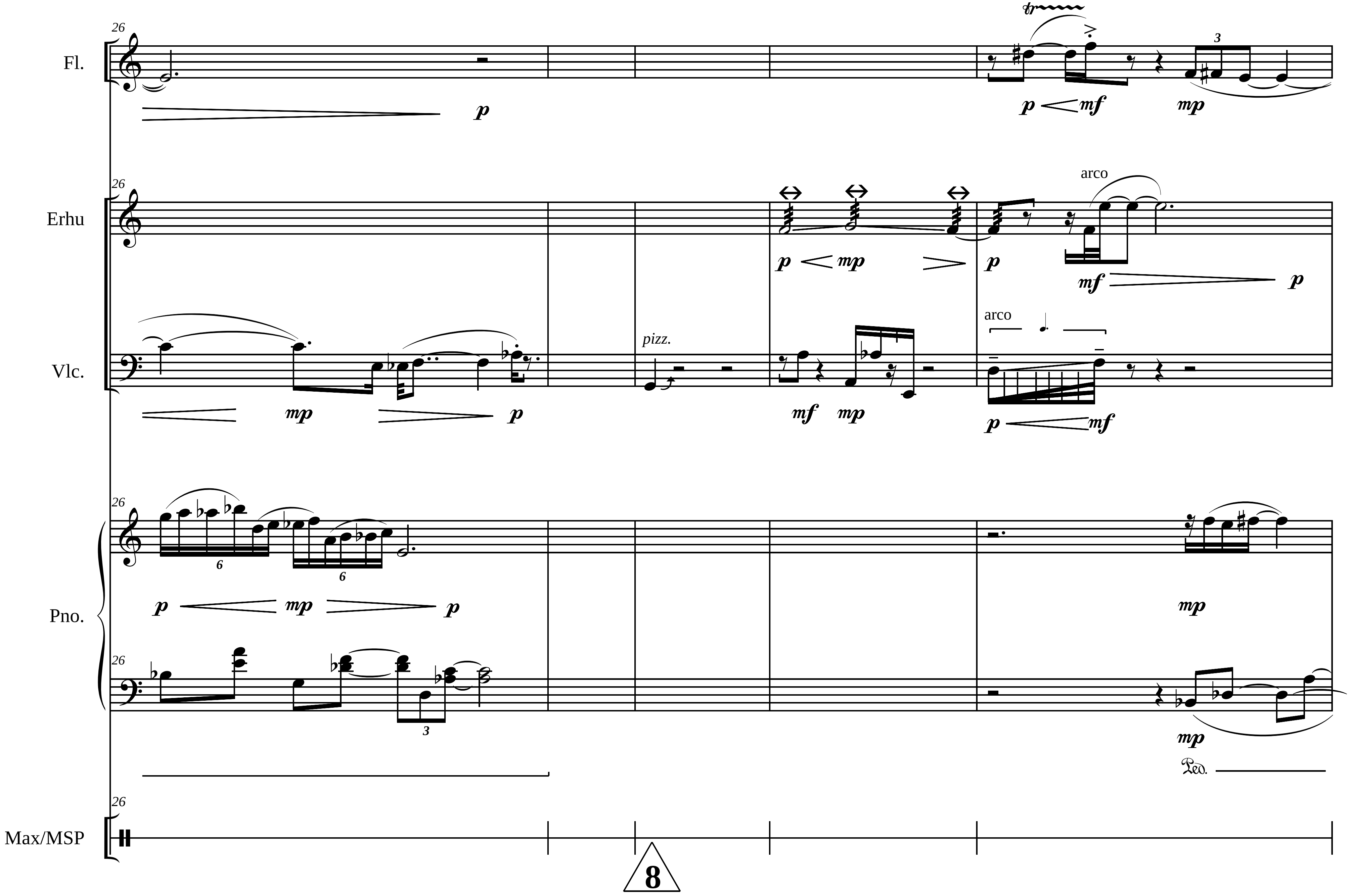

Fl.
Erhu
Vlc.
Pno.
Max/MSP
5
pizz.
R.H.
pizz.
pizz.
ca. 10"
ca. 10"
ca. 10"
ca. 10"
ca. 10"
(indefinite number of notes)
ca. 8"
p
mf
pp
mp
f
mf
pp
p
mp
mf
pp
f
pp
f
mf
mp
f
8vb
3
4
5

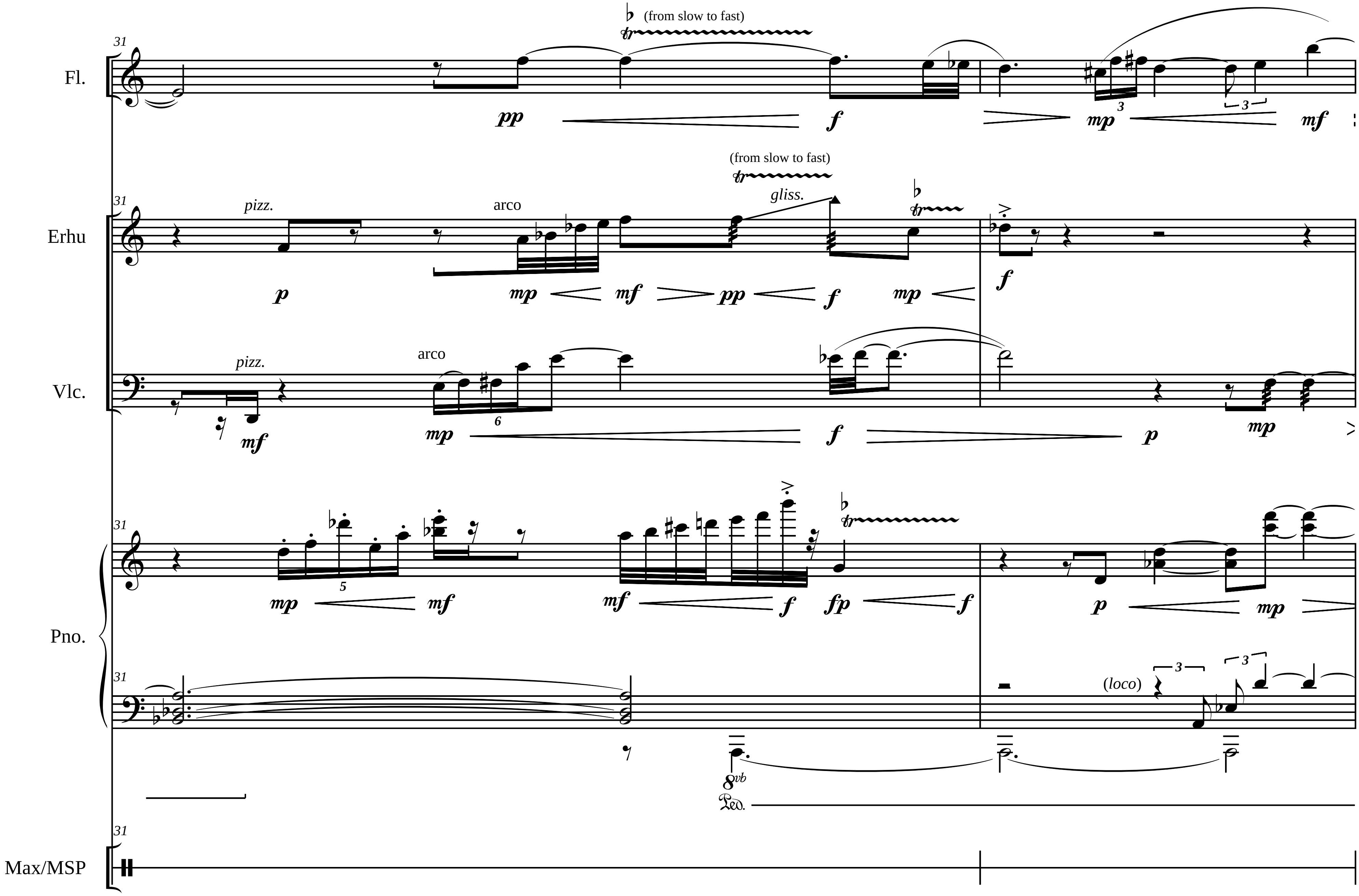

11
Fl.
Erhu
Vlc.
Pno.
Max/MSP
ca. 50"
pp
mf
R.H.
L.H.
ca. 10"
p
mf
p
Start after silence.
arco
mf
pp
gliss.
p
mf
arco
on the bridge
pp
mp
pp
(indefinite number of notes)
mf
pp
mf
3
f
p
mp
p
6
7

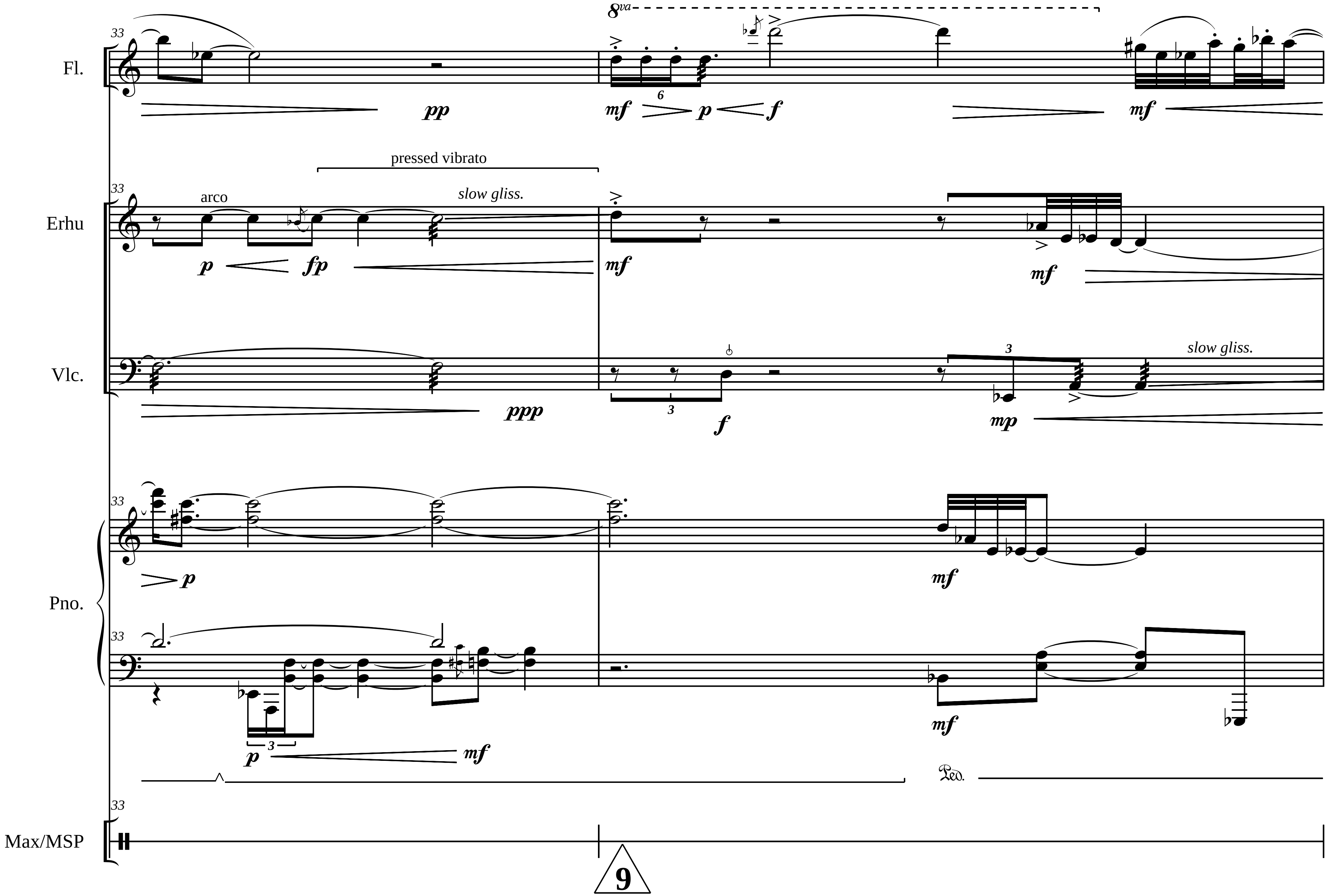

17
Fl.
Erhu
Vlc.
Pno.
Max/MSP
sul tasto
8va
3
p
pp
mp
f
mf
fp
sfz

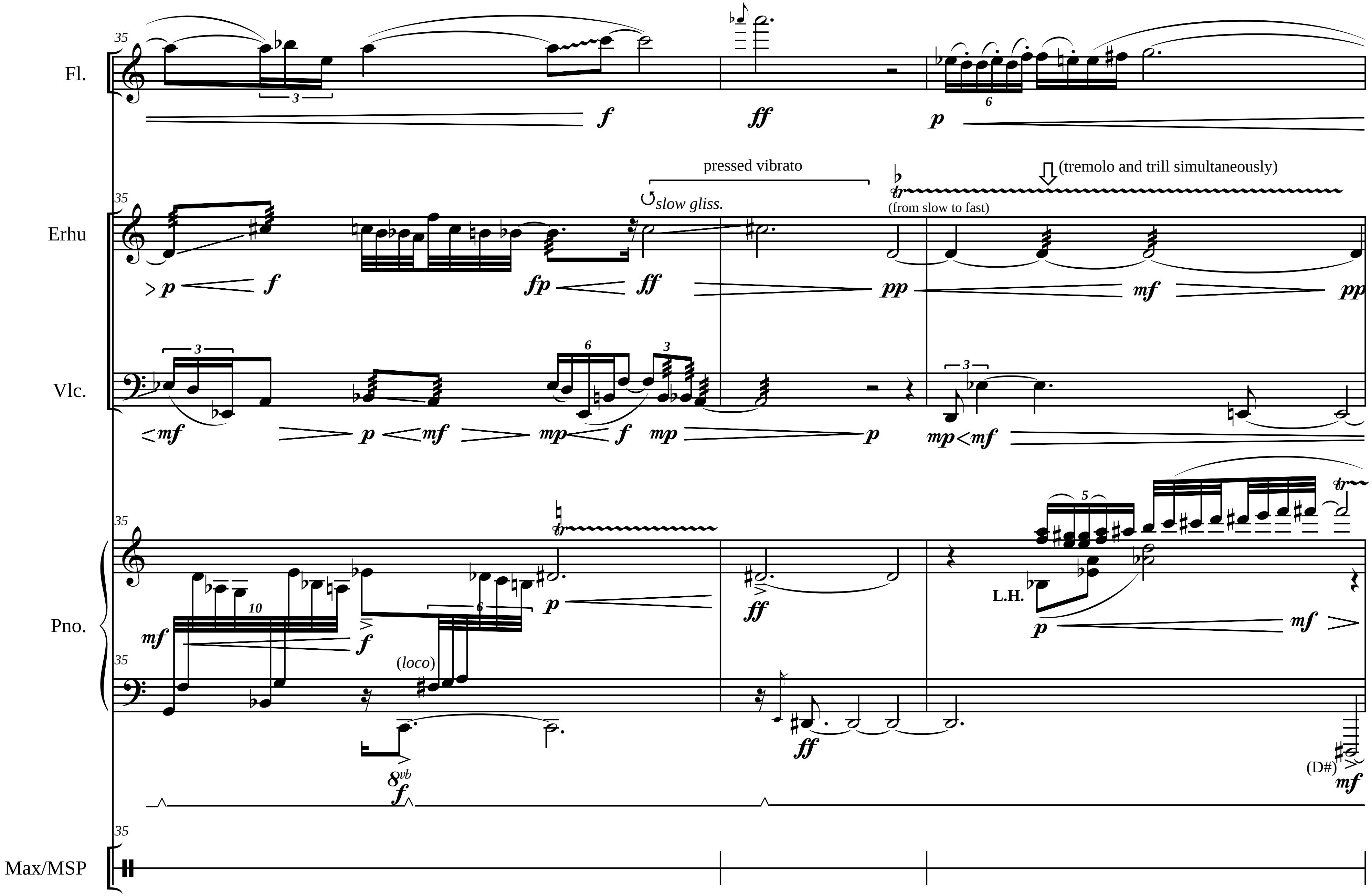


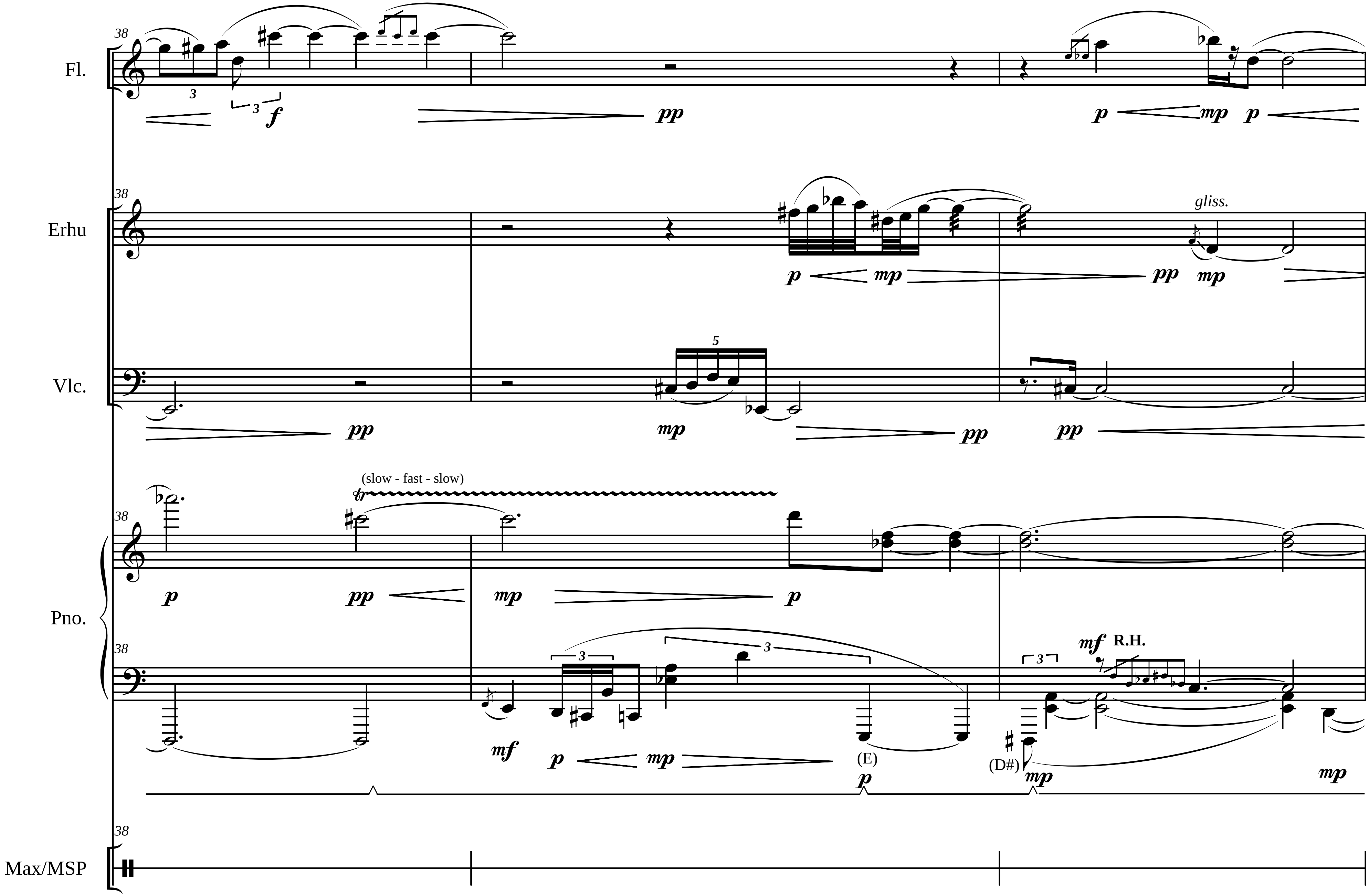
Fl.
Erhu
Vlc.
Pno.
Max/MSP
20
R.H.
pizz.
pizz.
arco
sempre
(loco)
3
6
7
pp
mf
p
mp
f

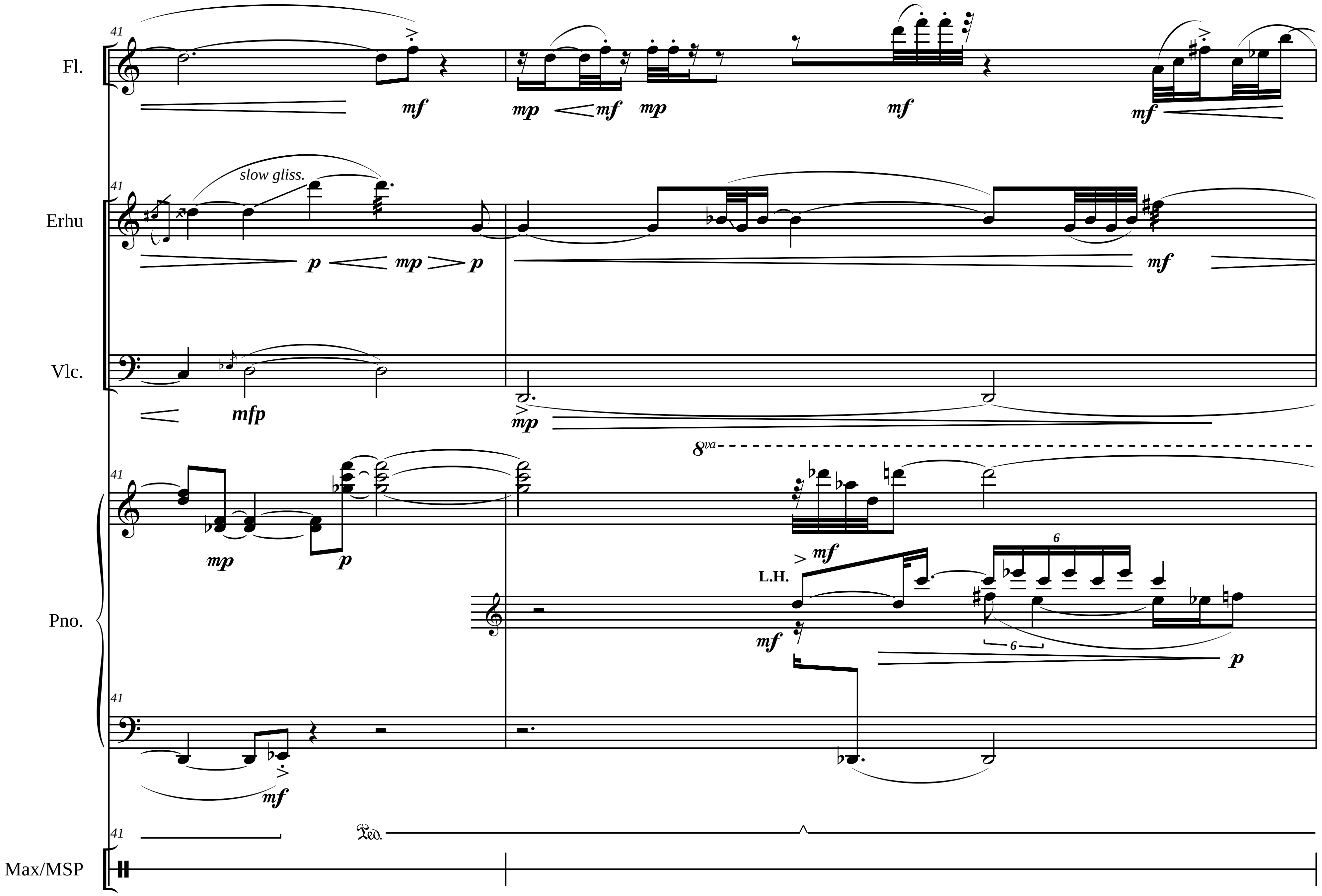

26
Fl.
Erhu
Vlc.
Pno.
Max/MSP
p
mp
mf
arco
pizz.
3
6
8

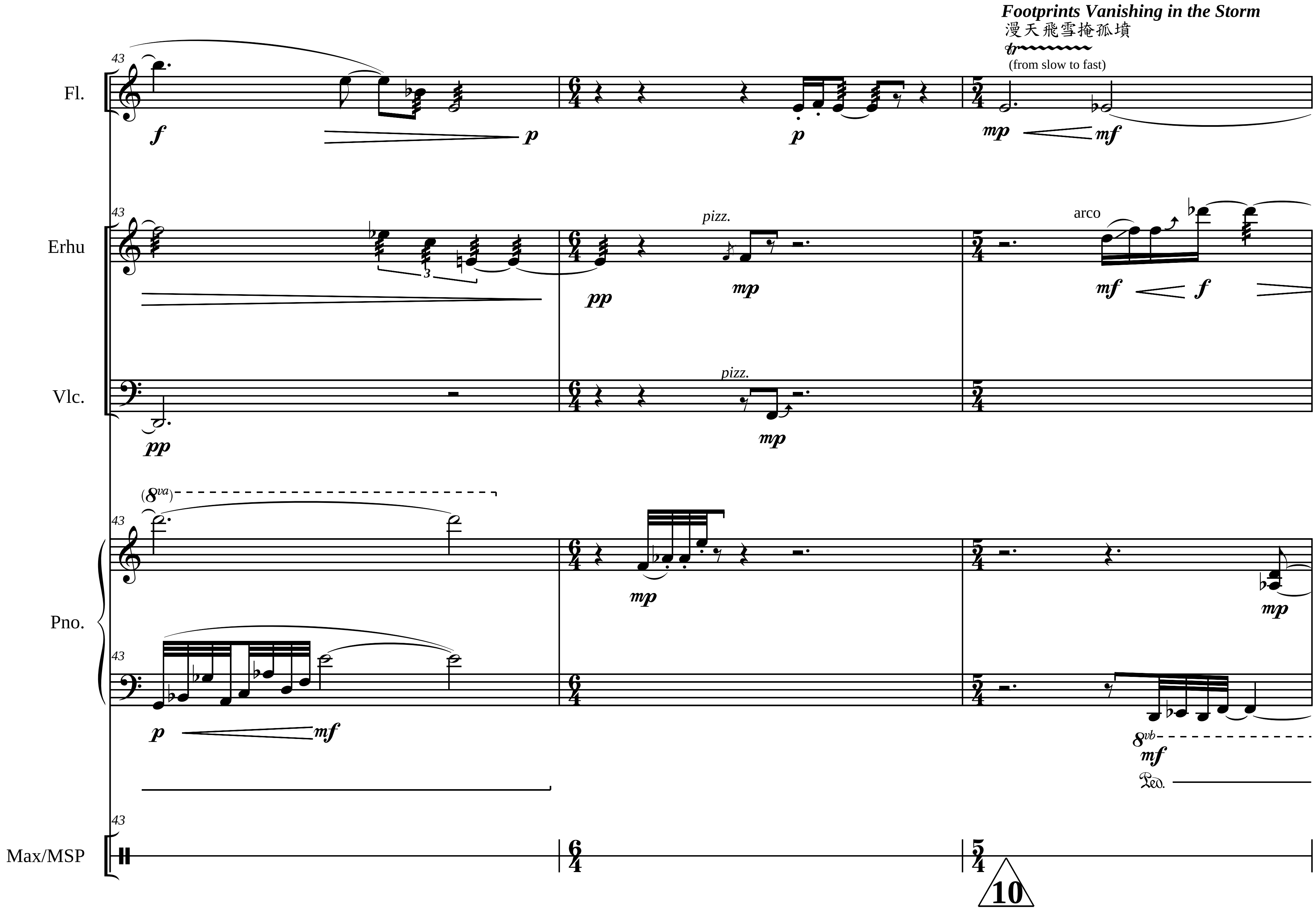

Fl.
Erhu
Vlc.
Pno.
Max/MSP
31
(from slow to fast)
pizz.
arco
gliss.
(loco)
8vb
Ped.
pp
f
mp
mf
p
fp
3
5
6

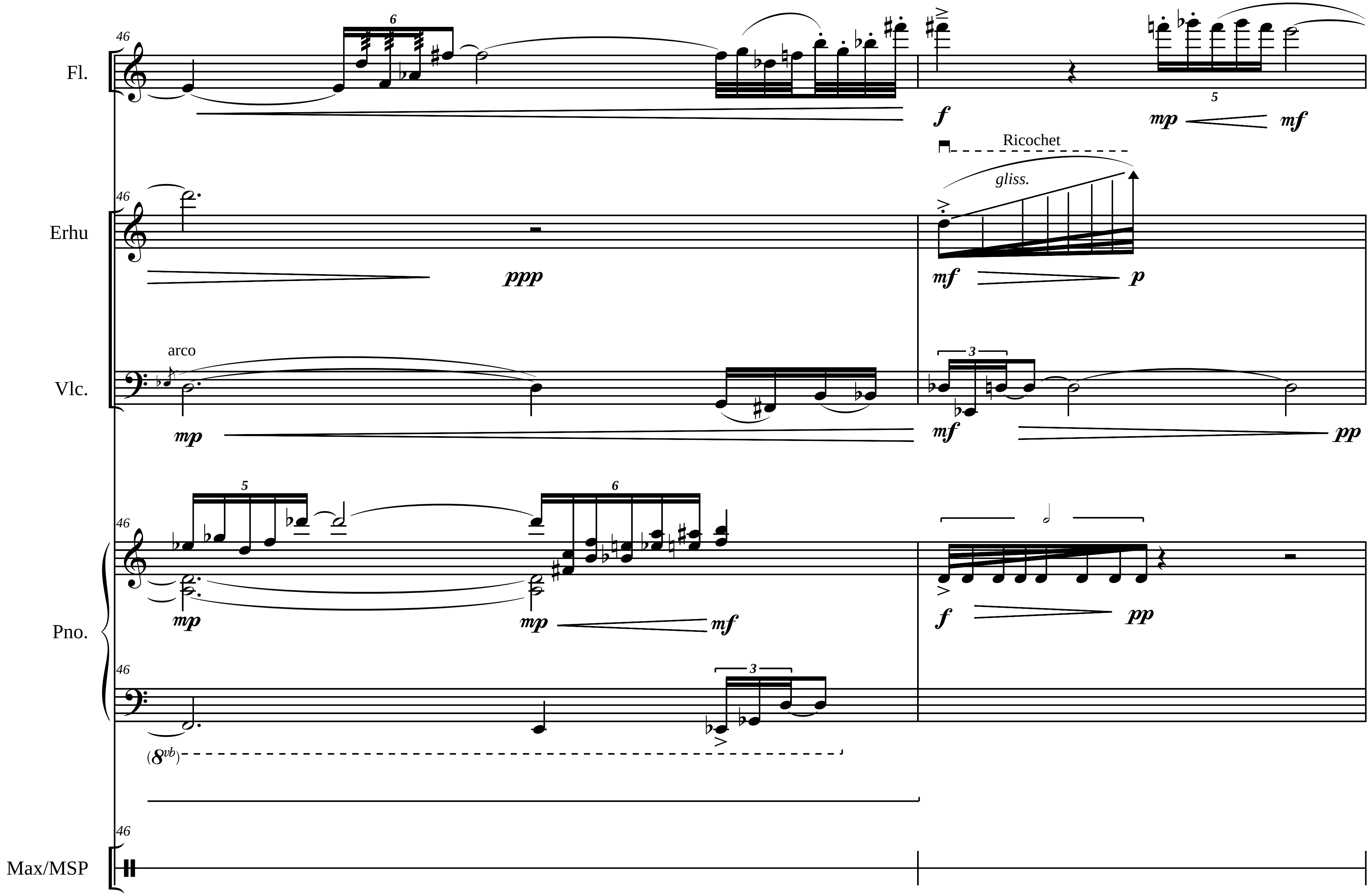

Fl.
Erhu
Vlc.
Pno.
Max/MSP
33
8va
pp
mf
6
p
f
pressed vibrato
arco
slow gliss.
fp
3
ppp
mp
Ped.
9

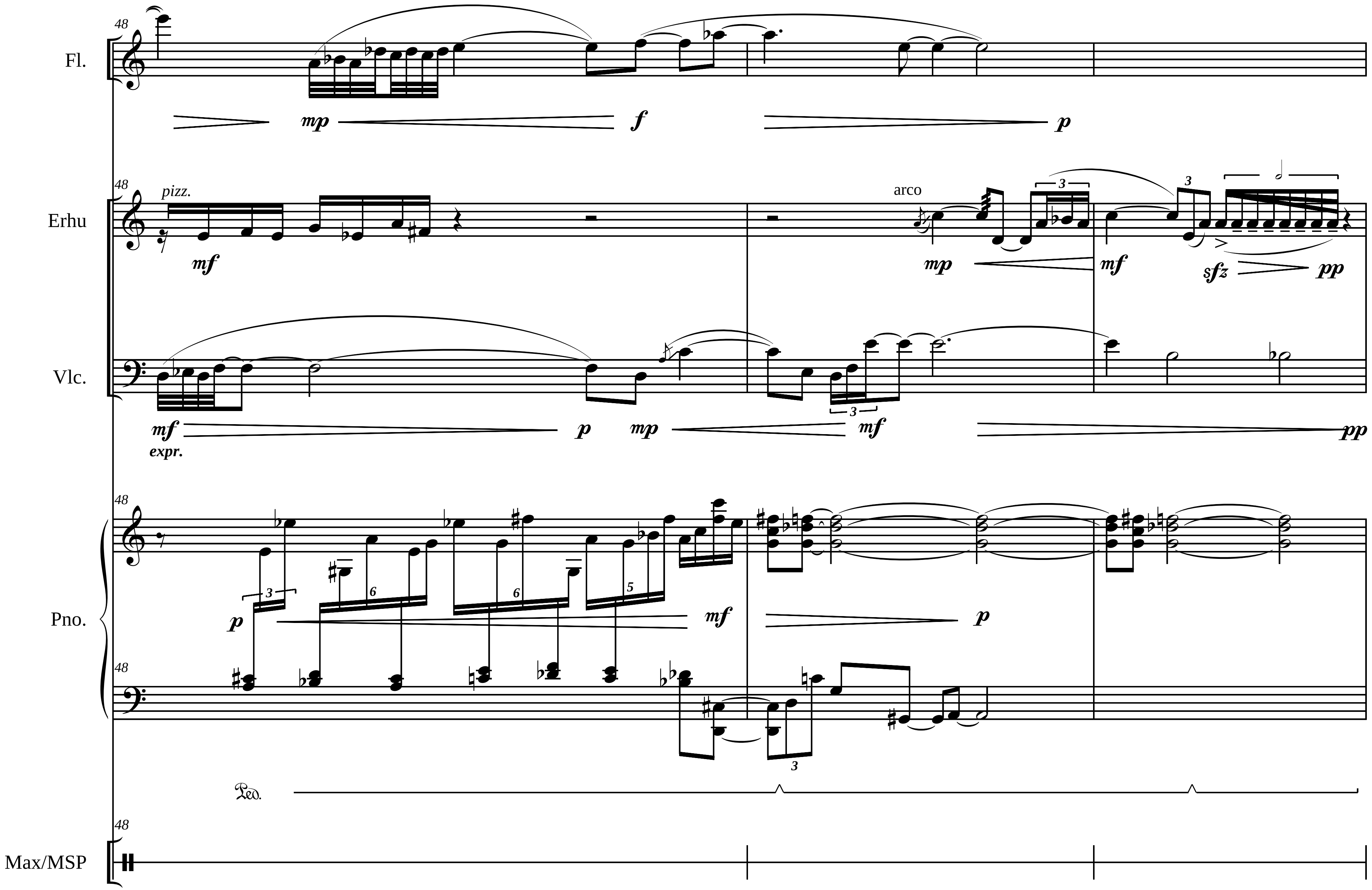

35
Fl.
Erhu
Vlc.
Pno.
Max/MSP
pressed vibrato
slow gliss.
(from slow to fast)
(tremolo and trill simultaneously)
(loco)
L.H.
(D#)
8vb

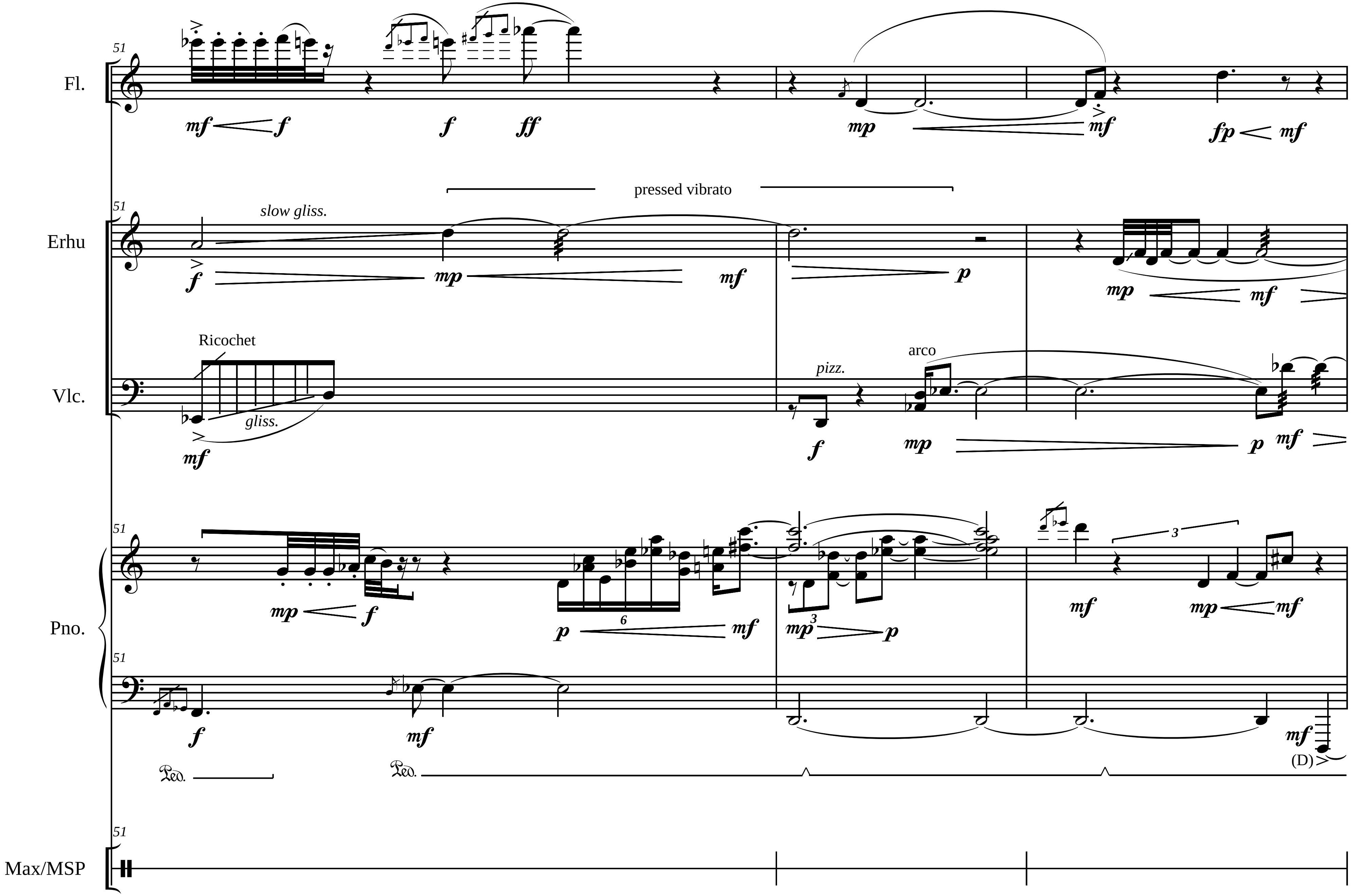

38
Fl.
Erhu
Vlc.
Pno.
Max/MSP
f
pp
p
mp
gliss.
(slow - fast - slow)
tr
mf
R.H.
(E)
(D#)
3
5

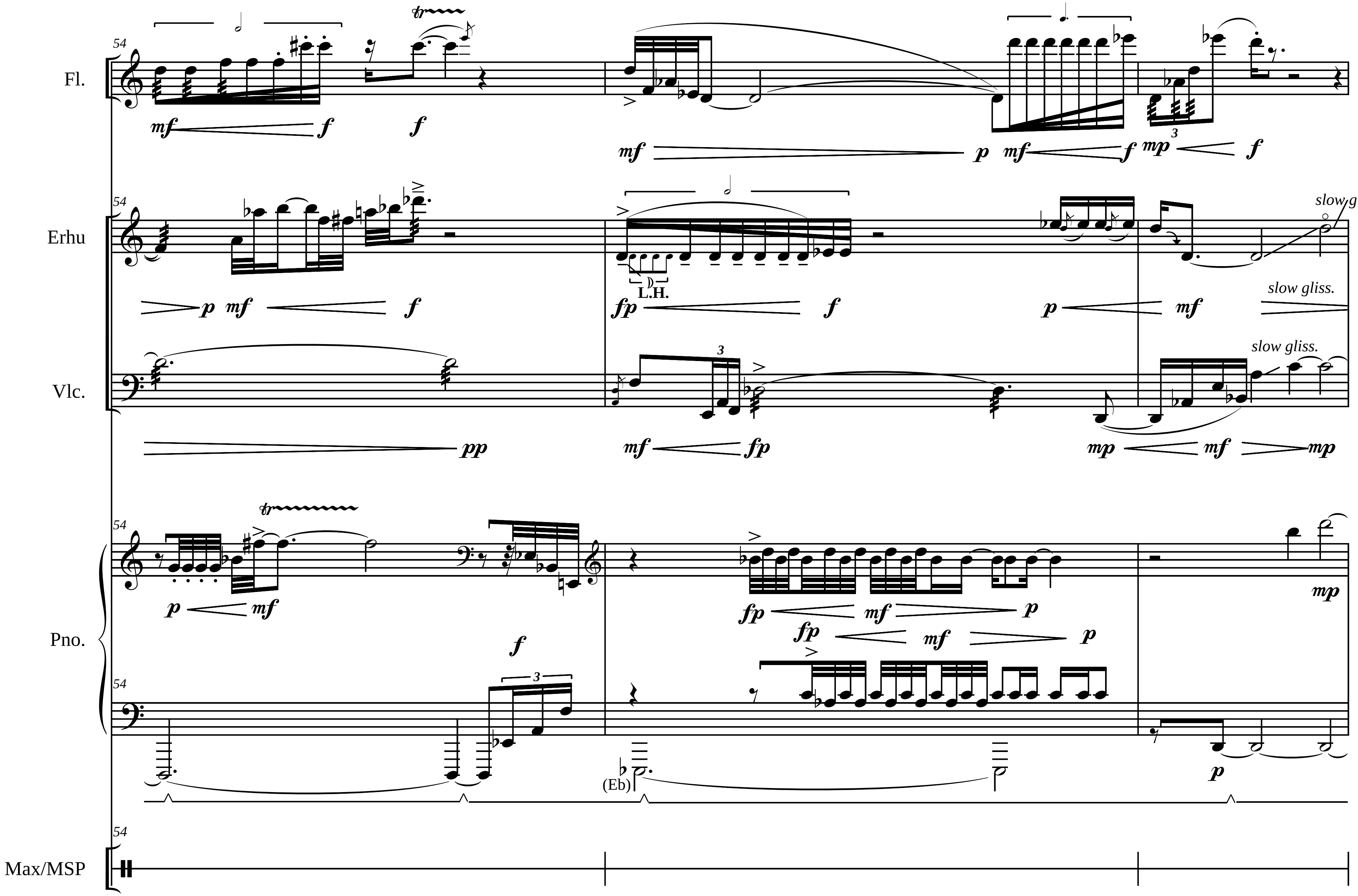

41
Fl.
Erhu
Vlc.
Pno.
Max/MSP
slow gliss.
mf
mp
mp
mf
mp
p
mfp
8va
L.H.
6
Ped.

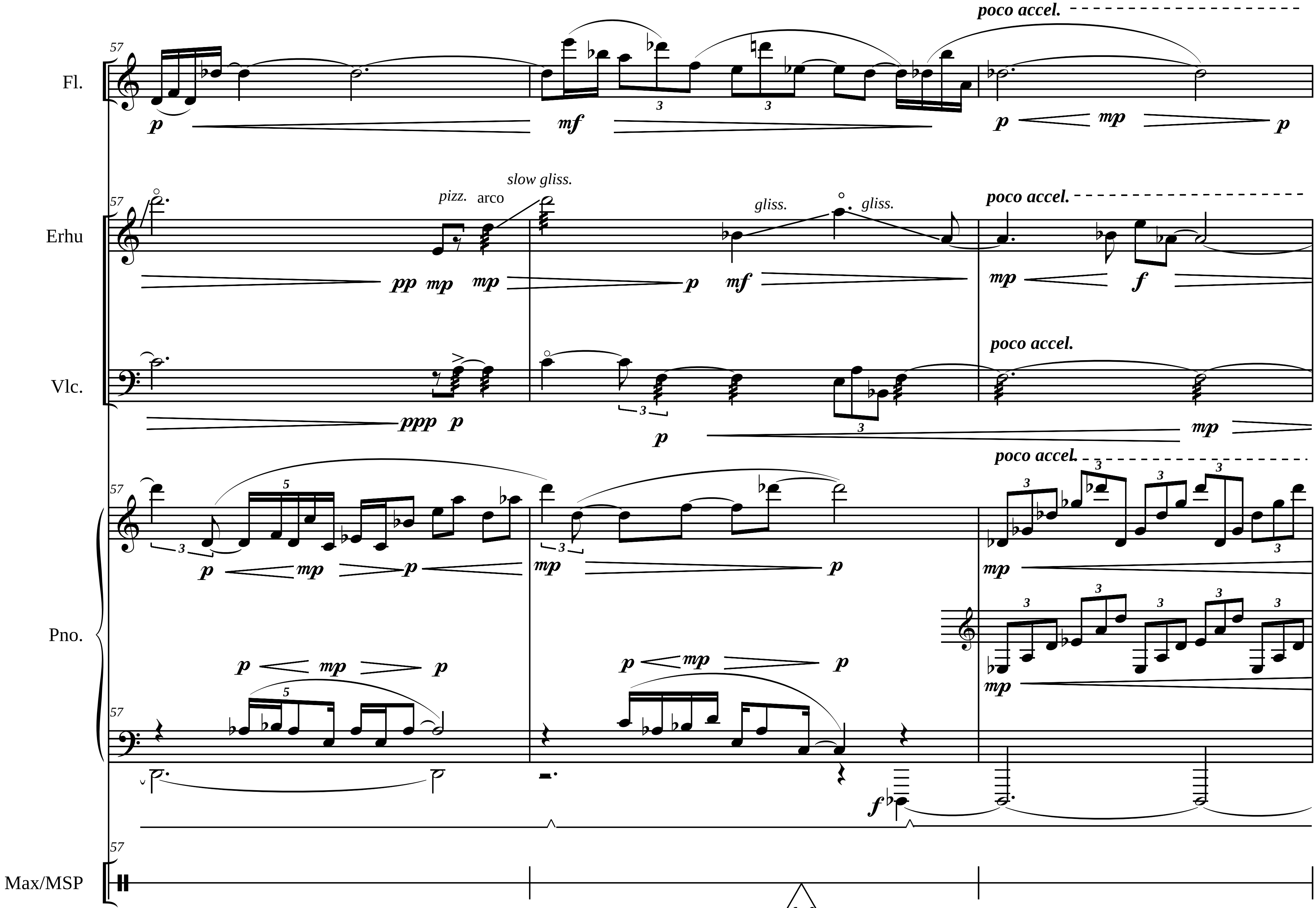

Footprints Vanishing in the Storm
漫天飛雪掩孤墳
tr
(from slow to fast)
43
Fl.
f
p
p
mp
mf
Erhu
3
pizz.
arco
pp
mp
mf
f
Vlc.
pizz.
pp
mp
(8va)
Pno.
mp
mp
p
mf
8vb
mf
Ped.
Max/MSP
10

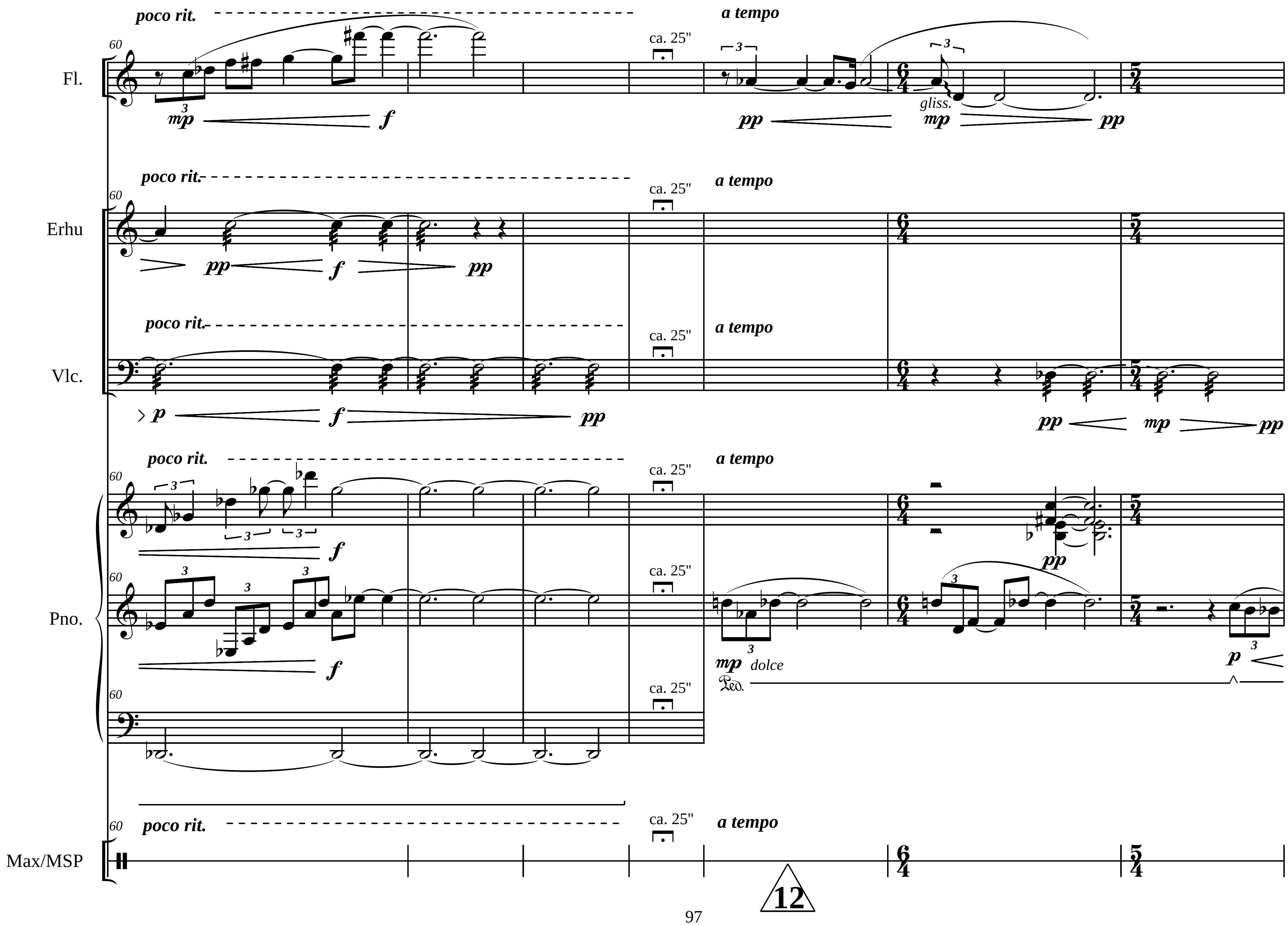

46
Fl.
Erhu
Vlc.
Pno.
Max/MSP
6
5
3
arco
Ricochet
gliss.
(8vb)
f
mp
mf
ppp
p
pp

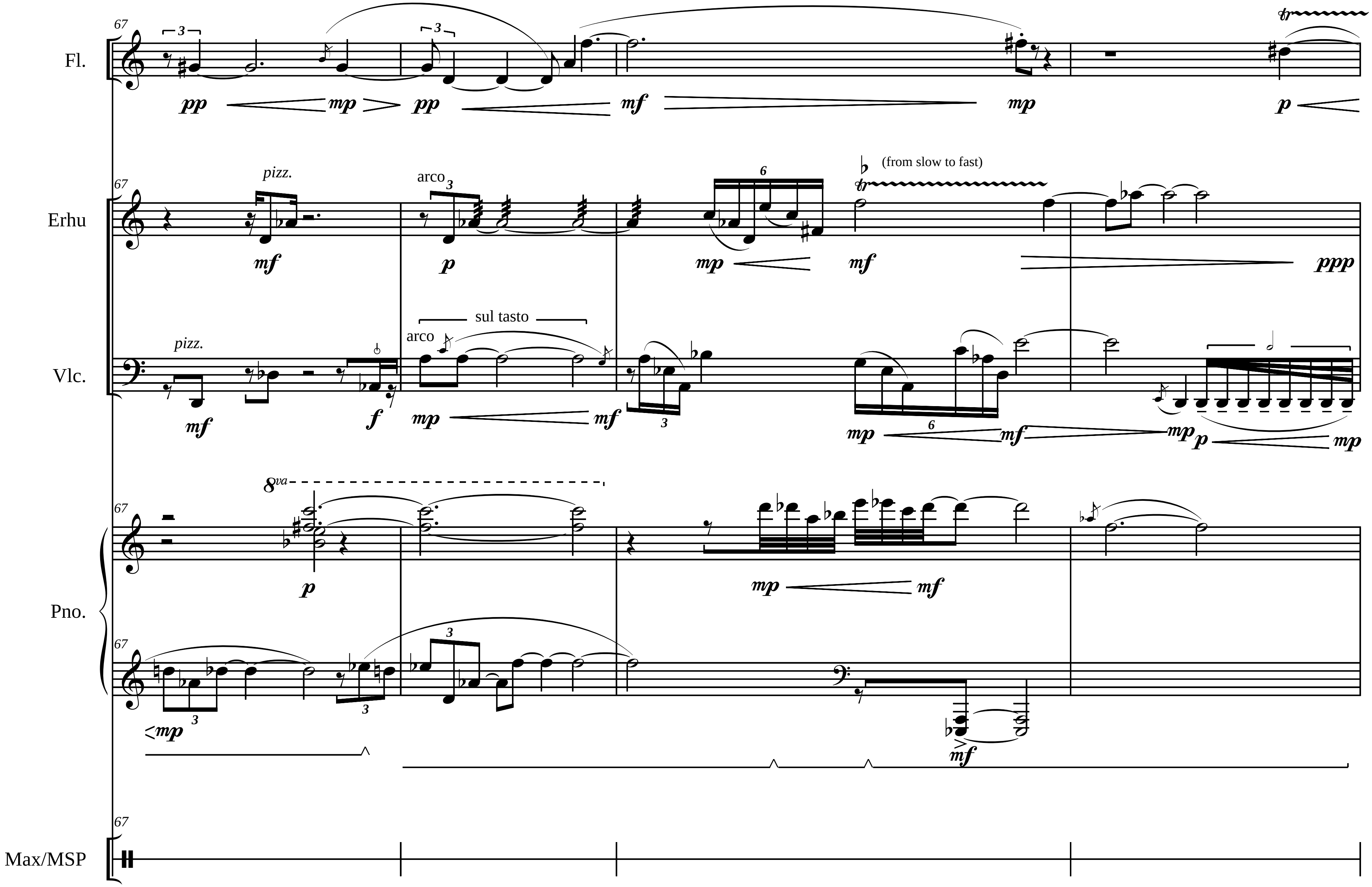

48
Fl.
mp
f
p
48
pizz.
Erhu
mf
arco
mp
mf
3
3
sfz
pp
Vlc.
mf
expr.
p
mp
3
mf
pp
48
Pno.
3
6
6
5
p
mf
p
48
3
Ped.
48
Max/MSP

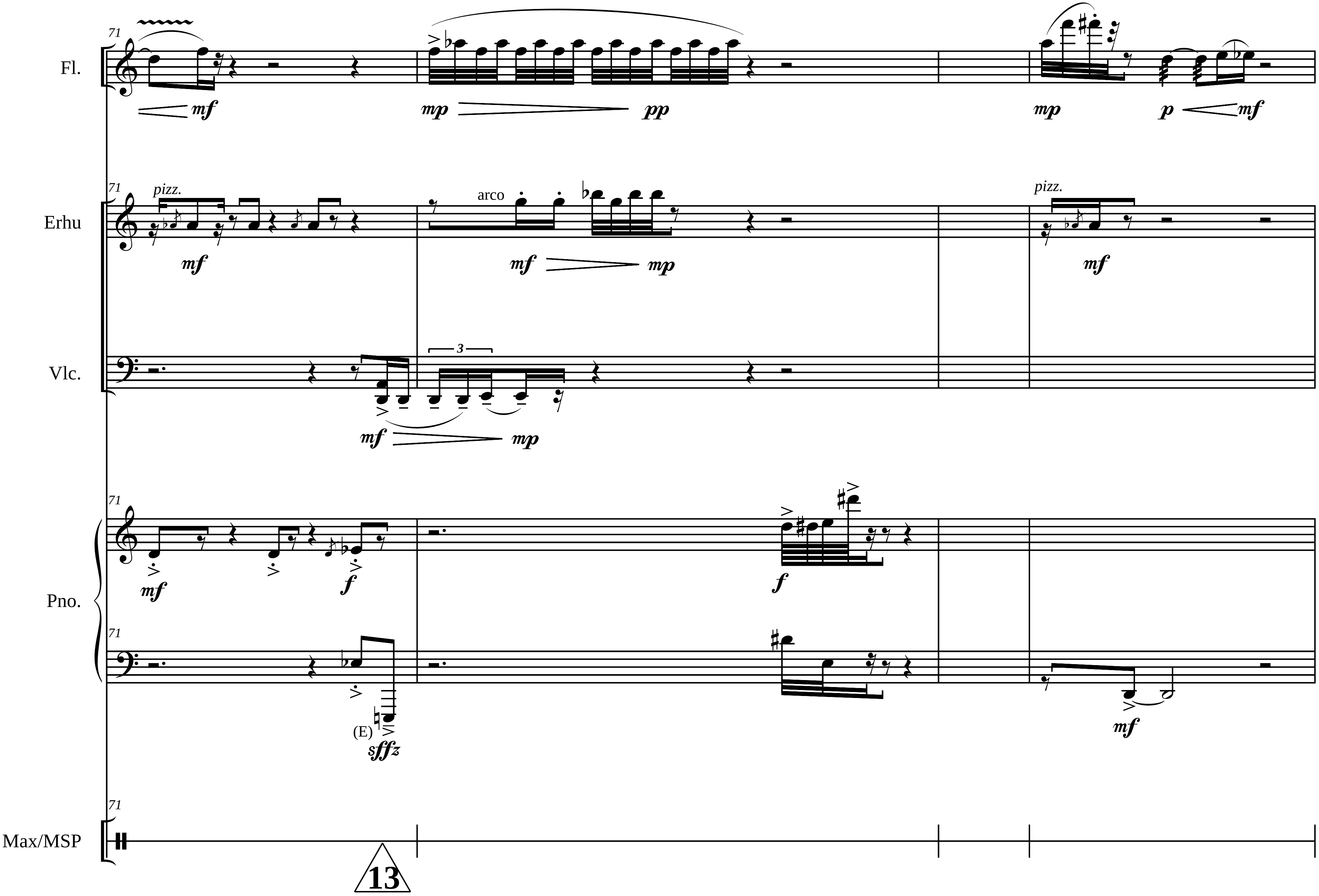

51
Fl.
mf
f
f
ff
mp
mf
fp
mf
pressed vibrato
Erhu
slow gliss.
f
mp
mf
p
mp
mf
Ricochet
Vlc.
gliss.
mf
pizz.
f
arco
mp
p
mf
Pno.
mp
f
6
p
mf
3
mp
p
mf
mp
mf
3
f
mf
mf
(D)
Ped.
Ped.
Max/MSP

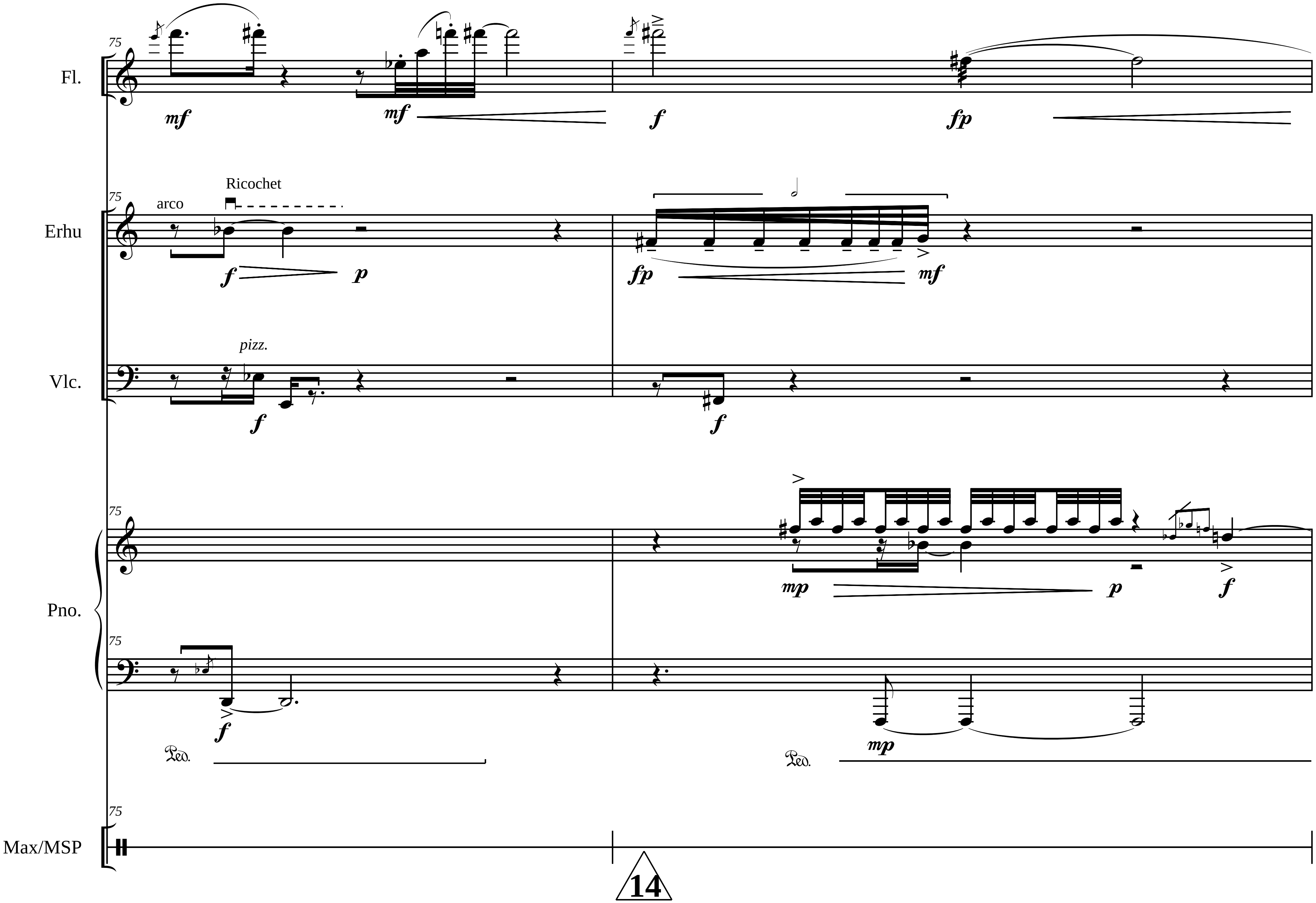

54
Fl.
Erhu
Vlc.
Pno.
Max/MSP
mf
f
mp
p
pp
fp
3
tr
slow gliss.
L.H.
(Eb)

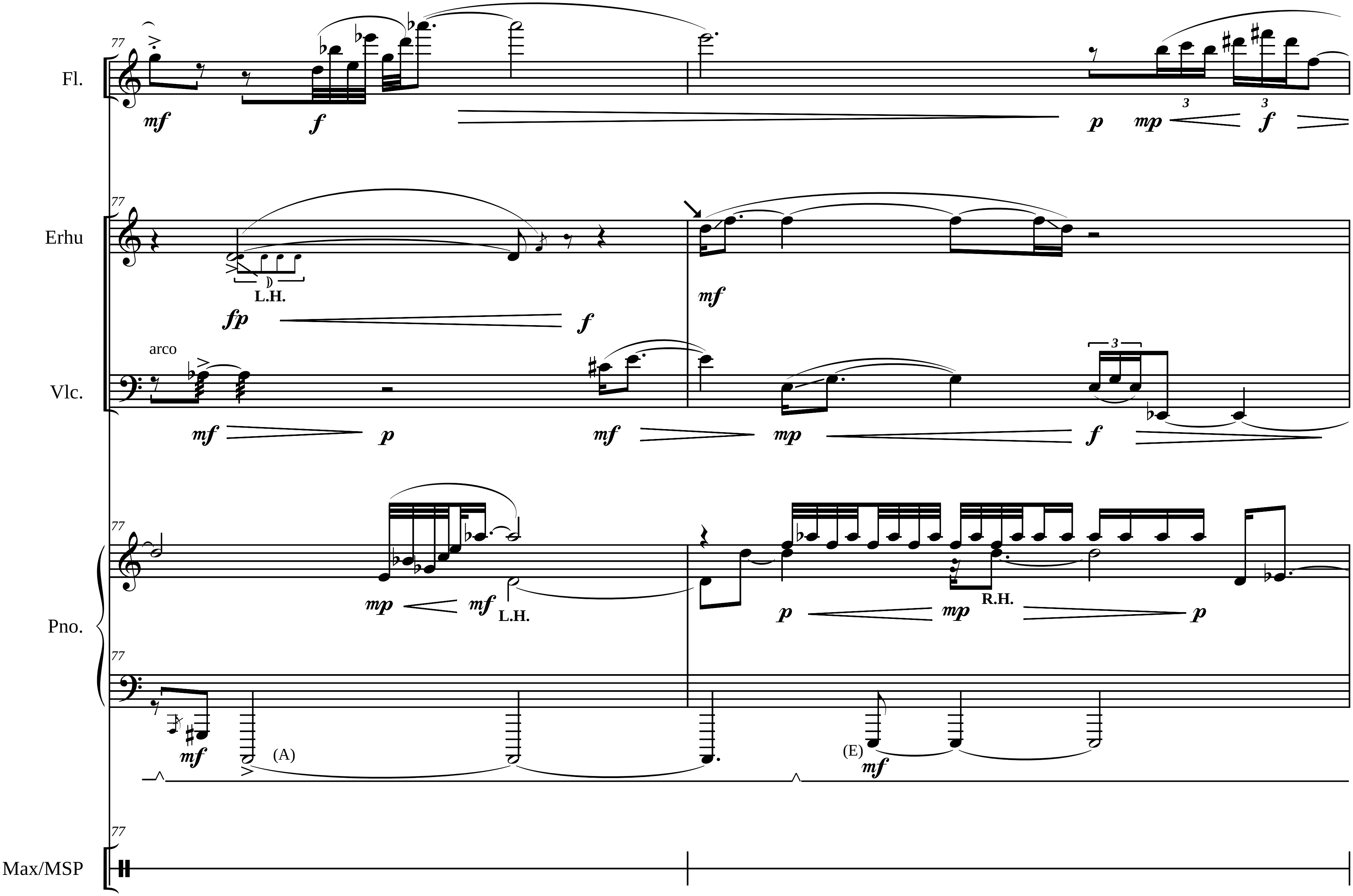

poco accel.
57
Fl.
p
mf
3
3
p
mp
p
slow gliss.
pizz.
arco
gliss.
gliss.
57
Erhu
pp
mp
mp
p
mf
mp
f
poco accel.
57
Vlc.
3
3
ppp
p
p
mp
poco accel.
57
5
3
3
3
3
3
3
3
Pno.
p
mp
p
mp
p
mp
p
mp
p
mp
p
p
mp
p
f
57
Max/MSP
11

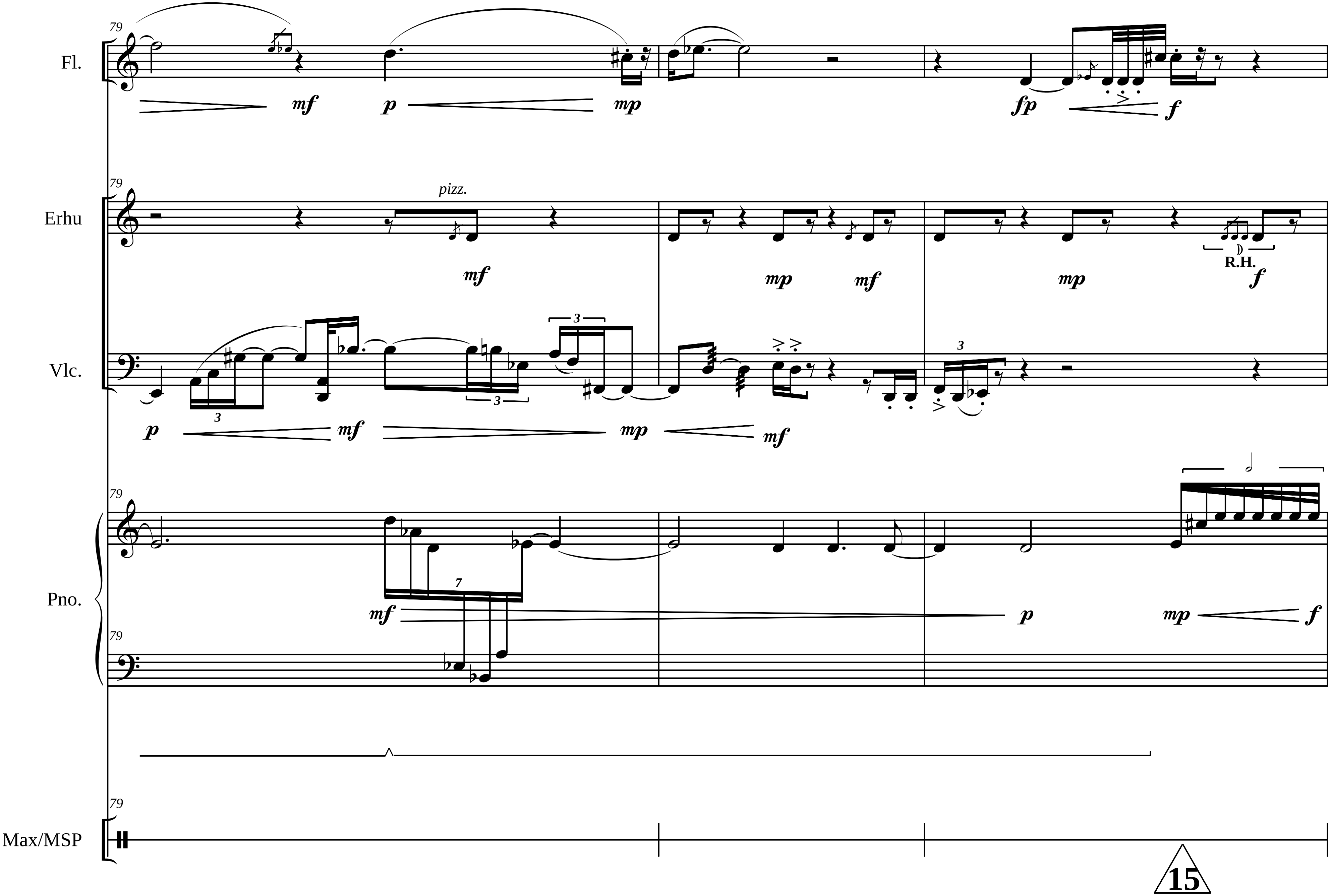

poco rit.
a tempo
ca. 25"
60
Fl.
Erhu
Vlc.
Pno.
Max/MSP
mp
f
pp
gliss.
p
mp dolce
12


67
Fl.
Erhu
Vlc.
Pno.
Max/MSP
pizz.
arco
sul tasto
(from slow to fast)
8va
pp
mp
mf
p
ppp
f
3
6

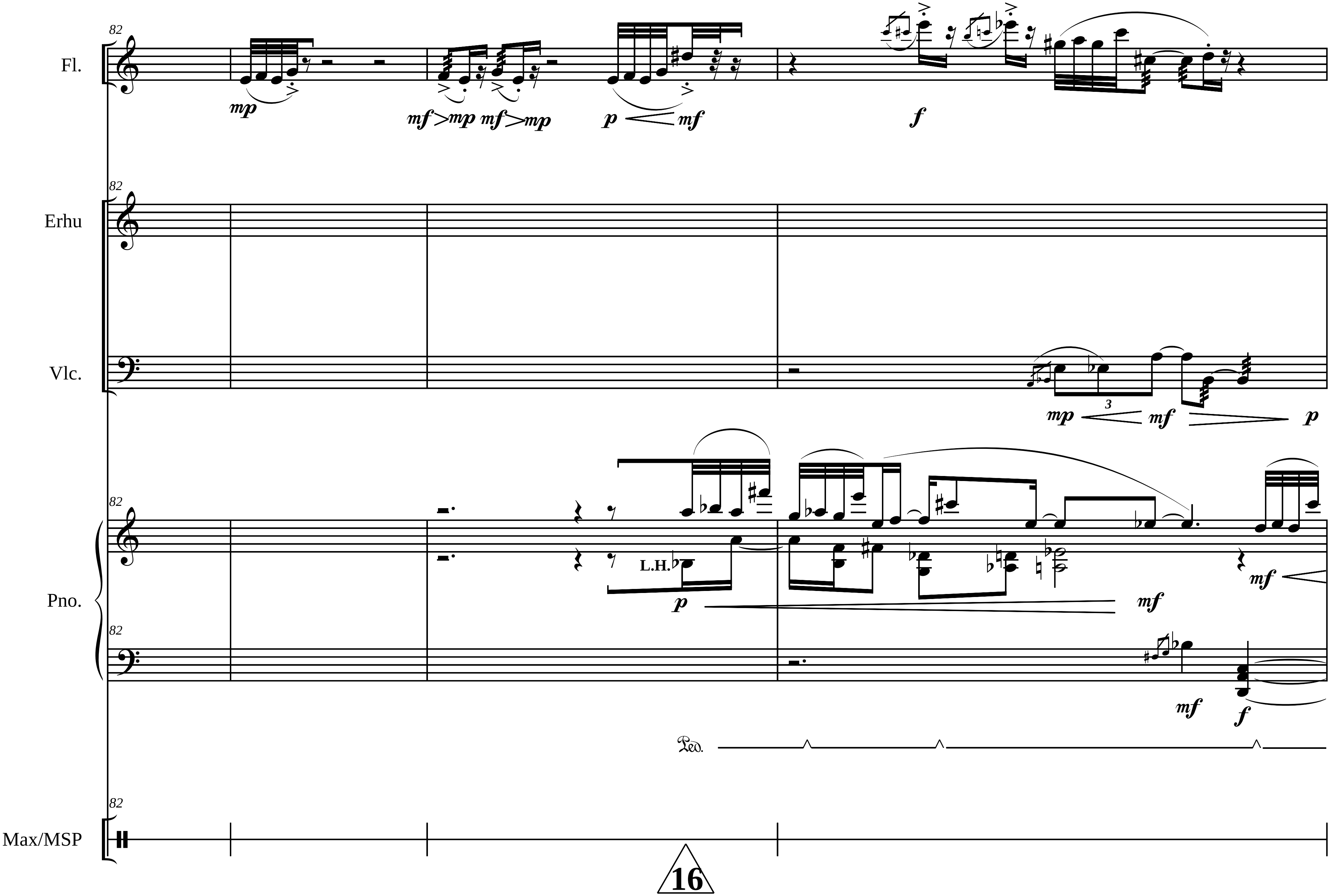

71
Fl.
mf
mp
pp
mp
p
mf
Erhu
pizz.
arco
mf
mp
pizz.
mf
Vlc.
3
mf
mp
Pno.
mf
f
f
(E)
sffz
mf
Max/MSP
13

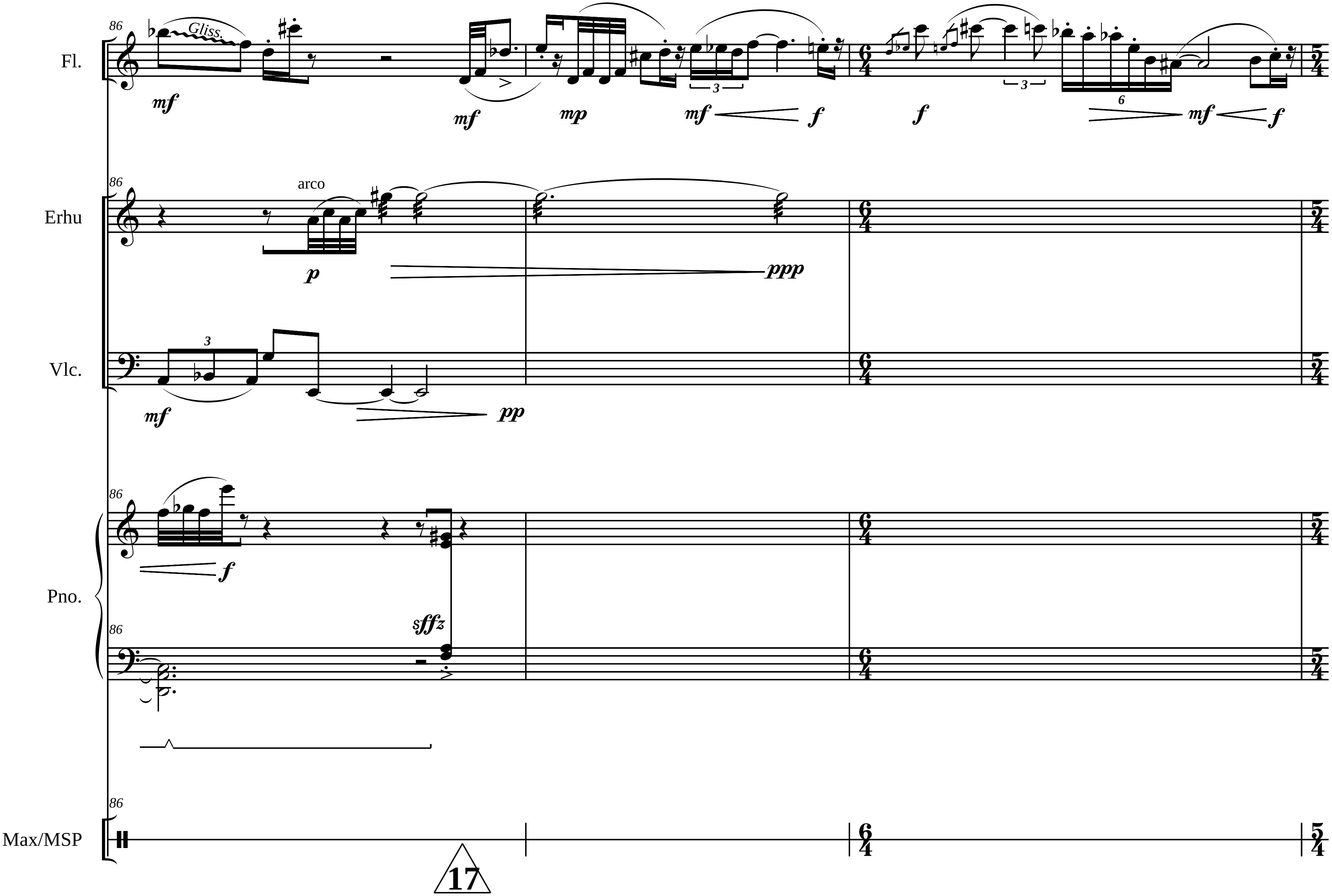

75
Fl.
mf
mf
f
fp
Erhu
arco
Ricochet
f
p
fp
mf
Vlc.
pizz.
f
f
Pno.
mp
p
f
f
mp
Ped.
Ped.
Max/MSP
14

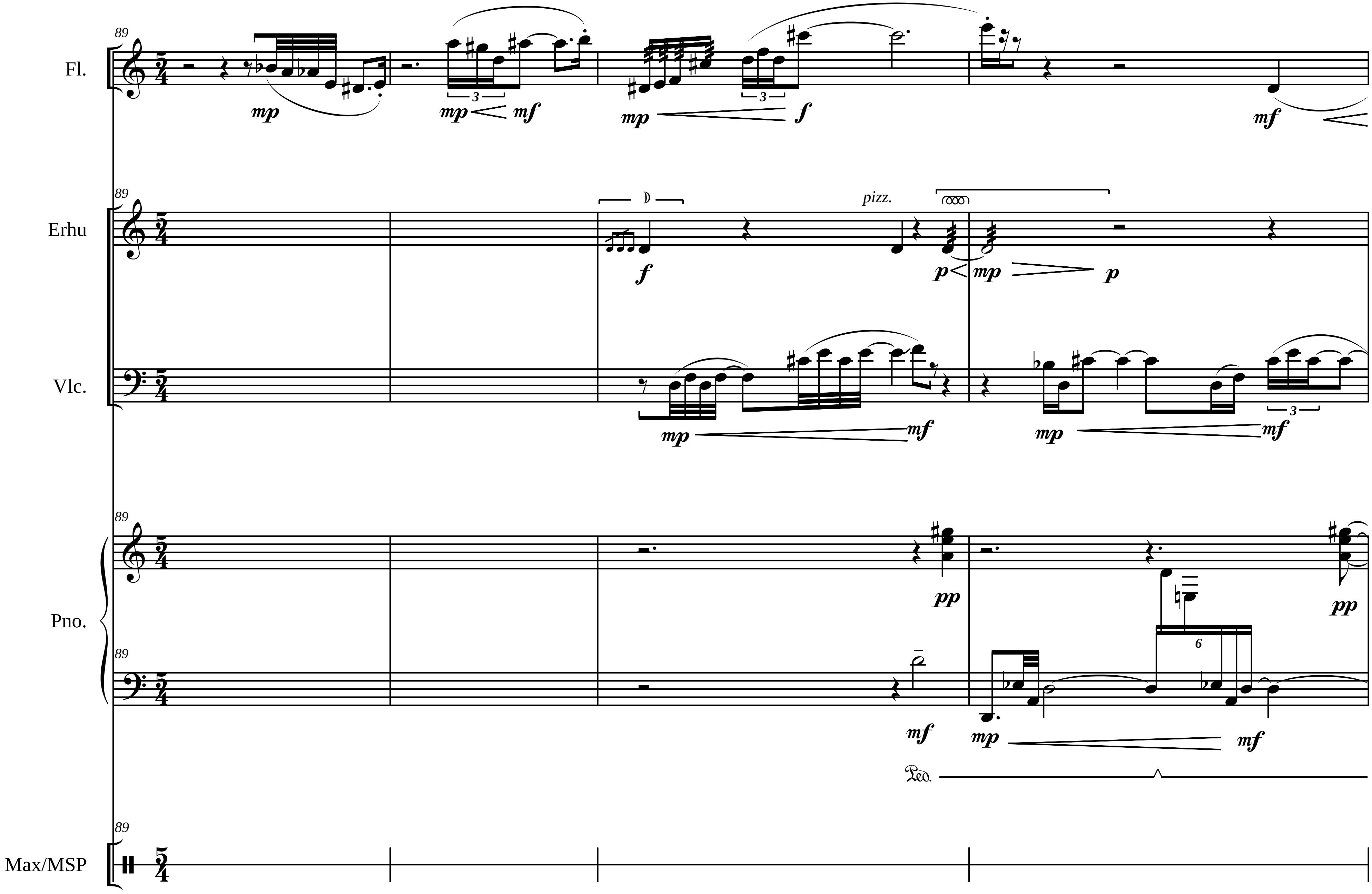

77
Fl.
mf
f
p
mp
3
f
Erhu
L.H.
fp
f
mf
arco
Vlc.
mf
p
mp
f
Pno.
mp
mf
L.H.
p
R.H.
(A)
(E)
Max/MSP

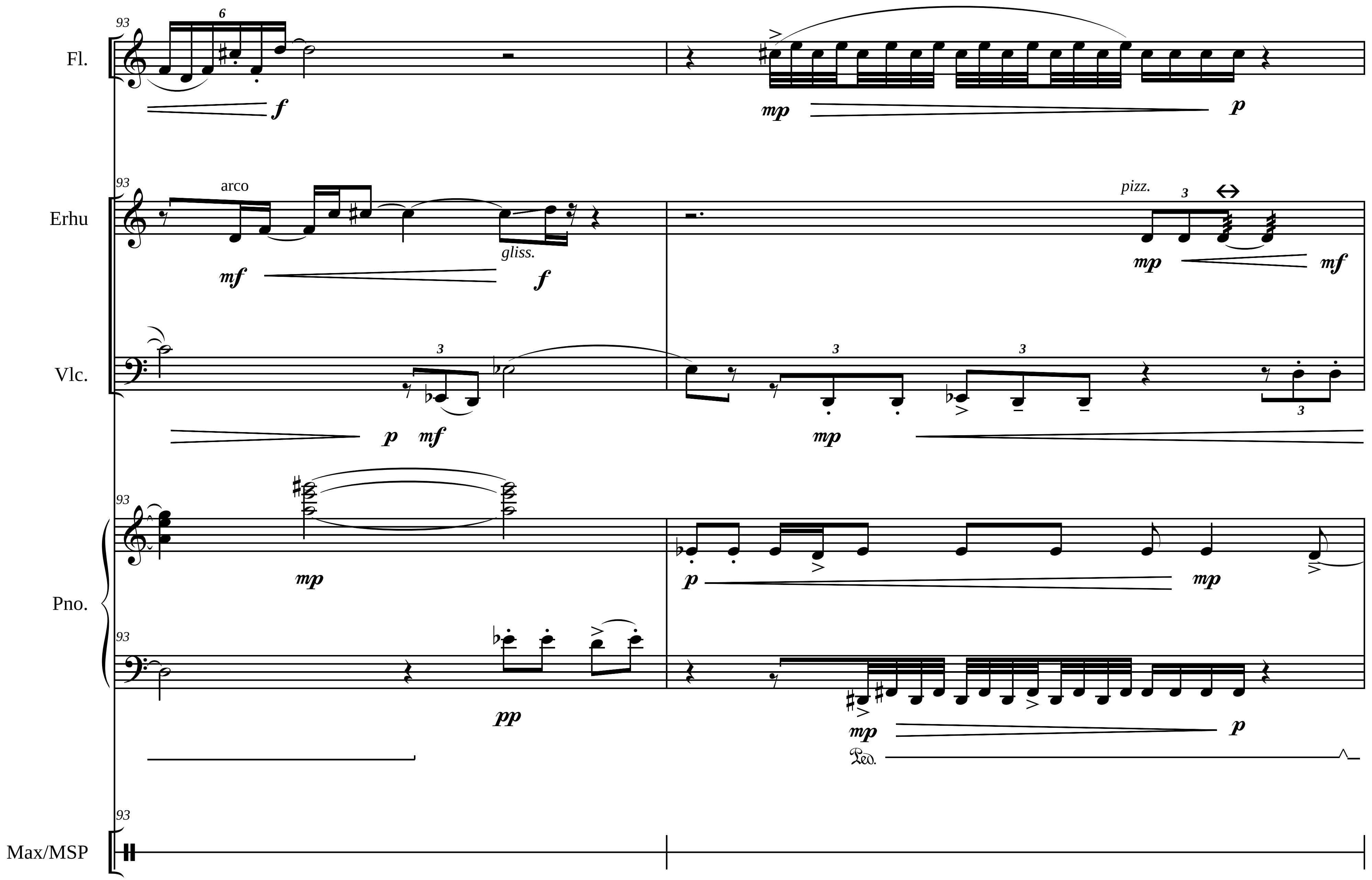


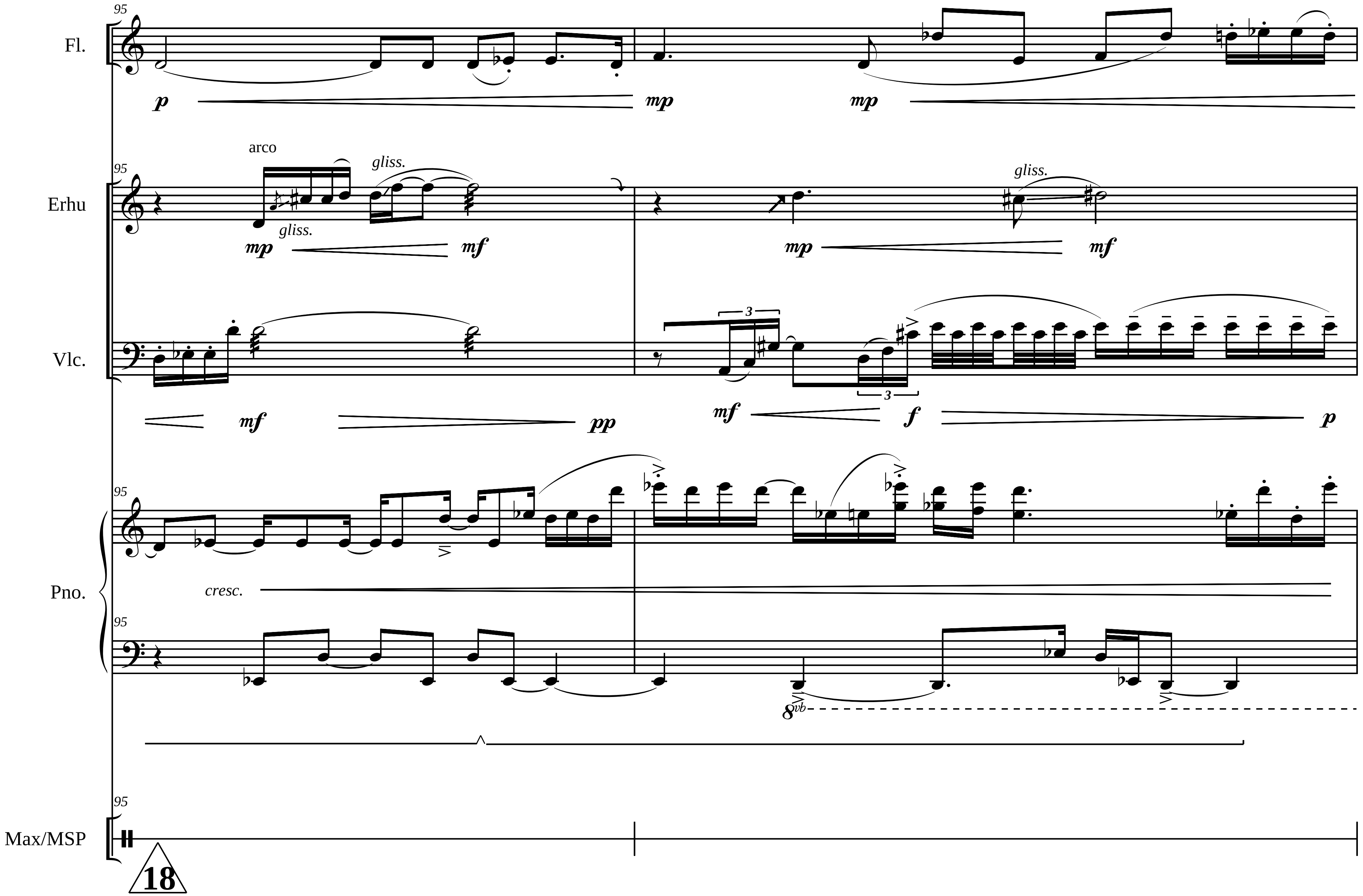
79
Fl.
Erhu
Vlc.
Pno.
Max/MSP
pizz.
R.H.
mf
p
mp
fp
f
3
7
15

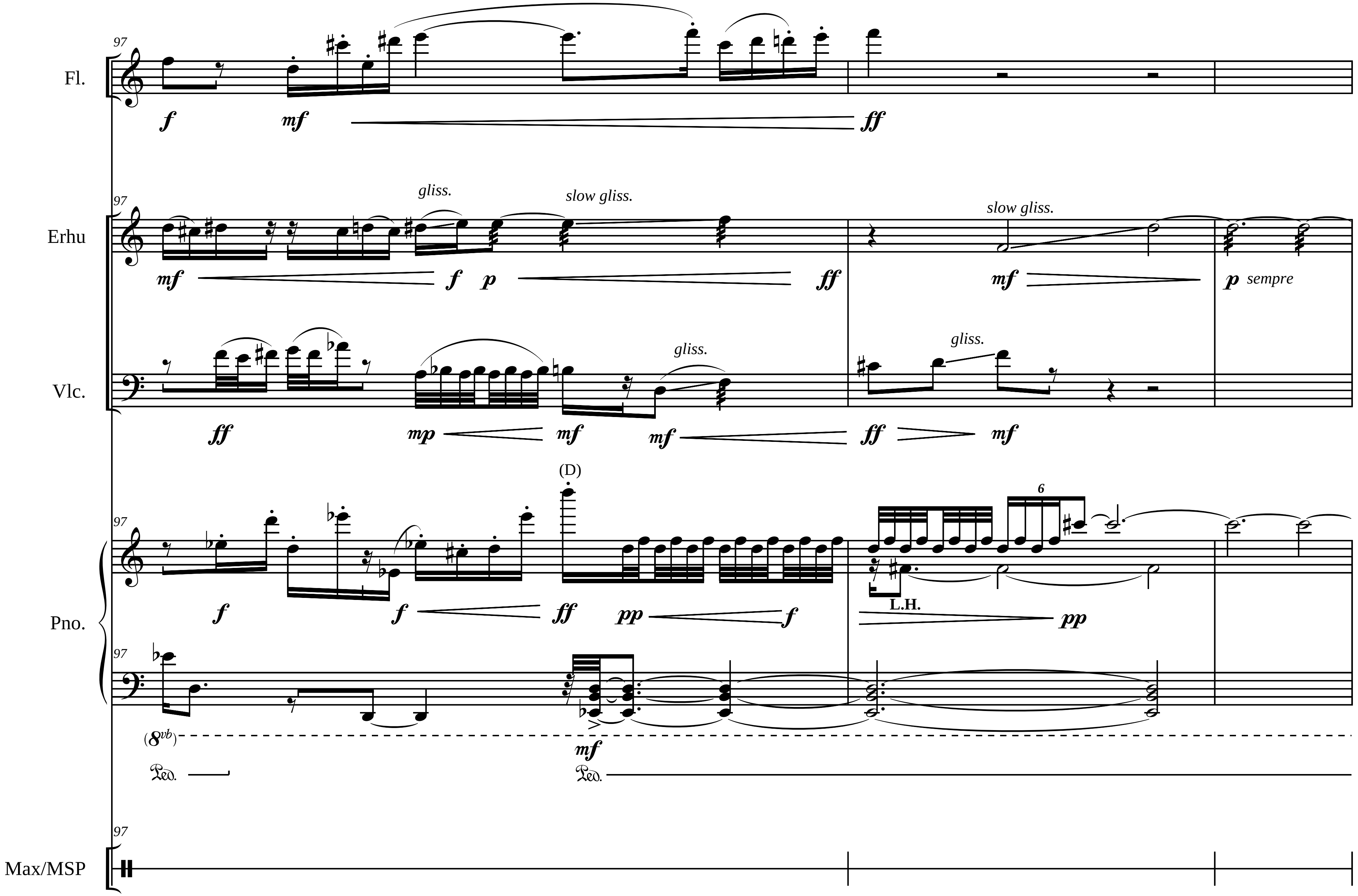

82
Fl.
mp
mf
mp
mf
mp
p
mf
f
Erhu
Vlc.
3
mp
mf
p
Pno.
L.H.
p
mf
mf
mf
f
Ped.
Max/MSP
16

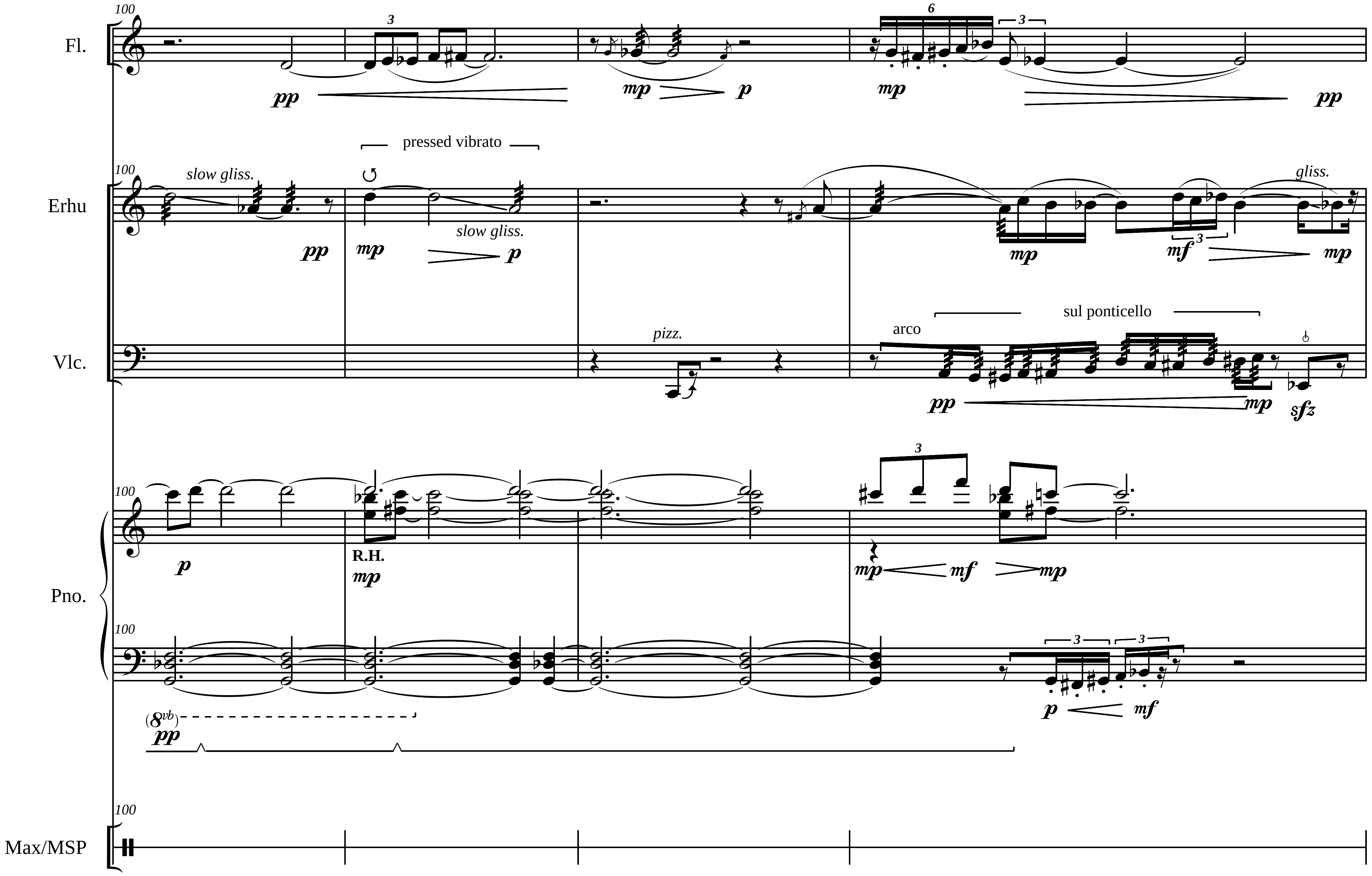

86
Fl.
Gliss.
mf
mf
mp
mf
f
f
mf
f
3
3
6
Erhu
arco
p
ppp
Vlc.
3
mf
pp
Pno.
f
sffz
Max/MSP
17

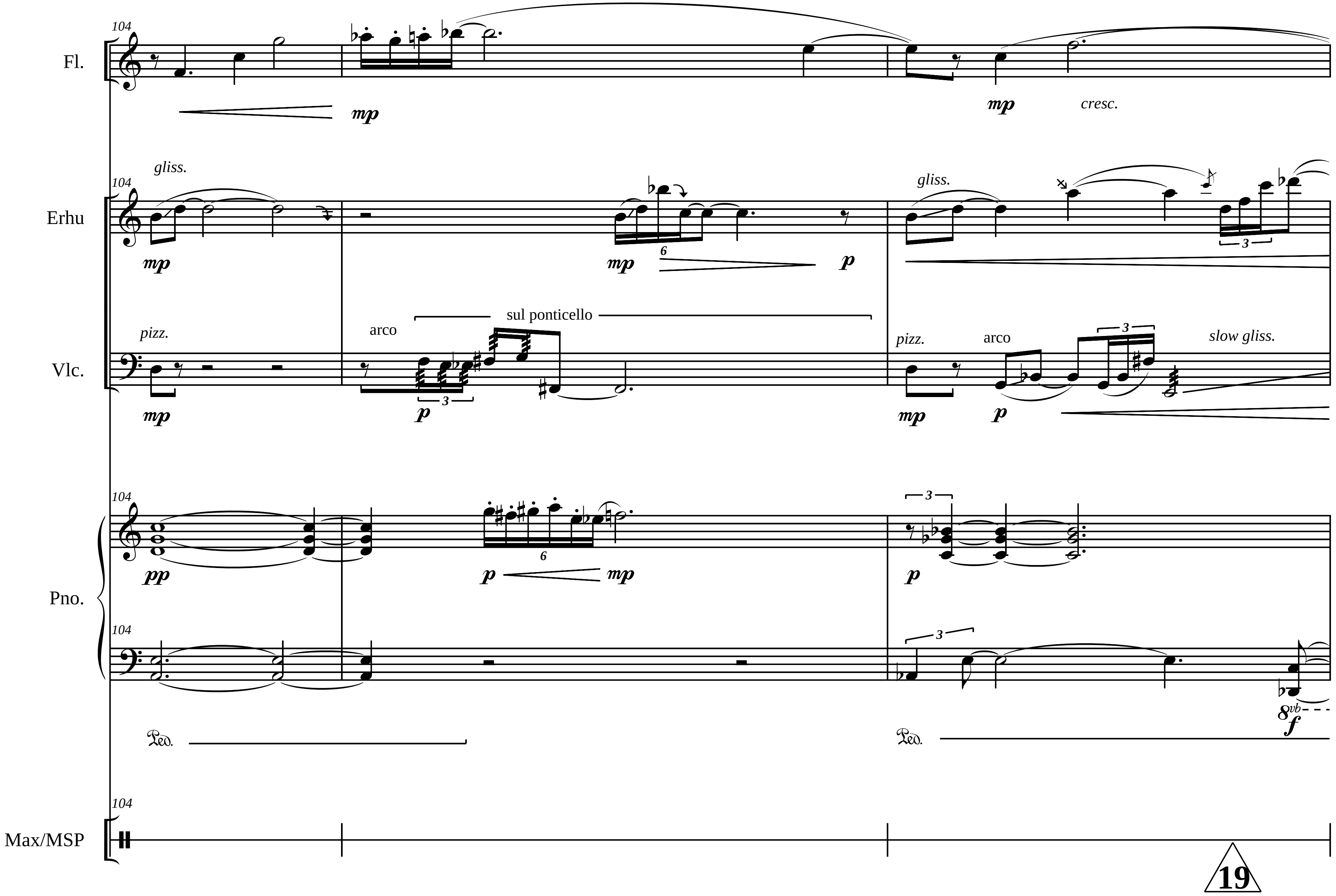

89
Fl.
Erhu
Vlc.
Pno.
Max/MSP
mp
mf
f
pizz.
p
pp
6
3
Ped.

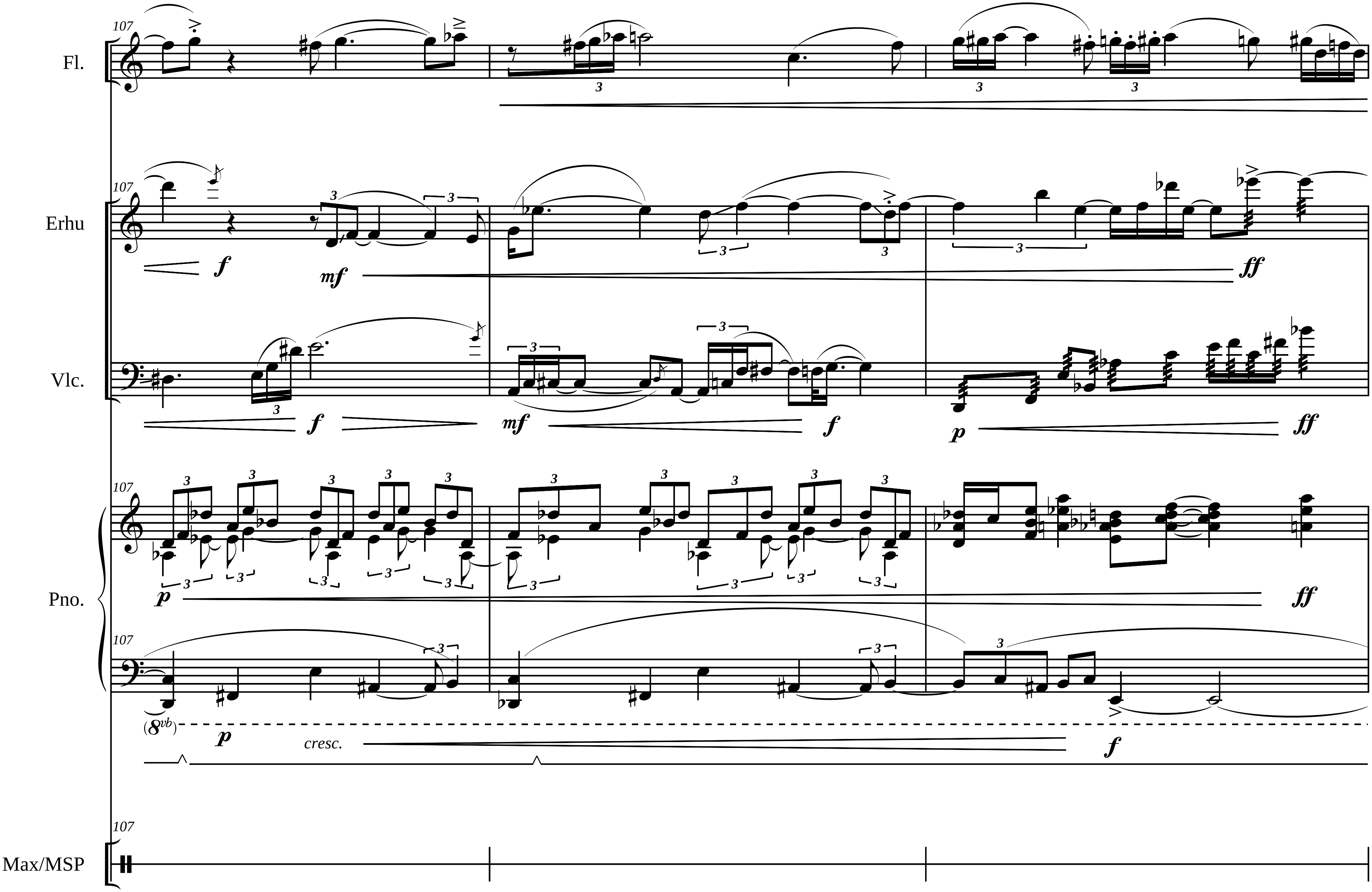

93
Fl.
6
f
mp
p
Erhu
arco
mf
gliss.
f
pizz.
3
mp
mf
Vlc.
3
p
mf
mp
Pno.
pp
Max/MSP

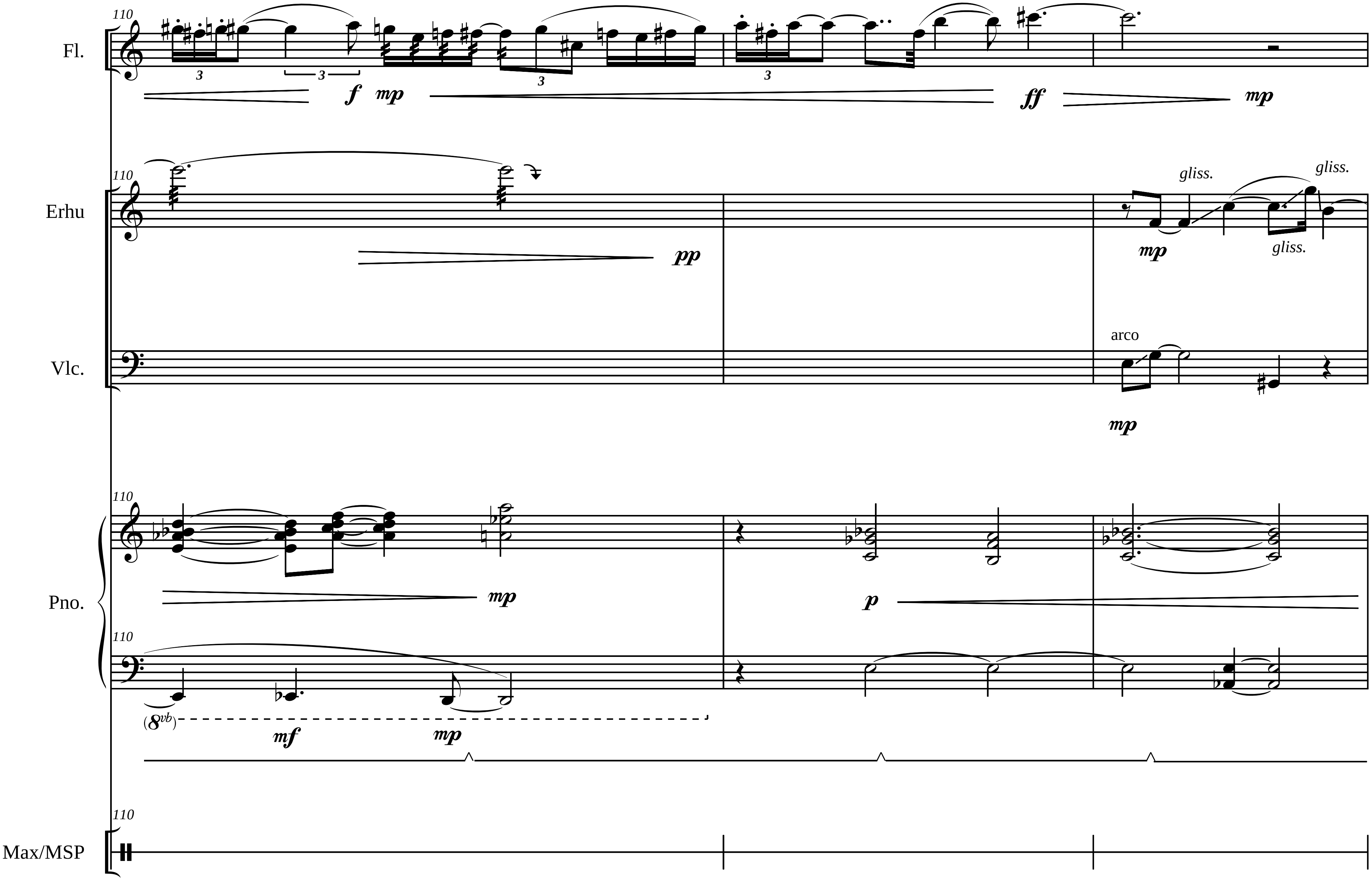

95
Fl.
p
mp
mp
Erhu
arco
gliss.
gliss.
mp
mf
mp
gliss.
mf
Vlc.
mf
pp
3
3
mf
f
p
Pno.
cresc.
8vb
Max/MSP
18

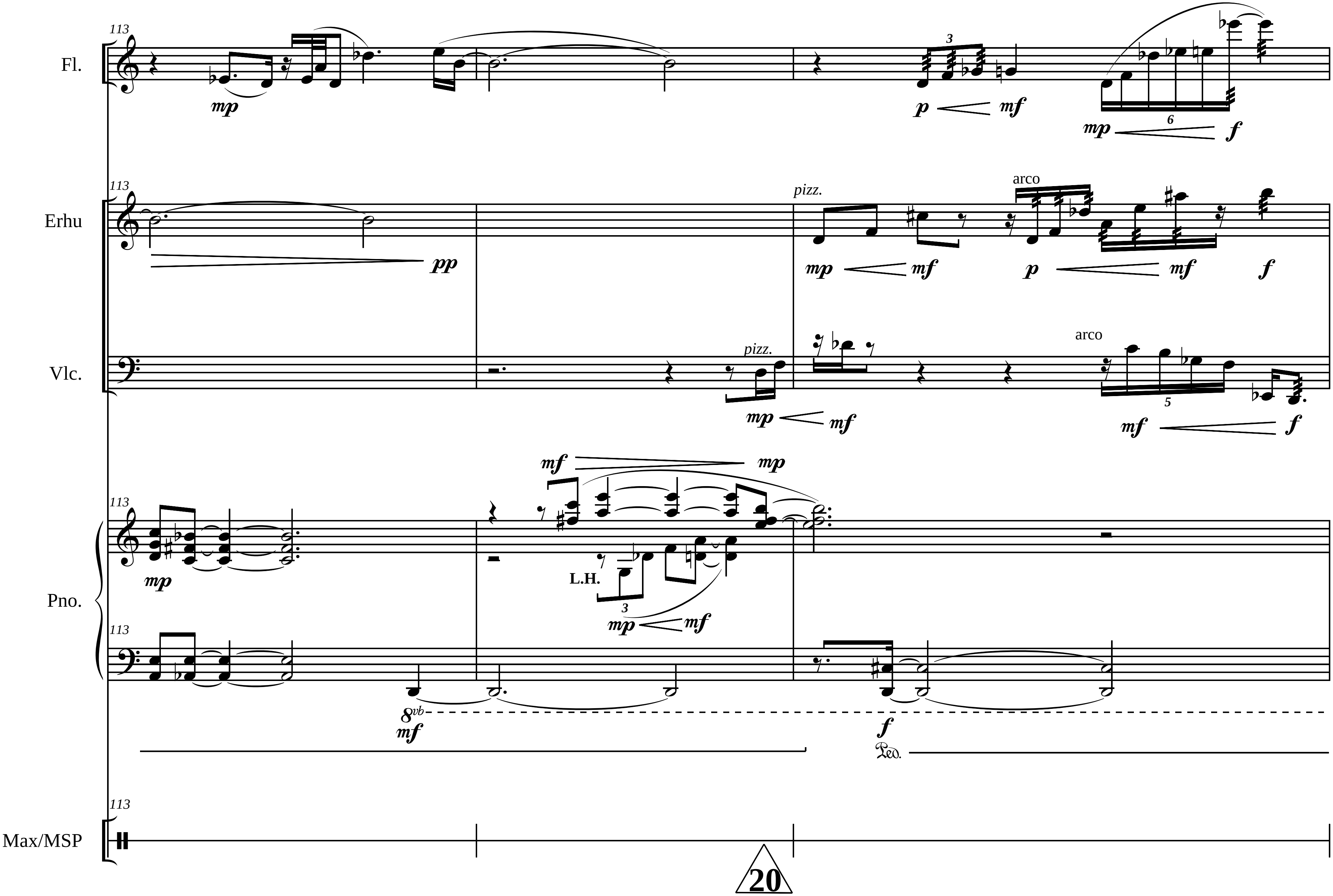

97
Fl.
f
mf
ff
Erhu
gliss.
slow gliss.
mf
f
p
ff
slow gliss.
mf
p sempre
Vlc.
ff
mp
mf
gliss.
mf
ff
gliss.
mf
(D)
Pno.
f
f
ff
pp
f
6
L.H.
pp
(8vb)
mf
Ped.
Ped.
Max/MSP

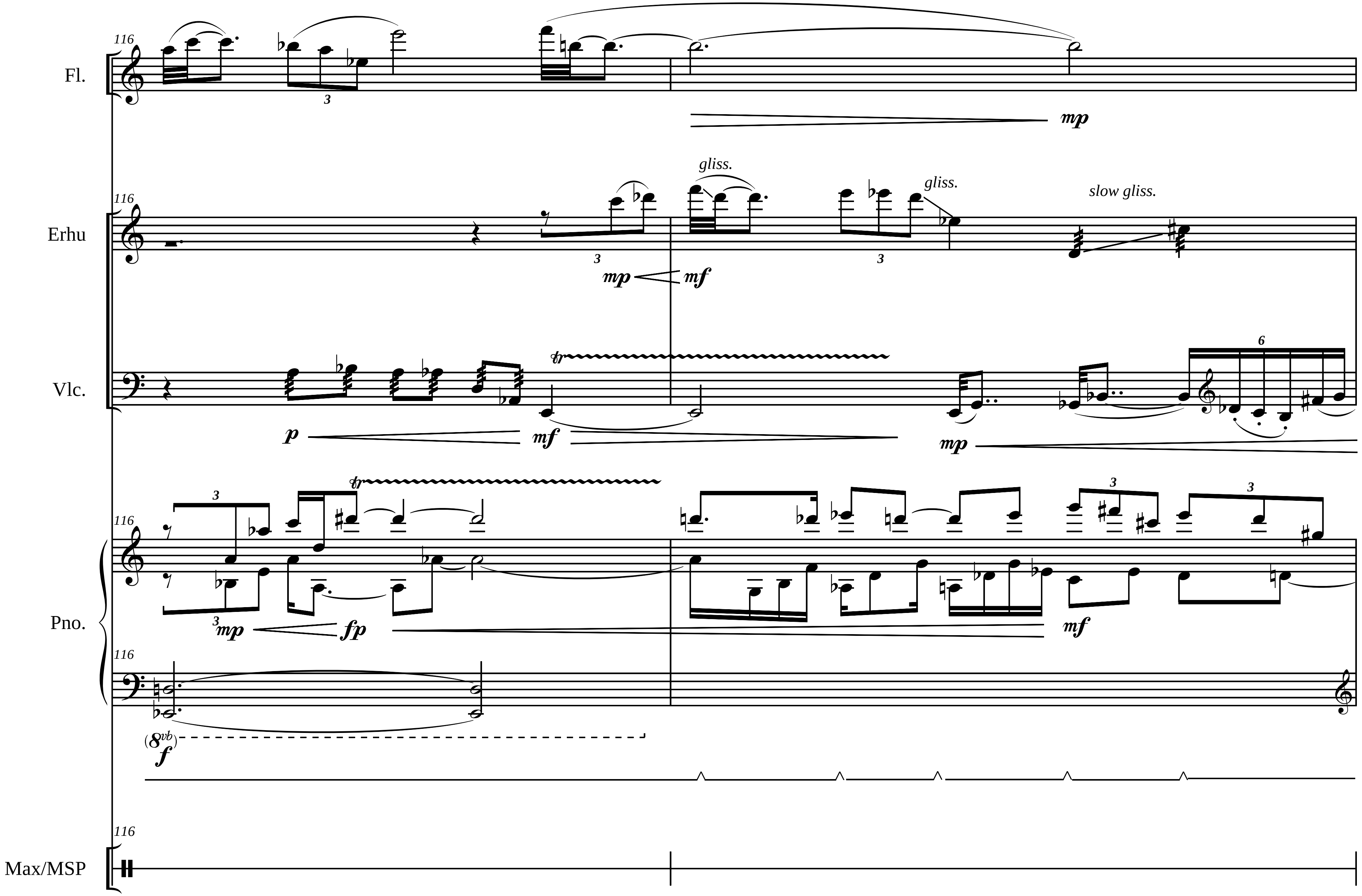

100
Fl.
3
pp
mp
p
6
3
mp
pp
pressed vibrato
Erhu
slow gliss.
pp
mp
slow gliss.
p
mp
mf
3
mp
gliss.
sul ponticello
arco
Vlc.
pizz.
pp
mp
sfz
3
Pno.
p
R.H.
mp
mp
mf
mp
3
3
p
mf
(8vb)
pp
Max/MSP

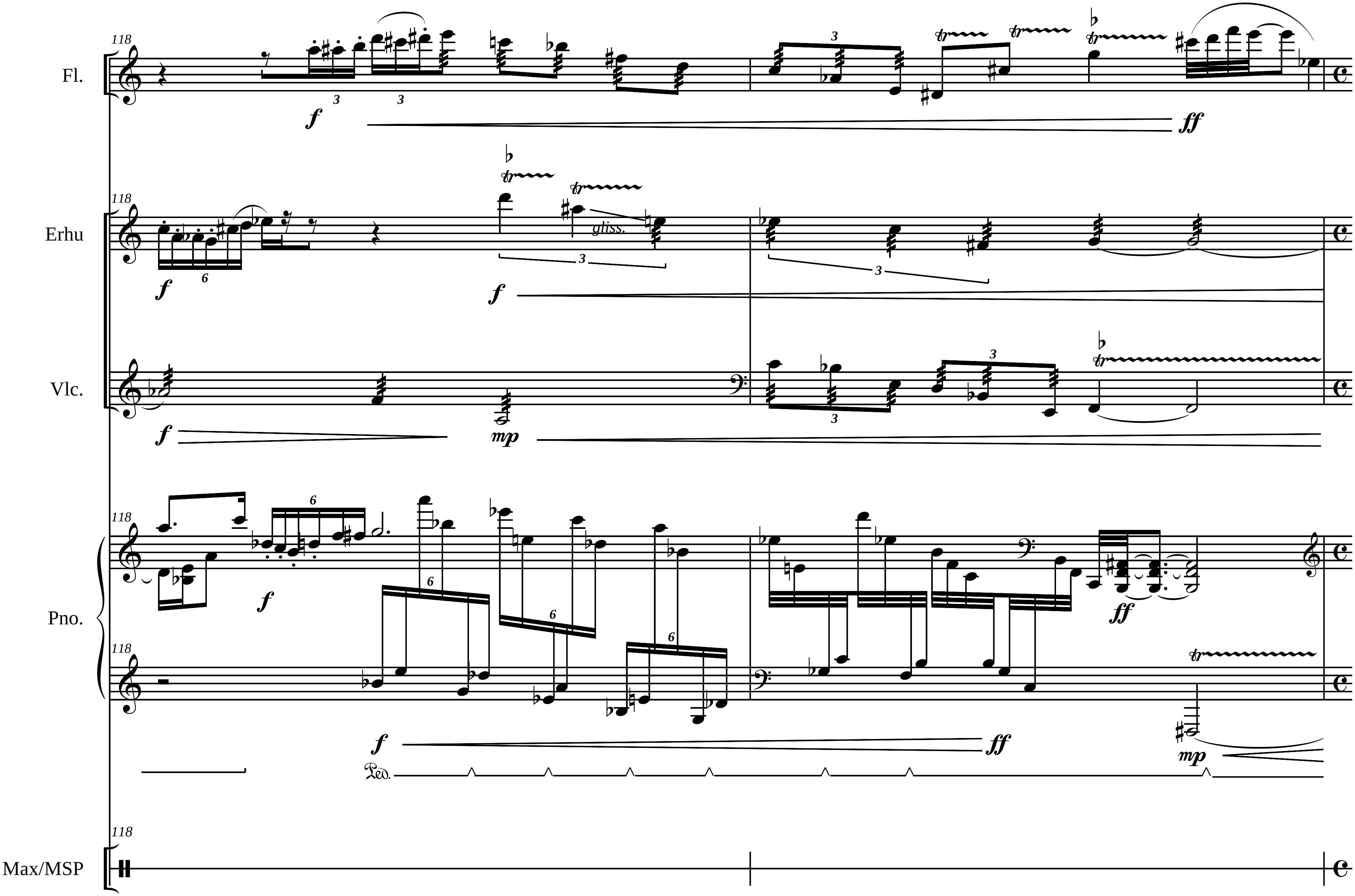

104
Fl.
mp
mp
cresc.
Erhu
gliss.
mp
6
mp
p
gliss.
3
Vlc.
pizz.
mp
arco
sul ponticello
3
p
pizz.
arco
3
slow gliss.
mp
p
Pno.
pp
6
p
mp
3
p
3
8vb
f
Ped.
Ped.
Max/MSP
19

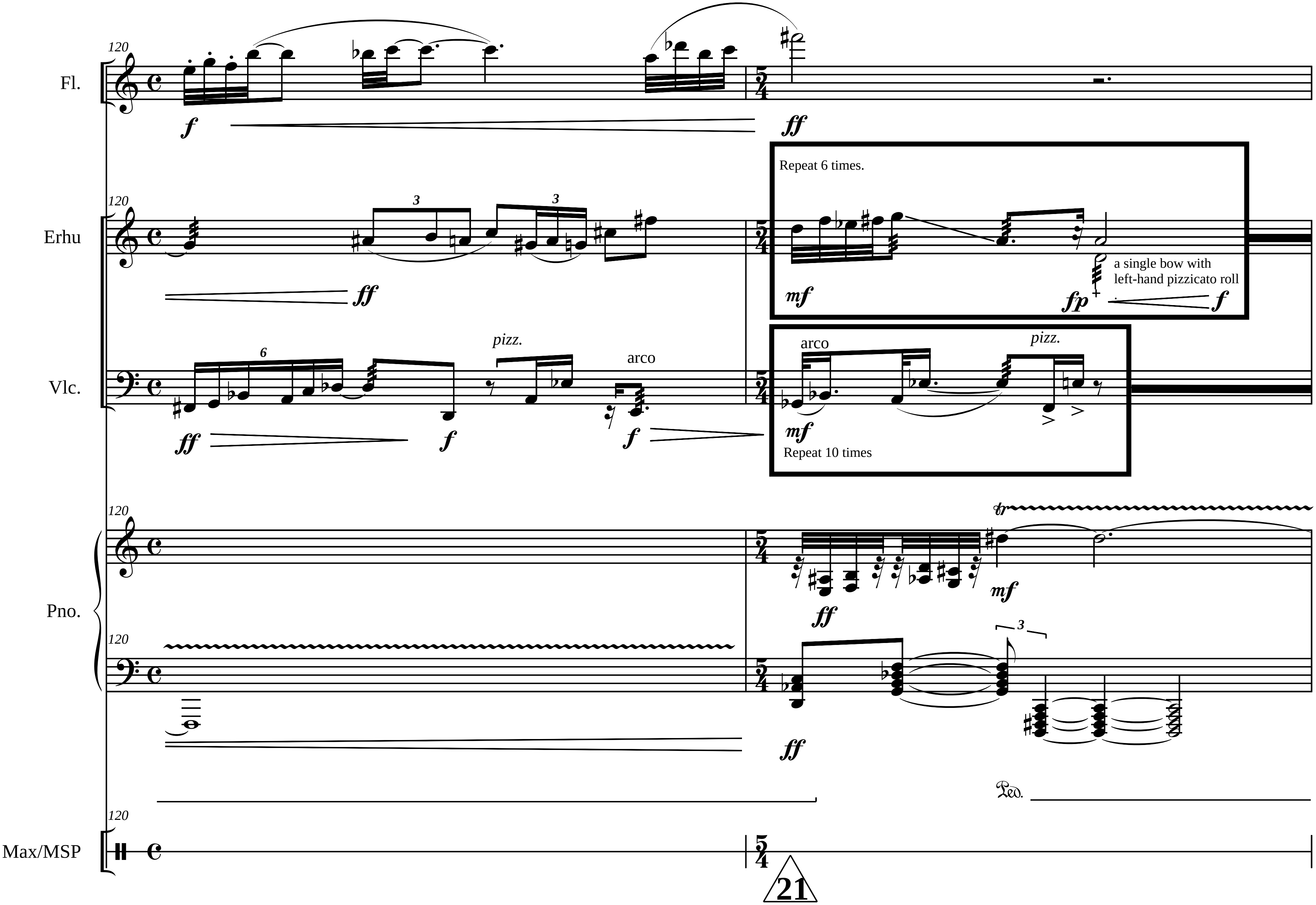

107
Fl.
Erhu
Vlc.
Pno.
Max/MSP
f
mf
ff
p
cresc.
(8vb)
3

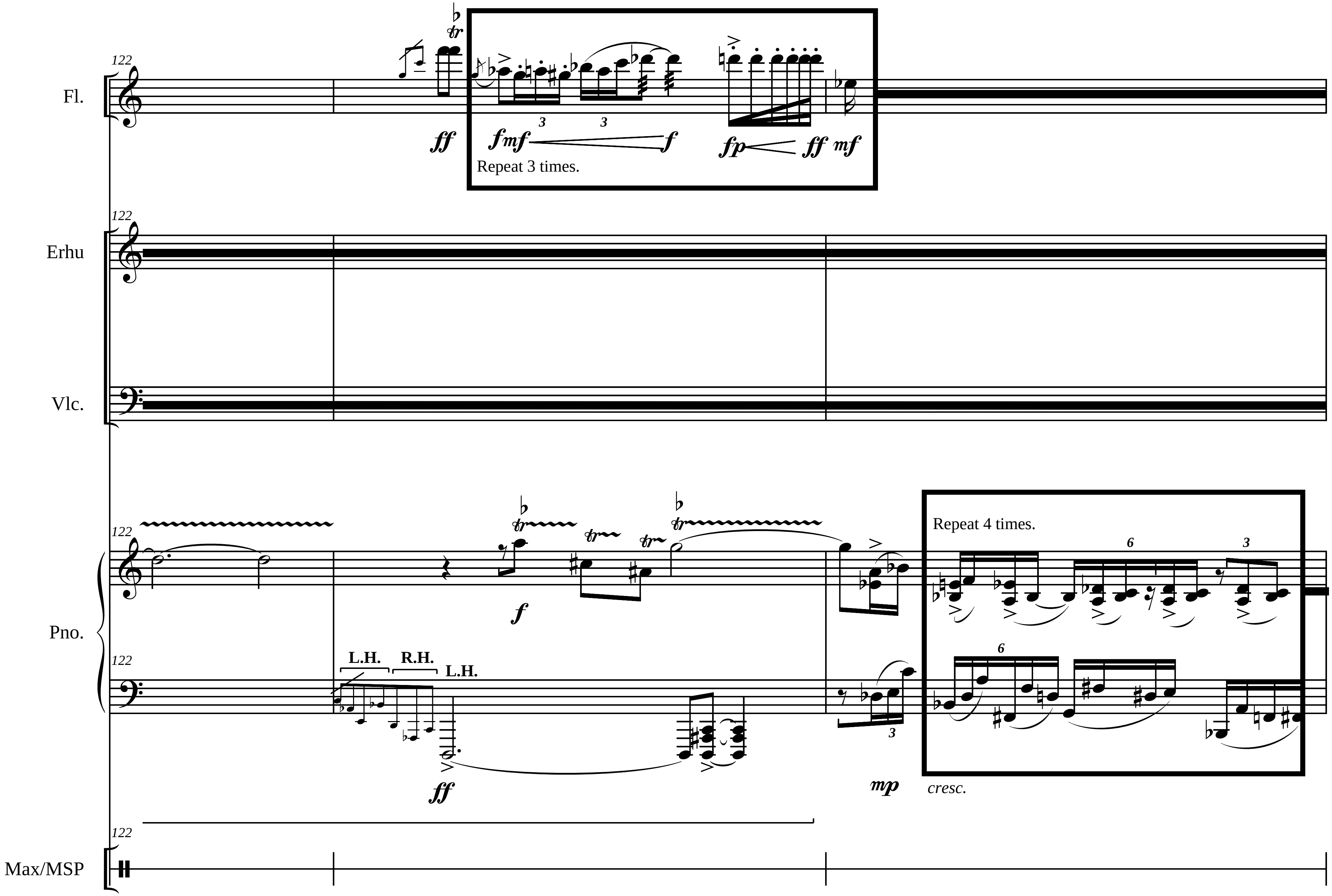

110
Fl.
3
3
3
3
f
mp
ff
mp
110
Erhu
pp
gliss.
gliss.
gliss.
mp
arco
Vlc.
mp
110
Pno.
mp
p
110
(8vb)
mf
mp
110
Max/MSP

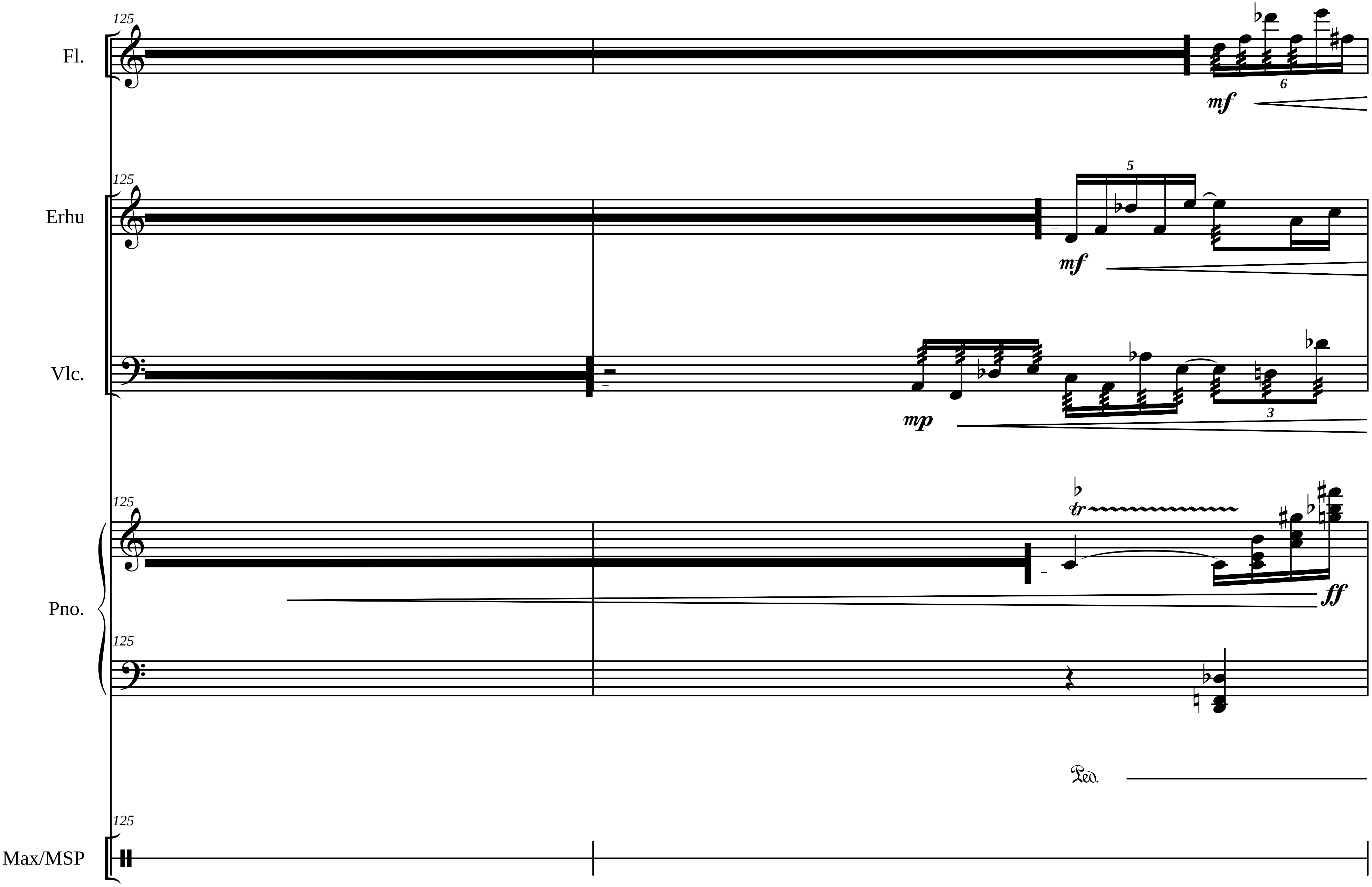

113
Fl.
mp
3
p
mf
mp
6
f
113
Erhu
pp
pizz.
mp
mf
arco
p
mf
f
Vlc.
pizz.
mp
mf
arco
5
mf
f
mf
mp
113
Pno.
mp
L.H.
3
mp
mf
113
8vb
mf
f
Ped.
113
Max/MSP
20

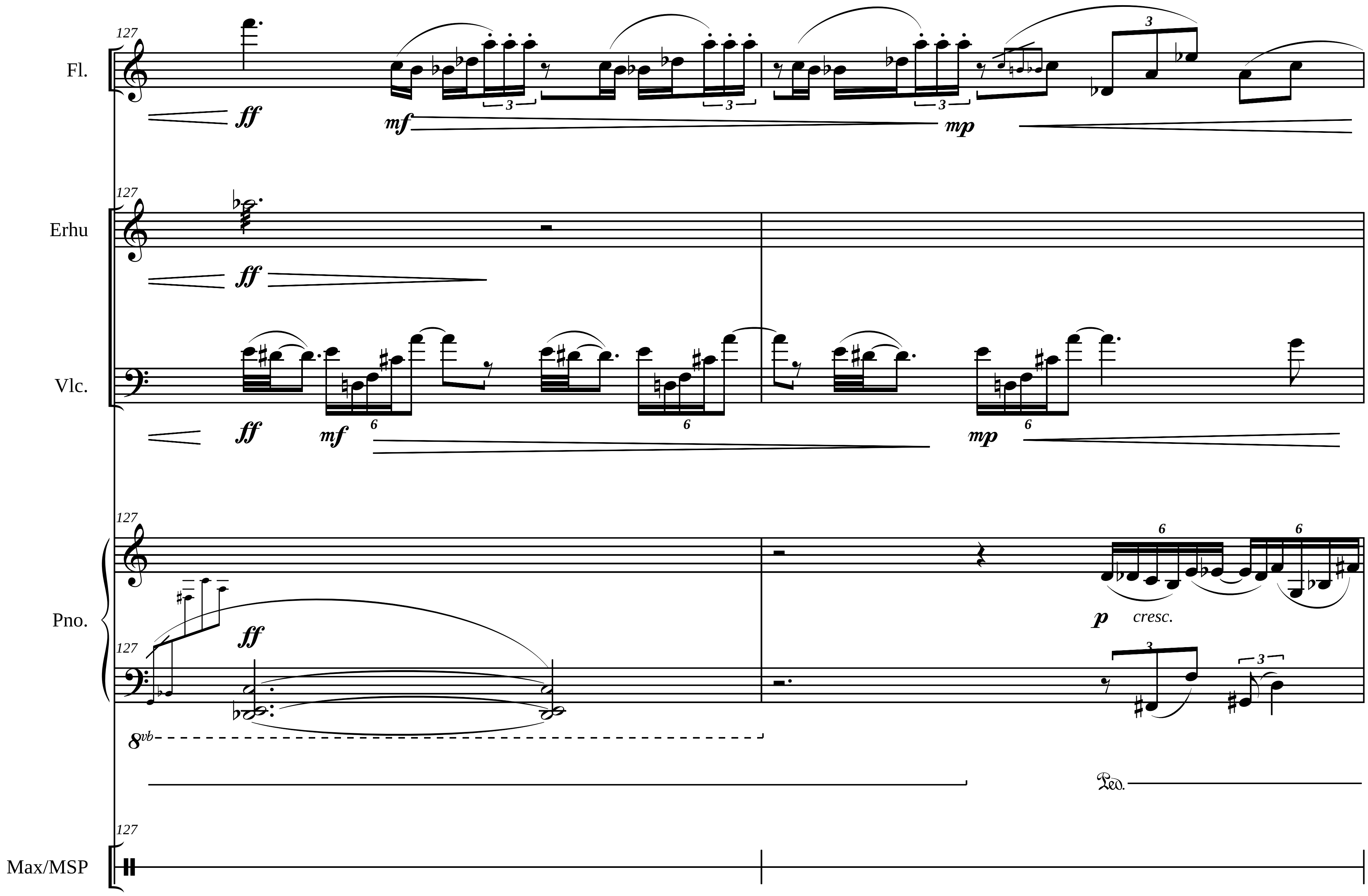

116
Fl.
3
mp
116
Erhu
gliss.
gliss.
slow gliss.
3
3
mp
mf
Vlc.
tr
6
p
mf
mp
116
Pno.
3
3
tr
3
3
mp
fp
mf
116
(8vb)
f
116
Max/MSP

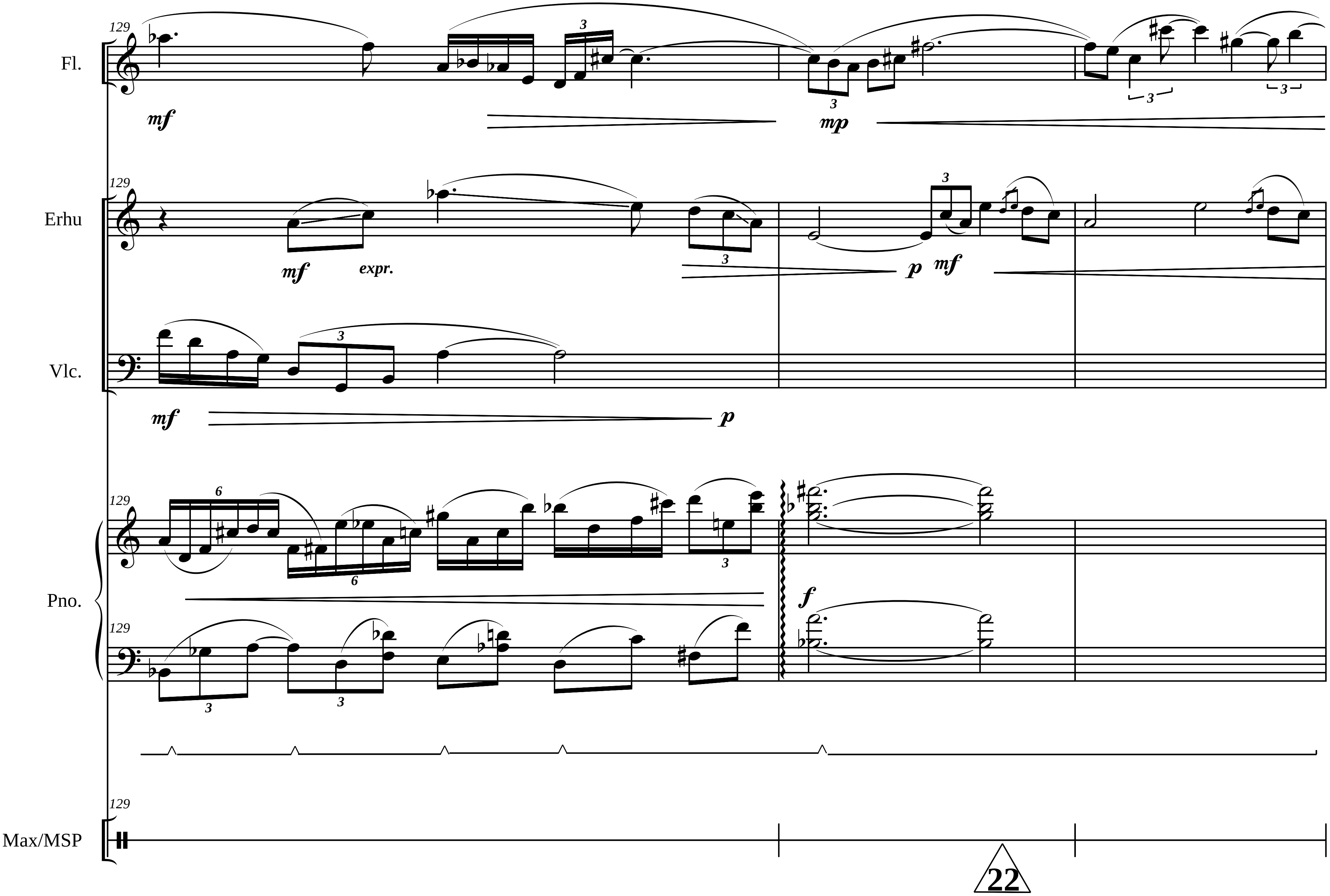
118
Fl.
Erhu
Vlc.
Pno.
Max/MSP
f
ff
mp
gliss.
3
6
Ped.

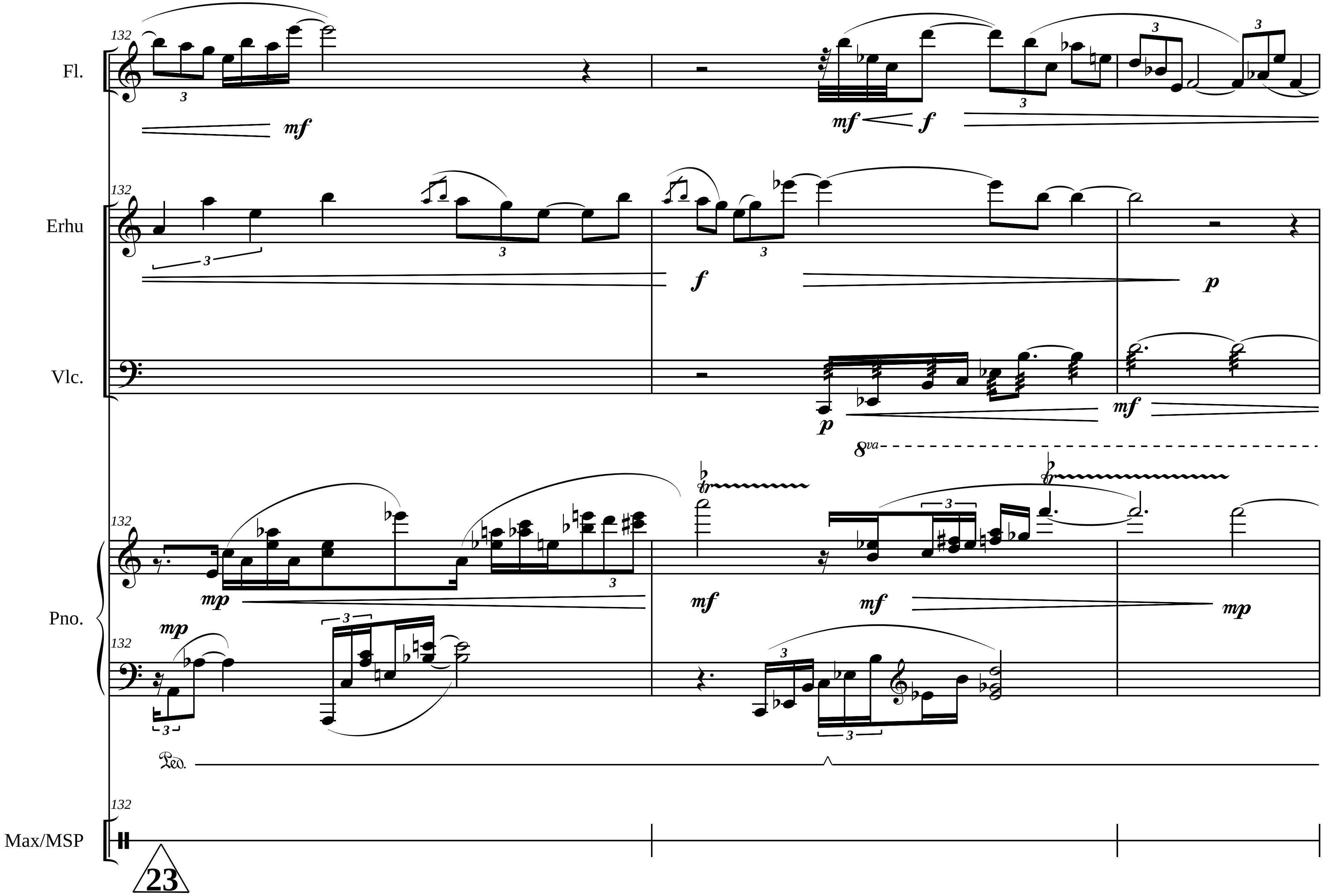

120
Fl.
f
ff
Repeat 6 times.
Erhu
3
3
ff
mf
fp
a single bow with
left-hand pizzicato roll
f
Vlc.
6
pizz.
arco
ff
f
f
arco
pizz.
mf
Repeat 10 times
Pno.
tr
mf
ff
3
ff
Ped.
Max/MSP
21

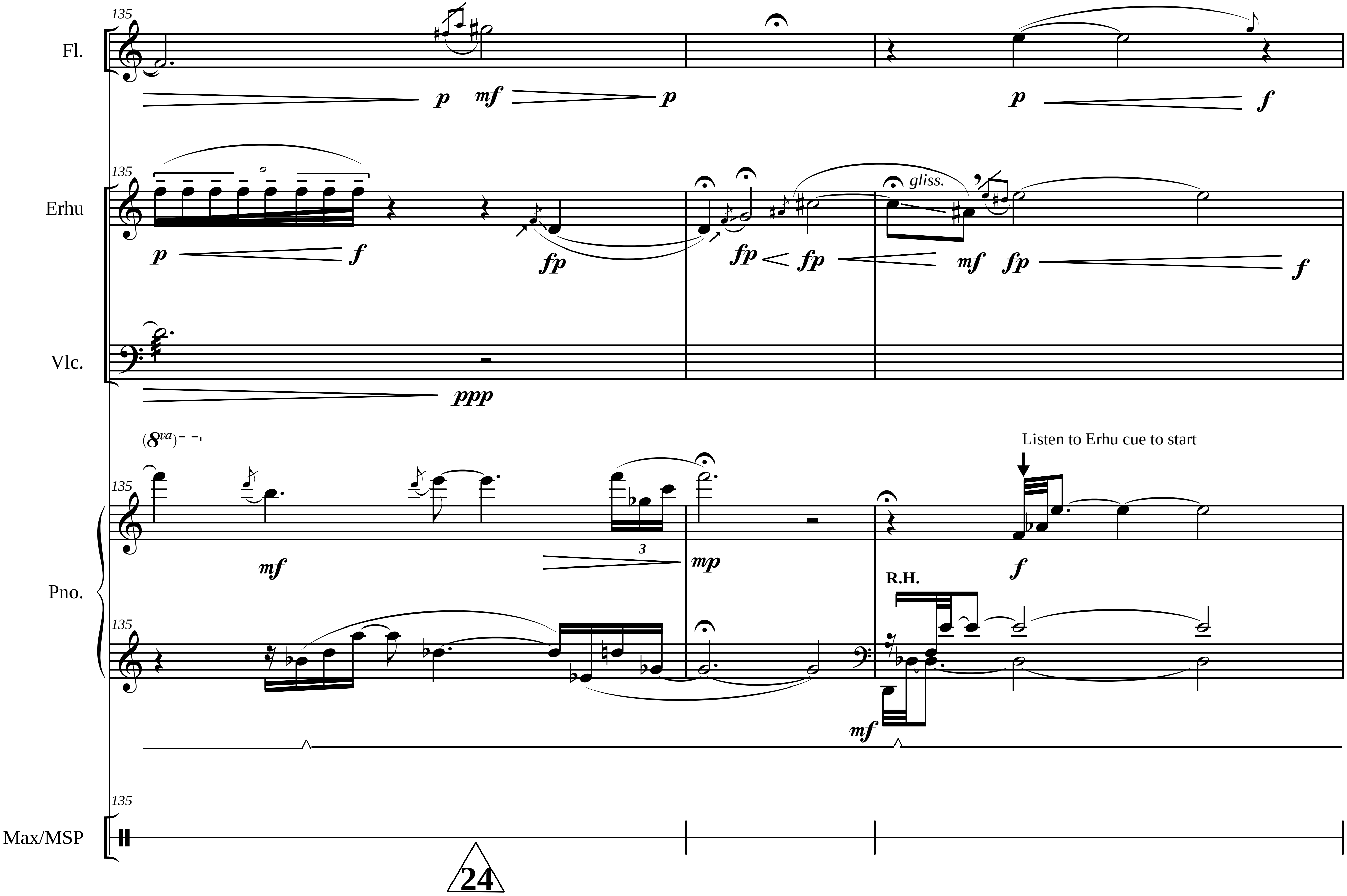

122
Fl.
Erhu
Vlc.
Pno.
Max/MSP
ff
fmf
f
fp
ff
mf
Repeat 3 times.
L.H.
R.H.
L.H.
f
ff
mp
Repeat 4 times.
cresc.
3
6

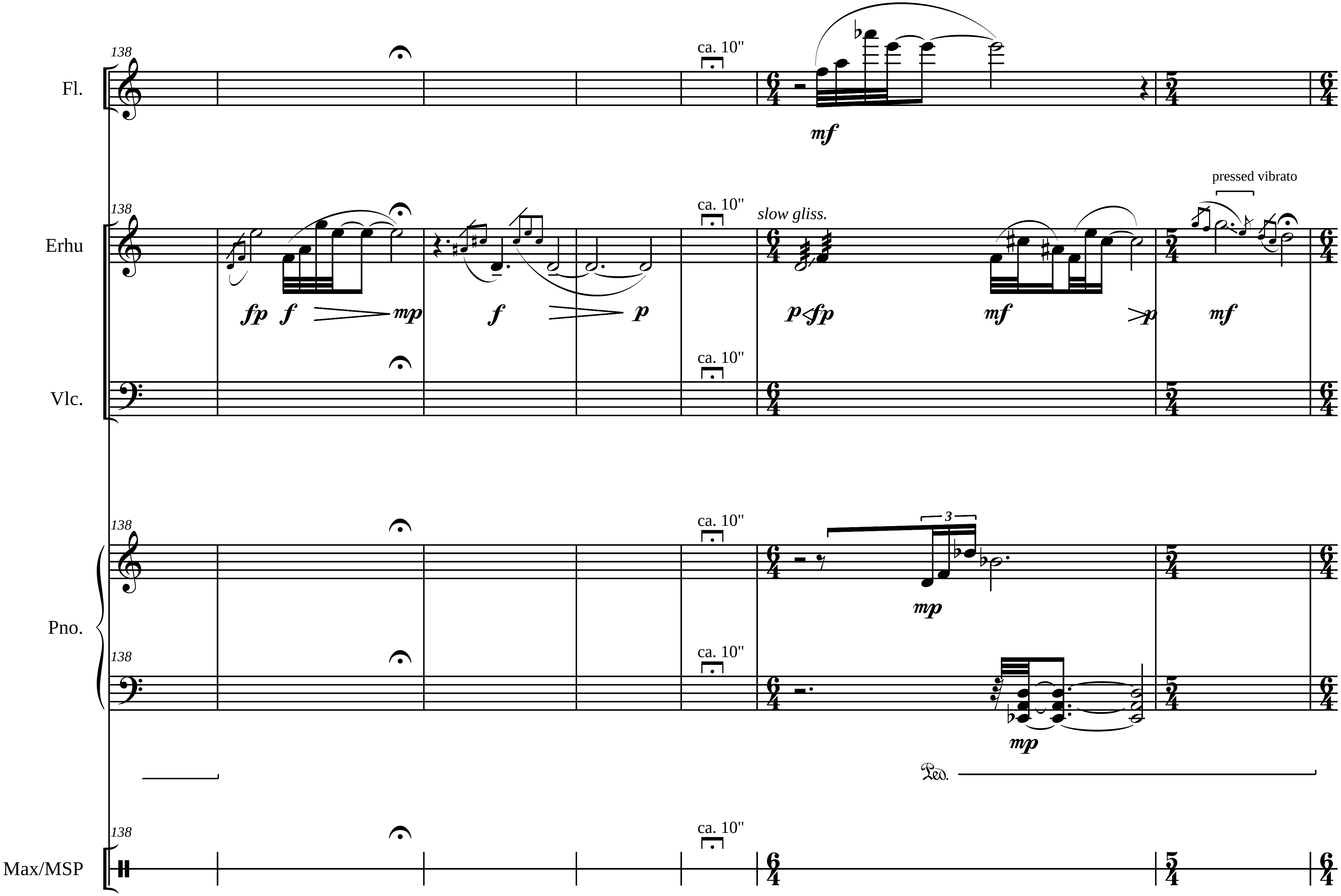

125
Fl.
mf
6
Erhu
5
mf
Vlc.
mp
3
Pno.
tr
ff
Ped.
Max/MSP

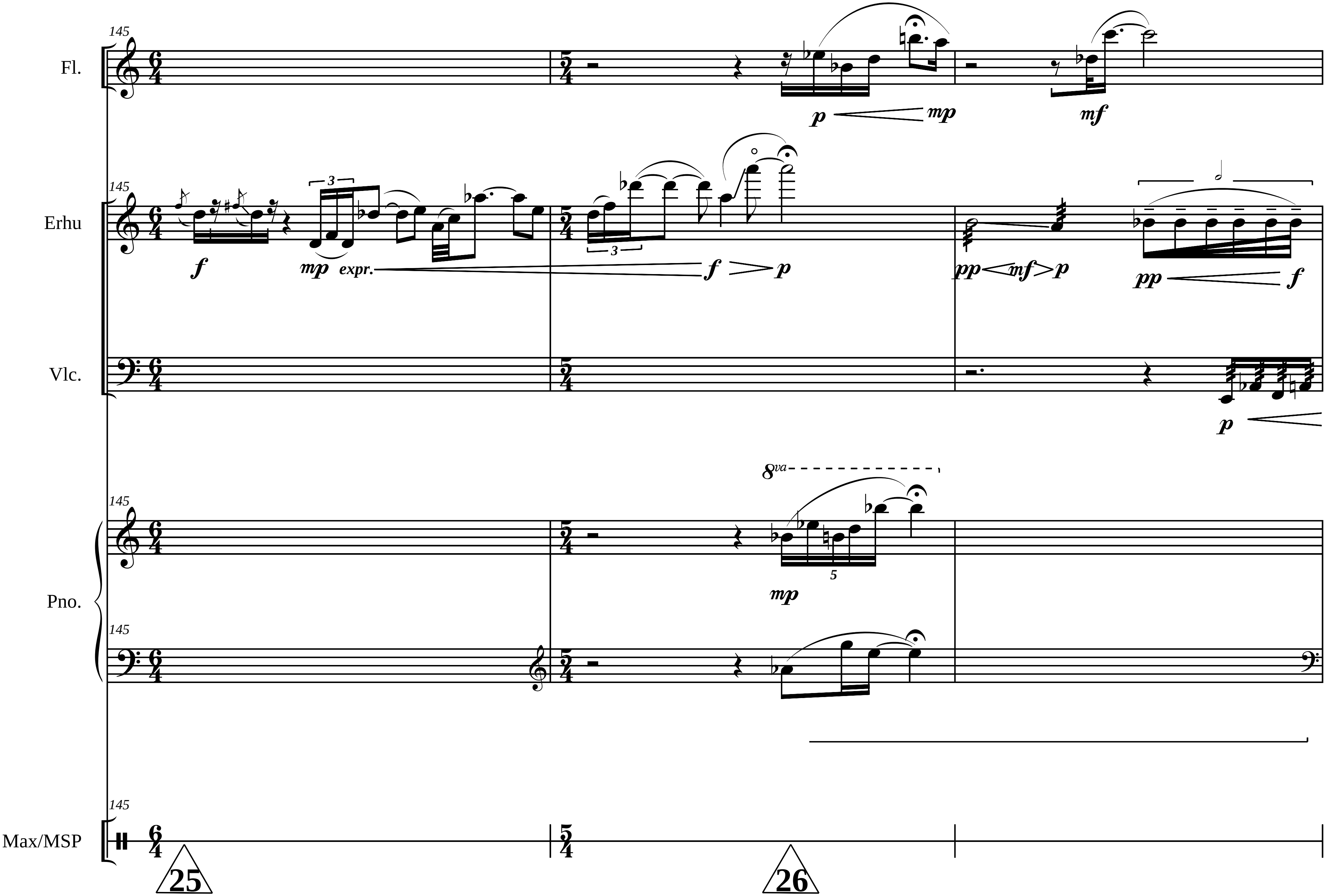

127
Fl.
ff
mf
mp
3
Erhu
Vlc.
6
Pno.
p
cresc.
8vb
Max/MSP

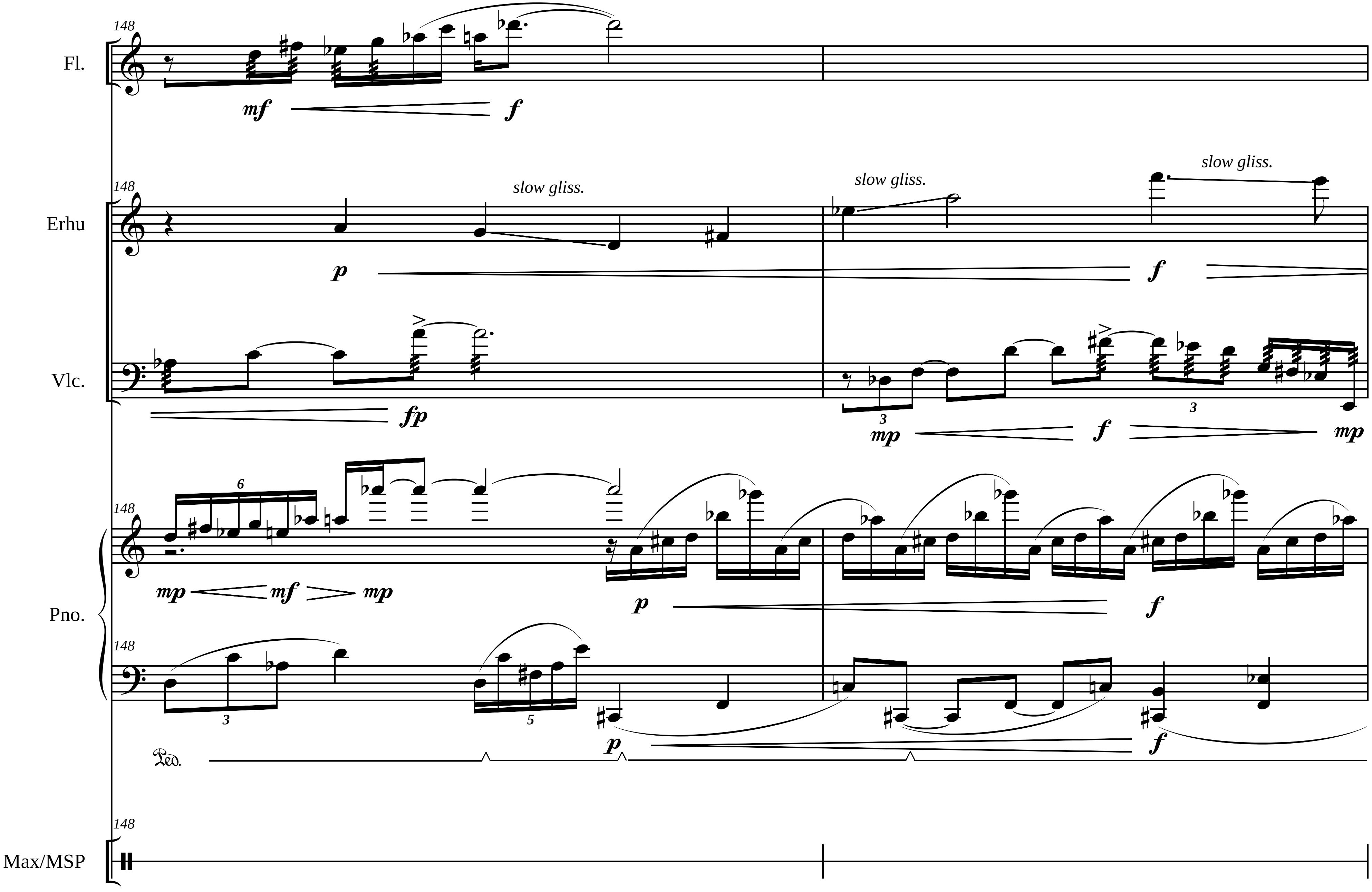

129
Fl.
mf
3
mp
Erhu
mf
expr.
p
Vlc.
Pno.
6
f
Max/MSP
22

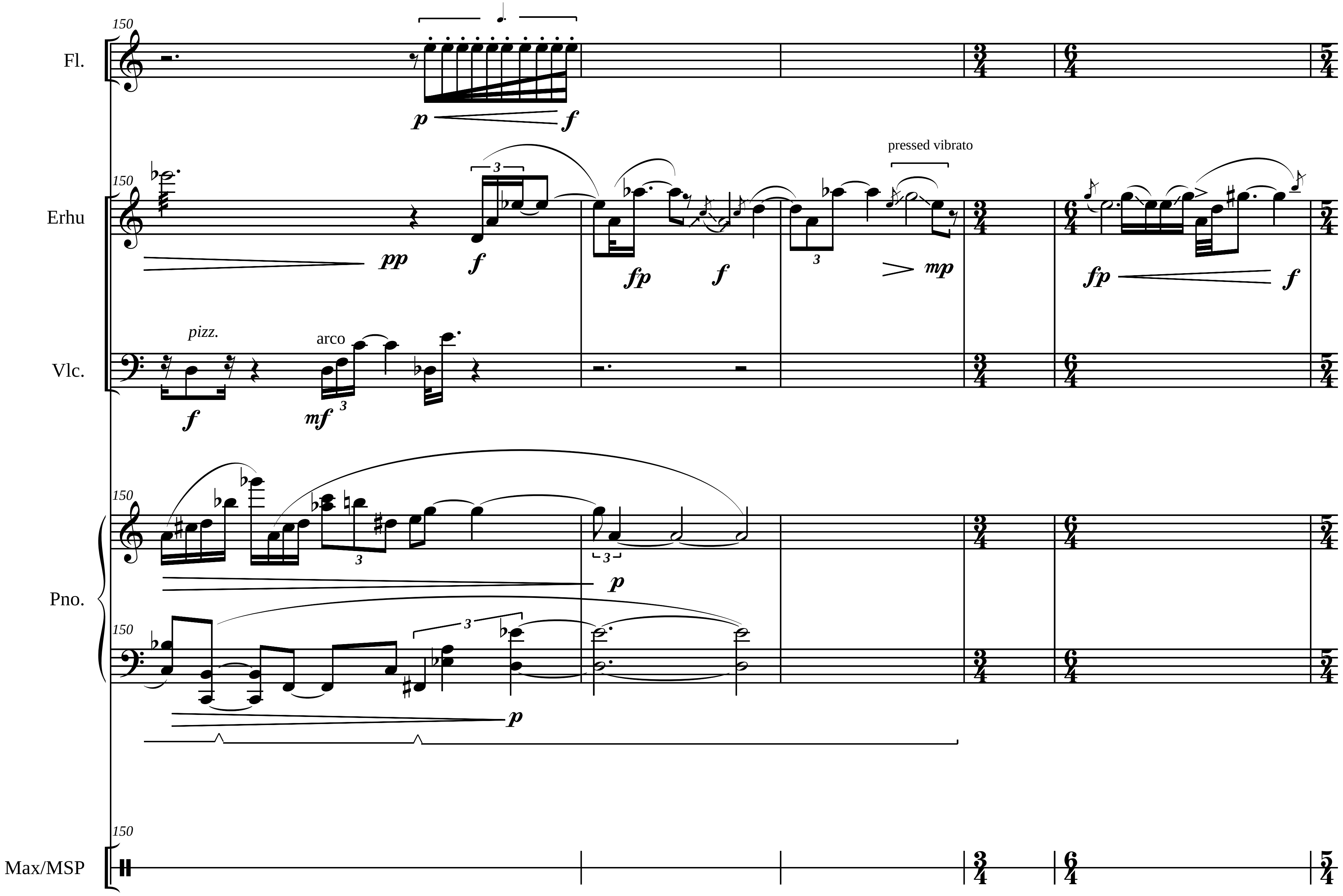

132
Fl.
Erhu
Vlc.
Pno.
Max/MSP
mf
f
p
mp
8va
Ped.
23

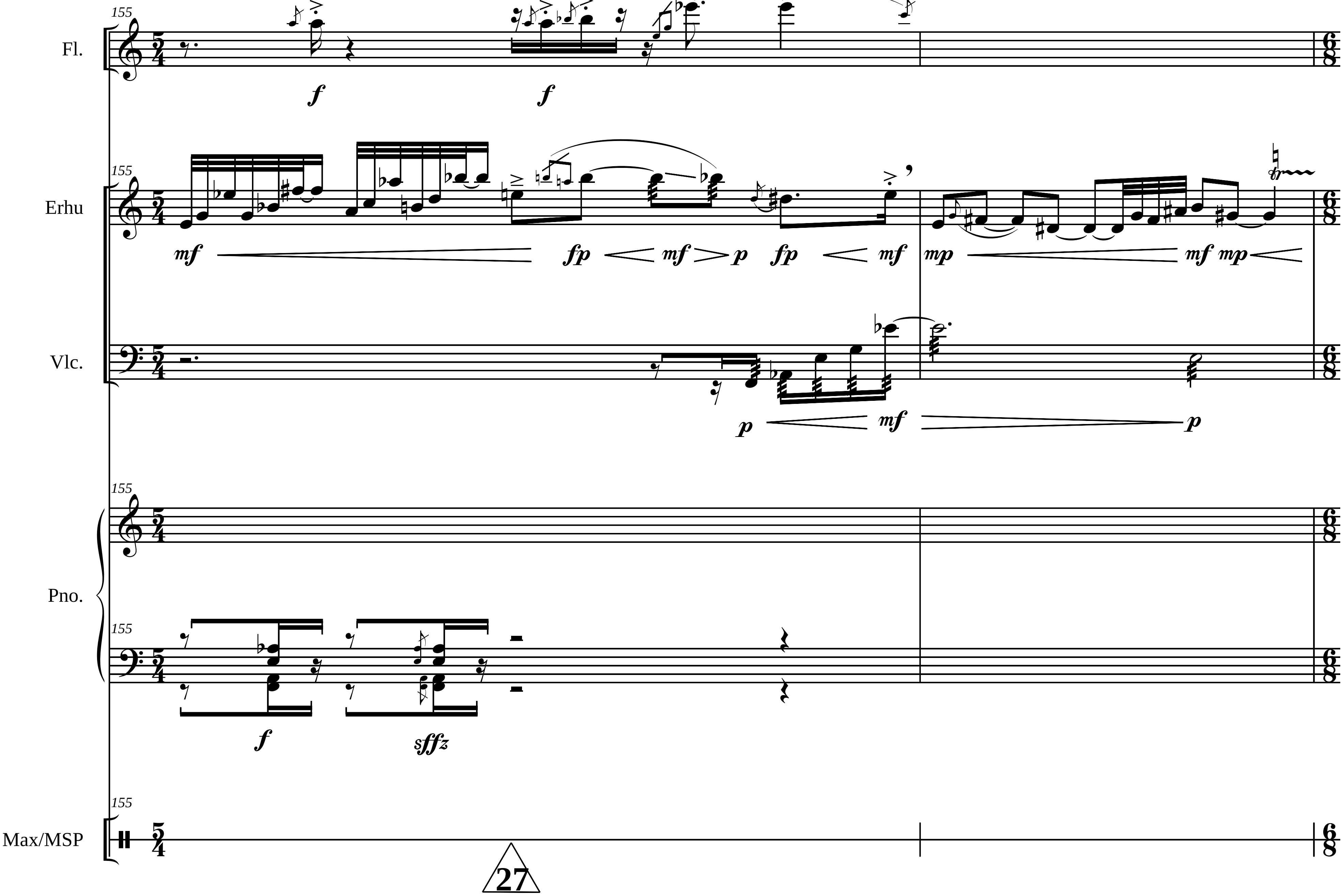

Listen to Erhu cue to start
135
Fl.
p
mf
p
p
f
Erhu
gliss.
fp
fp
fp
mf
fp
f
Vlc.
ppp
(8va)
Listen to Erhu cue to start
Pno.
mf
3
mp
R.H.
f
mf
Max/MSP
24

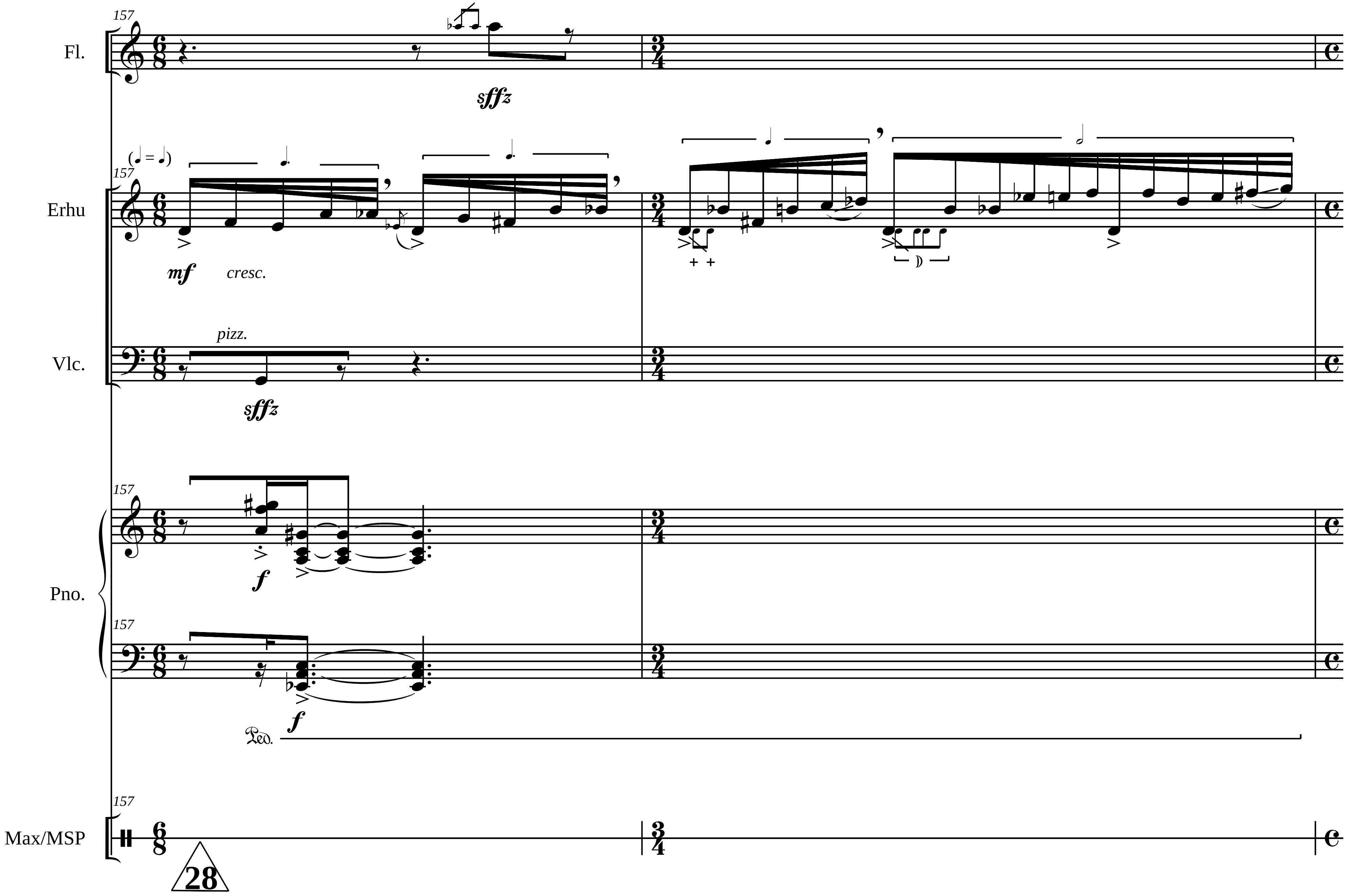

138
Fl.
ca. 10"
mf
Erhu
slow gliss.
fp f mp
f p
p fp
mf
p
pressed vibrato
mf
Vlc.
Pno.
3
mp
mp
Ped.
Max/MSP

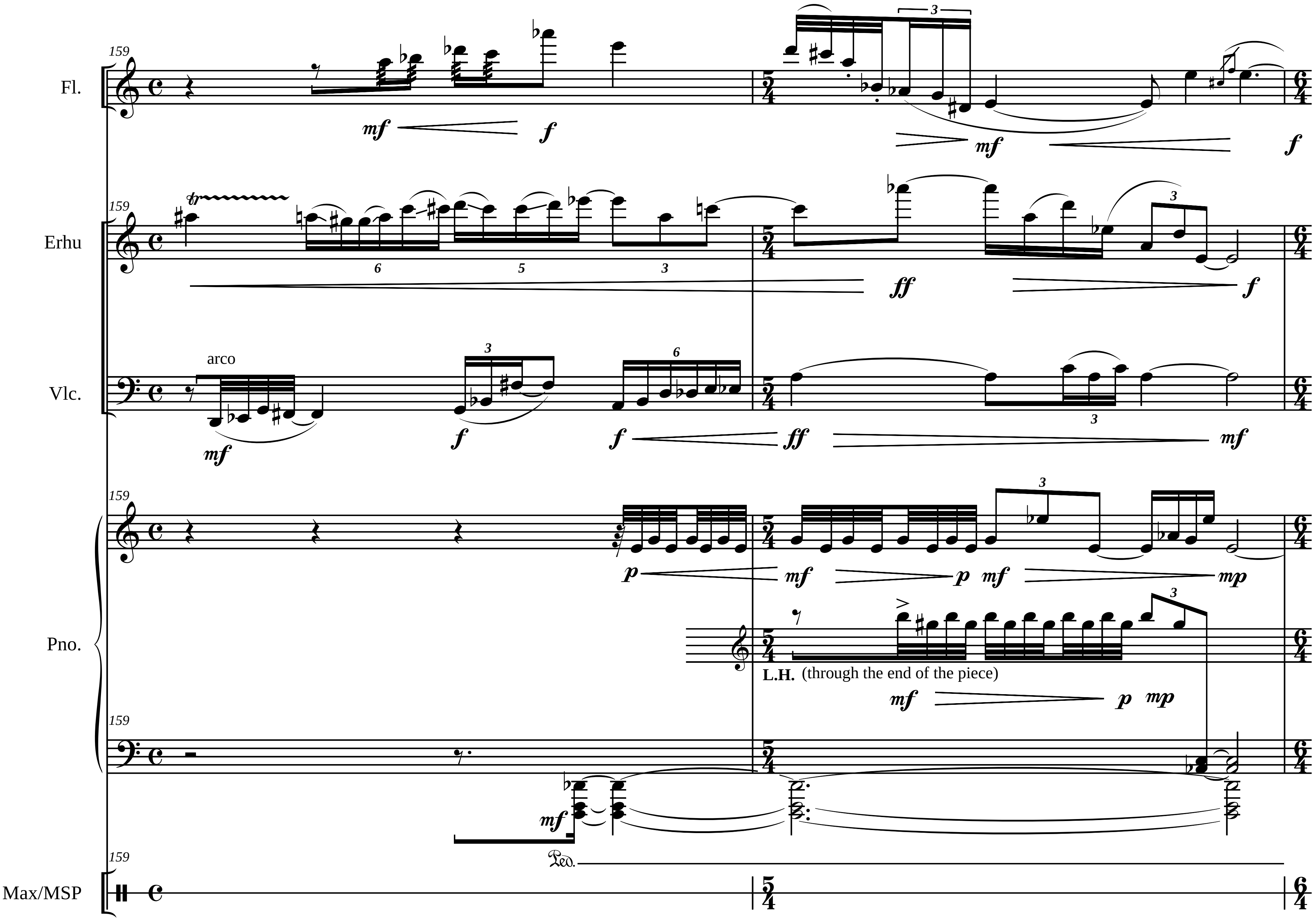

145
Fl.
Erhu
Vlc.
Pno.
Max/MSP
p
mp
mf
f
expr.
pp
3
5
8va
25
26

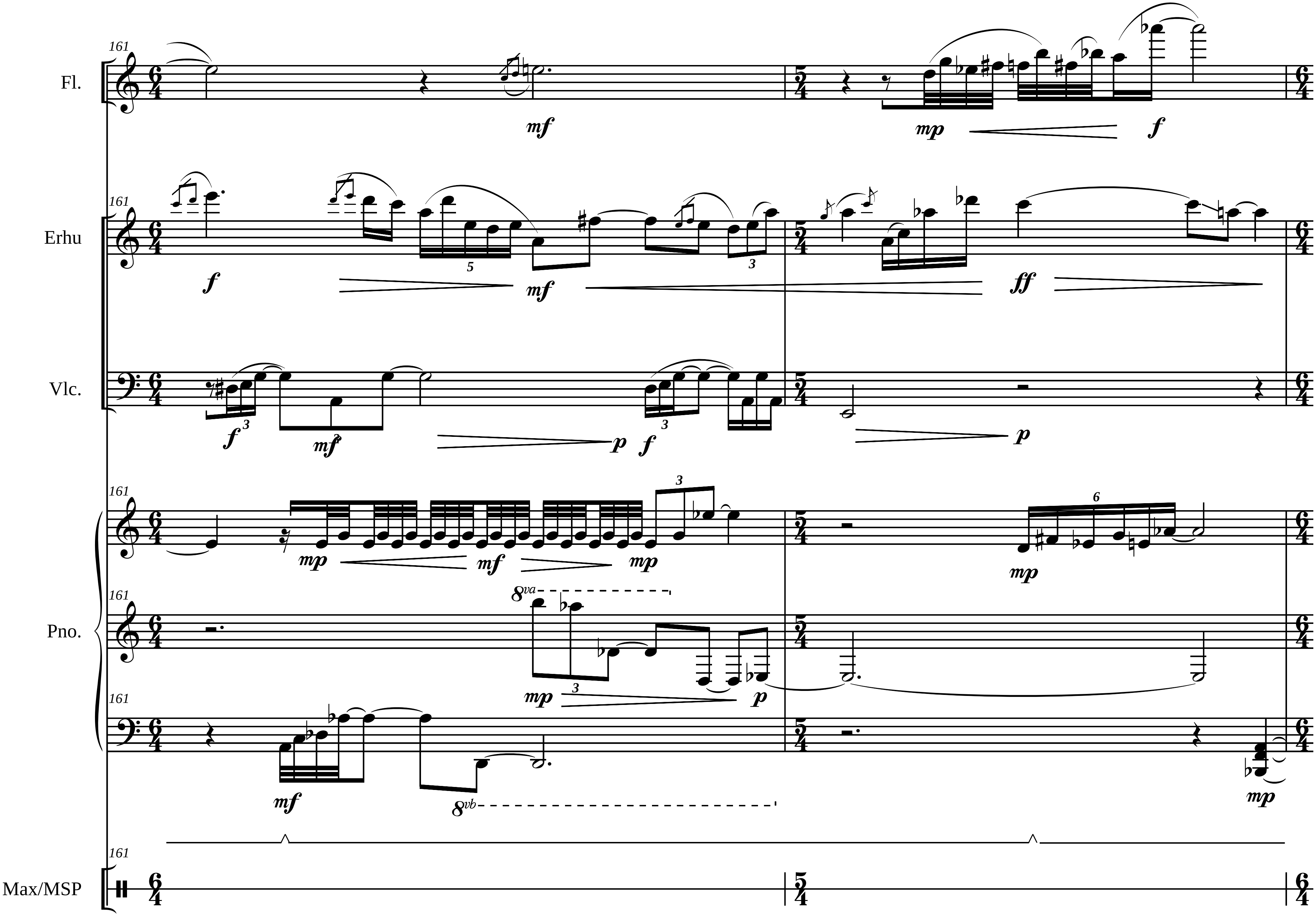

148
Fl.
mf
f
148
Erhu
slow gliss.
slow gliss.
slow gliss.
p
f
Vlc.
fp
3
3
mp
f
mp
148
6
Pno.
mp
mf
mp
p
f
148
3
5
p
f
Ped.
148
Max/MSP

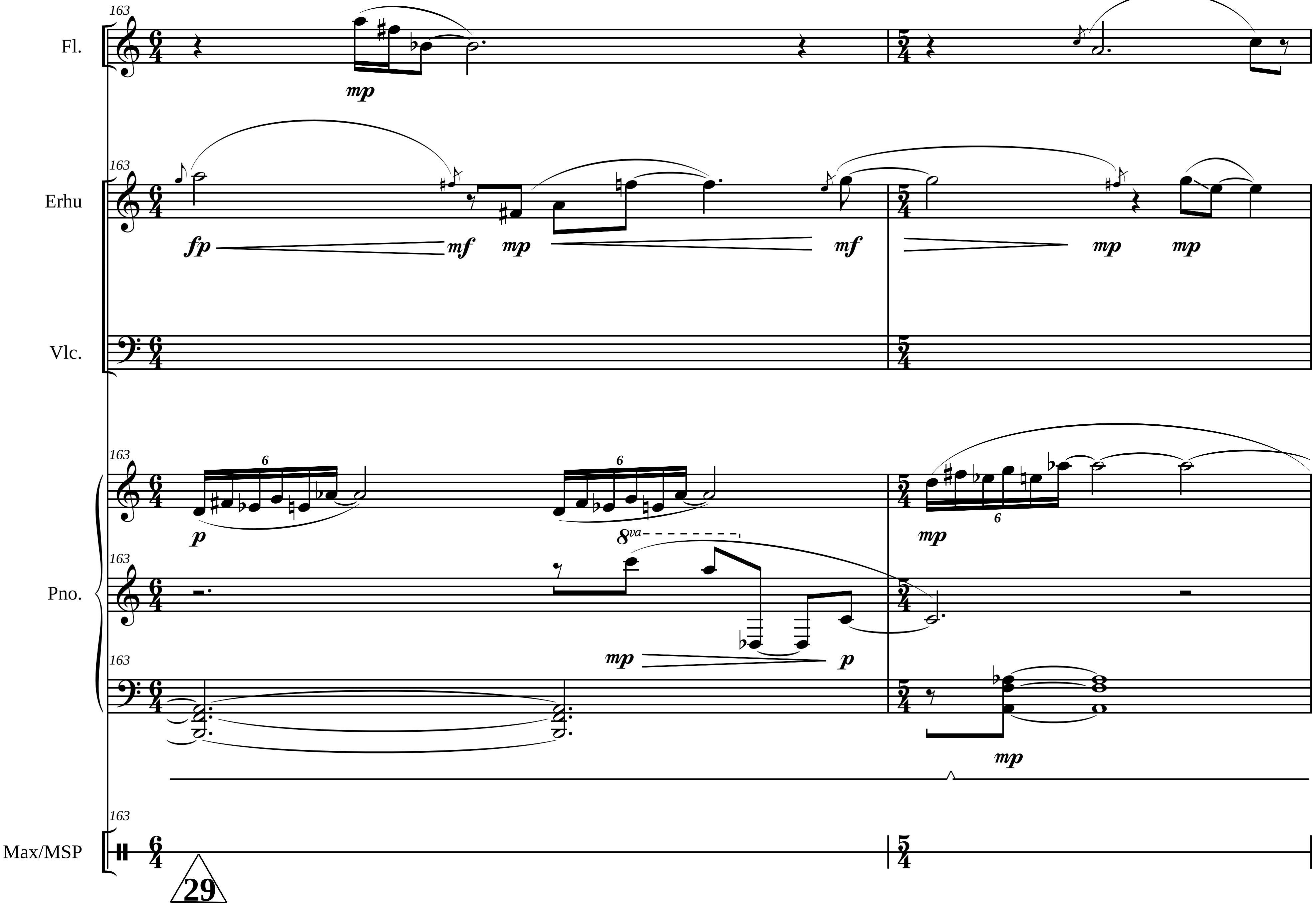

150
Fl.
Erhu
pressed vibrato
pizz.
arco
Vlc.
Pno.
Max/MSP
p
f
pp
fp
mp
mf
3

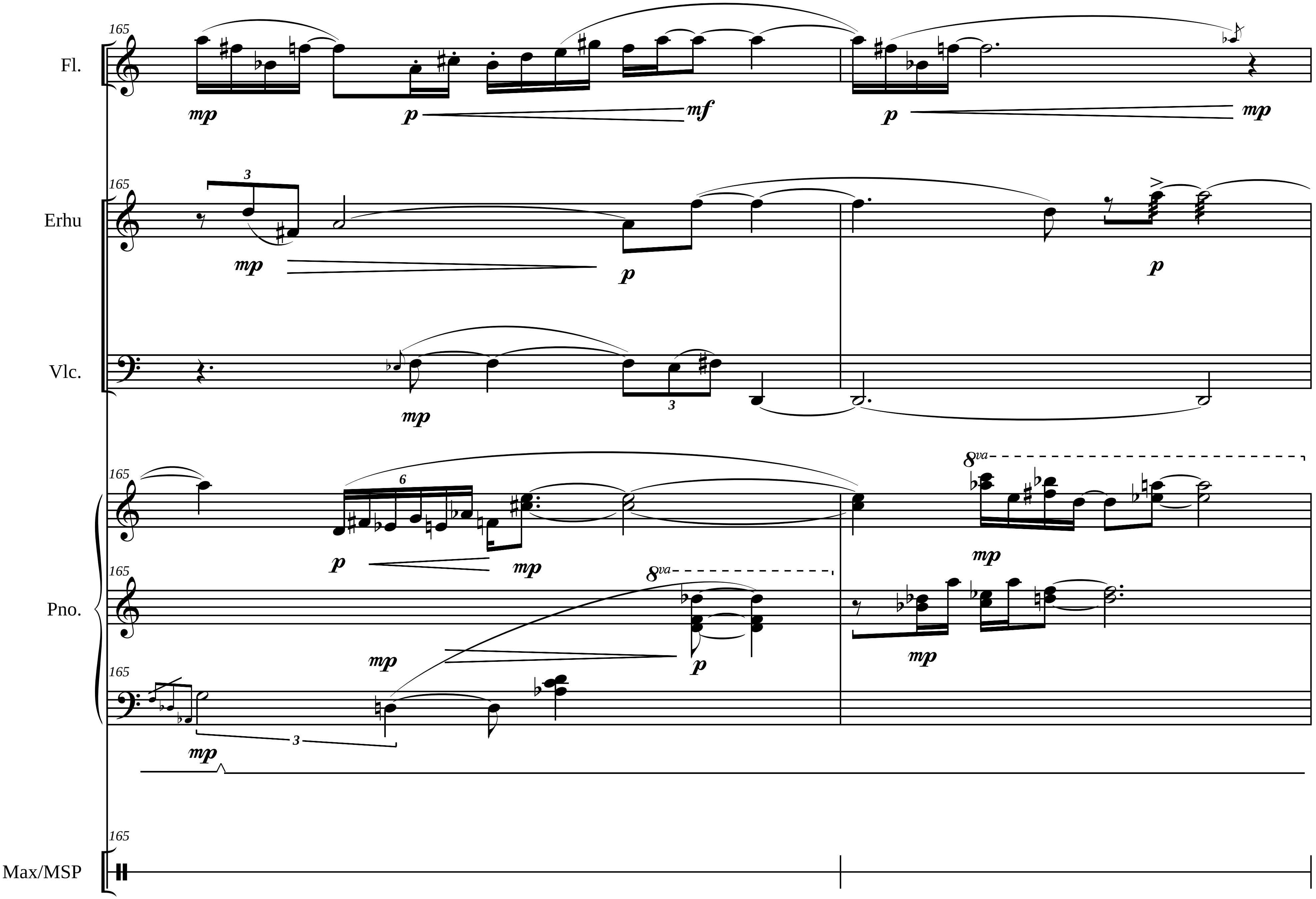

155
Fl.
Erhu
Vlc.
Pno.
Max/MSP
f
f
mf
fp
mf
p
fp
mf
mp
mf
mp
p
mf
p
f
sffz
27

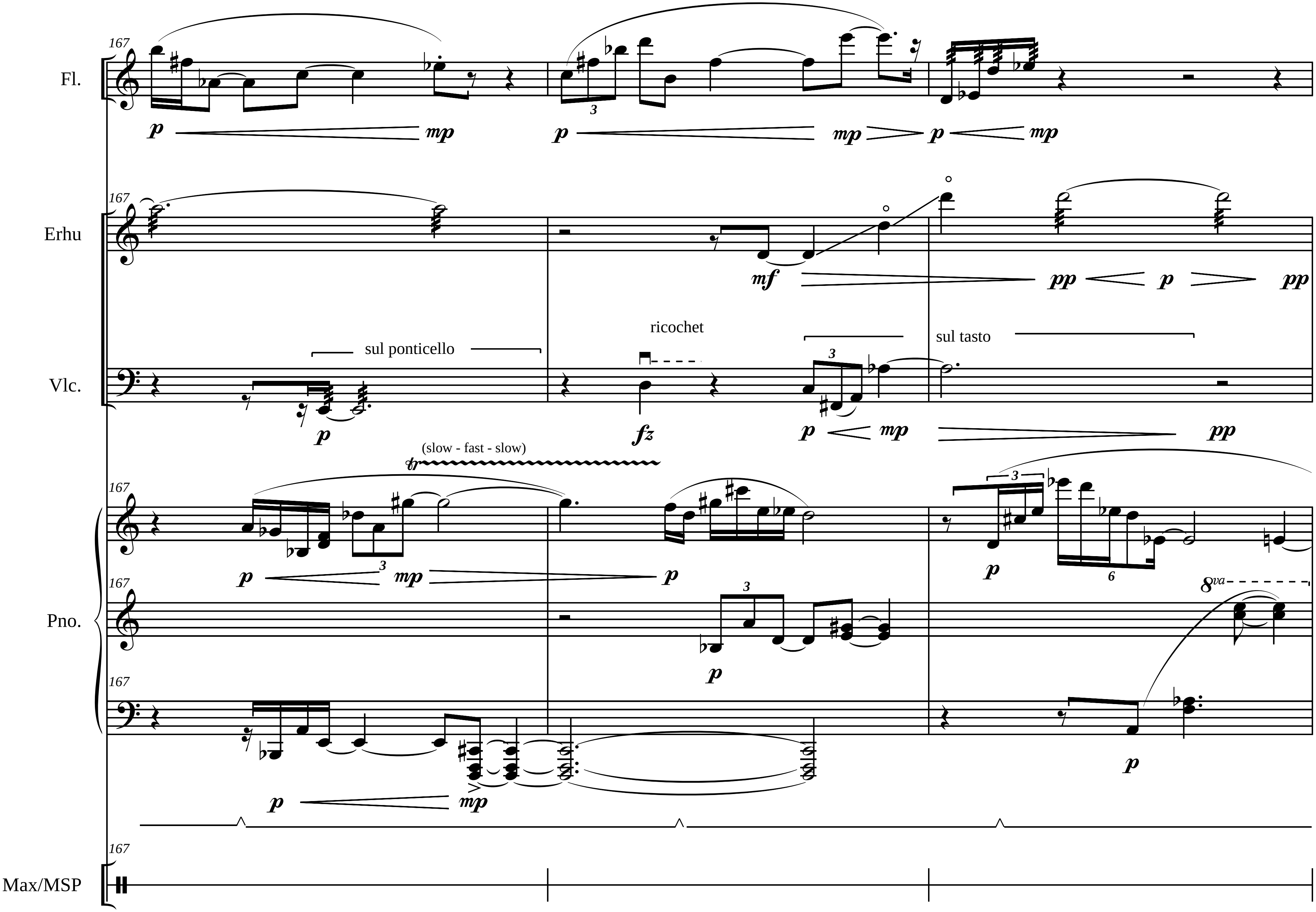

157
Fl.
sffz
(♩ = ♩.)
Erhu
mf
cresc.
pizz.
Vlc.
sffz
Pno.
f
f
Ped.
Max/MSP
28

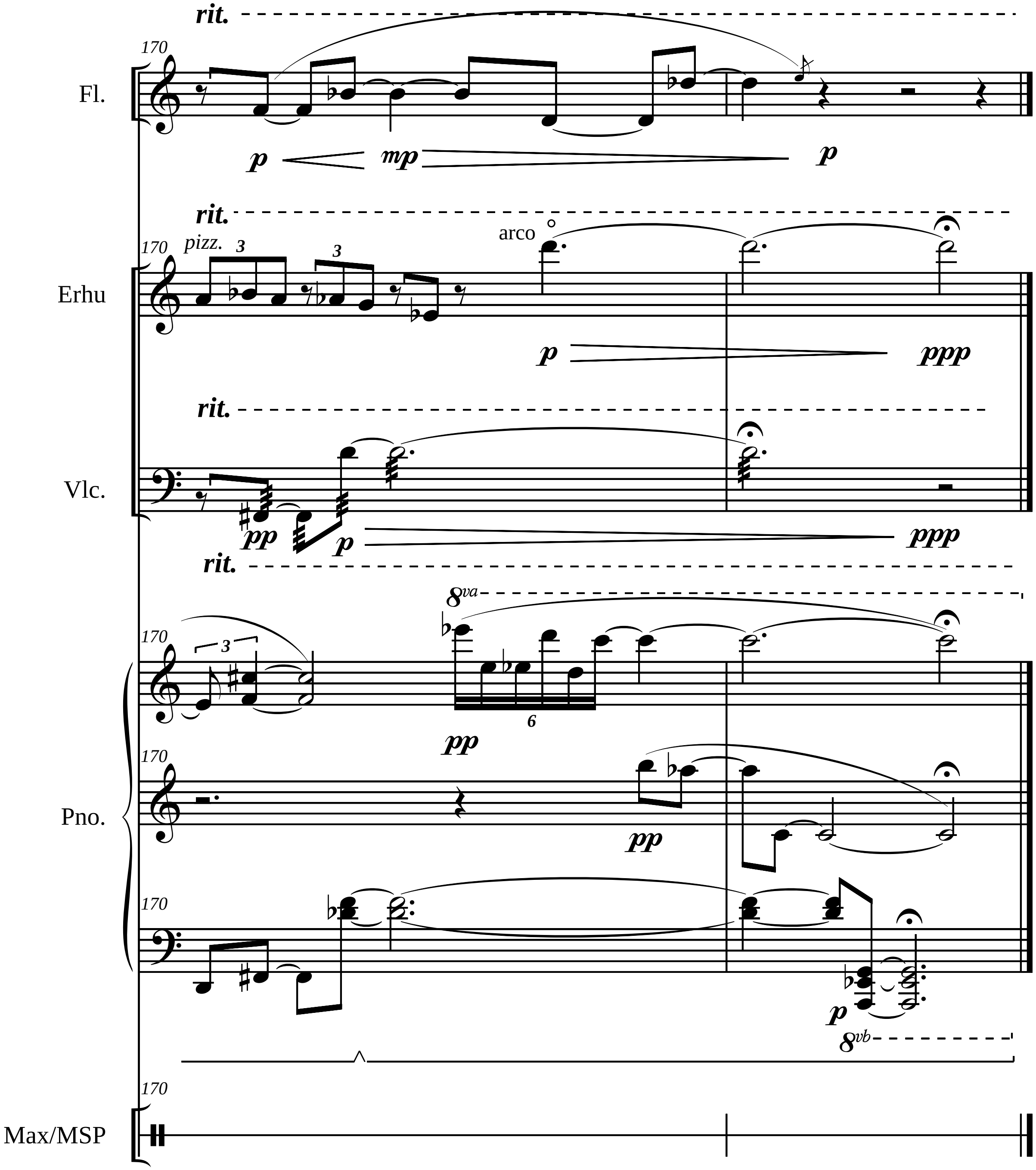

159
Fl.
Erhu
Vlc.
Pno.
Max/MSP
arco
L.H. (through the end of the piece)
mf
f
ff
p
mp
3
5
6


161
Fl.
Erhu
Vlc.
Pno.
Max/MSP
mf
mp
f
ff
p
8va
8vb
3
5
6


163
Fl.
Erhu
Vlc.
Pno.
Max/MSP
mp
fp
mf
p
8va
6
29

165
Fl.
Erhu
Vlc.
Pno.
Max/MSP
mp
p
mf
3
6
8va


167
Fl.
p
mp
p
mp
p
mp
3
Erhu
mf
pp
p
pp
sul ponticello
ricochet
sul tasto
Vlc.
p
fz
p
mp
pp
3
(slow - fast - slow)
tr
Pno.
p
mp
p
p
3
3
6
8va
p
p
mp
Max/MSP


rit.
170
Fl.
p
mp
p
rit.
170
pizz.
3
3
arco
Erhu
p
ppp
rit.
Vlc.
pp
p
ppp
rit.
8va
170
6
pp
170
Pno.
pp
170
p
8vb
170
Max/MSP