CLEM, D. TRAVIS, M.M. Interval Serialization and Its Use in *Exploring the Third Major Nebula*. (2009)
Directed by Dr. Mark Engebretson. 46 pp.

Interval serialization is a musical process I designed in order to generate a large amount of music based on a relatively small amount of initial material. The process begins with a small collection of pitches. While maintaining interval size and order, the direction of each interval is altered until every permutation of directions is exhausted. Once a composer has generated material through this process, a variety of methods can be used to combine the resulting cells to make a composition.

The thesis includes a written description of the technique of interval serialization, my original composition, *Exploring the Third Major Nebula*, for cello and electronics, and Pure Data computer files necessary for performing the piece.

The written portion of this thesis describes the process of interval serialization in detail, likening it to the systematic process of counting. Examples are given that demonstrate how cells may be formed, permutated, combined and deployed in a musical setting.

The paper further describes aspects of the piece, how interval serialization is used in the composition, and how the resulting material is combined in a variety of ways. *Exploring the Third Major Nebula* is based on four different tunings of the major third. This concept inherently gives the work certain pragmatic challenges, but these challenges are solved and described in the paper.

INTERVAL SERIALIZATION AND ITS USE IN

EXPLORING THE THIRD MAJOR NEBULA

by

D. Travis Clem

A Thesis Submitted to
the Faculty of The Graduate School at
The University of North Carolina at Greensboro
in Partial Fulfillment
of the Requirements for the Degree
Master of Music

Greensboro
2009

Approved by

____

Committee Chair

APPROVAL PAGE

This thesis has been approved by the following committee of the Faculty of The Graduate School at The University of North Carolina at Greensboro.

Committee Chair ______________________________

Committee Members ______________________________

____

____

Date of Acceptance by Committee

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF EXAMPLES

CHAPTER I

INTRODUCTION

I have created a process I call interval serialization. This thesis is an attempt to explain that process so other composers will be able to use it as a compositional tool if they so desire.

Chapter II presents a description of how to generate musical material from a cell of pitches by systematically changing the directions of the intervals between the notes until every permutation of directions is exhausted. I describe how it is analogous to counting, and through this analogy, explain the process. I also describe several ways that individual cells can be combined to form a composition.

Chapter III presents an explanation of how I use interval serialization in my composition *Exploring the Third Major Nebula*. I lead into the composition itself by describing how I used the process in two previous works. Though I do not describe every facet of *Nebula*, I present its compositional goals, its concept, and how I overcome certain pragmatic challenges created by the concept.

By virtue of its name, an observer might want to compare interval serialization with other types of serial processes (the twelve-tone method or integral serialism of various types). Schoenberg's twelve-tone system is a process by which a composer could organize all the pitches in a systematic way. Integral serialism applied the same systematic process to even more characteristics of music such as articulation, dynamics rhythm, and so on. Notes very often are repeated in interval serialization and the process does not inherently address articulation and dynamics. Based on these characteristics, one might wonder why the idea of serialism is invoked at all.

What this process does is systematically manipulate the direction of intervals within a given cell. Since it finds every possible permutation of interval directions, since it does so in a methodical manner, and since it assigns the permutations an ordered number, it qualifies as a form of serialism. Schoenberg's twelve-tone method used a systematic process by which to control the pitch. Interval serialization is a systematic process as well, and though it does not specifically address pitch, it can be used as a tool for music making.

Obviously, this process was not created in a vacuum. Though interval serialization differs from the twelve-tone method in many distinct ways, it owes much to the compositional techniques of Arnold Schoenberg and Anton Webern. Alan Forte's set theory analyzes complicated post-tonal sonorities by manipulating the pitches into consistent patterns; his vector description (methodically describing the interval content) of pitch sets is an important precursor to interval serialism.[1] And contour theory, as described by Ellie Hisama, is an important forerunner to this method.[2]

---

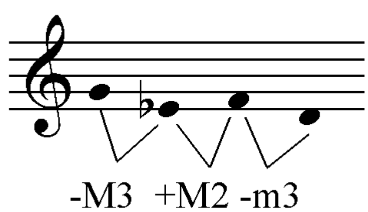
[1] Stefan Kostka, *Materials and Techniques of Twentieth-Century Music* 2nd ed. (Upper Saddle, New Jersey: Prentice Hall, 1999),179-182. Also see David Cope, *Techniques of the Contemporary Composer* (United States: Schirmer Thomson Learning, 1997), 77-83.
[2] Ellie Hisama, *Gendering Musical Modernism* (New York: Cambridge University Press, 2001), 49-51.

CHAPTER II

GENERATING MATERIAL

This chapter will focus on how one generates material using the process of interval serialization. Towards the end, different methods of combining resultant cells will be described.

One starts with an abstract collection of pitches hereafter called a cell (void of rhythm, articulation, dynamics, etc). The order is important and should be maintained. Then one analyzes the intervals and their direction between the notes. For this chapter, the pitch collection from the opening of Beethoven's Symphony No. 5, Movement 1 will serve as an example:



Example 1: First four pitches from Beethoven's Symphony No. 5, Mvmt I

"Interval serialization" refers to the notion that direction of interval will be treated in an orderly and systematic way while maintaining the interval size (term "interval" is applied generally, not specifically). Two types of interval serialization exist: bi-directional and tri-directional. In bi-directional interval serialization, the intervals either ascend or descend. In tri-directional interval serialization, the intervals can ascend, descend, or remain stationary (neither ascend nor descend).

Interval serialization is a systematic process that pairs interval directions. In many ways, it is analogous to simple counting. Each place value of counting has a limited number of symbols that fill that place. Our everyday system of counting has ten symbols (0, 1, 2...). The number of symbols in the system (base ten, base eight, base sixteen, etc.) and their order (1, 2, 3, etc.) are agreed upon, so most people take them for granted. When the symbols within a place value reach their limit, the order begins again and the place value to the left advances to its next symbol. For example, in base 10 (our everyday counting system), <19> advances to <20>; the one's place cycles back to the agreed upon symbol (0) and the ten’s place advances to its next symbol (2). In a base two system, two symbols are used (0 and 1), and their agreed upon order is 0, then 1. Three digits in a base two system produce eight individual numbers:

Table 1: The first eight numbers in a Base 10 and a Base 2 counting system

Base 10	Base 2
1	000
2	001
3	010
4	011
5	100
6	101
7	110
8	111

Interval serialization is like counting, except instead of base two, we see a bi-directional system and instead of place values, we see intervals. Instead of 0 and 1 being the agreed upon symbols in the agreed upon order, we have ascending and descending intervals. Therefore, the previous eight numbers in base two system translate into the following eight interval permutations (perm.) in a bi-directional system:

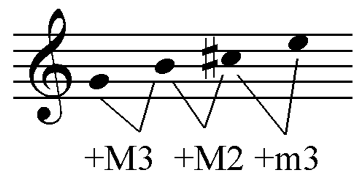
Table 2: Translation of a Base 2 system to bi-directional serialization

Base 10	Base 2	Interval Serialization's Translation
1	000	ascending, ascending, ascending
2	001	ascending, ascending, descending
3	010	ascending, descending, ascending
4	011	ascending, descending, descending
5	100	descending, ascending, ascending
6	101	descending, ascending, descending
7	110	descending, descending, ascending
8	111	descending, descending, descending

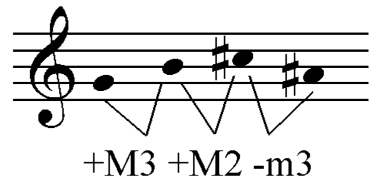
This is not to say that 0 necessarily needs to translate into ascending and 1 into descending. The point is that the symbols involved have a consistent order (in this case ascending then descending), and when that order runs its gamut, it starts over and the place value (or interval) to the left advances its symbol.

To apply this process to actual music, take the cell of notes from earlier (the reduced beginning of Beethoven's Fifth Symphony) and serialize its intervals. If the agreed upon symbols are ascending and descending intervals (in that order), the permutations produced by the process are as follows:

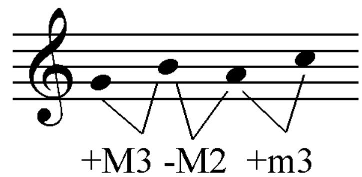
Perm. 1 - ascending, ascending, ascending



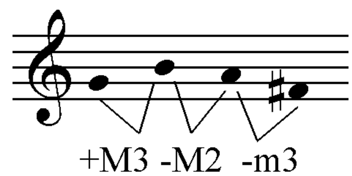
Perm. 2 - ascending, ascending, descending



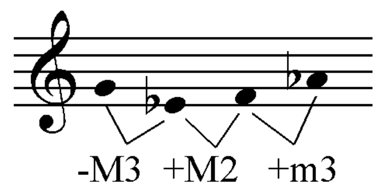
Perm. 3 - ascending, descending, ascending



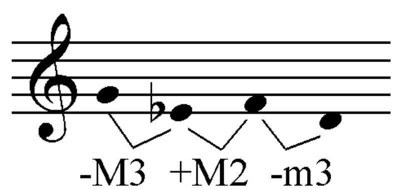
Perm. 4 - ascending, descending, descending



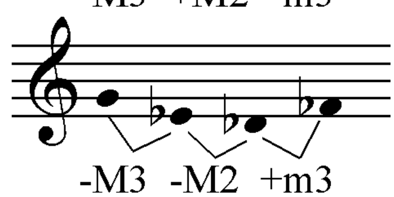
Perm. 5 - descending, ascending, ascending



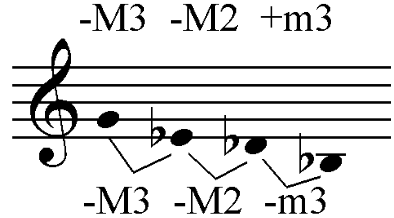
Perm. 6 - descending, ascending, descending



Perm. 7 - descending, descending, ascending



Perm. 8 - descending, descending, descending



Example 2: Permutations 1-8 in bi-directional interval serialization

Example 2 presented bi-directional interval serialization. Tri-directional works in the same way only another direction is introduced into a single interval's options. To return to the counting analogy, another symbol must be used before a digit returns to its starting point and the digit to its left advances. In a base three system, three digits produce the following numbers:

Table 3: The first twenty-seven numbers in a Base 10 and a Base 3 counting system

Base 10	Base 3
1	000
2	001
3	002
4	010
5	011
6	012
7	020
8	021
9	022
10	100
11	101
12	102
13	110
14	111
15	112
16	120
17	121
18	122
19	200
20	201
21	202
22	210
23	211
24	212
25	220
26	221
27	222

When the one's digit reaches the last symbol in its series, it returns to its first symbol and the digit to its left progresses. This process, in essence, is interval serialization.

For tri-directional serialization, the directions (symbols) a single interval (place value) can go are ascending, stationary, and descending (the possible symbols and their agreed upon order). Therefore, the previous three digit numbers in a base three system would translate to the following twenty-seven interval permutations:

Table 4: Translation of a Base 3 system to tri-directional serialization

Base 10	Base 3	Interval Serialization's Translation
1	000	ascending, ascending, ascending
2	001	ascending, ascending, stationary
3	002	ascending, ascending, descending
4	010	ascending, stationary, ascending
5	011	ascending, stationary, stationary
6	012	ascending, stationary, descending
7	020	ascending, descending, ascending
8	021	ascending, descending, stationary
9	022	ascending, descending, descending
10	100	stationary, ascending, ascending
11	101	stationary, ascending, stationary
12	102	stationary, ascending, descending
13	110	stationary, stationary, ascending
14	111	stationary, stationary, stationary
15	112	stationary, stationary, descending
16	120	stationary, descending, ascending
17	121	stationary, descending, stationary
18	122	stationary, descending, descending
19	200	descending, ascending, ascending
20	201	descending, ascending, stationary
21	202	descending, ascending, descending
22	210	descending, stationary, ascending

23	211	descending, stationary, stationary
24	212	descending, stationary, descending
25	220	descending, descending, ascending
26	221	descending, descending, stationary
27	222	descending, descending, descending

Every time an interval (place value) returns to ascending, the interval (place) to its left moves to its next direction. The musical translations of the previous twenty-seven permutations are not included in this paper for reasons of space but would contain twenty-seven different permutations of Beethoven’s cell.

The number of resulting permutations is a function of how many intervals the original cell has. Our original cell (Beethoven’s Fifth) has three intervals. Bi-directional has two directions for any given interval; tri-directional has three. The number of resulting permutations can be calculated as the number of directions in the system raised to the exponential power of how many intervals the cell has. Our original cell in a bi-directional system produces eight permutations (two to the third power); our original cell in a tri-directional system produces twenty-seven permutations (three to the third power). In this way, one can predict how many permutations interval serialization will produce before one begins.

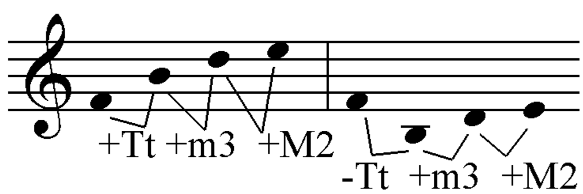
Figure 1: Mathematical equation for predicting resulting permutations

$$\langle \text{Directions in the system} \rangle^{\langle \text{Intervals in the cell} \rangle} = \langle \text{Resulting permutations} \rangle$$

$$2^3 = 8 \qquad 3^3 = 27$$

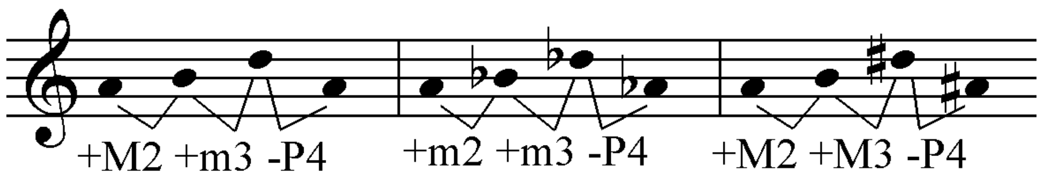
The original cell determines the number and variety of resulting permutations. To keep the number of permutations manageable, it is wise to limit the number of intervals in the original cell. For example, a cell with four intervals would produce sixteen permutations in a bi-directional system and eighty-one permutations in a tri-directional system; a five-interval cell would produce thirty-two and two hundred forty-three permutations in the same respective systems.

To maximize pitch class variety, avoid tritones in the original cell (although this may suit certain compositional purposes). Example 3 demonstrates how a tritone produces the same pitch class (and successive pitch classes) whether in ascending or descending form.



Example 3: Permutation 1 and 5 of a possible serialization illustrating tritone use

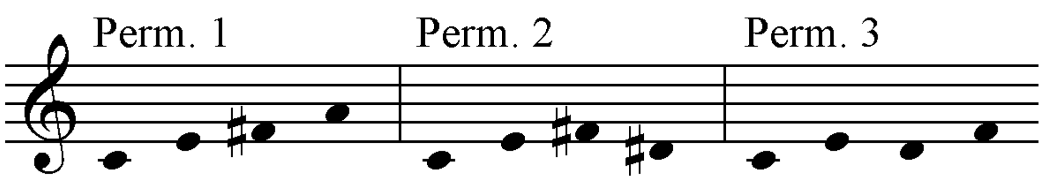
One should also be aware of smaller intervals that combine to make a larger interval within the cell. In Example 4, a major second and a minor third are followed by a perfect fourth; a future permutation will have the first two ascend and the last one descend, landing on the same pitch class one began with. The last two measures of the example would create higher pitch class variety, since the ascending-ascending-descending permutation will produce a different ending pitch than the starting pitch.



Example 4: Smaller intervals combining to a larger interval and two alternatives

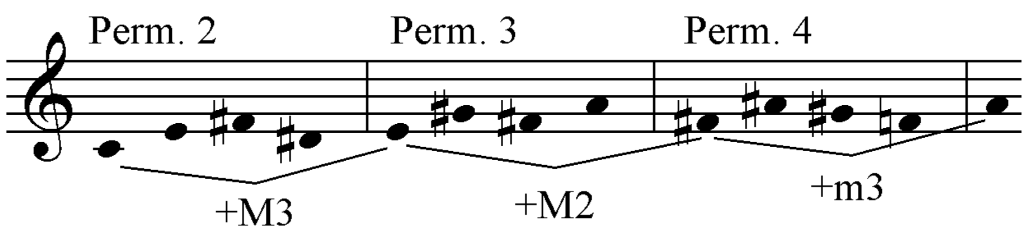
Once one cell has become many, the cells need to be put together to make a composition. Interval serialization generates musical building blocks; how those blocks are put together affects the final composition as much as the original cell's construction. The following is an introductory list of ways the individual cells can be combined.

The cells may consistently start on the same pitch.

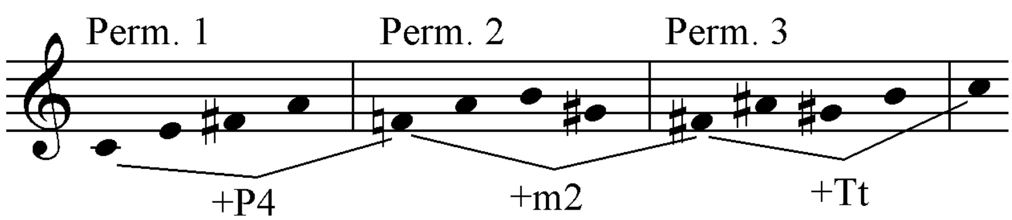


Example 5: Restarting cells on the same pitch

The cells may start on different pitches. My own preferences towards musical organicism dictate that there should be a purposeful relationship between the starting pitches of the cells. This relationship could also be controlled with interval serialization (though as yet I have not attempted this method in a composition).

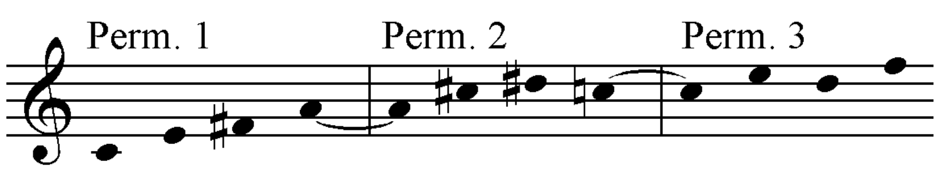


Example 6: Restarting cells on different pitches related by Permutation 1



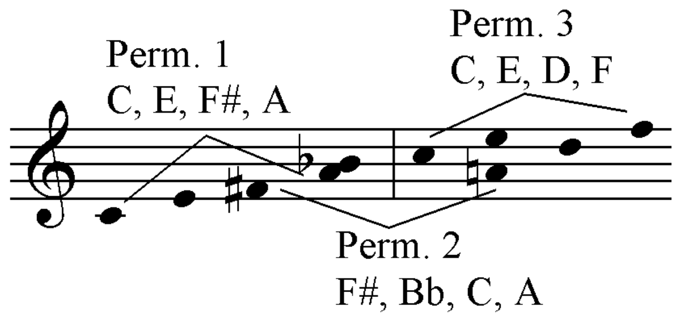
Example 7: Restarting cells on different pitches related by intervals not in the original cells

The cells may elide into each other (the first note of the new cell is the last note of the old cell). This method has been my consistent preference and can be seen in practice in the cello melody of *Nebula*. To me, it shows the process's focus on intervals over the specific notes on the page.



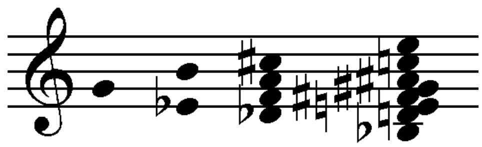
Example 8: Eliding cells together

The cells may overlap each other. The third (or second, or fourth, etc.) note of each cell can be the first note of the next cell.



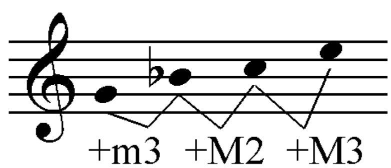
Example 9: Overlapping cells on the third note

All the cells could occur at the same time, but this would make for a rather short composition.



Example 10: All permutations starting at once on the same pitch

The order of the intervals within a cell can be reversed, in essence presenting the permutation in retrograde.



Example 11: Permutation 1 in retrograde

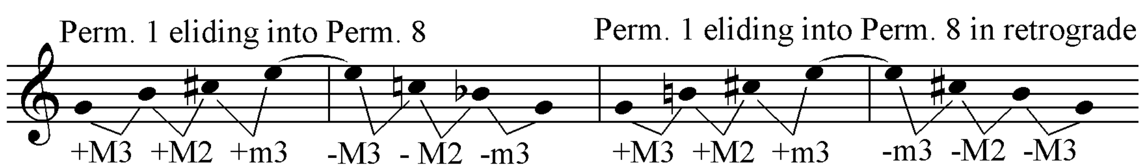
A cell does not need to be inverted because another resultant permutation already is that cell's inversion. Considering this aspect, there is symmetry to interval serialization. Halfway through the process, the resultant permutations become inversions of a previous permutation (Perm. 5 is an inversion of Perm. 4; Perm. 6 is an inversion of Perm. 3, etc). A cell's inversion can be calculated using the following equation:

Figure 2: Mathematical equation for finding a permutation's inversion

(<Total resultant permutations> + 1) - <A permutation>
= <That permutation's inversion>
(8 + 1) – 5 = 4   (8 + 1) – 6 = 3

The same trend and equation holds true in a tri-directional system.

When combining a cell with its inversion using elision, a greater pitch class variety can be achieved by keeping a consistent cell presentation for both permutations (either both cells in retrograde or neither). Altering the second cell's presentation will cause it to backtrack onto the same pitches as its inversion. Example 12 demonstrates this occurrence.



Example 12: A cell and its inversion elided together with consistent and inconsistent presentation

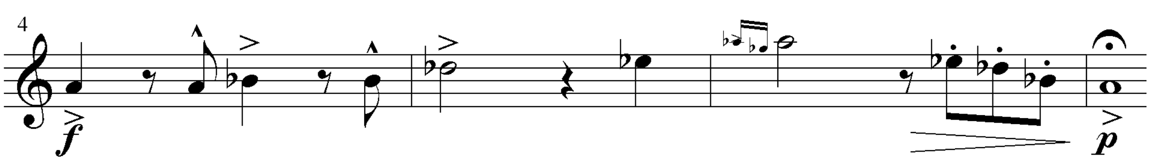
To compose using interval serialization, one must consider the aforementioned cursory list of ways to adjoin cells. One must also consider the order in which the cells are presented (does the composer maintain the order the system produced or purposefully rearrange it). The composer must decide whether or not to use all of the resultant permutations. Does one compose a purposeful relationship between a cell and its inversion in the overall composition? Quite separately, does one present cells in retrograde and if so, which one(s)? Though these choices appear daunting, they are all considerations that can be made (or ignored) when composing using interval serialization.

CHAPTER III

THE ARCHITECTURE BEHIND *NEBULA*

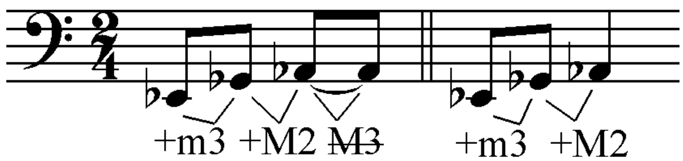
The subject of this chapter is *Exploring the Third Major Nebula* (2009) and explaining certain details of it. *Nebula* is the culmination of what I have taught myself concerning interval serialization. It is also extends many of my insights to proportions I had not yet pursued.

The first piece I composed using interval serialization was a work for solo clarinet called *Grown from Seed* (2005). The starting set of notes came from the Sigma Alpha Iota chorale (with many notes omitted) and I serialized the intervals in a bi-directional way. In this piece, individual cells are elided into each other and I ordered the cells in such a way as to reflect the symmetrical properties of the process in the formal layout of the piece (for every ascending interval of the process, a descending form appears somewhere else; the process causes the melody to start and end on the same note). The rhythms, dynamics, and articulations for the piece were all freely chosen.



Example 13: m. 4-7 of *Grown from Seed*, demonstrating freely composed rhythm, articulation, and dynamics

The next composition was for solo piano titled *A Lecture on Sharp Things* (2007). It employed tri-directional serialization and differed from the previous compositions in a number of marked ways. The interval serialization process determined the rhythm of the melody. The smallest rhythmic duration of the composition is an eighth note and any longer duration is a product of tied together stationary intervals.

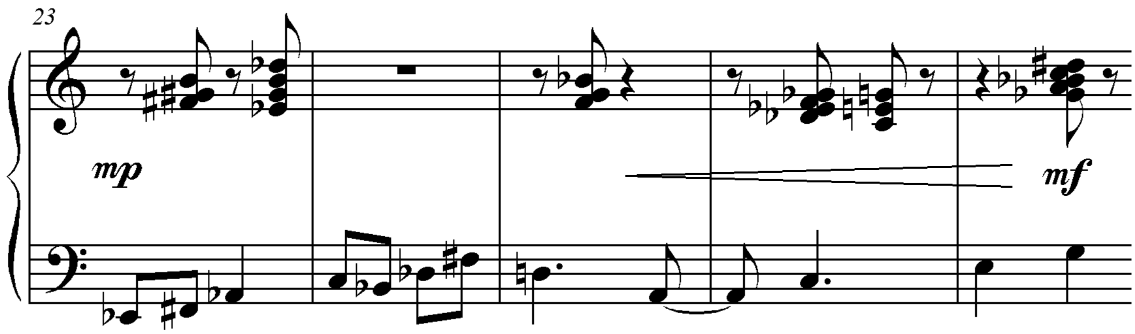


Example 14: Demonstration of rhythm derived from a tri-directional approach

Also for this composition, I wanted to take advantage of the fact that the piano can play chords. I took each cell from the process (at least each cell that had more than one note) and compressed its horizontal melody into a vertical chord. I then analyzed each chord (using the vectors from Forte analysis) to find its consonance/dissonance level.[3] I used this measure of consonance/dissonance to organize the chords' placement in the composition; I placed chords with greater consonance after chords with greater dissonance, imitating the dominant seven (dissonant) to tonic (consonant) movement of tonality. Though altering the chords' octave placement on the piano, I kept the pitch class content the same as the cells' melodic version. The melody's cells elided into each other but instead of ordering them in such a way as to reflect symmetry, I chose instead to order them asymmetrically. The composition still starts and ends on the same pitch class,

[3] Cope 78.

however a cell two measures from the beginning will not find its inversion two measures from the end.



Example 15: m. 23-27 of *A Lecture on Sharp Things* demonstrating use of chords and rhythms derived from tri-directional serialization

As I began to compose *Nebula*, I had a few ideas that I wanted to guide me. I wanted high electronic sounds and a low acoustic instrument. I wanted the electronics to be a little ambiguous in their function and the bass instrument to define them, give them acoustical perspective. If possible, I wanted to juxtapose a bi-directional approach to interval serialization with a tri-directional approach. And, in possibly my most non-traditional idea for the piece, I wanted the serialization process to organize four different tunings of a major third.

In previous compositions, I noticed the process had the potential for a lot of backtracking, or creating a melody that doubles back over pitch classes it had already covered (refer to Example 4 in Chapter II). This characteristic is inherent to the process, and depending upon your compositional aim, this can either be avoided (as described in the previous chapter) or embraced. But at some point, I entertained the idea that the process could be used to organize intervals besides tempered ones (I had been doing a lot of studying of Harry Partch's justly intoned scale).[4] If a cell had a just minor third and just major second that did not combine to exactly the same size as the next perfect fourth, then the ascending-ascending-descending permutation would end that cell on a different frequency than the one on which it started.

I found this sonic possibility very exciting and at the same time pragmatically challenging. Would I build my own instrument(s) like Partch? No, because the cells could be combined in such a large variety of ways, and would possibly be able to go to so many different individual frequencies, that the instrument would be huge, completely daunting, and virtually unplayable. To shrink down the number of possibilities would render the instrument applicable to only one composition, which is simply impractical. My other two solutions were to use electronics as an instrument (speaker systems can play most frequencies in the audible spectrum) or to use existing instruments that are not absolutely bound to the tempered scale (string instruments without frets, brass instruments with slides, woodwind instruments with a variety of alternate fingerings, the human voice, etc.). For *Nebula,* I decided to use both options.

The four tunings of the M3 in *Nebula* are just intonation, Pythagorean intonation, equal temperament, and a ratio derived from the golden mean. Just intonation was arrived at from the overtone series. Pythagoras's tuning is a product of four ascending perfect fifths reduced by two octaves. An equal tempered M3 comes from logarithmically dividing an octave into twelve parts and using the first four divisions. Since the golden

---

[4] Richard Kassel, *Barstow as History: An Introduction to the Sound World of Harry Partch* (Madison,Wisconsin: A-R Editions, 2000), xvii.

mean is a ratio, it can be used as a musical interval; James Tenney used it as such in his electronic composition *For Ann (rising)*.[5] The golden mean comes out to a sharp version of the flat sixth, but when the interval is inverted (when one note of the interval is transposed an octave), the interval comes out to roughly a major third.

These tunings are not of equal size. The table below presents the tunings (from largest to smallest). The table also presents the mathematical formula for each tuning, what the frequency would be if each tuning ascended from A 440, and how many tempered cents each tuning differs from a tempered M3.

Table 5: Four tunings of a M3, their mathematical formulas, their frequency above A 440 and their difference from the Tempered M3 in cents

	Tuning Math	Frequency from A 440	Cents Off Tempered
Pythagorean	81/64	556.88	7.82
Tempered	$2^{(1/3)}$	554.37	0
Just	5/4	550	-13.69
Golden Mean	$5^{(1/2)}-1$	543.87	-33.09

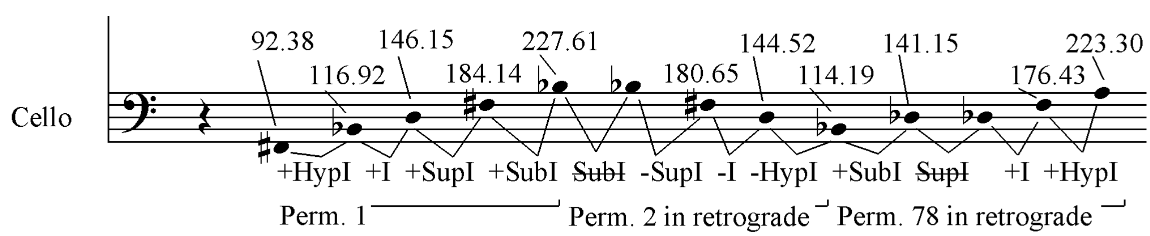
For the purposes of this thesis, the largest tuning (Pythagorean) will be referred to as an Hyper Integral Third, the next largest tuning (Equal-Tempered) will be referred to as a Super Integral Third, the next smallest tuning (Just) will be referred to as a Integral Third, and the smallest tuning (Golden Mean) will be referred to as a Sub-Integral Third.

With these general details in mind, I began to flesh out specific details of the composition. I finalized an order for the thirds (Hyper Integral, Integral, Super Integral,

[5] James Tenney, *Selected Works 1961-1969*, compact disc liner notes, (New World Records, 2003), 29.

Sub-Integral) within the cell. I made a cello the low, acoustic instrument and generated his/her melodies using a tri-directional approach. I made the electronics only vertical sonorities and (even though the electronics share the same original cell as the melody) its cells should be generated through a bi-directional approach. I decided that the individual cells of the melody should elide (refer to Example 8 in Chapter II), but that the electronic frequencies should be determined by another means (more on how I arrived at those frequencies later).

The cello's melody became finalized, as well. An order to the permutations was decided upon that uses every permutation once in the entire composition. The order also subtly emphasizes that tri-directional serialization produces every permutation of interval directions that bi-directional serialization does. During every bi-directionally-derived electronic sonority, the cello presents its corresponding tri-directional permutation in melodic form. Every cello permutation that does not correspond to an electronic sonority is presented in retrograde. Example 16 presents the first three permutations of the cellist's melody and each note's frequency in Hz.



Example 16: The first thirteen notes of the cello melody and their frequencies

With these decisions came certain challenges to the composition's pragmatism. If the thirds were of different sizes, then they had the probability of not perfectly backtracking in the melody as described above. The cello could start on an A2 (110 Hz) and by the fifth or sixth note, his/her melody could have backtracked to a 110.5 Hz or an even more specific 109.842 Hz. The great variety of pitches produced by this tuning scheme result in serious notational difficulties with respect to pitch.

My solution to this difficulty was to take advantage of the cellist's preexisting musical skills. The cellist wears a set of headphones. An electronic, pure tone (what I call an aural cue) is played into the headphones and the cellist matches the note they play to that pitch. The score altogether avoids the issue of how specifically to notate the melody's pitches and instead presents the closest note in the tempered scale to what the cellist is hearing. The performance instructions describe the purpose and accuracy of the written music and how they should reconcile what they hear in the headphones with what they see on the page.

With this solution comes another question of rhythm and pacing. How much time does a performer need to find the pitches? Each performer is different; maybe some need more time. If the aural cue plays back from a CD, it may advance before the cellist has found their pitch. Or conversely, it may advance too slowly, in which case the part (and composition as a whole) become too long, slow, and boring.

This problem was solved within a music making computer program. Pure Data is a free, open source version of Max/MSP and capable of powerful sound manipulations. Within the program, there is a list of frequencies that will correspond to the cello melody. A footpad transmits MIDI information to a counting feature within the program. The cellist depresses the footpad when (s)he is ready to advance the aural cue to his/her next note. Each time the footpad is depressed, the counter advances the list to its next frequency, which plays into the cellist's headphones.

Since the program produces the electronic sounds for the cello, it produces the high electronic part that makes the other half of the composition also. In essence, the program has a bank of oscillators with a list of what frequencies will sound throughout the composition. When the footpad/counter reaches certain values, half of the oscillators are given their frequency, activate and fade in. When the footpad/counter reaches other values, the remaining oscillators are given their frequencies and fade up while the first batch simultaneously fade out. This way, the high electronics have a fairly seamless flow between different sonorities produced by interval serialization.

This solution to the rhythm/pacing problem places the cellist squarely in charge. If at any point the composition disagrees with the cellist's musical sense, they can progress through it a little more quickly. While in another section, if his/her instincts respond favorably to the composition, they can prolong sonorities, too. My initial conception of the piece placed it in the realm of drone composition, and though for certain passages I want to make heightened drama and even have a climax, it is perfectly acceptable to have the cellist control the pace.

As stated before, the individual cells of the melody elide into each other; this method generates the individual frequencies. The high electronic frequencies were arrived at through a different process. I wanted to try a method of connecting cells other than elision, which heretofore had been my only approach. I also knew I did not want to restart each cell from the same frequency every time. Instead, I decided to place the cell where the two closest notes of the cell would have a beating interaction. In live, acoustic instruments, beating is often undesirable, a sign of two players being out of tune. However, in electronic music, I have found beating to make for a more complex, enjoyable sound.

Instead of having the last note of the cell elide into the first note of the next cell, or having the highest note elide into the next cell's lowest note, the higher note in the pair that have a beating interaction elides into the lower note of the next cell's pair that have a beating interaction. In Table 6, the closest frequencies that have a beating interaction in the second electronic sonority are 1478.082 Hz and 1496.558 Hz (they beat at 18.476 beats per second). In the third electronic sonority, I calculated which notes of the cell would have the closest frequencies and made the bottom frequency of that pair 1496.558 Hz (the higher frequency of the previous chord's beating pair). The other frequencies of the sonority were calculated around this fixed one.

Table 6: The frequencies and nearest note in the tempered scale of *Nebula*'s second and third electronic sonorities, demonstrating how the electronics elide

Frequency in Hz (Nearest Tempered Note): *Nebula*'s second electronic sonority	Frequency in Hz (Nearest Tempered Note): *Nebula*'s third electronic sonority
1870.697 (B^{b}6)	1885.544 (B^{b}6)
1496.558 (F$^{\#}$6)	1849.847 (B^{b}6)
1478.082 (F$^{\#}$6)	1508.436 (F$^{\#}$6)
1187.819 (D6)	1496.558 (F$^{\#}$6)
960.9654 (B5)	1191.85 (D6)

Globally, the electronic part's vertical sonorities are ordered in a nearly symmetrical way (imitative of the interval serialization itself). However, the point of symmetry (and the location of the most dissonant chords) is moved from the temporal middle of the piece to the temporal golden mean of the piece. Also at the temporal golden mean of the piece, the cello reaches a climax by ascending to its highest note and by frantically rearticulating notes (the cell composed of entirely stationary intervals can be found at this point).

Obviously, many more details and much more compositional thought has gone into *Nebula* than has been expounded upon here. This chapter is only meant to provide a little insight, a glimpse at the architecture so to speak. It is not meant to be a note by note analysis.

CHAPTER IV

CONCLUSION

After expounding upon how interval serialization generates material and how I used the process in *Nebula*, certain clarifications are necessary. These clarifications speak to overarching, philosophical concerns.

The views expressed in this thesis are in no way a value statement on twelve-tone music or integral serialism (or tonality for that matter). I am not creating a new method out of a perceived fault or shortcoming in other methods of composition. I simply came upon this method, and after making some music using it, I enjoyed the sonic result. This process is not trying to compete with or attempting to supplant Schoenberg’s process. It is simply intended as another tool for composers.

With serial composition, the question may arise as to whether or not the observer should be able to perceive the serial processes. “Should I be able to hear that this note, not the one before it or the one to follow, completes the tone row? Or that these twelve notes are a transposition of the previous twelve? Or that the original twelve are now being presented in a backward order?” As for my own method, I have no illusion that the audience will be able to tell one group of interval directions from another; in the middle of my own compositions, I certainly cannot.

This answer is generally followed by the question of the purpose of composing in this manner. Why have such a systematic method of organizing musical ideas if the audience do not perceive (let alone understand) the method? This question can be directed at a myriad of compositional devices, from the medieval era's rhythmic *taleas* and melodic *colores*, to the Baroque's motivic inversions and retrogressions in a fugue? Yet theorists and composers strongly assert that these compositional constructs are present and vital to the music.

As for myself, I have found that the act of composing is not primarily an act of creating music. Once a person has tried his/her hand at it for a while and learned many of the standard compositional devices, an enormous wealth of musical material is at his/her fingertips. Composition is the act of pursuing some musical ideas and disregarding others. When I compose, I work from the mindset that, "This piece is about X and I will not use any musical ideas that detract from or distract from X."

The technique of interval serialism is therefore not meant to be heard by the audience, as such. What the process does is give the composition a very strong sonic unity. Why have an ascending perfect fourth at this point of the composition? Because there is a descending one thirty-two bars prior. Also, because of how tightly self-referential and self-generating this composition is, changing this interval to a tritone would require changing almost the entire composition. This process creates a sonic world where certain ideas belong and others do not. For the listener then, *Exploring the Third Major Nebula* has a cohesion that cannot be denied. By using interval serialization, *Nebula* is an organic, unified whole.

BIBLIOGRAPHY

Cope, David. *Techniques of the Contemporary Composer*. United States: Schirmer Thomson Learning, 1997.

Hisama, Ellie. *Gendering Musical Modernism*. New York: Cambridge University Press, 2001.

Kassel, Richard. *Barstow as History: An Introduction to the Sound World of Harry Partch*. Madison,Wisconsin: A-R Editions, 2000.

Kostka, Stefan. *Materials and Techniques of Twentieth-Century Music*. 2nd ed. Upper Saddle, New Jersey: Prentice Hall, 1999.

Tenney, James. *Selected Works 1961-1969*. Compact disc liner notes. New World Records, 2003.

APPENDIX A

Exploring the Third Major Nebula

for cello and Pure Data

D. Travis Clem

Instructions

Required Hardware:
Laptop
Digital/Audio Interface (capable of sending 3 channels of audio and possibly having a MIDI port)
Cable to connect the interface to the laptop (kind varies with interface)
Mixer (capable of receiving 3 channels and sending 3 channels)
A Speaker System (either the performance hall's or your own; capable of performing 2 channels)
Headphones
Audio cables (kind and number varies with setup)

Optional Hardware (strongly encouraged):
MIDI foot-pedal
MIDI cable
Contact Mic (if the cellist wishes to hear his/her instrument in their headphones without the drone)

Required Software:
Pure Data (software can be downloaded at http://crca.ucsd.edu/~msp/software.html)
Digital/Audio Interface's Software (the specific software varies with setup)

Setup at the Premier:
Laptop with Microsoft Windows Home Edition operating system
Tascam US-1641 Digital/Audio Interface with USB 2 cable
Yamaha MFC10 Foot-pedal
The versionn of Pure Data was PD 0.41-4

Setup Instruction:

Load Pure Data (PD) on the laptop. Load the Digital/Audio Interface's software on the laptop. Test the setup to make sure PD can send sound out of the interface (if using a foot-pedal, test it and the interface to make sure PD is receiving MIDI data).

Using the audio cables, connect the interface to the mixer and the mixer to the speaker system so that Channel 1 is the Left Speaker, Channel 2 is the Right Speaker, and Channel 3 is the cellist's headphones (if using a contact mic, its sounds need to be routed into the cellist's headphones also). For a rough sketch of the setup, see the opposite page.

Performance Instruction:

Before each performance, make sure that PD is communicating with the interface using the "Media" tab. Open the file "3rdMajorNebula.pd" in PD. Click on the "A" radio button to begin the program at the "A" marking in the score. Click on the button "; pd dsp 1". Raise the vertical slider to a good level; it controls the volume of the pitches in the cellist's headphones.

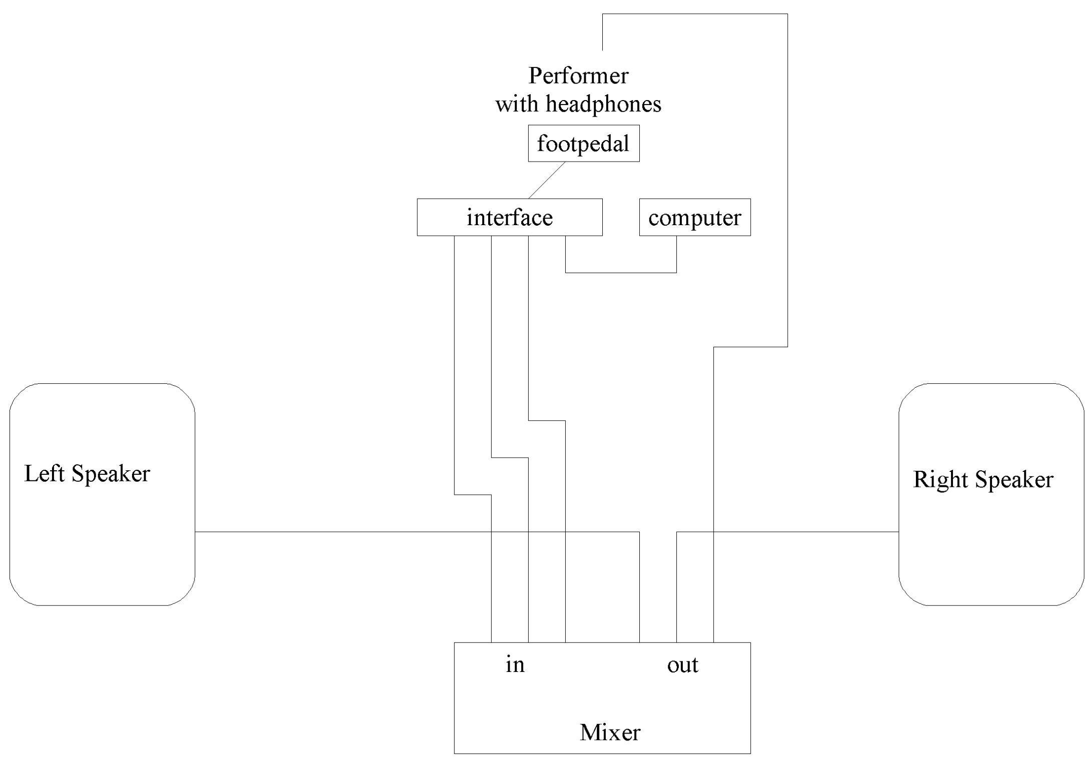
The cellist wears the headphones and matches the pitches (or aural cues) presented in them. To advance the the program to the next aural cue, send a controller number of "1" using the MIDI foot-pedal (if not using the foot-pedal, a second person hits the "Enter" button on the laptop). The electronic sonorities in the speakers are programmed to change at specific points in the aural cues; the person advancing the program doesn't control them directly. At the end of the performance, send a controller number of "2" on the foot-pedal (if not using the foot-pedal, a second person hits the "Backspace" button on the laptop) to fade out the electronics in the speakers. This action is also used to reset the program.

The rehearsal marks in the score correspond to the radio buttons in the program. To go to a rehearsal mark, reset the program (as described above) and click on the desired button.

Other Notes:

The written pitches in the music are a rough facsimile of the aural cues that are in the headphones. The aural cues provide the pitch, the cellist matches it, and the score shows them the general area they will depress the string. The score accurately portrays articulation, bowing style, and dynamics and should be followed in these regards.

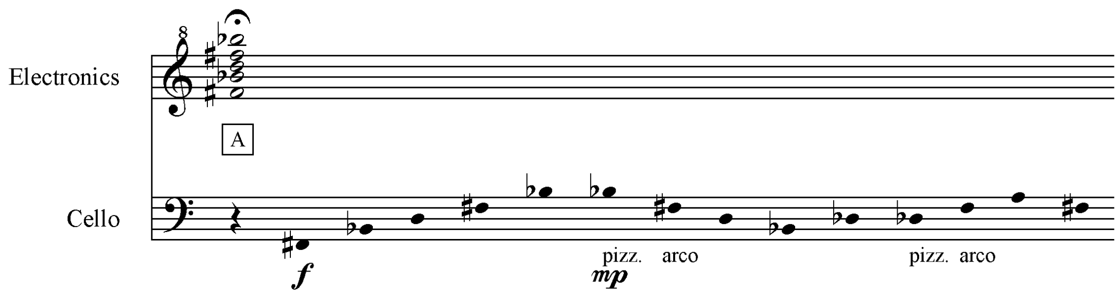
The general mood and tempo of the composition is that of a drone piece. This mood is broken, however, for the climax in the second stave of page seven, a written accelerando leads into it and a deccelerando leads to the end. Use slow, deliberate pacing at the beginning and end to give as much drama as possible to the climax.

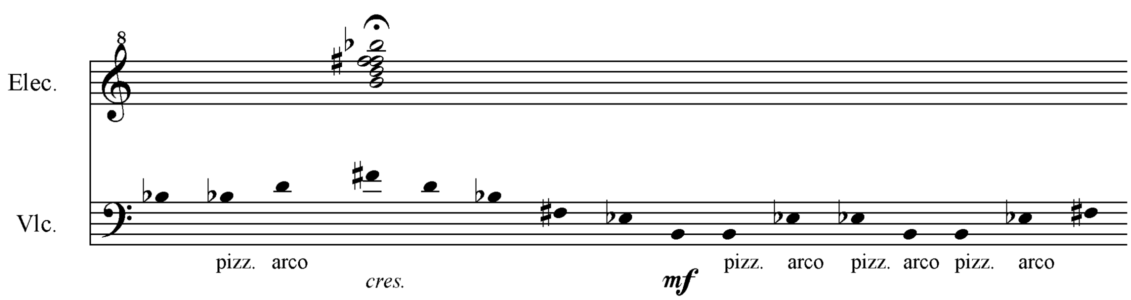


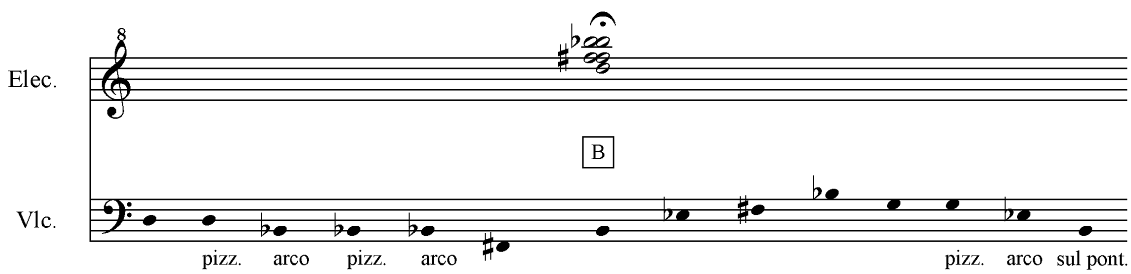
Score

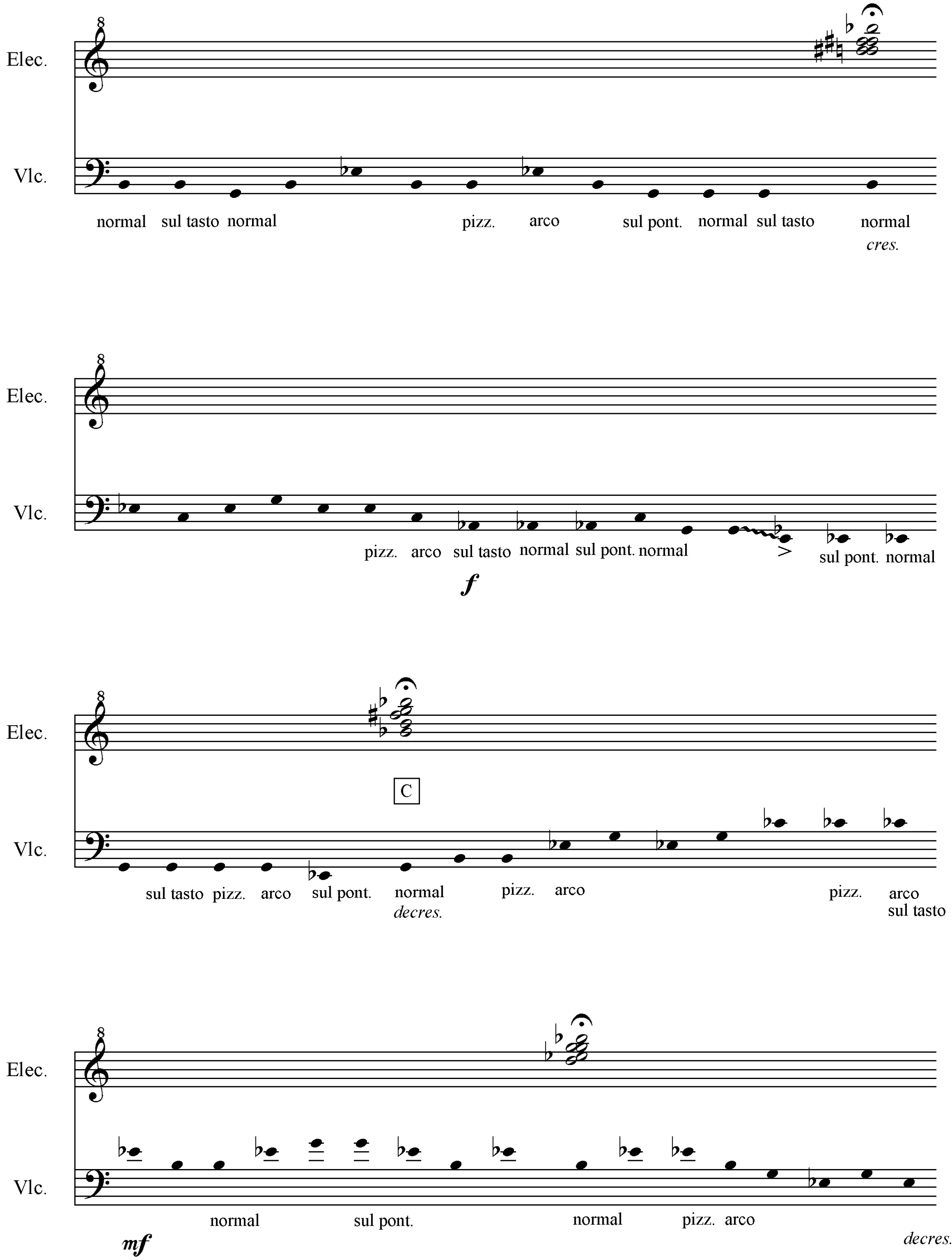
Exploring the Third Major Nebula

D. Travis Clem

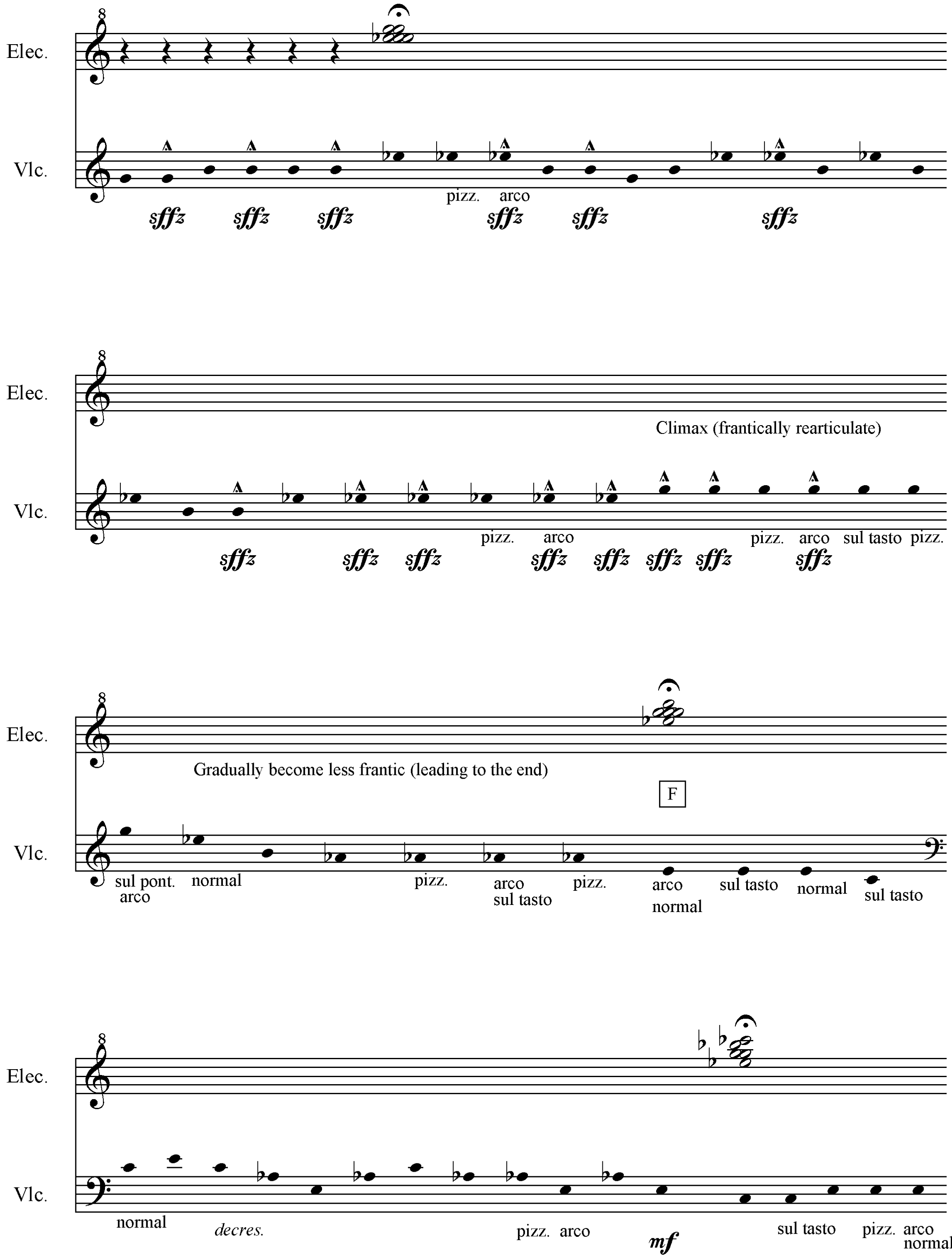


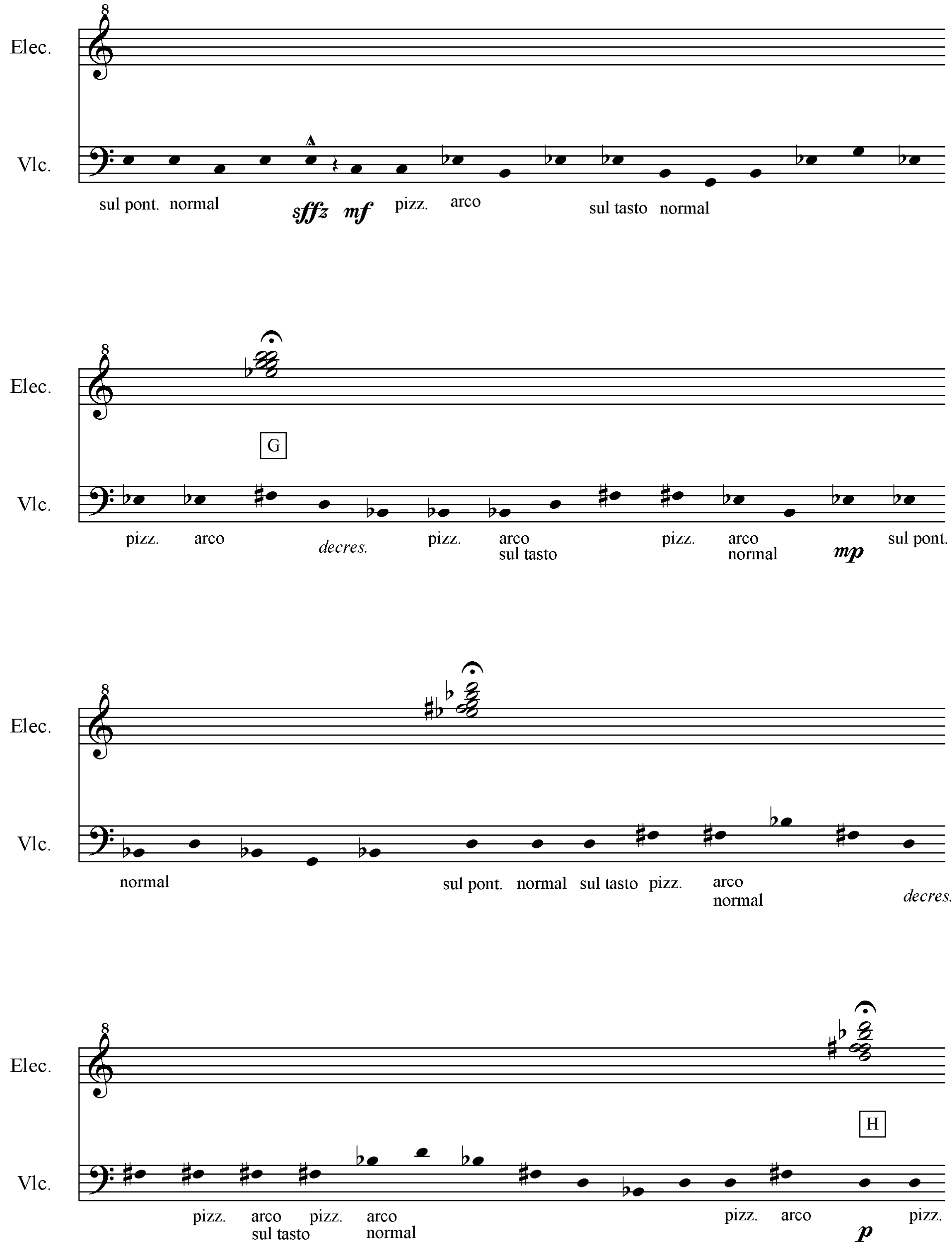


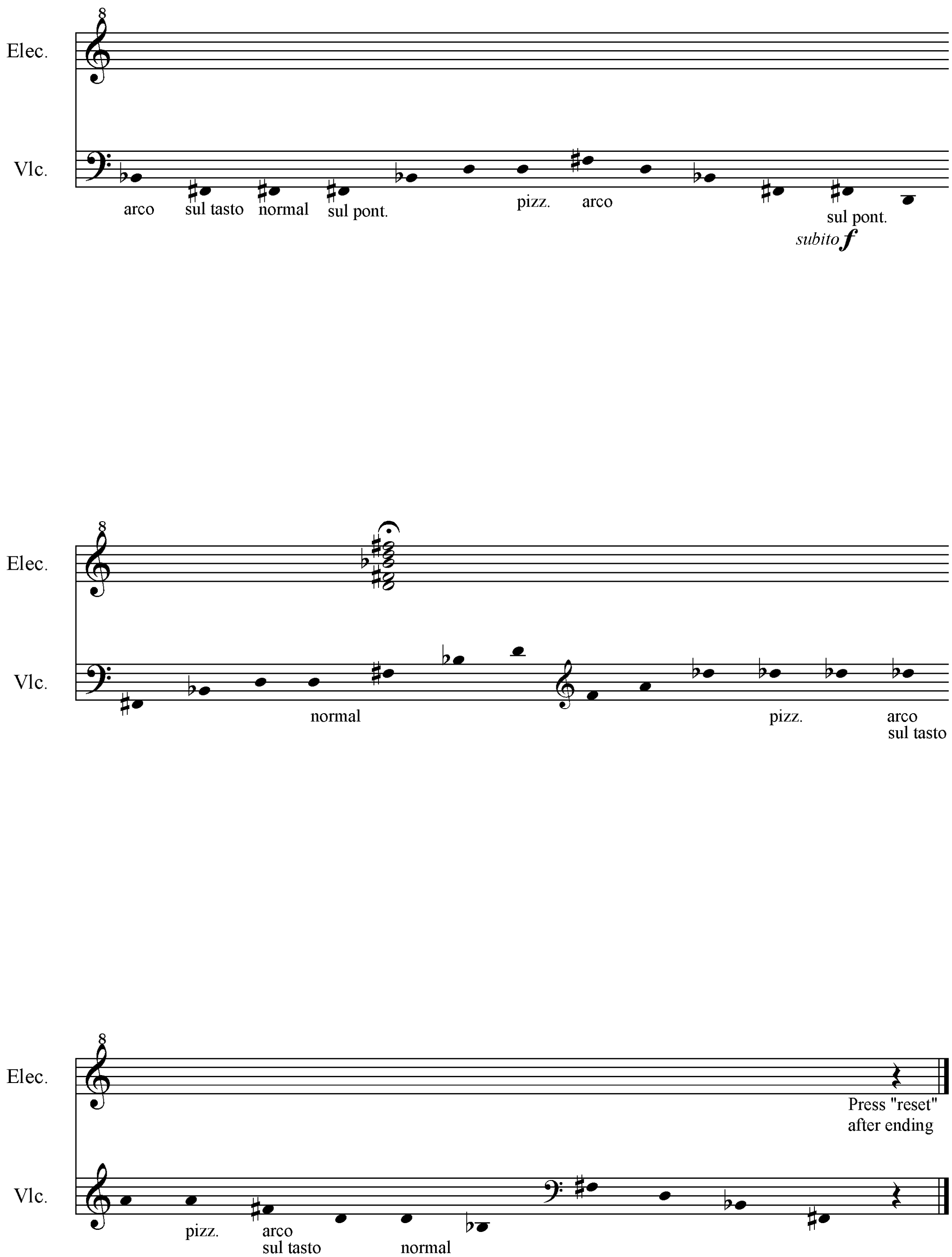



Elec.
Vlc.
normal sul tasto normal pizz. arco sul pont. normal sul tasto normal
cres.
Elec.
Vlc.
pizz. arco sul tasto normal sul pont. normal sul pont. normal
f
Elec.
C
Vlc.
sul tasto pizz. arco sul pont. normal pizz. arco pizz. arco
decres. sul tasto
Elec.
Vlc.
normal sul pont. normal pizz. arco
mf
decres.


Elec.
Vlc.
D
pizz.
arco
sul tasto
normal
mp
pizz.
arco
sul pont.
pizz.
Elec.
Vlc.
arco
normal
sffz
p
sffz
mf
sul pont.
normal
sffz
Elec.
Gradually become more frantic (leading to the climax)
E
Vlc.
sffz
sffz
sffz
sffz
sffz
Programmed fade out
Elec.
Vlc.
sul pont.
normal
sul pont.
normal
sffz
sffz
cres.
sffz


Elec.
Vlc.
sffz
pizz.
arco
Climax (frantically rearticulate)
sul tasto
Gradually become less frantic (leading to the end)
F
sul pont.
normal
decres.
mf


Elec.
Vlc.
sul pont. normal
sffz mf
pizz. arco
sul tasto normal
G
pizz. arco
decres.
pizz. arco
sul tasto
pizz. arco
normal
mp
sul pont.
normal
sul pont. normal sul tasto pizz.
arco
normal
decres.
pizz. arco
sul tasto
pizz. arco
normal
pizz. arco
H
p
pizz.


Elec.
Vlc.
arco
sul tasto
normal
sul pont.
pizz.
arco
sul pont.
subito f
Elec.
Vlc.
normal
pizz.
arco
sul tasto
Elec.
Press "reset"
after ending
Vlc.
pizz.
arco
sul tasto
normal

APPENDIX B

PURE DATA PATCHES FOR *NEBULA*

The files attached to this document are the patches associated with *Exploring the Third Major Nebula*. To be able to access these files, one needs to have Pure Data loaded on their computer. The program is free to the public and can be downloaded at <http://crca.ucsd.edu/~msp/software.html>.

"3rdMajorNebula.pd" is the required patch for a performance of *Nebula*. "3rdMajorNebulaSt.pd" is an example patch. It is not meant for performance, but is provided so that anyone with a computer and stereo speakers may hear the electronic sonorities and the cellist's aural cues. The cues do not represent dynamics, bowing styles, or even repeated notes. They do let the reader hear the result of the four major thirds in interval serialization without an actual performance.

To use either patch, open the file using PD. Using the "Media" tab, connect the program to one computer's sound system one would like to play the sound through. To turn on the patch, click on the button "pd dsp 1". Click on the radio button below "A" to start the patch; if one wishes to start at any other point in the composition, these radio buttons correspond to the rehearsal marks in the score. To adjust the level of the aural cues, raise or lower the slider near "cellovolume". To advance the aural cues (and by proxy the composition), press the "Enter" key on the computer's keyboard; to restart the aural cues and to fade out the high electronics, press the "Backspace" key on the computer's keyboard and click on the radio button under "A". To turn off the patch, click on the button "pd dsp 0". For instruction on how to set-up a performance of *Nebula*, see page two of the score in Appendix A.