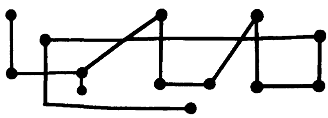
THE DYAD SYSTEM (PARTS TWO AND THREE)



JAMES DASHOW

II. THE ELECTRONIC SOUND DIMENSION

THE IDEA OF the Dyad System began with my early work on finding means for controlling the frequency content of certain classic sound-synthesis techniques in a more musically interesting fashion, in particular, the "grandaddy" of them all, John Chowning's frequency-modulation (FM) algorithm.[1]

Initially, I discovered that the frequency-modulation algorithm, now under precise computer control, could be piloted by pitch-pairs, and that each pitch-pair was capable of being reinterpreted by the algorithm to produce wide varieties of sounds *all of which had at least the frequencies of the pitch-pairs in common.* This approach was extended to any algorithm for which it was possible to find a precise mathematical representation of the frequencies in the generated spectrum; and by working backwards from the general mathematical expression to a specific frequency pair, it was possible to determine the equations by which the computer could calculate, for any pitch-pair, the input values for the synthesis algorithm that would produce sounds that *always* contained that particular pitch-pair as components of sounds that were nevertheless distinctly different from each other in timbre. This pitch-pair is called the *Generating Dyad*. It became immediately clear that the Generating Dyad's specific *pitch interval* was the controlling factor, i.e., a major ninth produces spectra vastly different from the same two pitch classes realized as an octave and a minor seventh, or as two octaves and a major second. Further, there emerged the added benefit of declaring one of the two pitches of the Generating Dyad as a phase-reversed frequency, or negative frequency, which, to the ear, is the same thing as a positive frequency, but for the mathematics of the algorithm radically alters the result and, in effect, doubles (or more) the number of possible sounds that a single algorithm can produce containing the Generating Dyad.

Composing for the electronic medium can be, then, in terms of Generating Dyads, a process of building up structures based on dyad manipulations. It became apparent, too, that one could imagine electronic sounds and, with a little experimentation, find the Generating Dyads that would produce those sounds; once having become acquainted with certain kinds of sounds associated with particular dyads, one could compose in terms of sounds available from certain dyads, and develop compositions timbrally via dyad structures, the complement to elaborating pitch structures (construed as dyads) via the sounds they can be made to generate.

Thus the concept that the Dyad System fully *integrates* pitch and electronically generated timbre, making them functionally interdependent: particular dyads *are* their timbres, particular (electronic) timbres *are* their dyads. Structural decisions in one domain mean that important characteristics in the other have already been determined. This interdependence suggested also the notion of a circular hierarchy, where different elements (dyads, timbre, intervals, pitch) dominate the compositional process at different times during the course of the work.

A second concept that has emerged during work with the System, one of the segments of the circular hierarchy, is that of considering the electronic sounds that are produced by the Generating Dyads as *harmonizations* of the dyads themselves. Some of the most interesting, and certainly the most characteristic, sounds in the electronic domain are the *inharmonic* sounds, made up of collections of frequencies that do not match any particular intonation system. This notion of the electronic spectra as harmonizing specific pitches with what might be called *inharmonic chords* suggests conceiving of these sounds as *prolongations*, in the traditional sense, of the Generating Dyads via timbral sequences. At one level one can design a sequence of radically differing electronic sounds, a rapidly changing surface that would communicate a high degree of energy, variety, and intricate detail; yet at another level these sounds would all be produced by a very small number (three or two or even one) of Generating Dyads, and hence express a different kind of rhythm at that level, where they would be heard as (slowly evolving) prolongations of those dyads via a rapid turnover of surface materials.

Interval prolongation can be similarly reinterpreted: a specific interval or small collection of intervals can be prolonged not only with differing kinds of electronic sounds, but also simply by using transpositions of a single particular sound which often significantly alters the latter's timbral character. The potential for linear development by interval prolongation is striking, especially in a multichannel (quadraphonic or more) sound environment where spatialization (the illusion of sound moving in three-dimensional space) can enhance the independence and clarity of multiple lines.

Lastly, specific pitch-class prolongation, where the same pitch-class pair of a given dyad is constantly being reinterpreted as to its interval type, suggests a potential "background" level that can yield the emergence of multiple developmental structures mutually (and subsequently) influencing each other at the dyad level, the interval level, and at the timbral level. The system offers, then a wide range of possibilities for structural design, from a basic direct approach all the way to a multi-tiered dimensioning of great complexity.

Most of the Generating Dyad procedures have been fully documented elsewhere, hence they are presented here in a rather terse form; I refer the interested reader to the original publications.[2] The Dyad System approach to additive synthesis has been expanded since the earlier articles and is described here in detail.

In all the examples that follow, the Generating Dyad, now in its *sounding* interval form (that is, interpreted as a particular pair of pitches), is communicated to the algorithm through the score data in 8.pc (octave point pitch-class) notation, and converted to frequency values either in the score or in the sound-synthesis instrument (algorithm) itself, prior to being subject to the Generating Dyad equations. Thus, the actual numeric values in the following equations are the converted pitch-to-frequency (or sampling increment) values. Conceptually, one can imagine them as pitches.

1. FREQUENCY MODULATION

The FM algorithm requires as input, minimally, a carrier frequency and a modulating frequency. We call the upper pitch of the Generating Dyad *HZU*, the lower pitch *HZL*, and their position in the FM spectrum, their sideband numbers, are *SBU* and *SBL*, respectively. The modulating and carrier frequencies are then calculated:

$$MOD = (HZU - HZL)/(SBU - SBL)$$

$$CAR = HZU - (SBU * MOD)$$

For the same two pitches *HZU* and *HZL*, changing the difference in their positions (*SBU* – *SBL*) changes the *MOD* and *CAR* values and hence the timbre. One of the two pitches may be a reflected (negative) frequency, which again changes significantly the kind of timbre, while maintaining the presence of the two pitches in the interval that forms the Generating Dyad.

2. RM COMPLEX (RING MODULATION, OR SIGNAL MULTIPLICATION)

Two sound-producing oscillators are multiplied together. The resulting sound is the *sum* and *difference* of their two frequencies, while the original frequencies are no longer audible.

If the two oscillators are already producing a compound signal then the multiplications yield a larger number of sum and difference tones. For RM Complex, each of the two oscillators produce sounds consisting of two different frequencies; each oscillator output is the sum of integer multiples of the input frequency, and we specify those integers as supplementary data to the algorithm.[3] The multiplication of two signal pairs yields eight frequencies in the generated spectrum. Working backwards we find sixteen equations that can represent any Generating Dyad. If one of the frequencies of the Dyad is declared reflected (negative) these same sixteen equations produce sixteen more possibilities, thirty-two all together. The sixteen equations follow, where UP is the upper pitch of the Generating Dyad, LP the lower pitch, x and m are the partial numbers in the look-up table for oscillator A (or the values to multiply the A frequency for the FLTOSC version), y and n are the partial numbers (or values) for the B oscillator, A and B are the two oscillators' input frequencies calculated by the algorithm from the pitch and partial number data supplied by the musical score. (See Example 3.)

1. $\mathrm{UP} = xA + yB$
   $\mathrm{LP} = xA - yB$
   $B = (\mathrm{UP} - \mathrm{LP})/2y$
   $A = (\mathrm{UP} - yB)/x$

2. $\mathrm{UP} = xA + yB$
   $\mathrm{LP} = xA + nB$
   $B = (\mathrm{UP} - \mathrm{LP})/(y - n)$
   $A = (\mathrm{UP} - yB)/x$

3. $\mathrm{UP} = xA + yB$
   $\mathrm{LP} = xA - nB$
   $B = (\mathrm{UP} - \mathrm{LP})/(y + n)$
   $A = (\mathrm{UP} - yB)/x$

4. $\mathrm{UP} = xA + yB$
   $\mathrm{LP} = mA + yB$
   $A = (\mathrm{UP} - \mathrm{LP})/(x\text{-}m)$
   $B = (\mathrm{UP} - xA)/y$

5. $\mathrm{UP} = xA + yB$
   $\mathrm{LP} = mA - yB$
   $A = (\mathrm{UP} + \mathrm{LP})/(x + m)$
   $B = (\mathrm{UP} - xA)/y$

6. $\mathrm{UP} = xA + yB$
   $\mathrm{LP} = mA + nB$
   $A = (n\mathrm{UP} - y\mathrm{LP})/(xn - ym)$
   $B = (\mathrm{UP} - xA)/y$

7. $\mathrm{UP} = xA + yB$
   $\mathrm{LP} = mA - nB$
   $A = (n\mathrm{UP} + y\mathrm{LP})/(xn + ym)$
   $B = (\mathrm{UP} - xA)/y$

8. $\mathrm{UP} = xA + nb$
   $\mathrm{LP} = xA - nB$
   $A = (\mathrm{UP} + \mathrm{LP})/2x$
   $B = (\mathrm{UP} - xA)/n$

9. $\mathrm{UP} = \mathrm{x}A + nB$
   $\mathrm{LP} = mA + \mathrm{y}B$
   $A = (y\mathrm{UP} - n\mathrm{LP})/(xy - mn)$
   $B = (\mathrm{UP} - xA)/n$

10. $\mathrm{UP} = xA + nB$
    $\mathrm{LP} = mA - yB$
    $A = (y\mathrm{UP} + n\mathrm{LP})/(xy + mn)$
    $B = (\mathrm{UP} - xA)/n$

11. $\mathrm{UP} = xA + nB$
    $\mathrm{LP} = mA + nB$
    $A = (\mathrm{UP} - \mathrm{LP})/(x - m)$
    $B = (\mathrm{UP} - xA)/n$

12. $\mathrm{UP} = xA + nB$
    $\mathrm{LP} = nA - nB$
    $A = (\mathrm{UP} + \mathrm{LP})/(x + m)$
    $B = (\mathrm{UP} - xA)/n$

13. $\mathrm{UP} = mA + yB$
    $\mathrm{LP} = \mathrm{mA - yB}$
    $B = (\mathrm{UP} - \mathrm{LP})/2y$
    $A = (\mathrm{UP} - y\mathrm{B})/m$

14. $\mathrm{UP} = mA + yB$
    $\mathrm{LP} = mA + nB$
    $B = (\mathrm{UP} - \mathrm{LP})/(y - n)$
    $A = (\mathrm{UP} - y\mathrm{B})/m$

15. $\mathrm{UP} = mA + yB$
    $\mathrm{LP} = mA - nB$
    $B = (\mathrm{UP} - \mathrm{LP})/(y + n)$
    $A = (\mathrm{UP} - y\mathrm{B})/m$

16. $\mathrm{UP} = mA + nB$
    $\mathrm{LP} = mA - nB$
    $A = (\mathrm{UP} + \mathrm{LP})/2m$
    $B = (\mathrm{UP} - mA)/n$

A = carrier frequency, B = modulator frequency
General Version Equation: $(\cos(xA) + \cos(mA)) * (\cos(yB) + \cos(nB))$
NOTE: For reflected LP, substitute –LP for LP in all A and B equations.

EXAMPLE 3: RM-COMPLEX

Several variants on the above are possible. In particular the A and B signals could be saved prior to the multiplication and then summed into the result, which would increase by twenty the number of possible sounds containing the Generating Dyad.[4]

Another algorithm, called AMRM, uses three oscillators in a mix of amplitude modulation and signal multiplication (in this case we have a Generating "Tryad"—see Dashow, "Three Methods").

An alternative is to use a straightforward amplitude-modulation algorithm, rather than beginning with the algorithm output, it is possible to calculate backwards with respect to the routines that generate the compound partial tables (the look-up tables) used by the interpolating oscillators. In this instance there are three kinds of calculations that can be done, two with a Generating Tryad, the third with a Generating Dyad; interesting inharmonic sounds can be obtained by adding supplementary controls to the partial generating data to create still other timbral choices (see Dashow, "Spectra as Chords").

3. FOLDOVER

Foldover (also called aliasing) is one of the "defects" of digital signal synthesis and processing. It is not possible to call for the digital synthesis of a frequency higher than half the Sampling Rate (*SR*); if such a frequency (*HZ*) is called for, the result will not be the desired frequency but rather the frequency (*FO*) equal to *SR* – *HZ*. Often these foldover frequencies are a source of inharmonic frequencies. By calling for a specific frequency to be generated *as* a foldover frequency, and using a compound table with several, arbitrarily chosen, partials for the interpolating oscillator, it is possible to generate a desired frequency as a foldover phenomenon accompanied by a variety of inharmonic sounds. Generating Dyad input requires, then, two of these foldover circuits together, each one being driven by the following equation (evaluated by the computer) for the upper or lower note of the Generating Dyad:

$$HZ = \frac{N * SR \pm Note}{PN}$$

where *HZ* is the frequency input to the interpolating oscillator, *N* is any integer, *Note* is the frequency value of the desired pitch which will be produced *as a foldover frequency*, and *PN* is the partial number in the oscillator look-up table which will generate the desired pitch (see Dashow, "New Approaches").

4. ADDITIVE SYNTHESIS

The Generating Dyad approach to additive synthesis yields a singularly rich variety of timbres that significantly increases the potential for a highly varied prolongation of the Generating Dyad. The several algorithms that divide up the frequency space around a pitch-pair are subjected to the "working backwards" method to derive equations for producing the Generating Dyad as members of a set of frequencies generated by repeated (successive) application of some kind of constant ratio. By continuing the synthesis according to the derived differences or ratios, the dyad is embedded in frequency collections that often have inharmonic relationships among themselves and/or to the dyad members. The specific kind of frequency ratio or difference that the Dyad can be made to generate determines the character of the surrounding frequencies, that is, the timbral quality of the spectrum.

The primary unit generator employed here is a multiple interpolating oscillator, called OSCILM in the MUSIC30 sound-synthesis language.[5] OSCILM and related unit generators can be emulated by ordinary "brute force" additive synthesis in other software synthesis languages simply by summing the appropriately driven number of OSCILI signals.

The factor that contributes most to the richness of these sounds is the possibility of generating *each* frequency of the OSCILM with a harmonic table that is *already* a compound construction of the sum of an arbitrary number of partials with whatever desired phase, amplitude, and frequency ratios. For example, if the call to OSCILM is for (the summed output of) six oscillators, and the table being read for each of the six contains four partials, the total output of the OSCILM will be a signal containing twenty-four frequencies. By calling for the Generating Dyad to be produced on *any one* of the partials in the table, we have the possibility of generating extremely rich sounds with an immense variety of timbral contrast that can range from radical to extremely subtle, all for the same dyad.

Conceptually, each single oscillator output, in either the compound OSCILM version or in a standard additive synthesis summation of single oscillators, is considered here a "component" or "sideband" of the complex spectrum that is the *result* of the oscillator summation.

The first version of additive synthesis mimics the kind of spectrum obtained from one-sided frequency modulation (the upper sidebands only), without the attendant amplitude (Bessel function) variations. In fact the additive-synthesis timbre as developed here is static, not dynamically variable as in FM or wave-shape modulation. This kind of sound, and its mode of deployment, is appropriate for musical contexts quite different from those that are designed using FM.

Given the Generating Dyad *HZU* and *HZL*, we consider that the two frequencies will be generated as any two of the sidebands in the additive synthesis spectrum, that is *HZU* and *HZL* can be frequency number *NU* and *NL* respectively, where *NU* and *NL* are integers from 1 onward (maximum of 12 for OSCILM). Then the frequency *difference* between them, *HZU* – *HZL*, can be divided into equal parts or steps to obtain the frequency to be added to each successive component in the spectrum:

$$AddHz = (HZU - HZL)/(NU - NL).$$

Alternatively, if we declare only the bottom pitch of the dyad (*HZL*) and the *interval ratio* (*NTRVL*) that *HZU* forms with *HZL*, then we can calculate *AddHz* by

$$AddHz = \frac{NTRVL - 1}{NU - NL} * HZL$$

If the bottom pitch of the Generating Dyad, *HZL* is also the frequency of the first component in the additive synthesis spectrum, then we simply begin synthesis with *HZL* and add *AddHz* to each preceding frequency to obtain the frequency of each successive sideband, which yields *HZU* as sideband *NU*. But we can also start on a frequency *less than HZL*, which will generate frequency components below the bottom pitch of the Generating Dyad, so the general expression for the initial frequency component in the spectrum is

$$Hz(0) = HZL - (NL - 1) * AddHz$$

For example, we can divide up a particular interval, say, the 8ve + mn7 9.10 (= 932.33 Hz) and 8.00 (= 261.63 Hz), declaring the 9.10 to be the sixth sideband, and 8.00 the second:

$$AddHz = (932.33 - 261.63)/(6 - 2) = 167.675$$

and the lowest component frequency will be

$$Hz(0) = 261.63 - (2 - 1) * 167.675 = 93.955$$

If we then call for a density of seven components beginning with 93.955 *Hz* and successively adding 167.675 *Hz* to each one, we generate a spectrum with the following sidebands:

oscil num	*Hz*	*ratio to preceding Hz*
1.	93.955 = *Hz*(0)	
2.	261.63 = 8.00	2.78 = ca. 17.73 semitones
3.	429.305	1.64 = ca. 8.56 semitones
4.	596.98	1.39 = ca. 5.7 semitones
5.	764.655	1.28 = ca. 4.27 semitones
6.	932.33 = 9.10	1.22 = ca. 3.44 semitones
7.	1100.005	1.18 = ca. 2.86 semitones

If the harmonic partials table being referenced for the additive synthesis consists of, say, only partial numbers 6, 13, and 28, and we want the frequencies to be generated on the sixth partial, then both the *AddHz* value and the *Hz*(0) value must be divided by 6; in this case *each* frequency will be generated as shown on the sixth partial along with the frequency $Hz(n) * 13/6$ and $Hz(n) * 28/6$, a total of twenty-one frequencies with a rather thick spectrum of inharmonic relationships. Or, we could call for synthesis to take place on the thirteenth partial of the same table, simply by dividing *AddHz* and *Hz*(0) by 13; we would obtain, then, all frequencies as shown accompanied by $Hz(n) * 6/13$ and by $Hz(z) * 28/13$. This would have the effect of shifting the complex sound down by the ratio 6/13 = .46 = down 13.4 semitones, and significantly altering the timbre *while maintaining the Generating Dyad 9.00 8.10 in its place.* By changing the partial table, and/or the number of steps between the Generating Dyad ($NU = 4$, $NL = 1$, for example), we find that the potential for timbral transformation and development, even for a single Generating Dyad, becomes quite large.[6]

Alternatively, the frequency distance between *HZU* and *HZL* can be divided in equal *proportions*, which yields a constant ratio (interval) between successive steps. In this case the additive synthesis output is such that the frequency value of each sideband *N* is given by:

$$Hz(N) = Hz(N-1) * MltFac$$

or alternatively

$$Hz(N) = Hz(0) * MltFac^N$$

where *MltFac* is the constant ratio between successive frequencies, and $Hz(0)$ is the first (lowest) frequency to be generated.

Once again, we work backwards from the specific values *HZU* and *HZL* to be generated by oscillators *NU* and *NL*: if we call the ratio $HZU/HZL = NTRVL$, then

$$MltFac = NTRVL^{(1.0/(NU - NL)}$$

and the initial (base) frequency of the spectrum is

$$Hz(0) = \frac{HZL}{MltFac^{(NL-1)}}$$

If the harmonic table for the oscillator routine is the sum of various partials, then it is necessary to divide the frequency values used in the calculations by the integer corresponding to the partial number that will generate the dyad frequencies, just as we did for the first example above.

And as before, all these calculations are user-programmed into the algorithm, so that the composer's score (data) consisting of the specific Generating Dyad values, and the *NU* and *NL* component numbers, are transformed by the computer into the data required for the OSCILMX unit generator, or other additive synthesis process.

Using the same interval as before, the 8ve + mn7, with 9.10 and 8.00, the additive synthesis spectrum for this version will have the following frequencies (since the ratio between frequencies is constant, the interval in semitones is the same):

$$MltFac = (932.33/261.63)^{(1.0/(6-2))} = 1.374$$

$$Hz(0) = 261.63/(1.374^{(2-1)}) = 190.41$$

oscil num	*Hz*	*ratio to preceding Hz*
1.	190.41 = $Hz(0)$	
2.	261.63 = 8.00	1.374 = 5.5 semitones
3.	359.485	1.374 = 5.5 semitones
4.	493.93	1.374 = 5.5 semitones
5.	678.65	1.374 = 5.5 semitones
6.	932.47 = 9.10	1.374 = 5.5 semitones
7.	1281.213	1.374 = 5.5 semitones

It should be mentioned that the fractional discrepancies in frequency precision (9.10 was originally set at 932.33 Hz, while here it comes out as 932.47) are due to rounding-off errors. Most music synthesis is done with at least 32-bit floating point precision, (the T.I. TMS320C30 dsp has 40-bit extended precision as standard) and as such, rounding errors, if they occur at all, have (musically) unnoticeably small magnitudes.

If we take the expression for the natural harmonic series, where the *N*th partial, $Hz(N)$, is N times the fundamental frequency $F(0)$, for all integers N,

$$Hz(N) = F(0) * N,$$

we can rewrite it as

$$Hz(N) = F(0) * (N^S)$$

where S is a distortion or "stretch" factor,[7] and for harmonic partials is, in effect, hidden as $S = 1$. If S is anything other than 1, in particular non-integer, the partial series generated for $N = 1, 2, 3 \ldots$ is, to various degrees, inharmonic.

The Generating Dyad is considered as two frequencies of one of these stretched-partial systems, but rather than declaring S, the factor is calculated as before in terms of the Generating Dyad and, as before, their positions in the spectrum given by their component number, N.

The Generating Dyad frequencies HZU and HZL as the *NU*th and *NL*th frequency in the series are

$$HZU = F(0) * (NU^S)$$

and

$$HZL = F(0) * (NL^S)$$

The stretch factor S is then given by

$$S = \frac{\log(HZU/HZL)}{\log(NU/NL)}$$

The initial frequency, $F(0)$, is then

$$F(0) = \frac{HZU}{NU^S}$$

Again, if any other partial than the first in a compound table is being used to generate the spectrum, all frequency values must be divided by that partial number.

Using the same interval as before, the 8ve + mn7, with 9.10 and 8.00, but with the Generating Dyad in the spectrum now as the fifth and second component, respectively, the additive-synthesis spectrum for the stretched-partials algorithm will have the following frequencies:

$$S = \log(932.33/261.63)/\log(5/2) = 1.38685$$

$$F(0) = 932.33/5^{1.38685} = 100.04$$

oscil num	*Hz*	*ratio to preceding Hz*
1.	100.04 = $F(0)$	
2.	261.63 = 8.00	2.62 = ca. 16.64 semitones
3.	459.08	1.75 = ca. 9.74 semitones
4.	684.17	1.49 = ca. 6.9 semitones
5.	932.33 = 9.10	1.36 = ca. 5.36 semitones
6.	1200.55	1.29 = ca. 4.38 semitones
7.	1486.72	1.24 = ca. 3.70 semitones
8.	1789.18	1.20 = ca. 3.21 semitones

Further expansions of this approach can introduce still other "hidden" factors into the standard equations representing the chromatic scale or harmonic frequency patterns: these factors are hidden when their regular values in ordinary use (such as the twelfth-root-of-two value for dividing the octave into twelve equidistant steps, or integer values for the multiplications that determine the harmonics of a fundamental frequency) are 1 (if multipliers) or 0 (if exponents or if added), hence they have no effect, and are not even included in the standard mathematical representations of the semitone or of harmonics. But by rewriting these standard equations to include these factors set differently, or solving the equations in terms of the Generating Dyad (as above) with these factors included, we can obtain an enormous variety of "inharmonic" harmonic patterns and very different scale-step ratio divisions, all of which can be made to produce the Generating Dyad as members of the generated scale or inharmonic spectrum.

For example, the previous "stretch factor" generating procedure can be written as a Multiply, Add algorithm:

$$Hz(N) = F(0) * (N^S) + A * N$$

or the *N*th sideband (frequency) in the spectrum is the result of first multiplying the starting frequency $Hz(0)$ by N raised to some stretch factor S, then adding some frequency A multiplied by the same N. In terms of a Generating Dyad this procedure requires the composer to specify the pitches of the Dyad, their sideband positions NU and NL in the spectrum and *either* the stretch factor S *or* the add factor (in *Hz*) A. The procedure then calculates $Hz(0)$ and whichever of the two supplemental factors was *not* specified, i.e., the A or the S, respectively.

A spectrum with the same octave and a mn7 Generating Dyad, dyad positions 2 and 5 in the spectrum and declaring a stretch factor of 3.73 produces an *Add Hz* as 125.844 Hz and generates the following set of frequencies:

Stretch fac = 3.73 *Add Hz* = 125.844

oscil num	*Hz*	*ratio to preceding Hz*
1.	0.749 = *F*(0)	
2.	261.63 = 8.00	349.33 = ca. 101.38 semitones
3.	422.62	1.615 = ca. 8.30 semitones
4.	635.24	1.503 = ca. 7.05 semitones
5.	932.33 = 9.10	1.467 = ca. 6.64 semitones
6.	1353.39	1.452 = ca. 6.45 semitones
7.	1944.20	1.436 = ca. 6.27 semitones
8.	2756.45	1.418 = ca. 6.04 semitones

In actual practice, the first frequency component would not be played.

We can compare the exact same conditions as the previous spectrum, but now with the Generating Dyad realized as the second and sixth component of the spectrum. The *Add Hz* is almost the same, but what is of primary interest is the interval content inside the spectrum.

Stretch fac = 3.73 *Add Hz* = 129.524

oscil num	*Hz*	*ratio to preceding Hz*
1.	0.194241 = *F*(0)	

2.	261.63	= 8.00	1346.92	= ca. 124.745 semitones
3.	400.27		1.53	= ca. 7.36 semitones
4.	552.29		1.38	= ca. 5.57 semitones
5.	726.23		1.31	= ca. 4.74 semitones
6.	932.33	= 9.10	1.28	= ca. 4.33 semitones
7.	1182.44		1.27	= ca. 4.11 semitones
8.	1489.99		1.26	= exactly 4 semitones

Again, the first component would not be played, and the net effect is of the Generating Dyad on the first and fifth positions. We can use stretch factors less than one, or very large, or negative *Add Hz* values to get different interval relationships inside each spectrum. Modest differences in the stretch factor (with the Generating Dyad in the same spectral position) produce only slight differences in the generated spectrum, which are ideal for gradually evolving timbral transformations.

The add first, then multiply version of the stretch factor procedure is written:

$$Hz(N) = N^S * (F(0) + A * N)$$

Another kind of Add, Multiply algorithm is

$$Hz(N) = M^N * (F(0) + A * N)$$

or, the Nth frequency is the result of adding some value multiplied by N to the starting frequency $F(0)$ and multiplying that sum by a factor M raised to the Nth power.

Using this algorithm with the same octave and a minor seventh Generating Dyad we generate completely different sounds. In the first of the two spectra, the positions of the Dyad pitches are the second and sixth component, in the second the positions are second and seventh. The Multiply factor is again given as 3.73 and the computer calculates the Add frequency, a negative value.

Multiply fac = 3.73 *Add Hz* = −17.212

oscil num	*Hz*		*ratio to preceding Hz*
1.	87.35	= $F(0)$	

2.	261.63	= 8.00	2.99 = ca. 18.991 semitones
3.	736.39		2.81 = ca. 17.916 semitones
4.	1853.49		2.52 = ca. 15.980 semitones
5.	3581.74		1.93 = ca. 11.405 semitones
6.	932.33	= 9.10	0.26 = ca. −23.301 semitones

Here the frequency components go beyond the upper pitch of the Generating Dyad (component 4 and 5) before returning to the upper pitch. This kind of frequency succession is not unusual (the effect of the *negative Add Hz* value) while producing some unexpected and often useful timbres. The seventh and eighth components in the above spectrum are frequencies well beyond the range of human hearing, so they are not played (one *could* digitally synthesize them in order to include their resultant *foldover* frequencies as part of the overall sound, but the effect will be different for each octave and a mn7 interval, whereas the interval ratios in the frequency range of 0 *Hz* up to half the sampling rate are always the same for all specific transpositions of the pitches forming the Generating Dyad).

Now with the Generating Dyad at positions 2 and 7:

Multiply fac = 3.73 *Add Hz* = −13.959

oscil num	*Hz*		*ratio to preceding Hz*
1.	84.09	= *F*(0)	
2.	261.63	= 8.00	3.11 = ca. 19.648 semitones
3.	781.65		2.98 = ca. 18.948 semitones
4.	2191.17		2.80 = ca. 17.845 semitones
5.	5471.04		2.49 = ca. 15.841 semitones
6.	10328.47		1.89 = exactly 11 semitones
7.	932.33	= 9.10	0.09 = ca. −41.636 semitones

Again, the too-high eighth component has been deleted.

Other "hidden factors" can be in the exponent or the multiply factor itself, as the result of some inner arithmetic operation. One such procedure is written:

$$Hz(N) = F(0) * M\wedge N + A * N\wedge X \rightarrow X = N + xx/N + yy$$

where the *N*th frequency component is the result of multiplying the starting frequency *F*(0) by some factor *M* raised to the *N*th power, then adding some frequency *A* multiplied by *N* raised to the *X*th power, where *X* itself is the result of dividing *N* plus some value *xx* by *N* plus some value *yy*; that is the *X* value might be something like 2 + 1.13 divided by 2 + 3.87 (yielding *X* less than 1) or the inverse (yielding *X* greater than 1). Depending on the Generating Dyad and its position in the spectrum, changes in any of these factors will yield varying degrees of timbral differences.

Lastly, a more complex algorithm that produces a large variety of extraordinarily rich sounds is written:

$$Hz(N) = M * (Hz(N-1)^2 / Hz(N-2))$$

The *N*th frequency is the result of dividing the square of the immediately preceding frequency in the series ($N-1$) by its immediately preceding frequency ($N-2$), and multiplying the result of that division by the (composer chosen) factor *M*. (If *M* were 1 and the ratio between $Hz(N-1)$ and $Hz(N-2)$ were 1.0594631, we would generate the standard chromatic scale.)

For all values other than 1, the multiply factor *M* generally behaves like an interval compander: values less then 1 produce decreasing intervals between successive spectrum components, whereas values greater than 1 produce *increasing* intervals between components. Furthermore, under certain conditions, the frequency series will first expand then contract, or vice versa. Using function tables like those shown below[8] for the synthesis of each of these frequencies can have particularly stunning results. The frequencies of the Generating Dyad are, of course, always two of those components.

The following two examples are realized with the same Generating Dyad as in the other examples. In the first the multiply factor *M* has been set to a value greater than 1 (1.0779), while in the second *M* is less than 1 (0.8987). In both, the Generating Dyad is realized as the second and fifth frequency component.

Interval M fac = 1.0779 (intervals expand)

oscil num	*Hz*	*ratio to preceding Hz*

1.	199.01 = $F(0)$		
2.	261.63 = 8.00	1.31	= ca. 4.74 semitones
3.	370.74	1.42	= ca. 6.04 semitones
4.	566.27	1.53	= ca. 7.33 semitones
5.	932.33 = 9.10	1.65	= ca. 8.63 semitones
6.	1654.58	1.77	= ca. 9.93 semitones
7.	3165.09	1.91	= ca. 11.23 semitones

Interval Mfac = 0.8987 (intervals contract)

oscil num	*Hz*	*ratio to preceding Hz*	
1.	138.34 = $F(0)$		
2.	261.63 = 8.00	1.89	= ca. 11.03 semitones
3.	444.66	1.69	= ca. 9.18 semitones
4.	679.19	1.53	= ca. 7.33 semitones
5.	932.33 = 9.10	1.37	= ca. 5.48 semitones
6.	1150.16	1.23	= ca. 3.63 semitones
7.	1275.17	1.11	= ca. 1.77 semitones

It should be recalled that all of these examples are realized with this algorithm using the *same* Generating Dyad: the same two pitch classes and the same specific interval, and almost the same supplementary factors (component position, Multiply or Add factor, etc.). The intervallic variety (and hence, electronic timbral variety) that just one single Generating Dyad can produce as demonstrated in these few examples should give some idea of the considerable potential that this approach to the synthesis of electronic sound most certainly possesses.[9]

Each of these additive synthesis procedures has its own particular optimum frequency and interval range, and this will be true in general for any algorithm that uses the Generating Dyad procedure for determining its input parameters: each algorithm will have more or less interesting (musically suggestive or appropriate) frequency ranges and interval types. As input to further elaboration by various signal-processing techniques, each of these sounds can be made to yield vast timbral and structural resources that this approach to the synthesis of electronic sound most certainly offers. A good deal of experimentation with the resources of this approach is necessary in order to get a feel for the musical potential of these sounds and their structural/expressive relationships.

Perhaps the most difficult task is the invention of convincing combinations, or simultaneities, of different kinds of sounds, getting them to mix dynamically, move around in space, interfere or reinforce, resonate with or cancel each other; and finally there remains the job of discovering ways of building the sounds of this system into a satisfying structural poetics. These are both technical and aesthetic problems: the composer's business as usual.

III. Using the System

A few examples are given here to demonstrate specific applications of the Dyad System, for both instrumental music and music for instrumentalists with electronic sounds.

These will, I think, show not only the flexibility of the System, but also, and especially, the nature of the partnership between the System and the composer, or rather, the role of the System in a real compositional context where the significant decisions are, as usual, left to the composer in working out the large- and small-scale dimensions of the musical idea.

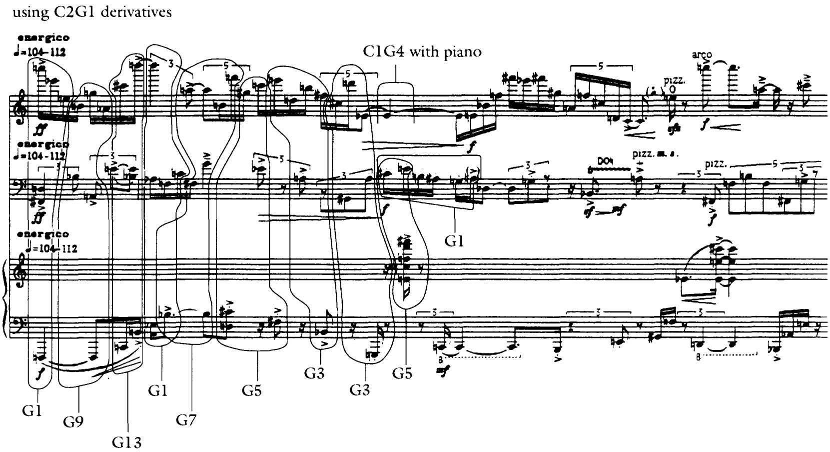
The first examples are from my *4/3-Trio*, for violin, cello, and piano.[10] Each of the work's three movements follows, or expands into, different paths through the resources of the Dyad System; each movement has its own ways of emphasizing certain kinds of intervals and interval successions, and more importantly, each movement works within and around its own characteristic sound (or better: its own characteristic motion through sound) due to the specific intervallic form each dyad takes at the moment of articulation. In other words, each movement uses different successions of groups and types and in different intervallic expansions, and each explores different means for hooking together varying kinds of group/type relationships.

The first movement, called "Dichiarazioni" (Statements), has several sub-sections each of which has its own group/type succession, where each sub-section succession is similar to that of the preceding subsection, but differing in a developmental sense—what I like to think of as "almost" variations, the implications of which are then made specific in the third movement, entitled "Varianti." The dense complex of developmental paths in the first part suggest points of departure for the second and third movements, generating a large-scale form that begins with a first movement accumulation of concentrated energy (the potential

EXAMPLE 4

departure points) that is gradually discharged over the remainder of the piece (second and third movements) as the workings out of the accumulated implications.

The beginning of the piece (Example 4) is constructed from differing groups, intertwinings of types that are designed from the very start to lay the groundwork for the energy complexes that accomplish the desired accumulations in part I. The primary groups are announced in the *fff* piano chords. The first is C2G1 (mn3, mn2, mj2, as D,F F♯,G E♭,D♭), the second is C2G3 (mn2, p4, mj2 as C,B G♯,C♯ G,A); the strings are already pulling out of the piano chord different intervals that point to other groups: the principle of keeping the same notes while changing their intervallic disposition is already evident. The piano figures in the bass immediately articulate other groups, including some of the possible regroupings of the principle group, as well as incomplete or intentionally ambiguous groupings which will become more precisely focussed during the first two pages; all of these groupings will influence the structural evolution later on.



EXAMPLE 5

These opening two introductory pages act as an upbeat to the first major arrival point, Example 5 (page 3 of the score).

These types are already a concentrated succession of groupings derived from C2G1 treated as a primary group: the three dyads are identical to another primary group, C1G1, from which in fact it is derived, but when expanded with rule 1 they yield several groups not found in the C1G1 expansion. Both groups sets have the same G3 (mn2, mj2, p4)—as well as others, of course—and are emphasized here and later used as the connecting bridge between the two group sets. This is a compositional choice based on an initial sound conception produced by combinations of mn2, mj2, p4.

In the sections that follow, a variety of new groups derived from C2 and C1 types are employed, at times becoming again ambiguous, for instance in the use of C1G11 (two mn3s and one mj3) as a (transposed) regrouping of C1G3 (mn2, mj2, p4), where the pairs of mn3s are made to blur the boundaries between specific types—the ambiguity being used to suggest a less well-defined harmonic progression as an upbeat to arrival points or as motion away from one area to another. Other procedures such as overlappings, suspensions, and anticipations of notes between types, are all employed so that there emerge rich harmonic textures which nevertheless have as their underlying basis the "standard" system types.

The Family II groups used in this piece are seldom used in the piece, and generally emerge only as passing areas or the incidental result of irregular dissolutions or constructions of Family I types; in this context, Family II is in effect treated as a sort of "dissonance". The end of the first movement (Example 6), with its coming to a suspended pause, is an example of this.

The second movement, "Giochi" (Games), is a high-energy development and expansion of the C2G3 group (mn2, mj2, p4)—one of the primary structural groups of the first movement—and extends into groups characterized by the tt, as well as once again using groups that have two or three of the same kind of dyad. For the overall form of the piece, I wanted to create a kind of plateau of fluctuating but generally high energy before the gradual discharge of the last movement. It seemed appropriate to whirl through a large number of closely related and nearly identical types, much the way a free electric source seems to be throwing off a gigantic quantity of charged luminous particles without actually running down.

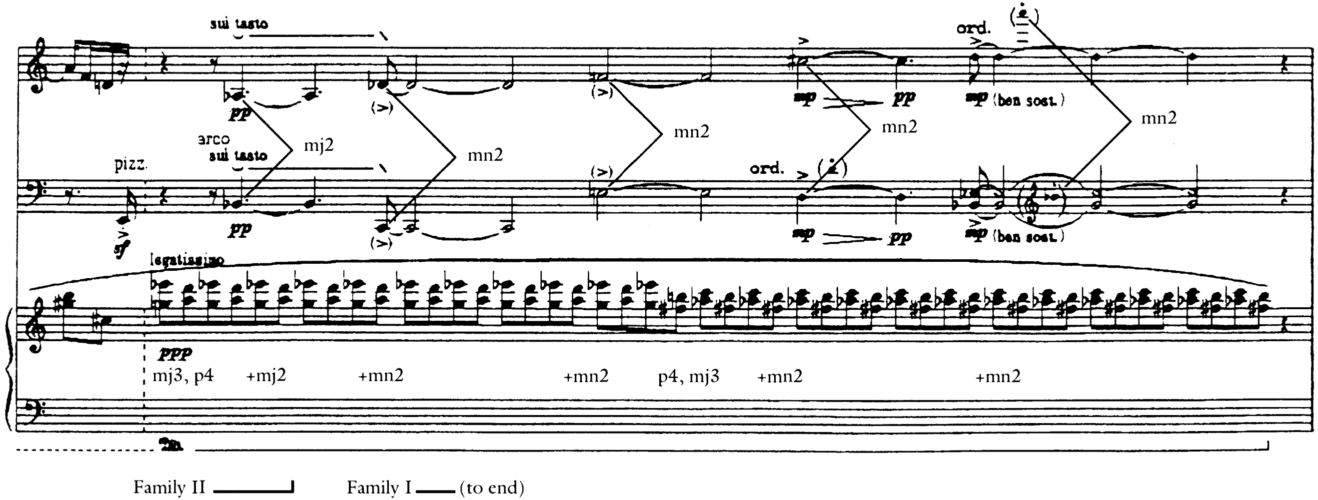
Example 7 shows the beginning of the movement, Example 8 is a fragment from close to the end.

The last movement, "Varianti," picks up other strands from the first movement and expands them in different ways, in contrast to the second movement which is dedicated to the workings out of a single kind of idea. The very beginning is already a variant of things already heard in movement 1, hence the indication "Var. 1" at the start. The first few moments are shown here (Example 9) along with the group succession, starting with C1G4 (mj3, p4, mn2), which is at the same time both an evolution of earlier implications and a ground-plan for coming expansions. The group succession is shown in the example, each group circled with its intervals indicated with their dyad notation, the reduced normal form.

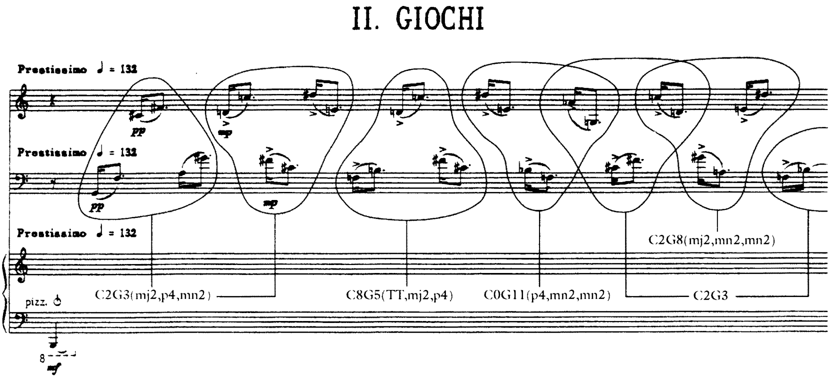
The start of Variant 2 is shown in Example 10 with its widening out of the C1G4 group.

After a great deal of textural change, expressed through a wide variety of group and type successions articulated with characteristically different turnover rates and phrase shapes, the piece moves to a conclusion with Variant 6 (Example 11) that is the movement's most complex intermixing of linear and vertical dyadic articulations; this last variant finally leads to an Epilogue that draws the entire work to a close, Example 12.

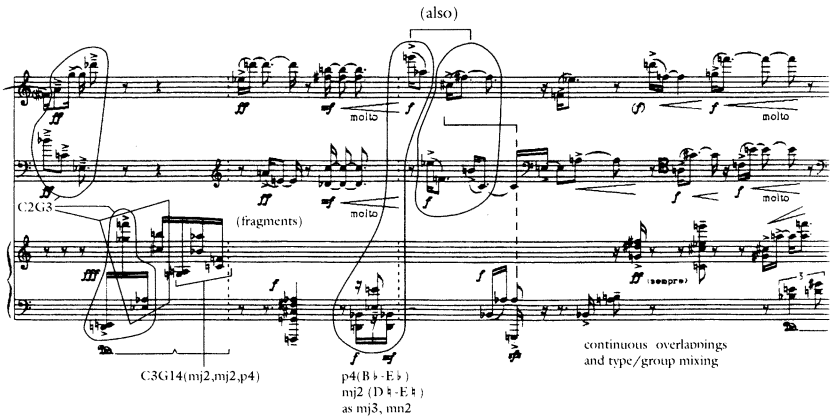
It should be noted that the above merely points out the chosen groups and types, and is by no means an analysis of the piece, any more than


sui tasto
arco
sui tasto
pizz.
pp
pp
sf
(>)
mj2
mn2
mn2
mn2
mn2
ord.
ord.
mp
pp
mp (ben sost.)
legatissimo
ppp
mj3, p4
+mj2
+mn2
+mn2
p4, mj3
+mn2
+mn2
Family II
Family I (to end)

EXAMPLE 6



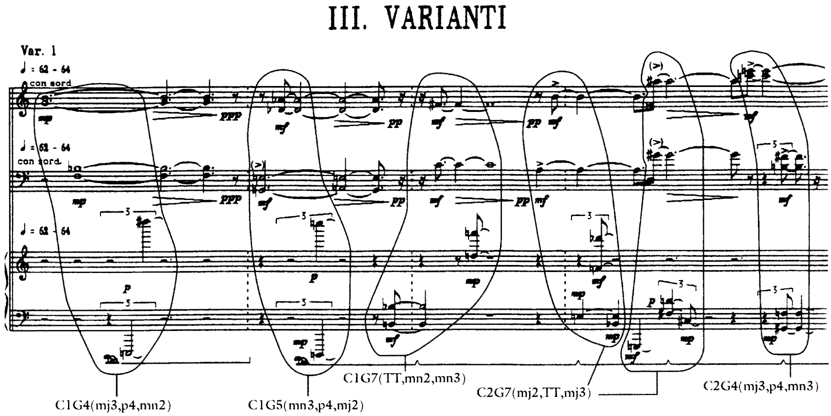
EXAMPLE 7



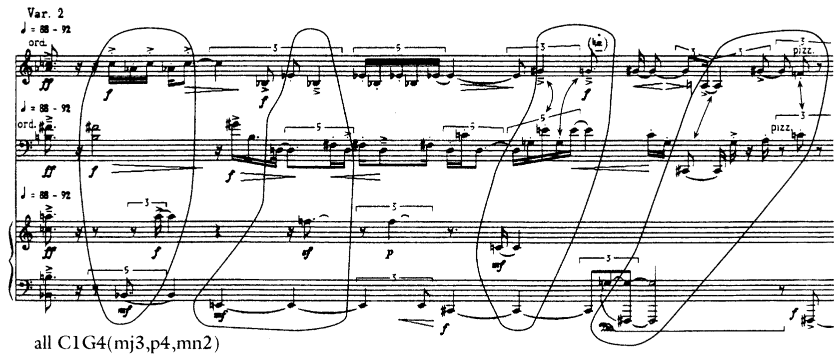
EXAMPLE 8

counting up to twelve or identifying tonics and dominants constitutes an analysis. The choice of specific groups and types, their sound and sense of progression, and how they function on various structural scales are the true concerns of the compositional process and together would be the subject of a fruitful analysis.

For the interaction of electronic sounds with live soloists, the Dyad System provides the source material of the Generating Dyads for the electronic sounds as well as the basis for the soloist's music. The examples



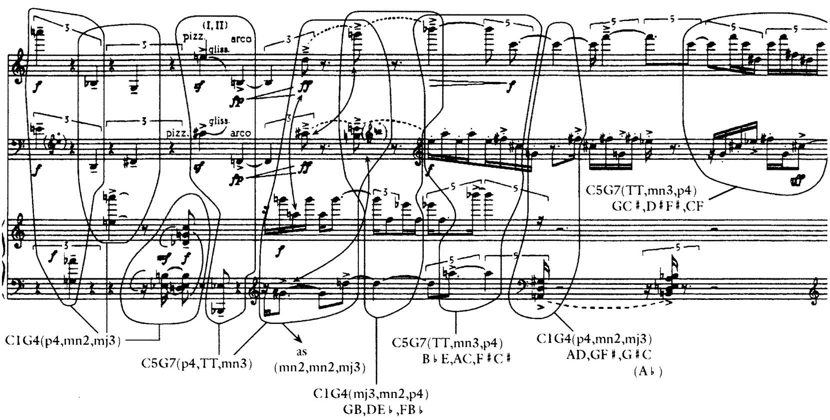
EXAMPLE 9



EXAMPLE 10

shown here do not completely represent the music since notation for the electronic sounds is rather difficult and certainly not informatative from an analytic point of view. I generally just draw simple graphic indications that provide the soloists with an approximation of "what is happening when" in the electronic part. Here I indicate some of the specific details as to how the sounds at particular points were generated.

Example 13 is from section 6 in Part I of my piano and computer piece, *First Tangent to the Given Curve.*[11] Among other things, this piece works with selections of groups from both Family I and Family II collec-



EXAMPLE 11

tions; in fact one of the structural aspects being explored is the "near relations" of groups from the two Families, and their differences in sound, ranging from the subtle to the radical. It should be emphasized again that both the choice of types and (especially) their means of articulation (voicing, orchestration, rhythm, etc.) are absolutely fundamental to the use and the perception of these differences.

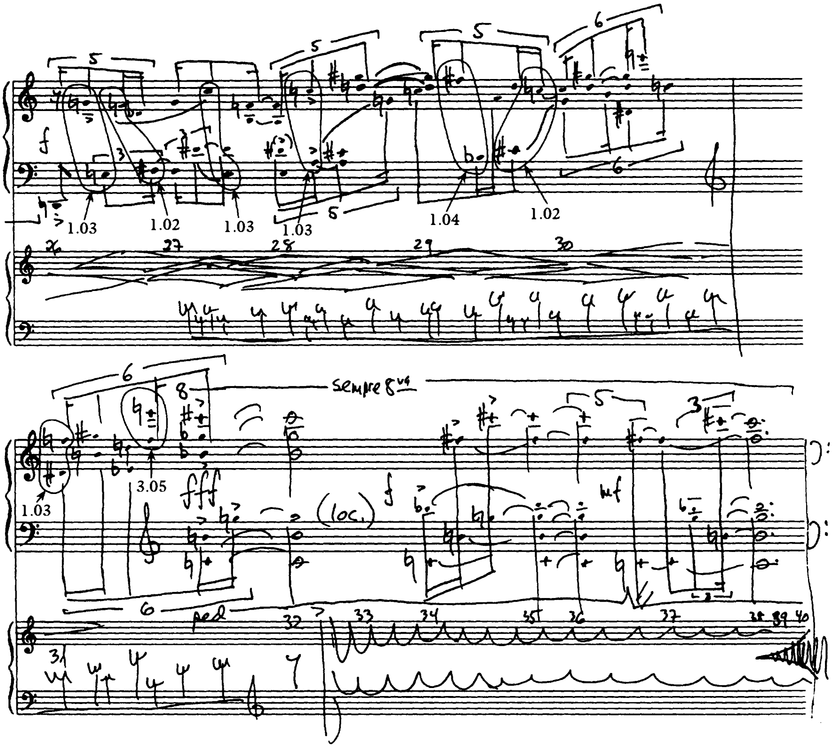
From beat 26 to 31, there is a fairly rapid change of chord-spectra produced by overlapping sounds generated with the second of the additive synthesis techniques discussed in Part II above, where the frequency distance between the generating dyad members is divided by equal proportions (which turns out to be division by equal *interval* differences). The generating dyads are shown circled in the piano part; the numbers 1.03 or 1.02, and so on, mean that the two notes of the generating dyad are heard as the first and third (1.03) or first and second (1.02) frequencies of the sound, where *each* frequency, as determined by the equal proportions algorithm, is played as the lowest frequency partial of a sum of partials; in these cases, all sounds in the passage 26 to 31 were generated by the same complex of partials, arbitrarily called *f*19 (function 19), with the relative amplitudes as shown by the MUSIC30 score data:

```
f19 0    7 1 0    17 .3 0    30 .2 0    39 .1 0    47 .1 0
```

(This is the same syntax as CSOUND except for the lack of the table size parameter, fixed in MUSIC30). The partials are 7, 17, 30, 39, and 47, their relative amplitudes are given by the value following the partial num-

EXAMPLE 12

ber, all beginning at phase 0. For all the sounds in the passage, all frequencies in the equal proportions spectrum are played by partial number 7, hence all frequencies calculations are divided by 7 prior to being generated by the OSCILMX unit generator. And *each* frequency of the basic spectrum will have further frequencies with ratios 2.43, 4.28, 5.57, and 6.71, in short, a rather dense inharmonic sound. The high mj7 is played



EXAMPLE 13

as 3.05, or rather the F is on the third basic frequency, the E on the fifth; hence the fourth frequency of the sound will have a ratio of 1.37395 to the F, or 5.5 semitones above the F, the second frequency will be 5.5 semitones below F and the lowest basic frequency will be a mj7 below F, G♭ (8.06). Again, all of these frequencies are produced with *f*19 and have, therefore, the added timbral densities of the other four partials.

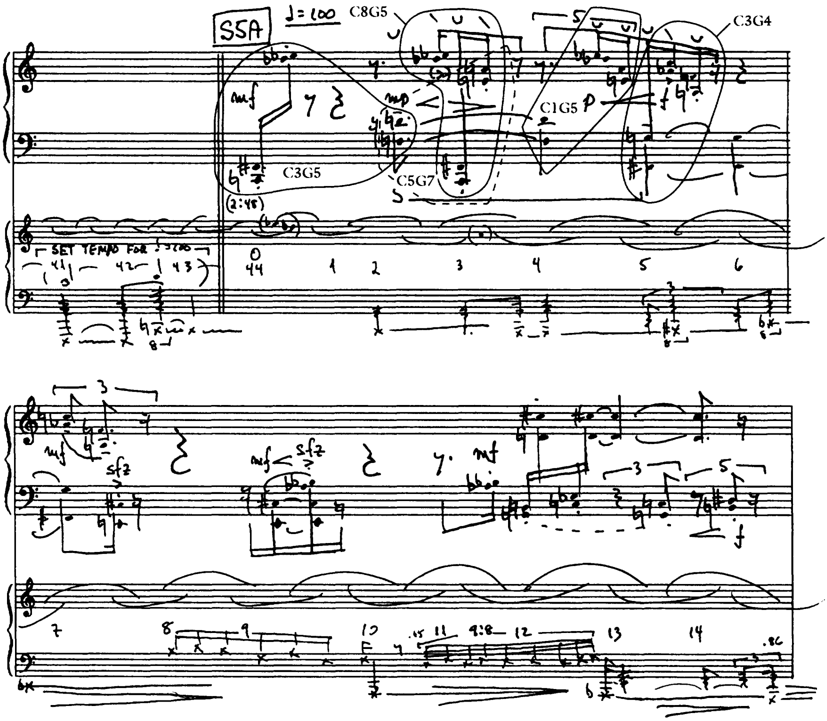
The second part of the Example 13, from beat 32 through beat 40, exploits the differing textural sense obtained by attacking the same notes but in different dyadic distributions: first as a conglomerate at the end of beat 31, then as (mj2, p4, tt), then as (tt, mn3, mj2). The accompanying electronic sounds are a very fast reattacking of each of the pitches in the entire six-note conglomerate, where each attack uses a different function (selection of partials). The effect is a strong complex flickering, a rapid and enriched timbral change in which the dominating piano pitches are embedded.

The next Example (14) is drawn from section 3 of my *Le Tracce di Kronos, i Passi* for clarinet, dancer, and computer.[12] The background electronic sounds, a kind of carpet of chord-spectra, is not shown graphically on the score, since the primary synchronization is between the clarinet soloist and the sequence of short fast attacks, generated by colored noise frequency modulation of a complex partial table. The background chord-spectra are produced by the FM procedure as described in Part II above. Again, the generating dyads are already in the clarinet part, or in the fast sounds in the computer part (N.B. the clarinet part is in B♭, so the real generating dyad pitches circled in the clarinet sound a major second lower), and the result is another kind of inharmonic harmonization of foreground material. The spectra (and hence the intervals that generate them) are designed to have an unobtrusive quality in order to provide a rich but quiet context for the more active computer and clarinet parts.

The pitches used for the generating dyad are circled and the numbers alongside refer to their position in the FM spectrum: –1.03 means that the lower pitch is the first lower sideband, the upper pitch is the third upper sideband; an *R* means that the lower pitch was produced as a reflected (reversed phase) frequency. The rise and overlap times of the sounds (each around seven seconds in duration) are long, roughly 1.75 seconds for the rise time and each sound overlapping the preceding one for approximately three seconds; the general strategy was to anticipate the clarinet pitches in the chord-spectra, so that perceptually the faster figurations seemed to be already "at home" in the sound flux, and, in a sense, emerging from it.

Lastly, Example 15 shows a fragment from section 5A of Part III of *First Tangent to the Given Curve* where the electronic sounds are made to prolong particular piano pitches as frequencies *internal* to complex chord-spectra. The technique is the first additive synthesis method (intervals divided by equal frequency differences) described in Part II above. In these sounds, the specified pitches or dyads are generated not on the lowest frequency of the sum of partial table function, but rather as one of the higher partials in the table. The rapid turnover of the chordal texture creates a foreground event of timbral change, while the ear manages to focus on the constant element, the back-and-forth motion between the two or three constant pitches within the spectral flux, only somewhat later in the sequence; a sense of what might be called timbral counterpoint emerges. A rather wide variety of partial table functions is used, all with different internal ratios between partials. They are listed in the footnotes.[13]

EXAMPLE 14



EXAMPLE 15

The notes being prolonged in this way, E♭ (9.03), D♭ (9.01), and to a lesser extent A (8.09), are shown in parentheses in the computer part. At this point in the musical structure, there is a foreground sense of Family II groups (circled in Example 14) with Family I groups (enclosed in rectangles) as a kind of shadow in the background: Family II types: C3G5 (tt, mj2, mn3); C8G5 (mj2, tt, p4); C3G4 (mn2, mj3, tt) with the shadow Family I types: C1G5 (mn3, mj2, p4) and C5G7 (mn3, tt, p4).

As I was developing this paper, I was fortunate enough to have been busy with a commission from the Koussevitzky Foundation. The result was *Far Sounds, Broken Cries* for twelve instruments and quadraphonic electronic sounds.

This piece explores primary collections of eight pitch classes, and in the process I did some preliminary work for developing primary collections of seven pitch classes, which became the basis for my next piece, *Sul Filo dei Tramonti* for soprano, piano and electronic sounds.

The electronic sounds for *Far Sounds, Broken Cries* were almost exclusively generated by the additive synthesis "ADD, MULTIPLY" kind of algorithm; the sounds were then considered source signals for such procedures as granular synthesis, convolution, time-stretching and time-compressing, complex delay and feedback processes, filtering, and so on. The "almost exclusively" in the above sentence refers to the fact that I used a single cello sound, a short loud low C, as the source for a great deal of signal processing; and two very rich natural noise sources, a sixteen-inch cymbal stroke and the sea, as second signals for convolution. Through the various electronic processes, the auxiliary acoustic signals refocussed certain characteristics of the synthetic sounds, and these fresh timbral transformations added rich detail to the interaction between the live ensemble and the electronic sounds. These transformations, however, were not allowed to obscure the Generating Dyad structure which initially produced the electronic sounds; on the contrary, I found I could design a more elaborate timbral development, which, in accordance with some of the concepts favored here, served to enrich and intensify the prolongation of the Dyad System–derived structure.

Far Sounds, Broken Cries takes advantage of the much larger number of groups derivable from any eight-pitch-class primary collection than from hexachordal primary collections, and the correspondingly greater number of types. A primary collection (C, D♭, D, F, E, G♭, E♭, G), chosen for the fact that it may be immediately grouped into four different dyads, (mn2, mn3, mj2, mj3), when operated on by rule 1, will generate 105 groups. Some of these groups have four different dyads, some have two the same and two different, still others have three the same and a fourth different, and three groupings where all four dyads.

What follows is a selection of some of the groups from this primary collection (the first group is labelled D1G1 and will be the source for later examples):

mn2	*mn3*	*mj2*	*mj3*					
C,D♭	D,F	E,G♭	E♭,G	mn2	mn3	mj2	mj3	**←D1G1**
C,D♭	D,F	E,E♭	G♭,G	mn2	mn3	mn2	mn2	
C,D	D♭,E♭	F,G	E,G♭	mj2	mj2	mj2	mj2	
C,D	D♭,G	F,E	G♭,E♭	mj2	tt	mn2	mn3	
C,F	D♭,G	D,G♭	E,E♭	p4	tt	mj3	mn2	
C,F	D♭,G	D,E♭	E,G♭	p4	tt	mn2	mj2	

mn2	*mn3*	*mj2*	*mj3*				
C,E	D♭,D	F,G♭	E♭,G	mj3	mn2	mn2	mj3
C,E	D♭,D	F,E♭	G♭,G	mj3	mn2	mj2	mn2
C,G♭	D♭,E♭	D,G	F,E	tt	mj2	p4	mn2
C,G♭	D♭,G	D,F	E,E♭	tt	tt	mn3	mn2

By applying Rule 2, each group yields 102 (or fewer) types, but there is a great deal of redundancy: most of the generated types are regroupings of the pitch classes of another type. If we elaborate D1G1 we obtain a full complement of 102 types, but only 22 *different* types in the entire type set. Here are three generated types (type 1, type 12, type 30, the highlighted collection at the top of each list) with their equivalent types generated later in the set. This kind of regrouping among *types* (where "both" the dyad content *and* the pitch class content are invariant—only the specific pitch classes constituting the dyads are changed) was a very useful resource in *Far Sounds, Broken Cries.*

Where in the hexachord collections we could regroup the pitch classes into pairs of trichords, with an eight pitch-class collection expanded opportunities in this regard present themselves: the various groups and

mn2		*mn3*		*mj2*		*mj3*			
C	**D♭**	**D**	**F**	**E**	**G♭**	**E♭**	**G**	**type1**	**trans0**
D♭	D	E♭	G♭	F	G	C	E	type3	trans1
C	D♭	E	G	E♭	F	D	G♭	type26	trans0
G♭	G	C	E♭	D	E	D♭	F	type64	trans6
G♭	G	D	F	D♭	E♭	C	E	type89	trans6
F	G♭	D♭	E	C	D	E♭	G	type90	trans5
C	**D♭**	**D**	**F**	**A♭**	**B♭**	**B**	**E♭**	**type12**	**trans0**
B♭	B	D	F	D♭	E♭	A♭	C	type28	trans10
D♭	D	F	A♭	B♭	C	B	E♭	type37	trans1
B	C	F	A♭	D♭	E♭	B♭	D	type56	trans11
D	E♭	A♭	B	B♭	C	D♭	F	type65	trans2
C	D♭	A♭	B	E♭	F	B♭	D	type81	trans0

mn2	*mn3*	*mj2*	*mj3*		
C Db	**E G**	**Gb Ab**	**Bb D**	**type30**	**trans0**
Gb G	Bb Db	C D	E Ab	type30	trans6
C Db	E G	Ab Bb	D Gb	type32	trans0
Gb G	Bb Db	D E	Ab C	type32	trans6
Db D	G Bb	E Gb	Ab C	type58	trans1
G Ab	Db E	Bb C	D Gb	type58	trans7
Db D	G Bb	Gb Ab	C E	type62	trans1
G Ab	Db E	C D	Gb Bb	type62	trans7

types can be re-partitioned into two trichords and one dyad, which may in turn form different internal patterns: {two trichords, one dyad}; {one dyad, two trichords}; {one trichord, one dyad, one trichord}.

We might proceed as follows: first generate the type set as above; then apply one or another of the groupings with trichords to the type set, then create the musical structure via trichord (or trichord + dyad, or trichord pairs, etc.) relationships inside the type set or between type sets. From the standpoint of the electronic dimension, a very small number of "octachords" can be made to generate a tremendous variety of electronic sounds.

The three sets of examples that follow are drawn from the same group, D1G1, the type names shown are those from the initial D1G1 four dyad grouping. The first three types in each example are the same (1, 5 and 18), in order to observe the intervallic differences that emerge for a single type.

as trichord, trichord, dyad:

C Db D mj2(mn2, mn2)	F E Gb mj2(mn2, mn2)	Eb G mj3	type1	trans0
C Db D mj2(mn2, mn2)	F Gb Ab mn3(mn2, mj2)	G B mj3	type5	trans0
C Db Eb mn3(mn2, mj2)	Gb F G mj2(mn2, mn2)	E Ab mj3	type18	trans0
C Db F p4(mn2, mj3)	Ab D E tt(mj2, mj3)	Eb G mj3	type38	trans0
C Db G tt(p4, mn2)	Bb Eb F p4(p4, mj2)	D Gb mj3	type68	trans0

C D♭ B mj2(mn2, mn2)	D E♭ F mn3(mn2, mj2)	E A♭ mj3	type91	trans0

as dyad, trichord, trichord:

C D♭ mn2	D F E mn3(mj2, mn2)	G♭ E♭ G mj3(mn3, mn2)	type1	trans0
C D♭ mn2	D F G♭ mj3(mn3, mn2)	A♭ G B mj3(mn2,mn3)	type5	trans0
C D♭ mn2	E♭ G♭ F mn3(mj2, mn2)	G E A♭ mj3(mn3, mn2)	type18	trans0
C D♭ mn2	F A♭ D tt(mn3, mn3)	E B E♭ p4(mj3, mn2)	type41	trans0
C D♭ mn2	G♭ A D p4(mn3, mj3)	E E♭ G mj3(mn2,mn3)	type54	trans0
C D♭ mn2	G B♭ G♭ mj3(mn2, mn3)	A♭ B E♭ p4(mj3,mn3)	type73	trans0

as trichord, dyad, trichord:

C D♭ D mj2(mn2, mn2)	F E mn2	G♭ E♭ G mj3(mn3,mn2)	type1	trans0
C D♭ D mj2(mn2, mn2)	F G♭ mn2	A♭ G B mj3(mn2,mn3)	type5	trans0
C D♭ E♭ mn3(mn2, mj2)	G♭ F mn2	G E A♭ mj3(mn3, mn2)	type18	trans0
C D♭ F p4(mn2, mj3)	A♭ D tt	E E♭ G mj3(mn2, mn3)	type38	trans0
C D♭ G♭ tt(mn2, p4)	A F mj3	G B♭ D p4(mj3,mn3)	type61	trans0
C D♭ A♭ p4(mj3, mn2)	B D mn3	E G♭ B♭ tt(mj2, mj3)	type78	trans0
C D♭ B mj2(mn2, mn2)	D A♭ tt	B♭ E♭ G p4(mn3, mj3)	type101	trans0

If we apply the rule 1 procedure to the 8 pitch class primary collection *initially* subdivided into, for example, {dyad, trichord, trichord}, we wind up with 280 groups. Each of these groups generates, with rule 2, a different number of types, as many as thirty, as few as seven. A representative selection of these groups derived from the primary collection (C, Db, D, Eb, E, F, Gb, Ab) follows:

1.	C Db	D Eb E	F Gb Ab	mn2	mj2(mn2,mn2)	mn3(mn2,mj2)
2.	C Db	D Eb F	E Gb Ab	mn2	mn3(mn2,mj2)	mj3(mj2,mj2)
3.	C Db	D Eb Gb	E F Ab	mn2	mj3(mn2,mn3)	mj3(mn2,mn3)
4.	C Db	D Eb Ab	E F Gb	mn2	tt(mn2,p4)	mj2(mn2,mn2)
19.	C D	Db F Ab	Eb E Gb	mj2	p4(mn3,mj3)	mn3(mn2,mj2)
20.	C D	Db Gb Ab	Eb E F	mj2	p4(p4,mj2)	mj2(mn2,mn2)
21.	C Eb	Db D E	F Gb Ab	mn3	mn3(mn2,mj2)	mn3(mn2,mj2)
22.	C Eb	Db D F	E Gb Ab	mn3	mj3(mn2,mn3)	mj3(mj2,mj2)
50.	C F	Db Gb Ab	D Eb E	p4	p4(p4,mj2)	mj2(mn2,mn2)
51.	C Gb	Db D Eb	E F Ab	tt	mj2(mn2,mn2)	mj3(mn2,mn3)
69.	C Ab	Db E Gb	D Eb F	mj3	p4(mn3, mj2)	mn3(mn2, mj2)
70.	C Ab	Db F Gb	D Eb E	mj3	p4(mj3, mn2)	mj2(mn2, mn2)
71.	Db D	C Eb E	F Gb Ab	mn2	mj3(mn3, mn2)	mn3(mn2, mj2)
72.	Db D	C Eb F	E Gb Ab	mn2	p4(mn3, mj2)	mj3(mj2, mj2)
209.	Eb Gb	C E Ab	Db D F	mn3	mj3(mj3, mj3)	mj3(mn2, mn3)
210.	Eb Gb	C F Ab	Db D E	mn3	p4(mj3, mn3)	mn3(mn2, mj2)
211.	Eb Ab	C Db D	E F Gb	p4	mj2(mn2, mn2)	mj2(mn2, mn2)
212.	Eb Ab	C Db E	D F Gb	p4	mj3(mn2, mn3)	mj3(mn3, mn2)

The internal variety of pitch class associations with respect to the specific dyads for a single group set derived in this way is, needless to say, rather immense. One can easily compose a work of rather large dimensions based entirely on the elaboration of just one or two of these groups (the Prologue to my planetarium opera, ARCHIMEDES, does exactly that).

In passing, it should be observed that the groups and types generated from *this* primary collection and those from the earlier eight pitch class primary collection exhibit the same kind of Family I, Family II relationship as noted with respect to hexachordal primary collections. There is no Family I type that is identical to a Family II type.

Finally, primary collections of seven pitch classes can be subjected to similar operations. Still other dyad-trichord associations emerge and suggest further expressive and structural possibilities. Here are a few of the 105 groups generated by rule 1 applied to the pitch class collection (C, Db, Eb, E, F, Gb,G) subdivided into {trichord, dyad, dyad}:

1.	C	D♭	E♭	E,F	G♭,G	mn3(mn2,mj2)	mn2	mn2
2.	C	D♭	E♭	E,G♭	F,G	mn3(mn2,mj2)	mj2	mj2
18.	C	E♭	E	D♭,G	F,G♭	mj3(mn3,mn2)	tt	mn2
19.	C	E♭	F	D♭,E	G♭,G	p4(mn3,mj2)	mn3	mn2
42.	C	F	G	D♭,G♭	E♭,E	p4(p4,mj2)	p4	mn2
43.	C	G♭	G	D♭,E♭	E,F	tt(mn2,p4)	mj2	mn2
60.	D♭	E	F	C,G	E♭,G♭	mj3(mn3,mn2)	p4	mn3
61.	D♭	E	G♭	C,E♭	F,G	p4(mn3,mj2)	mn3	mj2
75.	D♭	G♭	G	C,F	E♭,E	tt(p4,mn2)	p4	mn2
76.	E♭	E	F	C,D♭	G♭,G	mj2(mn2,mn2)	mn2	mn2
102.	E	G♭	G	C,F	D♭,E♭	mn3(mj2,mn2)	p4	mj2
103.	F	G♭	G	C,D♭	E♭,E	mj2(mn2,mn2)	mn2	mn2

Immediately after *Far Sounds, Broken Cries*, I completed *Sul Filo dei Tramonti*, for soprano, piano and electronic sounds (on poems by Gian Giacomo Menon) which uses exclusively seven pitch-class types derived from four different groups. Manipulations of the basic two dyad, one trichord structure yields a rich variety of pitch collections and, especially, of electronic sounds.

In all of these examples, it is hoped that the reader can get a sense of how these groupings (two or three, several, many, however many as would be required by a musical design) might *sound* when used in a musical work, how they might create different kinds of flow, texture and expressive structural distinctions, both by themselves in a purely instrumental composition as well as in terms of the electronic sounds they can be made to produce.

Notes

1. John M. Chowning, "The Synthesis of Complex Audio Spectra by Means of Frequency Modulation," *Journal of the Audio Engineering Society* 21, no.7 (September 1973): 526–34.

2. James Dashow. "Three Methods for the Digital Synthesis of Chordal Structures with Non-Harmonic Paritals," *Interface* 7 (1978): 69–94; James Dashow, "Spectra as Chords," *Computer Music Journal* 4, no. 1 (Spring, 1980): 43–52; James Dashow, "New Approaches to Digital Sound Synthesis and Transformation" *Computer Music Journal* 10, no. 4 (Winter, 1986); James Dashow, "Looking Into Sequence Symbols," *Perspectives of New Music* 25, nos. 1 & 2 (Winter and Summer, 1987): 108–137.

3. In the original implementation of RM Complex, this method was limited to integer multiples of the input frequency in order to use one interpolating oscillator (OSCILI) to read a single table consisting of two partials $N1$ and $N2$ that were the integers for the multiplication, i.e., the oscillator output was $InputHz^* N1 + InputHz^* N2$. This represented an optimized computation of the algorithm for the older generation of computer hardware available. In the most recent configuration of RM Complex using the author's MUSIC30 digital sound synthesis language for the T.I. TMS320C30 dsp chip, the single OSCILI is replaced by two resonating filter unit generators (FLTOSC) that are not only faster (even taken in pairs) than the OSCILI but produce a still cleaner signal. The independence of the two signals means that the components of each signal are no longer limited to *integer* multiples of the input frequency but can be any value whatever. The potential for gradual timbral transformation is considerably enhanced thereby.

   See also footnote 6.

4. i.e., output = $A^*(X+M)+B^*(Y+N) + ((A^*(X+M)) * (B^*(Y+N)))$

5. OSCILM (and its variants OSCILMX, OSCILMV) is a compound unit generator that produces the sum of from 2 to 12 interpolating oscillators per call using an internal looping structure. Thus, besides the significant increase in execution efficiency—real time in some MUSIC30 applications—it is possible to conceive of, control and manipulate the complex result of additive synthesis as produced by an OSCILM as a single sound object rather than a complicated synchronization of individual units. For each oscillator in the loop

(except the first which provides the starting frequency), OSCILM automatically adds a (performance time) frequency value to that of the previous oscillator in the loop; OSCILMX automatically *multiplies* the frequency value of the previous oscillator by a performance time value; in OSCILMV the current oscillator (*N*) frequency is obtained by multiplying the starting frequency value (for the first oscillator in the loop) by the *N*th value in an array of values that can be calculated in any way desired.

For a complete description of the MUSIC30 language (written for the Sonitech International accelerator board for IBM-compatible personal computers), see *Computer Music Journal* 19, no. 3 (Fall 1995): 83–85 (product review of MUSIC30), and *Computer Music Journal* 18, no. 4 (Winter 1994): 113–18 (user's report on the Sonitech Spirit30 accelerator board), both by the author of this paper.

6. Typical MUSIC30—and related languages—code for the algorithm would look something like the following:

```
*
*  parameter fields: p5=HZU, p6=HZL, p7=NU, p8=NL, p9=number of oscillators
*     p10 = partial number to generate Gen Dyad pitches, p11 = function table number
*
*     initialize as many arguments as possible!
*
hzusi     isipch    p5     ; convert pitches from 8.pc to sampling increment
hzlsi     isipch    p6
*  in MUSIC30, [i]divtab is generated at setup time and contains
*     1/N for N = 2 to (default maximum of) 80. This makes integer division
*     practical for perf-time necessities.
invpn     idivtab   p10    ; invpn = 1/partial number, now a multiply factor
*
addhzsi   idivide   hzu- hzl, p7 - p8     ; HZU-HZL/NU-NL
basesi    ival      hzlsi - (p8 - 1)*addhzsi
*
addhzsi   ival      addhzsi*invpn         ; now adjusted for partial number
basesi    ival      basesi*invpn
*     perf time call to oscilm does all the work
outsig    oscilm    basesi, addhzsi, p9, p11,0       ; phase init to 0
*     etc.
```

The amplitude of the outsig is adjusted "manually" outside the oscilm unit generator call. Similar procedures are used for oscilmx and oscilmv.

It should be noted that when generating the look-up tables for the oscillators with arbitrarily chosen partial numbers, it is not necessary to generate the "fundamental", i.e., the first partial. The table-generating routines produce ***any*** combination of partials, the only restriction being the table size (i.e., how much data the table can hold).

7. The notion of the harmonic "stretch factor" is due to Steve McAdams. I don't recall in which of his many highly suggestive articles McAdams first proposed this concept, but I do remember a long and fruitful conversation with him several years ago in Rome during which we discussed various aspects and potential compositional uses of several of his original ideas.

8. Functions used for the additive synthesis in Example 14

f13	0	9	5	1	0	7	.35	0	12	.25	0	19	−.25	0	31	.25	0	43	.3	0
f15	0	9	3	1	0	5	.35	0	8	.25	0	13	−.25	0	21	.25	0	35	.3	0
f16	0	9	6	1	0	11	.4	0	17	.3	0									
f17	0	9	6	1	0	13	.4	0	21	.3	0									
f18	0	9	7	1	0	15	.6	0	26	−.4	0	47	.35	0						
f19	0	9	7	1	0	17	.6	0	30	−.4	0	39	.35	0						
f21	0	9	7	1	0	10	.3	0	17	−.3	0	24	.4	0						
f23	0	9	11	1	0	18	.5	0	25	.4	0									
f26	0	9	9	1	0	16	.4	0	26	.25	0									
f27	0	9	7	1	0	11	.4	0	19	.25	0	30	.15	0						
f29	0	9	6	1	0	11	.4	0	29	.25	0									
f30	0	9	7	1	0	15	.4	0	26	.25	0									
f31	0	9	9	1	0	15	.4	0	22	.25	0									

9. A list of MULTIPLY, ADD and ADD, MULTIPLY algorithms so far developed (Spring 2000) for additive synthesis follows. Available at the author's website (www.jamesdashow.net) are the C routines for PC that allow the composer to see on-screen the list of frequency components of any spectrum realized with any Generating Dyad the C source code for these routines that may be compiled under other operating systems, and the version for MUSIC30 (as .OBJ and source code) that implements the real-time or deferred-time synthesis of these sounds.

MULTIPLY then ADD

fN = *f0* * *M*^ *N* + *A* * *N* -> no restrictions
fN = *f*(*N*-1) * *M*^ *N* + *A* * *N* -> ONLY in given *M* find *A* mode

fN = *f0* * *N*^ *M* + *A* * *N* -> Stretch Factor proc no restrictions
fN = *f*(*N*-1) * *N*^ *M* + *A* * *N* -> limited to sbu = sbl + 1 ALWAYS,

$fN = f0 * M^N + A * N^M$ -> ONLY in given M find A mode

the following work correctly ONLY with sbl > 1 and xx & $yy > 0$; xx and yy can be fractions.

$fN = f0 * M^N + A * X$ -> $X = N+xx/N+yy$
$fN = f0 * M^N + A * N^X$ -> $X = N+xx/N+yy$

ADD then MULTIPLY

$fN = M^N * (f0 + A * N)$ -> in given A find M, sbl can ONLY be 1 ($f0$ always LP)
$fN = M^N * (f(N\text{-}1) + A * N)$ -> limited to sbu = sbl + 1 ALWAYS, i.e. 2.03 3.04 4.05 etc.

$fN = N^M * (f0 + A * N)$ -> in given A find M, sbl can ONLY be 1 ($f0$ always LP)
$fN = N^M * (f(N - 1) + A * N)$ -> limited to sbu = sbl + 1 ALWAYS, i.e. 2.03 3.04 4.05 etc.
$fN = f(N - 1) * (N^M + A * N)$ -> limited to sbu = sbl + 1 ALWAYS, i.e. 2.03 3.04 4.05 etc.

$fN = f0 * (N^M + A * N)$ -> no restrictions

$fN = f0 * (M + A * N)$ -> no restrictions

$fN = f0 * (M^N + A * N)$
$fN = f(N\text{-}1) * (M + A * N)$ -> this is limited to sbu = sbl + 1
i.e. 2.03 3.04 4.05 etc.

$fN = M * f(N - 1)^2 / f(N - 2)$ -> M is a true interval compander
The intervals between successive $f(N)$'s expand ($M > 1$) or contract ($M < 1$)

$fN = M^N * (f0 + A * N^M)$ -> ONLY in given M find A mode

10. Copyright 1994 Edition Pro Nova, Sonoton Musikverlag-Musikproduktion, Munich; recorded on ProViva compact disc, ISPV 177 CD.

11. Copyright 1997, Via Veneto Edizioni Musicali, Roma, recorded on a Capstone Records CD CPS 8645, Daniele Roi, pianist.

12. Copyright 1996, Via Veneto Edizioni Musicali, Roma, recorded on vol. 24 of the CDCM Computer Music Series on Centaur CD, CRC 2310, Esther Lamneck, clarinettist. Also on Scarlatti Classica CD MZQ737707 (Roma), William O. Smith, clarinettist.

13. See note 8.

Discography

(Details available at www.jamesdashow.net)

Wergo CD WER 2018-50: James Dashow. *Archimedes and other works. Archimedes*, Scene 2 (first version).
Oro, Argento & Legno, for flute and computer (Marzio Conti, flutes).
Mnemonics, for violin and computer (Mario Buffa, violin).

ProViva ISPV 177 CD: James Dashow, Selected Works *Punti di Vista no. 1*, piano solo (Daniele Roi, piano). *Reconstructions*, for harp and computer. Lucia Bova, harp. *4/3-Trio*, for violin, cello, and piano (M. Wu, violin, L. Rath, cello, S. Kahan, piano). *Mappings*, for cello and electronic sounds (Luca Paccagnella, cello).

Capstone CPS-8659: *James Dashow Music for Small Ensembles. A Sheaf of Times*, septet (Helix! New Music Ensemble, Paul Hoffmann, director).
Oro, Argento & Legno, for flute and computer (Manuel Zurria, flutes).
Ashbery Setting, for soprano, flute and piano (New York New Music Ensemble: Lisa Pierce, soprano, Jayn Rosenfeld, flute, James Winn, piano).

Neuma CD 450-90: *Morfologie*, for trumpet player and computer (Mauro Maur, trumpets). *Punti di Vista no. 2*, for piano solo (Sandra Sprecher, piano). *Reconstructions*, for harp and computer (Lucia Bova, harp).

Capstone CPS-8645: *Songs from a Spiral Tree*, for mezzo-soprano, flute and harp (Constance Beavon, mezzo-soprano, Lauren Weiss, flutes, Lucia Bova, harp).
First Tangent to the Given Curve, for piano and computer (Daniele Roi, piano).

Pan Records PRC S20-12: *Conditional Assemblies*, computer music (stereo mixdown of quad original). *Partial Distances*, (analogue) electronic music. *Effetti Collaterali*, for clarinet in A and computer (Phillip Rehfeldt, clarinet).

Composers' Recordings, Inc. (CRI) LP SD-456: *Second Voyage*, for tenor and digitally synthesized electronic sounds (George Shirley, tenor) digitally remastered and reissued on Capstone CPS-8669.

Scarlatti Classica CD MZQ737707 (released in February, 2000): *James Dashow—Media Survival Kit ed altri percorsi. Le Tracce di Kronos, i passi, per clarinetto, danzatore e computer* (William O. Smith, clarinet).

Disclosures, per violoncello e computer (Francesco Dillon, cello). *In Winter Shine*, computer music (revised and digitally remastered). *Media Survival Kit*, a lyric satire for radio broadcast for voices, instruments and electronic sounds.

Centaur CD CRC 2310. *Le Tracce di Kronos, i Passi* (E. Lamneck, clarinet).

Associazione Culturale Secondo Maggio (Milano) CD SM001 *Le Tracce di Kronos, I Passi* (P. Giannetti, clarinet).

Neuma 450-75. *Disclosures* (Luca Paccagnella, cello).

RCA CD 74321 16825-2. *Morfologie* (Mauro Maur, trumpet).

ORF (Österreichischer Rundfunk) CD: Prix Ars Electronica 96. *Media Survival Kit.*

Edi-Pan LP LM006. *In Winter Shine* (original version, stereo mixdown).

MIT CD 25th anniversary of Experimental Studio/MediaLab, *In Winter Shine* (original version, stereo mixdown).

Cultures Electroniques/6 (Bourges). CD LDC 278055. *Whispers Out of Time.*

RCA CD CCD 3002. *Some Dream Songs*, for soprano, violin and piano (on poems by John Barryman) (J. Logue, soprano, M. Buffa, violin, G. Simonacci, piano).

CRI CD 578 *Some Dream Songs* (J. Logue, M. Buffa, G. Simonacci).

Cultures Electroniques CD (Bourges, in preparation) . . . at other times, the distances, quadraphonic electronic music (stereo mixdown).