Proportional Recursive Stochastic Composition Using COMP2, a Smalltalk-80 Composition Program Within the Kyma Digital Synthesis System

Brian Belet
Assistant Professor of Music
Department of Visual and Performing Arts
Clark University
950 Main Street
Worcester, Massachusetts 01610-1477, USA
e-mail: bbelet@vax.clarku.edu

Abstract

COMP2 is a proportionally governed recursive stochastic composition program written by the composer in 1991 using the Smalltalk-80 programming language within the Kyma digital synthesis system. COMP2 is designed to compose music in which all aspects of the composition are unified aesthetically and procedurally. This is accomplished through the use of a composer-defined set of initial values which directly or indirectly determine all of the required variable values through recursive stochastic procedures.

Like its 1985 Fortran predecessor COMP1, COMP2 uses twelve ratios as its generating "Ur set". These ratios correspond to twelve primary interval ratios (or proportions) of Just Intonation, and are used here as an aesthetic dedication to Ben Johnston (although any number of composer-defined values can be used). Two compositions were generated with COMP1, and to date one new work is currently being refined with COMP2 (for flute and Kyma system, with early 1992 performances scheduled). The resulting proportional recursive stochastic compositions utilize pseudo-curved aural space in several parameters to insure that parameter boundaries are neither exceeded nor actually reached. Not specifically microtonal, these works are micro-r-al where "r" represents every parameter and every relationship in the work. COMP2, operating on the Kyma system, permits real-time software synthesis in an intuitive, composer-friendly, yet extremely powerful desktop environment, which is a significant improvement over COMP1.

Introduction

It has been and remains this composer’s interest to aesthetically and procedurally unify the diverse aspects of human compositional and computer processes within a given musical work. A series of related compositions have been generated using algorithmic programming and software synthesis to explore the possibilities of creating resultant unified compositional complexity from initial aesthetic simplicity ("simplicity" is used here as "elegance"). This relates conceptually to the various explorations and experiments in astrophysics that seek to discover one or more Grand Unification Theories (GUTs) that will...
unify the four basic forces of nature (the strong nuclear force, electromagnetism, the weak nuclear force, and gravity; listed in decreasing order of strength) and all of the physical universe that is governed by these forces, into one universal "U-force" at the moment of creation.

Proportions have been selected as the central unifying programming factor for these compositions at ratios, and ratio scales, offer the most precision for generating, and thereby measuring, relationships in ordered (scalar) systems. In contrast to simpler nominal, ordinal, and interval scales, ratio scales permit an infinite number of points and relationships between any two established scale points (S. S. Stevens. "Mathematics, Measurement, and Psychophysics." *Handbook of Experimental Psychology.* 1951).

**COMP1**

The program **COMP1** was written by this composer in 1985 (revised in 1989) using Fortran to generate score data using proportionally guided, pseudo-recursive stochastic procedures. Two compositions were generated with this program and realized using Music 360 at the University of Illinois' Computer Music Project: *At Last* (SCIS 1a), for twelve live instruments arranged into four spatially separated trion, percussion, and computer tape; and *As Long As We Are Here* (SCIS 1b), for computer tape alone. The computer instruments included two types of additive synthesis (with proportionally determined added partials), AM, and FM (each with proportionally determined f1/fm and dynamic MI). These instruments stochastically called stored sine and just-derived audio wave tables for their required fundamental, added partial, carrier, and modulator signals.

While **COMP1** served as an initial exploration into the unification of the composition process and the resulting sound, there were several limitations presented by both the programming and the synthesis languages. Fortran is not inherently recursive, so extensive duplicate programming blocks were required to simulate recursive processes. Music 360, like many software synthesis languages before and since, separates the score and the orchestra in distinct, and therefore separate, conceptual operations. Events are inherently treated as notes and phrases, which affects all aspects, both aesthetic and procedural, of the resulting composition. Music 360 also does not permit real-time sound processing, which dramatically impedes the process of experimenting with raw sounds.

**COMP2**

**COMP2** was written during 1991 using the Smalltalk-80 object-oriented programming language within the Kyma digital synthesis system. Both the language and the system carry very little inherent aesthetic prejudice, which leaves the composer quite free to impart his or her own aesthetic concerns and algorithmic procedures throughout the compositional process. The dedicated DSP hardware component (CPyba) permits both real-time and sample-to disk software synthesis with CD-quality ADC and DAC. Operating on a host Macintosh workstation, the software component (Kyma) permits an intuitive continuum of approaches to constructing and controlling sound structures. There is no distinction between score and orchestra as all structures and procedures are treated as "Sounds", and an unlimited number of nested Sounds may be used. Locos are used to set up dynamic "patches" which can include Scales, Language Sounds, each of which contains a Smalltalk program.

Like its predecessor **COMP1**, **COMP2** is designed using only generative (additive) procedures so that any of the exact amounts of final data is generated, in contrast to slave (subtractive) procedures. The "U-force" material consists of twelve Just Intonation ratios, each manipulated to permit both increasing and decreasing change in variable values.
<table>
<thead>
<tr>
<th>Just Interval</th>
<th>Ratio</th>
<th>Value</th>
<th>L/Ratio</th>
<th>L/Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>M7</td>
<td>15.8</td>
<td>1.875</td>
<td>8.15</td>
<td>0.5335</td>
</tr>
<tr>
<td>m7</td>
<td>7.4</td>
<td>1.75</td>
<td>4.7</td>
<td>0.57</td>
</tr>
<tr>
<td>M6</td>
<td>13.8</td>
<td>1.625</td>
<td>8.13</td>
<td>0.615</td>
</tr>
<tr>
<td>m6</td>
<td>8.5</td>
<td>1.6</td>
<td>5.8</td>
<td>0.625</td>
</tr>
<tr>
<td>P5</td>
<td>3.2</td>
<td>1.5</td>
<td>2.3</td>
<td>0.666</td>
</tr>
<tr>
<td>A4</td>
<td>11.8</td>
<td>1.375</td>
<td>8.11</td>
<td>0.7272</td>
</tr>
<tr>
<td>P4</td>
<td>4.3</td>
<td>1.333</td>
<td>3.4</td>
<td>0.75</td>
</tr>
<tr>
<td>M3</td>
<td>5.4</td>
<td>1.25</td>
<td>4.5</td>
<td>0.8</td>
</tr>
<tr>
<td>m3</td>
<td>6.5</td>
<td>1.2</td>
<td>5.6</td>
<td>0.8335</td>
</tr>
<tr>
<td>M2 (large)</td>
<td>9.8</td>
<td>1.125</td>
<td>8.9</td>
<td>0.888</td>
</tr>
<tr>
<td>M2 (small)</td>
<td>10.9</td>
<td>1.114</td>
<td>9.10</td>
<td>0.9</td>
</tr>
<tr>
<td>m2</td>
<td>16.15</td>
<td>1.0666</td>
<td>15.16</td>
<td>0.9375</td>
</tr>
</tbody>
</table>

Table 1. Structural ratios derived from Just Intonation intervals.

These ratios are used to generate all aspects of the composition, from large-scale temporal structure and densities to detailed selection of event start time (including whether there be sound or silence), duration, sound source (live or computer-generated instruments), frequency, amplitude, timbre (including reverb), articulation, and location.

| In    | 0.0000 | 0.0250 | 0.0500 | 0.0750 | 0.1000 | 0.1250 | 0.1500 | 0.1750 | 0.2000 | 0.2250 | 0.2500 | 0.2750 | 0.3000 | 0.3250 | 0.3500 | 0.3750 | 0.4000 | 0.4250 | 0.4500 | 0.4750 | 0.5000 | 0.5250 | 0.5500 | 0.5750 | 0.6000 | 0.6250 | 0.6500 | 0.6750 | 0.7000 | 0.7250 | 0.7500 | 0.7750 | 0.8000 | 0.8250 | 0.8500 | 0.8750 | 0.9000 | 0.9250 | 0.9500 | 0.9750 | 1.0000 |
|-------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| 0.0   | 1.000  | 0.9825 | 0.9111 | 0.8166 | 0.7272 | 0.6333 | 0.5333 | 0.4333 | 0.3333 | 0.2333 | 0.1333 | 0.0333 | 0.0333 | 0.0333 | 0.0333 | 0.0333 | 0.0333 | 0.0333 | 0.0333 | 0.0333 | 0.0333 | 0.0333 | 0.0333 | 0.0333 | 0.0333 | 0.0333 | 0.0333 | 0.0333 | 0.0333 | 0.0333 | 0.0333 | 0.0333 | 0.0333 | 0.0333 | 0.0333 | 0.0333 | 0.0333 |
| 0.5   | 0.000  | 0.000  | 0.000  | 0.000  | 0.000  | 0.000  | 0.000  | 0.000  | 0.000  | 0.000  | 0.000  | 0.000  | 0.000  | 0.000  | 0.000  | 0.000  | 0.000  | 0.000  | 0.000  | 0.000  | 0.000  | 0.000  | 0.000  | 0.000  | 0.000  | 0.000  | 0.000  | 0.000  | 0.000  | 0.000  | 0.000  | 0.000  | 0.000  | 0.000  | 0.000  | 0.000  | 0.000  |

Figure 1. Proportional division of total time (normalized to 1.0) using L/Ratio values (0.5 < x < 1.0) and their reflections around 0.5. [1.0 - (1.0/Ratio)] (0.0 < x < 0.5).

The program uses a random number generator, focused by various probability tables (themselves generated by the initial set of ratios), to make the necessary decisions of temporal change in each event parameter. This process serves as the computer analog to the composer’s “intuition”, that undefinable process of human creativity.

Several parameters operate in pseudo-curved ranges, including frequency, amplitude, reverb, and stereo location (and its related panning operations). This procedure prevents these parameters from exceeding, or actually reaching, their respective boundaries. In these parameters (also normalized to 1.0), proto-values are first proportionally generated, tried against the established boundaries, and then curved if either boundary is exceeded.

\[ \text{PRO(n)} = \text{Ratio values} \times \text{increasing proportion a,b, where a \geq b} \]
\[ \text{ZPRO(n)} = 1.0/\text{Ratio values} \times \text{decreasing proportion b,a} \]

New (proto-) Value \( V_{\text{old}} \) = Old Value \( V_{\text{old}} \) \times \text{PRO(n)}

IMC 515
For increasing values:
  If $V_p >$ high boundary (HB), then

  $$V_p = (HB - V_{old}) \cdot ZIPRO(n) + V_{old}$$

For decreasing values:
  If $V_p <$ low boundary (LB), then

  $$V_p = V_{old} \cdot ((V_{old} - LB) \cdot ZIPRO(n))$$

Figure 2. Proto-values generated and curved, if needed, to remain within established boundaries.

Values thus determined in a global range are then mapped proportionally onto a specific, reduced range when needed. This occurs frequently in the frequency parameter for live instruments: proto-frequencies are generated in the global frequency range of the computer-generated sounds (e.g., between 20 and 10,000 cps, if SR=20,000), and then mapped to the reduced range of a specific instrument.

[Note: $\log = \log_{10}$]

- $R_g =$ global range
- $\delta HB_g =$ global high boundary
- $\delta LB_g =$ global low boundary
- $V_g =$ global proto-value
- $R_{fac} =$ range factor
- $R_s =$ specific (e.g., instrument) range
- $\delta HB_s =$ specific high boundary
- $\delta LB_s =$ specific low boundary
- $V_s =$ specific value (e.g., mapped frequency)

$$R_{fac} = \frac{\log HB_g - \log LB_g}{R_g} \quad \text{[Note: $R_g$ and $R_s$ are defined as constants in program initialization.]}$$

$$V_s = \text{unilog}(\delta HB_s \cdot R_{fac}) + \log LB_s$$

Figure 3. Global values mapped onto specific ranges.

This computing environment allows the composer to work more accurately towards the unification of process and structure, of live activity and computer processing, and of computer-assisted composition and computer-generated sound throughout a composition, while still permitting dynamic timbral control (which remains the great strength of software synthesis). A new work for flute and Kyma system, (GUTS 3e) in the series, is currently being refined using COMP2. It is being realized in two performance formats: one for flute and Kyma system operating in real time (for use when the Kyma system is available); and one for flute and previously recorded Kyma-generated tape (when the system is not available).

Summary

This composer's interest in aesthetically and procedurally unify the diverse aspects of human compositional and computer processes within a given musical work using proportional, recursive, stochastic algorithms is greatly enhanced and advanced through the use of the Smalltalk-80 programming language within the Kyma digital synthesis system.

ICMC 516