Realtime Stochastic Decision Making for Music Composition and Improvisation

Christopher Dobrian

Introduction

Iannis Xenakis formalized the application of the mathematics of probability in music composition.[1] He cited a perceived "crisis of serial music"[2] as leading logically to a statistical approach to composition rather than a melodic one. When polyphony is sufficiently dense and complex, he posited, "The enormous complexity prevents one from following the tangled lines and … what will count will be the statistical average of isolated states of the components' transformations at any given moment."

Xenakis' explications of statistical control of events, uses of randomness and probability, and granular methods of sound synthesis inspired subsequent generations of instrumental composers and computer musicians.[3] His influence has been particularly noteworthy in the field of computer music, not only for his invention and development of the UPIC system, but because from as early as the 1950s his compositional ideas implicated intensive calculation and the generation of large randomized samples, both of which are best carried out with the aid of a computer. In his own music and in the work of those influenced by him, stochastic pro-

[1] Iannis Xenakis. *Formalized music; thought and mathematics in composition* (Stuyvesant: Pendragon Press, 1992). [Original English publication: (Bloomington: Indiana University Press, 1971). Previously published as *Musiques formelles*, (Paris, 1963)].

[2] Iannis Xenakis. "La crise de la musique sérielle". *Gravesaner Blätter*, 1 (1955) 2-4.

[3] James Harley. "The Creative Compositional Legacy of Iannis Xenakis" in *Definitive Proceedings of the 'International Symposium Iannis Xenakis'* (Paris: CICM, 2006).

cesses—the systematic use of the mathematics of probability and randomness—have been applied in two closely related fields: music composition and sound synthesis.

The notion of taking such a rigorously mathematical approach to the composition of large-scale musical works was novel at the time, and remains foreign to many people. The meeting of several sophisticated disciplines in Xenakis' work—music, mathematics, and eventually computer programming—makes for a rich but very complicated and highly technical discussion. Even those who understand the gist of Xenakis' ideas find the level of sophistication and technicality in his writings intimidating, and relatively few composers have pursued stochasticism as a comprehensive methodology. Nevertheless, computers have made possible many mathematical approaches to composition involving intensive calculation, and have led to various usages and elaborations of Xenakis' ideas.

In this essay I attempt to describe some most basic applications of stochastics in algorithmic composition and granular synthesis in as plain a language as possible, and then to discuss how such techniques are applicable in realtime improvisation.

Basic Principles

Mathematical formulae can be employed to describe and control any musical parameter—pitch, loudness, duration, density of events, etc.—as well as the statistical distribution (the relative frequency of occurrence) of different types of event. What I am terming an "event" can theoretically describe an occurrence at any formal level, but here I will be focusing on "notes" (single unified sounds) or on even smaller acoustical quanta[4] called "grains" of sound, which are grouped together to form a sonic texture.

A "parameter" is, by definition, any factor or trait that is measurable and can therefore be described numerically. Any aspect of a music composition that can be quantified is subject to numerical description and manipulation. In the following discussion I most frequently use musical pitch as an example parameter, but it's important to remember that these principles of mathematical control can be applied equally well to *any* pa-

[4] Dennis Gabor. "Acoustical Quanta and the Theory of Hearing" in *Nature* 159:4044 (1947): 591-594 and Curtis Roads. *Microsound* (Cambridge: The MIT Press, 2002).

rameter. For that reason, when discussing numerical representations, I use the word "data" in order to reinforce the idea that "it's just numbers", and that the principles under discussion are generalizable to include many aspects of music. Indeed, the concept that music is describable as multiple simultaneous parametric data streams is crucial to this discussion.

One of the interesting aspects of this way of describing and controlling music is that the different parameters of the music can be treated "orthogonally"—with each parameter being potentially independent of the other parameters. Whether it be for musical notes or sonic grains, the parameters of pitch, amplitude, duration, rate of occurrence, timbre (source), articulation (envelope), and location for each constituent event of a sonic texture can all vary independently, as can more global characterizations of collections of events.

I will focus here on three characteristics that can be used to describe a set of data: 1) the range of possibilities, 2) the distribution of events within a given range of possibilities, and 3) the continuous transformation of range and distribution. Once a musical passage is so described, it can be composed with simple numerical controls.

Pitches in the twelve-tone equal-tempered system are most easily described with the standardized numbering system of MIDI, wherein middle C is 60 and each semitone step away from that is plus or minus 1; thus, for example, the pitch A at 440 Hz is designated as 69 (9 semitones above middle C), the range of a piano is 21 to 108, and so on. Pitch classes have long been represented as numbers 0 to 11 in music theory discourses focusing on serialism and set theory. Pitch class is easily derived from any MIDI pitch number with a modulo 12 operation. For example, the low A of the piano, 21, and the A above middle C, 69, are revealed to share the same pitch class because they are "congruent modulo 12". (21/12 and 69/12 both have the same remainder: 9.)

Other parameters can be similarly described as a range of numerical possibilities. For example, the relative amplitude of sounds (roughly comparable to their relative loudness) can be expressed on a decibel scale. We commonly use 0 dB as a reference to the greatest amplitude a computer can provide, and we establish some much lower amplitude (-60 dB, for example) as the softest musically practical amplitude, relative to the maximum. In this way, dynamics of notes or grains can be expressed on a linear scale—from -60 to 0 in this case—similarly to the way that pitch can be—from, say, cello low C 36 to flute high C 96. Spatial location of a sound in a stereo field can be described numerically as the change in polar angle from the listener's position. For example, with 0 being left and 1 being right, any intermediate number represents a point between the two extremes, on an arc from the center. The rate of events can be usefully described numerically as notes (or grains) per second. Duration can be described as a number of milliseconds (particularly appropriate in stochastic and granular textures).

In short, each attribute of each sound can be ascribed a number within a given range. Each range can be described in terms of its minimum value and its size (maximum minus minimum). Those range descriptors—minimum and size—can be held constant over time to create a static, globally unchanging texture, or they can be changed continually over time to create a gradually transforming texture.

In a totally random distribution of possibilities within a given range, all possibilities have an equal likelihood of occurring. In that case, over a large sampling they will tend to occur in equal amounts. This equal distribution is the musical equivalent of white noise. The random numbers can be shaped by control of their range, but the content of that range is neutral. To create a distribution in which some things are more likely to occur than others, we can ascribe different "weightings" of probability to each possibility. This lends a distinctive flavor to a set of events because some characteristics hold prominence simply by virtue of occurring more often. For example, if the pitch classes C, E, and G occur with three times the likelihood of all other pitch classes, the musical passage will decidedly imply C major.[5]

So, in addition to restricting the range of a set of data, we can weight its content by using a table of probabilities describing the likelihood of each possible event within that range. Each aspect of a stochastic texture—its pitch, amplitude, etc.—can be described by this range and distribution information. The combination of information for all important musical parameters constitutes a description of the texture; the description defines the boundaries and characteristics of the music.

[5] Carol L. Krumhansl. *Cognitive Foundations of Musical Pitch* (Oxford: Oxford University Press, 1990).

A Simple Example

Statistical or probabilistic descriptions of musical textures rely on large numbers of events for their characteristics to be evident. Statistical trends or biases in a population are more readily observable in a large sample. Dense clouds of notes, micropolyphonic melodies, and granular synthesized sounds all require specification of a great many parameters for a great many events. The multiplicity of sonic characteristics that need to be specified for such large numbers of events points naturally—as it clearly did for Xenakis—to an automatically generated stochastic approach.

That having been said, one can still apply a numerical description to a much smaller number of events, such as a single melodic line, with clear effect. The essential data characteristics mentioned above—range, statistical distribution of possibilities within the range, and the change in range over time—can be used in more traditional, non-computerized melodic composition. One can compose music within specific statistical constraints without employing randomness. As the number of specified constraints increases, the compositional process becomes increasingly systematized, and the composer's freedom to make decisions on the basis of "taste" or "intuition" is bounded by those constraints. If a composer rigorously adheres to a set of statistical descriptions, the resulting music will be guaranteed to exhibit those statistical properties even as other traits of the music may vary. This method of composition doesn't require large numbers of probabilistic choices; it can be implemented by using some permutation of a finite set of possibilities that already contains the desired statistical distribution, or it can be implemented simply by ongoing record-keeping of intuitively-made choices. The short musical example in Figure 1 on the following page illustrates this.

Figure 1 Linear changes.

In this ten-second melody for flute, ten notes occur in the first second, nine in the second second, eight in the third second, and so on, till the final second is a single note. A similar linear reduction can be observed in the number of pitches that belong to the key of F# major: 100% at the beginning of the melody changing to 0% by the end of the melody. The pitch range of the melody descends linearly from the flute's high range to its low range, and the loudness diminishes from very loud to very soft. All four parameters—note density, predominance of a given scale, pitch range, and loudness—diminish together in the same direct manner, giving a concerted impression of downwardness.[6] This was achieved in fact by a pre-compositional decision to restrict these particular musical parameters in very specific ways according to the graphs shown in Figure 2. Within those restrictions, the composer is free to exercise other methods of decision making for the exact pitches and rhythms, but the imposed statistical limits ensure a particular character and directionality in the musical result.

[6] The characterization of these trends as "downward" in a spatial metaphor is a subjective cross-domain mapping, but it's a commonly accepted one for each of these musical parameters, and is supported by the convention of larger numbers being spatially represented as higher on a vertical axis.

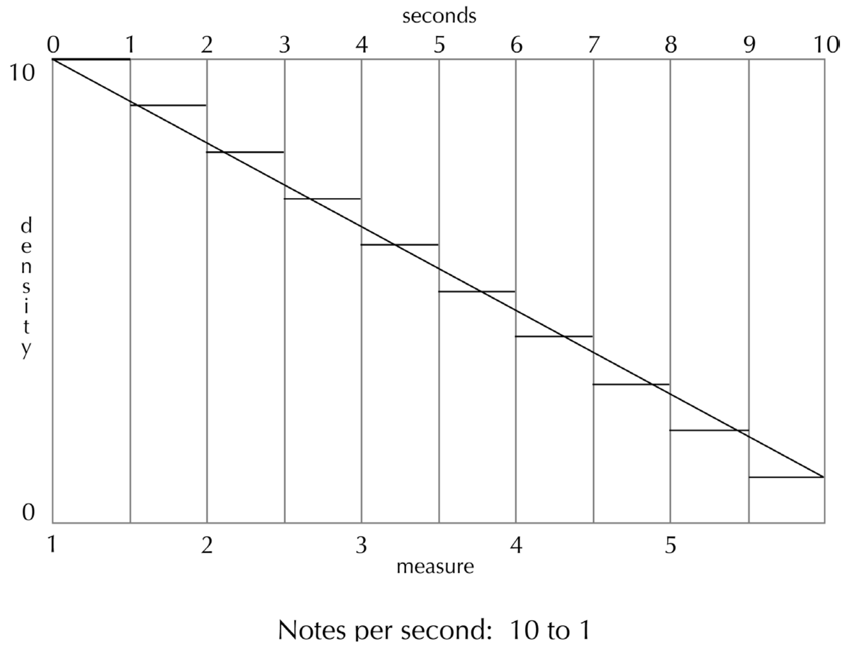


Figure 2a Density.

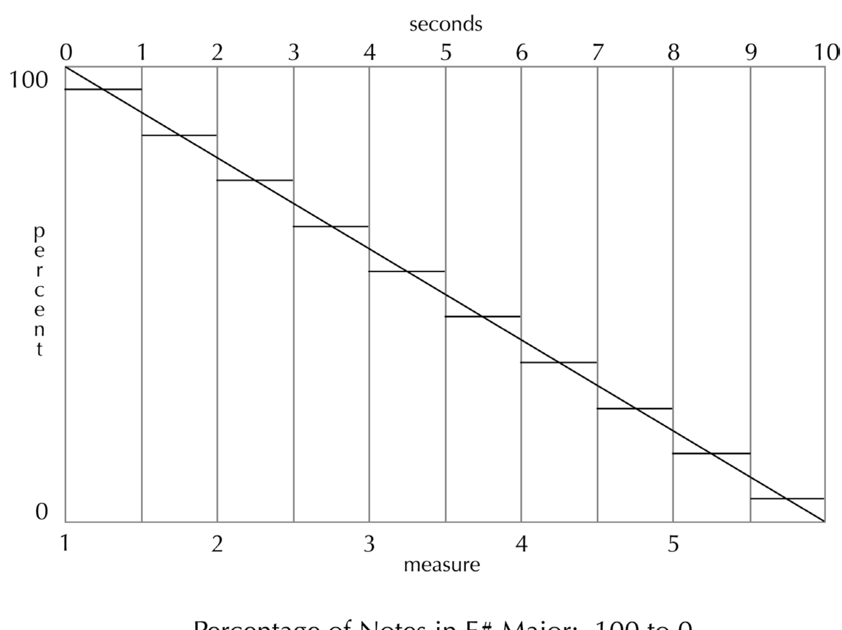
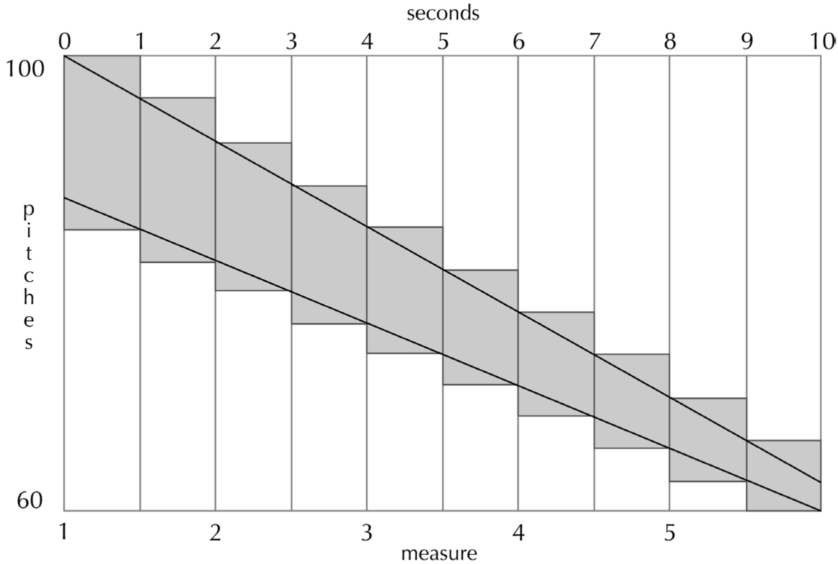
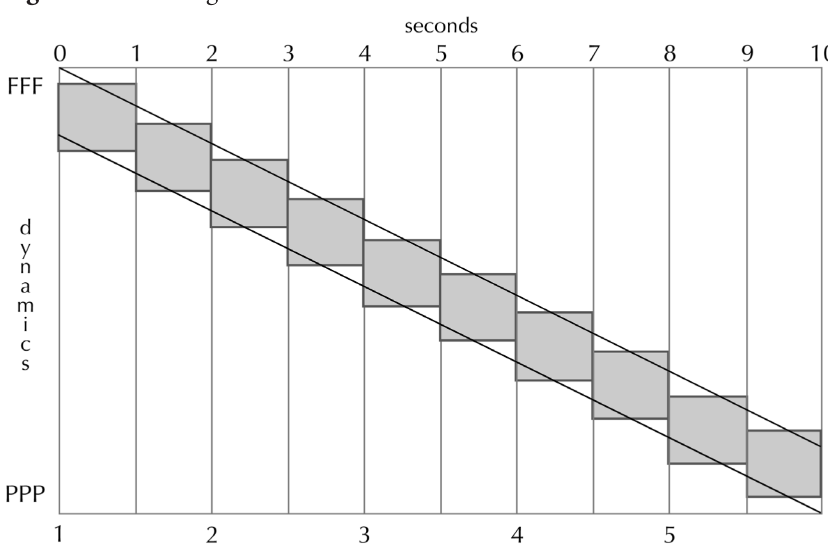


Figure 2b Percentage of notes.



Lowest pitch: 81 to 60 — Range size: 19 to 7

Figure 2c Pitch range.



Maximum: FFF to P — Minimum: F to PPP

Figure 2d Dynamics.
(**Figure 2a-d** Statistical descriptions of Figure 1).

A melody with those same statistical characteristics can be composed by a computer simply by imposing the necessary scaling factor and offset that will cause all decisions to be in the appropriate ranges.

Control of Randomness

All programming languages provide a means to generate pseudo-random numbers. The exact implementation varies from one language to another. The programmer's task is to a) establish a known number of randomly generated possibilities, b) spread those possibilities over a certain size range by multiplying them by a common factor, c) move that range up or down, by adding a common offset, to establish the desired minimum value, and d) optionally weight the possibilities so that some are more likely to occur than others.

So, for example, if we want the computer to choose a piano key to play at random, and our programming language provides us a means to generate one of a million random decimal numbers from 0 to 0.999999, we could multiply the chosen number by 88, thus scaling the size of the range to be from 0 to 87.999912, then ignore the fractional part, thus limiting it to one of 88 whole number possibilities 0 through 87, then add 21 to the result, to offset it into the range of possible piano notes 21 through 108.

If we want to select from a collection of discontiguous numbers, or we want to weight the distribution of possibilities, we can store an array of all the desired possible results in the computer's memory—with whatever duplicates we may need in order to skew the relative likelihood of the possible outcomes—and then use the randomly generated number as an index by which to look up the result in the array.

This relatively simple set of operations, applied to establish the boundaries and distribution of randomly generated data for different parameters of notes or grains, is actually sufficient to describe a substantial variety of stochastic textures. The benefit of such simplicity is that higher-level control can be achieved by specifying a fairly small set of constraints. If one then changes those constraints gradually over time, via linear or exponential interpolation from one set of parametric data to another, evolving stochastic textures result.

Figure 3 depicts, using the graphic programming language Max, this sort of simple manipulation of the range of random data. A fast-tempo metronome triggers random note data choices at a rate of 16 notes per second (one note every 62.5 milliseconds) while a very slow-tempo metronome chooses new constraint values—range size and minimum for pitch and loudness of the notes—every 5 seconds (5000 ms). Over the course of five seconds, the program interpolates linearly toward those new constraint choices, arriving at them just in time for the slow-tempo metronome to select a new set of destinations.

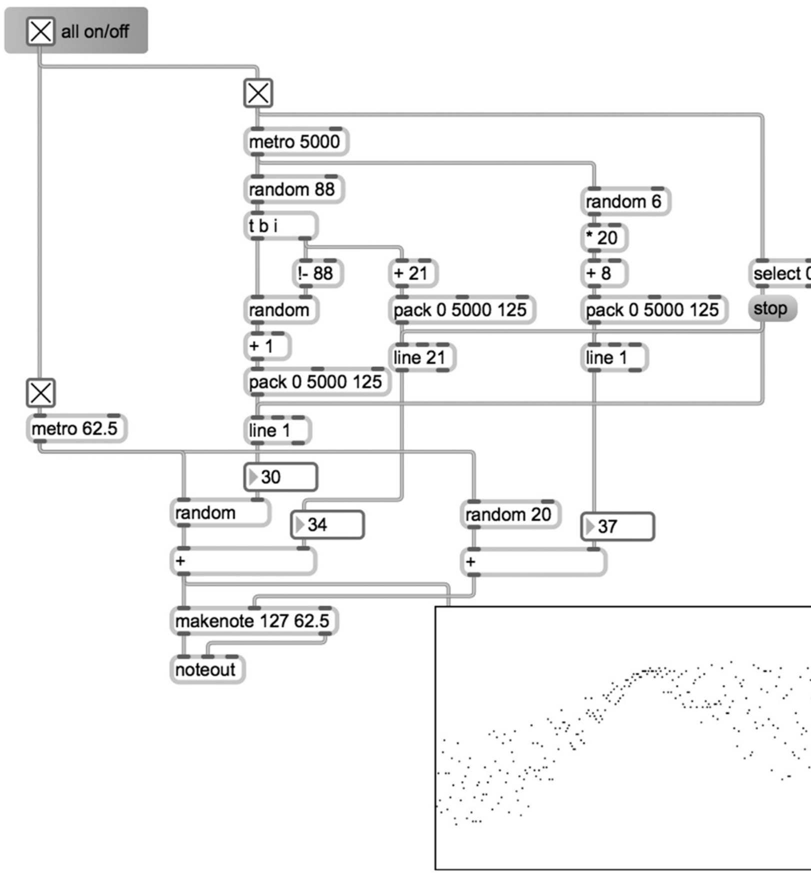


Figure 3 Moving range screenshot.

Macroscopic Formal Control

The large number of events in stochastic musical textures requires large sets of data. The data are produced by pseudo-randomly generated numbers that are limited within designated ranges and have particular weightings of likelihood. The textures can therefore be described just by stating those limits and weightings, which requires a smaller set of numbers. We can think of these as parametric data describing the texture as a whole, a description of a higher formal level than the grain or note level. Yet over the passage of time in a musical composition, those texture descriptors must change, so each parameter of the texture description must itself be thought of as a time-varying function (a shape over time).

Xenakis explained a technique of using what he called "screens": depictions of the frequency, amplitude, and density of events as they would occur in a single brief slice of time.[7] By collecting a great many screens in order, like pages of a book, he could describe a larger period of time. Each screen depicted the sonic texture at a given moment, and the ordering of the pages determined the changes the texture would undergo. This is in some ways analogous to the way that film and video divide time into discrete frames, each one a stationary snapshot of a single instant in a continually moving flow of time. To design and construct a film frame-by-frame, however, is a time-intensive and decidedly non-realtime proposition. Furthermore, depicting music as separate time slices allows an intimate view of a single moment in time but does not provide a picture of the more global formal structure viewed outside time, as does a more traditional musical score.

We can describe a stochastic cloud of notes or a granular synthesis texture in terms of the range and distribution of each of its aspects. In this way, we're considering its global characteristics and letting the computer make decisions within those constraints. That reduces the amount of information needed to describe the composition of the music. Yet even a global description requires specification of at least a half dozen or so parameters, each of which may vary independently of each other. If we want to view how those parameters change over time, it's possible to depict them as overlapping graphs with time as the common x axis; however, such a

[7] Iannis Xenakis. *Formalized Music*, op. cit. 40-78.

graph is difficult to read because the variation of each parameter over time is an independent shape, and the y axis represents something different for each one. Imagining and managing these multiple parameters and their interrelationships as they change over time is thus a fundamental challenge for the composer of stochastic music.

In my stochastic composition *Entropy* (1991) for computer-controlled piano[8], I reduced the number of parameters as much as I could, constraining the computer's composition of notes in terms of only three factors: pitch class, octave, and loudness. Each passage of the music was defined only as: a number of notes to be composed, a starting set of weighted probabilities for each factor, an ending set, and a curve of acceleration from beginning set to ending set. (A "passage" could be specified to contain any number of notes. The probabilities are realized more accurately as the number of notes in the passage increases.) The computer calculates probability weightings at each instant by interpolating between the beginning set and the ending set specified by the human composer and stored as arrays in the computer's memory. The composing algorithm thus composes a passage of any length by a) calculating an array of instantaneous probabilities by mapping a point on an exponential curve between elements of the starting array and ending array, b) making a stochastic choice based on those instantaneous probabilities, and c) incrementing toward the ending point and repeating the process. Because the input description is stated in terms of relative probabilities of different musical occurrences, one can easily specify music that ranges between totally predictable (negentropic) and totally unpredictable (entropic), and which can transform gradually or suddenly from one to the other.

Musically this technique demonstrated to me the usefulness of stochasticism not only as a model for the distribution of sounds in time, but also as a method of creating a new sort of harmonic "order". With this technique, one can modulate from one pitch class set to another by means of a "probability crossfade". During the period of transition, both sets are present to some degree, and a new, more ambiguous set comes into being. This type of modulation can be done between highly predictable (uneven) weightings and unpredictable (evenly distributed) ones, such that the degree of entropy itself becomes the focus in a passage of music.

[8] Christopher Dobrian. *Artful Devices* (compact disc recording)(Albany: EMF, 2000).

Realtime Control

Because computers are able to generate large numbers of sonic events instantaneously, stochastic composition can be achieved in real time, providing the opportunity for stochastic *improvisation*. Formal characteristics of such an improvisation can of course be provided to the computer beforehand to guide its decision making, but in that case is it really improvisation? To distinguish the notion of stochastic improvisation from any other sort of stochastic composition, let us stipulate that in an improvisation control information must be provided interactively in real time, in addition to whatever constraints have been provided beforehand.

How does one control so many different parameters or aspects of a stochastic structure in real time? Even though we're only trying to control global characteristics of an automated stochastic process, the number of parameters can be daunting for a human to manage on the fly. Two useful approaches were employed in the relatively early commercial improvising software *M* and *Jam Factory*[9] by the Intelligent Music company. The programs generated music stochastically, based on live note input provided by a human performer on a MIDI instrument. In those applications, the music being played by the human interlocutor was a primary source of control for the stochastic improvising program. The notes provided by the human were used directly to establish probability tables or Markovian transition tables. In this way, the computer in some way emulates the live performer, because the statistical distribution of its choices has been determined by the live performer, even if the computer does not play the individual notes in the same order.

M and Jam Factory also used an approach that can be likened to Xenakis' screens: the user of the program could, on the fly, choose to store a "snapshot" of the musical characteristics of a particular moment in time. Instead of storing a graph of the distribution of frequencies, amplitudes, and densities per se, as Xenakis did, Intelligent Music's snapshots stored the instantaneous settings of all control parameters and probability tables. The snapshots provided a collection of presets of control information as it had existed at specific historical moments, available to be recalled and reordered improvisationally at the discretion of the live performer.

[9] David Zicarelli. "M and Jam Factory" in *Computer Music Journal*, 11:4 (1987) 13-29.

In my own improvisation software for works involving improvisation I have employed this paradigm of presets that contain complete parametric descriptions of a particular stochastic texture, and I have added the capability to interpolate gradually from one preset to another. Thus, a performer can choose either a) to jump from one preset to another for an immediate change or b) to make a smooth transition from one state to another with the starting and ending presets being considered more as milestones in a continuous modulation of the texture. With this interpolating capability, an improviser can easily control in real time the multiple streams of parametric descriptors that might otherwise overwhelm one's abilities for multidimensional thought and manipulation.

Because stochastic improvisation demands control of these multiple streams of descriptive data, I propose that the morphology of stochastic textures can be usefully conceived as travel through a multidimensional parameter space. At any given moment in time, the listener is situated at a unique point in that multi-dimensional space, and is traveling with a particular acceleration in each dimension. This is a very intriguing and useful way to conceive of any music, and is particularly appropriate for stochastic music. However, for most people—myself included—the notion of navigating a universe of possibilities with many more dimensions than the three or four we're used to visualizing nearly exceeds the powers of imagination and certainly exceeds our ability to depict it in two dimensions. How then can a user best negotiate and control the exploration of such a space?

An orchestra conductor shapes the performance of an orchestra but does not play every note of the symphony. The conductor relies on the existing capabilities of the individual constituent members, confident that a comparatively simple gesture can adequately instigate and control a vastly more complex and multidimensional result. In short, the conductor relies on the musical "intelligence" of the system that is the orchestra. Likewise, the pilot of a jet aircraft does not directly control every mechanical part of the plane to steer it successfully. The physical control manipulated by the pilot merely sends instructions to a computer that enacts a more complex set of robotic instructions. Similarly, the multiple dimensions of a musical parameter space can be consolidated into a more manageable set of controls in order for stochastic music to be improvised in real time.

A "fly-by-wire"[10] model is indicated, one in which the computer enacts a one-to-many scheme of mapping a performance gesture to multiple controls.[11] Even beyond the one-to-many mapping of a simple gesture to a multidimensional result, a fly-by-wire system can incorporate algorithmic behaviors that mediate the gesture-to-result relationship in complex ways that themselves involve interactivity and intelligence.

Young people are constantly developing new virtuosity in the navigation of virtual space with kinetic game controls. Current interaction devices for computer games, on which many people have developed a considerable mastery for controlling motion in virtual space, include the Sony Dualshock Playstation game controller and kinetic game interfaces such as the Nintendo Wii and the Microsoft Kinect. On small mobile devices such as the Apple iPad, a multitouch screen surface allows one hand to provide several streams of continuous input simultaneously. These are but a few examples of recent interface tools (recent as of this writing!) that are rapidly changing the way that we can interact with computer software. Imaginative application of new technologies, combined with ever more intelligent (i.e., more profoundly considered) software programming, will lead to improved control and expression in realtime music making involving stochastics.

[10] Joel Chadabe. "The Limitations of Mapping as a Structural Descriptive in Electronic Instruments" in *Proceedings of the NIME-02 Conference on New Instruments for Musical Expression* (Dublin, Ireland: 2002).

[11] Marcello M. Wanderley and Marc Battier (eds.) *Trends in Gestural Control of Music* (Paris: IRCAM - Centre Pompidou, 2000).