Implementing the Dynamic Stochastic Synthesis

Peter Hoffmann, Skalitzer Str. 96, D-10997 Berlin, pyteuer@cs.tu-berlin.de

Abstract : *The Dynamic Stochastic Synthesis, conceived by Iannis Xenakis, is an algorithm that exploits probability fluctuations for the synthesis of lively, animated sound. Combined with an algorithmic procedure for the stochastic generation of the global sound structure of a musical piece, it is a landmark in his quest for an "Automated Art", lifting the creative act of composition onto the more abstract level of constructing a sound-producing automaton. The software implementation of this automaton on a computer system is essentially an engineering task. Yet there are decisions to be made in the realization of the algorithm that require some understanding of the artistic objectives of the composer. Along with a short recapitulation of the algorithm, some points of musical interest are discussed.*

1. Introduction

In the domain of digital music synthesis, the problem of composition may be formally viewed as the task to produce an appropriate sequence of sample amplitude values. Given the multitude of sample values to be determined (e.g. 44.100 values per second), one is obliged to take recourse to various synthesis models that help reducing the amount of sound information to a set of synthesis parameters and leave the job of deriving the raw sound data to a machine. Using probabilities is a way of generating a rich variety of musical structures out of some input data and a limited set of rules (i.e. an algorithm) that can be given to a computer to execute.

In science, probabilities are used to describe statistical phenomena, i.e. phenomena that are either too complex to describe in detail (the movement of molecules in a gaz) or exhibit an intrinsic statistical behavior (elementary particles) [Hoffmann 1994]. The interest may either lie in describing macroscopic features of the statistical whole (temperature, overall radioactivity) or the microscopic, individual behavior of one or some of its members (the moving of a molecule, the spontaneous disintegration of an atom). Stochastic laws link the macroscopic features of a statistic whole to the behavior of their microscopic components. For example, they give the probability for a certain molecule to move with a certain velocity at a given temperature, or for an atom to disintegrate within a certain time interval at a given radioactivity rate.

In the domain of arts, the interest in probabilities does not lie in describing but *creating* complex phenomena. In the stochastic synthesis as conceived by Iannis Xenakis, stochastic laws are used to generate the successive sample values of digital sound [Xenakis 1991]. While the overall aural aspect of the work is shaped in a way characteristic to the stochastic laws, the evolution of the music at every moment is rich in detail and always new due to probability fluctuations.

The stochastic synthesis has been put to work at the Research Center for Mathematical and Automated Music (CEMAMu) in Paris by Iannis Xenakis with the help of Marie-Hélène Serra and Gérard Marino. So far, two pieces of music have been composed with it: GENDY301, premiered at the Rencontres internationales de musique contemporaine, Metz (Nov. 1991, the premiere of a former version having taken place a month before at the Computer Music Conference, Montreal), and S709, premiered at the Journées UPIC à Radio France (Dec. 1994). Instrumental works composed by a stochastic computer program were already presented by Xenakis in the early sixties, and some stochastic sounds could already be heard in La Légende d'Eer (1977), but it was not before 1991 that Xenakis produced music entirely generated by probabilities.

2. The Model

2.1. Two Stages of Computation

In the stochastic synthesis, probabilities are exploited in a macroscopic and a microscopic aspect. The macroscopic aspect is adopted for the "macroscopic" structure of the musical piece: its formal aspects and the distribution of sound events. The microscopic aspect is adopted for the generation of the sound itself. Accordingly, there are two stages of stochastic computation: first the overall structure of the piece (its formal "architecture") is computed. Then, this structure is "filled" with sound. While probability fluctuations tend to average on a larger, macroscopic scale, they make themselves strongly felt when the phenomenon is looked at at a smaller scale. (While temperature and pressure of a gaz stay the same, a closer look to its molecules will show fierce bumpings!) Such a microscopic look is taken at the "particles" of sound: the waves of the sound signal. Its oscillations are generated by stochastic processes. These processes show a sensitive dependency on their starting conditions, because they accumulate and amplify the fluctuations of probability. This sensitive behavior is also to be found in complex dynamical systems. That is the reason why this kind of stochastic sound "generation" is called "dynamic", which gave the algorithm the name: "GenDyn".

2.1.1. The Formal Structure

The architecture of a piece generated with the GenDyn algorithm consists of a given number of sections ("sequences") each having a given number of sound routes ("tracks") with a sound evolution of their own. The sound tracks are cut into patches of sound ("fields"), the length and activity of whose (sounding/not sounding) is computed stochastically.

2.1.2. The Sound

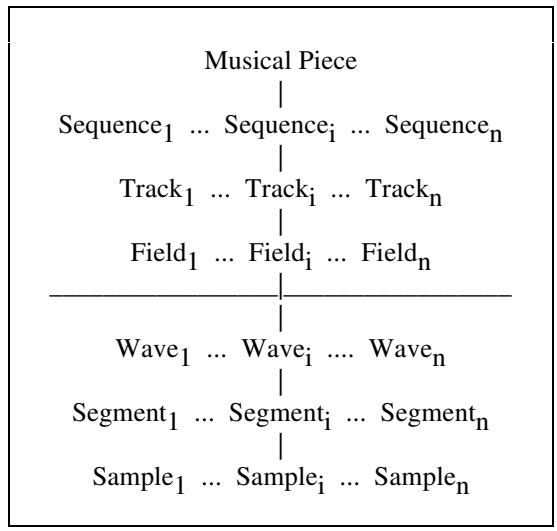
A contiguous stochastic sound is a succession of wave forms. Each wave form is divided into segments that represent in their turn a succession of sample values. The oscillating "particles" of sound are identified with the wave segments, that is, their width and their elevation over the zero amplitude is constantly changed by probabilities. Only the last sample value of a segment is computed, the other sample values being derived by linear interpolation between the segment ends. The wave is thus simplified to a polygonal shape. (The smooth wave shape can be later regained by filtering.)

We may conceive the vertex points of the wave form polygon as billiard balls moving in a two-dimensional space: the segment width on the horizontal and the amplitude on the vertial axis. With each repetition of the polygonal wave form, its vertices (= "balls") perform random steps in a vertical direction (changing its amplitude) as well as in a horizontal direction (changing the width of the segment). Each vertex of the wave form polygon performs an independent random walk, so there are as much random walks in parallel as there are segments in the wave form polygon. Because the random walks take place within the context of a closed wave form period, a periodicity is maintained in spite of the erratic movement of the "sound particles" [Xenakis 1995]. The composer calls this procedure "Polyonal Variation" because the shape of the wave form polygon is constantly varied and transformed by the random displacement of its vertex points.

2.2. Putting Things Together

The synthesis of the formal structure and the synthesis of sound to make is live are interfaced at the level of each sound track. Every track within a sequence is assigned a different kind of sonority by a different set of synthesis parameters.

As can be seen in the diagram below, a hierarchy of intermediary levels thus help breaking down the problem of generating a musical piece into the more specific problems of iteratively generating a sequence, a track, a field, a wave form and finally a wave segment. In other words, the problem of generating a linear sucession of sample values is tackled by designing a hierachy of abstraction levels above the raw sound data. The upper half of the diagram concerns the formal architecture, the lower half the sound generation as such.



2.3. Musical Interest

It is interesting to observe how Xenakis realizes his formal thinking in an algorithm by introducing the notion of "sequences". The concept of a piece of music as a succession of sections (sequences) with specific aural characteristics that either contrast or are part of a larger development is a fundamental formal concept in the music of Iannis Xenakis. In addition, in his algorithm, Xenakis realizes an "in-time" mapping of the sequences of the piece. The sequence's field duration structure is first generated as a structure "outside time" and then projected "in time". With the help of this projection, sequences can be arbitrarily reordered or repeated.

Xenakis tends to design musical textures that often have a strong coherence of themselves, but then he cuts them into pieces and lays them out in score. (A striking example is the tearing apart of the "paths" of symmetry group operations in Nomos gamma.) In the GenDyn algorithm, the tracks of continuous sound are cut into field patches. Sound generation in suspended when a field is declared as "not sounding". With the next sounding field, sound generation is continued where it broke off. This procedure has the same effect as if the sound had been generated in one piece and then be cut and laid out.

With the Dynamic Stochastic Synthesis, Xenakis approaches the problem of sound synthesis from another aspect than with his UPIC system. While both synthesis devices work directly on the sound signal without taking recourse to any of the known synthesis models, the sound of the UPIC tends to be "static" because of the identical repetition of a once designed wave form. This can be compensated, however, by altering and/or combining the other control instances of the UPIC tool (such as pitch arcs, envelopes, etc.). With the Dynamic Stochastic Synthesis, time variant sound generation is a "built-in" feature while, on the other hand, there is no direct control of the other musical parameters such as pitch, envelope or timbre. Yet, if the stochastic synthesis could be set up to work in real time, these musical parameters could be controlled by the interactive manipulation of the sound synthesis parameters while listening to the sound feedback.

3. The Algorithm

3.1. Synthesis of the Formal Architecture

The field structure of each track is incrementally computed with the help of the exponential probability distribution [Serra 1992]. This distribution assigns a probability value to the duration of each track field given a field density value as a parameter. Since it is a distribution function, it is monotone, so there exists a one-to-one mapping of a field duration to the probability of it appearing in a track. By drawing the probability value at random in an iterative procedure, the duration of each field is derived. The

individual fields vary in length but conform to the overall density of the track subdivistion into fields expressed by the distribution parameter. The length of the track is incrementally built by summing up the field durations. The activity of each field (i.e. sounding/ not sounding) is ascertained by a a simple coin tossing (where the coin can be arbitrarily biased by a parameter). The event density of a track is therefore a combination of two parametrizations: the density of its subdivision into fields and the probability for each field to be sounding. The event density of a sequence in its turn is determined by the superposition of tracks and therefore depends also on the fact how many of them are declared as "active" [Serra 1992].

3.2. Musical Interest

Due to probability fluctuations, the exact length of each track is different even when they share the same field density parameter. Since the sequence length is equal to the length of the longest track, the result is a sort of "thinning" out of each sequence towards its end as the tracks run out of activity one after the other. Consequently, there is a retardation of musical action just before the onset of a new sonority when the following sequence starts.

3.3. Sound Synthesis

Sound synthesis is done by iterating top-down through all hierarchy levels of computation. For each sample of a sequence, a "master" value is computed "in parallel" as the sum of the contributions of each active track, in analogy to the mixing of a multi-track tape. (The volume control can be done automatically, depending on the number of simultaneously active tracks.)

If the current field of a track is active, the track's contribution to the master sample is computed. It can either be an interpolated value or a value representing a vertex of the wave form polygone. The interpolated values are computed by adding an increment to a preceding value, depending on the "gradient" between two successive vertices. The vertices of the the current wave form are computed as a function of their values in the preceding wave form and a random displacement. Since they are independent from each other, they may be updated "in parallel".

In detail, to each vertex of the polygon are assigned two random walks, one for its horizontal coordinate (time) and one for it vertical coordinate (amplitude). Since time and amplitude are discreet, the horizontal coordinate stands for the number of sample values to be generated between two successive vertexes and the vertical coordinate for the integer value of the amplitudes. An integral part of the theory of random walks is the notion of barriers to confine the walk to a finite space. This fits in the picture of the moving billiard balls: a ball is reflected by the cushion if its trajectory would lead it to a location outside the table. The random walk "table" space is limited by a maximal amplitude value on the one hand and the audible frequency range on the other. (The summed up lengths of wave segments form the wave length and thus the inverse of the frequency). But the space may be limited much further in order to tame the sound evolution or to alter its timbre [Serra 1992].

The probability values for each random step are drawn from a set of specific probability distributions. Each probability distribution has its own characteristics concerning the variance of probabilistic values within a given range. If driven by different distribution functions, the "billiard balls" move in different ways. A distribution with a steep gradient will inhibit, a distribution with a smooth gradient reenforce their "diffusion" in random walk space. Gradient and value range of each function can be influenced by a coefficient parameter.

3.4. Parameter Set

The sound producing automaton simulated with the help of the stochastic synthesis algorithm takes a set of parameters as input data. The data are used to parametrize the generation of the formal achitecture and the generation of sound.

3.4.1. Formal Architecture Parameters

The following data are required to generate a sequence's formal architecture: track activity, number of fields, field density, and mean field activity for each track within the sequence. The formal architecture, once computed, can be saved as a list of field durations and activities and used as input data for the next synthesis.

3.4.2. Sound Synthesis Parameters

Sound synthesis parameters are specific to tracks. The following data are required to generate a sound: the number of wave form segments and the name of a distribution function to use, its coefficient, and the positions of the random walk mirrors.

3.5. Musical Interest

If the procedure of stochastic sound synthesis is called "Polygonal Variation", the question may be posed: What is, then, the original polygonal wave form to be varied? In fact, the original wave form to start with is silence: a polygon with all its vertex values set to zero. In this sense, the stochastic synthesis is "music out of the void": the coming of the sound into being is caused by probabiliy fluctuations only.

Unconditional probabilities have no "intention" nor "memory". Though, the musical ear works by memorizing and projecting intention and "sense" into sound processes. It is even able to "adjust" arbitrary moving pitches to the perception of musical intervals.

The two-dimensional random walk space is not a homogeneous space. Its two dimensions are of a different scale. The frequency of the generated sound depends very sensitively on the width of the polygon segments, especially if the wave length is short. While in order to obtain smooth glissandi, the wave segment width may vary only a few samples, the amplitude variations can be up to three magnitudes higher. Strong fluctuations in segment length produce a very agitated, "buzzing" sound and can even lead to nonlinear acoustic phenomena similar to frequency modulation.

Some of the synthesis parameters may be made time-variant. It could be interesting, for example, to open the random walk barriers widely at the attack of a sound and then restore their user-set position. In general, manipulating parameter settings during computation time makes it more difficult to explore the parameter space of the synthesis algorithm which by itself is already very large.

4. The Implementation of the Algorithm

Currently, there exists an object-oriented implementation of the stochastic synthesis working in real time. All synthesis is done in (short) integer arithmetics using a tabulation technique for the stochastic distribution functions. A graphical user interface helps editing input parameters interactively and graphically animates the functioning of the algorithm in order to show the user the impact of his parameter settings onto the synthesis procedure.

4.1. Object-Oriented Design

Objects are autonomous units of computation that are dynamically created during program execution. In the current implementation of the stochastic synthesis, the task of computing a sequence is carried out by a sequence object, the task of computing a track is carried out by a track object, and so on. The objects cooperate for the generation of a musical piece i.e. a sequence object employs a collection of track objects and so on. The wave form polygon of each track, the random walks working on it, and the barriers controlling the random walks are also modelled as objects. They are referenced through a superordinated "sound" object.

The use of objects for the implementation of the Gendyn algorithm has at least two advantages. First, synthesis data are rigorously structured in small seperated entities that are clearly organized on a higher level. Second, the computation is divided into small independent routines that cooperate in a straightforward way for the generation of a musical piece. Since data are closely tied to the routines working on them, there is no need to access global data structures which makes the implementation simple and easy to understand. It will also facilitate future development.

4.2. Tabulation of the Stochastic Distribution Functions

In the Gendyn algorithm, the fluctuation values for the random walks are governed by continuous distribution functions (such as the Cauchy, Logistic or Arcus Sine distributions [Serra 1992]). But the random steps to compute are in fact integer values since in digital sound synthesis, both amplitude and time are discreet. In the current implementation, the distribution functions are computed only once at their integer values and stored in a lookup table.

5. Future Work

With the current implementation, there exists a powerful tool for the computation of the stochastic synthesis. Still this tool has to be refined in order to fully meet the needs of musical creation. It could be developed even further by generalizing the idea of stochastic synthesis.

5.1. Adapting Parameter Values to Acoustic Constraints

It would be desirable for the musician user to have synthesis parameter settings scaled according to acoustic constraints (i.e. logarithmic frequency scale, logarithmic loudness scale etc.).

5.2. Building a Platform Independent Graphical User Interface

A final version of the user interface will be implemented in a fourth-generation-language which has backends to all current system platforms (PC, MAC, UNIX).

5.3. Parametrizing Wave Forms

The original wave form to start off could be defined by the user i.e. the wave form itself could be an element in the synthesis parameter set. If the original wave form could be set up by "sampling" sound input in real time, the stochastic synthesis could even serve as a powerful "live electronics" device. In this case it could be interesting to assign an individual pair of barriers to each vertex of the wave form polygon to preserve its global shape across the steps of stochastic transformation.

6. Conclusion

The algorithm of the Dynamic Stochastic Synthesis is very clear and straightforward. Yet there are many ways to implement it on a computer. Each of these ways definitely has influence on the audible result. That is why some crucial points of musical interest were pointed out in this paper. The sensitive dependency of the stochastic synthesis on the random numbers as well as numeric issues (number conversions, trigonometric functions, etc.) seem to make an "objective" comparison of different implementations (by looking up the sound data output) virtually impossible. There does still not exist a closed theory to link the abstract mathematical synthesis process to the aesthetic value of the concrete aural result. For the time being, we'll have to rely on our ears to tell the satisfying results from the less satisfying!

References

[Hoffmann 1994] Peter Hoffmann, *Amalgam aus Kunst und Wissenschaft: Naturwissenschaftliches Denken im Werk von Iannis Xenakis,* Peter Lang, Frankfurt/M. etc., 1994, pp. 65-105.
[Serra 1992] Marie-Hélène Serra, *Stochastic Composition and Stochastic Timbre: Gendy3 by Iannis Xenakis,* Perspectives of New Music 31 (1993). Quoted from the preprint (April 7, 1992).
[Serra 1993] Marie-Hélène Serra, *La synthèse dynamique stochastique de Iannis Xenakis,* Les cahiers de l'IRCAM 2 (1er trimestre 1993), pp.107-117.
[Xenakis 1991] Iannis Xenakis, *Dynamic Stochastic Synthesis* and *More Thorough Stochastic Music,* in: Formalized Music, Pendragon Press, Stuyvesant, NY, 1991, pp. 289- 321.
[Xenakis 1995] Iannis Xenakis, *Music Composition Treks,* in: Curtis Roads (ed.): Composers and the Computer, William Kaufmann, Los Altos, CA, 1985, pp. 172-191.