Brown, A. R. (2005). *Extending Dynamic Stochastic Synthesis*. International Computer Music Conference, Barcelona, Spain, ICMA, pp. 111-114.

EXTENDING DYNAMIC STOCHASTIC SYNTHESIS

Andrew R. Brown
Creative Industries Faculty
Queensland University of Technology
&
The Australasian CRC for Interaction Design
Brisbane, Australia.

ABSTRACT

This paper reports on the development of the Interactive Dynamic Stochastic Synthesizer and describes how its implementation of dynamic stochastic synthesis extends previous implementations, in particular with relation to the GENDYN implementation by Iannis Xenakis. The extensions include elaborations of existing features, parameterization to previously hard-coded attributes, new parametric controls, and interface design to support real-time interaction. In particular, the use of dynamic stochastic synthesis to produce a new class of percussive stochastic synthesis sound gestures is explained.

1. INTRODUCTION

The dynamic stochastic synthesis process was devised and first employed by Iannis Xenakis in the 1970s. Since then he and a hand full of other computer musicians [1] [2] [3] have implemented the dynamic stochastic synthesis process which produces probabilistic waveforms based on rapidly varying break point envelopes. The process produces a rich array of timbral results, but the same indeterminacy that creates this variety presents challenges to the real time control of the system. In this paper a software instrument, the Interactive Dynamic Stochastic Synthesizer (IDSS) is described, with an emphasis on explaining how it extends the original dynamic stochastic synthesis process and provides an effective interface for interaction. The IDSS was built in the language Java using the jMusic library [4] for audio and music event processing.

There were a number of driving forces behind the development of dynamic stochastic synthesis that the IDSS development has attempted to maintain. In particular these relate to the fact that dynamic stochastic synthesis manipulates data in the time domain and that frequency and spectra attributes are emergent. One of Xenakis' interests in creating dynamic stochastic synthesis was to find a synthesis method that did not rely on harmonic synthesis techniques based upon Fourier's theories [5]. When developing the IDSS, attention was paid to maintaining or reinforcing these emergent sonic characteristics. Also, the IDSS was intended as a real time instrument either for performance or composition and therefore many aspects of the implementation were designed to facilitate effective interaction and control of the dynamic tendencies.

2. BACKGROUND

Beginning in the 1950s Xenakis applied probability to the composition of instrumental works including *Metastasis* and *Pithoprakta*. Throughout his life Xenakis continued to apply stochastic techniques in various ways to instrumental works. From the 1960s he used the computer to assist with probability calculations and, seeing the processing possibilities of the computer, Xenakis began to consider synthesizing sounds based on random walks. These ideas were published in the first edition of *Formalized Music* in 1971 [6] and early experiments in stochastic synthesis were employed in parts of *La Légende d'Eer* in the 1970s.

By the 1990s Xenakis' ideas in this area had come together as a process he called Dynamic Stochastic Synthesis [5] which was realized in the GENDYN (GENeration DYNamique) software written by Xenakis in the language BASIC. GENDYN was used to create *Gendyn3* (1991) and dynamic stochastic synthesis techniques were also featured in *S.709* (1994).

Since the development of the GENDYN software a number of other software implementations of dynamic stochastic synthesis have been created. These include *The New GENDYN Program* [1], *Xenak* [2], *Stochos* [3] and the *Interactive Dynamic Stochastic Synthesizer* (IDSS) [7]. Given the greater computing resources of recent times, each of these implementations, unlike GENDYN, operates in real time and includes visual interfaces to assist control and interaction. The recent implementations vary in their complexity and purpose. The *Xenak* program is designed as an example of dynamic stochastic synthesis and is limited to a single voice. The *New GENDYN Program* is a faithful reproduction of the original GENDYN program. *Stochos* and the *IDSS* are designed for interactive control and extend dynamic stochastic synthesis in different ways.

3. DYNAMIC STOCHASTIC SYNTHESIS

The dynamic stochastic synthesis algorithm is based on random walks; a process where each subsequent value is probabilistically selected within a specified range above or below the previous value. A random walk waveform can be generated directly by specifying that each sample value is the next random walk value, however, the dynamic stochastic synthesis algorithm is somewhat more sophisticated than this. The waveform is described by interpolating between amplitude/time points, not unlike those used to describe a break point envelope, as is shown in Figure 1. In dynamic stochastic synthesis it is the position of these points that varies as a result of random walks applied to both the x (time) and y (amplitude) dimensions of each points' location.

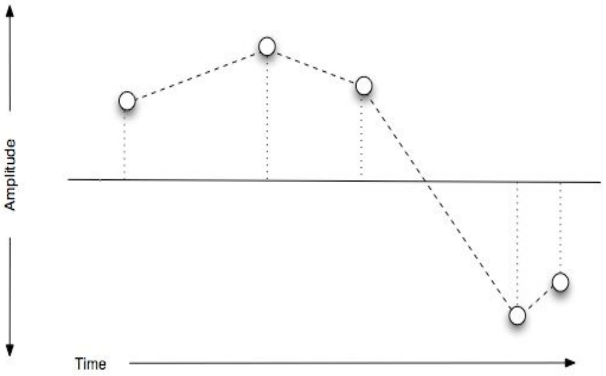


Figure 1. One cycle of a stochastic synthesis wave.

The curve described by the points makes up one cycle of the waveform and is updated each cycle. As a result of the rapid updating even small changes in the point positions have significant audible effects, and larger changes quickly result in noisy and chaotic sounds.

The behaviour of the random walks, and thus the waveform, can vary depending on the probability distribution used. Xenakis expended considerable effort exploring various probabilistic distributions[6], but in the IDSS only linear and Gaussian distributions are currently available. The points vary in two dimensions, time (left-right) on the x axis and amplitude (up-down) on the y axis. In Xanakis' design the value of each dimension of a point is controlled by two random walks in series; the output of the first affecting the maximum step size of the second which then determines the points' x or y value.

The range of values in each dimension is constrained by maximum and minimum limits referred to as "mirrors" because values that exceed the limits are "reflected" back by the same amount they exceed the mirror boundary. Fuller details of the operation of the dynamic stochastic synthesis algorithm, particularly as it is implemented in the GENDYN software, can be found elsewhere [1, 6, 8].

4. EXTENSIONS AND VARIATIONS

Taking the GENDYN implementation as the feature benchmark, the variations and additions that have usefully been incorporated in the IDSS are described here. The differences include elaborations of existing features, access to previously hard-coded parameters, new parametric controls, and interaction or interface features.

4.1. Stochastic Percussion Synthesis

The most significant innovation in the IDSS is the ability to articulate percussive gestures with the program; as opposed to the traditional format of an ongoing and potentially infinite sound field. These stochastic percussion sounds are achieved by automating a quick reduction in all random walk time-step amounts, which results in a rapid change from a bright complex timbre to a fixed and stable tone; an example wave created by this process is shown in Figure 2.

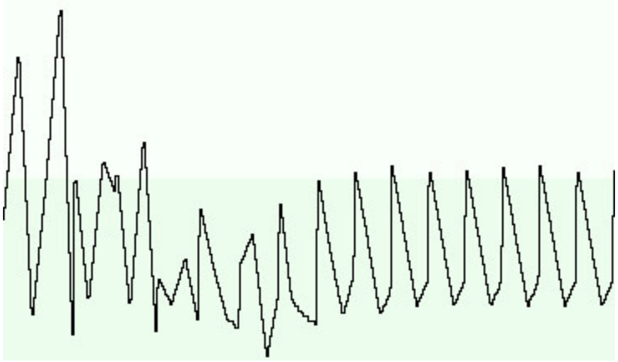


Figure 2. A stochastic percussion sound gesture.

The change in timbre has some structural similarities to the Karplus-Strong "plucked string" algorithm. A parallel reduction in amplitude step size can be applied so that the amplitude fades out toward the end of the gesture as well, increasing the similarity with the Karplus-Strong outcomes. The length of the percussive gesture is determined by the current sound field duration. The rate at which the waveform stabilizes can also be adjusted, in Figure 2 this rate was set to be about half of the total duration.

This method of creating waveforms reminiscent of vibrating physical objects adds a new performative element to the repertoire previously available with dynamic stochastic synthesis. The IDSS software interface provides a toggle switch to change a voice between percussive and the (normal) ongoing states and a slider to control the rate of stabilization.

4.2. Extended Parameters

Depending upon how the random walk step sizes are set the waveforms from dynamic stochastic synthesis vary between "all possible forms from square wave to white noise" [5:289]. However the spectral richness of a square wave, or even a triangle wave which is theoretically possible, is still quite broad and as a method of further increasing timbral variety the IDSS provides three interpolation curves, linear (standard), cosine and square. The cosine interpolation in particular enhances the range of timbres by opening up the less complex end of the timbral spectrum. Using the cosine interpolation, it is theoretically possible to create a sine wave with the IDSS, however, it is exceptionally unlikely that a pure sinusoidal wave is achieved and even less likely that it is sustained.

Another parameter that effects timbre and pitch is the number of points used to describe each wave cycle. A slider on the IDSS interface enables the user to specify the number of points between 2 and 60. A major effect of increasing the number of points is to raise or lower the pitch range. Interestingly, a resonant formant shifting effect not unlike speech or harmonic singing can result from continuously varying the number of points.

The audio fidelity of the sound is largely determined by the bit-depth of the signal. In the GENDYN program most values ranged between -100 and +100 which was close to a 7 bit resolution. According to Hoffmann [9:188] the amplitude in the *Gendy3* composition was confined to between –20 and +20 (close to a 4 bit range). The IDSS provides a slider that changes the bit depth between 1 and 16 enabling "breathy" lower-quality settings with varying amounts of quantization noise through to "smooth" high-fidelity settings.

The dynamic stochastic synthesis algorithm specifies that the last point in a wave cycle is the same as the first point in the subsequent cycle, this ensures phase coherence and avoids clicks resulting from a discontinuous audio stream. The position of this pivot point between wave cycles is otherwise unconstrained, which can result in wild-looking waveforms whose DC offset changes constantly. The IDSS has a toggle that, when activated, forces the wave to start and end at the zero crossing. This has little audible effect but significantly alters the visual appearance of the wave and maintains a consistently wider dynamic range. Some stochastic synthesis descriptions and applications assume this zero crossing as does, for example, the *Xenak* program.

A number of sections in *Gendy3* feature a soft and high-pitched introductory moment that then stabilizes at a more moderate pitch and amplitude level. This is a result of the fact that the GENDYN program used a starting position where all amplitude and time values were zero [9]. The IDSS has a "reset" button that sets the program to this zero-value starting position to mimic this behaviour, and a "randomize" button to initiate a random starting position. These can also be used as the program is running to jump to a new zeroed state or random position in wave space.

A gliding pitch change is characteristic of many dynamic stochastic synthesis settings and relies on a small step size in the random walk(s) on the time axis. In early implementations of IDSS gentle pitch slides were difficult to achieve even at the lowest step values. Hoffman reports similar results when implementing *The New GENDYN Program* [9:186] and explains the relationship between the time scale and pitch in dynamic stochastic synthesis; "The linear change in length of the generated waveforms corresponds to an exponential shift in pitch. Wavelength is the inverse of frequency, and frequency perception is logarithmic" [9:188]. To enable a more gentle pitch change in the IDSS two changes were made. Firstly, the random walk values are calculated as floating point numbers allowing more subtle variations and, secondly, the step size sliders use an exponential scale to allow is finer control over small step values.

4.3. Additional Control

As well as synthesizing sound the GENDYN program provided probabilistic macrostructure features that resulted in a voice alternating between sounding and silent periods (fields) of variable density. The large-scale organisation of dynamic stochastic synthesis sounds in *Gendy3* involved the probabilistic setting of "fields" of sound and silence in each voice. In *Gendy3* the overlapping of several such intermittent voices provided textural variation. IDSS provides similar on/off macrostructure but it is deterministic rather than probabilistic which enables regular sound fields to occur and, at short field durations, pulsed rhythmic effects.

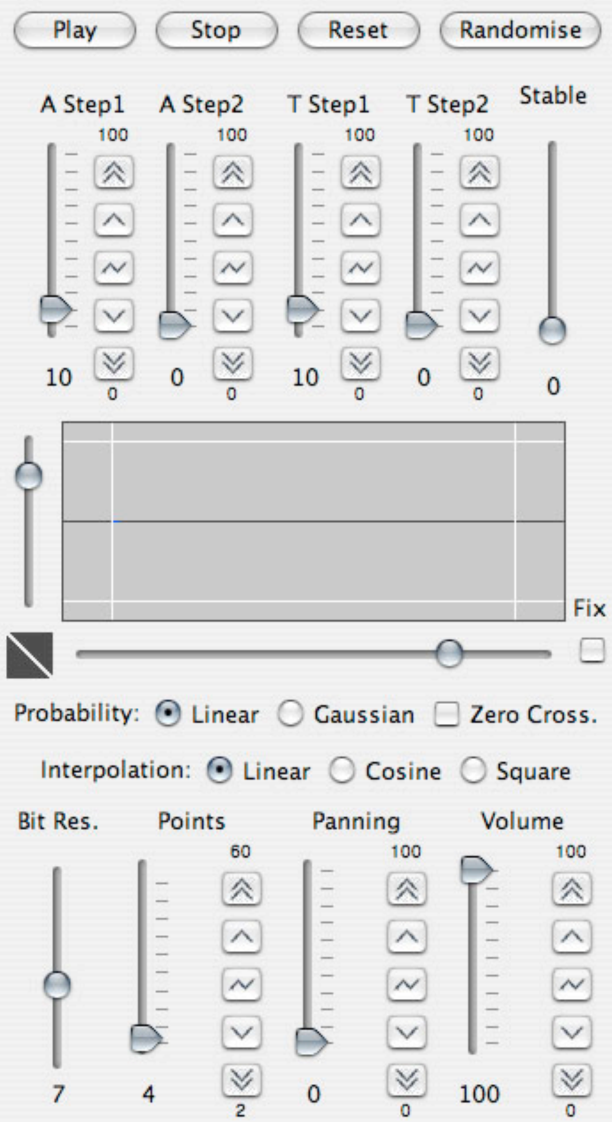
GENDYN provided a small attack and release envelope for the sound field, presumably to avoid clicks from large amplitude leaps as they began or ended. The envelope control of sound fields in IDSS (the duration of which can be specified or randomized) varies in two ways. Firstly, a sound field (called a "note" in IDSS) can have an attack and release fade length of any duration up to 50% of the sound field length. This provides for a smooth swelling of sound fields that enables a gentle mix and cross fade between polyphonic voices. Secondly, the field can have a probabilistic multi-segment break point envelope applied to it that can dramatically change the amplitude levels over time. This can provide an abrupt and punctuated dynamic activity to the music.

A waveform with a stable pitch can be achieved with dynamic stochastic synthesis by reducing the time step value to zero or by constraining the upper and lower time mirrors to the same value. To simplify this procedure the IDSS has a toggle that fixes all segment lengths to the value of the time mirror, therefore all segments have the same number of samples. The pitch (fundamental frequency of the waveform) can be altered by varying either the time mirror or number of points. These constraints combine to produce an emergent matrix of unconventional scales from which stable pitches can be selected. These scales are made up on pitches with fundamental frequencies that are multiples of the time mirror and number of point values.

4.4. Interface Features

A variety of interface features have been utilised in the IDSS design with a view to assisting real-time interaction. A general impression of the interaction potential can be grasped by looking at a portion of the interface shown in Figure 3.

These features include waveform and mirror display, slider automation on selected parameters, keyboard shortcuts and MIDI mapping of controls. These interaction design features are explored in more detail elsewhere [10].



Probability: ● Linear ○ Gaussian ☐ Zero Cross.

Interpolation: ● Linear ○ Cosine ○ Square

Figure 3. The IDSS visual interface for one voice.

5. CONCLUSION

This paper has presented a range of extensions to Xenakis' description of dynamic stochastic synthesis as they were implemented in the Interactive Dynamic Stochastic Synthesizer. This implementation of dynamic stochastic synthesis extends previous implementations in ways motivated by real-time performance requirements and the desire to provide a wide range of timbral and structural variety. This continued development of non-standard synthesis techniques extends a rich tradition where technical and creative ends reinforce one another to continually open up opportunities for musical expression.

6. ACKNOWLEDGMENTS

This work is supported by ACID (the Australasian CRC for Interaction Design) established and supported under the Cooperative Research Centres Programme through the Australian Government's Department of Education, Science and Training.

7. REFERENCES

1. Hoffmann, P., *The New GENDYN Program.* The Computer Music Journal, 2000. **24**(2): p. 31-38.
2. Chang, J.H., *Composing Noise*, in *Royal Conservatory*. 1999, Institute of Sonology: The Hague.
3. Bokesoy, S. and G. Pape, *Stochos: Software for Real-Time Synthesis of Stochastic Music.* Computer Music Journal, 2003. **27**(3): p. 33-43.
4. Sorensen, A. and A.R. Brown. *Introducing jMusic*. in *The Australasian Computer Music Conference*. 2000. Brisbane: ACMA.
5. Xenakis, I., *Formalized Music: Thought and Mathematics in Music*. Pendragon Revised ed. 1992, Stuyvesant NY: Pendragon Press.
6. Serra, M.-H., *Stochastic Composition and Stochastic Timbre: Gendy3 by Iannis Xenakis.* Perspectives of New Music, 1992. **31**(1): p. 236-257.
7. Brown, A.R. and G. Jenkins. *The Interactive Dynamic Stochastic Synthesizer*. in *Australasian Computer Music Conference*. 2004. Wellington: ACMA.
8. Xenakis, I., *Formalized Music*. 2nd Edition ed. 1991, New York: Pendragon Press.
9. Hoffmann, P., *Analysis through resynthesis. Gendy3 by Iannis Xenakis.*, in *Presences of Iannis Xenakis*, M. Solomos, Editor. 2001, Cdmc: Paris. p. 185-194.
10. Brown, A.R. *Playing with Dynamic Sound*. in *Interaction: System, Practice, Theory*. 2004. Sydney: Creativity and Cognition Studios Press.