Implementing stochastic synthesis for SuperCollider and iPhone

Nick Collins
Department of Informatics, University of Sussex, UK
N [dot] Collins]at[sussex [dot] ac [dot] uk - http://www.cogs.susx.ac.uk/users/nc81/index.html

Proceedings of the *Xenakis International Symposium*
Southbank Centre, London, 1-3 April 2011 - www.gold.ac.uk/ccmc/xenakis-international-symposium

This article reflects on Xenakis' contribution to sound synthesis, and explores practical tools for music making touched by his ideas on stochastic waveform generation. Implementations of the GENDYN algorithm for the SuperCollider audio programming language and in an iPhone app will be discussed. Some technical specifics will be reported without overburdening the exposition, including original directions in computer music research inspired by his ideas. The mass exposure of the iGendyn iPhone app in particular has provided a chance to reach a wider audience.

Stochastic construction in music can apply at many timescales, and Xenakis was intrigued by the possibility of compositional unification through simultaneous engagement at multiple levels. In General Dynamic Stochastic Synthesis Xenakis found a potent way to extend stochastic music to the sample level in digital sound synthesis (Xenakis 1992, Serra 1993, Roads 1996, Hoffmann 2000, Harley 2004, Brown 2005, Luque 2006, Collins 2008, Luque 2009). In the central algorithm, samples are specified as a result of breakpoint interpolation synthesis (Roads 1996), where breakpoint positions in time and amplitude are subject to probabilistic perturbation. Random walks (up to second order) are followed with respect to various probability distributions for perturbation size. Figure 1 illustrates this for a single breakpoint; a full GENDYN implementation would allow a set of breakpoints, with each breakpoint in the set updated by individual perturbations each cycle.

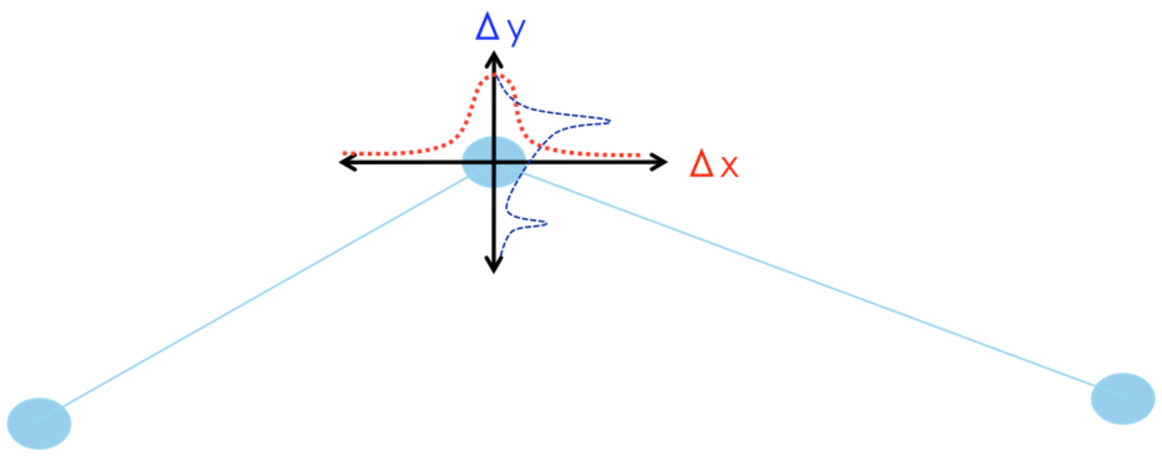


Figure 1. Probabilstic update of breakpoint. The central breakpoint (connected by linear segments to two surrounding points) is illustrating with one probability distribution on the x axis for the x update delta x, and a different distribution on the y.

The most famous electronic music compositions to thoroughly investigate these techniques are *Gendy3* (1991) and *S.709* (1992), based around second order random walks. Although GENDYN is the most well known algorithm, Xenakis also theorised about and explored more direct sample by sample synthesis with random variables or function composition of random variables (Xenakis 1992, Luque 2006). Direct sample-level selection from probability distributions, and first order prototypes of GENDYN, had appeared in the 1970s in a few works, such as *La Légende d'Eer* (1977).

Xenakis' sound synthesis work should be seen within the context of a wider investigation of 'non-standard' sound synthesis algorithms. Composers at the Institute of Sonology in particular intensively explored computer idiomatic DSP routines in the 1970s (they had the virtue of being very CPU efficient for the slow computers of that time), in such projects as Paul Berg's instruction synthesis, Herbert Brün's SAWDUST, and Gottfried Michael Koenig's SSP (Serial Selection Procedures) (Berg 2009). Much sound synthesis is concerned with recreating existing instruments, though we can create any sound which can come out of a loudspeaker: 'the uncharted galaxy of synthetic sound' (Roads 1996, p. 345). Without any acoustic model, the result tends to be raw, often noisy, sounds. Elsewhere, I have spoken of the potential of 'errant sound' synthesis, with Xenakisian GENDYN only one option amongst a multitude of technical possibilities (Collins 2008).

In techniques which directly treat the generation of time-amplitude waveforms without full grounding in real world acoustics and spectral (perceptual) models, some caution is important, however. As Max Mathews warns, 'For musical purposes, in the class ANY SOUND, almost all timbres are uninteresting' (Mathews 2007, p. 85); even in his original 1963 Science article (Mathews 1963), Mathews was careful to warn that knowledge of psychoacoustics was an important factor in sound synthesis research. The GENDYN algorithm is an example of how sounds with character can be created, without necessarily going the route of closely following real life acoustics. The random walk process gives a clear relation for the waveform from cycle to cycle, allowing a continuum from gradual change with small perturbations of breakpoint position, to very noisy timbres from fast variation.

Against this backdrop, interest in Xenakis's work has remained high amongst computer music practitioners. A number of software projects have investigated Xenakisian modes of sound synthesis, such as IanniX and the gliss iPhone app modelling UPIC, and implementations of GENDYN (Hoffmann 2000, Brown 2005). This article discusses some of my own contributions, in particular for the computer music environment SuperCollider, and built into an iPhone app for mass public use.

Implementation of GENDYN algorithms for SuperCollider

Rather than create a standalone package, in 2003, I devised three new UGens (Unit Generators) for SuperCollider (Wilson et al 2011), to make Xenakis' sound synthesis methods available as a resource within this powerful computer music system. Because SuperCollider is free and open source, so are these UGens; they are built into the core system and immediately available on download of the main application. The three UGens are:

1. Gendy1: first order random walks
2. Gendy2: second order random walks, following closely (Hoffmann 2000)
3. Gendy3: A desired frequency can be specified and achieved exactly; breakpoints are perturbed and durations fixed proportionally within the current period.

For all three, synthesis can be run at control rate for slower control signals, or full audio rate for timbral output. Cauchy, Logistic, Hyperbolic Cosine, Arc sine, Exponential and 'Sinus' distributions are available, following Xenakis' original code in Formalized Music. The 'Sinus' distribution can be used in an original manner by plugging in any other control rate Unit Generator available in SuperCollider; input is sampled to provide the distribution decision.

A particular novelty of construction was to use a circular list of breakpoints, where the list itself can be dynamically varied over time; the moment to moment control of the number of active breakpoints makes for an interesting timbral control (currently, time between breakpoints is normalized by the number of active breakpoints, so the overall cycle duration is stable with respect to the maximum number of breakpoints permitted). The algorithms allow for many further control parameters across both amplitude and duration data, including choice of distribution, scaling of perturbation, and perturbation limits. The Gendy2 UGen also has parameters for the Lehmer linear congruence pseudo-random number generator; by varying these to less statistically well-behaved values, some periodic sounds and weird timbral effects can be gained.

A strength of implementation within SuperCollider is that the Gendy UGens can be placed within arbitrarily complex networks of other sound synthesis and processing units. Figure 2 illustrates a non-trivial generative SuperCollider program using Gendy unit generators to create a thick texture. The number of Gendyn voices can be chosen at will, up to the CPU power of the machine. This is an advantage of using a programming language to specify the instructions for plugging unit generators together; SuperCollider is very efficient, and can run many hundreds of Gendy UGens per core.[1]

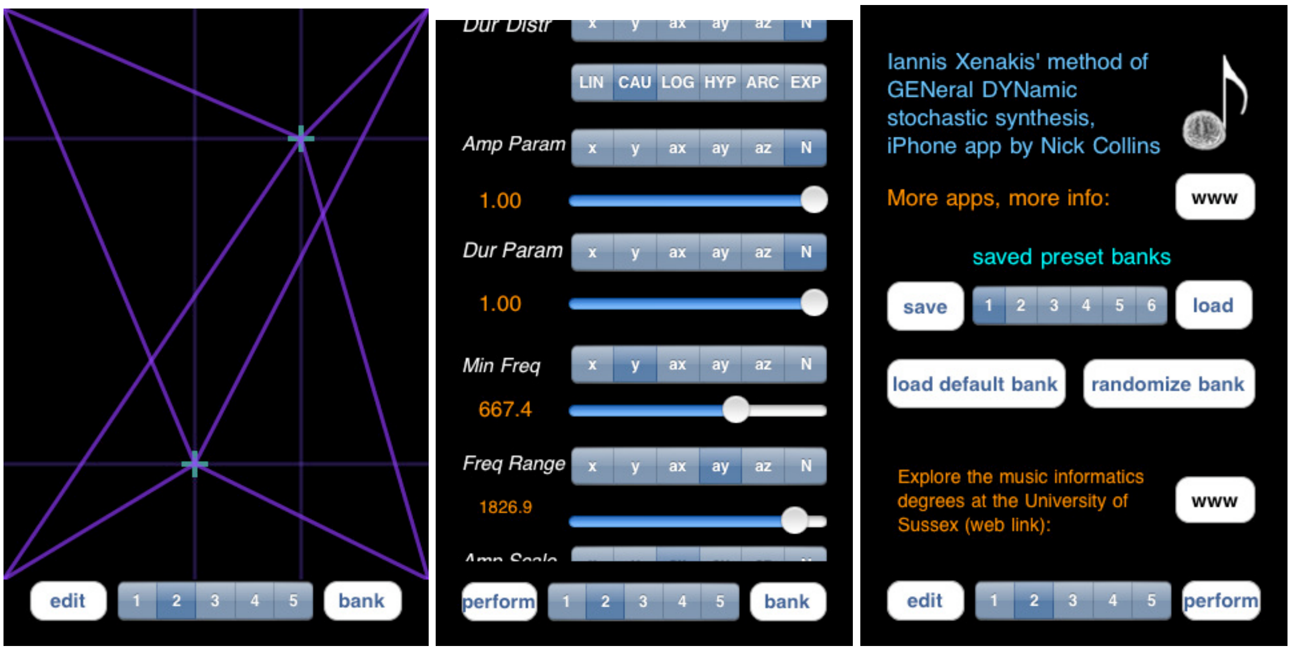
```
(
{
var n=10;

Resonz.ar(
Mix.fill(n,{ //mix together n Gendy2 UGens, each individually panned
var freq, numcps;

freq= rrand(50,560.3);

numcps= rrand(2,20);      //each Gendy2 has a different number of control points

Pan2.ar(Gendy2.ar(6.rand,6.rand,1.0.rand,1.0.rand,freq ,freq, 1.0.rand, 1.0.rand, numcps,
SinOsc.kr(exprand(0.02,0.2), 0, numcps/2, numcps/2), 0.5/(n.sqrt)), 1.0.rand2)
})
,MouseX.kr(100,2000), MouseY.kr(0.01,1.0), //mouse controls for filter settings
0.3);
}.play
)
```

Figure 2. Example code for a moderately complicated SuperCollider patch, summing up an arbitrary number of Gendy2 UGens

Implementation for iPhone

iDevices (iPod Touch, iPhone, iPad) are fun, if rather proprietary, with a large user base. They have proved highly popular for music software, having a ready audience, and providing a low latency platform for audio synthesis and processing (developer headaches though include chasing the continual version creep of iOS and the hardware). The iPhone is a useful source of controllers, including 3 axes of accelerometers (gyroscopes also since the iPhone 4), audio input (if a bit more awkward on iPod Touch, requiring a headphone+microphone set), and up to five touches detected simultaneously.

Following earlier work in SuperCollider, I have more recently worked on an iPhone app implementing stochastic synthesis, *iGendyn*. The App Store holds a host of music apps, though the grand majority are less than experimental, favoring diatonic modes, regular step sequencer rhythms and conventional timbres (beyond sampling synthesis, they usually stretch as far as typical modeling of analog synthesis via digital subtractive synthesis, but rarely go further into alternative sound synthesis methods). In my own approach to writing iPhone apps, I have tried to provide a more experimental flavour.[2] The porting of the SuperCollider Gendy UGens into an iPhone app provided an irresistible opportunity to place stochastic synthesis under multi-touch and accelerometer control. The iGendyn app was released as a free app in late May 2009, and has subsequently had around 10000 downloads.[3] Comments have generally been positive, though some users have no doubt been somewhat shocked by the noisiness of the sounds, and in some cases given up on the app immediately.[4] The app is not by any means as flexible for general sound design as the SuperCollider implementation, but it is much more widely accessible in this form. It has been used throughout the world, in recordings and on stage. To give one example, the noise artist Russell Haswell has performed with the app in live sets, reflecting Xenakis' own rich status within the noise music community.



Iannis Xenakis' method of GENeral DYNamic stochastic synthesis, iPhone app by Nick Collins

Explore the music informatics degrees at the University of Sussex (web link):

Figure 2. Three main pages of the iGendyn app; left pane, perform view, middle, edit view for selecting mappings, and right, bank view for saving, loading and randomising performance banks. The bank view includes links to a support page, and a bit of publicity for the music informatics degrees at the University of Sussex; the author is grateful to a small start-up grant from the informatics department to enable some initial computer music iPhone research in 2009.

Figure 2 reveals the three main pages in the current version (2.1) of the program. The main performance view consists of a large area where each touch is tracked (up to 5 human digits can be followed on an iPhone). The default mapping is that the x axis (on this page) corresponds to voice amplitude, the y to the minimum of the frequency range (essentially, allowed time spacing for breakpoints), and tilt parameters to control the depth of perturbation in amplitude and time, and the maximum of the frequency range. The user can set up, save and recall their own mappings via editing pages from the lower buttons. The available parameters correspond directly to the SuperCollider Gendy1 and 2's options. The choice of mapping includes touch x and y position, and accelerometer on x, y and z axes.

GENDYN Algorithm Extensions

Even in preparing the Gendy UGens for SuperCollider, certain extensions were added, such as the dynamic variation of the number of active break points, the live variation of the Lehmer random number generator parameters, and arbitrary control signals for distributions. Algorithm extensions have been explored by other authors, for example by Sergio Luque in sequencing of multiple GENDYN cycles (Luque 2006) and Andrew Brown in envelopes on perturbation size to create events with transitions from a noisier attack to more pitched decay (Brown 2005).[5] I will close by briefly mentioning a few more ideas currently being explored, in an informal way.

Breakpoint sets can be used at audio rate, at control rate for modulations, and in even slower variation in envelopes (breakpoint interpolation synthesis is essentially very fast looping through an envelope). So, slow variation of envelopes via random walks on their breakpoints is straight forward to implement. For the symposium, I presented a few demonstrations of spectral envelope fluctuation via random walk based breakpoint perturbation, including GENDYN based spectral envelope variation acting as a filter on a GENDYN noise source!

A second idea is that rather than using canned parametrised probability distributions, the probability distributions for breakpoint perturbation can be derived from live sampling (in both senses of the term). Statistics are gathered over captured audio, creating a first order Markov set of possible transitions; these determine for any given breakpoint amplitude position, a set of possible updates in amplitude. The initial effect is of changing through audio input the current texture of a GENYDN UGen. An experimental ModelGendy1 UGen has been created for SuperCollider; although a direct influence of audio input on transition behaviour of the

GENDYN oscillator can be demonstrated, the mapping itself is rather abstract at present. There are surely many more possible modeling possibilities, for example in fitting envelopes to determine possible breakpoint spacing distributions, or working more abstractly via audio features rather than direct signal.

A third extension for GENDYN is being investigated at Sussex this year by a third year music informatics student, Jonathan Young, for his project. He is exploring the scope of the sound synthesis space for GENDYN by using genetic algorithm search to try to find the optimal GENDYN algorithm parameters to match a given target sound. Whilst there is no expectation that real acoustic sounds are going to be well matched by an abstract synthesis process like GENDYN, the attempt is itself compositionally interesting. Further, at the expense of increased dimensionality, additional envelope controls can be added for time variation. Initial experiments indicate some interesting new GENDYN sounds can be created; Jonathan's report is currently being prepared for submission.

References

Ariza, Christopher. 2009. "Sonifying Sieves: Synthesis and Signal Processing Applications of the Xenakis Sieve with Python and Csound". International Computer Music Conference, Montreal.

Berg, Paul. 2009. "Composing Sound Structures with Rules". *Contemporary Music Review* 28/1: 75-87.

Brown, Andrew. 2005. "Extending dynamic stochastic synthesis". International Computer Music Conference, Barcelona.

Collins, Nick. 2008. "Errant Sound Synthesis:. International Computer Music Conference, Belfast.

Harley, James. 2004. *Xenakis: His Life in Music*. New York, NY: Routledge.

Hoffmann, Peter. 2000. "The New GENDYN Program". *Computer Music Journal* 24/2: 31-38.

Luque, Sergio. 2006. "Stochastic Synthesis: Origins and Extensions". Master's Thesis, Institute of Sonology, Royal Conservatory, The Netherlands.

Luque, Sergio. 2009. "The Stochastic Synthesis of Iannis Xenakis". Leonardo Music Journal 19: 77-84.

Mathews, Max V. 1963. "The Digital Computer as a Musical Instrument". *Science (New Series)* 142 (3592) (Nov. 1, 1963): 553-557.

Mathews, Max V. .2007. "Artist statement". In Nick Collins and Julio d'Escriván (eds.) *The Cambridge Companion to Electronic Music*, 85-6. Cambridge: Cambridge University Press.

Roads, Curtis. 1996. *The Computer Music Tutorial*. Cambridge, MA: MIT Press.

Serra, Marie-Helene. 1993. "Stochastic Composition and Stochastic Timbre: GENDY3 by Iannis Xenakis". *Perspectives of New Music* 31/1: 236-57.

Wilson, Scott, Cottle, David, and Collins, Nick, eds. 2011. *The SuperCollider Book*. Cambridge, MA: MIT Press.

Xenakis, Iannis. 1992. *Formalized Music*. Stuyvesant, NY: Pendragon Press.

Notes

1 On a two year old MacBook Pro, 500 Gendy1 UGens can be run at around 90% CPU use.

2 Other apps include explorations of live coding and instruction synthesis (TOPLAPapp, RISCy), concatenative synthesis on live streams and captured buffers (Concat), automatic remixing with beat tracking on any track in the music library (BBCut) and an iPad app based on photo to noise music mappings (PhotoNoise)

3 Exact figures are a little difficult to determine due to the limited time memory of Apple's tools for developers, and my informal calculation is from observing downloads since its release. iGendyn at the time of writing gets around 5 downloads a day; peak numbers occur around releases of version increments and other apps, and the original release year was naturally the busiest.

4 http://appcomments.com/app/id317986145/iGendyn_reviews has a selection of amusing responses by users, including 'It is either I am doing it rong [sic], or every sound I make is a screeching Banshee sound'

5 In the course of the symposium, Peter Hoffmann noted Xenakis' desire to explore the direct use of sieves for sound synthesis. I

quickly mocked up some synthesis examples in SuperCollider demonstrating this. Subsequent to the symposium, I discovered Chris Ariza's 2009 exploration of sieves in sound synthesis, using Python specified control signals within Csound. He demonstrates additive and subtractive synthesis with harmonic and filter position determined by sieves, and breakpoint interpolation synthesis, including interpolation between breakpoint sets determined by different sieves. Ariza's main tool is proportional representations for sieves, particularly for determined parameter values (like amplitude) between 0.0 and 1.0.