Dynamic Stochastic Control Voltage Generation: Adapting Iannis Xenakis' GENDY Program for Modular Synthesizer

Daniel McKemie
Independent Artist, New York, USA
daniel.mckemie [at] gmail.com
https://www.danielmckemie.com

Proceedings of Korean Electro-Acoustic Music Society's Annual Conference (KEAMSAC2021)
Seoul, Korea, 16-17 October 2021

A framework to use Iannis Xenakis' Génération Dynamique Stochastique (GENDY) program as a declarative and performative control voltage generator for a modular synthesizer.

Keywords: synthesis, stochastic, Xenakis, modular synthesizer, Csound, Python

Iannis Xenakis introduced the foundations of what would eventually become the GENDY (Dynamic Stochastic Synthesis / Génération Dynamique Stochastique) program in his paper *New Proposals in Microsound Structure* (Xenakis, 1992). In this paper he outlined a methodology for the implementation of stochastic processes in the construction of sonic waveforms. It was later in his essay *Dynamic Stochastic Synthesis* that these methods were placed in more concrete mathematical terms (Xenakis, 1992).

Early examples of Xenakis' use of stochastics to directly generate audio signals include his *Polytope de Cluny* (1972) and *La Légende d'Eer* (1977). He paused his research during much of the 1980s, instead focusing on instrumental works and his UPIC System (Xenakis, 1992). It was then in the 1990s that the GENDY program and concept was formalized. This ultimately led to the realization of *GENDY3* and *S. 709* which demonstrated a complete picture of stochastic processes that determined all aspects of a composition. The possibilities that are inherent in the GENDY system are vast, but for the purposes of this paper I will focus on the algorithmic construction and development of complex waveforms for the manipulation of external systems, both in real time and in fixed media.

The first example in this paper discusses a port to in Csound of the GENDY program in Csound to control the algorithm's parameters in a live setting. Given that the focus of computer music for the last two decades has centered more around live performance than fixed media, it is a suitable development in adapting this work to these modern uses.

The second example demonstrates the use of Xenakis' techniques to define musical form and emphasizes the evolution of this form over time by way of the inaudible control voltages that are generated. This process aligns more with the compositional world in which Xenakis worked: declaring parameters in advance and then running the program to sonically realize the musical result with no real time control or interference. Here, instead of treating the program's output as final, it is used as a control stencil in that the computed signals are output as voltage then patched into the modular synthesizer.

Hybrid System

These examples make use of both the computer and the modular synthesizer in one cohesive system. This combination seeks to expand the capabilities of both devices to fully maximize the possibilities of signal generation, modulation sources, and analog and digital computation in live performance (McKemie, 2021). The Hybrid System enables the computer to send control signals to the synthesizer which impacts its sonic output. Similarly, the modular synthesizer sends control signals back to the computer, informed by *and* informing the computer's actions in a quasi-recursive feedback loop. There are many options in how a Hybrid System can be configured based on user interests. More details and explanation can be found here[1].

In the same vein as *GENDY3*, the simultaneous realization of multiple tracks of the algorithm is possible (Hoffmann, 2009). Independent control over each track's parameters lends itself well to a variety of patch points of a modular synthesizer. The tracks can send either audio or control voltage (CV) signals that are derived from the mathematics of the algorithm in action and further modulated to whichever degree the end user's hardware is capable.

Figure 1. An out-of-the-box envelope follower, designed by Synthesizers.com

Inversely, the use of triggers and gates from the modular synthesizer enables time-based influence over various parameters in the algorithm, therefore expanding the feedback capabilities overall. These two directions of influence create an opportunity for the creation of an environment in which everything *can* influence anything else.

In one example, a sample and hold circuit is firing random triggers back into the computer to randomly change the parameters of a single track. This track is also acting as the source signal from which the sample and hold circuit/module grabs its values. While the computer could use its system information to provide random seeding[2], I opted instead to manually assign seed values to reproduce desirable sequences while manipulating other non-random variables. The analog white noise generator gives a unified bridge between the two systems. The speed at which the sample and hold sends a trigger is determined by the chaos level of a different track's y-axis movement. In this situation, as the amplitude distribution becomes more chaotic, the more likely it is that the sample and hold circuit will fire. The result is a self-generating feedback loop that plays itself without, at least theoretically, ever repeating. The white noise generator in the sample and hold provides what amounts to be one of the truest forms of randomness available to humans.

Adaptations of GENDY

CSound

Tito Latini ported the microsound generator portion of the GENDY program to Csound[3] as an opcode based on Nick Collins' Gendy1 ugen in SuperCollider (Collins, 2011). Within this ugen there are nine parameters:

1. Amplitude
2. Amplitude Distribution Curve
    a. Linear
    b. Cauchy
    c. Logistic
    d. Hyperbolic Cosine
    e. Arc Sine
    f. Exponential
    g. Sinusoid
3. Duration Distribution Curve
    a. (same as above Amp Distribution Curves)
4. Amplitude Distribution Range
5. Duration Distribution Range
6. Amplitude Multiplier (scale)
7. Duration Multiplier (scale)
8. Minimum Allowable Frequency of Oscillation
9. Maximum Allowable Frequency of Oscillation

Using TouchOSC (Open Sound Control)[4] on an iPad, these parameters can be manipulated in real time, providing continuous and tangible control of the algorithm while also allowing for playback at either audio or sample rate over multiple tracks. This is a tremendously powerful implementation of the technique for not only its mobile-control, but also being provided with a set of highly customizable touch components to control the algorithm in detailed ways. Figure 2 shows in Csound how OSC is wired to the GENDY opcode for live control.

```
;Dynamic Stochastic Synthesis (after ;Xenakis) OSC Receive
Snippet
instr 333
kSlider 1 init 0.0
kSlider 2 init 0.0
…
kSlider 7 init 0.0
Stext sprint “%i”, $R_PORT
ihandle OSCinit $R_PORT
kAction OSClisten ihandle, “/gendyCSound”, “fff”,
            kSlider1,
            kSlider2,
            …
            kSlider7


;FREQ controls should never hit 0
;Params scaled for effectiveness in ;live performance
aout gendy 0.7, 0, kSlider3, kSlider4, kSlider5, kSlider1+0.01,
kSlider2+0.01, kSlider6, kSlider7
outq1 aout
endin
```

Figure 2. Code snippet of OSC control of the GENDY Csound opcode

In my piece *Dynamic Stochastic Control Voltage Generation (after Xenakis)* (McKemie, 2020), multiple tracks of this algorithm are wired into the modular synthesizer and used as control voltage (CV) and audio sources to modulate various aspects of the instrument. These include anything from oscillator frequency, filter cutoff, envelope parameters, and more. The program also receives output(s) from the modular synthesizer in the form of trigger and gate signals which are used to randomize the algorithm's parameters, thereby changing the signal output from the computer. This creates a feedback mechanism between the two halves of the system and injects the performance with an added layer of indeterminacy, pushing the player into a trinity of human – machine – computer with each providing their own unique contribution to the order.

Aesthetically, the purpose of this work was to expand upon the use of Xenakis' algorithm, all the while paying homage to the composer. It is certainly clear for those familiar with the audible characteristics of GENDY that it was not my intent to mask that quality, but rather, to embrace it as a necessary part of the work. This stands in contrast to other adaptations of the GENDY program, which have sought to either update the platform in which the program is running (Hoffmann, 2009) or to faithfully reproduce new works aligned with how Xenakis may have realized them (Doornbusch, 2002).

One of the primary tools used here to convert the audio output of the algorithm to control signals in this work is a bank of envelope followers, and my aim with the work was to see what musical opportunities might arise when they were used to control the available modules in the system. With that came the challenge of mixing seven different GENDY instances into one system to achieve a stereo output that did not devolve into constant noise. This was a particularly important point as a product of something akin to pure noise due to a failure in mixing or appropriate signal flow design would negate the purpose of this adaptation. With the goal being a continuation of the canon, I felt it important to treat the process much like a performative interpretation of GENDY, rather than a gutting of it (Di Scipio, 1998).

Python

The second case is a program written in Python to generate values from a random walk that are then translated into points of a waveform and extracted to a wave file. Unlike the previous example this approach focuses more on variability and change over time, although not in *real* time. This specific use also moves away from the parameters and mathematics found in the original GENDY algorithm to focus more on the concepts of random walks and how different arrangements of stochastic points can be used to construct a waveform.

Stochastic math is what Xenakis used to define form, content and events throughout all his oeuvre, with the GENDY program doing the same, albeit algorithmically and with synthesized audio (Xenakis, 1985). Still using multiple realizations of the algorithm as tracks as Xenakis did, this follows the same aesthetic goal of Xenakis' design of Dynamic Stochastic Synthesis as outlined in his essay. The process here is defining a range of time in seconds as audio sample rate, the number of integers calculated by the program, the bit depth of each point, and finally encoding these values to construct a waveform. A truly random distribution is heard as white noise, with the most orderly sequence being heard as a simple waveform. While he never explored the effects of sample rate changes on the algorithm, he was aware interested in it being a parameter of exploration in future work (Hoffmann, 2009).

The declaration of sample rate correlates to the distance between two points on the x-axis of the waveform, with a higher sample rate allowing for higher density and lower rates for lower density. The sample rates explored here ranged from only 10-300 samples per second. The program compiles the values and renders them to a wave file which was then placed into a DAW and sent through individual outputs in a DC-coupled audio interface. Figure 3 illustrates an example snippet of constructing a basic set of integers and encoding the waveform.

```
# Time Declaration snippet
# setup
obj = wave.open('sample.wav', 'w')
prob = [0.25, 0.75] # walk direction
start = 0
values = [start]
np.random.seed(123)
rr = np.random.poisson(1, length)
downp = rr < prob[0]
upp = rr > prob[1]

# random walk
for idownp, iupp in zip(downp, upp):
    down = idownp.any() and values[-1]      > -32767
    up = iupp.any() and values[-1] < 32767 val-
    ues.append(values[-1] – down + up)
…
# Encode waveform
for i in values:
  data = struct.pack('<h', i)
  obj.writeframesraw(data)
obj.close()
```

Figure 3. Python snippet random walk and wave encoder

In my piece *Time Declaration*, the multiple tracks produce a slowly evolving shape that unfolds over some period of time. In the development of this piece, the program was run numerous times with all tracks sharing the same starting point, but the random walk was seeded differently each time in order to produce variations in the direction of each track. To emphasize or clarify the form, some of the signals are treated as triggers. This is achieved by measuring the amplitude of the signal in relationship to a predefined threshold, and executing a trigger command each time the signal crosses this threshold. This approach is similar to the routing assignments in the Csound version. The effects of these triggers on the form of the piece can be heard in the changes to the density of events in the excitation of the spring reverb tank and digital noise bands.

Aesthetics

Striking features of *GENDY3* and *S. 709* are the wide bands of noise and aliasing, both of which are often considered undesirable in audio signals, are a prime characteristic (Serra, 1993). In both *Dynamic Stochastic Control Voltage Generation (after Xenakis)* and *Time Declaration*, leaving these artifacts intact and magnifying them where applicable, was by design as an homage to Xenakis' work. Within the discourse about the qualities and preferences between analog and digital audio, digital audio holds a special place in this series of examples. Having extensively worked with tape early in his career, Xenakis reveled in the unique characteristics and qualities that digital audio synthesis provides, and in his last two computer music works he is not shy about pushing these characteristics to new levels.

The goal here is not to simply replicate the mathematics or its results as Xenakis intended, but instead to evolve and adapt the technique in different ways. Specifically, being able to control the algorithm in real time, which was not possible in the late 1980s. This real time control yields tremendously fascinating results and the performance device itself is incredibly engaging for the performer. Furthermore, having direct live control over the characteristics and shape of an audio source on a microlevel is wholly unique even by present day standards. A modular synthesizer allows the user to sculpt audio in direct and immediate ways, all while patching the system to be self-generative should it be desired. With the combination of the modular synthesizer and real time control of the GENDY algorithm, this offers a tremendously deep level of control that moves beyond the plane of chaos and order of the algorithm alone.

Like Xenakis, there were other composers and engineers working in similar music domains that documented their work in papers, books, and manuals since at least the 1960s, and the concepts presented throughout this paper could easily be modified and applied to these works to extend beyond where they left off.

Future Work

This project currently uses the Daisy Seed board to house both the algorithm and the control signal generator. This board is capable of housing the functionality needed for the GENDY program as well as the digital to analog and analog to digital conversions at both audio and sample rate, as demonstrated above. Moving forward I intend to adapt the entire tech stack illustrated in this paper into a more-streamlined solution.

On the Python side I intend to build more of the GENDY algorithm in code and to create a standalone application to run in the command line. Furthermore, to introduce new libraries that will enable the program to be integrated with other pieces of software as well as the internet, and to explore deeper mathematical possibilities in the fields of physical modeling, artificial intelligence, and live coding.

Acknowledgments. Many thanks to Ryan Ross Smith for his encouragement and insight along the way.

References

Collins, N. (2011). In *Proceedings of the Xenakis International Symposium*. London, UK. Goldsmiths, University of London.

Di Scipio, A. (1998). In Symposium *Présences de Xenakis.* Makis Solomos [Ed.] *Centre de Documentation de la Musique Contemporaine et Radio France*: 71-84. Paris, FR:

Doornbusch, P. (2002). *Corrosion. Music for Instruments, Computers, And Electronics*. CD, EMF 43. Albany, NY: Electronic Music Foundation.

Hoffmann, P. (2009). *Music Out of Nothing? A Rigourous Approach to Algorithmic Composition by Iannis Xenakis*. Dr. phil. Dissertation. Berlin, DE: Technischen Universität Berlin.

Luque, S. (2009). The Stochatic Synthesis of Iannis Xenakis. In *Leonardo Music Journal 19*: 77-84. Cambridge, MA: MIT Press

McKemie, D. (2020). *D/A /// A/D*. CD, VCR-1001. Brooklyn, NY: Voltage Control Records.

McKemie, D. (2021). *Control Surfaces: Using the Commodore 64 and Analog Synthesizer to Expand Musical Boundaries*. SEAMUS 2021 Paper/Presentation. Charlottesville, VA: University of Virginia.

Serra, M. (1993). Stochastic Composition and Stochastic Timbre. In *Perspectives of New Music 31*/1: 236-257. Princeton, NJ: Princeton University Press.

Xenakis, I. (1985). *Arts/Sciences: Alloys*. Hillsdale, NY: Pendragon Press.

Xenakis, I. (1992). *Formalized Music: Thought and Mathematics in Composition*, revised ed*.* Hillsdale, NY: Pendragon Press.

[1] toplap.org/live-coding-control-of-a-modular-synthesizer-with-chuck/

[2] docs.python.org/3/library/random.html#random.SystemRandom

[3] csounds.com/manual/html/gendyc.html

[4] hexler.net/touchosc