Gendynish: Stochastic Synthesis on the Arduino

Andrew R. Brown
Griffith University
Brisbane, Australia
andrew.r.brown@griffith.edu.au

Abstract

The Gendynish algorithm is software that runs on Arduino-class microprocessors to produces sounds similar to Iannis Xenakis' dynamic stochastic synthesis. This article outlines the origins of stochastic synthesis and describes the Gendynish algorithm and associated electronic instrument development and performance outcomes. This algorithm takes a somewhat unconventional approach to Arduino audio. In doing so it demonstrates how once cutting-edge computer music practices are now much more accessible and that the spirit of creative expression through audio technologies remains alive and well.

Keywords

Sound, music, audio, software, hardware, computer, microprocessor, Arduino, performance, interactive.

Introduction

In the 1990s Iannis Xenakis developed a new probabilistic audio generation method called Dynamic Stochastic Synthesis that produced quite unconventional sounds. This process reflected his interest in "human intelligibility [of] temporal periodicity and the symmetry of the [waveform] curves" generated by digital devices [1: 289]. Two notable works using this technique were composed by Xenakis, *Gendy3* and *S.709*. Xenakis applied probability theories to the construction of both musical structure and the generation of sound. This synthesis technique applies constrained random processes to waveform construction. The amplitude and time position of each break point are varied at each cycle of the wave by a constrained random walk function. The original implementation of dynamic stochastic synthesis was in the *Gendyn* (GENeration DYNamique) program, written by Xenakis in BASIC with the assistance of Marie-Hélèn Serra and rendered audio files to disk [2].

Inspired by this process and the works the Gendynish project took advantage of the rapid advances in technology to produce a simile of stochastic synthesis on the low-cost Arduino microcontroller. The algorithm runs in real time and emulates the four-voice architecture of Gendy3 and outputs quadraphonic audio.

The author previously engaged with dynamic stochastic synthesis in 2004 coding the first real time, interactive, implementation of the process. The IDSS (Interactive Dynamic Stochastic Synthesizer) is an implementation of the dynamic stochastic synthesis algorithm in the jMusic environment (Java programming language).

IDSS

The IDSS system was presented and performed with at the Australasian computer music conference in 2004 [3] and the International computer music conference in 2005 [4]. As well as being real time, IDSS enabled some other extended features. Stochastic percussion sounds were achieved by automating a quick reduction in all random walk time-step amounts. This resulted in a rapid change from a bright complex timbre to a fixed and stable tone. The interpolation between breakpoints in the waveform could be switched from linear to cosine interpolation with the effect of smoothing the wave and controlling harmonic complexity. IDSS was designed as a real time instrument performed via MIDI controllers. The performers guide the software between sets of probabilities that determine the sonic microstructure.

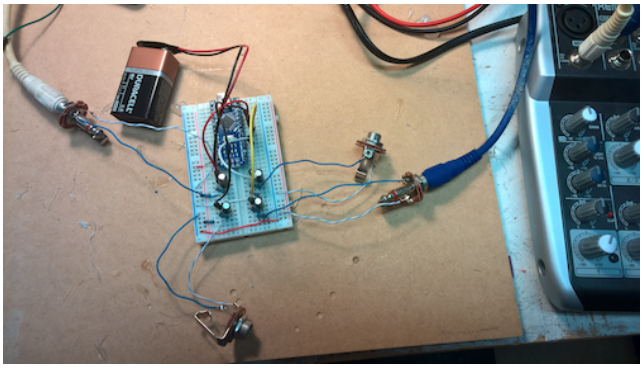

Figure 1: A Gendynish instrument prototype

Gendynish Algorithm

The Gendynish algorithm on the Arduino produces sounds similar to Iannis Xenakis' dynamic stochastic synthesis but is technically only 'in the spirit' of that process and is not a literal reconstruction, unlike IDSS. The Gendynish project arose as the aurthor was experimenting with generating audio directly from the pin outs on Arduino microprocessors, and noticed that some of the sounds were reminiscent

of those produced by the GENDY program. On the Arduino, a pin on the microprocessor produces a monophonic output, so four pins were used for the desired polyphony. Voltage output from a pin typically ranges from 0-5 volts and so scaling and DC offset was required to make the output suitable for connecting to a line level audio input. Figure 1 shows an early prototype with resistors and a capacitor on each of four pins to manage the output voltage.

The pin outs of an Arduino microprocessor lend themselves to pulse width modulation (PWM) output. This is what was used in the Gendynish algorithm and why is was only an approximation of the dynamic stochastic synthesis. To maintain the independence of the four pin outputs the timing of the pulse wave frequency was controlled within the main loop of the code, rather than by manipulating the microprocessor timers. This is computationally inefficient but allows more polyphonic flexibility. Because of the inefficiency a Teensy LC microcontroller was used for the performance because it is faster than many genuine Arduino microprocessors, yet software compatible. By probabilistically varying the frequency and pulse width of the pin output a surprisingly wide variety of timbral results were possible. Code for the Gendynish algorithm is available on GitHub.[1]

Performance 1 - A.001

The first performance of the Gendynish system was a composition by the author titled A.001 which ran in real time without human intervention and was played through a quadraphonic sound system. The work, performed in early 2018, had a number of contrasting sections and automatically moved between sections during the performance. A stereo recording of this piece is available online.[2]

Performance 2 – A.001.i

A second performance of the Gendynish system in late 2018 incorporated interactive control of parameters by the

Figure 2: Performing A.001.i

performer. The software was updated to include three rotary-push controls that adjusted probabilistic boundaries for frequency and timing and enabled manual sectional changes. LED lighting that synchronized with audio was also added for visual interest. The performance was based on the first composition, and thus titled A.001.i acknowledging the addition of interaction. A recording of this performance is available online.[3]

Conclusion

This project demonstrates that the past and present can come together in interesting and expressive ways. The Gendynish project draws inspiration from Iannis Xenakis and his innovative use of probabilistic processes to explore the nature of digital sound making. The project also utilizes accessible contemporary computing hardware whose cost to compute-power ratio would have astounded pioneers such as Xenakis. The value of open hardware and open source software, that is a feature of the Arduino project, is clearly demonstrated through outcomes such as these. Yet despite building on widely available platforms the Gendynish algorithm uses them in unconventional ways, continuing the artistic tradition of bending technologies to meet expressive desires.

References

[1] Xenakis, Iannis. *Formalized Music: Thought and Mathematics in Music*. Stuyvesant NY: Pendragon Press, 1992.

[2] Serra, Marie-Hélèn. “Stochastic Composition and Stochastic Timbre: Gendy3 by Iannis Xenakis.” *Perspectives of New Music* 31, no. 1 (1993): 236–57.

[3] Brown, Andrew R., and Greg Jenkins. “The Interactive Dynamic Stochastic Synthesizer.” In *Proceedings of the Australasian Computer Music Conference*, 18–22. Wellington, New Zealand: ACMA, 2004.

[4] Brown, Andrew R. “Extending Dynamic Stochastic Synthesis.” In *International Computer Music Conference*, 111–14. Barcelona, Spain: ICMA, 2005.

Author Biography

Andrew R. Brown is an active computer musician, computational artist, builder of creative software tools, researcher and supervisor of research students. He is Professor of Digital Arts at Griffith University in Brisbane and Program Director for the Bachelor of Creative and Interactive Media.

[1] https://github.com/algomusic/Gendynish/tree/master

[2] https://soundcloud.com/thejmc/a001

[3] https://soundcloud.com/thejmc/a001i-at-os11