ECHO
ORPHEUS INSTITUUT
a journal of music, thought and technology

Title: acrosynthetics
Subtitle: patch programmability, wavefolders and the demon at the end of the mirror.
Author(s): Pedro Ramírez-Agámez
Issue: #8
Publication date:
Review status:

Article DOI:

# acrosynthetics

*patch programmability, wavefolders and the demon at the end of the mirror.*

**Pedro Ramírez-Agámez**

ABSTRACT

This article discusses a practice I call acrosynthetics, developed on a hand-built Serge modular synthesizer, and the question that drove it: what actually separates analog from digital sound synthesis? The first section introduces the Serge and its resemblance to the analog computer. The second examines wavefolding, a circuit that continually adds harmonics upward and so presses against the ceiling of digital representation. The third develops a speculative framework drawn from cybernetics and Eugene Thacker's demonology, reading the wavefolder as an inverted, demonic filter. The final sections trace the practice itself across its analog, coded, and other implementations.

## Introduction

For the last four years, analogue and digital synthesis have been at the centre of my practice. I'm interested in them as tools, and equally in the ways of knowing they open through sound, and in the craft of their fabrication.

Although I've been exploring digital sound synthesis partly through sound coding environments for a longer period, my conversation with analogue machines is more recent, beginning with my postgraduate studies in Germany. I'm originally from Bogotá, Colombia, where, due to the purchasing power and currency value differences, the cost of hardware put analogue synthesisers, modular systems especially, largely out of reach for freelance working artists. When I moved to Europe for a master's degree, I had unbounded access to this equipment for the first time.

With it came a question that had been accumulating over years: what is the actual difference between analogue and digital synthesis? Is there something in the analogue domain that open-source sound coding environments cannot reproduce?

I decided to build a Serge Modular Synthesizer by hand, with the great help of supervisors and a small community of local builders. This came in parallel with self-study developing my practice in DSP, SuperCollider, and other computational environments for sound.

This essay traces some of what that process turned up by engaging with the Serge wave folding circuit as an exemplary case that displays the singular edges of analogue and digital sound synthesis.

Starting from a Thomas Ankersmit workshop[1], where he called the circuit "a sort of negative filter," I developed a framework that borrows from demonology, cybernetics and other speculative theory, taking the demon as a figure for the contrary, for mediation that runs in reverse. This is what I call *acrosynthetics*, which is at once a conceptual framework, a musical instrument and a set of compositions that demonstrate my perspective on building instruments as a form of inquiry.

# The Serge Modular Synthesizer

The choice of building a Serge Modular Synthesizer was both a practical and a conceptual one. Over the decades, and especially lately, this particular model of synthesiser has been growing in popularity for several different reasons. First, it presents itself with a consistent design philosophy replicated through a limited set of modules. And second, its inner-workings resemble and feel closer to working with voltage directly. Alongside this, the synthesiser has been kept alive through groups of hobbyists that by sharing modifications and circuit boards, have expanded the original designs and knit a tight community around it.

The Serge Modular Synthesizer was designed by Serge Tcherepnin at the California Institute of the Arts in the 1970s, drawing on an earlier tradition of instrument design that he traced back to Don Buchla, and the Pulsa collective.[2] It was originally designed as a DIY project, in which participants in workshops would have the capacity to assemble their own instrument.

Tcherepnin's particular contribution was an act of exposure. In Buchla's designs he found how much interesting circuitry was hidden inside each module: a voltage-controlled oscillator contained a sawtooth oscillator, a full-wave rectifier, a diode shaper, a comparator, all bundled and inaccessible. His response was to bring these components to the surface:

> "What struck me upon examining Don's designs, is how much very interesting stuff was hidden away in order to realize a given function . . . That's why from the beginning, I determined to make sub-functions such as comparators, rectifiers, waveshapers, bi-directional routers, and peak and trough tracers available for use by the musician interested in sonic DIY."[3]
>
> *Serge Tcherepnin*

He compared this approach to RISC architecture in computing: Reduced Instruction Set, where fewer, simpler operations give the user more expressive control than a smaller number of complex, pre-packaged ones. The result is an instrument that resists fixed description. A module like the Dual Universal Slope Generator can be a clock, a low-frequency oscillator, an audio-rate oscillator, a slew generator, a low-pass filter, an envelope follower, or a pulse divider, depending entirely on how it is configured. Tcherepnin called this *patch programmability*.[4] The character of the instrument is never fully determined by the builder: it continues to be built each time it's patched.

---

[1] Thomas Ankersmit, pers. comm. during a Serge Modular Synthesizer workshop, Kunsthochschule für Medien Köln, 12 April 2023.

[2] Meyer, Chris. 2018. "The PULSA." Learning Modular + Alias Zone, 19 August. Accessed March 2025. https://www.patreon.com/posts/pulsa-20848551.

[3] Meyer, Chris. 2018. "Conversations with Serge Tcherepnin." Learning Modular + Alias Zone, 31 July. Accessed March 2025. https://www.patreon.com/posts/conversations-20421552.

[4] ibid.

The Serge also refuses to separate audio signals from control voltages. In most synthesiser designs, a clear hierarchy separates the two: control signals govern audio signals. In the Serge, both are treated as voltage. This is reinforced by the banana jacks, borrowed from laboratory equipment, that encourage expansion and reconfiguration at every patch point through their capacity to be stackable. The instrument inherits the analogue computer's architecture and applies it to sound.

The extreme of this flexibility surfaces in the slope generators. A slope generator configured as an oscillator can be simultaneously reconfigured by its own input signal to act as an envelope follower, translating incoming amplitude into a control voltage. Connected back into the system, it becomes an ear: a module that monitors energy and responds. The system can be made to listen to itself.

On a different level, the Serge synthesiser bore a strong resemblance to the analogue computers of the 1950s, which preceded the von Neumann architecture that underlies all modern computing. Where modern computers operate by executing sequences of discrete instructions stored in memory, analogue computers worked through continuous voltage: they were built to model physical systems, to solve differential equations, to simulate the behaviour of fluids or mechanical structures in real time. As Robin Mackay writes in his liner notes for Florian Hecker's *Acid in the Style of David Tudor*: "The programmable patch-panel interface of the classic mid-1950s analog computer is already recognisable as the basis of the analog synthesisers that pioneers such as Moog and Buchla assembled in the 1960s, which in effect endow such modular signal-processing devices with a sonic 'realiser', putting the virtual laboratory of the analog computer in the service of sound production." He cites Dan Slater's observation that "the electronic analog computer is in many ways the aerospace equivalent of the analog modular music synthesizer."[5] The patch panel, the voltage-controlled functions, the absence of stored memory were inherited directly from the computing machines that preceded the digital era.

Where Moog's design intent was to make instruments with keyboards, continuing the tradition of musical instruments, and Buchla's focused on expressive playability through crafty and novel interface designs, the Serge kept its raw, grid-like design and low-level exposure of sound/voltage processing.

That's what drew me to it and made me choose it as the dissecting table for exploring the inner workings of making sound with voltage. Legible circuit logic and direct material exposure to electricity. Sub-functions exposed and connections available for direct use.

# Waves and folds

Of all the mechanisms the Serge exposes, the ones I've been most drawn to are waveshaping and wavefolding, and specifically the way they reveal a limit that runs through the difference between analogue and digital synthesis.

Waveshaping transforms an input waveform by running it through a transfer function: polynomial distortion, amplitude squaring, clipping. Wavefolding does something more particular: it amplifies a signal until it exceeds a threshold, then folds it back on itself rather than clipping it. Each fold adds odd harmonics: third, fifth, seventh, ninth. In the usual topology of a synthesiser, a filter takes a rich signal and attenuates its upper harmonics. Wavefolding inverts this logic: each fold generates new harmonics, inserting energy into the spectrum. Each fold functions like a mirror: reflected waveforms stack onto each other, multiplying complexity at every threshold.

The specific wavefolder I've been working with is the Lockhart cell, which appears in the central section of the Serge Wave Multiplier module. Its behaviour can be modelled mathematically through

5 Mackay, Robin. 2009. "Climate of Bass Hunter." Liner note for Florian Hecker, *Acid in the Style of David Tudor*. Editions Mego, eMego 094, CD. Citing Dan Slater.

the Lambert W function[6], a transcendental equation that gives the folded output voltage as a function of the input and the circuit's physical constants. The Serge Wave Multiplier cascades several of these cells in sequence, each one folding the output of the previous.

With regard to my inquiry into the analogue/digital tension, this was a really interesting point of exploration precisely because it exposes the limitations of both media. Because wavefolding continuously generates odd harmonics, it is always pressing towards higher frequencies. In the analogue domain, this is simply what the circuit does and carries no penalty: the harmonics keep climbing and nothing reflects them back. In the digital domain, there is a ceiling: the Nyquist frequency, which sits at half the sampling rate and marks the highest frequency a digital audio system can represent faithfully. Above this threshold, frequencies are reflected back downward, appearing as "ghostly" tones at lower frequencies. The Nyquist frequency acts as a mirror: every harmonic the wavefolder generates above it re-enters the audible spectrum as an alias. This is the characteristic sound of digital wavefolding pushed past its limits: a shimmer of spectral ghosts, frequencies that belong to reflections rather than sources.

Naturally, there are many ways to prevent aliasing from happening in more resource-expensive algorithms and tricks, and it's something we'll touch later in the article.

The practice of feeding back and heavily modulating the Triple Wave Shaper and Wave Multiplier modules in the Serge has become a flagship sound in the community, a technique that differentiates the instrument from other modular systems.

By way of demonstration, the interactive model below runs six Lockhart cells in cascade, self-oscillating through feedback. Although it uses ADAA anti-aliasing to mitigate the "ghostly" tones, they are still inherent in the soundmark of the digital.[7]

> *[Interactive media — cascade of six Lockhart cells, self-oscillating through feedback, with ADAA anti-aliasing. Available in the ECHO platform version of this article.]*

# Reflections on cybernetic ontologies

During my studies, while serving as a student assistant in a seminar on analogue synthesis, my professor hans w. koch talked about the limitations of digital vs. analogue synthesis in a way that captivated my curiosity by asserting: computers cannot predict the future.[8]

To elaborate: in the analogue domain, feedback is instantaneous, as voltage cycles through the circuit continuously, with no imposed ordering between past and present samples. In the digital domain, however, the calculation of each sample happens in sequence, and the output of a given sample cannot feed back into itself before the next one is computed. This imposes a direction onto time that analogue circuitry simply doesn't register. Even when the block size is reduced to a single sample, the causal ordering remains: each computation takes the previous sample as given, one step behind the loop it's trying to close.

The consequences of this process can be minor or imperceptible to the untrained ear, but the processes happening at this stage are still substantially different from the sonority of analogue

---

[6] Esqueda, Fabián, Henri Pöntynen, Julian D. Parker, and Stefan Bilbao. 2017. "Virtual Analog Models of the Lockhart and Serge Wavefolders." *Applied Sciences* 7 (12): 1328. https://doi.org/10.3390/app7121328.

[7] The web audio implementation uses Antiderivative Anti-Aliasing (ADAA) based on the Lockhart cell model described in Esqueda et al. 2017 (see previous note).

[8] hans w. koch, pers. comm. during a seminar on analogue synthesis, Kunsthochschule für Medien Köln, summer 2024.

feedback experiments. The sampling mechanism, the discretisation happening at the level of DACs and ADCs may lead to delays, phase mismatches or other accidents that can alter the sonic capacities of the circular feedback topology. Analogue circuits introduce phase shifts of their own, of course; what differs is the imposed ordering that produces them.

## Cybernetics and "demontology"

When trying to grasp the state of these low-level voltage and digital processes, the topic of cybernetics became an unavoidable step to dive further. Firstly, by examining the crossings between cybernetics and theories on the nature of time, and secondly, through an observation of an antagonistic comparison between wavefolders and filters, and how it relates to Eugene Thacker's perspective on what he calls a Demon Ontology. Here the "demon" would not only have resonances with thought experiments such as Laplace or Maxwell's demons, but also as an ontology of difference.

Norbert Wiener's *Cybernetics* (1948) described the discipline as a multidisciplinary perspective on self-regulating mechanisms through the use of feedback. These processes were developed alongside information theory, with Weaver and Shannon. Positive feedback was presented as a self-growing, exponential process. Counter to that, through negative feedback, processes of regulation were theorised. Wiener was already thinking about negative entropy as the principle that allows organised systems to resist disorder. He argued that information processing and negative entropy are two sides of the same phenomenon: a system that receives, processes, and acts on information is, in effect, working against the thermodynamic current. The CCRU's cybernetic theory, and Nick Land's writing in particular, extend this observation into a speculation about temporality. If entropy defines the direction of time, increasing disorder marking the future from the past, then a system that locally reverses entropy, that becomes more organised through its own operation, describes what Land calls *extropy*: a negative temporality that resists time's arrow from within.[9] The feedback network, by organising itself through self-listening, operates in this territory.[10]

This reversal is reminiscent of Maxwell's Demon, a thought experiment that imagined a demon sorting particles between two chambers, locally reversing entropy without expending work.[11] It was about information: what the demon uses is not force but knowledge of individual molecular velocities, a local reversal of thermodynamic direction through intelligence alone.

Eugene Thacker's work on demonology arrives at a related structure from a different direction. In later Christian theological tradition, angels are agents of mediation, carrying divine messages and serving as bridges between the godly and the mundane. The devil, himself an angel, operates through inversion of this function. As Thacker writes, citing the theologian Maggi: "if angels are messengers of the word of God, then whose words do demons speak? For Maggi, demons do not deliver anyone's message; they do not speak anyone's voice." The demon, in the extreme case of possession, becomes "the communication of noise, the mediation of nothingness." Their mediation takes the form of hyper-communication (babbling, speaking in tongues), the unspeakable body, or silence: demons

[9] Land, Nick. 2024. "Extropy." In *Xenosystems*, 224. Passage Publishing.

[10] Watson, Stu. 2024. "Cybernetics and the Evolution of Philosophy." *Do Not Research*. Accessed March 2025. https://donotresearch.substack.com/p/stu-watson-cybernetics-and-the-evolution.

[11] Maxwell's demon: a thought experiment in which an intelligent entity guards a partition between two chambers of a gas container, allowing only fast-moving particles through in one direction and slow-moving ones in the other. Over time one chamber heats and the other cools, apparently violating the second law of thermodynamics without expenditure of work by the demon itself.

"speak in order to subtract presence from the world." Where angels ensure clear communication, demons insert noise into the system. They mediate, but in a negative or contradictory way.[12]

The waveshaper/filter distinction maps precisely onto this structure. A filter operates by subtraction, selecting which frequencies to attenuate, reducing the harmonic content of a signal. A waveshaper operates by addition, generating harmonics that were absent, folding the signal through non-linear functions to produce complexity where there was simplicity. The filter and the waveshaper share the same domain while performing inverted operations: one reduces harmonic content while the other generates it. This inversion helped me to frame my practice with reinforced wavefolders as some sort of *demonic* synthesis: a practice built on recursion of transfer functions, of harmonic generation and non-linear deformation.

Working in adjacent territory, the South Korean computer musician Ryu Hankil writes in the liner notes for his album ③:

> "The expression 'demon' has negativity. What we don't know is the demon's trick or evil power. At the same time, in order to think about the unknown, it is necessary to assume the existence and operative mechanism of the transcendental energy whatever it is expressed in. The negativity represented as the demon here is irrational and fictional. . . . But rather than identifying the demon as a being of religious negativity, it would be better to reconfigure it as something like Prometheus, who brings fire to humans and pays the penalty of eons. It is because God never wants humans to understand the mechanism that makes God's powers work. As Paik Nak-jin has said, this is the basic framework of the narrative of the creation of the world and humans, which is common to all myths."[13]
>
> *Ryu Hankil*

He extends this to the scientific tradition: the phrase "demonic element," as he uses it, evokes thought experiments like Laplace's Demon, Maxwell's Demon, Descartes' Demon, Darwin's Demon, hypothetical forces theorised to function before they could be observed, whose eventual confirmation scientists sometimes describe as an exorcism succeeded. He also notes what is better called a fictional etymology of *impulse*: *imp* (a small demon, and a word related to the grafting of plants) embedded inside *impulse*. The system above is inert until it receives one.

The wavefolder turns out to be a productive site for all of these questions. Its constant generation of upper harmonics presses against the ceiling of digital representation, making those limits audible. Its inversion of the filter's logic makes it a sonic figure for the demonic ontology Thacker describes. And it is a mechanism that can only be fully understood through building and patching.

## Zeros and ciphers

As we're talking about self-recursive wavefolders, it's unavoidable to talk about aliasing, and I found it curious how there are techniques of anti-aliasing that are consonant with the line of speculative thought we have started bringing up here. An overall explanation of aliasing goes beyond the scope of this text, but in oversampling techniques, zeros are inserted between samples before non-linear computation, creating space for the wavefolder's harmonics to exist above the Nyquist mirror rather than folding back through it. The inserted zeros allow the non-linear function recursion to operate by

[12] Thacker, Eugene. 2011. "Occultural Studies 3.0: Devil's Switchboard." Metamute, 26 May. Accessed April 2025. https://www.metamute.org/editorial/occultural-studies-column/occultural-studies-3.0-devils-switchboard.

[13] Hankil, R. 2022. Liner notes on ③, self released.

giving space for foldings to happen. This recursion, as we explored before, possesses this interesting characteristic of chaotic emergence of sounds.

Curiously, and deeply ingrained in the CCRU's arsenal of metaphors, is the cipher: the numeral zero and its history in mathematics. The word "zero" comes from the Arabic *ṣifr*: "empty," "void," which became "cipher" in Latin. Medieval European mathematics met zero with resistance: if one represented God, zero could only represent its absence. It also introduced the concept of negative numbers, foreign to Roman arithmetic, and with it the unsettling idea of debt. In Portuguese, *chifre* (close in sound to "cipher") means "horn," an attribute of the devil.[14] Of course, the implications of these coincidences go nowhere but to entertain the imagination of the reader when considering what happens when the inquiry into sound gets microscopically smaller, and how the fabric of "reality" starts popping with entertaining mysteries.

# Praxis: patching the synth

Alongside the research and after finishing the construction of the synthesiser, I started building a documentation methodology and a performance practice that I called *acrosynthetics*. From the Greek ἄκρος: edge, extreme, outermost, *acrosynthetics* is organised around feedback cells: small clusters of modules routed into each other, connected into larger networks organised by function, i.e., the *demonic core* (a wavefolder feedback loop), *intelligence* cells (weighted configurations of the Analog Shift Register that produce rungler-like behaviour), and *ears* (envelope followers that track the energy of the system and feed it back as modulation). Connected together, they constitute a general patch, one that changes character each time it is assembled by emphasising different weights or areas.

The instrument consists of nineteen modules assembled across two panels. The patch began around a single feedback entanglement between the Wave Multiplier and the Resonant Equalizer, and grew outward as new cells were added. Rather than mapping the full web of connections, I broke the documentation into functional clusters, a diagram that describes the organisation without pretending to capture the behaviour.

In the scope of the present article I prepared two patches of recursive wavefolding cells. Firstly, starting on different combinations of each of the three sections of the Wave Multipliers (WM) module in combination with the Resonant Equalizer (RESEQ) and later, the coupling between the Triple Waveshaper (TWS) and the Phaser (PHA). These patches not only demonstrate the sonic palette I'm after but also the acrosynthetic dynamics of the instrument, based heavily on the different ways the same modules can be used for different purposes due to their patch programmability.

> *[Audio media — two Serge patches: Wave Multiplier with Resonant Equalizer, and Triple Waveshaper with Phaser. Available in the ECHO platform version of this article.]*

# Going digital: writing code

Trying to find parallels in my sonic coding practice, I've been working over the last two years on different implementations of this feedback topology in the SuperCollider programming environment which, at the time of the original draft of this article, was limited in its capacity to embrace single-sample feedback loops. Since then, several additions have facilitated my research, namely the

---

[14] Plant, Sadie. 1998. *Zeros and Ones: Digital Women and the New Technoculture*, 53. London: Fourth Estate.

integration of DynGen[15] and FaustDef[16], which allow the user to dynamically define new unit generators at runtime using either EEL2 or Faust programming languages, with the delight of allowing feedback loops on a per sample basis akin to gen~ found in Max.

In the following examples, the reader will find two different sketches: One with an oversampled Serge Folder uGen, using the traditional method of defining the block size to `1` and `LocalIn.ar()` and `LocalOut.ar()`. And another version where most critical DSP has been defined within DynGen, making sure there's no extra space than the tight-knit single sample knot.

In the same vein as the demonstration in the analogue section, these experiments attempt to digitally recreate a *demonicCell*, which grows from the coupling between a Wave-Folding function and a Filter and/or Phaser, from simple, to more crafted iterations.

```
s.options.blockSize_(1);
s.reboot;

(// SinOsc phased modulation with feedback, SergeFoldOS in the fb loop

Ndef(\demonCell_04, {
    var snd, local, local1, trig, fbSend;

    local = LocalIn.ar(2) + Impulse.ar(0);
    local1 = (LPF.ar(local[0], 30) * 1000);// lowpass the feedback to get a clock signal

    trig = PulseDivider.ar(local1.abs > 0.2, 7); // use a threshold to trigger the pulse
divider

    snd = SinOscOS.ar(    // 5 OS osc,
        {1600.0.rand}!5 * (1 + (local * MouseY.kr(0,5))),  // fb modulates a base freq
        Latch.ar(local, trig).lag(3) % 2pi, // sah modulates phase
        4   // oversample factor
    );
    snd = snd * MouseX.kr(-10,10); //gain, invert polarity
    snd = Splay.ar(snd.scramble);
    fbSend = SelectX.ar(
        LFNoise0.ar(0.5).linlin(-1,1,0,1),//sah selects between two wavefolders
        [
            SergeFoldOS.ar(snd, oversample: 4), //oversampling  4x
            BuchlaFoldOS.ar(snd, oversample: 4)
        ]
    );

    LocalOut.ar(fbSend);
    LeakDC.ar(snd.tanh);
}).play;
)
```

Amongst many iterations of the experiments, I found the example of the code above produces sounds in a similar direction to the analogue patches, this time by phase and frequency modulating five oversampled sine oscillators and alternating between oversampled models of Buchla and Serge wavefolders amongst other tricks.

Lastly and continuing from the first embedded example of the *demonic Cell* featured above, the following example features a model of four different demonic cells coupled together dynamically, with their parameters being controlled by a one-to-many relationship on the faders, with the capacity of allowing a shuffle of the weights in the parameter of each cell.

> *[Audio and code media — four coupled demonic cells under one-to-many fader control. Available in the ECHO platform version of this article.]*

---

[15] Scheiba Dennis, DynGen. https://github.com/capital-G/DynGen

[16] Scheiba Dennis, FaustDef. https://github.com/capital-G/sc_faust

## Thinking further: top-down digital representation

Both implementations approached the problem from the bottom up: model the mechanism, reproduce the circuit function by function, and arrive at a similar sound through the same operations. In doing so we exposed aliasing, block latency, and single-loop constraints, but also the possibilities of digital multiplicity.

A different approach would come at the matter the other way around: from the way the *demonic cell* makes itself audible, and how in doing so it makes evident its own internal logic of sound composition by mirroring my aesthetic preferences.

I have been exploring this through sampling and contemporary neural network frameworks: the variational autoencoder found in IRCAM's RAVE model,[17] or a fast latent diffusion model on Stable Audio 3 trained with a custom LoRA such as Underfit.[18] That work remains outside the scope of this article.

However, since these methods require a specific imprint and profile drawn from a significant corpus of audio samples, I came to the idea of reaching it by a more streamlined and direct path: a web-based instrument that documents my practice while interacting with the user's native browser experience.

That instrument is *acrosynthetics: cascade*

Instead of only replicating the circuits, the *acrosynthetic* approach seeks to stretch the ways of thinking about sound synthesis by testing the plasticity of concepts and their epistemic capabilities across different media. What began in the low-level approach of the Serge Modular Synthesizer and its patch programmability became the igniter of a myriad of techniques and thoughts around instrument making.

The practice of *acrosynthetics* is not resolved in any single medium but continues across them, and from each, new lines of reflection radiate outward and inward through their own Nyquist frequencies, the limits of their faithful representation, expanding the edges of the known.

[17] Caillon, Antoine, and Philippe Esling. 2021. "RAVE: A Variational Autoencoder for Fast and High-Quality Neural Audio Synthesis." arXiv preprint arXiv:2111.05011. https://arxiv.org/abs/2111.05011

[18] Dadabots. n.d. "Underfit." GitHub. Accessed July 2026. https://github.com/dada-bots/underfit