Functional Iterative Sequence Warping: An Open Framework for Exploring Warped Ramps, Exponential and Euclidean Rhythms, and Multi-Part Swing

Morgan Jenks
Texas A&M University
morganjenks@tamu.edu

Jeffrey M. Morris
Texas A&M University
morris@tamu.edu

ABSTRACT

This paper presents an open framework that uses a uniform approach to deviate from regular metrical grids. Building on the concepts of general circle maps and functional iteration synthesis and decoupling the timing ramp from event patterns, this approach encapsulates the circle map's modulation term as a variable for sequential transformations, here with a small set of example modulations that can be mixed freely by a text-command syntax. The method is applicable for conventional rhythmic values, compound meter, and poly-rhythmic swing, as well as arbitrary step counts and more extreme manipulations of time. We present a variety of musical applications.

1 Background

We draw primarily from two overlapping practices in computer music. The first stems from J Dilla [1] who pioneered techniques using drum machines like the Akai MPC series, creating humanized rhythms by embracing rhythms outside the homogeneous grid. The second thread diverges from the first and comes from electronic musicians exploring mathematical approaches to rhythm, colloquially using the term *exponential rhythms* to describe the overwhelming accumulative effect of hand-warped ramps and sampler loop-point manipulation [2], often appearing in single instances and obfuscating the underlying metrical structure. Interest in using exponentially warped rhythms that are periodic within a metric cycle appeared more recently but via freely drawn curves [3]. The dissemination of the techniques includes online tutorial videos from Mr. Bill [4], FITNESSS / A S D [5], and Joshua Eustis of Telefon Tel Aviv [6]. The artists and many others have explored varying formulations of swing rhythms [7]. In a separate strain of developments, Euclidean rhythms emerged as a way to generate rhythms and reproduce many rhythms found in culturally significant music around the world [8].

Sample-accurate step sequencing typically relies on driving musical processes with a sawtooth wave ramp, as recently detailed in depth by Taylor and Wakefield [9]. Many, including the artists discussed above, have found it useful to manipulate the clock, and further examples of phase modulation for timing purposes are found in circuit bending and related hardware hacking [10], in commercial modular synthesizers [11], [12], [13], mobile applications [14], and DSP scripting environments like BitWiz [15]. The BitWiz bytebeat interpreter for iOS exposes the sample rate as a variable that can be shaped, and Morris documented some uses of a warped derivative clock in bytebeat coding [16]. Certain compositions have also explored this notion, such as [17], designed to challenge the authority of the clock.

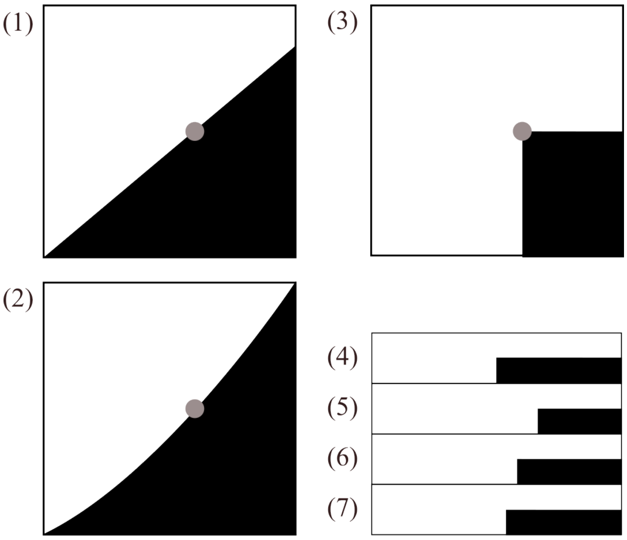


Figure 1. A table comparing (1) linear timing skew, (2) power curve, (3) Euclidean quantization of 2 pulses out of 5, with dots indicating the upbeat of the swing tuple. Also, note that the euclidean method is not continuous, and the values converge on 0.5 as the euclidean sequence grows in length. this figure shows (4) an even subdivision at length 2, then (5) length 3, (6) length 7, and (7) length 13.

Our work connects iterative wave generation and phase modulation with exponential rhythm and swing, so swing can be expanded on a continuum into truly exponential rhythmic gestures in a way that intersects with and can be combined with Euclidean rhythms. Our contribution makes this technique applicable in a procedural way and merges it with the parallel increasing trend of interest in more complex meters and tuplet swing, combining complex polyrhythmic concepts with electronic production methods.

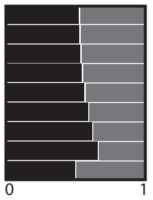
2 Musical Concept

We are working with a timing ramp for an entire phrase or loop. Given a phase-locked, 0-to-1 ramp driving a sequence, an exponent applied to this ramp will bend it to be convex or concave (shifting events earlier or later) while keeping 0 and 1 in place. Further, a sequence of 8 equally spaced events can be converted into 4 pairs of swinging eighth notes, their corresponding ramp segments can be scaled to the range 0–1 and have an exponent applied, and each pair can be repackaged into the full ramp with the result of shifting every second step in time. This allows for realizing any type of swing or anti-swing (like the Scotch snap), including tuplet-based swing, as well as morphing continuously among them.

Euclidean rhythms can also realize tuplet-based swing by distributing 2 events evenly over any odd number of subdivisions of a beat (not the whole sequence). While there are several approaches to achieving Euclidean rhythms, many produce results in rotated forms relative to each other, with some being more left-packed, others right-packed, and yet others with events shifted toward the center. Whereas Toussaint's approach yields many rhythms found in musics worldwide [8], the approach based on the modulus operation consistently yields swinging rhythms, such that $E(2, n)$ will yield tuplet swing with any odd value of n as graphed in Figure 2. Toussaint has notated Euclidean rhythms in the form $E(k, n)$, where k is the number of events and n is the total number of subdivisions in the pattern. Whereas the exponential approach allows continuous morphing, the discrete Euclidean algorithm allows rhythms to evolve incrementally between paying just the downbeat ($E(1, n)$) to playing every subdivision ($E(n, n)$, with two-part swing being just one possibility in a larger related set of rhythms. This ability alone allows for sweeping through more- or less-dense rhythms based on the same number of subdivisions.

Combining Euclidean rhythms with exponential curves brings even more possibilities. By describing playing every eighth note in every swung pair as $E(2, 2)$ and achieving swing by applying an exponent to curve the ramp over that period of time, this swinging pattern can evolve not only via the exponent by also by changing the Euclidean rhythm articulating it. For example, changing $E(2, 2)$, playing every eighth note in a swung pair, to $E(3, 3)$ yield a three-part swing, and this pattern can be continued for any amount of *multi-part swing* (rather than the ordinary 2-part swing).

This highlights an intersection between exponential and Euclidean approaches, each of which can expand in multiple directions. These approaches can be applied to the whole or subsets and in combination, and, since each stage of rewriting the ramp (stored as a table) keeps the ramp in the range of 0–1, in most cases, the transformations can be done in any order (they are commutative). Figure 2 shows several types of tuplet-based swing as well as a swing proportion based on the golden ratio (ϕ), in terms of the percentages, exponents, Euclidean rhythms, and tuplet subdivisions that could each be used to achieve them. Figure 3 demonstrates how a linear ramp stored as a table can be transformed to realize a variety of outcomes by combining exponential and Euclidean approaches. Figure 4 shows an example beat pattern with each musical voice using a different combination of techniques and different parameters, including an early backbeat (after J Dilla) but here achieved with a subtle exponent which allows notes to creep forward over time instead of statically rotating them, which would also be possible. It additionally features a quintuplet swing hi-hat cymbal pattern common in J Dilla's productions, a Latin American *cinquillo* pattern played by a shaker ($E(5, 8)$, with rotation depending on the appraoch used), and in the castinet, a concave exponential burst after the technique used by Mr. Bill but with rests induced by a Euclidean rhythm instead of playing every warped subdivision.



Percent	Exponent	Euclidean	*n*-Tuplet
53.3%	1.10	*E*(2,15)	15-tuplet
53.8%	1.12	*E*(2,13)	13-tuplet
54.5%	1.14	*E*(2,11)	11-tuplet
55.6%	1.18	*E*(2,9)	9-tuplet
57.1%	1.24	*E*(2,7)	septuplet
60.0%	1.36	*E*(2,5)	quintuplet
61.8%	1.44	n/a	φ (golden ratio)
66.7%	1.71	*E*(2,3)	triplet
50.0%	1.0	*E*(2,2)	straight

Figure 2. Several proportions of swing and their corresponding values in reference to percentages, exponents, Euclidean rhythms, and tuplets.

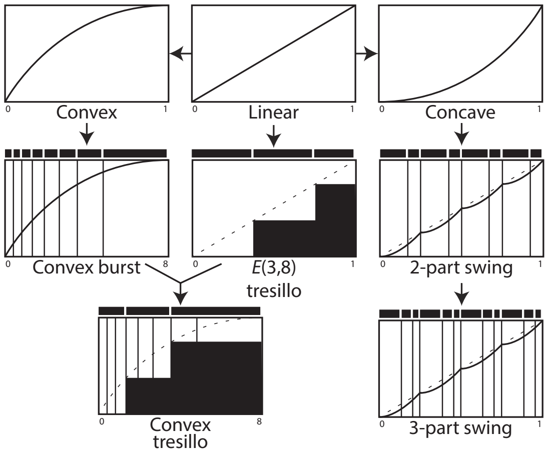


Figure 3. A linear ramp and possible paths of transforming it by applying exponents to the whole or parts of the ramp, Euclidean rhythms, and combinations of these. Resulting rhythms are notated along the tops of tables in piano-roll style.

Toussaint's work in 2005 revealed how the Euclidean algorithm, originally from particle physics via Bjorklund's [18] research, could generate rhythmic patterns found worldwide. This algorithm provides an elegant method for specifying micro-timing rhythmic deviations, especially given the algorithm's phenomenologically salient output of macro-scale rhythms. Considering the notion of toggle rhythm swing, the $E(2,\text{'odd-}n\text{'})$ rhythmic unit, as shown in Figure 1, and how these deviations emerge from the algorithm's attempt to distribute events with optimal evenness in quantized space, we arrived at how the resulting patterns create an asymptotic approach to an even subdivision, and drew the connection to modulating the timing ramp's progression through steps, rather than the pattern the steps themselves.

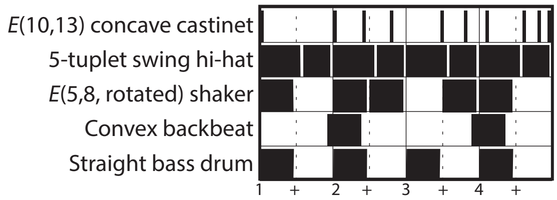


Figure 4. A single beat pattern displayin in piano-roll style with each musical voice using a different combination of techniques and different parameters.

3 Framework

In the realm of cyclic calculation, Essl's General circle maps provide a unified mathematical framework connecting various oscillatory synthesis methods, including waveshaping and phase modulation [19]. A related technique, functional iteration synthesis (FIS) involves iteratively applying nonlinear functions to generate sound [20]. Both of these methods have been discussed in terms of audible frequency range synthesis, but they can also provide a systematic way to generate and manipulate timing sequences through iterative phase functions.

Discussions of the circle map and FIS generally assume that the same function f is used consistently throughout the iterations of the map. The application of these two techniques also has historically involved dynamic nonlinear formulas, where the next sample is dependent on the previous sample value. The general forms of these two methods, however, do encompass closed-form and linear equations. Our work presents a subset of FIS and generalized circle maps that utilizes an arbitrary stack of modulations with an index term in order to more broadly incorporate the practical usefulness of various types of functions into generating warped sequencer ramps.

$$\Theta^{n+1} = f_n(\Theta^n, p_n) \tag{1}$$

where $f_n \in \{f_0, f_1, \ldots, f_k\}$, $p_n \in \{p_0, p_1, \ldots, p_k\}$, p is a payload of arguments for the current function f, n is the iteration applied to the ramp from 0 to k, Θ may be a finite sequence of samples that are phase bound, wrapping from the end of the sequence back to the beginning, and the functions f may incorporate the entirety of the sample sequence Θ at the previous iteration to return the new values of the next Θ.

Our implementation elaborates this general form with an iteration over potentially different functions on each pass of n. This modular, or phase-locked, formulation, in combination with a serial iteration approach opens the doors to computing exponential and Euclidean swing within the same function and opens the space of possibilities to other approaches.

While dynamic and chaotic systems can theoretically fit within this implementation framework, their musical application presents challenges in terms of predictability and control. This area represents a potential direction for future research in extending the framework to incorporate more complex mathematical systems.

4 Implementation

The formulation of the approach is intentionally generic to accommodate a wide variety of applications. The presented implementation in Max/MSP 8's ES5 uses a higher-order function to wrap individual iteration functions. The provided implementation in Python 3 simply calls sequential iterations in main. This section discusses some details of the implementation that are useful for making rhythms, and how each of them fits into the calculation of the ramps. Our formulation places handling the particulars of any method (including integration between iterations and recursion) within the scope of the f_n method from Equation 1 at each iteration.

4.1 Interpreter Stage for Commands

For the purpose of interacting with the model from Max, we have developed a simple textual command syntax with the form

$$f\ [x_a, \ldots, x_n] : \tag{2}$$

where f is a string denoting the method, and the terms in brackets are optional arguments that may be defined for a specific method. Commands terminate with a colon to accommodate arbitrary argument counts.

We have found text to be an effective format for rapidly creating and exploring different combinations of methods, and note that further development of command parsing for the system is likely be a fruitful path for future work (notwithstanding a possible object based GUI implementation). While recursion can be implemented within the method itself, compound and looped calls to methods can be used to generate metrical structure on top of the features generated by a single method. Some suggestions for musically relevant ways to notate duration and subdivision include Barlow's prime factorization $(1{\times}3{\times}2{\times}2)$ [21], and Gotham's recursive metrical structures $((3,3,2),(2,2))$ [22], and Morris's Duotresation [23].

4.2 Phase Arguments

For nearly all of the sample methods provided here, the computation of each modulator is run over a finite phase in real numbers. We implement a beginning phase and ending phase in a normalized 0–1 range, but not bound in that range. Using phases beyond the normalized range enables computing dynamic formulas over multiple periods into the buffered set of values (for example, 10 passes over the phase-space, beginning phase -9 and ending phase 1), due to the phase-wrapped nature of the model.

Also, of particular note in the context of sequencer ramps, we have found shorter phase ranges useful for operating piecewise on a discrete subdivision within the sequence as shown in Figure 5. Multiple passes of mutation - each operating on a single tuple - can be applied, each with their own way of incorporating the previous state of the ramp buffer.

$$\theta_i^{n+1} = f(\theta_i^n), \quad \forall i \in \{b, b+1, \ldots, e\}, \tag{3}$$

where: i is the range of samples from beginning to end. θ_i^{n+1} and θ_i^n are the individual values at index i for the next state and current state of the ramp, and where

$$\theta^n = \{\theta_b^n, \theta_{b+1}^n, \ldots, \theta_e^n\}$$

and

$$\theta^{n+1} = \{\theta_b^{n+1}, \theta_{b+1}^{n+1}, \ldots, \theta_e^{n+1}\}$$

represent the individual values on iteration n and the transformed set of values at iteration $n+1$ respectively.

The operations of an individual iteration may be calculated in the normalized space of the beginning and ending phase, which is a critical prerequisite for applying exponentiation such that the beginning and ending are invariant. The iteration-normalized values can then be transformed back into their place along the larger ramp, including scaling and phase bias for the input and output axes, as seen in Figure 5.

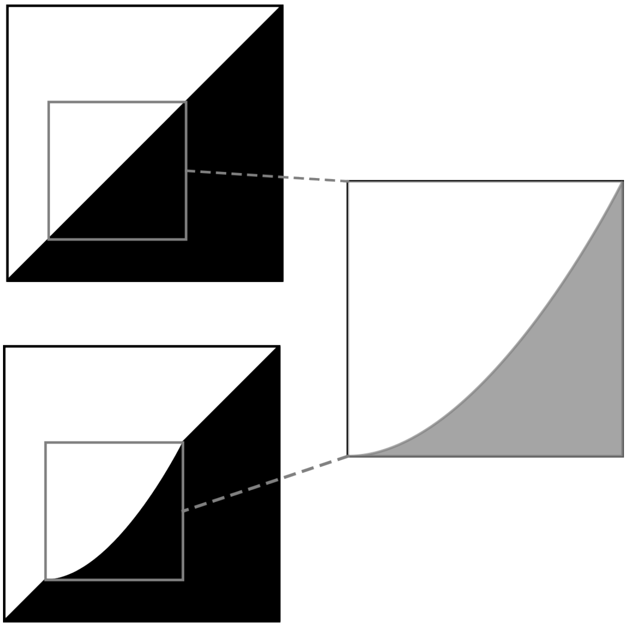


Figure 5. A piecewise iteration, calculated in a normalized space and transformed back into the ramp space.

4.3 Frequency Term

An optional frequency parameter is included in all of the methods here, which further subdivides the span of the modulations. Among these example implementations, we keep the frequency term relative to the normalized iteration space. Once the iteration is normalized within the begin-to-end range$\{b, b+1, \ldots, e\}$, the formula

$$y = \frac{f(sx \bmod 1) + \lfloor sx \rfloor}{s} \tag{4}$$

applies the function to multiple sweeps where, as outlined in Figure 6, s is the frequency, and $f(\cdot)$ is the function to be applied.

4.4 Alpha Term

We also include an optional blend factor in our set of modulations. This term controls how much of the previous iteration is overwritten by the next by applying linear interpolation to the new value. Notably this is different from the way the new value itself may also be effected by the previous value or neighboring values; the alpha is applied as a final computation. Perhaps also notable is the similarity between this blend and moving across a wave table, although the handling of the wave will likely differ in the context of timbral control. This blend produces musically interesting results, particularly where a blend from diagonal slope to a stepped slope creates proportionally shifted subdivisions, as illustrated in Figure 7.

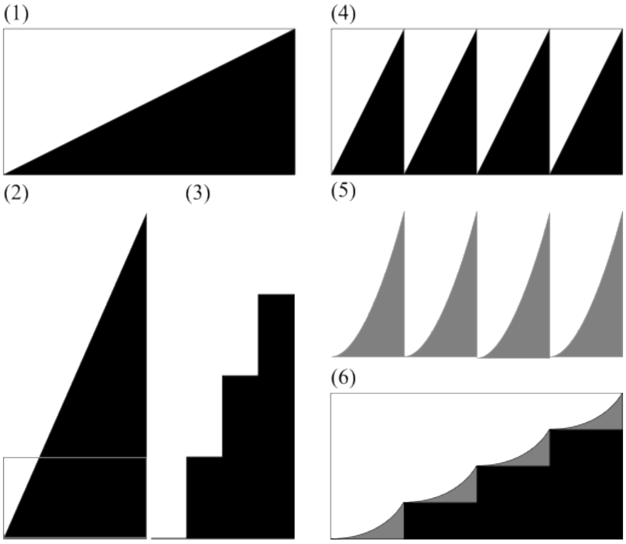


Figure 6. (1) The unmodulated ramp, (2) Scaling the ramp by the Subdivision term, in this case 4, (3) The floor of the scaled subdivision ramp, (4) the modulus 1 of the scaled ramp, (5) using the modulus to apply the function on each subdivision, (6) combining and renormalizing the floor step and the applied function

5 Application

We present four examples that illustrate interesting aspects of the iterative functional ramp approach. Some of these examples focus on demonstrating the system's ability to produce complex rhythmic patterns through non-linear timing curves, while others highlight its capacity for seamless transitions between different timing modes. These cases underscore the versatility of the iterative functional ramp approach, revealing its potential for applications ranging from experimental music composition to adaptive control in interactive audio systems.

Figure 8 shows a sequence of 7 that is skewed by a piecewise division of 4, each containing exponential sweeps of increasing prime-numbered subdivisions. Of particular note is the conforming of the mixed meter into an even subdivision feel while the doubly subdivided hi-hats do not repeat the same swung tuplet.

Figure 9 presents an example of linear interpolation between iterations of the ramp transformation. In this example, two sections (phase segments 0.3–0.5 and 0.8–1) are subdivided into Euclidean 2 hits over 6 steps and remapped into the local region of the phase slice. The euclidean ramp is linearly blended with the previously present values at 50% strength, generating the discontinuous and fractional slope noted earlier. Phenomenologically, the result is a slightly skewed triplet feel and a breaking of the palin-

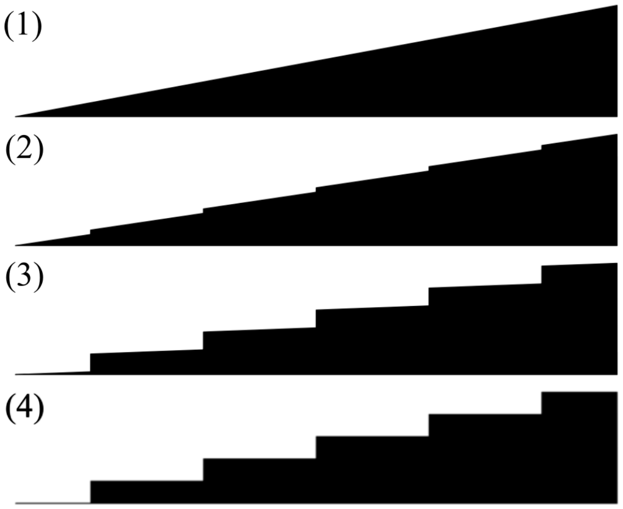


Figure 7. A Euclidean quantization $E(5, 15)$, applied at increasing scales of interpolation from the unmodulated ramp, (1) 0%, (2) 20%, (3) 80%, (4) 100%

dromic sequence.

Figure 10 presents another case of Euclidean quantization mixed in at low strength, this time combined with a slight exponential scoop applied over 4 subdivisions. The triplet feel of the previous example is present here as well, with an additional exponential accelerando and tempo drop each beat. It transforms a rather plain house music beat to be more off-balance in a way that accentuates syncopation.

Finally, Figure 11 skews a sequence of 32 steps into a Euclidean rhythm of $E(42, 11)$, or 11 hits out of 42. This produces a tight, yet wobbling, swagger when weighted at 0.1, and a very irregular phasing polyrhythmic feel when weighted at 0.7.

6 Future Work

This study opens several avenues for further exploration. Future work could involve expanding the interpreter stage to allow for more sophisticated command structures and greater flexibility in authoring complex rhythmic patterns. For example, recursive hierarchies and additional notational paradigms would immediately enable a more diverse warping of the sequencer progression. Implementing the combination of the various iterations in the model described here as a fully dynamic polyphonic voice model, rather than the precomputed approach, is another promising direction. A real-time synthesis approach to the ramps could also be interoperable with the precomputed method.

Furthermore, the development of more methods is intentionally meant to be extendable for future work. Application of chaotic or stochastic systems within the proposed framework could yield interesting musical results. Finally, investigating user-driven customization and adaptive learning algorithms could help tailor rhythmic modulations to specific musical contexts or preferences.

7 Conclusion

This paper presents a novel approach to rhythm generation by extending the frameworks of circle maps and functional iteration synthesis to encompass sequencer timing. The implementation demonstrates the musical potential of this method, offering a framework that supports a continuum of rhythmic expressions through a modular, phase-based methodology, compounded by the use of iteration. The system enables smooth transitions between a variety of rhythmic structures, ranging from conventional swing to complex polyrhythms and freely warped timings. Within the implementation, we outlined and discussed a number of design features and their applicability towards defining rhythm, and followed this discussion with musical use cases. This work builds on established traditions of rhythmic exploration in electronic music and establishes a foundation for ongoing innovation in rhythmic synthesis.

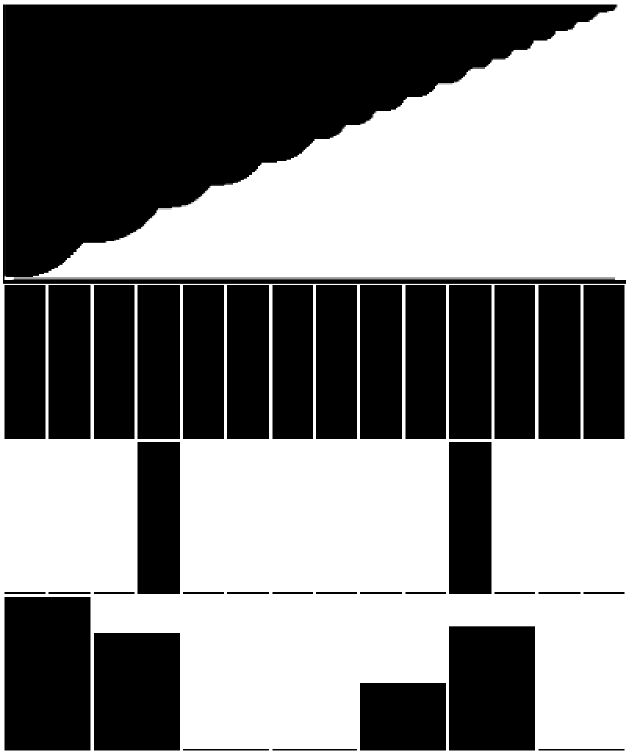


Figure 8. A step pattern of 7 is swung by quarter notes of increasing prime subdivision. Audio Examples: Unwarped [LINK], and Warped [LINK]. Max Patch Commands:
exp 0 0.25 2 3 :
exp 0.25 0.5 3 3 :
exp 0.5 0.75 5 4 :
exp 0.75 0.99 7 5 :

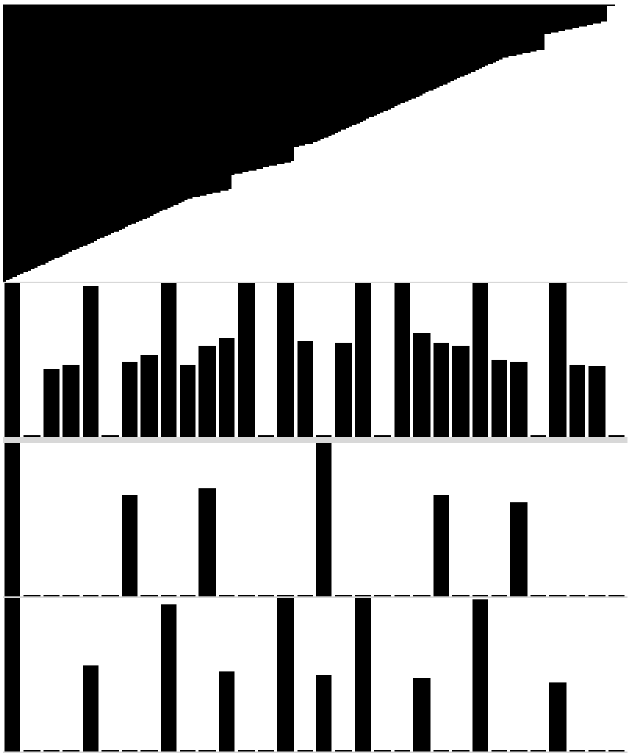


Figure 9. Euclidean Tuples on the backbeat create a sense of ebb and flow, and break the symmetry of the underlying 32 step pattern. Audio Examples: Unwarped [LINK], and Warped [LINK]. Max Patch Commands:
bjo 0 1 16 10 :
exp 0 1 4 1.3 :

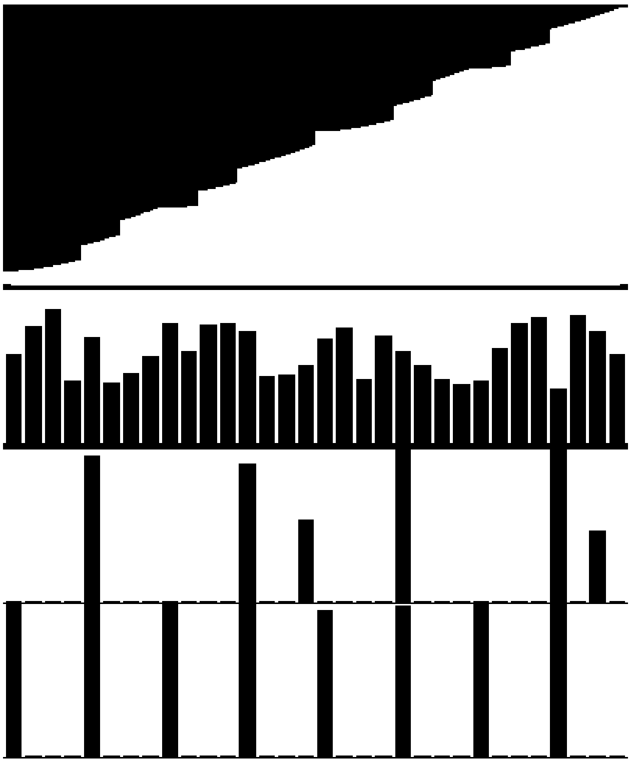


Figure 10. Simultaneous exponential and euclidean swing. Audio Examples: Unwarped [LINK], and Warped [LINK]. Max Patch Commands:
bjo 0.3 0.5 6 2 :
bjo 0.8 1 6 2 :

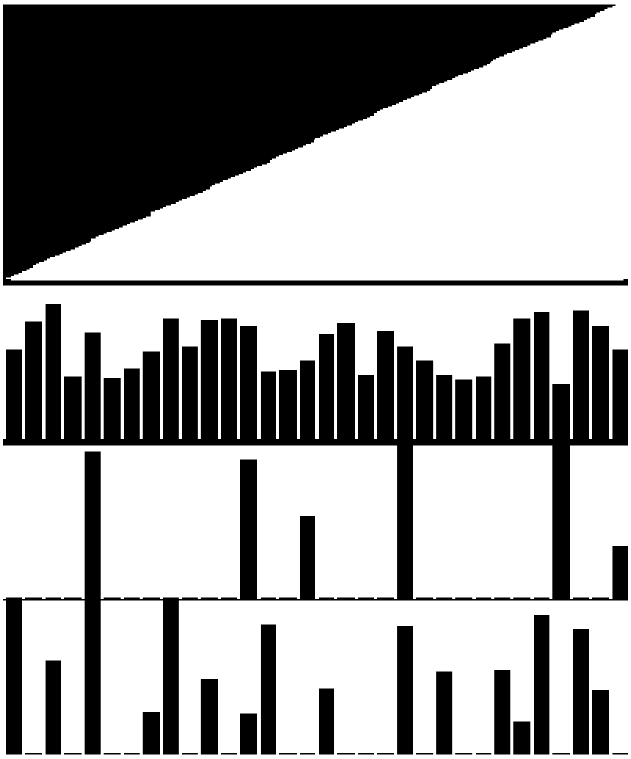


Figure 11. The slightest perturbation can have significant auditory effects. Graphed here is 0.1 blend weight from a series of increasing blends similar to Figure 7. Audio Examples: Unwarped [LINK], 0.1 weight [LINK], 0.3 weight [LINK], 0.7 weight [LINK], fully weighted [LINK]. Max Patch Commands:
bjo 0 0.99 42 11 0 0.1 :

8 References

[1] Charnas, *Dilla Time: The Life and Afterlife of J Dilla, the Hip=Hop Producer Who Reinvented Rhyhm.* MCD, 2022.

[2] Point Blank, "Ableton Live Tutorial: Creative Exponential Rhythms," https://www.youtube.com/watch?v=E2rHj-uu-wQ, 2014.

[3] J. Timothy, "Ableton 10- How Exponential Rhythms Work." [Online]. Available: https://www.youtube.com/watch?v=5vx3fwuNBSM

[4] Mr. Bill, "Mr. Bill - Ableton Tutorial 47: Exponential Rhythms," https://www.youtube.com/watch?v=hbsQ3w0hPcw, 2014.

[5] Ableton Sound Design, "ableton sound design — EXPONENTIAL RHYTHMS [1/5] Free Rate Arpeggiator," https://www.youtube.com/watch?v=WXzDesXWA3I, 2022.

[6] Telekom Electronic Beats TV, "Tech Talk: Inside Joshua Eustis / Telefon Tel Aviv's Sound Lab (Electronic Beats TV)," https://www.youtube.com/watch?v=z5Jv7hzUdTs, 2019.

[7] A. D. (Ed.), *Musical Rhythm in the Age of Digital Reproduction.* Routledge, 2010.

[8] G. T. Toussaint, "The Euclidean Algorithm Generates Traditional Musical Rhythms," in *Proceedings of BRIDGES: Mathematical Connections in Art, Music, and Science*, Banff, Alberta, Canada, July 31 to August 3 2005, pp. 47–56.

[9] G. Wakefield and G. Taylor, *Generating Sound & Organizing Time: Thinking with gen$\tilde{B}$ook 1.* Cycling '74, 2022, section on ramp manipulation and time bending.

[10] N. Collins, *Handmade Electronic Music: The Art of Hardware Hacking.* New York, NY, USA: Routledge, 2006.

[11] WMD, *Metron User Manual*, WMD Devices, Sep. 2022, version 1.51.

[12] Endorphin.es, *Ground Control User Manual*, May 2022, section 8.1: TEMPO AND SHUFFLE. [Online]. Available: https://endorphines.info/manuals/Ground_Control_manual_v2.pdf

[13] Intellijel, *Metropolix Manual*, Intellijel Designs Inc., Apr. 2022. [Online]. Available: https://intellijel.com/downloads/manuals/metropolix_manual_v1.4_2022.04.04.pdf

[14] Olympia Noise Co., "Patterning 3: Drum Machine," May 2024, iOS application. [Online]. Available: https://apps.apple.com/us/app/patterning-3-drum-machine/id1631913793

[15] Kymatica, "BitWiz Audio Synth." [Online]. Available: https://kymatica.com/apps/bitwiz

[16] J. Morris, "Untitled," in *Proceedings of the 22nd Generative Art Conference*, Rome, Italy, Dec. 2019, pp. 168–240. [Online]. Available: https://generativeart.com/GA2019_web/28_JeffMorris-168x240.pdf

[17] ——. (2020) Persistence of Elusion. Videopoem, Audiovisual Composition, University of Central Florida. [Online]. Available: https://projects.cah.ucf.edu/mediaartsexhibits/uncontinuity/Morris/PersistenceOfElusion-Videopoem.mp4

[18] E. Bjorklund, "The Theory of Rep-Rate Pattern Generation in the SNS Timing System," Spallation Neutron Source, Technical Report SNS-NOTE-CNTRL-99, 1999.

[19] G. Essl, "Iterative Phase Functions on the Circle and Their Projections: Connecting Circle Maps, Waveshaping, and Phase Modulation," in *Perception, Representations, Image, Sound, Music*, ser. Lecture Notes in Computer Science, vol. 12631. Marseille, France: Springer Verlag, October 2021, 14th International Symposium, CMMR 2019, Revised Selected Papers.

[20] A. D. Scipio, "Iterated Nonlinear Functions as a Sound-Generating Engine," *Leonardo*, vol. 34, no. 3, pp. 249–254, 06 2001. [Online]. Available: https://doi.org/10.1162/002409401750287029

[21] C. Barlow and H. Lohner, "Two Essays on Theory," *Computer Music Journal*, vol. 11, no. 1, pp. 44–60, 1987.

[22] M. Gotham, "Meter Metrics: Characterizing Relationships Among (Mixed) Metrical Structures," *Music Theory Online*, vol. 21, no. 2, 2015.

[23] J. Morris, "A Dynamic Model of Metric Rhythm in Electroacoustic Music," in *Proceedings of the 2004 International Computer Music Conference.* Miami, Florida, USA: Michigan Publishing, November 1-6 2004.