

THE EVOLUTIONARY SOUND SYNTHESIS METHOD

Jônatas Manzolli^{1,2}, Adolfo Maia Jr.^{1,3}, Jose Fomari^{1,4} and Furio Damiani^{1,4}

¹Interdisciplinary Nucleus for Sound Studies – NICS

²Arts Institute Music Department – DM/IA

³Applied Mathematics Department – IMECC

⁴School of Electrical and Computer Engineering – DSIF/FEEC

University of Campinas- UNICAMP

BRAZIL

[jonatas, adolfo, fomari, furio]@nics.unicamp.br

ABSTRACT

A mathematical model for interactive sound synthesis based on the application of Genetic Algorithms (GA) is presented. The Evolutionary Sound Synthesis Method (ESSynth) generates sequences of waveform variants by the application of genetic operators on an initial population of waveforms. We describe how the waveforms can be treated as genetic code, the fitness evaluation methodology and how genetic operations such as crossover and mutation are used to produce generations of waveforms. Finally, we discuss the results evaluating the generated sounds.

INTRODUCTION

As it is found in the recent literature of electronic music, genetic algorithms (GAs) have been applied to produce evolving trajectory of musical material. Koza [12,14] defines genetic programming as a domain-independent problem-solving approach in which computer programs are evolved to solve, or approximately solve, problems. Garcia [13] is being working with GAs to solve the problem of automating the design of the sound synthesis techniques and sound synthesizer topologies. Biles [4] presented a genetic algorithm-based program that mimics a student learning to improvise jazz solos under the guidance of a human mentor. In Horowitz's [5] development, an interactive system uses GAs to develop a criterion for distinguishing rhythmic patterns producing a large number of variations. We have also studied applications of GAs to interactive composition [6]. Similar to the approaches described above, our previous research used MIDI data to control music events in real time. Yet, using a different heuristic, we created a system named *Vox Populi* [7], a hybrid system composed of an instrument and a compositional environment. *Vox Populi* produces sound moving from clusters to sustained chords, from pointillist sequences to arpeggios, depending upon the number of chords in the population, the duration of the generation cycle, and drawings made by the user over a graphic interface (GUI) control pad.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

MM'01, Sept. 30-Oct. 5, 2001, Ottawa, Canada.

Copyright 2001 ACM 1-58113-394-4/01/0009...\$5.00

Evolutionary Computation is being used by Johnson [9] to develop a computer system for sound design. Since it is, in general, difficult to combine quantitative and qualitative descriptions of a given sound, we apply the concept of Evolutionary Systems to develop a methodology for Timbre Design. This was named as *ESSynth Method* and is the focal point of this paper. *ESSynth* uses a set of Target waveforms to describe a timbre tendency generated by, from an initial population of waveforms, new variants (generations) "similar" to those ones in the target. This similarity is measured by evaluations of a Fitness Function. From an algorithmic point of view *ESSynth* can be seen as a man-machine process that uses an implicit set of rules for generating waveform variations. The main goal of this research is to formulate robust mathematical and computational models for measuring waveform similarities and to define genetic operations such as *crossover* and *mutation* (see definitions below) to operate as waveform transforms. By controlling the Target Population as well as the Initial Population, it is possible to create organized sound patterns or, at a higher level, a musical composition. For the latter, an external device linked to the *ESSynth* will be necessary, to obtain musical sequences in real time.

Inspired by Risset's sound variants idea [2], it is possible to imagine variants as a kind of genetic transformation applied into a population of sound patterns. Smalley's [3] integration of time and spectra induces to think in a timbre evolution over time or, using another terminology, a dynamic process in which an *Evolutionary Timbre* is generated. In *ESSynth*, waveform populations are taken as genotype sets and the resultant transforming timbre is taken as phenotype. In this sense the *genotype* is changed (i.e. the waveforms in the population), but the *phenotype* is preserved (i.e. the overall timbre) producing a variant. In the way Risset interpreted Schaffer's concept we can interpret waveform (i.e. genotype) as "form" and timbre (i.e. phenotype) as "matter" in the case of *ESSynth*. Therefore, these two elements are used in the same way evolution uses genetic information to generate new individuals.

1. EVOLUTIONARY MANIPULATION

The *ESSynth* method is a man-machine interaction cycle. Firstly, the user specifies a set of Target Waveforms. Secondly, the computer produces generations of waveforms using the target set as the fitness criterion. The user is free to change the target set any time. When this happens, a new population generation starts, and so forth. There are three basic structures of control:

- $B^{(n)}$, the n -th waveform population generation. The waveforms initial set is denoted by $B^{(0)}$.
- T , the waveforms Target set.
- f , the *Fitness Function* used to evaluate the best waveform w^* of each generation.

We define w^* as the most similar or, in mathematical terms, the closest waveform of $B^{(n)}$ to the target set T . In each generation its best waveform w^* is sent to a buffer and played as a wavetable cyclically (see Fig 1.).

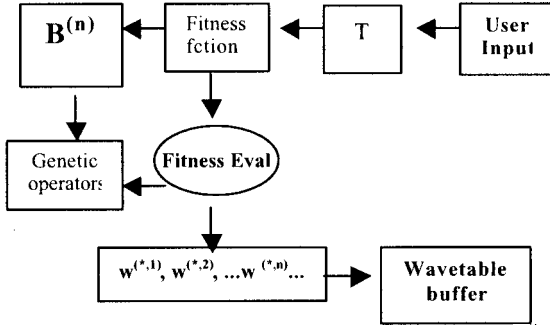


Figure 1. Basic control structures of the ESSynth. $w^{*,n}$ denotes the best waveform of the n -th generation sent to the wavetable buffer.

2. FITNESS EVALUATION

Firstly, we need to define an auxiliary metric or distance function. Our mathematical model considers waveforms as vectors in a real vector space $W = \mathbb{R}^{1024}$, i.e. each vector in the space has 1024 components.

Given two vectors v and w in W , we define the usual Euclidian Metric between them:

$$d_2(w, v) = (\sum_{i=1}^{1024} (w_i - v_i)^2)^{1/2} \quad (1)$$

This metric induces the norm $\|w\| = (\sum_{i=1}^{1024} (w_i)^2)^{1/2}$ and gives the total energy of the resultant sound. However, other metrics could be used and tested.

Now we define a distance function between two sets. So, let $T = \{t^{(1)}, t^{(2)}, \dots, t^{(L)}\}$ to be the *Target waveform set* and $B^{(n)} = \{w^{(n,1)}, w^{(n,2)}, \dots, w^{(n,M)}\}$ the n -th waveform generation set. Since these are sub-sets of W , we can define the distance between them as follows:

$$d(T, B^{(n)}) = \min \{d_2(t^{(j)}, w^{(k)})\} \quad (2)$$

with $j = 1, \dots, L$ and $k = 1, \dots, M$, and L is the number of waveforms in T and M is the number of waveforms in $B^{(n)}$.

As pointed above, T and $B^{(n)}$ are finite sets, therefore the minimum in Eq. (2) is obtained for at least one vector in $B^{(n)}$, which we denote by $w^{(n,*)}$. This vector is the n -th generation best waveform, obtained using the metric of Eq. (2). Now we define the n -th generation Fitness Function $f: T \times B^{(n)} \rightarrow B^{(n)}$ as

$$f(T, B^{(n)}) = w^{(n,*)} \quad (3)$$

which indicates the best individual of n -th generation population.

In a process of genetic improvement the best individuals with better-adapted phenotype, survive in the next generation. In Nature adverse conditions select these individuals. In our model this is accomplished by a distance function that measures how far a new waveform population departs from a Target Set.

3. THE GENETIC OPERATIONS

Waveform variants are produced by applying genetic operations such as crossover and mutation to $B^{(n)}$. An interesting ESSynth feature is to make this waveform patterns dynamical sequence in real time. Biologic evolution produces species diversity: with ESSynth one can create and manipulate a complex generation of sound material. Crossover increases the waveform co-variance and mutation produces random population variations.

Starting with a Crossover Vector described as $\alpha = [\alpha_1, \alpha_2, \dots, \alpha_M]$ where $0 \leq \alpha_i \leq 1$ chosen by the user, it is possible to define a kind of continuous waveform crossover. The n -th generation best waveform is used as a *Parent Waveform* $w^{(n,*)} = (s_1, s_2, s_3, \dots, s_{1024})$. Any other $B^{(n)}$ waveform is denoted by $w^{(n,i)}$ with $0 \leq i \leq M$. The following steps define the n -th generation Crossover Operation:

1. Set a random integer number generator in the interval $[1, 1024]$.
2. Take two integer numbers $(k_1^{(n)}, k_2^{(n)})$ in the interval $[1, 1024]$ with $k_1^{(n)} < k_2^{(n)}$.
3. Select the waveform segment in $w^{(n,*)}$ as $S^{(n,*)} = (s_{k_1}, \dots, s_{k_2})$.
4. Combine the waveform segment with a equivalent waveform segment $S_i^{(n,i)}$ in $B^{(n)}$ applying a Hamming Window $H()$ on $S_i^{(n,i)}$ as follows:

$$S^{(n+1,i)} = \alpha_i \cdot H(S_i^{(n,*)}) + (1 - \alpha_i) S^{(n,i)} \quad (4)$$

- Notice, we denote the new segment as $S^{(n+1,i)} = (s'_{k_1}, \dots, s'_{k_2})$.
5. The crossover operation is the replacement of each $S^{(n,i)}$ in the original waveform making $w^{(n+1,i)} = (s_1, s_2, \dots, s'_{k_1}, \dots, s'_{k_2}, \dots, s_{1024})$.
 6. Repeat steps (4) and (5) for all waveforms $w^{(n,i)}$ in $B^{(n)}$ with $w^{(n,i)} \neq w^{(n,*)}$.

On any living organism, mutation makes strong modifications on population individuals, generally due to external factors. It can be understood as a kind of disturbance of the reproduction process. We used this characteristic to define the mutation below. It starts with a definition of a *Mutation Coefficient* $0 \leq b \leq 1$ that sets the amount of disturbance applied to $B^{(n)}$. Since the waveforms belong to $W = \mathbb{R}^{1024}$, a *Mutation Vector* is generated with 1024 entries randomly generated in the interval $[1-b, 1]$, named as *disturbance interval*. Now the *Mutation Operation* is defined on the n -th generation by the following steps:

1. Create the *Mutation Vector* $\beta = [\beta_1, \beta_2, \beta_3, \dots, \beta_{1024}]$, where each β_j belongs to the disturbance interval $[1-b, 1]$.
2. Apply the disturbance $w^{(j,n+1)} = w^{(j,n)} \cdot \beta_j$ on all elements of $B^{(n)} = \{w^{(1,n)}, w^{(2,n)}, \dots, w^{(M,n)}\}$.
3. Repeat steps (1) and (2) in every generation.

The mutation strength is controlled by parameter b in the real interval $[0, 1]$. The closer is b to 0 the weaker is the mutation. As b gets close to 1 the mutation gets stronger. Notice that ESSynth Mutation Operation can be seen as a *wave-shaping* process where the waveforms are modified by a random waveform in an amount given by the Mutation Coefficient.

4. GRAPHICAL ANALYSIS OF SOUNDS

In this section we analyze graphic examples of sound results. These were obtained using Matlab 6.0 to simulate the ESSynth method. A Target Waveform Population was used with two diverse Initial Populations:

- a) Random distributed sine waveforms
- b) Harmonic distributed sine waveforms,

as shown in Table 1.0.

Table 1. Experimental parameters used to simulate ESSynth

ESSynth Examples			
Index	Population Description	Waveforms in the population	Frequency Distribution (Hz)
01	Target Waveform	05	100, 200, 300, 400, 500
02	Random Distributed Sine Waveforms	10	From 180 to 16,000
03	Harmonic Distributed Sine Waveforms	10	100, 200, 300, 400, 500... 10,000

All waveforms have 1024 values normalized in the real interval $[-1, 1]$. It is possible to "see" complex waveform patterns as Target and Initial Populations. Our strategy is to show ESSynth behavior with simple patterns.

As shown in Fig.3 and Fig.4, the method converges making erosions in the waveform population produced by the Crossover Operation. Thus, the sound envelope produced by ESSynth starts with strong amplitude followed by a smooth decay, as instrumental sounds behave. This sound is also dynamically changed in its spectral contents due to the Mutation Operations.

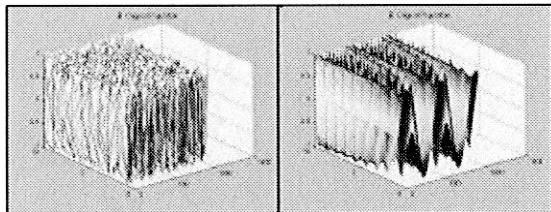


Figure 3. 3" plot of the populations used in the examples. The plot on the left side is the Random Distributed Sine Waveforms population. The plot in the right side is the Harmonic Distributed Sine Waveforms population. The X-axis displays the 1024 values waveform. The Y-axis displays the waveform pattern in an increasing order and the Z-axis is displaying the samples amplitude in the interval $[-1, 1]$.

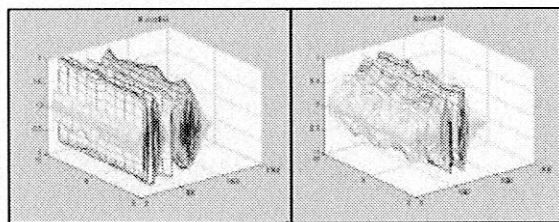


Figure 4. 3D plot of the waveform populations after 500 interactions. It is apparent how the ESSynth works making erosions in the waveform population as the Crossover Operation dumps waveforms.

ESSynth produces a "drive" disturbance that reflects a sound tendency controlled by the user input Target Set. During the Matlab simulations we realized that the crossover operation produces a waveform sequence that converges to the Target Set i.e. a waveform attractor set. This numerical behavior produced

sounds with more similarities among themselves as well among those in the Target Set. On the other hand, despite the mutation operation that can add drastic changes on the waveform trajectory, it did not affect the overall sequence that still converged to the Target. Finally, it is always possible to modify the Target and/or the Initial Population leading to a very strong departing from the actual waveform pattern" as shown in the graphic examples.

5. CONCLUSION

We developed a new methodology for sound generation and a mathematical model to constrain the waveform variants to a Target Set. ESSynth is a "cw sound synthesis method that integrates the Mathematical Approximation Theory to the Genetic Algorithms. It can be seen" as a new framework for Timbre Design. The basic features of the method are: a) the "use of a Target Set of waveform patterns, b) a Fitness Criterion implemented by a distance function (Hausdorff Distance) which measures how much a waveform population departs from the Target Set and c) an evolutionary process, based on crossover and mutation operations, applied to waveform sets to produce a sequence of waveform generations drive" by the Target Set.

REFERENCES:

- [1] Schaeffer, P. "Traité des objets musicaux". Editions du Seuil. Paris 1966.
- [2] Risset, J.-C. 1991. "Timbre Analysis by Synthesis: Representations, Imitations, and Variants for Musical Composition". In *Representations of Musical Signals*, ed. De Poli, Piccialli & Road, Cambridge, Massachusetts: The MIT Press, ISBN 0-262-04113-8, pg. 7-43.
- [3] Smalley, D. 1990. "Spectro-morphology and Structuring Processes". In *The Language of Electroacoustic Music*, ed. Emmerson, pg. 61-93.
- [4] Biles J. A.. GenJam: "A Genetic Algorithm for Generating Jazz Solos". *Proceedings of the 1994 International Computer Music Conference*, (ICMC'94), 131-137, 1994.
- [5] Horowitz D., "Generating rhythms with genetic algorithms". *Proceedings of the 1994 International Computer Music Conference*, 142-143, 1994.
- [6] Manzoli, J., A. Moroni, F. Von Zuben & R. Gudwin. 1999. "An Evolutionary Approach Applied to Algorithmic Composition". *Proceedings of the VI Brazilian Symposium on Computer Music*, Rio de Janeiro, p. 201-210.
- [7] Moroni, A., Manzoli, J., Von Zuben, F. & Gudwin, R., 2000, "Vox Populi: An Interactive Evolutionary System for Algorithmic Music Composition". *San Francisco, USA: Leonardo Music Journal - MIT Press*, Vol. 10, p. 49-55.
- [8] Fogel D. B.. "Evolutionary Computation - Toward a New Philosophy of Machine Intelligence". *IEEE Press*, USA, 46 - 47, 1995.
- [9] Johnson C. G.. "Exploring the sound-space of synthesis algorithms using interactive genetic algorithms in G. A." Wiggins, editor, *Proceedings of the AISB Workshop on Artificial Intelligence and Musical Creativity*, Edinburgh, 1999.
- [10] Koza, J. R., Bennett III, F. H., Andre, D., Keane, M. A., Dunlap, F., "Automated Synthesis of Analog Electrical Circuits by Means of Genetic Programming," *IEEE Transactions on Evolutionary Computation*, Vol. 1, NO. 2, July 1997.
- [11] Cheung, N. M., Horner, A., "Group Synthesis with Genetic Algorithms," *Journal of the Audio Engineering Society*, 44(3): 130 -147, 1996
- [12] Koza, John R.. "Genetic Programming". *Encyclopedia of Computer Science and Technology*. 1997.
- [13] Garcia, Ricardo A. "Automatic Generation of Sound Synthesis Techniques". *Proposal for degree of Master of Science. MIT - Fall 2000*.