Waveform Synthesis Using Evolutionary Computation

Jônatas Manzolli[1,2],Adolfo Maia Jr[1,3],José Fornari[1,4],Furio Damiani[1,4]

[1]Interdisciplinary Nucleus for Sound Studies - NICS
[2]Music Department – DM/IA
[3]Applied Mathematics Department – IMECC
[4]School of Electrical Engineering – DSIF/FEEC
University of Campinas – UNICAMP
BRAZIL

{jonatas,adolfo,fornari,furio}@nics.unicamp.br

Abstract. *ESSynth, a new method for the synthesis of wave sound patterns is presented. We take advantage of the Evolutionary Systems theory to develop an approach that uses a set of wave patterns as "target" to embed variations in another set of waveform called "population". This paper covers how the wave patterns are treated as genetic codes, their fitness evaluation methodology and how the genetic operations, such as crossover and mutation, are used to produce new waveforms.*

Resumo.*ESSynth é um novo método para síntese de padrões de forma de onda. A partir da teoria de Sistemas Evolutivos, desenvolvemos uma abordagem que utiliza um conjunto de padrões de forma de ondas como "objetivo" para produzir variações em outro conjunto de forma de ondas denominado de "população". Este trabalho apresenta como as formas de ondas são tratadas como código genético, a avaliação de seu fitness e as operações genéticas de cruzamento e mutação utilizadas para gerar novas formas de onda.*

1. Introduction

Evolutionary Computation is being increasingly used in Computer Graphics to create scenes and animations (Foley, 1996). In these applications the rules are *learned* by the system through its interaction with the user (Fogel,1995). We used this property to generate sound pattern variants. We already studied other applications for interactive composition (Manzolli et all, 1999), using MIDI data to control music events with a heuristic approach (Moroni et all, 2000).

From an algorithmic point of view, *ESSynth* can be described as a man-machine process that handles a set of rules to generate sound material. The main goal is to formulate a robust mathematical model for measuring waveform similarities and then defining genetic operations such as *crossover* and *mutation* to generate interesting sound transformations. By controlling the waveform target as well as the initial waveform population the user is able to obtain well-organized sounds or, at a higher level, to perform music composition. Such process can be understood as the next evolutionary step in the paradigm for musical composition. In each generation, the best waveform is sent to a sound output (a wavetable engine). ESSynth is an excellent tool for real-time

synthesis where the user can manipulate the Target set as well as the initial population in order to get a musically significant sequence of waveforms and consequently new sounds. An external controller device may be linked to the *ESSynth* algorithm in order to control the production of sound patterns in real-time. *ESSynth* can be seen as an experimental environment for sound synthesis. Our method is a kind of macro-structural synthesis so that the Fourier partials are treated in batches. From the operational point of view, it might be seen like carving tools that *sculpt* the set of waveforms.

In the following sections we present the method, its mathematical formulation and experimental results. This paper does not include the concepts of Evolutionary Computation that can be found in previous publications (Manzolli et all, 1999; Moroni et all 2000).

2. Evolutionary Manipulation of Waveforms

The ESSynth method can be described as a man-machine interaction cycle. First, the user specifies a set of target waveforms. It is important to underline that the user is free to change the target set at any time. Second, the computer produces generations of waveforms using the target set as the raw material for the fitness evaluation process. When the user changes the target, the computer generates a new set and so forth. There are three basic control structures: the population of the *n-th* generation denoted by $\mathbf{B}^{(n)}$ (the initial generation is denoted by $\mathbf{B}^{(0)}$), the set of target waveforms **T** and the *Fitness Function* $\boldsymbol{f}$. The fitness function is used to find the best waveform **w*** of each generation $\mathbf{B}^{(n)}$. The **w*** is defined as the *most similar* or, in mathematical terms, *closest* waveform $\mathbf{B}^{(n)}$ to the set **T,** measured by a distance function (see the appendix for a definition). In each generation the best waveform **w*** is sent to a buffer and is played as a periodically scanned wavetable.

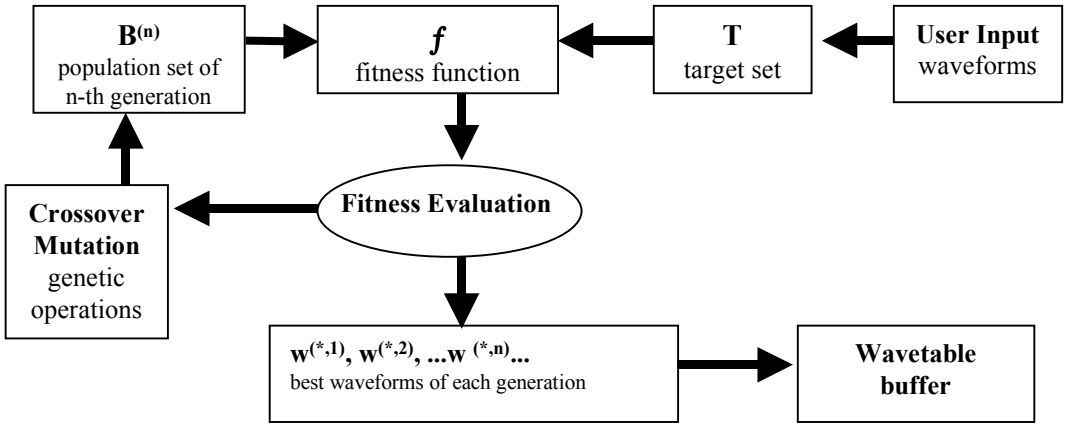

Figure 1. ESSynth basic control structures. $\mathbf{w}^{(*,n)}$ denotes the best waveform of the n-th generation, which is sent to the output wavetable buffer

The waveforms are normalized as floating-point arrays of 1024 values (or points) defined in the real interval [-1,1]. The user always defines the **T** set and optionally the $\mathbf{B}^{(0)}$, which is otherwise randomly generated. When the user changes the **T**, the system reacts in either of two ways depending upon its previous state: $\mathbf{B}^{(n)}$ is kept unchanged (as the initial $\mathbf{B}^{(0)}$) or a new $\mathbf{B}^{(0)}$ is randomly generated/user defined. The first situation produces a mutation in the waveform generation, but the overall characteristics of the

sound pattern are kept unchanged. In the second case, the inputs will start a new generation of variant waveforms that will create new generations of sound patterns. The overlap-and-add technique is used to interlace the sequence of best waveform of each generation $\mathbf{w}^{(*,i)}$, in order to obtain a smooth audio output. To do so, a Hanning window is applied to each $\mathbf{w}^{(*,i)}$. The interlacing is done so that the amplitude doesn't change on the overlapped regions, as shown in figure 2.

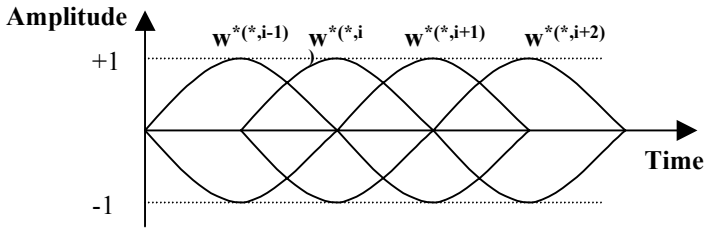

Figure 2. Hanning windowed best waveforms $w^{(*,I)}$ overlap-and-add at50% .

3. Crossover, Mutation and Fitness Evaluation

Waveform variants are produced by applying genetic operations such as crossover and mutation in the $\mathbf{B}^{(n)}$ population. ESSynth dynamically generates waveform sequences. Similar to the biologic evolution that produces diversity in Nature, it creates and manipulates a complex generation of sound material in real-time. In statistical terms the crossover increases the waveform co-variance and the mutation produces random variations in the population. These two operations are defined below.

3.1 Crossover

Similar to the genetic code of living organisms reproduced by meiosis, the crossover operation mixes the *waveform codes* of population $\mathbf{B}^{(n)}$. In order to get better-fitted individuals, ESSynth crosses population $\mathbf{B}^{(n)}$ with the elements of the T set. This operation performs a *natural selection* in the domain of waveforms. As the user can interfere at any time, it would be more accurate to say that ESSynth uses a *driven genetic selection*.

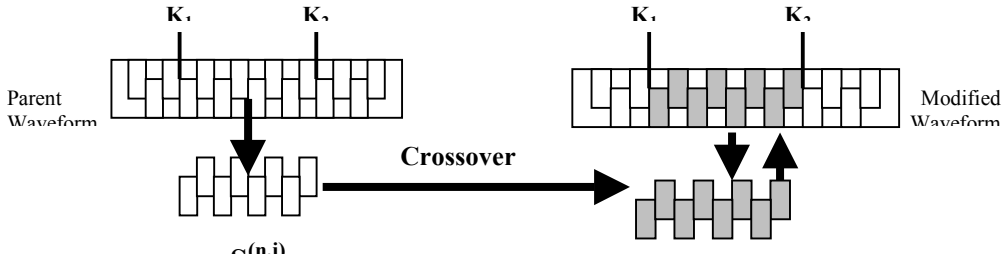

Figure 3. Crossover operation diagram, where a segment from a parent waveform (defined by two randomly generated values K1 and K2) is merged with an equivalent segment of a waveform in the $B^{(n)}$ set.

In order to avoid glitch noise due to the crossover process, a convex combination and a Hanning window are applied to smooth the $\mathbf{S}^{(n,i)}$ waveform segment edges.

3.2 Mutation

In living organisms mutation implies a modification, usually due to external factors. It might affect individual phenotypes. This concept was used for our waveform mutation operation. We defined a mutation parameter **b** in the real interval [0,1] to quantify the mutation strength. When **b** is close to 0 the mutations are non-existent; for **b** close to 1 the mutations are strong. This operation will add high frequency spectral components to the population of waveforms.

3.3 Fitness Evaluation

In the process of genetic improvement, it is necessary to measure which individuals are better fitted to survive. In Nature, an adverse environement selects these individuals. In the waveforms world this can be accomplished by a function that measures the distance between an n-th generation population set $\mathbf{B}^{(n)}$ and the target set **T**. Therefore it is necessary to have a function to measure the distance between two sets (population and target) not necessarily having the same number of elements (waveforms). In mathematics, this is called *Hausdorff Distance*. This is well suited for distance measurement between sets of points in a N-dimensional real space. In our case, N = 1024.

4. Conclusion and Results

A new methodology for sound generation was developed. Its mathematical model was established, constraining the sound variants in a coherent domain. ESSynth features include the use of a target set of waveforms, a fitness criterion using Hausdorff distance, an evolutionary process based on crossover and mutation operations.

ESSynth Table of Parameters				
Index	**Population Description**	**Frequency distribution (Hz)**	**Number of Individuals**	**Comments**
01	Target set / harmonically distributed sine waves	100, 200, 300, … 2000	20	10% mutation
02	Population set / randomly distributed sine waves	From 180 to 16000	25	100 interactions

Table 1. ESSynth parameters used to generate a waveform sequence.

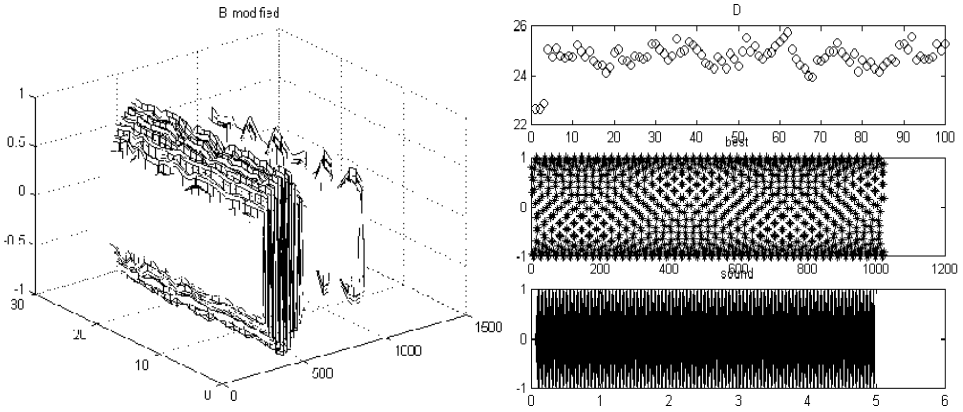

Figure 4. (left side) Population of waveforms after 100 generations. (right side, from top to bottom) 1) Fitness variation of n-th generation population $B^{(n)}$ to T set; 2) Best (fitted) waveforms; 3) Resulting waveform after overlap-and-add

5. Appendix: Mathematical Model

An auxiliary metric, or distance function is firstly defined. This mathematical model considers waveforms as vectors in a real vector space $W = \Re^{1024}$, i.e. each vector in the space has 1024 components (points). Given two vectors **v** and **w** in **W**, we define the usual Euclidian Metric between them:

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

As it is known, this metric induces the norm: $|\mathbf{w}| = (\Sigma_{i=1\ldots1024} (w_i)^2)^{1/2}$ Since we are considering the vectors as waveforms, this norm gives the total energy of the resultant sound. We arbitrarily opted to use the Euclidian metric, although other metrics might be tried and used as well.We are now ready to define the distance function between two sets. Let $T = \{\mathbf{t^{(1)}}, \mathbf{t^{(2)}}, \ldots, \mathbf{t^{(L)}}\}$ to be the *Target Waveform Set* and $B^{(n)} = \{\mathbf{w^{(n,1)}}, \mathbf{w^{(n,2)}}, \ldots, \mathbf{w^{(n,M)}}\}$ the set of the n-th generation of the waveform population. Since these are sub-sets of **W**, we can define the distance between them as follows:

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

where: **j = 1, … , Lk = 1, … , M**

L is the number waveforms in the target **T.**

M is the number of waveforms in the n-th generation of the population $\mathbf{B^{(n)}}$.

The measure (2) is called the "Hausdorff Distance" between two sets. As pointed above, T and $B^{(n)}$ are finite sets, therefore the minimum in Eq. **(2)** is given by at least one vector in $B^{(n)}$, which we denote as $\mathbf{w^{(n,*)}}$**.** This vector is the best waveform in the n-th generation of $\mathbf{B^{(n)}}$ which means the closest waveform in population to the given target set **T**, that is using the metric defined in Eq. **(2)**. Therefore, the Fitness Function of the n-th generation $\boldsymbol{f}$: T x $B^{(n)} \rightarrow B^{(n)}$ is

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

where T is the target set, $B^{(n)}$ is the n-th generation population, $w^{*(n)}$ is the best waveform.

5.1 Crossover Operation:

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

The following steps define the crossover operation in the n-th generation of the population:

1. Generate random integers in the interval [1,1024].

2. Take two of these integers as $k_1^{(n)}$ and $k_2^{(n)}$, where $k_1^{(n)} < k_2^{(n)}$.

3. Select the segment of the waveform $\mathbf{w}^{(n,*)}$ defined by k1 and k2 as $\mathbf{S}^{(n,*)} = (s_{k1}, \ldots, s_{k2})$.

4. Combine the waveform segment $\mathbf{S}^{(n,*)}$ with the correspondent segment of the waveform $\mathbf{S_i}^{(n,i)}$ in $B^{(n)}$

5. Apply a Hanning Window $\mathbf{H}(\mathbf{S_i}^{(n,*)})$

6. Combine the windowed segment with the equivalent segment $\mathbf{S}^{(n,i)}$ of $\mathbf{B}^{(n)}$.

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

where $\mathbf{S}^{(n+1,i)} = (s'_{k1},..,s'_{k2})$.is the new segment.

The crossover operation is the replacement of each $\mathbf{S}^{(n+1,\ i)}$ in the original waveform making $\mathbf{w}^{(n+1,i)} = (s_1, s_2,..,s'_{k1}..s'_{k2},..s_{1024})$.

Repeat steps (4) and (5) for all waveform $\mathbf{w}^{(n,i)}$ in $B^{(n)}$, so that $\mathbf{w}^{(n,i)} \neq \mathbf{w}^{(n,*)}$.

Obs: A Hanning window is applied in steps (5) and (6) a also a convex combination to fade the borders of the waveform segment $\mathbf{S}^{(n,i)}$.

5.2 Mutation Operation

This operation starts with the definition of a *Mutation Coefficient* $0 \leq b \leq 1$ that fixes the amount of disturbance applied on $\mathbf{B}^{(n)}$. Since the waveforms belong to $\mathbf{W} = \Re^{1024}$, a 1024 points *Mutation Vector* β is randomly generated in the *disturbance interval* [1-b, 1]. The *Mutation Operation* is defined on the n-th generation by the following steps:

1. Create the *Mutation Vector* $\boldsymbol{\beta} = [\beta_1, \beta_2, \beta_3 \ldots \beta_{1024}]$, where each β_j belongs to the disturbance interval [1-b, 1].

2. On the elements of $\mathbf{B}^{(n)} = \{\mathbf{w}^{(1,n)}, \mathbf{w}^{(2,n)}, \ldots, \mathbf{w}^{(M,n)}\}$ apply the operation of disturbance:

$$\mathbf{w}^{(j,\,n+1)} = \mathbf{w}^{(j,n)} . \beta \quad (5)$$

Repeat the steps (1) and (2) for every next generation.

References

Biles, J. A.(1994) *GenJam: A Genetic Algorithm for Generating Jazz Solos*, Proceedings of the 1994 International Computer Music Conference, (ICMC'94), 131—137.

Foley J. D.(1996) Andries van Dam, Steven K. Feirner and John F. Hughes, *Computer Graphics Principles and Practice*, Addison-Wesley Publishing Company, p 1018.

Fogel, D. B. (1995) Evolutionary Computation - Toward a New Philosophy of Machine Intelligence, IEEE Press, USA, 46 – 47.

Horowitz, D. (1994) *Generating rhythms with genetic algorithms*, Proceedings of the 1994 International Computer Music Conference, 142 – 143.

Johnson,C.G.(1999) *Exploring the sound-space of synthesis algorithms using interactive genetic algorithms*. In G. A. Wiggins, editor, Proceedings of the AISB Workshop on Articial Intelligence and Musical Creativity, Edinburgh.

Manzolli, 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.

Moroni, A.., Manzolli, 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.

Polansky, L. (1996) *Morphological Metrics*, Journal of New Music Research, 25, pp. 289-368.