A NON-LINEAR SOUND SYNTHESIS METHOD

Jônatas Manzolli[1,2], Furio Damiani[1,3], Peter Jürgen Tatsch[1,3], Adolfo Maia Jr.[1,4]

[1]Interdisciplinary Nucleus for Sound Studies
[2]Fine Art Institute, Music Department
[3]DSIF/School of Electrical and Computer Engineering
[4]Applied Mathematics Department
400 Albert Einstein Av., 13081-970
UNICAMP – University of Campinas, SP
BRAZIL
furio@fee.unicamp.br, jonatas@nics.unicamp.br, peter@dsif.fee.unicamp.br, adolfo@nics.unicamp.br

ABSTRACT

This paper presents a new method for sound synthesis based upon non-linear dynamics. The procedure is compact and can be interactively controlled in real-time. The sounds thus synthesized have dynamic characteristics and rich spectra. It is an economical method, for it uses only a cyclic buffer to control the audio output and a non-linear map to generate the waveforms.

INTRODUCTION

This method, FracWave, uses a parametric control of non-linear maps for sound synthesis(1,2). It explores the structural control of non-linear dynamics (NLD) in a computer-based environment. It applies recursive sonic constructs, integrating the macro and micro levels of the sound synthesis process, thus enabling a fast generative method.

NLD in the acoustic domain has been investigated by Lauterborn[3], has been used for micro-structural control of sounds and connected with Granular Synthesis (4,5).

In the domain of the synthesis of sounds using digital simulation of acoustic instrument models, Lindeman[6] describes a physical model for a woodwind instrument reed, showing that a simple model could exhibit sonic responses varying from harmonic to chaotic.

NLD and physical modelling belong to two different classes of digital sound synthesis: the first uses abstract algorithms and the second makes use of traditional musical instruments models. The latter generally expends much calculation time to generate the output; the first is dependent on the algorithm: if it is less calculation intensive, it can be used for real time sound production, as for instance, the FM(7).

It is necessary to formulate new models for timbral design, taking advantage of the new and cheap breeds of Digital Signal Processing (DSP) hardware and of new user interfaces(8) enabling real-time gestural control of synthesis parameters.

1. OVERVIEW

Waveforms of natural sounds and acoustic instruments are time dependent(9). FracWave uses numerical sequences produced by simple non-linear equations to create continuously transforming waveforms. It samples a circular buffer in which numerical sequences are stored. This buffer is a dynamic lookup table, which we will call *dynamic wavetable,* or **DW**.

To produce complex sounds, but also to create degrees of sound complexity, a linear interpolation after a sub-sampling process was applied to foster the control of the frequency spectra. This has to be done, because if one would directly use the sequence of numbers produced by the non-linear map, the sound produced would always possesses too rich a spectra, with intense energy in high frequency components.

One can find similarities between FracWave, the Karplus-Strong Algorithm(10), and the LASy Technique(11). A diagram with the basic blocks of the FracWave process is shown in Fig 1.

2. NUMERICAL ORBITS AS WAVEFORMS

A **DW** stores the values produced by a process in which a non-linear map is recursively iterated. The basic idea is, given a starting point $\mathbf{x_0}$ and a non-linear map $\mathbf{F(x_k)}$, a sequence of values $\mathbf{x_k}$, also called a *numerical orbit*, is obtained by the expression $\mathbf{x_{k+1} = F(x_k)}$.

A sequence $\mathbf{x_k}$, generated by the non-linear map, is mapped into the interval [-1,1]; then it is followed by a linear interpolation, generating a new sequence of samples $\mathbf{x_n}$. To obtain suitable numerical orbits, we studied a number of non-linear maps, extending those found in the NLD literature, such as the Logistic and Henon maps or the Barry Martin algorithm(12).

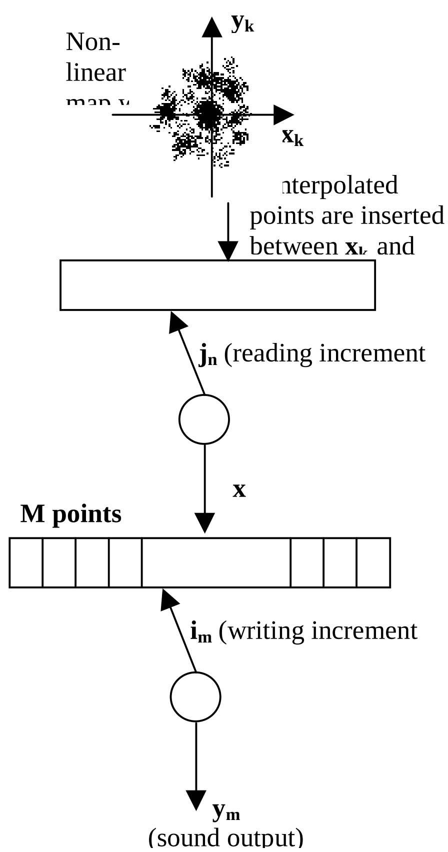

Fig. 1 – FracWave structure diagram.

Calling K the total number of points in the non-linear map and M the number of points in the DW, they should satisfy the relation M > α K. We have been using 2 ≤ α ≤ 100. Thus, the sub-sampling rate F_{sub} on the non-linear map, to obtain a CD quality waveform with F_s =44,100 Hz in the sound output will be given by F_{sub} = 44,100 Hz/α.

The set of equations used in this work are listed below:

$$\mathbf{x_{k+1} = y_k - sgn(x_k)*sqr(abs(B* x_k -C))} \qquad \text{Eq. 1}$$
$$\mathbf{y_{k+1} = A - x_k}$$

$$\mathbf{x_{k+1} = y_k - sgn(x_k) + sqr(abs(B* x_k -C))} \qquad \text{Eq. 2}$$
$$\mathbf{y_{k+1} = A - x_k}$$

$$\mathbf{x_{k+1} = y_k - sin(abs(B* x_k -C))} \qquad \text{Eq. 3}$$
$$\mathbf{y_{k+1} = A - x_k}$$

$$\mathbf{x_{k+1} = y_k - sin(B* x_k -C)} \qquad \text{Eq. 4}$$
$$\mathbf{y_{k+1} = A - x_k}$$

where A, B and C are parameters and all equations start with $\mathbf{x_0}$ = 0 and $\mathbf{y_0}$ = 0. The equations above can be described by the following general formula:

$$\mathbf{x_{k+1} = y_k - perturbation(B,C)} \qquad \text{Eq. 5}$$
$$\mathbf{y_{k+1} = A - x_k}$$

Equations 1 to 4 define two-dimensional maps in which a set of parameters {A, B, C} and initial conditions {x_0, y_0} control a delay factor and a perturbation produced by a non-linear term as described in Equation 5.

The orbit phase space {x_k,y_k} representation shows behaviours varying from quasi-periodic to chaotic. This will give out timbres with a defined pitch and quasi-periodic spectral (small) variations: these are natural sound characteristics.

To analyze the non-linear maps one can use the phase space plot $\mathbf{x_k}$ ***versus*** $\mathbf{y_k}$. On such plots, clusters of points mean that there is an attractor; cluster distribution regularity identifies quasi-periodic numerical behavior; irregularity typifies a chaotic one. Plots showing the phase space behaviour can be seen in Figs. 2 and 3.

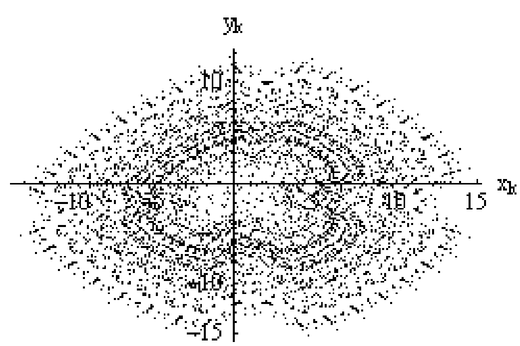

Fig.2 Eq. 1 non-linear map phase space diagram, with A = - 0.54 , B = - 3.9, C = 1.4.

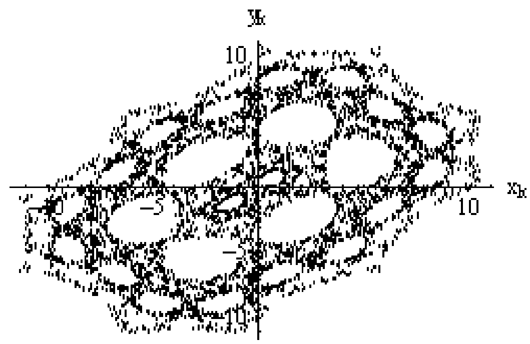

Fig. 3 Eq. 3 phase space non-linear map diagram, with A = - 1.4, B = - 3.9, C = 1.4

2.1 DW CONTROLLERS

The reading and writing increments, j_n and i_m, are defined by:

$$j_n = n\,\phi \text{ rmod}(M) \qquad \textit{with } n = 1,2,..N \qquad \textbf{Eq. 6}$$

$$i_m = \text{round}(m\,\omega) \text{ mod}(M) \qquad \textit{with } m = 1,2\ldots M \qquad \textbf{Eq. 7}$$

where

$$\phi = M(Fr/Fs)$$

$$\omega = M(Fw/Fs)$$

and

N is the number of interpolated x_k values from the non-linear map, **M** is the number of samples in the DW (M= 1024, M=2048 or M=4096), **Fr**, **Fw** and **Fs** are the reading, writing and sampling frequencies (in Hz) respectively. The operator **rmod** gives the remainder of a floating point division and the operator **mod** gives the remainder of an integer division; **round** returns the closest integer.

2.2 INTERPOLATION AND SUB-SAMPLING

Linear interpolation is applied to smooth the sequence x_k generated by the non-linear map and to improve the precision of the reading increment. Given two consecutive values, x_k and x_{k+1}, they will be inserted into the DW at constant sub-sampling intervals h, given by:

$$h = \text{int}\,(F_s/F_{sub}), \text{ with } F_{sub} \ll F_s \qquad \textbf{Eq. 8}$$

where F_{sub} is the sub-sampling frequency; the operator **int** truncates a number, returning an integer value.

The algorithm for the linear interpolation and sub-sampling is defined by:

```
FracWaveInter(n, k, h )
   When  (n mod h = 0) make
        begin
        x_k = x_k+1
        x_k+1 = f(x_k+1)
        x_k+1 = g(x_k+1, min, max)
        end
Return [(x_k+1 - x_k)*n)/h + x_k ]
```

Note that the function $\mathbf{f(x_{k+1})}$ returns a new value produced by the non-linear map iteration and the function $\mathbf{g(x_{k+1}, min, max)}$ maps the value $\mathbf{x_{k+1}}$ between [-1, 1] linearly. The interval [**min, max**] can be found experimentally, applying a large number of iterations over $\mathbf{f(x_k)}$ and verifying the upper and lower limits of the resultant attractor in the phase-space horizontal axis.

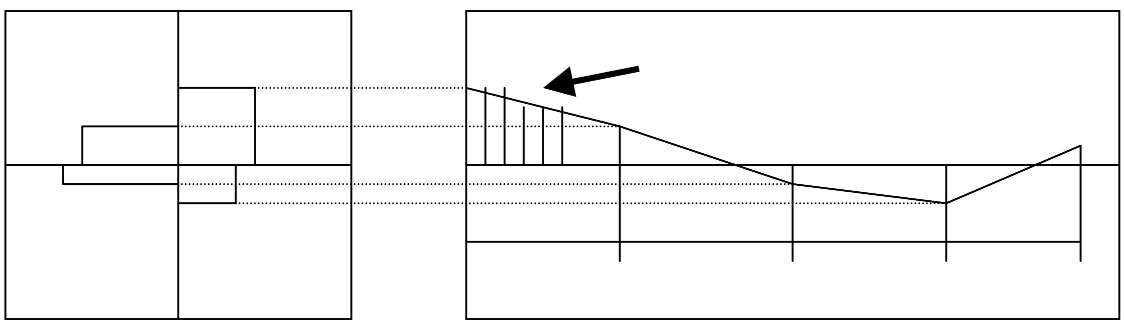

Fig. 4 Diagram of the Linear interpolation applied to smooth the sequence generated by the non-linear map. In this example, values from the vertical axis $\mathbf{Y_k}$ are used to draw the Waveform. Note that it is the same procedure for the horizontal axis $\mathbf{X_k}$.

The reading increment $\mathbf{j_n}$ defined in **Eq. 6** above, a floating-point value, is used to interpolate two consecutive values of the DW. The procedure below performs it:

```
FracWaveInter(n, k, h )
  A1 = int(j_n)
  A2 = (A1+1) mod M
Return W(A1) + frac(j_n) (W(A2) –W(A1))
```

where W(...) is a DW and the function **frac** returns the fractional part of a number.

The following steps are used to start FracWave sound generation and subsequently iterations:

STEP 0
fill the DW with $\mathbf{x_k}$ without interpolation

STEP 1
dynamically refresh the DW using the increments $\mathbf{i_n}$ to write values and $\mathbf{j_m}$ to read them
subsequent call of the FracWaveInter( ) and FracWaveInter( ) functions

STEP 2
Repeat **STEP 1** until the end

3. GESTURAL CONTROL

Due to its compactness, FracWave is well suited for live performances. A number of user interfaces can be used to control the parameters. We used a mouse on a Microsoft Windows environment GUI to enter the parameters A, B and C. We are developing, a small portable gestural interface that is worn on a finger by the performer and gives out a temporal sequence of the finger spatial coordinates[8] to establish the values of the relationship $\mathbf{F_r/F_w}$ in real-time.

To produce the necessary parametric changes we used a mathematical tool, which we developed, called Sound Functors[13]. For this application, curves generated with the gestural interface are used to transform sound parameters.

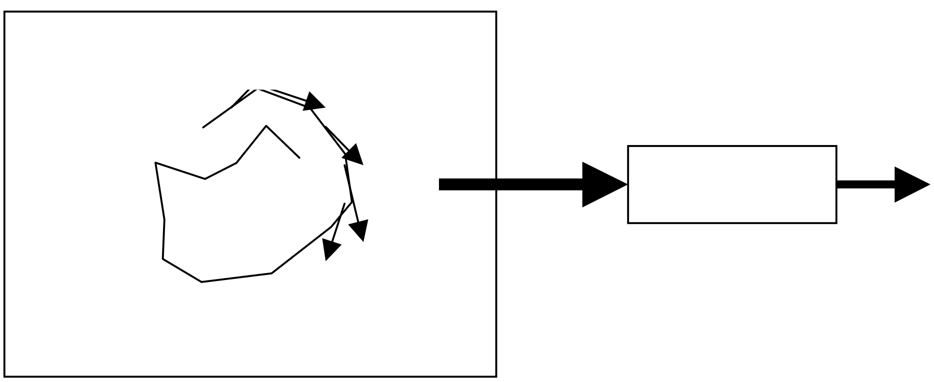

Fig. 5. General idea of the gestural control applied based on a vectorial field to be applied in FracWave

3.1 IMPLEMENTATION

A sequence of points (x_p,y_p) generated by the gestural interface, is used as a control device as showed in **Figure 5**. We take the sequence of points (x_p, y_p) to be equivalent to $(\mathbf{F_r}(p),\mathbf{F_w}(p))$. In this way we define a discrete curve **D** in the plane $\mathbf{F_r}$—$\mathbf{F_w}$, where $\mathbf{F_r}$ and $\mathbf{F_w}$ are the reading and writing frequencies of FracWave. Let $\mathbf{v}_p = (x_p, y_p)$ the vector defining the position of a point on **D.** Now suppose we have a *vector field* **u** defined on **D.** We can assume, without loss of generality**,** **u** to be a normalized vector field, that is**,** $||\mathbf{u}||=1$.

At each point $\mathbf{v}_p=(x_p, y_p)$ we have a corresponding vector $\mathbf{u}_p = (a_p, b_p)$. A deformed curve **D'** is then defined by a new position vector

$$\mathbf{v}_p = \mathbf{v}_p + \quad _p\mathbf{u}_p \qquad \textbf{Eq. 9}$$

where $\ _p$ is a real number which measures the strength of the deformation.

In order to generate variations of an initial curve **D** we must define the deformation vector field **u** and strength parameter . Taken the points (x_{p-1}, y_{p-1}), (x_p, y_p) and (x_{p+1}, y_{p+1}), we define an approximated tangent vector field on **D** as follows

$$\mathbf{t}_p = (x_{p-1} - x_{p+1}, y_{p-1} - y_{p+1}) \qquad \textbf{Eq. 10}$$

In extension, a normal vector field on **D** can be defined as

$$\mathbf{n}_p = (-y_{p-1} + y_{p+1}, -x_{p-1} + x_{p+1}) \qquad \textbf{Eq. 11}$$

and we further define a normalized normal field as

$$\mathbf{u}_p = \mathbf{n}_p/||\mathbf{n}||. \qquad \textbf{Eq. 12}$$

As discussed above, we are interested to produce small deformations in the original **D** curve. We can restrict our parameter to a continuous, bounded function defined as follows:

$$_p = \quad (x_p, y_p) = 1/1+sqr(x_p^2 + y_p^2) \qquad \textbf{Eq. 13}$$

In Non-linear sound synthesis in real-time, this is particularly interesting because the performer can draw any curve, listen to the result and interact with the curve. It is similar to play a musical instrument, but in this case the performer controls a synthesis method.

After obtaining an interesting result, the performer may want to re-produce a family of sounds with the same behavior. If the curve D is complex, it is difficult to re-draw the sequence of points.

Having this in mind, we developed a model with the following features:

- an intuitive control curve **D** drawn with the portable gestural interface
- a family of control curves **D**( ).

CONCLUSION

The aim of this investigation was to produce sounds with a dynamic and rich spectra. We obtained interesting sounds with colourful timbre. It is possible to imagine a new generation of sound synthesis devices using gestural control to produce sounds based upon non-linear dynamics.

This investigation provides a theoretical framework, which expands the techniques for digital sound synthesis. Instead of imitating known sonic behaviour, the goal is to produce unique sounds. The user experiences a deep interaction with sound material, an immersion on soundscapes produced by his own intuition. It is necessary to formulate new models for timbral design, for there is a continuing need of alternative sound sources for Art Music.

ACKNOWLEDGMENTS

This project was realised through the support of the Foundation for Research of São Paulo State (FAPESP) gave to the Gesture Interface Laboratory in which FracWave was developed.

REFERENCES

[1] J. Manzolli, *Musical application derived from FracWave sound synthesis method*, Preprints of the AES Convention, Berlin, 1993.
[2] F.Damiani, J.Manzolli and P. J.Tatsch, *A non-linear algorithm for the design and production of digitally synthesized sounds*, ICMP'99 Proceedings, Campinas, Brazil, 1999.
[3] W. Lauterborn and U. Parlitz, *Methods of Chaos physics and their applications to acoustics*, Journal of the Acoustic Society of America, 84(6):1975-1993, 1988
[4] B. Truax, *Chaotic non-linear systems and digital synthesis: an exploratory study*, Proceedings of the 1990 International Computer Music Conference, 1990
[5] A. Di Scipio, *Composition by exploring of non-linear dynamic systems*, Proceedings of the 1990 International Computer Music Conference, 1990
[6] E. Lindeman, *Routes to chaos in a non-linear musical instrument model*, Proceedings of the 1998 Audio Engineering Society Convention, 1988
[7] J. M. Chowning, *The synthesis of complex audio spectra by means of frequency modulation*, Journal of the Audio Engineering Society, 21(7):526-534, 1979
[8] F.Damiani and G.Mendes, *Interface gestual ultrassônica*, IV Iberchip, Mar del Plata, Ar, 1988
[9] M. V. Mathews, and J. C. Risset, *Analysis of musical-instrument tones*, Physics Today 22(2):23-30, 1969
[10] Karplus and Strong, *Digital synthesis of plucked -string and drum timbres*, Computer Music Journal 7(2):43-51, 1983
[11] Chareyron, *Digital sysnthesis of self-modifying waveforms by means of linear automata*, Computer Music Journal 14(4):25-41, 1990
[12] K. Becker and M. Dorfler, *Dynamical systems and fractals*, Cambridge University Press. 1989.
[13] J. Manzolli and A. Maia Jr., *Sound Functors Applications*, Proceedings of the V Brazilian Simposium on Computer Music, Belo Horizonte, Brazil, 1998.