Possibilities for Dynamical Wave Terrain Synthesis

Stuart James

West Australian Academy of Performing Arts, Edith Cowan University
email: stuart_g_james@hotmail.com

Abstract

For the most part, Wave Terrain Sound Synthesis (WT) has remained within a conceptual domain defined by linear topographical structures deriving essentially from Euclidean and Cartesian geometry. Consequently, the technique has been characterized by simple oscillator and modulator types due to an inflexible process of modifying the phase state within the system. After addressing the limitations inherent in existing methodology, this paper discusses some possible alternatives to linearity by describing various ways of introducing dynamical and pseudo-dynamical systems into the Wave Terrain Synthesis model.

1 Introduction

Wave Terrain Synthesis (WT) (Gold 1978), or what is alternatively referred to as *Two-Variable Function Synthesis* (Borgonovo and Haus 1986), uses a "trajectory" of coordinates to read from a function of two variables (i.e. a data array) in order for the creation of a raw audio stream. Like many synthesis techniques, *WT synthesis* conceptually derives from a fundamental notion of *Wavetable Lookup*, where an indexed table of values is scanned by a linear trajectory generated by incrementing in the positive or negative direction. *WT synthesis* essentially extends this principle to the scanning of a virtual three-dimensional "wave terrain."

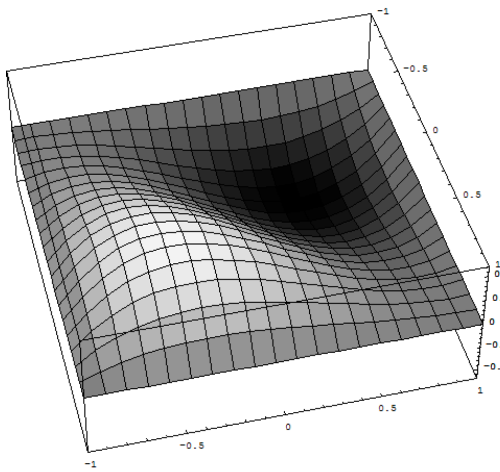

Figure 1. A typical terrain function used for *WT synthesis* described as $f(x, y) = (x - y)(x - 1)(x + 1)(y - 1)(y + 1)$; Image generated by the author using *Mathematica*.

The notion of a virtual "wave terrain", within this context, was first realized by R. Gold (Bischoff, Gold, and Horton 1978). Conventionally this terrain function has been described as a function of (x, y). It has been the function of the trajectory curve that has described the temporal evolution of the system. This curve has been typically expressed as a pair of Parametric equations (i.e. $x = f(t)$, $y = g(t)$) that specify the coordinates (x, y) with respect to time (t). Within this conceptual model, it is the height of the terrain surface (z) at the discrete coordinates described by the trajectory that determines the resulting audio signal.

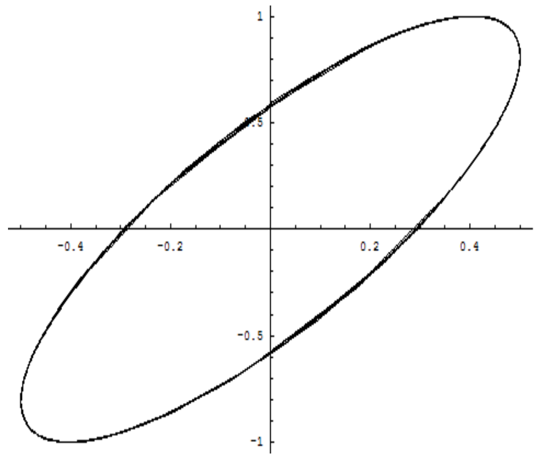

Figure 2. A typical trajectory structure defined by the Parametric equations: $x(t) = 0.5\sin(8t + \pi/5)$; $y(t) = \sin(8t)$; Image generated by the author using *Mathematica*.

The technique has been investigated by a small number of computer music researchers including R. Gold through consultation with Leonard Cottrell (Bischoff, Gold, and Horton 1978), Y. Mitsuhashi (1982), A. Borgonovo, and G. Haus (1984, 1986). More recently *WT synthesis* has been adapted by H. Mikelson (2000), J. C. Nelson (2000), and J. M. Comajuncosas (2000) in association with the readily available freeware programming language *CSound*; a language that has also recently seen the inclusion of a basic *WT synthesis* opcode within canonical version 4.19 (Conder, *et al*. 2002). Other implementations include a very simple linear *terrain~* object as part of the *PeRColate* library for *Max/MSP/Nato* and *Pure Data* (Trueman and DuBois 2001). *Max/MSP* also includes an object *2d.wave~* that uses two instances of a single wavetable for use as a *Two-Variable Function* for scanning by a trajectory (Zicarelli, *et al.* 2001).

Conclusions regarding *WT synthesis* have consistently raised two fundamental observations. The first being the potential scope of the technique due to the interesting and novel effects that it produces; and the second being the need for more thorough research in order to establish an overall conceptual and scientific theory detailing the extended practical use of such a technique (Roads, *et al*. 1996; Nelson 2000). Depending on the methodology, both the concept as well as the results of *WT synthesis* seem to have hovered somewhere within the realms of *Wavetable Lookup* (Roads, *et al*. 1996), *FM synthesis* (Mitsuhashi 1982), *Waveshaping* and *Distortion synthesis* (Comajuncosas 2000), *Morphing* (Comajuncosas 2000), and wavetable cross-multiplication (i.e. *Ring Modulation*) (Borgonovo and Haus 1984, 1986; Nelson 2000). If this might be the case, there begs the question: what exactly might *WT synthesis* actually be? Perhaps for now it is fair to assume that, due to its multi-parameter structure, it represents a "synthesis" of elements drawn from all of these techniques.

2 The Domain of Linearity

In order to predict the resulting waveform, the majority of research into *WT synthesis* has stipulated the need for simple mathematical expressions in the range $[-1 \leq x \leq 1, -1 \leq y \leq 1]$ for use as terrain functions (Mitsuhashi 1982; Borgonovo and Haus 1986). Nevertheless, with the onset of abstract data structures, one could supposedly translate any three dimensional data for use as a virtual wave terrain. For example, constrained Algebraic, Trigonometric, Logarithmic/Exponential, Complex, and Composite/Hybrid mathematical functions (Gold 1978; Mitsuhashi 1982; Mikelson 2000), data extracted from analyses of global seafloor and land topography, or simply a digital image file.

The *Jitter* extendable library to the *Max/MSP* visual programming environment provides a variety of tools for dealing with multi-dimensional data structures (Bernstein, *et al*. 2002). *Csound* on the other hand, without the present facility for handling arrays, has seen users derive terrain functions by cross-multiplying two f-tables (Nelson 2000), or by initialising a list of f-tables that derive a terrain function through the linear interpolation of various one-dimensional curves (Comajuncosas 2000).

Research has also seen the use of simple mathematical structures for trajectory orbits. These have conventionally been in Parametric/Cartesian, Polar, or Algebraic form; structures that, depending on their use, most typically exhibit inherent periodicity.

For the purposes of introducing variables into the trajectory system – if the equations themselves were lacking in useful parameters – one could easily modify them in order to introduce the necessary variables for more flexible control. As another alternative, the trajectory could simply be mathematically processed by a series of functions such as those for affine transformation. In this way the scale (*x, y*), translation (*x, y*), and rotation (θ) of an existing trajectory curve may be altered within the two-dimensional space. In order to ensure the continuity of the resulting waveform while performing *WT synthesis* – in the case of the trajectory exceeding the domain range of the terrain function – the curve could simply be "folded" or "reflected" inward at the terrain boundary points.

By altering the variables involved with the generating or processing of the trajectory, the timbre of a generated tone by *WT synthesis* may be modified with respect to time. If the orbit were to remain periodic, the resulting sound would be characterized by a simple oscillator type with a fixed and static waveform. Below is a list of the various classes of trajectory orbits. Not all of these structures are entirely exclusive, but rather, it is by default that they fall into these existing categories.

Periodic	Lines, Curves, Lissajous Figures, Spirographs
Quasi-Periodic	Invariant Tori, Spirals
Chaotic	Strange Attractors
Stochastic	Random Walks

3 Probable Limitations in Methodological Approach

Like any other sound synthesis technique, *WT synthesis* is recognized by its unique control over the parameters of pitch, intensity, duration and timbre. Pitch is directly related to the speed of periodic oscillation in the trajectory, intensity to the maxima and minima points of the terrain function, and duration for how long the trajectory might continue to change as a function of time. Timbre has usually been more problematic to describe; one description states that timbre may be characterized by the evolution of many parameters within a system (Hourdin, Charbonneau, and Moussa 1997). It is already clear that the output signal generated by *WT synthesis* has mutual dependence on both the wave terrain function and the trajectory orbit. If both of these structures were to be defined via simple means, the timbral result may be one of sonic uninterest.

Is it possible that there may be existing limitations in the many current approaches to *WT synthesis*? In order to make the point, some of this research involved the use of two-dimensional representations of real-world samples using a discrete phase space and pseudo-phase space method. Through this process it was possible to compare, by observation, the differences in complexity between real-world sounds and Euclidean structures. To anyone familiar with deterministic chaos, this activity might seem like comparing the Tibetan Alps with cardboard boxes, but it merely brings one back to the understanding that there exists a fundamental difference between the two. One structure may be adequately defined in terms of Classical Euclidean concepts of dimensionality, and the other requires additional and more specific definitions in terms of box-counting dimension, correlation dimension, and Hausdorff dimension to name only a few. The difference here is that Classical geometrical objects have dimensions defined by whole numbers; the structures in phase space arising from real-world sounds are characterized by a real number dimensionality with an arbitrary fractional part that increases with respect to its geometric complexity.

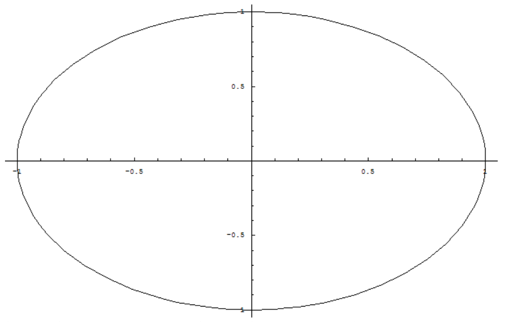

Figure 3a. Illustration of a periodic sinusoidal signal in a two-dimensional phase space (i.e. velocity and amplitude) typifying an archetypal Euclidean structure that may be adequately defined by a topological dimension of 1 (simple closed curve); Images generated by the author using *Mathematica*.

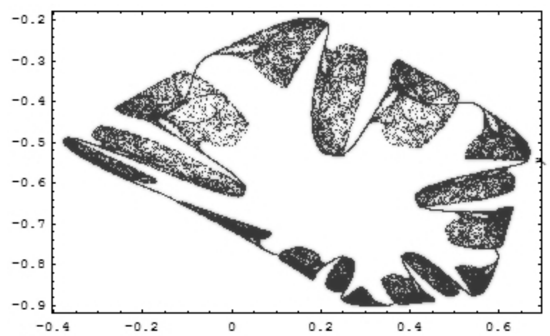

Figure 3b. A chaotic attractor in two-dimensional phase space typifying a complex geometrical structure that requires more specific definitions of dimensionality; the quadratic map is described by a pair of discrete state equations (Sprott 1993)

$$x_{n+1} = -0.5 - 0.6x_n + 0.8{x_n}^2 - 0.5x_n y_n - 0.9y_n + 0.3{y_n}^2$$

$$y_{n+1} = -0.5 + 0.1x_n + 0.6{x_n}^2 - 0.6x_n y_n + 0.2y_n - 0.5{y_n}^2$$

The application of two-dimensional discrete phase space and pseudo-phase space orbits for use in *WT synthesis* were largely successful for both the recreating of the real-world signals as well as the reshaping of those signals. As in conventional *Waveshaping synthesis*, if the shaping function is described by *f(x) = x*, the signal that is input to the system is reproduced. The same situation applies for *WT synthesis* when using an embedded plot in pseudo-phase space; to reproduce the signal, one may use a ramp function *f(x, y) = x* as illustrated in Fig. 4a. The terrain function in this case takes only one parameter of the trajectory signal into consideration. Alternatively, if the terrain were defined by a more complex function of two variables, the incoming trajectory signal would be reshaped with respect to the temporal evolution of both parameters. As in the case of Fig. 4b, this would result in a nonlinear harmonic modulation of the real-world signal.

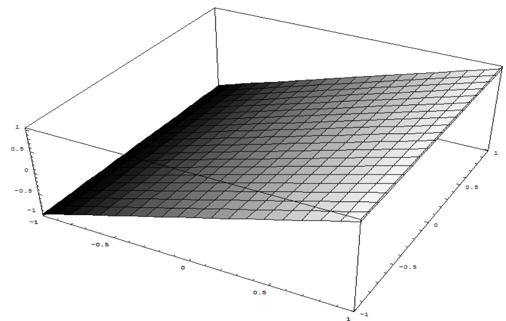

Figure 4a. A ramp terrain function $f(x, y) = x$; Images generated by the author in *Mathematica*.

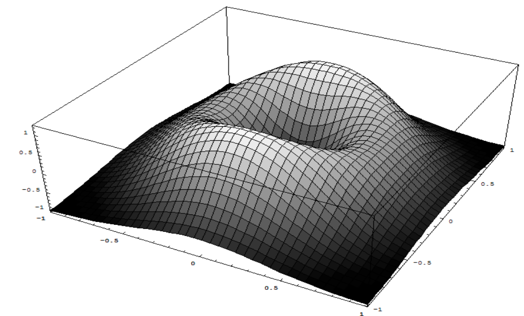

Figure 4b. A more complex contour defined by the function

$$f(x,y) = 2(8x^2 + 12y^2)e^{-4x^2 - 4y^2} - 1$$

Of course, there are a multitude of shaping alternatives that may be applied for *WT synthesis*, and there is already much existing literature concerning these issues with respect to *Waveshaping synthesis*. One may refer to Bruns article (Brun 1979) for a comprehensive theory of varied approaches to one-dimensional systems that may also be applied to the design of two-dimensional *Waveshaping* contours for *WT synthesis*.

What makes the use of phase space and pseudo-phase space representations of real-world sounds useful is for determining how the *WT synthesis* model deals with varying levels of transient complexity; this has been partly for reasons of establishing what roles the trajectory and terrain functions should play for the purposes of increasing the expressive potential of the technique. One may not have realised at this point that it is the trajectory that completely determines to what extent of bearing the terrain function has on the system. For many of the conventionally used approaches – for the purposes of creating pitched sounds for musical application – it is becoming clear that *WT synthesis* technique has been largely restricted to the world of simple oscillator types due to the way in which trajectory orbits have been derived; a situation that – on the surface – seems to render the possibility of using a terrain data array impractical. For example, a small trichromatic 24-bit image file of dimensions 320 x 240, requiring 230,400 bytes of memory, seems far too large a wavetable if the result is much like a simple oscillator. This is largely problematic when the trajectory orbit is periodic, as only a small percentage of the terrain data is accessed for the resulting sound.

Unlike the inherent periodicity found in many conventional approaches to generating trajectories, real-world pitched musical instrument sounds represented in phase space are commonly classified as being quasi-periodic. For example, the embedded plot in Fig. 5 illustrates such a trajectory which exhibits slight signs of chaos, but which conforms to a strong periodic orbit of attraction. While real-world sounds may flexibility move between various states, existing approaches to trajectory synthesis will not allow for wide variational evolution in phase space complexity. It is becoming clear that in order to introduce such complexity into the system, such as that found in the phase space and pseudo-phase space representations of real-world sounds, other options are needed.

Figure 5. A quasi-periodic trajectory generated from an audio signal of a female singing at a fundamental frequency of 364Hz that was embedded in pseudo-phase space with a delay (τ) of 485 samples; Generated by the author using *Max/MSP/Jitter*.

Might there be a more organic approach to the *WT synthesis* model where the user is able to flexibly move between different phase states allowing for high degrees of expressive freedom? Ideally, one wants a system that is capable of functioning autonomously, yet may also be sensitive to changes in parameter. Perhaps certain means for filling the two-dimensional space may be required, and since *iterative function systems* and fractal dimension cross over with the theory of space-filling curves, it could be recursive properties that hold a possible solution to this problem.

Certainly, the introduction of *nonlinear dynamical systems* to the *WT synthesis* model may be a logical step for introducing flexibility into what is already a hugely multi-parameter system. It is through the application of *dynamical systems theory* that one may be able to use simple mathematical structures for producing a great range of complexity (Rodet and Vergez 1999).

4 The Domain of Nonlinearity

The notion of a dynamical terrain within the context of *WT synthesis* seems to have been first alluded to by C. Roads in *The Computer Music Tutorial* (Roads, *et al*. 1996). In this paper, Roads describes the notion of a wave terrain as an undulating surface of wave motion.

If time dependant, a *dynamical system* is characterized by a state that evolves with respect to time. This temporal evolution may be determined both by the current state of the system, as well as by the state of any input parameters used for influencing how the system responds during this evolution. These systems are conventionally described with a series of *discrete iterative functions* or *continuous differential equations*; the inclusion of a *nonlinear* element introduces complexity and sensitivity in terms of how the system evolves and responds to input parameter changes.

Depending on the level of attractive stability inherent within a system, dramatic changes in the input parameters may give rise to a complex behavior known as *chaos*. It is important to understand the fundamental difference between what is *chaotic*, and that which is described as being *random*. *Chaos* is a completely deterministic phenomenon reserved for those problems where extreme complexity results from a sensitivity to initial conditions as applied to *nonlinear* equations (Hilborn 1994). *Randomness* and *chance*, on the other hand, are mathematically understood as *stochastic* processes for which not all parameters can ever be entirely known.

As recent research has proved, *dynamical systems theory* has continued to reveal new ways of understanding the temporal evolution of physical systems. It is the possibility for these systems to move between various phase states, by controlling the relative proportions of periodic and chaotic components in a signal, which is why they have been recognised as being of importance to both the synthesis of sound and the modelling of real-world instruments. Though despite many approaches to the modeling and synthesis of natural sounds having already been developed, researchers have continued to establish new possibilities, some of which have fulfilled some fundamental characteristic properties of real musical instruments: richness of the sonic space, expressivity, flexibility, predictability, and ease of control of sonic results (Rodet and Vergez 1999). Whatever the case may be, it seems that new and experimental investigations into approaches of generating sound may be encouraged for the means of potentially enlarging the scope of available sound processing methods (Röbel 2001).

The next section of this paper briefly addresses some of my own experiments in applying dynamics within the paradigm of the *WT synthesis* model. Some pertinent issues are discussed as to the practicality of such methods, including some other suggestions for future avenues of research.

5 Possibilities for Dynamical Wave Terrain Synthesis

There are fundamentally two exciting prospects for the *WT synthesis* model: firstly its multi-facetted and multi-dimensional structure for the purposes of creating a single audio signal, and the second being the separation of two parts of the model that each have – in turn – mutual bearing on the resultant waveform. In other words, dynamics could be introduced in a great variety of ways within this conceptual framework. To some extent, it seems that the use of dynamics is naturally complimentary to the *WT synthesis* model. These systems give rise to the possibility of crossing between high dimensional and low dimensional structures, they are not restricted by multi-parameter frameworks, they may evolve autonomously, and may also be responsive to parameter changes specific to the system. For many of the problematical issues having been raised, *dynamical systems* seem to be relevant, and may aid in the process of finding practical solutions to these problems in the future.

The idea of applying *dynamical systems theory* to this synthesis paradigm is certainly not motivated by a need to introduce a myriad of new parameters to the existing model, but is rather aimed at a greater hope that *dynamical systems* may potentially introduce more useful and meaningful parameters for the expressivity of the existing model. There certainly could be many examples of *dynamical systems* that do not interact effectively for the purposes of *WT synthesis*. That having been said, care must be taken in either the choice, adaptation, or the design of a system so that it might respond in ways that are going to be useful for *WT synthesis*. What is more, the parameters involved need to be refined such that they are specific to both the problem and the system.

For example, a dynamical terrain function should exhibit quasi-periodicity rather than extreme chaos. This is substantiated by the problem that audio resulting from *WT synthesis* may show signs of DC offset, aliasing, and extreme fluctuations in amplitude. These issues have been typically addressed in the following ways: firstly, the z-domain range of the terrain function should generally be rescaled within the range $[-1 \leq z \leq 1]$ to ensure the resulting waveform is within digital audio range; and secondly, due to the possibility of audio aliasing and clipping, it is recommended that mathematically derived terrain functions remain continuous. Similarly for discrete sets of data, it is recommended that there exhibit a strong continuous tendency characterized by a low deviation. A *dynamical* trajectory system, on the other hand, is not quite so restrictive. Unlike the terrain system, the trajectory may be required to exhibit a great range of phase state complexity for reasons of expressive flexibility.

Based on the knowledge that the resulting sound is determined both by the terrain and trajectory structures, it has been my intention at this initial stage to systematically place equal emphasis on the application of dynamical systems to both of these systems. However, most of the approaches discussed in this paper are as applied to terrain systems.

Terrain systems have been realised in several ways depending on whether the structures are a direct result of the *dynamical system*, or a secondary result due to the dynamical influence of various parameters involved in the linear processing of these functions. Some of my own experiments have included applying *iterative function systems* to video processing for generating fractals, two-dimensional Cellular Automata, and video feedback, as well as applying dynamical systems for the control of parameters involved with the geometrical construction of NURBS (non-uniform rational B-spline) surfaces. Other experiments have dealt with dynamics applied for the purposes of influencing parameters of linear mathematical abstractions, and a process of dynamic modulation using streaming audio signals.

Finally there is a return to the positive argument for introducing dynamics as means of controlling the temporal evolution of the trajectory. Chua's circuit is introduced as one of the most well known systems for efficiently generating a wide range of trajectories (Rodet and Vergez 1999). This system has been described as a series of *continuous differential equations.*

5.1 Video Signal Processing

The reason for my applying video processing within the *WT synthesis* model stems essentially from two fundamentals: firstly, the notion of mapping a multi-dimensional signal process to one that is primarily for audio synthesis, and secondly, to find a way in which to allow for the *WT synthesis* model to become more intuitive. And in order to comprehend the structure of a terrain, in the case of *dynamical systems*, one really needs to be able to "see" the process unfolding. Without having some form of visual representation of the terrain structure, it is difficult to make any judgement as to the best approach to proceed for *WT synthesis*.

There are a number of options for *iterative function systems* in multi-dimensional signal processing. Some of these include classical fractal structures based on the Mandelbrot and Julia set, two-dimensional Cellular Automata, and video feedback.

In my experiments, however, most of these systems have posed unique problems for *WT synthesis*.

Mandelbrot and Julia sets are characterized by the iteration of rational maps in the complex plane. Through this process, some points are "trapped" (remain near their starting point) and others "escape" (move far away). The trapped points make up the set, and are conventionally coloured black, the remaining points being coloured white. From my experience, it appears that unless one introduces gradations of colour – such as a gradation where colour intensity is directly proportional to the distance a point is from the set – the results will remain much like step-noise. Furthermore, due to inherent fractal complexity, a trajectory may need to be rescaled and translated to make use of only specific localized regions of the set in order to both increase the harmonic stability and reduce audio aliasing in the resulting audio signal. In this way it might be possible to produce sounds that are both stable and pitched. Nevertheless, with a linear trajectory system (i.e. periodic) the results are still characterized by a simple oscillator type.

The results of Cellular Automata, in my experience, created waveforms that could be characterised as phase-distorted square waves for both periodic and quasi-periodic scanning trajectories. Upon initializing the automaton, the initial tone produced showed inherent stability. However, this tone slowly digressed as the automaton was left ad infinitum. What is more, due to a lack of a gradual contour, as with the Mandelbrot and Julia sets, it seems that Cellular Automata do not naturally produce the wide timbral variation that might be needed for an expressive *WT synthesis* model. Instead, the sounds are characterised with a very rich spectrum that is prone to becoming more unstable as the automaton progresses.

Video feedback, on the other hand, seems to have been largely more successful for *WT synthesis*. In my experience this was largely due to two significant factors: the terrain remained mapped to the entire region of the data array, allowing for more flexibility with respect to the trajectory orbits. Secondly, the data retained a gradual evolving contour useful for the generation of audio waveforms. Terrain contours produced through this feedback method resulted in ever-evolving and changing kaleidoscopes of unfolding self-similarity. Essentially, for the purposes of generating pitched and stable sounds, problems only arose when the system tended toward the extremes of chaos or periodicity. Chaotic complexity resulted in a tendency toward audio aliasing in the resulting audio signal. Attractive periodicity resulted in a contour much too simple for maintaining a sounding waveform via *WT synthesis*. It seems from this, the answer for dynamical terrain systems lies somewhere in the middle ground between chaos and periodicity for rich and stable musical sounds. For this reason, more investigation is needed in order to determine how the parameters of the system may be refined for easily generating a wide range of quasi-periodic states.

It is in the *Jitter* extendable library for *Max/MSP* that one may be able to realise all of these varied systems. As apposed to the generation of Cellular Automata, for which there is already an object that performs the necessary calculations internally, the approach to generating fractals in *Jitter* is largely inefficient. Problems result when trying to build in functionality for the system to deal with the points that "escape" and tend toward infinity; these points do not need to be recalculated when they become excessively large. The digital approach to video feedback may be implemented within this environment quite easily by using the basic affine transform (i.e. scale, translation, and rotation) with additional control for the user to alter colour scale and proportion levels for red, green, and blue channels during the evolution process. The new frame of video is calculated as the average contour function between the old frame and the transformed frame.

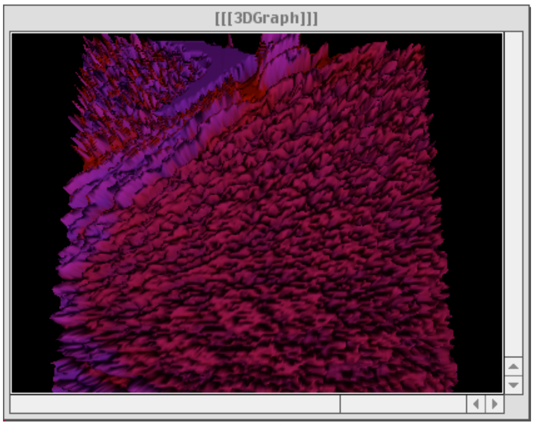

Figure 6a. A three-dimensional OpenGL representation of video feedback showing a chaotic tendency that may ultimately result in aliasing; Both images generated by the author using *Max/MSP/Jitter*.

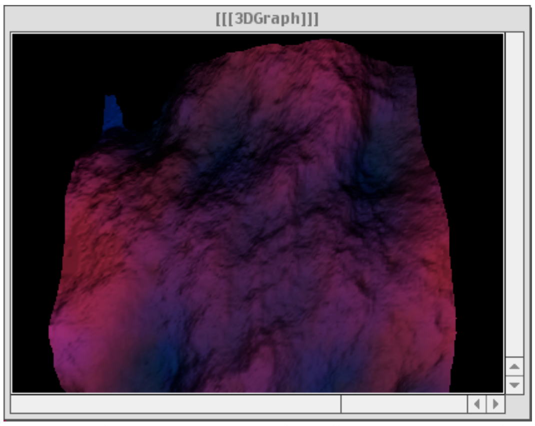

Figure 6b. A three-dimensional OpenGL representation of video feedback showing results from a more quasi-periodic state for producing more stable digital sounds via *WT Synthesis*.

Video feedback runs at about 30fps on a G4 iMAC 800MHz with 256MB of SD-RAM running OS 9.2. However, the frame rate may drop to as low as a third of this initial speed when audio processing is initialised for the trajectory system. One can hear significant audible changes in the resulting waveform that occur precisely with the change of each frame; what might be alternatively described as a frequency artifact resulting at the video refresh rate. This side-effect is certainly not encouraging for the application of video signal processing in *WT synthesis*. What is more, the sheer quantity of data that must be processed at one time for multi-signal processing may also be one of the more significant drawbacks for video processing in *WT synthesis*.

Perhaps another alternative is to use the dedicated *OpenGL API* (Application Programming Interface). This 3D graphics modelling library interfaces with dedicated graphics hardware for both computational performance and efficiency. It means that the process of terrain generation may be potentially less demanding for realtime processing. The *OpenGL* library includes a powerful set of tools as part of its *GLU* utility library, and one of these functions in particular is of great interest for *WT synthesis*.

NURBS (non-uniform rational B-spline) *surfaces* represent geometrical structures that are entirely malleable within virtual three-dimensional space. These surfaces are specified by a series of vertices that are connected in an *n*-dimensional wire-mesh. *NURBS* are used as a higher control function within the *OpenGL* library for producing smooth flowing surfaces by interpolating B-spline (bi-cubic splines) curves through these geometric mesh points.

Within the *Jitter* extendable library to *Max/MSP*, the user is able to export this geometrical information into a separate data array. This structure stores information for every vertex of the geometric structure and the interpolated vertices between them. This information is stored in 12 different planes that give *x*, *y*, *z* coordinate information (planes 0-2), coordinates of textures at vertices (planes 3-4), lighting at vertices (planes 5-7), red, green, blue, and alpha channel components of vertex color (planes 8-11), and the edge flag value for specifying the connections to adjacent vertices (plane 12). It is plane 2 that stores the *z*-domain contour information; this may be applicable for use as a terrain function for *WT synthesis*.

This model provides the user with a flexible and intuitive approach for distorting a terrain function. In realtime, the user has the option for pulling and pushing values that directly correspond to specific geometric control points on the terrain. This control causes slow evolutions in the *Waveshaping* of trajectory signals. The main problem here is finding solutions to how the user might control a large number of parameter values at the one time.

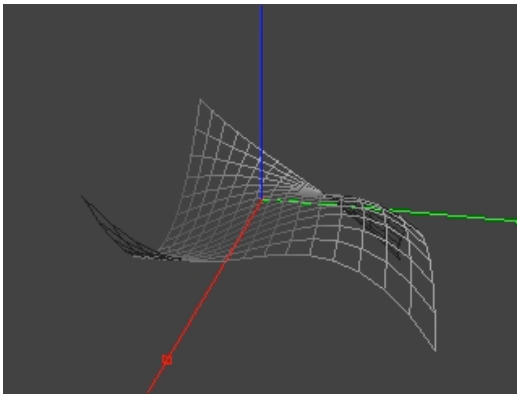

Figure 7. A wire-mesh view of a 16x16 geometrical structure that, when interpolated by variable-order B-spline curves, forms a virtual two-dimensional surface in three-dimensional geometrical space; Image generated in Max/MSP/Jitter.

The idea of using dynamical control is complementary to the high level of control required of the geometric construction. Even though most of my own experiments have involved discrete *iterative function systems* for dynamic control, it certainly may be worth exploring the application of *continuous differential equations* as control systems for NURBS surfaces.

There are certain advantages to this approach for the generation of audio. This is most significantly due to the terrain contour having been constructed from a discrete set of points. Rather than the case of video feedback where structures of self-similarity might derive every such detail of the terrain contour, problems due to audio aliasing are not an issue with NURBS surfaces as applied to *WT synthesis* since the contours never become complex enough in order to produce such effects. However, it seems that this approach does lack a certain amount of the complexity as compared with the video feedback approach due to default geometrical structures defined with smaller numbers of geometrical control points. Unfortunately, if one were to increase this number, the total number of control parameters would increase exponentially. This creates a very difficult controller situation. Perhaps in this situation the user might apply several different dynamical systems for parameter control.

5.2 Audio Signal Processing

The main advantage for generating terrain structures in abstract form is that they may not require such an intensive degree of processing for realtime application. Video processing requires a system to calculate an entire multi-dimensional matrix of data for each evolutionary stage of a system; an approach that might seem like a complete waste of computational resources when one may only be pointing to one sample value at a given time for the purposes of creating the resulting signal. Nevertheless, the introduction of dynamical audio signal processing also means that the rate of processing is increased to the audio sampling rate (some realtime systems, like *Max/MSP*, account for a *vector buffer size* which determines how long a system has for processing in proportion to the sampling rate).

Certainly, an advantage to this abstraction of dynamical terrain generation may be that the sonic effects of the system also occur at audio rate, so unlike the video feedback process, there are consequently no low frequency artifacts in the resulting waveform. In this respect, the terrain is certainly worthy of being derived in this way. The only problem here is that the user would not be able to refer to a visual representation of the terrain. If any unusual problems were to arise during *WT synthesis*, it might not be clear as to what process might be occurring.

This first option for generating dynamical terrains is based on Mikelson's work in creating dynamically modulated surfaces (Mikelson 2000). These terrain functions are derived firstly by linear equations describing the terrain function with respect to two variables. From here, the discrete linear system is substituted with equations that describe a nonlinear dynamical process. In this way, features that might have been introduced by a linear equation – such as the curved surfaces produced by sine, cosine, and logarithmic functions – become subject to movement over the terrain resulting in an ever evolving nonlinear landscape of contour. Within this conceptual framework, the dynamical system controls the positions of various hills, ridges and valleys describing the terrain contour. This depends largely on how the linear terrain function is structured. By separating various components of the equation through processes of addition or subtraction – rather than multiplication or division – certain features of the curve may be controlled independently. Of course, the introduction of more variable parameters may also reflect the way in which the contour may be realized.

This approach, despite having been investigated using only a small number of dynamic systems, has shown promising in producing evolving drones and pads of sound (Mikelson 2000).

Another conceptual extension to the *WT synthesis* paradigm is the generation of terrain functions by the dynamical modulation of two streaming audio signals. This approach may not be a true *nonlinear dynamical system*, but might be termed pseudo-dynamical since it is ever changing and evolving with respect to the dynamics inherent in the two audio streams. Mathematically, the basic notion is not that different to that used by J. C. Nelson (2000) who has generated terrain functions by the cross-modulation of two static wavetables. Obviously, the main difference here is that the terrain function is not static, it is consistently ever changing with respect to the sampling rate.

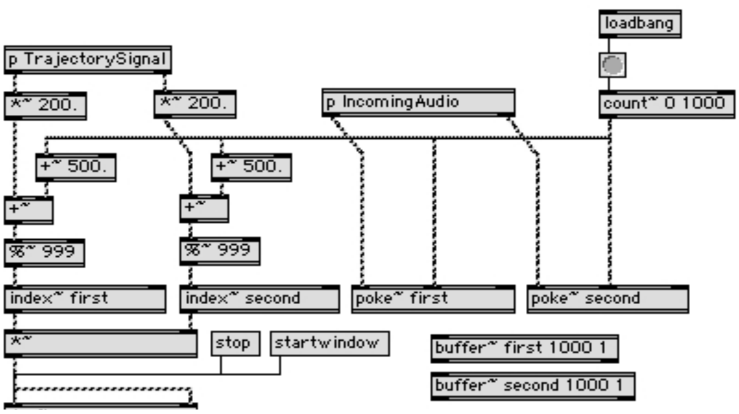

Figure 9. An implementation of Dynamic Wavetable Modulation in *Max/MSP*; By the author.

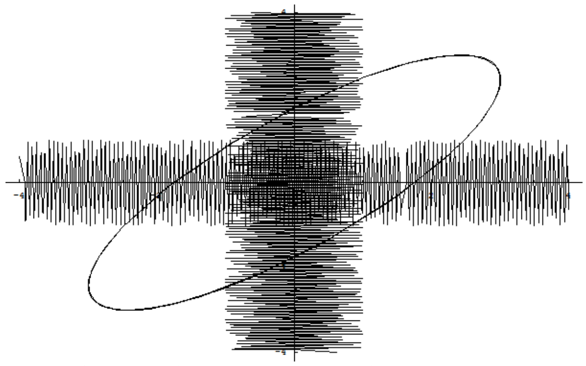

Figure 10a. Dynamic Wavetable Modulation involves the cross multiplication of two streaming audio sources that form a dynamical terrain surface which is subsequently scanned by a trajectory; Images generated by the author using *Mathematica*.

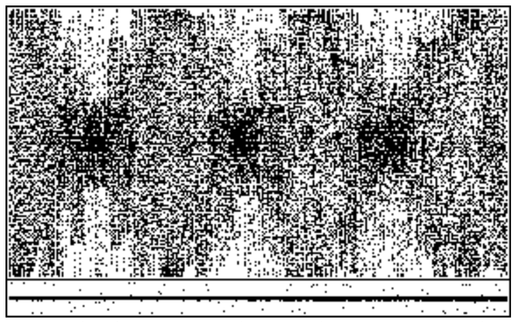

Figure 10b. A sonograph of a result from two sinusoid functions at 1000Hz used for a dynamically modulated terrain for scanning by a trajectory function.

It is unclear as to whether or not this has been implemented, though it is fairly easily constructed. Essentially one requires a pair of allocated memory buffers that, to be safe, need to be at least twice proportional to the amplitude range of the trajectory signal (i.e. how wide the orbit swings in the *x* or *y* direction). In Fig. 9 the trajectory signal is allowed a minimum and maximum orbit deviation of -200 and +200 samples. This could be increased to -400 and +400 if the user might wish. One also needs to account for the shifting changes in parameter values of the pointers reading from the memory buffers so that the information read from the buffers does not cross over where the write pointers are for each sample cycle. This is possible if the reading pointer is positionally offset by half the buffer size (i.e. by +500 samples).

Traditionally, the linear process of multiplying two waveforms has been understood as the digital implementation of *ring modulation*. The addition of a trajectory system that modulates the index value of each buffer value introduces a distortion of the wavelength of the recorded audio stream with respect to time, resulting in an effect known as *frequency modulation* (Chowning 1973). The sonogram illustrated in Fig. 10b exhibits an example of a sound generated this way using two sinusoidal functions; the result is one of extreme spectral complexity.

Finally is an example for what may potentially be an ideal system for generating trajectory curves. This system has become known amongst many for its unusually rich repertoire of dynamical phenomena (Madan 1992), and therefore may serve as an example for generating a broad set of trajectories and periodic orbits for *WT synthesis*. The system is based on a set of *continuous differential state equations* that describe Chua's circuit; it was through the addition of a delay element in the feedback loop that allowed for this system to become particularly useful for sound synthesis (Rodet and Vergez 1999). It seems that the time-delayed Chua's circuit may be particularly useful for generating trajectories for *WT synthesis* due to both its flexibility in the changing of the phase state of a system, as well as its ability to control the frequency produced during the evolution of a signal.

The Time-Delayed Chua Circuit can be related to many sustained instrument models. It is the delay-line that allows for more flexible control over frequency for both periodic and chaotic phase states. The delay-line feedback loop also helps to stabilize whatever nonlinearity is generated by the system.

This conceptual model is much similar to the present requirements of trajectory synthesis where the user may need to specify parameters for controlling pitch, as well as the geometrical structure of a function. The use of a dynamical trajectory system essentially provides the flexible functionality required for sound synthesis by allowing the user to shift between various system states via simple means; this may also mean that the user is not necessarily preoccupied by a myriad of alternative control parameters for generating complexity through multiple processes of linearity.

Like many other useful equations in dynamical systems theory, Chua's circuit is described as a series of *continuous differential equations*. The use of continuous systems as applied to audio signal processing is extremely promising due to inherent signal "continuity." Unfortunately at this stage, for my own experimental purposes, this system has not been implemented to the stage where it may be adequately tested in conjunction with *WT synthesis*.

6 Conclusion

For the most part, the application of dynamical systems to *WT synthesis* was a response from the observation of trajectory complexity in real-world signals when represented in discrete phase space and pseudo-phase space. As a result of effectively recreating and *Waveshaping* real-world samples within the *WT synthesis* model – using two-dimensional discrete phase space and pseudo-phase space plots as trajectory orbits – it has become clear to what extent *WT synthesis* may have remained in the realms of simple structural types due to the use of Euclidean functional archetypes. For this reason, elements of complexity have largely not been fully tried and tested in the *WT synthesis* model.

Even though *WT synthesis* is not that different from *Waveshaping synthesis*, it is potentially more adaptable. Due to its multi-parameter flexibility, the technique can be modified and adapted in terms of methodological approach. It is this multi-parameter and multi-facetted flexibility that needs to be explored in order to achieve a further understanding of the *WT synthesis* paradigm.

It seems that the introduction of *dynamical systems theory* may be a complimentary for the *WT Synthesis* model. Dynamics will allow for a more organic control over the sound synthesis technique by providing more flexibility over the phase state of the system at any given time, and may serve to interface with the many control parameters already inherent within the existing *WT synthesis* model. It seems earlier approaches may too restrictive for expressive control.

The introduction of *dynamical systems* has been discussed at various stages of the terrain function model, and various issues pertinent to both the terrain and trajectory systems have been addressed, including those relating to multi-dimensional signal processing and computational efficiency.

Further research may involve a thorough testing of dynamical systems within the *WT synthesis* model. Other avenues of investigation might include a thorough assessment regarding the use of vector bundles as a means of developing control over the form of a passive periodic element within the dynamical framework of a trajectory orbit. Another possibility may be the introduction of a system for the *dynamic modeling* of system attractors (Röbel 2001) involving various stages of analysis of successive representations of embedded plots in pseudo-phase space.

These structures have great potential for application within this synthesis paradigm. It seems there are many options for future investigation, and *Dynamical systems theory* – an immense area alone – may only be the start of further investigation. It is certain that a great deal of work is still necessary to make any particularly useful and conclusive judgements based on the extended practical use *WT synthesis*. It is also clear that this work is still a long way from drawing any form of comprehensive conclusions concerning the applications of dynamics to *WT synthesis*.

References

Bernstein, J., *et al*. 2002. *Jitter: Tutorials and Topics*. Cycling '74.

Bischoff, J., R. Gold, and J. Horton. 1978. "A Microcomputer-based network for live performance." *Computer Music Journal* 2(3):24-29.

Borgonovo, A., and G. Haus. 1984. "Musical Sound Synthesis by means of Two-Variable Functions: experimental criteria and results." In D. Wessel, ed. *Proceedings of the 1984 International Computer Music Conference*. San Fransisco: International Computer Music Association. pp. 35-42.

Borgonovo, A., and G. Haus. 1986. "Sound Synthesis by means of Two-Variable Functions: experimental criteria and results." *Computer Music Journal* 10(4):57-71.

Brun, M. Le. 1979 "Digital Waveshaping Synthesis." *Journal of the Audio Engineering Society* 27(4):250-264.

Chowning, J. 1973. "The Synthesis of Complex Spectra by means of Frequency Modulation." *Journal of the Audio Engineering Society* 21(7):526-534. Reprinted in C. Roads and J. Strawn, eds. 1985. *Foundations of Computer Music*. Cambridge, Massachusetts: The MIT Press. pp. 6-29.

Comajuncosas, J. M. 2000. "Wave Terrain Synthesis with Csound." In R. Boulanger, ed. *The CSound Book: perspectives in software synthesis, sound design, signal processing, and programming*. Cambridge, Massachusetts: MIT Press.

Conder, K., et al. 2002. *The Alternative CSound Reference Manual Edition 4.22-1*. Massachusetts: MIT.

Hilborn, R. C. 1994. *Chaos and Nonlinear Dynamics: an introduction for scientists and engineers*. New York, Oxford: Oxford University Press.

Hourdin, C., G. Charbonneau, and T. Moussa. 1997. "A Multidimensional Scaling Analysis of Musical Instruments' Time-Varying Spectra." *Computer Music Journal* 21(2):40-55.

Madan, R. N. 1992 (Washington, DC, 9-12 Aug.). "Observing and Learning Chaotic Phenomena from Chua's Circuit." *Proceedings of the 35th Midwest Symposium on Circuits and Systems*. pp. 736-745.

Mikelson, H. 1999. "Sound Generation with the Julia Set." *The Csound Magazine*. http://www.csounds.com/ezine/summer1999/synthesis/

Mikelson, H. 2000. "Terrain Mapping Synthesis." In R. Boulanger, ed. *The CSound Book: perspectives in software synthesis, sound design, signal processing, and programming*. Cambridge, Massachusetts: MIT Press.

Mikelson, H. 2000. "Terrain Mapping with Dynamic Surfaces." *The Csound Magazine*. http://www.csounds.com/ezine/spring2000/synthesis/

Mitsuhashi, Y. 1982c. "Audio Synthesis by Functions of Two Variables." *Journal of the Audio Engineering Society* 30(10):701-706.

Nelson, J. C. 2000. "Understanding and Using Csound's GEN Routines." In R. Boulanger, ed. *The CSound Book: perspectives in software synthesis, sound design, signal processing, and programming*. Cambridge, Massachusetts: MIT Press. pp. 65-97.

Roads, C., *et al*. 1996. *The Computer Music Tutorial*, Cambridge, Massachusetts: MIT Press.

Röbel, A. 2001. "Synthesizing Natural Sounds Using Dynamic Models of Sound Attractors." *Computer Music Journal* 25(2):46-61.

Rodet, X., and C. Vergez. 1999. "Nonlinear Dynamics in Physical Models: Simple Feedback-Loop Systems and Properties." *Computer Music Journal* 23(3):18-34.

Rodet, X., and C. Vergez. 1999. "Nonlinear Dynamics in Physical Models: From Basic Models to True Musical-Instrument Models." *Computer Music Journal* 23(3):35-49.

Sprott, J. C. 1993. "Automatic Generation of Strange Attractors." *Computer & Graphics* 17:325-332. Reprinted in C. A. Pickover, ed. 1998. *Chaos and Fractals, A Computer Graphical Journey: Ten Year Compilation of Advanced Research* (Ed.). Amsterdam, Netherlands: Elsevier. pp. 53-60.

Trueman, D., and R. L. Dubois. 2001. *PeRColate: a collection of synthesis, signal processing, and video objects (with source-code toolkit) for Max/MSP/Nato v. 1.0b3*. Computer Music Centre: Columbia University.

Zicarelli, D., *et al*. 2001. *MSP: Getting Started; Tutorials and Topics; Reference*. Cycling '74.