Extensions to Dynamic Wave Terrain Synthesis for Multidimensional Polyphonic Expression

Sohejl Zabetian

Masters Thesis
MSc. Sound and Music Computing

Faculty of Engineering and Science
School of Information and Communication Technology
Aalborg University Copenhagen

August 2018

Abstract

This paper describes the design and implementation of a real-time polyphonic wave terrain synthesis instrument utilising MPE technology to enhance musical expressivity. It is argued that MPE provides solutions to many of the inherent difficulties faced by wave terrain synthesis, and that visual feedback from wave terrain synthesis can provide a more intuitive understanding of MPE control mechanisms. This hypothesis is examined through the mapping of synthesis parameters to that of an MPE interface, and system analysis based on sound synthesis and digital instrument design theories.

Extensions to wave terrain synthesis include dynamic surface modulation and the use of nonlinear dynamical systems as an effective means of introducing flexibility into the wave terrain synthesis model.

User evaluations and analysis of the instrument are provided, indicating a successful instrument design and general interest in visual based sound synthesis methods and expressive interfaces.

Acknowledgements

I would like to express my gratitude to Stefania Serafin and the whole Sound and Music Computing team at Aalborg University Copenhagen for providing a course which helps satisfy both creative and scientific curiosities. I would also like to thank all my fellow SMC students, who have been family for the past two years.

Contents

Chapter 1

Introduction

Since the dawn of electricity, musicians, engineers, and enthusiasts have been attempting to exploit technologies at their disposal to create new and intriguing musical instruments. Some have succeeded, few have defined music for generations, but sadly most have passed by unnoticed or forgotten. The reality is, design of electronic and digital musical instruments is fraught with hidden complexities. Even to this day, interfaces of most successful digital instruments are built upon or are deeply inspired by already familiar acoustic instruments. Digital technology has given us unprecedented access to a diversity of sounds, yet the abstract nature of sound synthesis makes finding a musically expressive means of control a complex undertaking.

In acoustic instruments, the interaction between sound and gesture is defined by real-world physics, but in the digital world, the rules are determined by us. This is not a trivial matter - the feedback and physical interaction of acoustic instruments are difficult to model, and even more difficult to adapt to abstract forms of sound generation. Yet through technological developments in sensor technology, material science, and signal processing we are coming ever closer to achieving this. The designs of digital musical instruments seem to be shifting focus towards capturing the most desirable trait of acoustic instruments - ’musical expressivity’, be it in the form of gestural nuances or physics-inspired haptic and acoustic feedback.

The notion of expressivity is one engraved in human nature, and is often best understood through intuition. Yet to create a system that encourages musical expressivity, it needs to be understood not only through intuition, but also through technical and scientific methodologies. With a deeper understanding of these aspects, instruments of the future may inspire creativity in ways we cannot yet envision, and open up a world of sounds we have not yet heard.

This project is but one new exploration of this fascinating topic.

1.1 Why Wave Terrain Synthesis? Why MPE?

Wave terrain synthesis is a relatively obscure and unexplored form of sound synthesis first envisioned in 1978. Since then, it has received little attention besides a small number of academic studies. The reasons for this may be diverse, but it is most likely due to the its multifaceted nature (it can behave as a combination of many other synthesis and processing methods), and a lack of appropriate hardware technology and interfaces. Recent developments in touch technology and MPE interfaces however, seem to offer a solution to these issues and seem well adept to providing an interface for effective gestural control of sound synthesis parameters.

To fully grasp *Wave Terrain Synthesis*, it can be useful to understand it conceptually first. To generate an audio signal, wave terrain synthesis relies on two independent structures - a *terrain* and an *orbit* (also referred to as a trajectory). A terrain is analogous to the dips and peaks found in the grooves of vinyl records, which describe the spectral content of the sound waves etched onto them. An orbit is comparable to the stylus that vibrates within the grooves of the record to produce sound, but one that is not limited to following the grooves in a cyclical manner and that can comprise of any shape or size. In essence, an orbit defines a series of coordinates that are used to read from a terrain function describing a three-dimensional surface - the result of which produces an audio spectrum dependent on the properties of both structures.

MPE (Multidimensional Polyphonic Expression) is concerned with improving the unresponsive and discontinuous interfaces of most digital musical instruments by offering continuous control, finer resolution, and more degrees of freedom. Despite being a relative newcomer, MPE instruments have gained plenty of publicity and are being praised for improving expressive capabilities of digital musical instruments. It is the belief of many, that digital instruments of the future will all follow in the footsteps of MPE and its goal of improving musical expressivity.

1.2 Research Aims

This project aims to accomplish four main objectives:

1. Implement wave terrain synthesis as a Max/MSP instrument and investigate its use with an MPE interface - namely, the ROLI Seaboard.

2. Extend the wave terrain synthesis algorithm explore potentially useful modulation techniques.

3. Investigate the use of nonlinear dynamical systems in wave terrain synthesis.
4. Evaluate wave terrain synthesis through user testing and identify strengths, weaknesses and potential future improvements.

Chapter 2

Background

2.1 Related Sound Synthesis Methods

This section will provide a concise theoretical outline of the sound synthesis techniques relevant to Wave Terrain Synthesis and digital musical instrument design. The concepts and formulae discussed will be useful in the analysis and understanding of Wave Terrain Synthesis. For a more detailed overview of sound synthesis methods consult [1] and [2].

2.1.1 Early Methods and Graphical Synthesis

The relationship between sound and image has a rich and illustrious history. As early as 1877, after his development of the phonograph, Thomas Edison described the idea of a "Kinetoscope" that would "do for the eye what the phonograph does for the ear". Interestingly, the first abstract sound synthesis method was developed through motion picture technology - where hand-drawn waveforms were photographed and photo-optically printed onto film to be played back as sound [3]. This provided the spark for future innovations in optical techniques and graphical sound synthesis. Optical techniques have even be used to control analogue synthesis. Daphane Oram 1973 would draw control functions for analogue synthesisers onto transparent film. An optical scanning head would then transform the image into an electronic control voltage fed into various modules of a synthesiser. The control functions would determine pitch, vibrato, tremolo, filter settings, and amplitude level of several voices. Graphical control of sound synthesis began with experiments of Mathews and Rosler. Perhaps the most prevalent was the UPIC system, combining various synthesis methods with a flexible graphical user interface to create a unique approach to sound composition.

2.1.2 Wavetable Synthesis

Wavetable synthesis is a widely used sound synthesis method for efficient creation of musical tones. Similar to most current software synthesis systems, it is based on a table-lookup procedure, which is generally the most computationally efficient method of generating a periodic digital signal or sample of a single period of an instrument tone.

Its efficiency relies greatly on pre-computing the Discrete Fourier Transform (DFT) of the waveform spectrums, rather than computing them in real-time. A single period of the signal is stored in a circular buffer of N sample values, constituting a single period of the desired signal. For a table length of N samples, a sample rate of fs, and a phase increment $p = 1$ sample per time step, the output signal will have a fundamental frequency of fs/N. The table pointer (or phase accumulator) can be incremented at variable rates by skipping samples, effectively shrinking the size of the wavetable in order to generate different frequencies, and resulting in non-integer increment values for most values of N, f and fs. This is problematic because wavetable values are located by integer indices, which means that an integer value needs to be derived from a non-integer increment [1].

The simplest solution would be to truncate or round the fractional part of the phrase increment value to the nearest integer. This introduces unwanted noise to the signal, so an interpolation function is usually used, the most common being *linear* interpolation [4]. It works by effectively drawing a straight line between two neighbouring samples and returning the appropriate point along that line. More specifically, if η is a number between 0 and 1 which represents how far we want to interpolate a signal y between time n and time $n+1$, then we can define the linearly interpolated value $\hat{y}(n+\eta)$ as follows [5]:

$$\hat{y}(n+\eta) = y(n) + \eta \cdot [y(n+1) - y(n)] \tag{2.1}$$

The interpolation technique used can have a significant effect on the quality of the resulting signal. More advanced techniques include cubic and bandlimited interpolation, or higher-order integrated wavetables [6], [7], [8].

For musical purposes it is generally desirable to use time-varying waveforms. This can be achieved in wavetable synthesis with two main strategies - *wavetable crossfading* (also known as *vector* or *compound synthesis*) where instead of scanning repeatedly, the oscillator crossfades between two or more wavetables over the course of an event, and *wavestacking* where each sound event results from the addition of several waveforms [1], [8], [9].

2.1.3 Waveshaping Synthesis

In its simplest form, waveshaping synthesis involves passing a signal x through a *shaping function* to alter the shape of the input wave. In other words, if f is the identity function $f(x) = x$ then the signal will pass through unchanged. Similarly to wavetable synthesis, waveshaping can be accomplished by means of a table-lookup procedure. Computing the $f(x)$ simply involves looking up the value corresponding to x in an f table, and possibly interpolating between values. There are however several extensions to this process that allows for waveshaping synthesis to cause interesting musical variance in a spectrum.

An example of such an extension would be to multiply the input signal x by some value a before applying f to it, giving us the function $f(ax)$. By varying values of a over time, different sub-ranges of the function f will be scanned, causing the output wave to vary in shape and produce differing harmonic content as a result. One can further extend this to allow for inharmonicity by multiplying the output signal with a sinusoid with some arbitrary frequency C. This causes every harmonic spectral line of frequency $k\omega$ and amplitude $h\omega$ to be replaced by two spectral lines of frequencies $C + k\omega$ and $C - k\omega$, each with amplitude $h_k/2$. This can be expressed as

$$\begin{aligned} \cos Ct \sum_{k=0}^{\infty} h_k \cos k\omega t &= \sum_{k=0}^{\infty} h_k \cos C_t \cos k\omega t \\ &= \sum_{k=0}^{\infty} \frac{h_k}{2} (\cos(C + k\omega)t + \cos(C - k\omega)t) \end{aligned} \tag{2.2}$$

Choices in the value of C can have a significant effect on the output spectra. For example, if values of C cannot be expressed as a ratio of integers, inharmonicity is produced. If C is represented by a simple fraction, some noise or "roughness" is introduced, but the impression of harmonicity remains. A similar effect can be achieved through *iterated waveshaping* (see section 2.1.4).

2.1.4 Functional Iteration Synthesis

The term "functional iteration" is commonly found in chaos theory (the mathematical modelling of nonlinear dynamical systems, see section 2.2) to refer to the iterated application of nonlinear functions. This process was adopted and applied to sound synthesis by Di Scipio, to create a class of synthesis techniques based on the same approach (FIS) [10], [11], [12].

A stream of audio samples are calculated by applying a set of transformations f_i^m to a set of data $x_{0,i}$ to obtain a sequence of output sample streams $x_{n,i}$. Essentially the output is a sequence of nth iterates of some m-parametric function applied to a set of initial data:

$$x_{n,i} = f_i(f_i(...)f_i(x_{0,i})...)) = f_i^m(x_{0,i}), \tag{2.3}$$

where n is the index of the iterate, i is the index of discrete time, and m is a set of parameters for function f. Any nonlinear f can be used, and smoothly changing control functions can be used to update parameter values at each iterated process [10], [12]. It should be noted, that although the characteristics of the output will be altered by the choice of f, the process of iteration has a far greater influence on the output spectra.

The musical functionality of FIS methods can be found within their chaotic structures. Despite possessing inherent aperiodic behaviours (see section 2.2.3), there is a natural order to be found within chaotic systems. This can be useful for the modelling of natural acoustic sounds which inhibit similar behaviours [10]. Furthermore, the iterative process of FIS can be used with other forms of synthesis, such as waveshaping - by feeding the output sample back into the waveshaper, allowing for the generation of chaotic nonlinear distortion through *iterated waveshaping*.

2.1.5 Scanned Synthesis

Inspired by psychoacoustics and haptic interaction, scanned synthesis aims to provide a platform for real-time control of timbre [13]. It is based on principles of physical modelling synthesis, specifically a finite element model of a 'circular string' of damped mass-spring systems. The string is vibrated at *haptic frequencies* of (0Hz to 15Hz) as this provides the frequency range of spectral changes that are observed to be musically interesting, and that also adhere to the frequency rate of human motor control abilities [14]. To create an audible signal, the series of masses are "scanned" at sampling rate similarly to wavetable synthesis (see section 2.1.2), with table interpolation between successive updates. The rate at which the scanning function is called, determines the pitch of the output, and is independent from the dynamic system control which is used to create variance in timbre. Extensions to this method include higher dimensional scanning trajectories, advanced excitation models, and the freedom of masses to move in two dimensions [15], [3]. Scanned synthesis could be considered an extension to wave terrain synthesis in many ways, reasons for which will become clear in section 2.4.1.

2.2 Nonlinear Dynamical Systems

The study of nonlinear dynamical systems is a vast subject. This section will provide a brief overview of key concepts and definitions, alongside details of its relevance to musical applications. A detailed description of this field can be found in [16] and [17].

2.2.1 Dynamical Systems

Dynamics is a subject that deals with change in systems that evolve in time. A dynamical system can be described as any set of equations giving the time evolution of a systems state from a knowledge of its previous history [18].

Dynamical systems whose state is known only at a discrete set of times (iterated maps) can be defined by the relation

$$x_{n+1} = F(x_n) \tag{2.4}$$

where x_n is the current state of the system, and $x_n + 1$ is the state of the system after one interval of time has passed. By iterating this equation multiple times, we can find subsequent states of the system. Systems in which F is not time-dependant (such as the Lorenz system) are referred to as autonomous, whereas time dependant systems (such as the Duffing system), are referred to as non-autonomous [19].

2.2.2 Phase Space and State Space

Phase space is a space which represents all possible states of a system by depicting how each possible state corresponds to a unique point in the phase space. The evolving state of a system traces a path (*phase space trajectory*) through the high-dimensional space over time. The path of the trajectory represents a single possible progression through the phase space for particular initial value (or set of values) x_0.

The movement of trajectories are influenced by *attractors*, which are defined as sets of points that influence the movement of trajectories in the phase space. The sets of points whose orbits converge to an attractor of a system is called the *basin of attraction* of the point [20]. Various types of attractors exist, such as fixed points that can be either stable (attracting trajectories within their basin of attraction) or unstable (repelling trajectories), and chaotic attractors which are highly sensitive to initial conditions and do not produce periodic behaviour. Nonlinear systems often contain multiple attractors which may overlap in their basins of attraction, producing potentially complex and chaotic results [19].

2.2.3 Nonlinearity and Chaos

By definition, nonlinear systems are systems in which the output is not proportional to the change of the input. Specifically, they do not adhere to the superposition principle or homogeneity properties of linear systems:

$$\begin{aligned} f(\alpha x) &= \alpha f(x) \\ f(x+y) &= f(x) + f(y) \end{aligned} \tag{2.5}$$

A consequence of the inclusion of nonlinear terms in dynamical systems, is that their behaviour may become highly sensitive to initial conditions [19]. This is known as the *butterfly effect*, which describes how a small change in one state of a deterministic nonlinear system can result in large difference in a later state. Although inherently deterministic, small differences in initial conditions (such as rounding errors in numerical computation) yield widely diverging outcomes for such dynamical systems, rendering long-term prediction of their behaviour impossible in general. The behaviour of these systems and the underlying patterns within the apparent randomness of complex systems is the basis of chaos theory. It was summarised by Edward Lorenz as:

> "*When the present determines the future, but the approximate present does not approximately determine the future*"

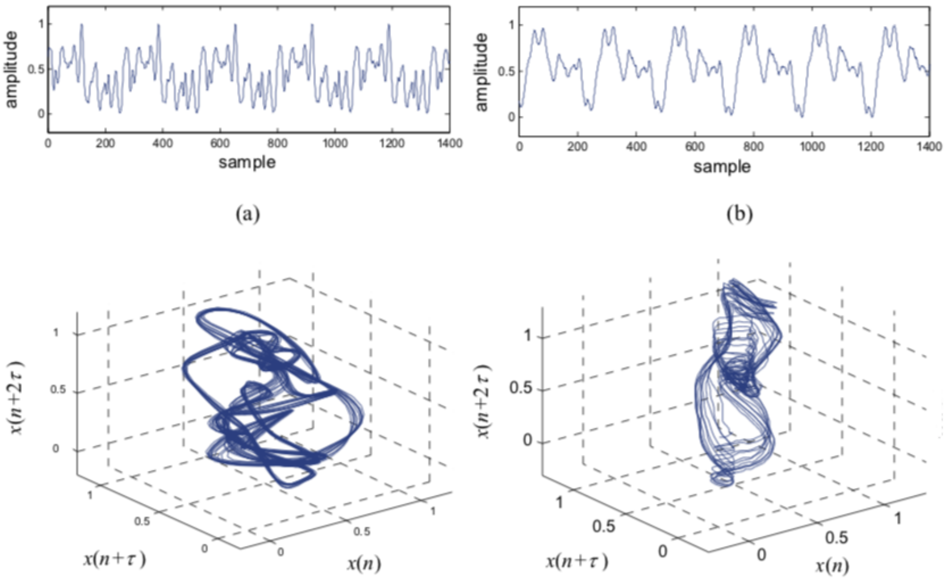


Figure 2.1: Temporal waveforms and phase space portraits of a clarinet and cello, exhibiting signs of chaos but conforming to a strong periodic orbit of attraction [21].

2.2.4 Music and Nonlinear Dynamical Systems

Nonlinearity is an inherent characteristic of acoustic instruments. It is often generated as a result of the complex relationship between instrument control mechanisms and sound generators. Nonlinear feedback from acoustic instruments can often affect aspects such as timbre, pitch, and dynamics - consequently influencing interaction with the instrument in a feedback loop. The exact way in which this complexity influences sound and interaction is entirely dependent on the physical nature of the instrument. Many traditional instruments can be driven to a state of chaotic behaviour, characterised by noisy, rapidly fluctuating tones [22]. Examples of this can be found in saxophone styles in which the vocal sounds interact directly with vibrations in the saxophone [23], or in the use of feedback with the electric guitar, where the guitar becomes part of a complex driven system and the instruments behaviour is dependent on factors such as room acoustics, and the location of the performer in relation to loudspeakers [24].

The inherent complexity of acoustic instruments can be visualised by phase space visualisation techniques (see figure 2.1). This complexity is one of the main characteristics that often make them superior to digital instruments in terms of interaction and expressivity. To achieve the same kinds of results in digital musical instruments, particular attention must be paid to capturing nuances in gestural input and its mapping to sound synthesis parameters. For accurate models of these interactions, one must look to physical modelling synthesis [5]. Abstract synthesis methods however, have no direct link to sound generation parameters of acoustic parameters, yet they can benefit enormously from acoustic inspired complexities such as nonlinearity, chaos, and convergent/divergent mappings (see section 2.3.2).

2.3 Design of Digital Musical Instruments

The design of digital musical instruments is a broad and interdisciplinary subject, which includes highly technological areas (e.g. electronics and sensor technology, sound synthesis and processing techniques, computer programming), human-related disciplines (associated with psychology, physiology, human-computer interation), plus all the possible connections between them (e.g. mapping techniques) and most essential of all, music in all of its possible shapes [25]. A digital musical instrument (DMI) can be defined as a system which affords real-time control over digital audio synthesis algorithms using some sort of user interface. While approaches to user-interface and media synthesis may vary considerably, a unifying feature of DMIs is that, unlike acoustic systems there exists no physical relationship between user input and its effect on the system state. Instead, associations between gesture and sound (or other media) must be designed or otherwise generated before the system becomes functional; this mapping from sensed gestures, postures or other phenomena to media control dramatically impacts the response of the system and thus the experience of the performer and audience [26].

This section will provide a theoretical overview of elements of DMI design that are relevant to the expressive control of sound synthesis. For an in-depth insight into this topic as a whole, consult [25] and [26]. The latest research and articles on the topic can be found in the proceedings of New Interfaces of Musical Expression (NIME) conference [27].

2.3.1 Expressiveness in Digital Instruments

The term 'expressiveness' is widely regarded as the communication of feelings, emotions and sentiments amongst other things. Much like in other art forms, expression is the inherent purpose of music and musical performances and thus suffers from the same complexities in being measured and described scientifically. Nonetheless, musical expression can be defined acoustically by temporal variations in timing, pitch, dynamics, timbre and space, which demonstrates the fascinating idea that intentions and emotions of the mind can indeed be encoded into sound. Expressive variations in acoustic signals have been previously considered in the context of modelling expressive performances for analysis [28], [29], [30], synthesis [31], [32], [33], and adaptive audio effects based on expressive features [34].

For a digital musical instrument to be expressive, it should provide a platform where the performer is able to convey expressive intentions. That is, it should provide intuitive control over acoustic parameters. The way in which a digital instrument accomplishes this is based entirely on its design, and it is usually desirable for it to do so in a unique manner. This can be demonstrated with a comparison of the piano and violin. Aside from differences in the sound source, the physical design of the instruments encourage completely different playing styles, thus being able to achieve musical expressivity in unique ways.

2.3.2 Parameter Mapping Strategies

In acoustic instruments, the relationship between control mechanism and sound generator is bound by physical laws and is consequently complex. In digital musical instruments however, control parameters of the instrument (derived from gestural interaction) and parameters of the sound synthesis are separate, meaning that the relationship between them has to be defined. The process of establishing this relationship is known as mapping. The importance of the mapping strategy in DMI design can not be understated. By establishing the mechanics of the instrument, mapping dramatically impacts the response and characteristics of the system and thus the experience of the performer and audience [26], [35].

A number of typologies for considering mapping approaches exist. Firstly, the complexity of the mapping scheme must be defined. Mappings may be *one-to-one*, *convergent* (in which multiple source parameters affect a single destination), *divergent* (in which one source parameter affects multiple destinations), or a combination [26]. Studies have found that instruments with complex mappings are generally preferred and that real-time control can be enhanced by multi-parametric interfaces [36]. Mappings are also usually assumed to be instantaneous, in that the output of a mapping destination depends on its input at that time. However, interesting behaviours can also be produced by keeping track of the history of mapping functions. Examples include modifying the responsiveness of the instrument based on input/output history, measuring average activity, and predicting gestures [25]. Another consideration is as to whether the mapping strategy should be determined using generative mechanisms (such as neural networks) or explicit mapping strategies. The former provides a strategy by means of internal adaptations of the system through training, and the latter involves explicitly defining relationships between parameters [37].

2.3.3 Gestural Control of Sound Synthesis

Gestural control of sound synthesis can be seen as a highly specialised branch of HCI involving the *simultaneous control of multiple parameters, timing, rhythm and user training* [38]. Advances in sensor technology and computing has lead to a surge in tactile DMI controllers that are able to capture the nuances of human gestures (a great example being MPE, see section 2.3.4). The aim of such instruments are to reproduce or expand upon complex interactions found in acoustic instruments, most often to enhance expressive capabilities of the performer. To achieve this, it is crucial to determine an effective mapping scheme between gestural input and sound synthesis parameters. This is the primary concern of this section. Other important considerations include typologies of gesture, gesture acquisition and suitable synthesis algorithms [39].

The variety in characteristics of sound synthesis variables makes gestural mapping highly dependent on the sound synthesis method chosen. A spectral form of synthesis such as additive synthesis for example, would have parameters such as amplitudes, frequencies and phases of sinusoidal partials. The mapping between gestural variables and these parameters is abstract, much like the synthesis method itself. A physical model however, has variables describing physical parameters of an instrument, thus allowing for a direct mapping between gesture and synthesis to be made. Determining suitable mapping strategies for abstract synthesis methods can be difficult. Most often they are based on perceptual variables such as timbre, dynamics and pitch, but they can also be based on other perceptual characteristics of sound, or be chosen with arbitrary reasoning [40].

An issue often found in parameter mapping is the need for entirely different mapping strategies to be implemented for individual controllers. An interesting solution to this is discussed in [41], where two independent layers of mapping are used- One for the mapping of control variables to intermediate parameters, and another for the mapping of intermediate parameters to synthesis variables. This essentially creates one layer of mapping that would need not be changed when changing between controllers, thus simplifying the process and creating continuity in the way the system is mapped.

2.3.4 Multidimensional Polyphonic Expression

As of January 28th 2018, the MIDI Manufacturers Association (MMA) ratified a new extension to MIDI, MPE (MIDI Polyphonic Expression) in an effort to respond to the new expressive capabilities offered by MPE controllers, such as the ROLI Seaboard, Linnstrument, K-Board and Continuum [42], [43], [44], [45]. In the absence of MIDI, this technology is called multidimensional polyphonic expression.

In MPE, each note is assigned its own MIDI channel, so that channel-wide expression messages can be applied to each note individually. Prior to this, control messages and pitch bending were applied globally to all notes in the channel, thus affecting all notes being played. This prevented polyphonic pitch bends and polyphonic Y-axis control (which uses control change messages) over a single MIDI channel. The main differences in MPE and traditional MIDI can be summarised by the following [43]:

- Response to finger pressure: In a traditional MIDI keyboard, pressure (aftertouch) is sent only after the key is fully pressed. MPE instruments improve on this by sending a continuous stream of pressure messages, so as to measure both the lightest and heaviest touch.

- Response to Y-axis movements: Adding another dimension of control for the performer. This is usually implemented as a gestural performance control for continuous modification of timbre.

- Response to X-axis movements: Allows for micro-tonal control of pitch by removing note restrictions based on the 12-note scale.

The exact implementation and gestural control of these aspects depends on the hardware and sensors used by the MPE instrument in question. As an example, the ROLI Seaboard treats its instrument as one whole surface, meaning that gestural movements can be made on the entire surface of the instrument (see section 3.1.2). It is important to note that parameters of MPE can be mapped to anything, meaning that interaction with MPE instruments is defined by the mapping strategy utilised.

2.4 Wave Terrain Synthesis to Date

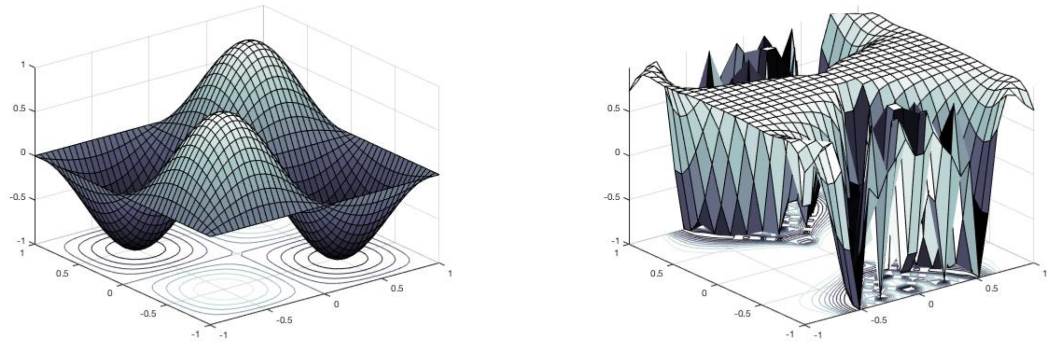
2.4.1 Theoretical Definition

The notion of wave terrain synthesis was first conceptualised by Gold [46] as a new means of generating audio waveforms, with the first subsequent implementation named the 'terrain reader' [47]. It was Mitsuhashi however who used the concept to formally propose a new sound synthesis technique, which he describes as audio signal synthesis by *functions of two variables* [48]. Fundamentally, the technique is based on generating an audio signal by the sampling of an n variable function along an orbit determined by n expressions of time-dependent variables. In most studies, $n = 2$ due to complexities and processing requirements of $n > 2$. Audio samples can be calculated directly using an arithmetic approach, or read by means of an n dimensional table lookup [49]. The orbit is given by $x = x(t)$ and $y = y(t)$, where x and y can be varied within the range of -1 to $+1$. The use of an orbit to sample a two-variable function $f(x, y)$ demonstrates their mutual dependency and bearing on a resultant waveform. Especially the function $f(x, y)$, as it establishes the general properties of the waveform.

Suitable two-variable functions within the range of $-1 \leq x \leq 1$ and $-1 \leq y \leq 1$ are described by Mitsuhashi as having the following properties [48]:

- Both the function and its first-order partial derivatives are continuous in the area of definition.

- The function is zero on the boundaries of definition $f(\pm 1, x) = 0$ and $f(\pm 1, y) = 0$.



(a) $f(x,y) = sin(x\pi) * sin(y\pi)$

(b) $f(x,y) = cos\left(\frac{sin(4xy)}{x^2y^2}\right)$

Figure 2.2: Terrain examples generated in Matlab

- The first-order partial derivatives $\partial f/\partial x$ and $\partial f/\partial y$ are zero on the opposite boundaries $x = \pm 1$ and $y = \pm 1$, that is $\partial f(x,y)/\partial x = 0$ and $\partial f(x,y)/\partial y = 0$.

Amongst the limitless possibilities for orbit signals, Mitsuhashi considers a sampling orbit capable of producing a wide range of time-varying curves (including both periodic and aperiodic structures) for evaluating wave terrain synthesis [50]. The orbit is given by

$$\left.\begin{aligned} x &= 2f_xt + \Phi_x + I_x(t)sin(2\pi F_xt + \Psi_x) \\ y &= 2f_yt + \Phi_y + I_y(t)sin(2\pi F_yt + \Psi_y) \end{aligned}\right\} \tag{2.6}$$

where f_x, f_y, F_x, F_y are frequencies, $\psi_x, \psi_y, \phi_x, \phi_y$ are initial phase values, and $I_x(t), I_y(t)$ are time dependant orbital parameters. Evaluation of the system reveals some important and distinctive properties of wave terrain synthesis.

For instance, if an orbit is periodic then the resulting waveform will be periodic too, and if an orbit is fixed then the resulting sound is a fixed waveform characterised by a static spectrum. To generate a time-varying waveform, the orbit has to be changed over time. Although Roads [1] envisioned an extension to this where the orbit is fixed and the wave terrain is time-varying, much like the tracing of waves on the surface of the sea (see section 3.3.2). Naturally, this idea has served as an inspiration for scanned synthesis. Sonic results from wave terrain synthesis have been compared to pitched sounds reminiscent of analogue synthesisers, frequency modulation [50], amplitude modulation with greater dynamism [51], dynamic waveshaping synthesis, and more. The sonic characteristics of wave terrain synthesis are of course highly dependent on the orbit and terrain, but interaction and complex mapping within the system can also serve to drastically transform these characteristics. In truth, the multifaceted nature of wave terrain synthesis makes it somewhat ambiguous and difficult to categorise - it may be only ever be best described as a mixture of elements drawn from many types of synthesis. This is very much both its strength and weakness, and what makes it difficult to implement yet potentially rewarding if done so successfully.

2.4.2 Previous Implementations

Wave terrain synthesis is a fairly unknown sound synthesis technique and has only been investigated by a few researchers in the field. As mentioned previously, the concept and initial research was conducted by Gold [46], [47]. Formal implementations and analysis of the synthesis were subsequently carried out by Mitsuhashi, and Borgonovo and Haus [49]. Early research mainly focused on simple orbit and terrain functions, often using polynomials and trigonometric functions, to be able to study the technique systematically. Di Scipio investigates the use of iterated nonlinear sine map models [10], [11], [12], and Cafagna explores the use of Elliptic functions for sound synthesis [52].

Research has also been done in the development of interfaces for wave terrain synthesis by Hsu [53], James [54], and Overholt, whose MATRIX controller is capable of dynamically generating and distorting the frequency of two-dimensional maps [55]. Harons development of higher dimensional wavetables can also be relevant for generation of terrains [56]. More recent implementations of wave terrain synthesis have also been developed by James who proposes a visual methodology [50] and uses wave terrains for exploration of timbre spatialisation [57].

Chapter 3

Design and Implementation

3.1 Hardware, Platforms and Code Libraries

3.1.1 Max/MSP and OpenGL

Max/MSP [58] is a modular real-time graphical programming environment for the development of music and multimedia applications. Although written in C, it aims to take care of low-level programming ‘under the hood’ to provide users with a high-level and intuitive environment for rapid prototyping. The environment can be split into two main domains - MSP for audio signal processing, and Jitter, which utilises the OpenGL protocol for real-time graphic rendering and video processing.

Max/MSP was chosen to develop this application for several reasons. Firstly, the vast range of efficient objects available provided ease of use whilst retaining a high level of programming flexibility. This provides an ideal platform for prototyping experimental sound/processing methods without having to consider low-level complexities. At the time of writing, Max/MSP is also one of the few frameworks to support the MPE, which it does by incorporating objects made specifically for handling MPE data streams, allowing for easy and efficient integration with MPE instruments. As well as having great integration between audio and graphic processing, Max/MSP also integrates seamlessly with Ableton Live (a popular music production and live performance software) which makes the application simple to distribute to a wider audience.

Perhaps the most important feature of Max/MSP for the implementation of this project were Gen Objects. These were used extensively as they utilise their own scripting language called *GenExpr*, helping merge the advantages of a graphical environment with that of a low level scripting language. Essentially, Gen generates and compiles native CPU machine code, and allows for custom pre-compiled MSP and Jitter objects to be created within Max itself. Gen patchers are specialised for both MSP (`gen~`) and Jitter (`jit.gen`) with specific functionality dependent on their domain, such as delay lines in `gen~` and vector operations in `jit.gen`.

3.1.2 Hardware

In the digital age, many new instruments and interfaces are often proposed, yet comparatively few become established and widely accepted [59]. At the time of writing, the ROLI Seaboard is one of those few to have gained significant commercial attraction and approval from the music community, and is perhaps the most well known of the MPE instruments. The success of the instrument is owed partly to its piano keyboard interface, which whilst unique, provides users with a familiar and intuitive instrument. The interface is made of soft silicone and is designed as one continuous, non-flat surface, where the raised and recessed areas of the surface correspond with the centres of white and black keys [59]. The surface rests upon an array of force sensing resistor (FSR) sensors which measure variations in pressure and location of pressure peaks in the sensor array, thereby forming a representation of which notes the user is playing, and how they are playing them.

The ROLI Seaboard was chosen as a suitable hardware interface for testing the wave terrain synthesis application for two main reasons. Firstly, the familiarity of its interfaces provides an ideal platform for evaluating MPE and wave terrain synthesis as it allows users to explore the new capabilities of the system without having to adapt to an entirely new interface. This also improves the potential longevity of the application due to its compatibility with these types of interfaces. Secondly, the way in which the Seaboard is designed as one surface is advantageous to controlling wave terrain synthesis, as one is able to continuously traverse a terrain through intuitive gestural control, whilst maintaining the musical possibilities of a keyboard interface. Finding an effective means of doing this has been a major drawback of WTS systems, but MPE seems to offer a possible solution.

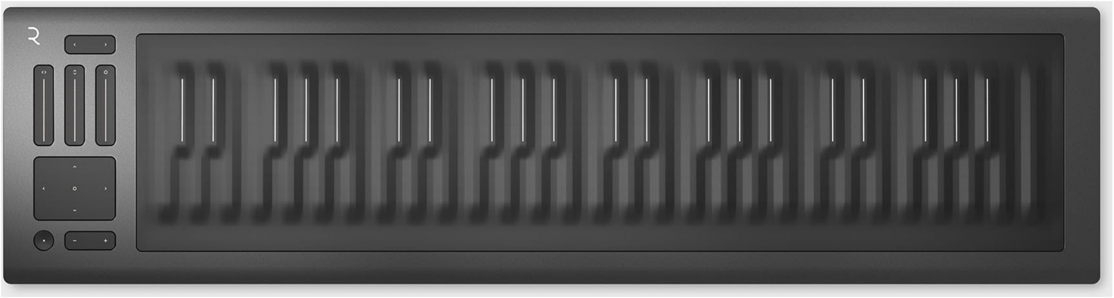


Figure 3.1: The ROLI Seaboard RISE

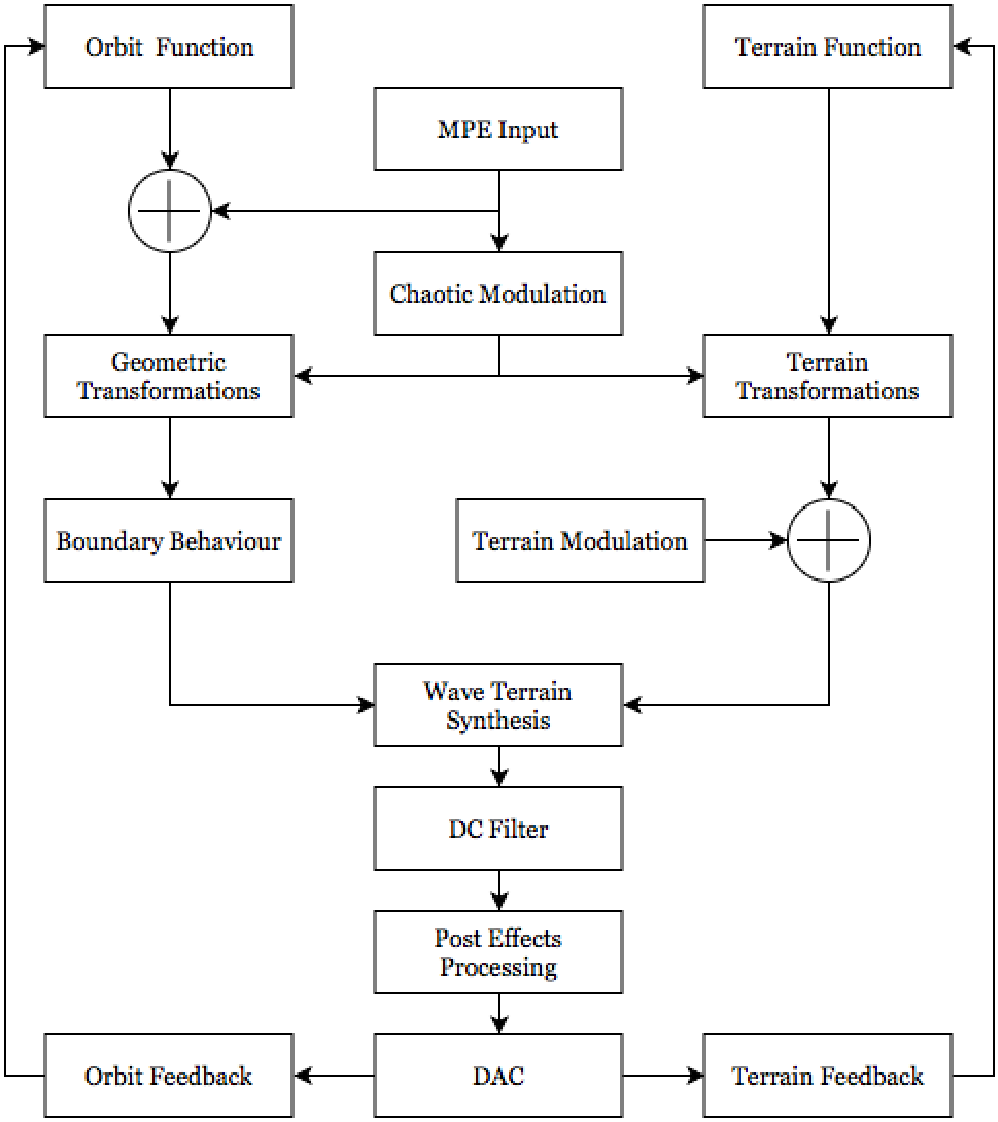


Figure 3.2: Instrument Schematic

3.2 System Overview

The entirety of the system was built in Max/MSP with no external library dependencies. The system is split into two independent sections, one of which handles the audio (MSP) whilst the other renders graphics (Jitter). The sections are then linked together through accurate one-to-one mappings between their parameters. The reason for this approach was to minimise run-time dependencies on Jitter so as to reduce audio latency and potential system malfunctions.

Similarly, orbit and terrain functions were also implemented independently, as dictated by wave terrain synthesis theory. The terrain functions are generated inside a `jit.gen` object that outputs coordinate data into a `jit.matrix` which is graphically rendered in real-time with `jit.gl.mesh`. An orbit however, relies on separate terrain and orbit `gen~` objects for MSP, and a separate `jit.gen` for Jitter, which is connected to the same `jit.mesh` as the terrain for graphic rendering. With six-voice polyphony, the system relies on a total of 19 `gen` objects for a great deal of the processing. This approach allows for each orbit to be entirely independent in terms of both audio generation and graphics, which when combined with external parametric control of all `gen` objects, creates an immensely flexible platform for mapping control of the system. Interaction with the system relies on the Seaboard, which communicates with Max via serial. Data from the Seaboard is collected with the `mpetrack` object, which outputs data based on the touch number detected. By using separate `mpetrack` objects, MPE data can be routed and unpacked individually for each 'touch' of the Seaboard, which is used to implement polyphony and to control parameters within the MSP and Jitter sections. An overview of the system is visualised in figure 3.2.

3.3 Terrain Function Generation

An arithmetic approach to terrain generation was selected over other possible methods (described in section 2.4.2). Whilst not the most computationally efficient method, this approach was chosen due to its flexibility, ability to experiment with more advanced mathematical constructs (such as iterative maps), and for providing infinitesimally accurate solutions to any function $f(x, y)$ [50].

3.3.1 Functions of Two-Variables

As mentioned previously, terrain functions are generated in `gen` objects. In the MSP section, the x and y orbit signals are sent to the two inlets of a `gen~`. The two signals are then simply multiplied by the variables `xTerrainIterations` and `yTerrainIterations`, the results of which are used to generate the function. A code example of this process can be seen in figure 3.5. Graphical generation of terrains are implemented in Jitter in a similar way. A three-dimensional matrix is initialised and filled with values generated from a `jit.gen` object. The matrix is then rendered with OpenGL using `jit.gl.mesh`.

A total of ten terrain functions were implemented, ranging from simple geometric functions to complex ones. Three main control functions are provided for manipulation of the terrain: `xTerrainIterations`, `yTerrainIterations`, and `terrainScale`. These three parameters alone offer significant control over the geometry of the terrain. Generally, an increase in terrain iterations creates a more complex terrain, and subsequently more harmonically rich waveforms as the orbit travels over it. ‘Noisy’ waveforms can be generated with enough iterations, although the amount of iterations required is dependent on the nature of the terrain function. Scaling of the terrain correlates directly with amplitude, so a completely flat terrain would essentially be a non-driven system.

```
terrain9(xInput, yInput, xTerrainIterations, yTerrainIterations,
    terrainScale) {
        x = xTerrainIterations*xInput;
        y = yTerrainIterations*yInput;
        z = terrainScale*(sin(22*(pow(x,2)) + pow(y,3)));
        return z;
}
```

Figure 3.3: Terrain function in `gen~` based on: $f(x,y) = sin(22x^2 + y^3)$

3.3.2 Dynamic Surface Modulation

Dynamic modulation of the terrain is inspired by scanned synthesis (see section 2.1.5) and Curtis Road’s envision of a fixed orbit over a time varying terrain [1]. The overall flexibility of the system allows for this to be implemented simply by creating time-based variations of `xTerrainIterations` and `yTerrainIterations`. By altering terrain iteration parameters using a cyclical process, a wave-like modulation of the terrain can be achieved. Depending on the way in which it is mapped, the effect of a fixed orbit over a modulated terrain can be perceptually equivalent to parallel dynamic waveshaping and amplitude/frequency modulation. The timbral quality of the effect is dependent on the terrain and orbit structures, and the rate at which the terrain is modulated. The performer is also able to gesturally move the orbit whilst the terrain is modulating, and also has independent control over the speed of modulation. Overall, this results in a rather unique interactive and time-varying audio processing effect.

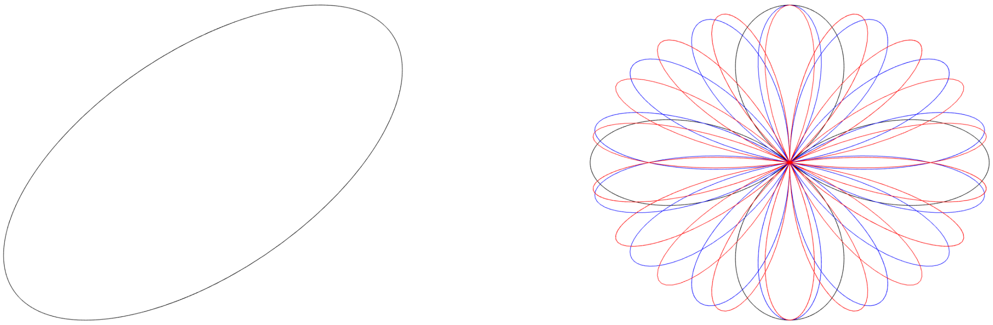
3.4 Orbit Function Generation

The orbit functions are based on parametric equations as they lend themselves well to geometric transformations and wave terrain synthesis due to being based on separate x and y equations [60]. The equations are driven by either a sine, sawtooth or triangle oscillator which the performer has the option to choose from. Prior to being fed as inputs into the MSP section of the terrain generator, the equations undergo transformations such as signal folding and movement in the x and y position over the terrain. The structure of an orbit is crucial in determining the final audio output. The frequency or pitch of the instrument is completely dependent on the orbit, and the size and position of the orbit have a significant impact on the timbral and harmonic content of the final audio output.

3.4.1 Periodic Orbits

There are a wide variety of orbit structures that can be created, characteristics of which can range from periodic, quasi-periodic, chaotic or stochastic. To keep in line with creating a 'musical' instrument, it was decided for only periodic orbits to be implemented. That is, orbits with a fixed path geometry [50]. Despite this, orbits can exhibit quasi-periodic and chaotic characteristics due to a chaotic modulation parameter that can be controlled both graphically and with gestural interaction (see section 3.5).

There are ten orbit functions available to users, all of which offer a great variety in their various shapes and structures, thus affecting the resultant waveforms in different ways. Many aspects of the orbits are designed to be controlled gesturally using the Seaboard, details of which are discussed in section 3.6.1. Nonetheless, some orbit parameters are available for adjustment using the graphical user interface. An option between `wrap` or `fold` for instance, determines the orbit's behaviour on the boundaries of the terrain. The `waveforms` selector allows the user to switch between different oscillators, and `orbitSizeLimit` restricts the maximum size of the orbit, which is dependent on the force of touch measured by the Seaboard.



(a) $x = sin\left(2\pi t + \frac{\pi}{5}\right), y = sin(2\pi t)$ (b) $x = cos(4t)cos(t), y = cos(4t)sin(t)$

Figure 3.4: Orbit examples generated in Matlab

```
}


orbit45(coord, xTerrainIterations, yTerrainIterations, terrainScale,
xOrbitIterations1, yOrbitIterations1, xOrbitPosition1, yOrbitPosition1,
orbitSize1, orbitMode, orbitRotation)
{
        swiz = swiz(coord, 0);
        xT = xTerrainIterations;
        yT = yTerrainIterations;

        r = cos(4*(xOrbitIterations1*swiz));

        if(orbitMode == 0) {
        xp = fold(orbitSize1*r*((sin(xOrbitIterations1*swiz))) +
        xOrbitPosition1,-1);
        yp = fold(orbitSize1*r*((cos(yOrbitIterations1*swiz))) +
        yOrbitPosition1,-1);
        x = swiz(xp, 0);
        y = swiz(yp, 0);

        z = terrainScale*(((8*(pow((xT*x),4))) - (8*pow((xT*x),2)) + 1) *
        ((8*(pow((yT*y),4))) - (8*pow((yT*y),2)) + 1));
        return vec(x, y, z);
        }

        if(orbitMode == 1) {
        xp = wrap(orbitSize1*r*((sin(xOrbitIterations1*swiz))) +
        xOrbitPosition1,-1);
        yp = wrap(orbitSize1*r*((cos(yOrbitIterations1*swiz))) +
        yOrbitPosition1,-1);
        x = swiz(xp, 0);
        y = swiz(yp, 0);

        z = terrainScale*(((8*(pow((xT*x),4))) - (8*pow((xT*x),2)) + 1) *
        ((8*(pow((yT*y),4))) - (8*pow((yT*y),2)) + 1));
        return vec(x, y, z);
        }
}
```

Figure 3.5: `jit.gen` function for the 4th orbit sampling over the 5th terrain

3.4.2 Temporal Evolutions

It has been demonstrated that the spectra of interesting timbres change with respect to time [14]. With that in mind, the evolution of an orbit over time should be considered an important aspect for reasons of sound synthesis interaction and musical expression. One of the main advantages of using the Seaboard for wave terrain synthesis is that it provides a surface which can essentially be used as a physical surface of the virtual terrain. This means that a performer could use gestures to evolve an orbit over a terrain as a function of time. This approach allows for the user to not only have the musical interaction benefits of a keyboard interface, but to also be able to have significant control over the timbre at the same time. This could of course be implemented in many different ways, and is dependent on the nature of the instrument, but MPE does offer potential freedom in controlling the temporal evolution of orbits, and consequently in the sounds one can produce using wave terrain synthesis.

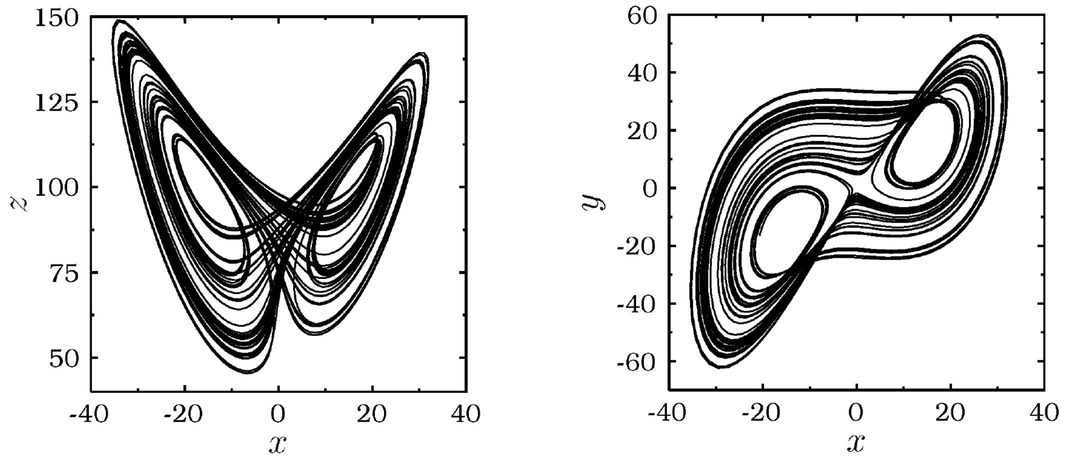
3.5 Chaotic Modulation

The inclusion of chaotic modulation was inspired by nonlinear interactions in acoustic instruments as described in section 2.2.4. Its aim is to add minor chaotic fluctuations to the performance of the instrument. To achieve this, the Lorenz attractor was chosen arbitrarily (different attractors would be indistinguishable in the context of this application) and implemented in gen$\sim$.

Two modes of chaotic modulation have been implemented - fixed and adaptive. The user is able to choose between modulation of the orbit position or its orbit size.

Fixed modulation simply uses the x, y outputs of the Lorenz attractor to modulate the selected parameters based on their current value. This ensures that the modulation is relatively controlled and only occurs within a set range. The advantage of this approach is that it can help maintain the 'musicality' of the synthesis by allowing modulation to influence the system without overpowering it. The sensitivity of the attractor and the range of modulation are control parameters provided to the user. An example effect that can be produced using this method is 'chaotic vibrato', which works by setting the modulation to orbit position with a relatively low range of modulation and sensitivity.

Adaptive chaotic modulation aims provide performers of the instrument with gestural control of its parameters. It does this by measuring average fluctuations of x and y movements or velocity from the Seaboard, and using these values to control modulation parameters. This creates a direct link between performer and



(a) label 1 (b) label 2

Figure 3.6: Phase portraits of a chaotic attractor from the Lorenz equations [17] - used to drive chaotic modulation of the orbit or terrain.

the modulation - the more 'chaotic' the movement performance, the more chaotic the resulting acoustic output will be. Alternatively, a chaotic attractor could have been used directly as an orbit. However, in sight of keeping the system inherently musical, it was decided for it to be best utilised purely as modulation for periodic orbits.

3.6 Instrument Design

Before discussing instrument design decisions, it is important to identify the creative aspirations and criteria for the instrument as a whole.

- The instrument should provide a platform for expressive musical intentions.
- The instrument should leverage the capabilities and advantages of MPE for use with wave terrain synthesis and provide unique gestural control opportunities.
- The instrument should not lose its 'musicality' in offering complex interaction and gestural manipulation of the system.
- The instrument should be intuitive to use yet require practice to learn proficiently, and should provide a clear and concise graphical user interface.
- Gestural interaction with wave terrain synthesis should take inspiration from inherent complexities in acoustic instruments.

The points mentioned above were identified as important criteria in designing a 'successful' wave terrain synthesis instrument. This section will discuss design decisions that were made to achieve this.

3.6.1 Mapping Strategy

The parameter mapping strategy is perhaps the most important design decision and is fundamental to interaction and feedback from the instrument (see section 2.3.2). The implemented mapping scheme is inspired by the visual nature of wave terrain synthesis and the gestural control parameters available from MPE.

Achieving acoustic complexities through interaction with sound synthesis often requires complex mapping strategies, however in wave terrain synthesis many of these complexities are inherent within the synthesis technique itself, and so mappings can be relatively more straightforward whilst achieving similar results as divergent or convergent mappings. For example, as can be seen in figure 3.8, x and y positioning of the orbit are mapped to their direct counterparts - `pitch bend` and `slide value`. Although they are implemented as one-to-one mappings, varying these parameters offers complex interaction with the acoustic feedback and outgoing waveform. This includes effects such as frequency modulation, amplitude modulation, waveshaping and distortion of the waveform.

Despite this, significant design decisions had to be made for the mapping of MPE to synthesis parameters. Perhaps the most important of these was the way in which the orbit would be gesturally controlled by the user whilst continuing to maintain musicality (based on a discrete 12-note scale). Other important considerations related to mapping included sensitivity of parameters and nonlinear scaling of input.

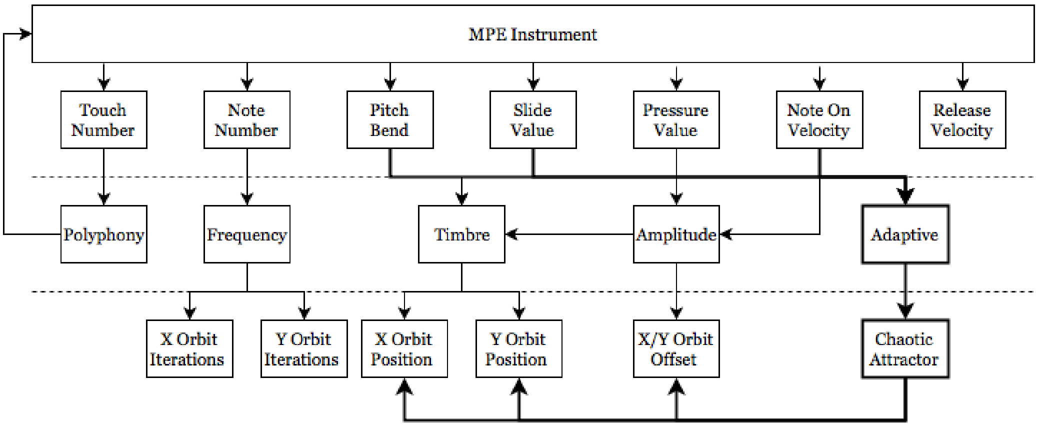


Figure 3.7: Mapping Scheme: input parameters of the ROLI Seaboard -> perceptual descriptors -> orbit control parameters

3.6.2 Design Decisions

During the implementation process a variety of different design decisions had to be made in regards to the inner workings of the system. This section will describe a few of these decisions.

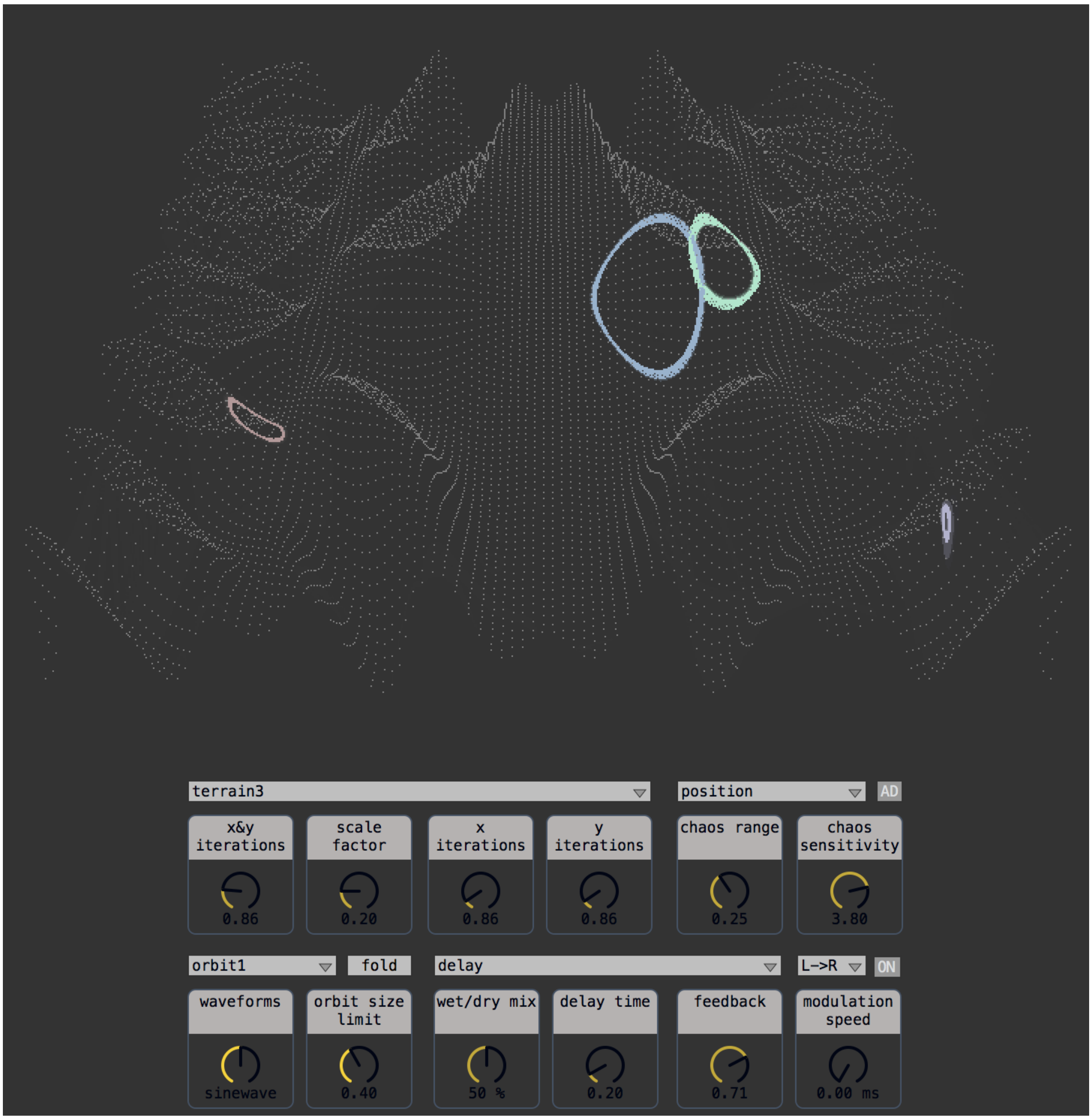
An important consideration for the instrument was how to maintain musical control whilst also being able to manipulate parameters of the synthesis itself. Initially, the frequency of the orbit was set by an external parameter, allowing for the performer to focus on navigating the terrain primarily for sculpting the timbre of the sound. Whilst this would be useful for sound design purposes, a more musical approach was desired. As a result, the final implementation uses the note number parameter of MPE to control the frequency, an approach similar to standard MIDI controllers. The difference in this MPE implementation is that it uses deviations from the first selected note in the form of pitch bend and slide value to traverse the terrain. This approach allows for complex, yet musical manipulation of timbre.

Using MPE in this way raised the question of what the relationship between gestural input and position of the orbit was. Through experimentation, the most effective and intuitive way to treat gestural input with the Seaboard was identified as treating the Seaboard like the surface of the terrain itself - that is, that the Seaboard boundaries should coincide with the boundaries of the terrain. This approach however, also meant that the position of an orbit would be defined by the note being played, which would limit the flexibility of the instrument. To solve this, each note generates a random position on the terrain, and control of the movement of the orbit is relevant to the initial position it was assigned to. This was somewhat inspired by the idea of aleatoric music, where a controlled element of randomness is used as part of performances.

Efficiency was an important consideration in creating an instrument that is usable for real-time interaction and performance. Therefore, careful consideration had to be made into the architecture of the code so as to maximise efficiency. After experimentation it was found that separating Jitter and MSP operations improved both efficiency and stability. Most likely because MSP processing tends to be less computationally demanding. They essentially work in parallel, and relevant parameters of both are mapped one-to-one to create accurate communication between them.

3.6.3 Graphical User Interface

The most important criteria for the graphical user interface was to be able to display a three-dimensional visualisation of wave terrain synthesis with real-time control and manipulation of orbits and terrains. A dark background is provided for this purpose with each orbit being distinguished by different colours. Most of the real-time manipulation is performed directly through the Seaboard, however some manual control parameters are available. The aim of the interface was to refrain from providing the user with a myriad of ambiguous controls, and instead opt for a high-level and clear format. This results in a more streamlined and intuitive graphical interface.



x&y iterations 0.86
scale factor 0.20
x iterations 0.86
y iterations 0.86
chaos range 0.25
chaos sensitivity 3.80
waveforms sinewave
orbit size limit 0.40
wet/dry mix 50 %
delay time 0.20
feedback 0.71
modulation speed 0.00 ms

Figure 3.8: Graphical user interface

3.7 System Observations

This section will provide details of the system and abnormalities or malfunctions within the system and discuss the reasons and potential solutions for them.

With a sampling rate of 44100Hz and buffer size of 512, the CPU usage of the instrument at run-time ranges from 2-7% and includes 14 signals and 110 function calls. The relative efficiency of the system (when taking into consideration all aspects of the instrument), is largely down to the optimisation of Max/MSP and the system architecture of the implemented code. As supported by user evaluation (see chapter 4), there is no perceptual latency when using the system, aside from minor visual lag due to intensive graphical rendering. This would likely not be observed in modern computing systems.

There are two main issues with the implementation that affect the user experience of the instrument. The first is a fast fluctuating random movement of the orbit. This only occurs in a very specific case: if the seaboard detects two or more touches, one touch is removed (whilst the others remain), and put back on, the orbit of that particular function jumps around the terrain. The reason for this seems to be due to the structure of the `mpetrack` object. When the first touch is detected again, the 'touch number' parameter from the `mpetrack` does not update. This means that the system can not update the functions related to the touch number in this scenario, and it produces glitching.

The second issue is purely an aesthetic one based on the graphical rendering of orbits. Due to the way Jitter is processed, it stores and visualises each iteration of an orbit. This is generally a useful feature, however when a terrain is not flat it causes all iterations of the orbit to follow it along the z axis, which can cause a visual distortion of the original orbit shape.

Chapter 4

Analysis and Evaluation

This section will describe the user testing process, evaluation and analysis of the system. The user testing utilises two frameworks to provide a mixture of quantitative and qualitative results. The first is based on a framework for the evaluation of digital musical instruments [61] with focus on the performer's/composer's perspective. The second utilises an extended version of the well established system usability scale [62] to provide a reliable measurement of the usability of the system as a whole. To further analyse the system, an objective evaluation of the sound synthesis based on [63] is presented by the author.

4.1 Evaluation of Digital Musical Instruments

The framework [61] is inspired by design and evaluation methods from the fields of human-computer interaction (HCI) and music computing, examples of which include [38] and [64]. It identifies four main stakeholders for digital musical instruments - the audience, performer/composer, designer and manufacturer. Each stakeholder is then evaluated in one or more terms of *enjoyment*, *playability*, *robustness*, and *achievement of design specifications*. At this stage of the development of the instrument, it was determined for the performer/composer to be the most notable and significant perspective to evaluate from. This perspective involves qualitative evaluation of *enjoyment*, and quantitative methods for *playability* and *robustness*.

4.1.1 Testing Procedure

A total of 12 participants were evaluated, with approximately 40 minute tests for each. Testing occurred in various locations using a 24-key ROLI Seaboard Block with a 2014 model Macbook Pro. All participants had a background in music technology and good technical understanding of audio synthesis. Despite this, only half of participants identified as musicians (with an average of 7.84 years experience), four were well versed in piano/keyboard, and none had previous experience with MPE instruments.

Prior to testing, participants were given a five minute explanation of wave terrain synthesis, the ROLI Seaboard/MPE, the mapping scheme, and an overview of the graphical user interface. After the introduction, participants were given 15-20 minutes to experiment with the instrument, with the author available for technical support if required. Once this was complete, participants were asked to complete a digital questionnaire requiring both qualitative and quantitative answers. As mentioned previously, the questions were based around the enjoyment, playability, and robustness from performance aspects of the instrument. The order in which the topics were presented varied between participants, and the questions were developed to contain a mixture of negative and positive wording so as to remove potential biases.

4.1.2 Quantitative Results

Quantitative results are based on a numerical scaling from 1-10, where 1 is to strongly disagree and 10 is to strongly agree. The first table evaluates the system from a playability standpoint, and the second is about robustness of the system.

Quantitative Evaluation: Playability	**Mean**
Graphical visualisation helped me understand the synthesis technique	9.25
Visualisation negatively influenced my performance and interaction	2.12
Mapping between the seaboard and synthesis was intuitive	8.96
Control parameters were overly sensitive to gestural input	1.83
The seaboard allowed me to be more expressive in my performance	8.88
The graphical interface was intuitive	8.21

Quantitative Evaluation: Robustness	**Mean**
Perceived latency did not affect the playability or my performance	8.34
Parameters did not work as they should have	3.92
There were no auditory or graphical glitches in the system	4.34
The audio quality did not maintain its standard	2.3

4.1.3 Qualitative Results

The following outlines statements that most participants seemed to agree upon about various aspects of the instrument.

Perceived sound quality and stability: Participants were impressed with quality of sound synthesis, rating it as of commercial quality. System was generally considered stable aside from a few parameters.

Interaction, mapping and MPE: The Seaboard/MPE interaction was very well received and many claimed for it to be an ideal controller for WTS. Gestural mapping was intuitive and flexible.

Explorabillity and controllability: Both aspects were well received, although some participants would have liked more control over the shape of the orbit and structure of the terrain.

Learnability: Opinions were very mixed, most likely due to diversity in musical experience. Synthesis was generally considered easy to learn, but MPE had a steep learning curve.

Graphical User Interface: The interface was deemed clear and simple to navigate with a suitable amount of control parameters available.

Chaotic modulation: Participants enjoyed a means for unpredictability and being able to choose its sensitivity. The adaptive chaos however, was not perceived as a particularly useful feature.

Other comments: Most participants relied heavily on the graphic visualisation and claimed they would not have understood WTS without it. The synthesis had a distinctive sonic character.

4.2 System Usability Test

An extension to the standard system usability scale is proposed in [62] to help further validate results. The extension is based on a seven-point adjective-anchored Likert scale appended as an eleventh question. This test was used primarily to support results from the digital musical instrument evaluation (see section 4.1).

Results are based on a five-point scale, where 1 is to strongly disagree and 5 is to strongly agree. They were interpreted using the SUS formula. The resulting score for the system is 78.3, which is favourable compared to the average score of 68. They were also asked the adjective based 11th question: *Overall I would rate the user-friendliness of this product as:* to which over 80% answered 'Excellent'.

System Usability Test (Extended)	**Score**
I think that I would like to use this product frequently	4.12
I found the product unnecessarily complex	1.55
I thought the product was easy to use	3.89
I think that I would need the support of a technical person to be able to use this product	1.84
I found the various functions in the product were well integrated	4.31
I thought there was too much inconsistency in this product	2.07
I imagine that most people would learn to use this product very quickly	4.02
I found the product very awkward to use	1.33
I felt very confident in using the product	3.65
I needed to learn a lot of things before I could get going with this product	1.88

4.3 System Analysis

This section will follow relevant sections of the paper [63] for self-evaluating wave terrain synthesis and its implementation in terms of usability of parameters and efficiency of the technique. The intention of this section is to evaluate the system as a whole and to identify parts of the system that are suitable or unsuitable for a specific purpose.

How intuitive are the parameters? The intuition of parameters is often related to the mapping of musical attributes of the synthesis. For example, do controls relate to musical dynamics and articulation or are they mere mathematical variables with little correlation to real-world perceptual or musical experience [63]? Despite wave terrain synthesis being based on an abstract form of synthesis, the correlation between interaction, visual feedback, and acoustic output makes it relatively easy to understand and intuitive to perform with. This is supported by the evaluation of participants, nearly all of who shared this view. As one would expect however, the answer to this question remains highly dependent on the instrument design.

How perceptible are parameter changes? This question is related to the sensitivity and mapping of parameters. If not treated carefully, divergent and convergent mappings can make it potentially difficult for a user to control and perform with the instrument. It is therefore important to ensure stability in the system. Overly sensitive parameters can also cause problems by influencing the system in a way not intended by the performer. This aspect was considered during implementation of the instrument, and user evaluations show satisfaction with sensitivity and mappings of parameters.

How physical are the parameters? A physical parameter is one that mimics the complex acoustic behaviours of real-world instruments and applies it to digital sound synthesis. Abstract synthesis methods are limited to using this as inspiration, since a true physical parameter would have to rely on physical modelling synthesis techniques. The WTS implementation does however benefit from the capabilities of MPE for crudely replicating complex behaviours. For example, the orbit size is mapped to the pressure exerted onto the Seaboard, which relates perceptually to both amplitude and spectral complexity of the output waveform (dependent on orbit and terrain characteristics). This correlates roughly to the relationship of force and spectral output in real-world instruments. Participants felt that the instrument was more 'physical' with MPE interaction in comparison to standard digital keyboard interfaces.

How well behaved are the parameters? This question is linked to the predictability and sensitivity of control parameters, and overall stability of the system. Despite mostly positive remarks by participants on the characteristics of control parameters, some did observe instability in certain parameters for specific situations. However, these issues were mostly minor and did not have a negative affect on the overall impression of the instrument. Technical details of these issues can be found in section 3.7.

How robust is the sound's identity? The identity of a sound is concerned with how well it retains its identity in the context of variation - it is the expression of synthesis [63]. For a synthesis technique to have an identity, the sounds it can produce should be perceptually related so that they are distinctive enough to be recognised as coming from the same instrument. That is not to say that the instrument should not be capable of producing a variety of sounds and timbres, but that the sounds should exhibit common sonic characteristics. The importance of sonic identity can be demonstrated by well known techniques such as subtractive or FM synthesis, and even specific synthesisers - which are all recognisable and famed for their sonic qualities.

Wave terrain synthesis, which can be thought of perceptually as a mixture of different synthesis techniques, could be expected to struggle with identity. This is not the case however. The strength of wave terrain synthesis lies in the level of synthesis interaction it offers. Despite being capable of producing a multitude of possible spectrums, the sound of an orbit being moved around the steep peak of a terrain is immediately recognisable.

How efficient is the algorithm? Efficiency is an important consideration in real-time sound synthesis, and is dependent on a number of different factors such as memory and processing power. A digital musical instrument will usually also include hardware, which should also be considered. Generally, the time delay between input and output for a real-time musical system should not be over 30ms, or else it will be perceptible and have a negative affect on performance aspects of the system. Details on the efficiency of the wave terrain instrument can be found in section 3.7.

Chapter 5

Conclusion and Further Work

Musical expressivity is a difficult subject to fully understand, yet alone to capture and facilitate. However, recent developments in digital interfaces are becoming increasingly more adept at capturing the subtleties of human touch and subsequently, expression. This more detailed stream of information requires new digital synthesis methods capable of leveraging the advantages it has to offer. Wave terrain synthesis was identified as a suitable candidate for this purpose due to its multifaceted and visual nature. To test this hypothesis, a digital wave terrain synthesis instrument was built, extended on, and linked harmoniously with an MPE instrument - the ROLI Seaboard.

The overall conclusion is that wave terrain synthesis is inherently well suited as a synthesis technique for MPE systems, and that their relationship is two-fold. Not only does the level of control provided by MPE suit the highly parametric nature of the synthesis - but the visualisation of WTS actually helps a performer understand their interaction with the MPE interface and exactly how their gestures are manipulating the sound synthesis algorithm. This is an interesting situation as both interface and synthesis help improve each others shortcomings. User evaluation of the instrument supports these ideas.

In summary, the instrument works efficiently without major flaws, allows for intuitive control of sound synthesis parameters that encourage musical expressivity, and extends the wave terrain synthesis model in several ways - it was deemed a success.

5.1 Summary of Contributions

A summary of results based on the research aims of the project (section 1.2).

1. A real-time polyphonic wave terrain synthesis instrument was implemented in Max/MSP and designed to be controlled with MPE - namely, the ROLI Seaboard. The interface proved to be an ideal control mechanism, and solved many of the previous difficulties in providing effective control of WTS parameters.
2. The WTS algorithm was extended with dynamic modulation techniques. Most notably - dynamic surface modulation, which was evaluated to be effective and musically valuable. Chaotic modulation received a similar evaluation, whereas adaptive modulation was deemed redundant.
3. Nonlinear dynamical systems have been previously proposed as a suitable means of controlling or evolving elements of WTS. Effective mapping between MPE and WTS parameters solved most of the motives behind this reasoning. Nonetheless, principles from the field were used as inspiration for modulation parameters with relative success.
4. Wave terrain synthesis was evaluated through user testing and conclusions were made as to its strengths, weaknesses, relationship with MPE, and future possibilities.

5.2 Further work

Due to the multifaceted nature of wave terrain synthesis, there are countless ways in which it could be designed, implemented, and improved. During the implementation of this project there were a few future additions that were identified as potentially useful. Firstly, a 'rotation' control could be created for orbits, which would cyclically rotate the orbit at a speed set by the user. This could serve as an interesting audio effect (particularly at terrain boundaries), although its characteristics would depend on the terrain and position it is sampling from. The orbit could also benefit from a graphical interface where the shape of an orbit can be manually drawn or where there is significant parametric control over its shape. A similar concept could be applied to terrains, where the terrain structure is generated through an imported image file. Alternatively, deeper control of terrain structures could be offered through additional control variables to mathematical functions or through extended control of wavetable shapes, depending on the implementation method. As terrains have a greater influence on the resulting waveform, it is suggested that efforts are better placed into terrain control systems.

This project has introduced control of wave terrain synthesis using the ROLI Seaboard, but MPE instruments can vary significantly in their structure and interfaces. It would be a useful to have a general framework for MPE control of WTS that can be applied to a variety of current and future MPE instruments. Perhaps the use of independent mapping layers would be a useful approach for this task. Alternatively, research could be made into creating MPE specific interfaces, although the use of commercial interfaces would perhaps make it more likely for WTS to be popularised.

Wave terrain synthesis could also find use in non-musical settings. As suggested by Wegner [65], WTS could be used for audio rendering of 3D surfaces, essentially providing real-time acoustic descriptions and sonifications of the surfaces. This could be useful as an audio feedback system for medical surgery or other unexplored areas.

It is clear that the potential of wave terrain synthesis has yet to be fully explored, but has outstanding potential as a sound synthesis technique. Yet for this potential to be fully realised, more research, creative exploration, and publicity is required. The author is confident that with the efforts of a few, wave terrain synthesis will be enjoyed by many.

Bibliography

[1] C. Roads, *The Computer Music Tutorial.* The MIT Press, 1996.

[2] P. Cook, *Real Sound Synthesis for Interactive Applications.* CRC Press, 2003.

[3] R. H. Tubb, "The wablet: An investigation into scanned synthesis and multitouch," Master's thesis, Queen Mary University London, 2011.

[4] T. Stilson and J. Smith, "Alias-free digital synthesis of classic analog waveforms," n.d.

[5] J. Smith, *Physical Audio Signal Processing for Virtual Musical Instruments and Audio Effects.* W3K Publishing, 2010.

[6] G. Geiger, "Table lookup oscillators using generic integrated wavetables." Proceedings of the 9th Conference on Digital Audio Effects (DAFx06), 2006.

[7] A. Franck and V. Vlimki, "Higher-order integrated wavetable synthesis." Proceedings of the 15th Conference on Digital Audio Effects (DAFx12), 2012.

[8] R. Bristow-Johnson, "Wavetable synthesis 101, a fundamental perspective," in *Proc. 101th AES Convention*, 1996.

[9] A. Horner, J. Beauchamp, and L. Haken, "Methods for multiple wavetable synthesis of musical instrument tones." Audio Engineering Society, 1993, vol. 41, pp. 336–356.

[10] A. D. Scipio, "Synthesis of environmental sound textures by iterated nonlinear functions." Proceedings of the 2nd Conference on Digital Audio Effects (DAFx99), 1999.

[11] ——, "Iterated nonlinear functions as a sound-generating engine," 2001, vol. 34, no. 3, pp. 249–254.

[12] ——, "The synthesis of environmental sound textures by iterated nonlinear functions, and its ecological relevance to perceptual modeling," 2002, vol. 31, no. 2, pp. 109–117.

[13] B. Verplank, M. Mathews, and R. Shaw, "Scanned synthesis." Interval Research Corporation, 1999.

[14] J.-C. Risset, "Analysis of instrumental tones." Physics Today, vol. 2.

[15] R. Boulanger, P. Smaragdis, and J. Fitch, "Scanned synthesis: An introduction and demonstration of a new synthesis and signal processing technique." Proceedings of ICMC 2000, 2000.

[16] S. H. Strogatz, *Nonlinear Dynamics and Chaos with Applications to Physics, Biology, Chemistry, and Engineering.* Perseus Books, 1994.

[17] M. Lakshmanan and S. Rajasekar, *Nonlinear Dynamcs: Integrability, Chaos, and Patterns.* Springer-Verlag Berlin Heidelberg, 2003.

[18] E. Ott, T. Sauer, and J. A. Yorke, *Coping with Chaos: Analysis of Chaotic Data and the Exploitation of Chaotic Systems.* John Wiley Sons, Inc., 1994.

[19] T. Mudd, "Nonlinear dynamics in musical interactions," Ph.D. dissertation, Centre for Research in Computing, The Open University, 2017.

[20] R. Gilmore and M. LeFranc, *The Topology of Chaos: Alice in Stretch and Squeezeland*, 2nd ed. Wiley - VCH, 2011.

[21] R. Rui and C. Bao, "Musical instrument classification based on nonlinear recurrence analysis and supervised learning," 2013.

[22] S. J. Puig, "Digital instruments and players part ii: Diversity, freedom and control." International Computer Music Conference, 2004.

[23] D. Menzies, "Composing instrument control dynamics," 2004.

[24] J. Paradiso, "Electronic music: New ways to play," 1997, vol. 34, no. 12, pp. 18–30.

[25] S. J. Puig, "Digital lutherie: Crafting musical computers for new music performance and improvisation," Ph.D. dissertation, Universitat Pompeu Fabra, 2005.

[26] J. Malloch, “A framework and tools for mapping of digital musical instruments,” Ph.D. dissertation, McGill University, 2013.

[27] NIME, “Proceedings of the international conference on new interfaces for musical expression,” http://www.nime.org/archives/, accessed: 17/08/2018.

[28] A. Friberg and E. Schoonderwaldt, “Cuex: An algorithm for automatic extraction of expressive tone parameters in music performance from acousitc signals,” 2007, vol. 93, no. 3.

[29] A. Bhatara, A. K. Tirovolas, L. M. Duan, B. Levy, and D. J. Levitin, “Perception of emotional expression in musical performances,” 2011, vol. 37, no. 3.

[30] R. A. Kendall and E. C. Carterette, “The communication of musical expression,” 1990, vol. 8, no. 2.

[31] U. Zölzer, *DAFx: Digital audio Effects*, 2nd ed. Oxford, England: John Wiley Sons, Inc., 2011.

[32] L. Mion and G. D’Inc, “Analysis of expression in simple musical gestures to enhance audio in interfaces,” 2006, vol. 10, no. 1.

[33] C. Oshima, K. Nishimoto, Y. Miyagawa, and T. Shirosaki, “A concept to fascilitate musical expression.” Proceedings of the 4th Conference on Creativity and Cognition, 2002.

[34] J. Holfelt, G. Csapo, N. Andersson, S. Zabetian, M. Castanieto, D. Overholt, S. Dahl, and C. Erkut, “Extraction, mapping, and evaluation of expressive acoustic features for adaptive digital audio effects.” Proceedings of the 14th Sound and Music Computing Conference, 2017.

[35] A. Hunt, M. M. Wanderley, and M. Paradis, “The importance of parameter mapping in electronic instrument design.” Proceedings of Conference on New Interfaces for Musical Expression (NIME 2002), 2002.

[36] A. Hunt and R. Kirk, “Mapping strategies for musical performance.” IRCAM - Centre Pompidou, 2000.

[37] A. Hunt, M. M. Wanderley, and R. Kirk, “Towards a model for instrumental mapping in expert musical interaction.” International Computer Music Conference, Berlin, 2000.

[38] M. M. Wanderley, “Evaluation of input devices for musical expression: Borrowing tools from hci,” 2002, vol. 26, no. 3, pp. 62–76.

[39] M. M. Wanderley and P. Depalle, “Gestural control of sound synthesis,” 2004, vol. 92, no. 4, pp. 632–644.

[40] D. L. Wessel, “Timbre space as a musical control structure,” 1979, vol. 3, no. 2, pp. 45–52.

[41] M. Wanderley, N. Schnell, and J. Rovan, “Escher - modeling and performing composed instruments in real-time.” IEEE International Conference on Systems, Man, and Cybernetics, 1998.

[42] ROLI, “Seaboard: The future of the keyboard,” https://roli.com/products/seaboard, accessed: 30/07/2018.

[43] R. L. Design, “Linnstrument: An expressive replacement for the midi keyboard,” http://www.rogerlinndesign.com/linnstrument.html, accessed: 30/07/2018.

[44] K. M. Instruments, “K-board pro 4: Expressive smart instrument,” https://www.keithmcmillen.com/products/k-board-pro-4/, accessed: 30/07/2018.

[45] H. Audio, “Haken audio continuum fingerboard,” http://www.hakenaudio.com/Continuum/, accessed: 30/07/2018.

[46] J. Bischoff, R. Gold, and J. Horton, “Music for an interactive network of microcomputers,” 1978, vol. 2, no. 3, pp. 24–29.

[47] R. Gold, “A terrain reader.” The BYTE Book of Computer Music.

[48] L. Mion and G. D’Inc, “Audio signal synthesis by functions of two variables,” vol. 30, no. 10, pp. 701–706.

[49] A. Borgonovo and G. Haus, “Sound synthesis by means of two-variable functions: Experimental criteria and results,” 1986, vol. 10, no. 3.

[50] S. G. James, “Developing a flexible and expressive realtime polyphonic wave terrain synthesis instrument based on a visual and multidimensional methodology,” Master’s thesis, Edith Cowan University, 2005.

[51] R. Boulanger, *The Csound Book*. The MIT Press, 2000.

[52] V. Cafagna and D. Vicinanza, "Synthesis by mathematical models: Elliptic functions and lacunary series." Proceedings of the 7nd Conference on Digital Audio Effects (DAFx04), 2004.

[53] W. Hsu, "A flexible interface for wave terrain synthesis." PERformance and NETworking Colloquia, 2002.

[54] S. James, "A multi-point 2d interface: Audio-rate signals for controlling complex multi-parameteric sound synthesis." Proceedings of Conference on New Interfaces for Musical Expression (NIME 2016), 2016.

[55] D. Overholt, "The matrix: A new musical instrument for interactive performance." Proceedings of Conference on New Interfaces for Musical Expression (NIME 2001), 2001.

[56] A. Haron and M. Wright, "Wave voxel synthesis." Proceedings of the 12th Sound and Music Computing Conference, 2015.

[57] S. G. James, "Spectromorphology and spatiomorphology: Wave terrain synthesis as a framework for controlling timbre spatialisation in the frequency domain," Ph.D. dissertation, Edith Cowan University, 2015.

[58] Cycling74, https://cycling74.com/products/max/, accessed: 1/08/2018.

[59] R. Lamb and A. N. Robertson, "Seaboard: A new piano keyboard-related interface combining discrete and continuous control." Proceedings of Conference on New Interfaces for Musical Expression (NIME 2011), 2011.

[60] J. J. Shah and M. Mäntylä *Parametric and Feature-Based CAD/CAM: Concepts, Techniques, and Applications.* John Wiley Sons, Inc., 1995.

[61] S. O'Modhrain, "A framework for the evaluation of digital musical instruments," vol. 35, no. 1, pp. 28–42.

[62] A. Bangor, P. Kortum, and J. Miller, "Determining what individual sus scores mean: Adding an adjective rating scale," vol. 4, no. 3, pp. 114–123.

[63] D. A. Jaffe, "Ten criteria for evaluating synthesis techniques," 1995, vol. 19, no. 1, pp. 76–87.

[64] C. Fritz and D. Dubois, "On perceptual evaluation of musical instruments: The case of the violin put into perspective," 2014, vol. 100.

[65] K. Wegner, “Surgical navigation system and method using audio feedback.” International Community for Auditory Display, 1990.