UNIVERSITY OF
PLYMOUTH

AN INVESTIGATION INTO THE USES OF MACHINE LEARNING FOR ELECTRONIC SOUND SYNTHESIS

By

LAURENCE METTERS

A thesis submitted to the University of Plymouth in partial fulfilment for the degree of a

RESEARCH MASTERS

School of Humanities and Performing Arts

March 2021

Acknowledgements

Throughout the course of writing this thesis, I have received a great deal of support and assistance from those around me. I would first like to give credit to my Director of Studies and Supervisor, Professor Eduardo Miranda and Dr Alexis Kirke. The former has always been more than happy to present advice and direction when I struggled with the project, and the latter recommended the NSynth to me and helped shaped the project whenever I needed assistance for which I am extremely grateful.

I would also like to give thanks to Dr Edward Braund for recommending the ResM Computer Music to me in the first place, allowing me to continue to develop my skills as a researcher and academic in an area of study that I enjoy greatly. Sam Pearce too, who helped me better understand the construction of synthesizers and a great deal of python coding.

In addition, I would like to thank my family for their encouragement and support throughout my time at university, especially the last year and a half. You told me it would be worth the hard work in the end and I'm sure you were right.

Declaration

At no time during the registration for the degree of Research Masters has the author been registered for any other University award without prior agreement of the Doctoral College Quality Sub-Committee.

Work submitted for this research degree at the University of Plymouth has not formed part of any other degree either at the University of Plymouth or at another establishment.

Word count of main body of thesis: 23,274

Signed: Laurence Metters

Date: 20/03/2021

Abstract

This thesis presents an investigation into the uses of machine learning and artificial intelligence for electronic sound synthesis, specifically the creation of new synthesised sounds for composition and research. Using the Magenta Labs Neural Synthesizer (NSynth), a synthesis tool that uses deep neural networks to generate new sounds based on data input from electronic synthesizers, this research project aimed to produce a system where bespoke synthesizers could be used to produce interesting sound combinations consisting of approaches to electronic sound synthesis that would not conventionally be used together. Combinations of different approaches to electronic sound synthesis produced interesting results when choices were made based on sonic characteristics of individual synthesisers, such as the plucked timbres of a Karplus-Strong synthesizer and the smooth extended notes of a frequency modulation synthesizer.

In this thesis, contextual information into both electronic sound synthesis including justification for the use of each synthesis method as well as an investigation into Artificial Intelligence(AI) and machine learning techniques relevant to the use of the NSynth has also been carried out with the intention of producing an informed and researched final product in the form of a composition. The summary of this research project culminated in a final composition utilising the sound samples produced by the NSynth, arranged into a piece inspired by computer music research compositions including John Chowning's *'Stria'* and '*Switched on Bach'* by Wendy Carlos. The synthesizers in this research project were produced in Max and designed with specific sonic qualities of the instruments in mind, with versatility to produce more samples following the same process for further research and application by other composers inspired by the work. The resulting composition included a number of interesting sounds with plenty of variation in sonic qualities that resembled computer music composition as well as standard western composition, demonstrating the versatility of the concept of using AI and bespoke synthesisers to create new and interesting sounds.

Page of Contents

Introduction

With the prevalence of emerging technologies in the fields of Artificial Intelligence and electronic sound synthesis, it is no surprise how often these fields are encountered in daily life, examples ranging from prediction algorithms used by entertainment services to email spam filters. Artificial Intelligence, and specifically machine learning technologies are constantly being developed and improved to become even more efficient and practical to the everyday user. Similarly, electronic sound synthesis technologies have been utilised in everything from smart home assistants to pop music.

With the creation and prevalence of machine learning and sound synthesis technology, it is inevitable that artists and musicians would attempt to utilise this technology to push the limits and understanding to discover new ways in which the research can be used. These include ways of repurposing existing ways of musical and creative expression to produce new forms of art. This trend has been observed throughout human history, including early experimentation in times of conflict to produce the radio or musicians cutting apart speaker cones to produce the first distortion effects. The point is, wherever there is technological development in a field, artists and creators will seek to find ways in which this technology can be used for the advancement of the arts.

During the course of this thesis, I will present the steps I have taken towards producing a system where developments in machine learning technology and sound synthesis technology can be combined to produce new sounds which could potentially have a wide variety of uses, including composition, live performance, artistic expression and further research. This paper will also follow the processes I have undertaken to reach the final project, including developing an understanding of

machine learning and electronic sound synthesis coming from a non-computer music-based background, to becoming familiar with software such as Max and Python to help carry out the projects, and many more strands of research that all contributed towards the completion of this thesis.

The final sum of this master's degree project will be presented in the form of an in-depth investigation into the uses of machine learning for electronic sound synthesis, as well as a practical element which consists of an electronic instrument made up of new sounds as the result of combining machine learning algorithms with self-produced electronic synthesizers, carefully selected for the characteristics of the sounds they are capable of producing. In addition to this, I will also present surveys into the methods of sound synthesis and machine learning that informed my studies and allowed me to reach a definitive point in my research process.

I will begin this thesis with a survey of electronic sound synthesis methods with a discussion into the background of a few select methods of sound synthesis and why they have been chosen for this research project. This survey will also include a section of contextual information regarding the field of electronic sound synthesis. Finally, this chapter will go into depth about my own practices with electronic sound synthesis including discussion of my own attempts at creating synthesizers, my own practice with them and justification for their place within this research project.

Following this section, I will also conduct a similar survey into the fields of Artificial Intelligence and machine learning which is a sub-category of the former. This section will discuss relevant technological developments that allowed for the creation of the research practices I have used thus far, and historical and contextual information. This section will also discuss real world uses of

machine learning for music and a discussion of my own attempts at creating a basic perceptron model in order to better understand the field.

Next, this paper will include a reflection upon the path I took towards deciding my final project and the steps I took to get there. This includes the practical implementation of a number of electronic sound synthesisers and basic neural networks, and my research into TensorFlow, Magenta Labs, deep neural networks and the Neural Audio Synthesizer (Nsynth). This will include a critical reflection of the difficulties faced during the research phase of this project.

Finally, I will discuss the neural audio synthesis system that I have utilised for my project in depth as well as my own contributions to the system, as well as the rationale for the selection of samples I chose for the instrument and any steps that were taken to reach the conclusion of which samples were selected. I will also discuss any issues I encountered while learning to better utilise Max and Python in order to reach this point in my research, including problems faced during the testing phase and while developing the systems in place. As the final sum of this project also includes a composition inspired by the early computer music work of artists such as John Chowning, there will also be a section for discussion of the composition and the research that went into it.

1. Survey of Synthesis Methods

Electronic sound synthesis is a generic term for any form of sound that is produced electronically using a computer system or electronic system. Any sound that is produced through the use of electronic software and hardware can be classified as electronic sound synthesis, and the practice of using electronic machines to synthesise sound (Hass, 2017). A basic additive synthesizer functions by combining an oscillator with a soundwave produced by a noise generator to produce a basic noise which can be altered through the use of various filters, effects such as reverb and envelopes to modify a sound before it is propagated by an amplifier for use as a musical instrument (Smith, 2012). Historical context and relevant musical examples which helped me draw inspiration for the range of synthesizers and what sort of sounds can be achieved through electronic sound synthesis will be discussed in the following section.

1.1 Brief History of Electronic Sound Synthesis and Music

Some of the initial experimentation around the field of electronic sound synthesis included the French 'Musique Concrète' (Palombini, Carlos 1993) with Pierre Schaeffer which consisted of heavily modified recordings of musical instruments through effects such as tape manipulation, as well as the German c ounterpart, Elektronische Musik. The latter, pioneered by Karlheinz Stockhausen focused on the production and manipulation of electronically produced sounds from early noise generators rather than the manipulation of recorded samples, and has been described as 'pure' sound synthesis compared to the recording based 'Music Concrete' (Eimart, Herbert 1972). However, technology and experimentation in the field quickly disregarded the separation between the French and German research practices with pieces such as Stockhausen's *Gesang der Jünglinge* (Stockhausen, 1960)

combining both recorded sound and electronically synthesized sound to produce a unique and interesting piece of music.

The most popular image of a synthesizer to the average consumer of music is something controlled by a keyboard, perhaps recognisable from brand such as Yamaha as commercially available instruments (Milano D, 1975). From the 1960's, electronic sound synthesis was becoming more widely available thanks to the work of inventors such as Robert Moog who designed the envelope generators for the RCA Mark II (Peter, 1996), as well as a number of other modular instruments which were much more affordable than previous electronic synthesizers. With these cheaper analogue modular synthesizers, signals would be routed via patch cords and the instruments were brought into the mainstream attention with musical releases such as '*Switched-on Bach*' (Carlos, 1968). As music technology has developed over the years, electronic sound synthesis has become popular in a digital format. From the 1970's onwards as microprocessors and integrated circuits became inexpensive and cheaper to incorporate into technology, digital synthesizers became far more commonplace and synthesizers took the form of electronic keyboards, becoming easier to control than ever, such as the Yamaha DX7 (Yamaha, 1987).

While this development in the realm of physical synthesizers was taking place, research was also being carried out into the creation of coding languages that could use rapidly developing computers to control the frequency and other characteristics of sound, and to describe the sounds and music they wished to create synthesizers to produce. One such researcher was Max V. Mathews of Bell Labs, who created what can be considered a predecessor to modern digital sound modelling software such as Max. MUSIC, developed by Mathews at Bell Labs in 1957, was the first computer program that allowed for the generation of digital audio waveforms through direct synthesis and was one of the first computer programs for making digital sound and music (Manning, 1993). By the

time Mathews have developed MUSIC V, the software was developed enough to be considered a fairly advanced way of producing digital sound synthesis, similarly to the way in which analogue modular synthesizers could have been considered as a complete synthesizer system with a wide range of capabilities. The development of MUSIC V and similar software such as Csound (Vercoe, 1980s) and Music 10 (Chowning, Stanford 1968) eventually led to the development of standardised systems for digital music interfaces such as MIDI in 1991 (Swift, Andrew 1997) which would attempt to create a framework for converting program numbers into musical values, making modern digital recording technology and sound synthesis practice possible.

1.2 Introduction to Relevant Methods of Sound Synthesis

Historically speaking, the field of electronic sound synthesis has been researched and developed for the better part of a century, and as a result there are several different approaches to the topic that each have their advantages and disadvantages. In this section of the thesis, I will investigate several methods of electronic sound synthesis that have played a role in my research and been considered for their suitability at some stage during the length of this research project.

Frequency modulation synthesis, often abbreviated to FM synthesis, was more commonly used methods of sound synthesis for electronic keyboard-driven synthesizer instruments, such as those sold by Yamaha from the 1970's onwards (Milano D, 1975). The process of frequency modulation synthesis is to take a noise source such as a sine wave or a square wave and 'modulate' it with another sound wave of a different frequency. The first wave is often referred to as the carrier wave, and the modulators are often called operators or oscillators (Chowning, 1973). The synthesis was developed by John Chowning in 1973 and alterations such as key scaling were made to the technique in its analogue form such as signal distortion, although digital frequency modulation

eliminated this issue altogether (Holmes, 2008). While any two or more sound waves can be combined together for frequency modulation synthesis, generally the best results occur when the sounds are of a similar frequency range such as 220 Hz, which is the scientific pitch notation of the note “A” and 440 Hz, which is the same note but an octave higher. Frequency modulation as a method of sound synthesis is extremely effective at synthesising complex and interesting timbres using a limited number of oscillators, such as harmonic bell sounds or inharmonic percussion, depending on the relationship of the frequencies used as mentioned above (Dodge & Jerse, 1997). Computational efficiency was an important factor when debating which methods of electronic sound synthesis to utilise for producing samples for this thesis project due to the time it would take to produce and render the audio, as well as the limitations of the available computer systems. Examples of the uses of frequency modulation synthesis include early computer game systems such as the SEGA Genesis and the SEGA Megadrive (Kent, 2001), which incorporated ‘16bit music’ (Collins, 2008) and had very little memory to work with so efficient synthesis with acceptable sound quality made for some interesting results that could benefit from resynthesis making frequency modulation an excellent candidate for the research project. Frequency modulation was also popularised and featured in the Yamaha DX7 keyboards in analogue format and many of the issues associated such as pitch instability became irrelevant with the transition to digital (Milano D, 1975). Despite the elimination of analogue issues with the transition to digital synthesis as the mainstream approach, functional transformation still suffers from the drawbacks of other synthesis methods such as additive synthesis where trying to use too many oscillators to produce a lifelike, realistic sound can cause strain on a standard computer system, heavily restricting the usefulness of the system, although the goal here is not to produce realistic synthesis – simply to use synthesizers to produce a wide range of samples that a neural network can be trained on.

Subtractive Synthesis is another commonly used method of sound synthesis, although it differs a great deal from most methods of sound synthesis. Rather than combining soundwaves together and adding multiple oscillators to create interesting timbres, subtractive synthesis seeks to 'subtract' from a source sound to alter the noise source (Buchanan, 2011). Whereas additive synthesis creates a sound by adding together noise sources and oscillators, subtractive synthesis starts with a sound that contains all of the required harmonic criteria for the final sound, but a modifier is applied to remove any unnecessary harmonic qualities and to shape the sound through the envelope (Russ, 2009). For example, a lowpass filter applied to a subtractive synthesis technique would filter out sounds that are higher than a defined point in the frequency range, and allow the lower frequency signals to pass through, resulting in a bass heavier sound. The synthesizers created by Robert Moog were analogue subtractive synthesizers and were the first widely commercially available subtractive synthesizers (Moog, 1964) and used voltage controls to shape the filter envelope that would determine the sound. Much like frequency modulation, subtractive synthesis is a cheap and easy way of creating unique electronically synthesized sounds and can become increasingly realistic depending on the number of oscillators employed by the system, especially in digital format. Subtractive synthesis is even occasionally used in conjunction with other methods of digital sound synthesis due to the flexibility offered by the simplicity of the system (Collins, 2008). Although the methods of producing synthesized sound between frequency modulation and subtractive synthesis may share some common aspects, the sounds produced by the two are so diverse and different that the use of both can be justified for the sample set and will offer the potential to create even wider sets of samples if more data was to be needed for the research project. According to Martin Russ (Russ, 2009), subtractive synthesis is formed around the idea that there are three parts that any real instrument can be broken down into: the sound source, the modifier, and a controller. In the case of subtractive synthesis, the noise source is generally a sine or sawtooth wave, the modifiers are the variables that can be altered to change the sound e.g. the shape of the volume envelope/ADSR (attack, decay, sustain, release), and the controller is the way in which the user interacts with the

instrument. This is vital for understanding the contextual history for the use of synthesizers as real instruments that are capable of producing a wide range of sounds, which allowed for the development of technology leading up to this point. All of this also made subtractive synthesis a suitable candidate to make samples for use with the neural network.

Additive Synthesis is a method of sound synthesis that creates timbre by adding together soundwaves, usually sine waves. With any musical instrument, the timbre consists of a number of harmonic and inharmonic partials, and each partial is a sine wave of a different frequency that changes based on the ADSR envelope of the additive synthesis model. Additive synthesis is one of the more simplistic methods of electronic sound synthesis in that sine waves of different frequencies are simply added together to create interesting timbres (Reid, 2000). Additive synthesis is the original spectrum modelling technique and is based in Fourier's theorem which is the rule that any periodic function can be modelled as a sum of sinusoids, or waves, at different amplitudes and harmonic frequencies (Marchand, Lagrange, 2001). Fourier analysis is the mathematical technique that is used to decipher the timbre parameters from an overall sound, including non-musical sounds such as birds or water flowing, and allow a researcher to figure out how to recreate these sounds through the use of the correct harmonics, sinusoidal waveforms and oscillators (Evans, 1998). In a musical context, the lowest frequency of a note is referred to as its fundamental frequency, for example the fundamental frequency of 'middle C' is 261.6 Hz and is generally agreed that this is the frequency that the note is played at, although this is just one of several harmonics. Additive synthesis aims to reproduce a sound by producing the exact frequencies that are contained within the sound to give it a rich, harmonic texture (Sami, 1999).

Modern day implementations of additive synthesis, including my own for the purposes of this research project, are predominantly digital. Wavetable synthesis can be seen as a form of time-

varying additive synthesis, wherein periodic waveforms are used to add sound waves together to synthesize a sound at a low computational cost (Andreson, Uwe, 1979). Group additive synthesis is an extension on this, where partials are collected into harmonic groups where each group has a different fundamental frequency, and wavetable synthesis is then used to synthesize each group separately before mixing the results together (Smith 2011). additive synthesis is one of the synthesizers used for this research project, and the justification for this well be explained further on in this thesis. Inverse FFT synthesis is another application of additive synthesis method where an inverse fast fourier transform is used to synthesize frequencies that can be evenly divided in the transform period (Heideman, 1984). Using the discrete fourier transform frequency-domain representation, it is possible to create a form of additive synthesis by synthesising sinusoids using a series of overlapping frames, or sections of the transform period of the function.

Musically relevant applications of additive synthesis that inspired my own use of the synthesis technique include early speech synthesis, which was reported in the pages of 'Popular Science Monthly' (Popular Science Monthly, 1924). Research in this article was the first to state that the human vocal cords produce a harmonically rich tone which is filtered by the vocal tract to produce different tones, at the same time as the first additive Hammond organs were available to the public. Although these organs were expensive due to the number of oscillators required to produce a viable sound, which functioned the same way as a digital additive synth where soundwaves were combined. My own implementation of the additive synthesis technique for this project produces sounds using a form of group additive synthesis, to which the same basic principles apply but on a more complex scale.

1.3 Physical Modelling Approach to Sound Synthesis

One of the fascinating and more complex schools of electronic sound synthesis is physical modelling, which attempts to emulate the physical processes behind the source of a sound rather than approximating the acoustic qualities of a sound or creating new ones (Smith, 2010). Physical modelling uses mathematical algorithms which calculate the physical processes behind the characteristics of a real instrument such as sympathetic resonance with instruments such as the Balinese gamelan (Perrin, 2014) or altering the velocity of an excitation source to randomly create variances in dynamics to emulate a real plucked string's inconsistent dynamics (Fathy, 2004). Two methods of physical modelling were explored for this project; Karplus-Strong synthesis and functional transformation synthesis.

Karplus-Strong synthesis is a method of physical modelling sound synthesis that recreates the sound of a plucked string and percussion instruments by looping a brief noise burst through a delay line with a filter applied to shape the sound (Karplus and Strong, 1983).

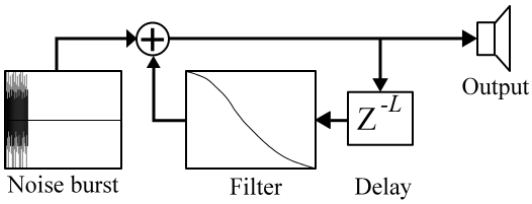

Fig 1. (Karplus and Strong, 2003)

Karplus-Strong synthesis shares similarities to the way in which subtractive synthesis works as both techniques use a filter to remove partials from a noise source, and the Karplus-Strong method uses a feedback loop to recreate the decaying energy of a plucked string by subtracting a percentage of the energy with each cycle to recreate the effect of a string's vibration dissipating into the surrounding air (Fig 1). The label 'Z' in the equation refers to the algorithm's use of Z transform analysis to calculate pitches and decay times of harmonics, contributing to the delay effect in the sound, whereas 'L' simply refers to the length of the note in question. (Karplus and Strong, 1983). The Karplus-Strong algorithm was first produced in 1983 at Stanford University by Alexander Strong and

Kevin Karplus while experimenting with waveguide synthesis for 8bit microcomputers, where the ideas of filtering the wavetable with each pass through the system and using random number generation to determine the velocity of the noise burst to emulate inconsistent human instrument usage were used, leading towards more realistic electronic sound synthesis (Nikol, 2016). The most well-known demonstration of this synthesis method is a piece titled 'Silicon Valley Breakdown' (Jaffe, 1987), for which the Karplus-Strong algorithm was further developed on to improve the control and realism over the instrument by including musical characteristics of real instruments such as tuning, control over brightness, dynamics and alterations to timbre, the end result of which was the Extended Karplus-Strong Algorithm (Smith, 2010). The extended version of the Karplus-Strong algorithm saw the incorporation of a number of features including a lowpass filter which would alternate between different presets for different pick directions, a pick position comb filter, and filters for string stiffness and string dampening which would allow better control over sound and dynamics, albeit at a much higher computational cost due to the additional filters and processes taking place (Smith, 2010). The Karplus-Strong model, however, is still a low computational cost method of physical modelling sound synthesis and was used to create a number of the samples used during this research project. The Karplus-Strong model has been used in the past to produce a computationally efficient physical model of a harpsichord, for example (Valimaki, 2004). Computational cost is kept to a minimum in this model by only generating a half second long noise burst, and creating a tail using a reverb effect that can be lengthened or shortened to create different dynamics and vibrational effects. Interestingly, through experimentation with my own iteration of the Karplus-Strong algorithm, I found that using a 'pink noise' source which is a noise burst where each octave carries an equal amount of noise energy (Szendro, 2001) produced a sound which resembled a harpsichord more than a plucked guitar. The flexibility of the Karplus-Strong algorithm when different types of noise are used allowed for some vastly different sounds to be produced from one synthesizer, which made it an excellent candidate for the research project.

Functional transformation sound synthesis is another method of physical modelling sound synthesis that was considered for the research project. Functional transformation is a mathematics-based method of synthesis that uses the 'Sturm Liouville' transformation in which the fixed variable *Y* is a function of the free variable *x,* and although barely used in acoustic signal processing due to technological limits related to computational efficiency, many issues with standard physical modelling synthesis are avoided. One such issue is discretization, the process wherein continuous functions variables and models are transferred into discrete counterparts which can cause issues (Brown, 1996), referring to the fact that the use of a Sturm-Liouville equation prevents a process called discretization from occurring, which is the process of converting data from a frequency based, equation heavy modelling approach such as modal synthesis into a format which works better with digital sound synthesis and computer systems, simply because the output is more easily recognised by the systems operating system, making optimisation easier. (DeCarlo, 1989)

A discretization error occurs when numbers become finitely small and computer systems struggle to differentiate from data that is too similar to other data. However, as computer systems continue to improve and develop as stated by Moore's Law (Moore, 1965), this becomes less and less of an issue as time goes on. The functional transformation was first introduced in 2003 by Dr Lutz Trautmann and Dr Rudolf Rabenstein, and according to their paper on the introduction of the technique, the first step of the functional transformation method is 'the mathematical description of the sounding object, in terms of a PDE with several and initial boundary conditions'. A PDE, or partial differential equation, which are used to mathematically formulate and solve physical problems with several variables that are considered with physical modelling synthesis such as the propagation of sound/heat, electrodynamics, vibration etc (Evans, 1998). The use of PDEs in sound synthesis allows for several traits of an instrument such as transfer of force and the effect that has on dynamics, resonance in a realistic environment, sympathetic vibrations and more to be modelled in an efficient way. Partial differential equations are often used outside of computer music and physical modelling for a wide variety of purposes, ranging from medicine to model the growth or spread of disease to

physics for describing the movements of waves, pendulums and chaotic systems (Koss, 2017). Although functional transformation synthesis was not used to produce any of the samples for the research project, an implementation of the method was still considered and therefore merited discussion. As a counterpoint of physical modelling synthesis to the Karplus-Strong approach which is grounded in attempting to physically emulate the characteristics of the plucked string, a more mathematical approach to sound synthesis opens another avenue of discussion that could be relevant in developing this project post-thesis.

1.4 My Approach to Electronic Sound Synthesis for the Project

The purpose of this survey section was to explain and present an understanding of the field of electronic sound synthesis, and the individual synthesis methods used in this project. As the main body of sounds for this research project and composition have been produced by synthesizers, I have built myself, arguing the benefits and shortcomings of each synthesis methods was vital in choosing which synthesis methods to include. Every synthesizer used was designed using Max and was chosen with a careful balance of research, testing and process of elimination which will be explained later in the thesis. To begin this section, I will briefly introduce the Magenta Labs NSynth. The Neural Audio Synthesizer, referred to in this paper as the NSynth is a combination of artificial intelligence and electronic sound synthesis that produces sounds unlike any other synthesizer available before its introduction. Unlike traditional synthesizers that use oscillators, wavetables and noise sources etc. to produce sounds, the NSynth uses deep neural networks to break down sounds, rebuild new sound combinations and present entirely new sounds which are a hybrid of sonic characteristics of the original sounds (Magenta, 2017). The NSynth instrument uses a virtual 16x16 grid to provide the user with control over timbre and dynamics, allowing new sounds to be created that would be difficult to produce with traditional methods of electronic sound synthesis. The NSynth functions from a huge dataset of musical notes and instruments to recognise a wide variety of input from different samples, as well as a machine learning algorithm that can accurately

represent instrument sounds learned by the algorithm. My own implementation of the NSynth is more focused on using the same process used by Magenta to produce the sounds for their instruments to produce my own samples from tailor-made electronic synthesizers rather than using the instrument itself as a composition tool, to experiment with using a mixture of wildly different sound synthesis methods to produce altogether new and interesting sounds. The synthesis methods used in this project are a mixture of traditional methods and physical modelling methods and are not conventionally used to produce sounds, and the hope for this research project was to prove that these sounds could be combined together to create new ways to compose with the approaches to electronic sound synthesis.

The main methods of synthesis used in the final project are listed below:

- Karplus-Strong Synthesis
- Frequency Modulation Synthesis
- Additive Synthesis (wavetable synthesis)
- Subtractive Synthesis

Using these four synthesis techniques, sixteen sets of samples varying in sound were produced and prepared for use with the NSynth's virtual grid. Using this prepared instrument, a composition based around the early computer music composition work by the likes of John Chowning was produced, in order to prove the instruments validity as a computer music tool for both research and composition. The methods and technology behind the development of this instrument will be explained during the relevant sections of the thesis related to the topics.

My own implementation of the Karplus-Strong synthesis technique is capable of producing a wide variety of sounds by combining the standard Karplus-Strong method but instead of using a standard sine wave as a noise source, the instrument allows for up to four different noise sources to be

blended together to produce a variety of timbres. For example, combining pink noise and a saw wave produces a timbre similar to that of a harpsichord, simple white noise produces a sound closer to that of the traditional plucked guitar. It is also possible to use the instrument to produce basic percussion sounds by altering the note length within the instrument's code. While the instrument was not the most flexible in terms of sounds produced and was actually rather restrictive and specific to plucked string samples, features in my own implementation of the synthesizer allowed for much experimentation with the different varieties and textures of sounds, making the final composition much richer in sound variation.

For the frequency modulation synthesis element of the project, the original implementation the ability to alter the shape of the ADSR envelope with two modulators to produce unique sounds. The sounds produced by this instrument more closely resembled a typical electronic synthesizer rather than the instrumental realism of the previous approach. However, variety of sound between the different methods of synthesis was an important element of the project in order to produce distinctly unique sounds with the NSynth's capability for combining and resynthesizing sounds.

My implementation of the additive synthesis (Fig. 2) is a standard additive synthesizer with five oscillators that function as filter banks which allow the user to manually change the shape of the ADSR envelope in order to manipulate the sound and create interesting sounds. The program allows the user to 'draw' in the shape of the envelope in order to create and experiment with different sounds as a result of the shape, and the boxes below allow the user to toggle the oscillators on and off to further customise the result. It is also possible to control the note length, with longer notes allowing the user to hear the results of their experimental combinations more clearly, which is a useful feature for creating dynamic and fluid sounds that could potentially be used in composition.

Some such sounds I managed to produce through experimentation with the synthesizer included natural 'wobbling' or vibrato sounds and harmonica-like tones as well as drones, percussion and

long, drawn out organic like sounds, which in the true spirit of the NSynth stray away from sounds you would typically expect to hear from synthesizers used to produce music (NSynth, 2017). Although most implementations of basic additive synthesizers feature just one noise source that is used for each oscillator, I added the option to choose between four noise sources (sine, saw, rectangle, triangle) to add diversity to the set of sounds the instrument is able to produce in the final package. I also added the ability to change the length of the notes so the sound could be further manipulated for use with creating samples for the NSynth. For the samples I created from this synthesizer, I chose to utilise a harmonica-like theme due to the fact that I felt a baseline ordinary synth sound was needed for the project rather than using every synthesizer to produce outlandish or unconventional sounds, especially as a composition was to be used to demonstrate the system.

Subtractive synthesis often functions in a way which is not dissimilar to additive synthesis and this particular implementation is no different. The synthesizer, designed in Max in the form of a noise generator, uses many of the same techniques as the previous synthesizer. However, in this implementation, the function objects act as filters that subtract from the sound rather than adding several together. As above, there is the option to change the noise source as well as alter the length of the notes, and the ADSR envelope can be used in the same way to shape the sound although the results are much different. My initial experimentation found that sounds produced by this subtractive synthesiser produced sounds that much more closely resembled traditional sound synthesis, specifically 1970s/1980s analogue synthesisers (Verlag, 2008). This synthesizer was one of the more limited in terms of the range of sounds it could produce compared to the Karplus-Strong synth or the additive synth, however the quality of samples produced proved that it was suitable to be one of the four synthesizers featured in the project due to its relative simplicity and to act as a more run-of-the-mill synthesizer to combine with others, similarly to the project's implementation of additive synthesis.

Rationale

In this section, I will provide justification and rationale for the major decisions made during the research process, including reasons for choosing to research particular synthesis methods and why these methods were used in the next steps of the project, as well as why certain approaches to electronic sound synthesis were used rather than others. I will also explain the process behind why the decision to construct my own synthesisers was made, rather than just use pre-existing options already available for use.

The decision to carry out research into frequency modulation was a fairly straightforward one due to the fact that I already knew that as a synthesis method, it was easy to use and simple to produce my own version of the synth and adapt it to my needs. As a method of synthesis, it is well known to be fairly efficient and able to produce a variety of sounds, depending on the types of operators that are being modulated. The reason that I carried out research into FM synthesis is to provide historical academic context for the use of it in my thesis, which in turn provided examples in the ways in which the method has been used and can be used for my own work. The effect that these decisions had on my work was that I had a simple baseline synthesizer to work with that was able to produce a number of sounds, though not overly complex, that could be easily manipulated for my needs during the composition and creation of new sounds.

During my research, subtractive synthesis was regularly mentioned as a form of electronic sound synthesis that was capable of producing interesting sounds in a different way from other methods of synthesis, and as such it seemed logical to include it in a project where the aim was to blend together unique and different approaches to synthesis. The rationale behind carrying out research into this method was to properly understand the context in which it has been used in the past, such

as the ones created by Moog and used in popular music (Moog, 1964), and the strengths characterised by the method. Due to the fact that subtractive synthesis has been proved to be a computationally cheap way of producing unique and increasingly complicated sounds depending on the number of oscillators and filters used, the obvious choice was to produce my own simple iteration of the synthesis approach. The impact that this decision had on my project was that I now had two simple synthesis methods where I could easily produce my own iterations that could create varying sounds in a time effective manner, which would be more suited to the final goal due to my own understanding of the types of sounds needed for the project, which would be discussed later on in the project as the research for the composition took place. The same thought process applied to the use of additive synthesizers in the project; simple to produce my own version and sounds, computationally effective with varying degrees of complexity and realism depending on how in depth the user may want to go. The fact that both synthesis methods allowed for more complex and realistic sounds to be created depending on time available, need for more complex sounds at the cost of computational efficiency had interesting implications for the project, specifically in terms of future work. With more complex sounds and, it is worth considering the impact that it could have on the project, further reinforcing the rationale behind choosing these methods of synthesis in the first place.

The use of physical modelling synthesis as a whole and particularly Karplus-strong sound synthesis was a partially aesthetic choice, as well as a practical choice due to the variation in sound produced by physical modelling-based synths compared to more conventional synthesis methods, which lined up with the initial aims of the project to use AI to combine types of sound synthesis that would usually not work together, which made it an easily justifiable use of research time. As for the aesthetic approach to using the synthesis method, the appeal of a realistic plucked string sound

offered interesting sonic characteristics for composition, particularly when taken into consideration with my own musical background as a guitarist and songwriter. The rationale for using this rather than just attempting to recreate these sounds with other non-physical modelling synthesis methods that may have been computationally cheaper to produce was to be more faithful to the original aims of the project, which was to utilise a variety of synthesis methods to produce the sounds in the first place, inspired by Magenta's work in combining musical instruments and other sounds not usually found in music.

The decision not to use functional transformation synthesis in the project was ultimately a decision made to manage the scope of the project, as well as the fact that the other synthesis methods selected covered the same project requirements in a more computationally efficient manner than functional transformation. However, carrying out the research into the synthesis method and ultimately deciding against using it was beneficial in the effect it had on the thesis project in the respect that it offered contextual information into concepts such as the sturm-liouville transformation and discretization errors, which allowed me to understand the synthesis method as an alternative for further work on the project, if it becomes an option to dedicate more time to the research and implementation of further ideas.

One of the key motivating factors for carrying out the synthesis survey research section of this thesis project was to gain enough contextual understanding on the strengths and weaknesses of different synthesis methods in order to understand them and best utilise them further on in the project. However, one logical way of saving time which could have been devoted into other strands of research would have been to use other existing implementations of the synthesis methods to

produce the samples rather than constructing my own versions of the synthesisers. The justification for producing my own synthesisers was that I found that my own implementations of the different synthesis methods allowed me to have more control over the sounds produced, particularly with my Karplus-strong synth, which allowed me to blend together different types of noises and soundwaves to further alter the samples. The whole point of producing my own synths and using different synths in the first place was to experiment with interesting sounds and blend them together, and the best way for me to generate interesting sounds was to build my own synthesizers from the ground up and have total control and understanding of what kind of sounds can be produced by the systems.

2 Survey of Artificial Intelligence and Machine Learning Techniques

Machine learning and AI are terms broadly used to describe computer systems that mimic cognitive functions and processes associated with the human mind, such as problem solving or learning behaviours and patterns (Nilsson, 1998). Artificial Intelligence is a term often misused to describe any learned behaviours by increasingly advanced computer systems and as these systems have continued to develop, practices that were perhaps once considered 'artificial intelligence' have just become standard computing, such as optical character recognition (Schantz, 1982), and others. Modern examples of Artificial Intelligence include human speech recognition which is implemented in modern smart home devices such as the Amazon Echo, self-driving autonomous cars, simulations that respond to human input and more. Musicians and artists have often utilised new technologies for creative purposes, and artificial intelligence is no exception. Examples of this include audio processing tasks such as audio classification and automatic music tagging as tools for categorising music, but also composition-based projects where a machine learning algorithm has been trained on the works of classical artists and been tasked with the project of composing something to resemble the composer (Herremans, 2010). Other musical applications of machine learning can include "Learning Features from Music Audio with Deep Belief Networks." (Eck, 2010) which focuses on feature extraction which has similarities to my own project, and "automatic music genre classification" (Silla Jr, Carlos N., Alessandro L. Koerich, and Celso AA Kaestner, 2008).

2.1 Relevant Historical and Contextual Information

"*We propose that a 2 month, 10 man study of artificial intelligence be carried out during the summer of 1956 at Dartmouth College in Hanover, New Hampshire.*

The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it. An attempt will be made to find how to make machines use language, form abstractions and concepts, solve kinds of problems now reserved for humans, and improve themselves.".

- Dartmouth AI Project Proposal; J.McCarthy et al.; Aug. 31, 1955.

As discussed above, the definitions of artificial intelligence and machine learning has shifted over the decades, which makes it difficult to discuss the emergence of the field. The origins of work into the fields of machine learning and AI can be traced back to the 1950s with the development of computing, and the field was technically founded in 1956 at a conference where the term 'artificial intelligence' was first used to describe computing based on human cognition and decision making processes. During this period, the question was not whether or not computers would develop to the point of being as advanced as human intelligence, but simply when and how this would happen. The 1956 conference at Dartmouth College hosted by John McCarthy brought together expert researchers from several fields including complexity theory, language simulation and neurology to plant the seeds of what would eventually become the field of artificial intelligence. This gathering created the field of AI which served and continues to serve as a backdrop for most computer research, as well as the public perception of artificial intelligence thanks largely in part to science fiction and movies and media in pop culture.

2.2 Uses of AI in Music

As with most technological developments, artists and musicians in particular have used machine learning and artificial intelligence to create, perform and experiment with music. Machine learning as a subset has been particularly utilised within music (Miranda, 2013) in fields of classification, prediction and data analysis and more. Naturally, music has often been used to show the capabilities of new AI technology as demonstration platforms, as well as creative exploits to, for example,

extract and analyse patterns that occur in classical compositions and use those patterns to compose new music in the style of the original (Cope, 1991, 1992). These substantial results of machine learning experimentation are useful as tangible ways of showcasing what machine learning technology is capable of to those who may not be familiar with the field, and to musicians who may end up creating using these tools which is not unlike the aims of my own thesis project which will be demonstrated with a composition. In the section below, I will briefly discuss projects that have incorporated music and AI which helped direct my research by providing context as to what creative material can be produced using the field and technology.

The first AI music experiment I will discuss in this section is Cypher (Rowe, 1992). Cypher is an interactive real-time system that functions with two components – the listener and the player. In essence, the 'listener' element of the system analyses streams of MIDI data and the 'player' uses several different algorithmic systems to produce a new musical output. The listener element functions by classifying data from the MIDI input and learning the behaviours of the data over time, at which point the classification becomes more accurate. Classification categories include speed, dynamics, harmony and rhythm, and a user can configure Cypher to react differently to different classifications to, for example, put more of an emphasis on rhythm or dynamics. A graphical interface, not dissimilar to the NSynth, allows for the user to specify the relationship between the classifications and the types of response produced by the algorithm. Presets and collections can be saved and recalled for later use to aid with experimentation, which is something that could be a potential feature included in the future development of my own project. Another interesting feature that Cypher utilises to produce musically coherent results is an 'internal critic' that has a set of programmed aesthetics to keep the output consistent. The system is also capable of producing music without an input, relying on the algorithms to recall previous input or generate new random but musically coherent output. Systems such as cypher are relevant examples of music and AI due to features such as the aesthetic limiter to keep results musically coherent, which is achieved in a sense

with my own project by carefully choosing the sample sets used by the system to ensure the results are usable in the composition.

Although optical music recognition may not still be defined as artificial intelligence, the two still share similar attributes in terms of categorising music and the way they can be used as tools for understanding music. Systems that are now assumed as commonplace such as the ability to take musical input and convert it into MIDI data so the characteristics of the piece such as pitch, timing, pitch and velocity can be pinpointed as a form of 'automatic transcription (Rebelo et-al, 2012) of course, technologies related to this have developed since and this is no longer considered AI, it is still relevant as a system that takes a musical input, breaks down the characteristics and uses this information to reproduce music, much like the NSynth and the focus of my own project. Optical recognition is, however, experiencing a form of revitalisation as researchers combine OCR tools with artificial intelligence practices to capture and comprehend information, which means that AI tools can be used to check for mistakes in these pre-existing systems without human intervention (Dixon, 2000).

Taking a more modern approach to applying AI technology to music and images, Magenta is an open source research project that seeks to use artificial intelligence and specifically machine learning as a tool for the creative process (Magenta, 2017). Magenta is essentially an open source python library which includes a number of projects and assets that can be used as ableton plugins for music production, utilising machine learning technology and showcasing the possibilities of the research carried out into the field, such as the NSynth which will be discussed in depth in this thesis. Other magenta based projects as part of the 'magenta studio' include a number of standalone applications that can be used to compose music, proving that combining artificial intelligence (to randomly generate music in this instance) is a valid and modern use of the technology (Souppouris, 2016).

Potential applications of machine learning and music are constantly becoming apparent as the technology develops, and technologies such as the NSynth are themselves recent developments into the field. Other potential applications of machine learning include the artificial intelligence musical composition project, The Watson Beat, produced by IBM research. The Watson beat uses reinforcement learning and deep belief networks to produce music based on a melody input and a selected musical style preset. Reinforcement learning is another area of machine learning which involves considering how software agents operate in an environment to maximise the reward through repetitive action, presumably input from a data set, and differs from supervised learning due to the fact that the focus is on balancing exploration of new data with exploitation of current data learned by the machine, to 'reinforce' current knowledge acquired by the system (Kaelbling et al, 1996). Deep belief networks are a similar to deep neural networks in the fact that they are made up of several hidden layers and a visible layer as variables where information acts as the input to the system, where each layer learns to transform the data into a more abstract and composite representation of the data in tiny increments, depending on the number of layers (Hinton, 2006). Music created by artificial intelligence with examples such as the Watson Beat proving that combining these two fields can produce comprehensive results gives validity to the idea of using AI to resynthesise synthesizers as an instrument to fuel new musical compositions and creative ideas.

2.3 Relevant AI Approaches

A field of research as broad as artificial intelligence is bound to have several approaches to achieving the end result, some of which are more relevant to music whereas others are specifically designed for other fields of research. In this section, relevant AI approaches will be discussed to provide contextual information and to demonstrate my understanding of the work that went into developing the NSynth's neural network.

The difficulty with applying any artificial intelligence algorithm to a creative problem is that the network is not capable of producing the human spark of creativity. As a substitute for this, randomness is often employed in the form of random number generators. Markov chains are often an engine for this solution and are crucial to the way in which many AI algorithms work. A markov chain is a mathematical system that transitions from one state to another according to pre-defined rules of probability (Osipenko, 2019) where the probability of the next state depends on the previous state and not on the previous states of the entire sequence. This allows the predictions to be more computationally efficient due to the fewer number of variables, and more 'random' as previous variables are not considered as much. Markov chains are useful for producing music by, randomly assembling chord sequences that are musically coherent based on pre-defined rules. This example of using an algorithm to substitute a human quality of music is intrinsic to the process of using artificial intelligence with the creation and production of new music and musical tools.

Deep learning and deep neural networks are the most relevant field of artificial intelligence to my project and is the foundation for which the Magenta Labs NSynth was based on. Deep learning is a type of machine learning algorithm that uses several layers to extract raw features from the input, with each layer extracting more and more intricate and detailed information. Therefore, the more layers, the more complex and intricate the resulting data is (Deng, 2014). In a musical context, a deep neural network with more layers may be better at accurately recreating and learning details of a piece of music or a sound, which is an important attribute when attempting to achieve accurate and musically coherent sounds for use in further research. The key attribute of deep learning that makes it so useful for the kind of result that the NSynth hopes to achieve is that with each level, the algorithm learns to transform the input data into a slightly more abstract version of itself. Thus, after several iterations, a fairly accurate and recognisable version of the original sound is produced and this abstract version of the sound can be used with others that have been through the same process

for further development and research, such as re-synthesis and composition. Feature retrieval features of a deep learning neural network are often used in Music Information Retrieval (MIR) systems, and features that are often targeted for abstraction include qualities related to spectral, timbral, temporal and harmonic characteristics which were qualities I ensured where defined in my own implementations of synthesizers that would be processed with the deep learning network. While this technology was originally designed for genre classification, this works particularly well especially with a wider data set that algorithms such as the NSynth were trained on as a majority of abstract sound characteristics will be easily recognised as vaguely representative of something else due to the in-depth level to which the features of a great quantity of sounds were broken down to(Magenta, 2017).

2.4 Wavenet, Tensorflow and NSynth

The following section will discuss the workings of the NSynth and contextual information relating to how my own thesis project relates to this work, as well as a discussion about Tensorflow and Wavenet, two projects that allowed the NSynth to be created and hold a great deal of research value and relevance to my own findings.

Wavenet is a deep generative model of raw audio waveforms that was originally designed for human speech generation but has also been demonstrated as capable of synthesizing other audio signals including music and producing accurate automatically generated pieces of music for the piano (Deepmind, 2016). Modelling of raw audio is usually avoided during research due to the fact that the number of samples per second (approx. 16,000), which is why building an autoregressive model in which predictions in the model are influenced by all previous observations proved to be a challenge for researchers. This led to the one-dimensional structure of Wavenet, which is a fully convolutional neural network where each of the layers have dilation factors that allow each level of the neural network to grow exponentially and produce a higher detailed abstraction of the original sound. This

allows for the creation of highly accurate versions of sampled sounds that are easier to work with rather than raw audio and its extremely high sample rate, allowing for projects such as the NSynth to exist.

Neural networks such as these are trained on real data to improve their functionality with practical applications, while Wavenet is trained on input data of real waveforms recorded from human speakers, at which point the process for synthesising the samples is carried out. This is not dissimilar to the way in which the NSynth functions with its wide range of sampled training data which will be explained further in depth below. This method of training the neural network is computationally expensive but is necessary for generating realistic sounding speech synthesis as well as other audio. When the system was trained on pieces of piano music instead, it produced fascinating results that sounded somewhat like a random note generator in the fact that it still retained an aspect of musicality, although my iteration of the NSynth was used for the synthesis of short audio samples so this particular musical application will not apply. However, it is still of interest that there are further musical applications of the technology that could be explored in further projects.

Tensorflow is a machine learning system that operates as a training platform for the NSynth in conjunction with Wavenet as the autoencoder for the sound samples (Rajat; et. al 2015). Tensorflow as a system functions by operating at a large scale in diverse environments for projects that require a large amount of data to be processed to train a neural network, which is highly suitable for machine learning projects such as the NSynth where datasets consist of millions of samples. The reason that Tensorflow systems can handle such a large quantity of training data is due to the fact that the system maps the nodes of the of the data flow across several machines in a simulated cluster, which in itself can operate across several computational devices or multicore computer systems or specialised ASICs known as Tensor Processing Units, although these are obviously only available to the developers of the system and designed as proof of concept and not for the average

user. This configuration allows for developers to experiment more with training data in the virtual environment more than past iterations of similar systems such as Distbelief (Perez, 2015). Several google-led projects and services such as the NSynth itself use Tensorflow as the base for their data training systems due to its flexibility and potential for real world applications, as demonstrated by the capabilities of the NSynth and the capability for users to train their own data on the NSynth system to produce music (Magenta, 2017). When training abnormally large data sets such as the case of the NSynth, Tensorflow utilises a distributed representation, which uses a training example as a pattern of activity across each different machine in the clusters so there is already a framework that the algorithm can follow. Due to the fact that training a model can take a long period of time, even several days and use a large number of machines, any sort of assistance to speed up the process is crucial to saving time and money in the process.

Finally, the NSynth, or Neural Synthesizer is a deep neural network-based synthesizer that uses sounds generated at the individual sample level in combination with other sounds to produce new and interesting sounds (Nsynth, 2017). Unlike traditional synthesizers, the NSynth uses deep neural networks such as Magenta to resynthesize sounds based on its enormous data set of nearly 3 million samples. NSynth provides artists and creators with absolute control over timbre, dynamics and other characteristics and my implementation of the NSynth endeavoured to increase this control by allowing users to produce their own sample sets from a number of pre-designed and carefully selected synthesizers chosen for their sonic characteristics. As mentioned above, the NSynth dataset is especially large even for the machine learning community, and was deliberately designed to be so for two reasons; to develop a creative tool for musicians so that any potential sounds inserted into the system can be at least somewhat recognised and sampled and return a fairly accurate result, and to push the limits of the machine learning community and create a challenge in terms of generative models for music. This development allowed for my project to exist due to the fact that my own samples come from self-designed implementations of pre-existing synthesis techniques which means that a Karplus-Strong synthesizer, for example, may still be recognised by the system as

reminiscent of something such as a guitar or percussion in the large data set. According to the creators of the NSynth, the motivation of the data set was that it allowed for the factorization of the generation of music into notes and other characteristics of music, and potentially more but for the sake of simplicity it was reduced to two variables (equation. 3)

$$P(\text{audio})=P(\text{audio}|\text{note})P(\text{note})$$

equation 1, NSynth Temporal Embedding Factorization Model(Magenta, 2017)

The aim of the equation is to model timbre under the assumption that the note aspect of the equation comes from the user input, represented by *P,* which produces results that are musically coherent. As mentioned above, the NSynth utilises Wavenet to train the system with temporal embeddings, in this context found in the form of speech or music. Temporal embedding is the way in which the NSynth system encodes samples so that the neural network can understand them after the sample size is small enough for the system to be able to handle it, even considering the size of the samples compared to the size of the training dataset (Liu, Et Al, 2015).

The best way to contextualise the use of the NSynth as an instrument is to view it as a tool that can be used to combine two unique sounds, musical or otherwise, and use them to produce a new sound which has timbre and dynamic characteristics of both original sounds. Although this is a massive oversimplification of the process that the neural network goes through to produce the end result, the NSynth itself was demonstrated in the form of an interactive instrument where one could combine the sounds of a saxophone with a thunderstorm or a flute with a bass guitar (Nsynth, 2017).The creators of the NSynth also relased the NSynth Super, which is a physical instrument where 16 source sounds are loaded into a small handheld synthesizer which operates on a 4x4 grid where the user can use the outcome of the neural network to create unique sounds. This is of course

far more restrictive than the full version of the NSynth that has been implemented in this project but is still a good way to increase public understanding of the uses of neural networks and artificial intelligence for the creation of music and art (NSynth Super, 2018).

2.5 Using AI in Sound Synthesis

Artificial intelligence and computer music have become heavily integrated as two fields of research and as a result of this connection, several uses for artificial intelligence have been created for music. One such use is evolutionary computing, which is a field of research concerning evolving synthesis parameters. Evolving synthesis parameters refer to an idea where, for example, sixteen random variations of a synthesizer could be generated. Then, based off a smaller number selected from the original sixteen, another sixteen variations based on those selected would be generated and so on. Evolutionary computing as a field of artificial intelligence is one with particularly interesting applications to musicians as compositional tools and simply expanding the boundaries of what can be produced by ordinary synthesizers, an idea which is at the heart of my own research project. Evolutionary computing algorithms have three basic properties: inheritance, random variation and selection. These properties are fairly self-explanatory in their function. However, the latter is the most crucial to producing the example of evolutionary computing with music explained above. In most computer implementations of selection, this is carried out by applying a 'fitness score' to variables to measure their suitability for the next step in the evolutionary process, allowing future generations to be produced with the characteristics of the variable, such as a certain sonic element or sound shape, to apply a musical context However, fitness criteria can be problematic when it comes to changing the aesthetic of a piece or instrument mid-way which is something that composer often do to create texture and variation within a piece of music. This also raises the issue that it is difficult to define what a 'good' sound is in a particular musical context due to the fact that this can be a matter of both circumstance and opinion. Despite this issue, evolutionary music is still

an interesting way in which artificial intelligence is being used with music to expand the boundaries of what can be done with electronic sound synthesis.

Although not an example of artificial intelligence, ChucK is an audio programming language used for real time electronic sound synthesis which is on occasion used in tandem with software reliant on AI, as well as composition and performance. It is well known for its ability to 'live code' which means that code can be added mid operation without the need to stop the code or restart the program. ChucK has been used in live performances by PLOrk (Princeton Laptop Orchestra) and for developing applications with the American mobile app developer, Smule. Smule developed an ocarina emulator that used AI to learn from samples and produce a realistically synthesized emulator with potential applications for composition, production and performance.

Rationale

In this section, I will justify decisions made throughout the stage of researching relevant approaches of AI to the project, as well as the steps that lead me to magenta labs research and the NSynth. This includes why research was carried out into both cypher and optical music recognition, why magenta was used and how that lead to the NSynth as well as why I opted to use the NSynth with my own synthesis implementations rather than use the NSynth super, which was already more a functional instrument with less work to produce compositions.

The main aim of carrying out research into other relevant approaches to artificial intelligence and music was to gain an understanding of different ways in which the two fields of research have been combined, and how these developments might have led to the NSynth at the core of my project. Certain aspects of the cypher project such as the graphical interface making it more intuitive to use and the ability to create presets are not only shared with the NSynth and the NSynth super, but are

common in many compositional tools such as virtual software instruments and plugin effects, and thus research can be justified in terms of practicality and aestheticism towards a functioning instrument at the end of the project. The impact that research into cypher had on my thesis project was providing context and inspiration in how to present my own musical AI project, and presenting ideas for ease of use such as the ability to save presets. While this approach of combining music and AI was not the most relevant compared to other work discussed in the main body of this chapter, research still proved to be valuable in terms of providing contextual information.

The decision to research into optical music recognition considering that it is strictly not considered AI was to gain contextual knowledge into similar work to the NSynth, which in this context is the shared functions between the two, such as the ability to categorise, break down and sort musical information, often in the form of music notation. The research had an impact on my own project in allowing me to gain more of an understanding on exactly how software like this is used to perform musical experimentation in other ways and provided some inspiration for my composition after seeing examples of the outcome of optical musical recognition systems, such as audible versions of music scores that have been input into the system (Alexander, 2019). This provided context for using composition as a demonstration of using AI and sound synthesis together, and academically justified the idea making the research worthwhile.

The decision to research into magenta was to provide context leading into the NSynth which was key to the practical stage of the project in producing the new instrument sound samples. Magenta itself was more of a collection of individual projects including the NSynth but understanding the predecessor projects that led to its development was instrumental in understanding how artificial intelligence was used in the context of the project, as well as the steps taken to develop the successor project. Features of magenta projects such as utilising machine learning technology and

showcasing the possibilities of the research with composition remain relevant to my own project and provide reason for the surveys.

Although there are alternatives to the NSynth to achieve the aim of blending together different sounds as explained in this thesis, justification for why the NSynth was chosen is required. From an aesthetic and conceptual standpoint, the aim of this project was to find ways in which AI and electronic sound synthesis can be combined, and other methods of achieving the same goal, and the NSynth met those requirements perfectly while providing room for modification and experimentation beyond the initial parameters of the project, in which the program was demonstrated by combining, for example, the sounds of a mellotron with the meow of a cat. The program was discovered early on in the research stage, and quickly became as core a part of the project as the synthesis methods that my own implementations of the instruments were based upon. As a result of work from Magenta, a more functional, easily accessible version of the NSynth called the NSynth super was produced and is discussed in this paper, and the choice to use the NSynth rather than this more easily accessible later version formed a key part of the future of the project and had a large impact on the research that came after, in later stages of the project. The decision not to use the Nsynth Super was to maintain more control over the samples produced rather than just using the pre-provided instruments, and the predecessor was easier to manipulate and modify to suit the needs of my project.

3. Project Approach

The purpose of this section is to discuss the project approach to demonstrate the process I took, from the inception of the project including initial ideas that led me to the project idea behind this thesis, the process of preparation and carrying out initial research, as well as early experimentation into designing the synthesisers and becoming familiar with the NSynth and neural networks as a concept. Other aspects of discussion in this section include changes to the project that were made throughout the process due to issues encountered or restraints such as time or access to technology, as well as simple creative decisions such as choosing how to demonstrate the project and how further research would help develop these ideas. Coding the synthesizers was one of the most important steps of the project in terms of selecting and refining them and as such I will discuss the process and testing that took place, as well as the decision to simply utilise an existing version of the NSynth rather than develop my own neural network due to issues related to the scope of the project. Finally, I will discuss issues encountered during the project and how this led to changes throughout the entire process and how this allowed me to refine the project into its current state. The process of testing and producing the samples as well as producing the composition and the inspiration for that will also be elaborated on in order to clarify the relevance of the research to composition and to contextualise the usefulness of the final project as a tool for composers and researchers.

3.1 Project Inception

The initial inception of this project came from an interest in designing synthesizers, particularly physical modelling-based synths such as the Karplus-Strong synthesizer. After an initial project attempt to construct a physical modelling-based synthesizer that would reconstruct the sounds of a blue whale for use as a research tool, the idea of creating a synthesizer that could be used as a research tool as well as a compositional tool led to looking for a way to accomplish this with synthesis as the main focus of a new project. After discovery of the NSynth and the work of Magenta

Labs (Magenta, 2017) the process of combining sounds to create new sounds proved to be suitable for the project scope and timeline. This also offered machine learning and artificial intelligence as a field of study, which proved to be a useful direction in which to take the project as the implications of using machine learning for creative purposes is a field with a great deal of existing research to use and allowed me to better understand how to use the NSynth. Some of the research from the initial project idea was able to be re-used as a large part of the focus of that project was to focus on methods of synthesis that offer non-conventional sonic qualities in the sounds that they produce, such as frequency modulation, where obscure synthesizer effects could be combined with less obscure and more standard sounds to fuel new compositional ideas for myself and for others. Much of the early computer music composition carried out by the likes of John Chowning (Chowning, 1973) and Max Matthews originated in using technology that was not designed with music in mind to produce compositions, which is in line with the NSynth and combining AI and synthesis to create new compositional ideas, so using these artists as inspiration felt suitable for the demonstration of the project. This demonstration of the technology with composition as a practical example also helps remove the barrier between musicians who may not be as familiar with computer music practices and electronic sound synthesis due to the contextualisation of the practices in a format that is much easier to access, which is another area of research within computer music that is often discussed, as much of the technology used within the fields such as EEG caps and brain-to-computer music interfaces allow individuals who may not necessarily be able to produce music have a creative outlet.

3.2 Process of Preparation and Research Methods

The first step in the preparation phase was to conduct research into electronic sound synthesis, AI and machine learning as well as computer music composition. Selecting synthesizers for the project depended on four key criteria; ease of use, computational efficiency, range of sounds produced and simplicity of construction (due to time constraints). Additive synthesis seemed like an obvious

candidate due to the fact that most implementations of the synth are simple to construct and are computationally efficient, able to produce a wide range of sounds in my own implementation built in Max on a machine that is not particularly powerful in terms of hardware. Computational efficiency was an important factor because part of the project was to provide the means for composers to design their own samples so the ability to run the synthesizers on any system was vital to the functionality of the project. After experimenting with different implementations of additive synthesis in Max including waveguide synthesis (Andresen, 1979), I found that a simple five oscillator additive synthesisers produced the widest range of synthesized sounds while remaining computationally efficient and were also simple to use by individuals who were not too familiar with how synthesizers work. The construction of the additive synthesizer was also fairly simplistic, and it was quickly decided that a MIDI keyboard in Max was the best way to control the synth in a way that is familiar to composers who use digital audio workstations (DAWs) to produce. Additive synthesis was suitable because it could produce fairly standard synthesized sounds with ease but to one more experienced with the instrument, it could also be used to create more obscure and unusual sounds.

Early research also proved the suitability of subtractive synthesis for many of the same reasons as additive synthesis, primarily due to the simplicity and ease to produce fairly standard synthesizer-like tones and was simple enough in construction within MSP to be easy to use and computationally efficient at the same time. However, the idea of granular subtractive synthesis (Collins, 2007) was also considered due to the interesting tonal qualities of sounds produced, but time constraints meant that a more stripped back approach to the synthesis method proved to be more suitable for the project, as well the simplicity assisting with understanding how to produce baseline sounds that could be more easily understood by users when used in conjunction with other synthesis methods in the NSynth.

Karplus-Strong synthesis was also an obvious choice for the project due to the fact that the results produced by the synthesizer differ so much compared to the previous two methods of electronic

sound synthesis. As discussed above, Karplus-Strong synthesis produces sounds that closely resemble a plucked acoustic guitar and can easily be altered to resemble other stringed instruments such as harps, plucked violins or harpsichords, presenting an even wider set of samples that can be used beyond the initial scope of the project. In addition to that, the recirculating delay loop mechanism of the Karplus-Strong synth maintains excellent computational efficiency and the design allows for ease of use which will be explained in depth in the discussion of the construction of the instruments below.

Frequency modulation synthesis was the fourth method that was researched and selected for the project due to its potential to develop interesting sounds and historical use as a diverse and capable synthesizer. As discussed previously frequency modulation was used in early Yamaha synths (Milano D, 1975) and has remained as computationally efficient in its digital format since the transition from analogue. Simple frequency modulation synths are still able to produce unique sounds although the increasing numbers of oscillators required for more complex sounds can begin to negatively impact computational efficiency. However, most basic implementations of frequency modulation are not aimed at producing realistic synthesis of real-world instruments so computational efficiency issues wouldn't have reached a point where the synthesizer would become too advanced to easily use.

Other methods of synthesis that were considered for the project but weren't found suitable for the included functional transformation synthesis, as well as digital convolution synthesis and sample-based synthesis. These methods weren't selected as they didn't meet the criteria, such as the time it would have taken to build a functional transformation synthesizer and have it able to effectively produce sounds with alterable parameters. In addition to this, using sample-based synthesis did not fit the creative scope of the project and was too close to what the NSynth demonstrated by Magenta had already achieved with animal and world sounds as part of the package, and did not offer the same benefits in a compositional setting compared to other simpler methods of synthesis. Digital convolution was not included in the project because digital convolution is difficult to achieve in Max

whilst remaining computationally efficient, and the results produced by the synth are too dissimilar to the rest of the project to produce musically coherent results for the compositional aspect of the system.

Once the synths had been chosen, the next step was to figure out how best to use the NSynth to produce the sounds and how to demonstrate the capabilities of the concept. Magenta labs provided a web-based demonstration of the capabilities of the NSynth which was presented in the form of a keyboard interface where the user could select pitch, and drop-down menus where the user could select two different sounds to combine e.g. a cat meowing and a trombone, which provided me with some context to how two very different sounds could work together when combined with the NSynth, forming the basis of the project. However, simply using the NSynth as an engine for resynthesizing sounds as a project still lacked research content which led me to conducting a survey of relevant artificial intelligence techniques in order to better understand how the NSynth extracts qualities from the sounds being produced by the synthesizers.

This research focused specifically on techniques related to deep neural networks such as cypher and TensorFlow. The original intention of this research was to assist in the development of my own neural network to demonstrate an understanding of the concepts behind the NSynth but due to time constraints and irrelevance to the overall project this was not further researched. Preparing the NSynth itself for the initial experiments did not take much work as much of the instruction on how to produce samples and prepare them for use with the NSynth was provided by Magenta Labs to encourage users to expand on the tools provided. Initial ideas to construct my own neural network or AI did lead to the process of learning python coding, however, which assisted with understanding much of the background processes behind the NSynth as well as understanding Max for the NSynth's provided software on both systems. TensorFlow as discussed above was used in the development of the neural network which uses Python which would have been the focus of the initial experiment to produce an AI similar to the NSynth although time constraints very quickly prevented this. Python

still played a part in the process of preparing the samples from the synthesizers for the NSynth, however, which will be discussed later.

3.3 Initial Experiments

One of the earliest tasks during the process was to conduct a phase of initial experiments to determine whether or not it was feasible to use the NSynth and selected electronic sound synthesis methods to produce instruments that could be used effectively as composition tools. Due to past experience with using Max to construct synthesizers from my undergraduate music degree program, I used this knowledge to build basic versions of a Karplus-Strong synthesizer and an additive synthesizer. The Karplus-Strong synthesizer was used to produce fairly standard plucked guitar like tones with a touch of reverb to make the sound more distinctive in the end result to differentiate how different sonic characteristics such as effects would be represented. Karplus-Strong synthesis was a good candidate for this initial testing stage of the project due to the fact that it does not resemble any other selected synthesizers in terms of timbre characteristics, whereas simple implementations of additive or subtractive synthesis may be hard to pick apart in a resynthesized sound as they both produce fairly typical sounds one might expect from an electronic synth instrument. Additive synthesis in its most basic form of a five-oscillator synth offered enough depth to the sound to be able to differentiate it in the mix and produced a high enough quality sound to prove the feasibility of the synthesis technique. The samples produced by the additive synth were much more simplistic during this stage of testing with no additional functionality such as blending noise sources or selecting of note lengths, functionality which would be added later on in the development of the instruments. Constructing the synthesizers during this stage of the testing was fairly time consuming which meant that I did not have the luxury of experimenting with several other types of sound synthesis to the same extent although I did construct basic versions of subtractive synthesizers and frequency modulation synthesizers that were later developed into the synths for the final project. The reason that only two synthesizers were tested during this stage of

the project was due to the fact that at this point it was not clear if this project would even yield results that sounded coherent and musically aesthetic, let alone suitable for composition.

During this testing stage of the project, the samples produced by the synthesizer prototypes still had to go through the process of being prepared to work with the NSynth which is carried out in Python and involves the sounds being broken down into tiny samples and configured to work with the neural network, a process which remains unchanged from the main project. However, due to technological issues surrounding the availability of a compatible system and the codes inability to run on a Windows system, the initial round of tests did not produce any results. However, this stage of testing allowed me to better understand the ways in which I could improve the synthesizers in terms of complexity and quality of sound samples produced by the machines, as well as how to better streamline the process of preparing the samples particularly in terms of file management. This issue with the NSynth code not working in the initial round of experiments was particularly time consuming especially considering how long the process of encoding the samples took in comparison to other aspects of the project, approximately 8 to 12 hours per batch of encoded samples. The workaround for this instead was to run the system on a Mac OS, which meant even more delays in the project timeline due to difficulty sourcing access to the equipment needed for enough time to produce the results.

Another aspect of this original round of experiments to consider was how the sounds could be demonstrated, and the best way to do this was to attempt to use them in a musical context. Attempting to compose with the samples required a work around as the output of the NSynth was designed to be used with the NSynth Max patch implementation, and not for composition on projects such as logic. However, it should have been possible to take the files produced by the process and map them to a keyboard by transforming the data to MIDI data to be compatible with composition and music software, which is a process I have used in previous projects to compose with custom built synthesizers so this part of the project was not expected to be too time consuming

as other aspects of the original testing phase had been. One issue encountered during this testing phase of the project is how difficult it was to alter characteristics of synthesis samples if the produced result after the NSynth preparation is unsatisfactory, in the fact that any alterations to the sound requires restarting the entire process with a new set of samples from the base synthesizers, restarting the encoding process all over again. This is more of a flaw with the project that may be altered one day in future developments by somehow combining the digital Max built synthesizers with the NSynth system, but that is not the aim of this project and is merely an inconvenience at this stage.

3.4 Constructing Synthesizers and Understanding Neural Networks

As stated previously, all the electronic sound synthesizers for this project were constructed in Max due to the fact that the software is extremely capable of allowing the user to produce high quality synthesizers in a way that other coding languages and software would simply not be able to. The workflow of Max as a program is extremely logical and works perfectly with designing synthesizers particularly in the way that the software allows for the user to very easily produce noise objects and sculp the sounds to produce basic additive or subtractive synthesizers with little prior knowledge, and the support offered online via the Max forums and community allowed me to fill in the gaps in their knowledge or troubleshoot issues and came to be extremely useful in the development of my synthesizers. While Max was the perfect tool for building the synthesizers, however, it lacked the capability to work with artificial intelligence and machine learning, so python was used for the elements of the project related to those fields instead.

Regarding my own implementations of the synthesizers, the process of designing and constructing them was a combination of both my own knowledge and information provided by the community. For example, I was already fully aware of how to construct a version of the Karplus-Strong synthesizer and an additive synthesizer, but I was unfamiliar with developing subtractive synths or frequency modulation-based machines. The idea to blend noise sources together to create a greater

depth to the variation of sounds produced by the synthesizers was inspired by a completely different design for an additive synthesizer that I encountered on the Max forums and subsequently made its way into the designs for the additive, subtractive and Karplus-Strong synthesizers. However, in the additive synthesizer this appeared more in the form of a selectable noise source per individual oscillator, while it functioned as a 'blender' like feature in the Karplus-Strong synthesizer where more than one noise source could be active at one time. This is due to the design of the synths, where all the noise sources in the Karplus-Strong synth were streamlined into one output and were the main ways in which the sound could be shaped, in addition to a function object acting as an ADSR envelope to control the shape of the sound produced. Constructing the subtractive synthesizer was the one that presented the most difficulty for two reasons; my unfamiliarity with the system, and the fact that I struggled to find any sort of guidance or instruction on how to construct anything other than a basic implementation of the synthesizer. This led me to using techniques that featured in other synthesizers to increase the range of sound produced such as an increased number of filters at the expense of computational cost, and the ability to change the noise source as mentioned above. The construction of the Karplus-Strong synthesizer for this project was based on a version of the synthesizer that I had previously constructed for another project and expanded on for increased functionality, such as the ability to change the length of notes, refinement with how different noise sources would work together to produce more than just plucked guitar tones as well as reusing an old algorithmic composition function to help refine the sounds while blending together for real-time feedback while balancing the often delicate selection of sound sources to get the most suitable sound before recording the samples. The Karplus-Strong synthesizer is perhaps my favourite out of the synthesizers I have produced purely due to quality of the samples the synth can produce, although restrictive in terms of the variety of sounds that can be produced, and the fact that it takes the field of physical modelling synthesis and combines it with more traditional electronic sound synthesis in a way that I have not encountered, at least in my own research. The frequency modulation synthesizer was perhaps the most basic of the four synthesizers in its final construction

due to the fact that it did not need to be particularly advanced to produce diverse samples like the other synthesizers, as more modulators can easily be added to increase the flexibility and realism of the sound, to an extent at the cost of computational efficiency. However, realism is not necessarily the goal of this particular implementation and the number of modulators that can be added before computational performance really starts to suffer is quite considerable. Overall, the process of designing and constructing the synthesizers was fairly straightforward, if time consuming. This is due to the fact that each synthesizer is based off an existing framework provided by the community or my own knowledge with alterations made by myself for the sake of diversity of sound samples produced and ease of use for future users of the NSynth. The reason that this step of the project was time consuming compared to other preparation aspects is that producing high quality synthesizers is no simple feat and troubleshooting issues with no real guidance except forums and prior knowledge can take time especially in software such as Max where it is easy to make small mistakes when dealing with particularly large and complex patches. For example, during one stage I encountered an issue where the subtractive synthesizer simply stopped producing sound, which was due to a simple issue with the gain control being incorrectly wired up and subsequently the signal for the whole patch was not being properly sent to the audio output. Other than minor issues such as the few discussed, this stage of the project went well and was completed during the timeline.

The next key stage of the project preparation phase was understanding how the processes behind the NSynth work in order to better use the software. As discussed above, the NSynth draws on a huge dataset of audio samples to recognise sonic characteristics of nearly any sound that is introduced to the machine (including my own samples which either resemble traditional synthesizers or real instruments) due to the developers desire to “develop a creative tool for musicians and also provide a new challenge for the machine learning community to galvanize research in generative models for music.” (Magenta Labs, 2017). The data set consists of more than 300,000 notes from over a thousand instruments so it was safe to assume that my methods of electronic sound synthesis would at least be somewhat recognised by the dataset in terms of

similarities to other instruments and processed in a coherent manner. If the results produced by the data set were not satisfactory, the workaround for this would have been to train an iteration of the NSynth myself with the samples I have produced. However, this would have dramatically increased the scope of my project and would not have been realistically possible to achieve the level of depth of training an entire neural network to produce results on a suitably sophisticated level during the time period. The acoustic qualities from learned instruments in the NSynth will pick up, for example, the guitar like qualities of the Karplus-Strong synth and the standard electronic synthesiser resembling sounds of the additive and subtractive synthesizers. Compared to other machine learning training data sets that focus on single objects such as pictures or features of an object such as words or symbols (Tensorflow, 2017) , the machine learning algorithm behind the NSynth focuses on single notes at a time, which will be further explained when discussing how samples are produced for the project. NSynth works in the same way as any machine learning network when learning from samples in the respect that repetition and recognising characteristics from very similar but slightly different sets of data means that there is a margin of error where the algorithm can recognise more ambiguous and hard to define samples, which is perfect for my own synthesizers which may still resemble other instruments to human ears, machines may struggle to make the distinction. The NSynth prepares data through a process called temporal embedding (Magenta, 2017). Temporal embedding is the context of isolating abstract behaviours and learned behaviours to predict future occurrences and behaviours to allow a neural network to better respond to input in the future (Liu, et al, 2015). In the context of audio samples, the temporal embedding process essentially unravels the structure of the data to reveal characteristics by breaking each sample down into thousands of tiny samples and isolating these features, replicating them and reorganising them into something new. These wavenet representations of the sounds are familiar and recognisable as the original but are uniquely different from the source sound. With the NSynth, two sounds are combined during this deconstruction and reconstruction phase and the new sound is the product of this process, where individual characteristics of the original sound can be isolated but clearly a new sound is the

main focus. The temporal embedding process is described as being similar to a nonlinear infinite impulse response filter, as the filter is currently limited to several thousand samples a second to capture the intricate details of the samples, a form of external signal to guide the process beyond the scope of a few thousand samples at a time for larger samples is needed. Magenta Labs resolved this issue by including a wavenet-style auto encoder in the temporal embedding algorithm so it was capable of learning its own temporal embeddings without external guidance, streamlining the entire process and making it simpler for other users to use the technology to carry out the process on their own samples. The samples go through thirty layers of computation, which results in the creation of a temporal embedding consisting of 16 dimensions for every 512 samples, capturing a great deal of depth and character from the original sample input. Magenta state that this can be viewed as a 32x compression of the original data then a subsequent unpacking for the reconstruction phase.

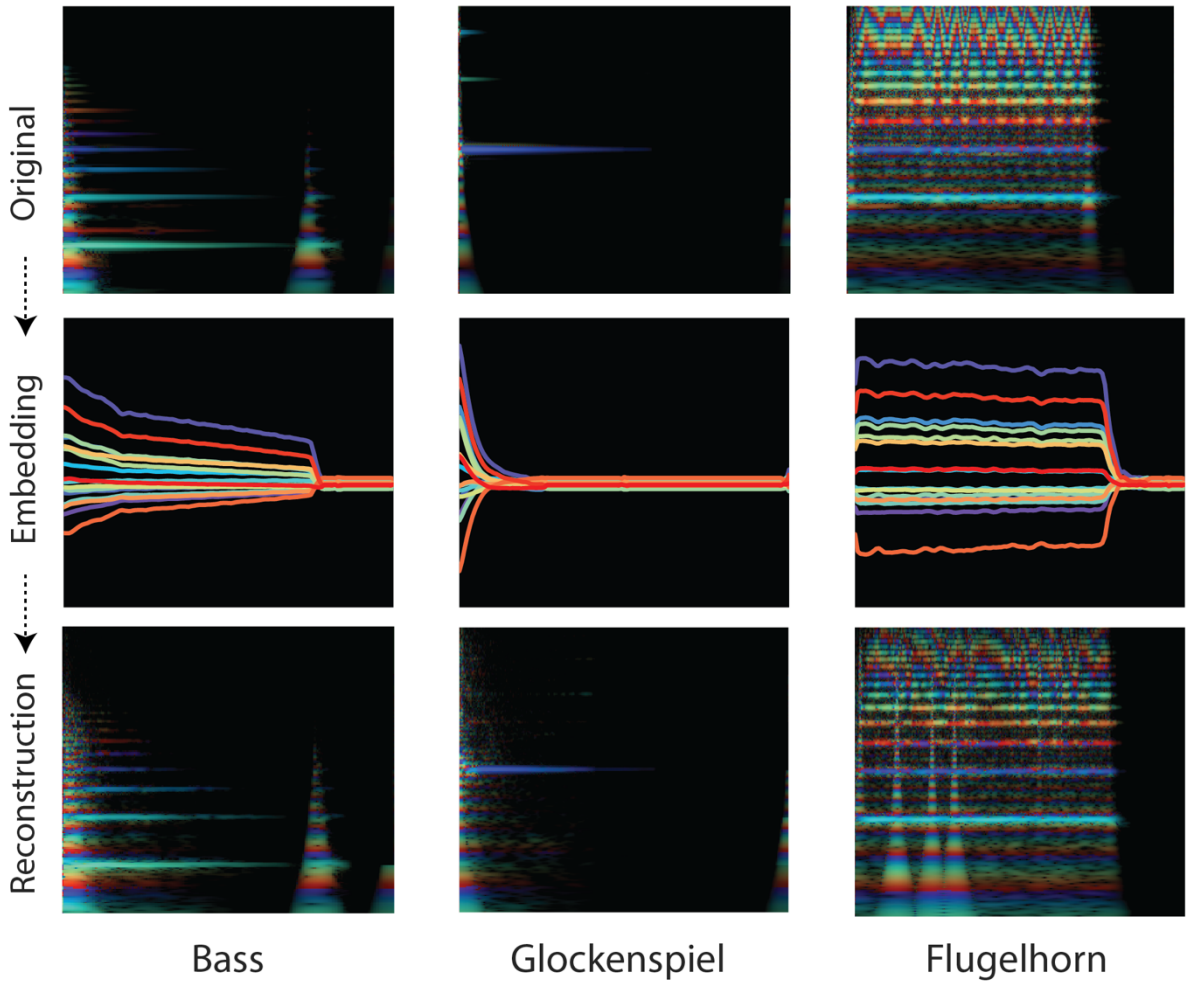

Fig 4 (Magenta Labs, 2017)

The rainbowgram (Fig 4). represents the original sample of three instruments before being processed by the WaveNet encoder, during the encoder and the subsequent reconstruction. The graph makes it fairly clear that reconstructed samples produced by the temporal embedding process very closely resemble the original sample in terms of audio quality with a few key distinctions, and audio examples provided by Magenta reinforce this point. (Magenta Wavenet, 2017). While the wavenet can only capture sound samples in a local context, this is suitable for the project as long as the samples that are output at the end of the process are able to respond to MIDI signals and used as composition tools either through the NSynth or other software to produce music. The model was tested at producing both single note sequences, as well as longer sequences of several notes which only reinforces the strength of the training dataset and the capability of the temporal embedding process, perhaps due to the miniscule size at which the sound samples are processed and reconstructed. Understanding this process paid off in the long term of the project preparation as it provided me with insight as to how the process was particularly adept at highlighting unique sonic features in the sound and gave me a better idea of what kind of samples to produce.

As the process of encoding and decoding audio samples in the NSynth relies so heavily on WaveNet, I felt it was important to conduct a brief research survey into WaveNet itself. WaveNet is a 'deep generative model of raw audio waveforms' (Deepmind, 2017). WaveNet was originally developed as a deep neural network that could generate speech that more closely resembles existing methods of text-to-speech sound synthesis, such as concatenative, which synthesises the structures of shapes and vowels to essentially patch together spoken words and speech (Mustapha, 2016). DeepMind demonstrated that the same neural network could also be used to process other digital audio signals for music and use the same training algorithm to produce realistic piano pieces with striking resemblance to the compositions the network was trained on. WaveNet is responsible for the element of the temporal embedding process that breaks the raw audio down to the sample level

due to the same reasons as the former mentioned; raw audio proves too difficult to work with due to the rate at which it ticks over, typically 16,000 samples a second. Compared to other methods of synthesized speech and real human speech, the WaveNet proved to be much more effective than its predecessors in accurately synthesizing speech (DeepMind, 2016).

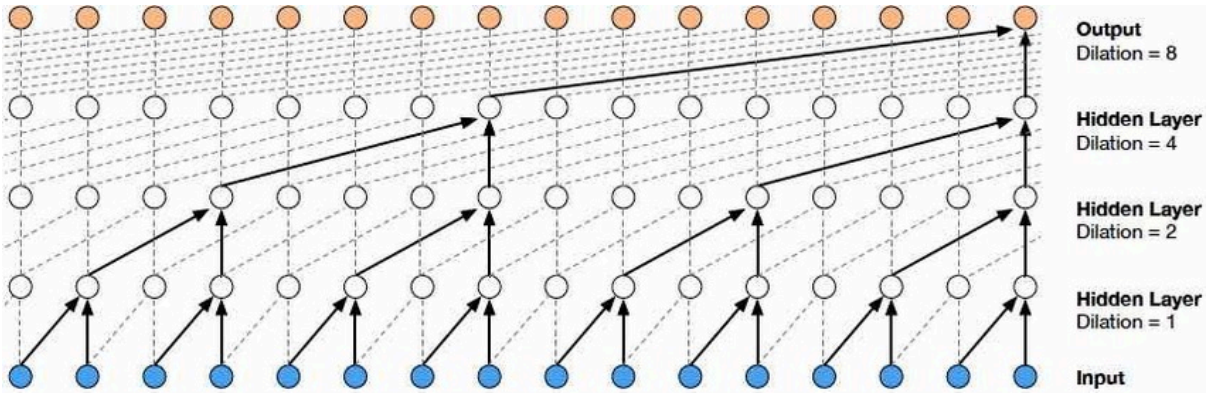

Fig. 5 (Tensorflow, 2017)

The structure of the Wavenet (Fig 5) is made up of several layers of data. There are several layers in this implementation of a convolutional neural network (Habibi; et. al 2017) where each layer has several dilation filters that split up data and allow its receptive area to grow exponentially with every layer and extract detail from thousands of time steps into which the data is separated.

This only reinforces the possibilities of the same technology when used to make the NSynth function, only reinforcing the systems strength as a tool of effectively producing usable synthesizers for composition. WaveNet was also proven to be capable of reproducing organic elements of human speech such as breathing and mouth movements, demonstrating the amount of sonic detail the process manages to capture and reproduce. The musical demonstrations of the WaveNet system independently of the NSynth warrant discussion at a later time due to the fascinating implications that training AI to reproduce the music of composers, perhaps long dead from the classical and baroque eras is an interesting research discussion of its own. Conducting this research allowed me to much better understand the workings of the NSynth and WaveNet, specifically with how the

system treats data and the quality of synthesizer needed in order to best utilise the attention to sonic characteristics and detail that the NSynth and temporal embedding process is capable of.

3.5 Testing and Producing Samples

As described in the Magenta Labs help files for the operation and expansion of the NSynth, there was a predefined process to carry out on the samples to make them compatible for use with the NSynth. For clarity, I will explain the process below with a flow diagram, and spectrograms of the input vs output samples.

Following the instructions, a set of 16 samples from each synthesizer had to be recorded directly from the instrument, each four seconds in length for the sake of consistency within the mixture of the resynthesized result.

The first step in the process was to record the samples from the synthesizer. Due to the nature of synthesizers built in Max, the easiest way to record the samples was to use a recorder from inside the Max patch itself, which could be programmed for length. This was important as each sample had to be the same length in time due to the encoding process of the NSynth. Sixteen samples per synthesizer had to be recorded, ranging from MIDI value 24 (C2) to MIDI value 84 (C7).

The next step in the process was to prepare the files to work with the python code.

In order to do this, the process started with the need to properly organise the files in a directory so the python code could recognise them. The python code used in the process was provided by the user instructions in the magenta NSynth files and consisted of a number of different parameters that would recognise specific markers in the file name of the samples produced in the previous stage. For example, ‘additivesynth_24.wav’ would be the sample for the additive synthesizer for the note MIDI value 24, C2.

Once the directory for the samples was prepared, the python code was used to produce the batch of samples. The output was produced in the form of a number of combined instrument samples, with individual samples of combinations of instruments, with note equivalents ranging from C2 to C7.

Once the samples were produced, the next step was to produce a spectrogram of the input sample vs the output sample to compare sonic similarities and use this information, as well as subjective opinions from listening, to the samples to decide if the input sample had to be tweaked to produce higher quality and more usable results. In the final iteration of my samples, this process had to be repeated around four times to produce output samples I felt were suitable for the composition I aspired to produce. Below is a flowchart explaining the process followed to produce the samples, as well as spectrograms comparing the samples from the synthesizers before the NSynth, and the subsequent output.

One final optional step that I performed for the sake of achieving a certain aesthetic quality was to re-record the output samples through a dynamic microphone to achieve a sense of 'room space' or reverb that would often be present in traditional instrument recordings. This aesthetic characteristic is an aspect that is often present in my own practice as a musician and is not entirely essential to the process but this falls in with the suggestion for the user to experiment with any one of the systems present in this research project to achieve whatever sonic characteristics they may wish to utilise.

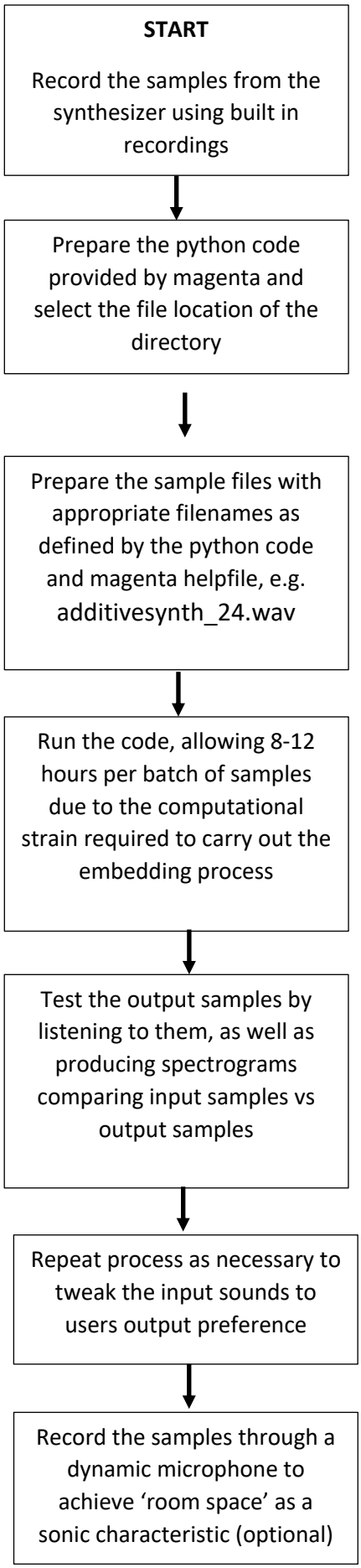
START
Record the samples from the synthesizer using built in recordings
Prepare the python code provided by magenta and select the file location of the directory
Prepare the sample files with appropriate filenames as defined by the python code and magenta helpfile, e.g. additivesynth_24.wav
Run the code, allowing 8-12 hours per batch of samples due to the computational strain required to carry out the embedding process
Test the output samples by listening to them, as well as producing spectrograms comparing input samples vs output samples
Repeat process as necessary to tweak the input sounds to users output preference
Record the samples through a dynamic microphone to achieve ‘room space’ as a sonic characteristic (optional)

Fig. 6 (Flowchart of sample production process)

Below are spectrograms of the input samples for the Karplus-Strong synthesizer and the additive synthesizer, compared to the subsequent output from a test run of a batch of two samples.

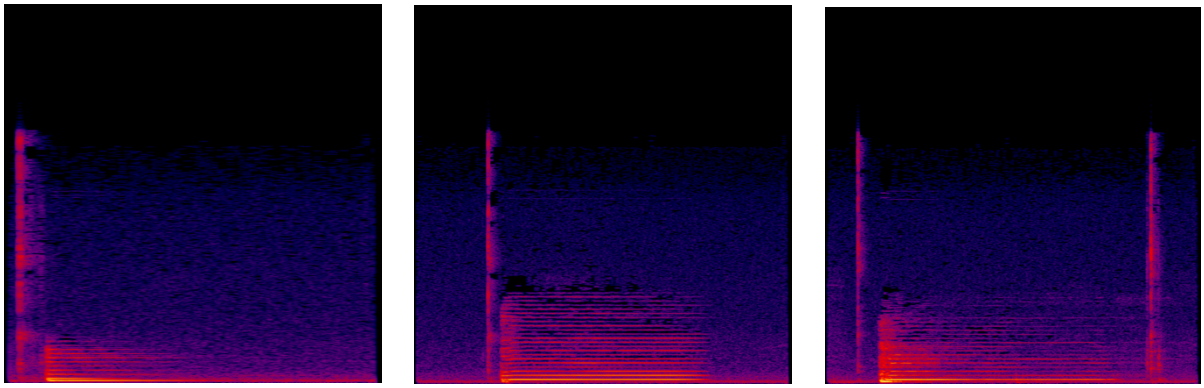

Fig 7. (Additive Synth input spectrogram, Karplus-Strong input spectrogram, NSynth output spectrogram)

As shown in the spectrograms above, characteristics present in both the input samples are present in the output, which provides evidence that sonic characteristics of the input synthesizers are still recognisable in the resulting NSynth output. The plucking sound of the Karplus-Strong synthesizer in *Fig.6* and *Fig. 8* is noticeable in the output sample spectrogram, as is the decay of the note in the additive synth sample. There is also an additional loud source of noise in the output sample as represented by the sharp line in *Fig. 8,* which is likely just distortion of the sound from the embedding process which gives sounds a grainy texture of their own, which is just a result of the process found in most of the samples produced for the this project. This reinforces the importance of the final step in the flowchart, which instructs the user to repeat the process as often as required to ensure the output sample represents the aspects of the combined sounds to a level suitable for their compositional requirements.

3.6 Composition and Inspiration

Electronic sound synthesizers were originally created with the end goal of being used to produce music, as was the NSynth. Furthermore, musicians and researchers have always been at the forefront of new technological developments to find a way to use emerging technology and theories to produce new and original compositions and methods of enhancing music. To this end, it was clear that the best way to demonstrate my own thesis project was with a composition. To draw inspiration from existing computer music applications of technology, I studied two distinct approaches to the concept. The first was *Stria* by John Chowning, and the second was *Switched on Bach* by Wendy Carlos. *Stria* is described (Zattra, 2016) as a key milestone in the history of computer music. The piece is fully generated with synthesizers, specifically frequency modulation, and is famous for being one of the first pieces to fully utilise this method of synthesis as well as the Golden Mean Ratio, which is a mathematical phenomenon where if two quantities are in the same ratio of their sum to the larger of the two quantities (Dunlap, 1997). Chowning's breakthrough in the FM synthesis technique allowed him to create synthesized 'metal striking' sounds and bell like sounds, which bore striking resemblance to physical real-world percussion. After six years of development with the algorithm which he engineered to be capable of synthesizing many instruments with varying degrees of complexity, including the human voice (Chowning, 1973). The structure of Stria itself was based on the mathematical properties of the golden mean, where the length of an individual segment of generated sound was decided via the length of previous sections and the sum of the ratio of individual sections to share the ratio of the overall piece. The golden mean was considered historically to be the representation of physical perfection displayed by natural examples such as the Fibonacci spiral (Lucas, 1891), which is presumably what Chowning was aiming for when he programmed the algorithms to produce the piece. However, for the sake of this project, I was more interested in the sonic properties of the piece and how resynthesized sounds could be used to achieve similar sounds, while the structure of the composition would follow something far more musically conventional as this is my area of strength as a songwriter and composer myself. In Stria,

the properties for every sound featured in the piece were generated by a source algorithm starting from the same globally referenced FM synth for the piece. Each sound was defined by approximately 30 parameters including begin time, duration of attack, carrier frequency, ADSR control etc.) which allowed for a wider range of sounds to be automatously produced by the algorithms due to the sheer number of parameters that could be altered, similar to my own synthesizers which could be heavily tweaked to alter the sound produced by the algorithm (Baudouin, 2007). Chowning gained control of this huge number of parameters to shape the sound by applying individual envelope generators to every oscillator which allowed control of each element of the process on an individual level rather than having to tweak the entire algorithm which must have dramatically reduced time limitations of the original composition while maintaining flexibility of sound. This range of sound was even expanded further by adding alternate envelope generators to each oscillator with a different set of controllable parameters. The alternate envelope generator was historically used in the climax of the piece to produce a 'shhh-boom' sound as described by Chowning, which was achieved by a step variation of the original carrier amplitude (Dahan, 2007). Overall, the sonic characteristics of *Stria* are so unique and such a distinctive showcase of the capabilities of unique synthesized sounds to produce composition, studying this piece was extremely helpful in understanding the scope of music that could be composed using non-conventional computer music synthesis methods such as the NSynth. (Meneghini, 2007).

Switched on Bach was composed as a counter argument to many computer music research compositions of the time produced by avant-garde composers regarded by Wendy Carlos as "ugly music" in an effort to compose "appealing music you could really listen to" (Carlos, 1999). Produced entirely with synthesizers, the project was a collection of renditions of classical Bach pieces reimagined in a new style with emerging technologies. This use of emerging technology at the time to bring a change to existing musical tradition and use of technology really resonated with me in conducting research into the history of composing with computer music to discover how to produce my own compositions with the NSynth. Switched on Bach was produced with a monophonic

synthesizer, meaning that each note had to be released before the next one could be played, resulting in a disconnected and unnatural sound common to monophonic synths. To overcome this, Carlos had to record each note one at a time which must have been a particularly arduous process for the technology of the time, especially on an unreliable synthesizer that "often went out of tune" (Miller, 2004). Upon listening to *Switched on Bach* in its entirety, the aspect of the compositions that stood out to me most was the way in which synthesizers were used to emulate the range of instruments featured in the original composition, particularly the way in which woodwind instruments were cheerfully recreated by whistle-like tones and the range of sounds produced by just one synthesizer (Carlos, 1968). In the piece '*Water Music Suite No. 2 in D*', the use of the synthesizer to recreate semi authentic horn sounds particularly resonated with me as a good example of the types of samples to produce to yield valid compositional results from the NSynth algorithm. To this end, it was clear that having a good intention of the types of samples I intended to produce with the NSynth was important to producing the best possible compositions with the results. This is in line with any endeavour into producing music with electronic synthesized sounds and was undoubtedly what took place with *Switched on Bach* in order to faithfully recreate the compositions.

Studying these two distinctly different approaches to using emerging computer music technology to compose music has allowed me to gain an insight into the process for my own composition, combined with my own background as a music producer. *Stria's* use of interesting and abnormal synthesized sounds and focus on individual sonic characteristics of a sound to produce a truly unique piece in addition to the foresight and planning of sounds used to produce a faithful but cutting edge demonstration of synthesis of *Switched on Bach* left me far more prepared for my compositional attempt with my produced sounds. As my own background as a writer of music lies primarily within the genres of indie and rock, producing a piece of music in a new style with unfamiliar instruments

certainly presented a challenge, although this challenge was overcome with the inspiration of recreating existing pieces to an extent, not dissimilar to the work of Wendy Carlos. For my own composition, I decided to use the synthesizers to create a piece inspired by of one of my personal favourite pieces of music. The piece of music in question is the *Exogenesis Symphony Pt. 3 (Redemption)* by the British rock band *Muse.* The piece itself is a fusion of classical and rock styles in the form of a 3-part symphonic suite and offered plenty of variety in instruments to demonstrate my own synthesized sounds. I chose this piece with the theory that I would be more comfortable composing with an element of the style I am experienced in especially considering the unfamiliarity of the instruments and would therefore produce a much better piece to demonstrate the strengths of the project. As for the instruments within the original piece, piano, strings, percussion and vocals heavily featured. In terms of producing samples to fit this instrument selection, I decided that a resynthesized sound with prominent features of the Karplus-Strong algorithm would best be suited to recreate the short but flowing notes of the introduction to the piece, as the sustain offered by the original algorithm in addition to the drone-like textures produced by some of the additive synth samples would accurately recreate this instrument that would continue to be a prominent part of the piece. As for the string sections, any other blend of synths would work as long as the timbre was distinctly different from the Karplus-Strong focused blend. As for the process of composing the piece, I realised that the most efficient method was to find a way of mapping the sets of resynthesized sounds to a MIDI keyboard that could be used with Logic Pro X as a production suite. Software such as Soundflower was capable of connecting MIDI output from Max to software such as Logic or Ableton. The composition took the form of a blend of the '*Exogenesis Symphony'* as an arrangement using some of the resynthesized samples, as well as original music with a wider range of samples inspired by Carlos and Chowning to demonstrate the capabilities of the results of the project. The shape of the composition was directed mainly by the need to demonstrate the capabilities of the NSynth samples, hence the wide variety of sonorities demonstrated in the piece

which resulted in a ten-minute-long piece that seems to change drastically throughout in terms of theme, dynamics and momentum.

3.7 Project Changes and Refinement

Throughout the course of the project, there were several changes made to the project for various reasons, as well as refinements to control the scope of the project to fit within the time constraints. In this section I will discuss how these project changes affected the overall process and why the changes were vital to ensure the success of the project. Firstly, the biggest change to the project in my own opinion was the decision to drastically reduce the number of synthesizers from sixteen in the original inception of the project (as dictated by the NSynth sample preparation documentation) to just four sets of synthesizers. The reason that this was necessary was primarily a time constraint concern, although focusing on fewer synthesizers allowed me to focus much more on individual synthesizers and making them as advanced and diverse in terms of sounds produced as possible, rather than having sixteen half-baked and indistinguishable from one another synthesizers. As a result, each of the four synthesizers were able to produce many more sounds for future use of the project, and is much more manageable to control four instruments in a complex mix rather than sixteen, especially in a proof-of-concept project aimed at composition where simplicity is valued over extreme complexity to the point of convolution. Another change that was made to the project was the decision not to produce a wider set of samples for potential users to play with. Not only was this a time constraint conscious decision, I felt that leaving users to their own devices when producing samples with the instruments better allows them to express their creativity, so detailed instructions on how to use the code and synths to produce new samples would be provided instead.

Originally, I intended to discuss digital convolution as a synthesis method which is also capable of combining sounds for resynthesis to an extent, although not to the depth and timbre of the sounds that the NSynth produces. I chose to omit this from the project fairly early on into the timeline as I felt that discussing a method of synthesis so close to the project I was trying to achieve would

confuse results and undermine the work with the NSynth, and invalidate the whole artificial intelligence and machine learning element of the project as a result. Even though the synthesis method was nowhere near capable of producing similar results, it simply did not require inclusion in the project past a simple passing mention during the research survey section. The final project refinement was to add detailed instructions on how to use the provided tools to make samples and use the NSynth rather than including a wider bank of pre-prepared samples. As mentioned above I felt that providing a larger range of samples might limit creative freedom but it also discouraged users from going through the same learning process I underwent myself when understanding what kind of results each different synthesizer would produce while used for producing samples and with the temporal embedding process. This learning process really informed my decision making when composing with the new sounds as I had a much better idea of the sonic qualities that would feature heavily in the piece. Overall, these changes helped the project feel a lot more concise and be much more manageable within the time scale provided.

Rationale

In this section, I will justify decisions made throughout the practical elements of the project, how research led me to make these decisions and the impact that said choices had on the thesis project from an objective standpoint.

As stated previously, the decision to use the synthesizers were based on four key criteria; ease of use, computational efficiency, range of sounds produced and how easy it was to construct my own version of it. These criteria were selected carefully due to a few key reasons, and these decisions had a huge impact on the outcome of my project. Firstly, one of the main aims of the project from the inception was to create a tool that could be used by others for composition and experimentation once the project was completed, and to that end, the synthesizers had to be easy to use so the user could produce their own samples and modify the synthesizers enough to suit their own needs. The

effect that this decision had on my project was that none of the synths became overly complicated and could easily be used to produce a moderate range of sounds, although this sacrificed the ability to create perhaps more complex individual sounds that could have been focused on if time constraints were not an issue. In addition to being easy to use, the synthesizers also had to be capable of producing more than just one simple sound, with the aim of experimentation in mind. To that end, range of sounds produced by the chosen synthesizers was also an extremely important criteria that the synths had to meet. The decision to include this as a criterion affected the project in that a delicate balance between finding synths that were easy to use while remaining computationally efficient and still possess the ability to produce an acceptable range of sounds was formed. Fortunately, simpler methods of synthesis such as FM synthesis and additive synthesis met these criteria, while approaches such as Karplus-Strong met fewer criteria but proved useful due to their unique sonic characteristics, which was more of an aesthetic decision to the project rather than an academically justified one. Synthesizers also had to be computationally efficient and the decision to include this as one of the requirements for the instruments is purely a practical one, due to the fact that I was producing the synths on a system with limited hardware capability and a very small budget. Without this as a consideration, the project would not have been possible at all due to simply not being able to use the programs if they were too computationally demanding. Needless to say, this was an easily justifiable decision to make, however I was fortunate enough that all of the synthesizers selected for the other criteria were manageable from a computational standpoint if kept moderately simple and not much further research into more efficient methods was needed. The final criteria related to how easy it was to produce my own version of the synthesizer in terms of the process of using Max to construct the synthesizer, and the decision made to include this was to keep the project scope manageable and to make sure that the process was completed during the allotted slot of the project timeline.

Carrying out initial experiments into using the NSynth with my own samples was an important part of my rationale to prove that the project was technically feasible in the first place. The decision to produce a small number of demonstration samples with my implementations of the Karplus-Strong and additive synthesisers was made because both synthesizers offered different sonic characteristics in quite a drastic manner so individual aspects such as plucked-string-esque timbres could easily be recognised in the newly created sample sound. This allowed me to distinguish from the experiments output to determine if a result that could still be recognised as having characteristics of the original sound and could be used in a similar manner to the original sound in terms of composition. This had a massive impact on the project in the respect that a sample containing sonic characteristics could be used in the same way that a guitar could be used in a composition, but distinct enough from the original instrument to be considered something unique and valid as a creative tool. The rationale behind the testing phase of this project was to determine if valid sounds could be produced and this was proved by the initial experiments.

The decision to use composition as the presentation method for the research was made very early on in the process due to the fact that with sound and music related technology, there is no better way to demonstrate the research than producing a piece of music to show the output in a practical way, rather than an abstract or theoretical explanations or demonstrations. Composition as a chosen method also complimented my own practice as a musician and composer, so this was easy to justify even if the composition method from the project was unconventional. However, as mentioned previously, this just encouraged experimentation to produce new music with new tools.

During the process of producing the samples with the NSynth, I had the choice between using the existing dataset of the NSynth to recognise characteristics of sounds based on three million samples the system had already been trained on, or to train the system on my own dataset. The reason I

chose to use the existing dataset is due to the fact that the latter would have been an extremely time-consuming process and could not guarantee the same standard of result as magenta's training data set. The impact that this had on my research was that time and practicality was favoured over a data set that may have been more bespoke to the specific instruments it was being used with, although no tests were carried out to confirm this so it cannot be stated whether or not a specific dataset would have produced better results.

The decision to produce a test batch of samples was made to discern whether or not the project was viable and could be continued, and the decision to produce a test batch of samples led to a testing phase that was initially unsuccessful due to a lack of access to technology with sufficient computational power and the correct operating system to do so. This led me to work on other aspects of the project until a unix capable machine could be acquired, which had the impact on the project of having to slightly adjust the timeline to ensure valuable time wasn't wasted by bringing the project to a halt. During this time, I emphasised my focus on research into inspiration for the composition aspect of the project. The outcome of this decision on my project was not too serious, however, and allowed more time to refine the plan for the composition once the final batch of samples had been produced. The rationale behind this decision was to test the system before the project was too far down the timeline to adapt and change it to overcome difficulties that may have been encountered further down the project.

This next paragraph explains and justifies the decisions I made during the planning of the composition and not the actual composing process itself. For the composition, I wanted to include inspiration from works of computer music that are related to the early work of electronic sound synthesis as well as inspiration from my own practice as a musician. From a self-taught musical background, my early days as a musician were defined by bands such as Muse, The Arctic Monkeys

and Jimi Hendrix. As a guitarist, I wanted to select a compositional style and feel that matched my own usual approach to composition with synths such as Karplus-Strong with clear inspiration from works such as *Stria* and *Switched On Bach*, as mentioned above. This led to the idea to produce a recreation of an existing piece based in my own musical background with elements of influence from computer music associated genres and works. The impact that this had on my project was the outcome of a well-researched and academically justified composition idea with aesthetic justifications rooted in my own musical practice.

During the composition process, the decision not to turn the project into a full-blown sample player made the composition much more difficult. As explained previously, the composition process was difficult due to the nature of the samples and the fact that they could only be arranged by essentially dragging and dropping them into Logic, which was an extremely time-consuming process. This would have negatively impacted the project timeline; however, I was fortunately ahead of schedule and had extra time to ensure the composition was given full attention. General issues included difficulty lining up the samples to be in time due to the nature of the output of the NSynth where different length of samples had to be trimmed down to size, however issues like this were overcome with patience and attention to detail and had no result on the final end result of the composition process.

4. Summary and Conclusion

The purpose of this section is to summarise and conclude the findings and results of this research thesis project. The main points discussed will be the research outcomes and main achievements of the project as well as primary issues faced throughout the process. Alterations that also occurred throughout the process of the project for various reasons and the implications of these changes will also be reviewed and summarised. Finally, the potential for further research into the fields of AI and sound synthesis using the findings of this project and further developments of the project within this thesis will be discussed.

4.1 Research Outcomes and Main Learning Achievements

Overall, this thesis project served to demonstrate that an overlap between artificial intelligence-based systems and specific ally designed electronic sound synthesis could produce sounds suitable enough to be used for a composition inspired by the works of previous computer music artists such as John Chowning and Wendy Carlos. The resulting sounds produced by the synthesizers once the temporal embedding process has taken place were used to produce a composition, and instructions on how to follow the same process to produce sounds with the aforementioned synthesizers were also provided so further compositions could be produced. The aim of this thesis project was to use the Magenta Labs Neural Synthesizer (NSynth) along with electronic synthesizers specifically designed and tuned based on their sonic characteristics to produce something resembling a specific style of composition with historical and contextual inspiration. One of the main learning achievements of this project, in my opinion, was producing the sounds that could be used to create a composition from particularly difficult to use instrument and create a viable, if slightly unusual composition. In addition to the composition, it also proved that there are far more applications for musical production by individuals far more skilled than me in producing electronic music to use their own synthesis implementations and sounds to create and compose with the NSynth. Research conducted into the NSynth during the project process as well as experimentation with the

instrument and the bespoke synthesizers found that certain combinations work better than others, and designing specific synthesizers to fill a certain role within a planned piece when combined together was a valid and methodical approach to composing with these tools, in my experience. However, this does not mean that it is impossible to combine random sounds from the synthesizer and produce a perfectly valid composition in any style or use the sounds to compliment other methods of composition including live recordings or MIDI pieces. The process of editing the result of a sound includes starting from the synthesizers and then repeating the temporal embedding process which proved frustrating at times and probably makes the latter method of composition more attractive to potential users, although the project proves that it is definitely possible to use AI and synthesis to produce music. The aim of the project was to conduct 'an investigation into the uses of artificial intelligence and machine learning for electronic sound synthesis'. Overall, this investigation culminated in a demonstrative composition, reviews of two different fields of study related to computer music and development of the research concept, leaving the potential for further research.

4.2 Primary Issues Faced

Despite the overall success of the project, there were a number of hurdles and issues faced throughout the process. I will discuss these issues one by one and how they affected the project, and alterations that were made as a result of these changes and the affect on the overall project as a result of this.

One of the issues faced early on in the project was finding a computer system suitable enough to carry out the temporal embedding process, which caused significant setbacks in the project timeline and nearly even forced the project idea to be reconsidered in favour of focusing purely on the theoretical ways in which AI and electronic sound synthesis could be used together, or instead focusing on resynthesis through digital convolution and abandoning the artificial intelligence aspect

altogether. For various reasons related to the way in which windows operating systems function, the temporal embedding process carried out in Python did not work on the computer systems I had available to me, so to resolve this issue I was eventually able to gain access to a UNIX operating system that the temporal embedding process could be carried out on. This caused project delays due to financial implications involved in the process that were eventually resolved and eventually finished, and the delay was adapted to by simply focusing on other areas of the project in the meantime. The rationale for addressing this issue was fairly straightforward in that it was a more practical solution to adapt the technology used in the project rather than massively changing the contents when much of the research and work had already been carried out. The implication of this decision is that there were delays that ultimately affected the time left available to produce the final composition, where I had intended to experiment more with different sounds produced by the algorithm, although the instruments available to me for composition were still adequate enough to provide a suitable proof-of-concept composition for the thesis project.

Another primary issue faced was the underestimated difficulty of producing a composition that I as a composer was satisfied with, due to both technological reasons and compositional reasons. Firstly, technological limitations of the implementation of the NSynth provided by Magenta Labs as a Max device meant that acquiring sounds and controlling the blend of sounds was easy, but there was no real way to import this output into composition and music production software such as Logic Pro X, except software such as Soundflower which merely records from the instrument which was difficult to use in terms of arranging a piece and editing segments as the output as one singular audio file, as if recorded with a microphone. To overcome this issue, I simply recorded the audio directly from my studio monitors with a microphone and played the desired notes within the instrument, then trimmed these segments and arranged them into the piece as necessary. This workaround was complicated and perhaps with more time it would have been possible to use the software such as sound flower to enable the instrument to be used for an entire composition more smoothly, but the project timeline meant that this workaround prevented prohibitive issues from holding back the

research project any more than it already had. Despite this time-consuming workaround, the quality of the output samples from the NSynth still proved to be usable for composition. Although the technological issues with the composition process were eventually resolved, using the sounds to produce a composition presented their own set of issues. The way in which the samples had to be imported into the logic file meant that each note had to be individually arranged into the piece with no way of using the MIDI keyboard as an instrument with which to compose. It was still possible to produce a composition this way, though it was time consuming and unappealing to other musicians as a compositional tool. This meant that once the samples had been recorded from the NSynth instrument in the process detailed above, there was no way of lengthening the notes short of looping them which involved changing huge aspects of the compositional plan. However, it was possible to shorten the notes by trimming them and using fades, equalisation and other mixing tools to remove harsher aspects which resulted from this lack of note control. If it were not for the limited time available to complete the composition, I would have created a sample player in Max with variable note length into which the samples could be loaded, which would have, in theory, functioned well as a compositional tool. This issue was overcome with time and patience in the compositional stage of the project, with potential to make this process easier with further development and research in the form of the sample player to act as a dedicated instrument. The decisions made to overcome these issues were partly aesthetic and partly practical. For example, the decision to use a dynamic microphone to re-record the samples directly from my studio monitors was an aesthetic choice, made in order to add the sonic characteristics of depth and more natural 'room' and reverb to the sound, and I found that this made the instruments take on a slightly different, less artificial sound which was an interesting approach to the final recording, coming from my own experience as a recording musician outside of this thesis project. The microphone used was a Shure SM57 which I often use myself for recording guitar amps and other stringed instruments worked particularly well with synths attempting to emulate strings in the composition, adding much of the same timbre that the equipment offers with similar live instruments. The impact of having to

manually organise individual sound samples definitely affected the scope of the final composition in the respect that it is an extremely irritating and impractical method of composition that may deter less experimental users from fully exploring the scope of the sounds made available from the project. However, in the same vein of the experimentation that came from cutting and splicing tape recordings, this can be viewed from an aesthetic standpoint as an unintentional homage to that, encouraging experimentation and composition that may not otherwise occur. This both negatively and positively affected the project, depending on how experimental the user wants to be. From my own perspective, I found it difficult to adapt to this particular method of composition, especially compared to my usual composition method of recording with live instruments and my own synthesizers, however I did find it interesting to take a new approach to using my own tools that I may not have intended when I first started the project. The decision not to make the instruments functional as a sort of sample player as mentioned above could have been justified with more time available and would have had an impact on the project in the sense that it would have been more easily accessible, but as mentioned previously, this can be justified by encouraging the user to adapt to the tools available like I, myself, had to.

4.3 Reflection and Further Research

Overall this research project was successful in demonstrating a new way in which electronic sound synthesis and AI can be combined for compositional purposes. Given more time, I would have liked to prepare the project as a more concise package with a proper method of use as a compositional tool rather than time consuming workarounds to make the process easier to follow for other composers and researchers. However, the guides provided by Magenta Labs proved simple enough to follow in order to allow others to use the synths to produce their own samples. Upon reflection, I feel that producing my own implementation of the NSynth would have broadened my understanding of the process behind the network even more than the research did, but that would have dramatically increased the amount of time taken to produce the final result and the results yielded

may not have been significantly different so dedicating valuable time to a learning exercise was not necessary. As a side note, part of my own research independent to this thesis project involved building a neural network, and skills learned from this process aided my research into the artificial intelligence aspect of the research. However, lacking my own implementation of the NSynth did not diminish the completed work as the research aim was to conduct an investigation into ways in which the two fields are used in tandem to produce a musical output and composition through AI and synthesis was simply the way in which I chose to demonstrate this connection.

Although I have briefly discussed the implementation of a sample player as a compositional aid to function with the synthesizer samples as a way of further developing the project, there are other ways in which I would further this research project, given the opportunity to keep developing it. It would have been interesting to expand the range of synthesizers beyond four implementations of basic synthesis methods and explored the sonic characteristics of electronic sound synthesis methods such as octave synths or granular synthesis, and discover new compositional avenues offered by these new possibilities. It would also have been interesting to further explore the results of combining, for example, physical modelling synthesis methods and more conventional but obscure synthesis methods. This was briefly explored in the project with the combination of the Karplus-Strong synthesis combined with the others and produced perhaps some of the most interesting sound samples which I was particularly fond of utilising during the creation of the composition. The issues encountered and the potential for expansion within this finished research project does not detract from the strides I have made with my own knowledge within the fields of artificial intelligence and electronic sound synthesis, and since the start of the ResM computer music course, and I look forward to experimenting further with this research in my own time. With the completion of this composition and the research project overall, this concludes the research project and production of the means to produce further work.

References

Andresen, Uwe (1979), *A New Way in Sound Synthesis*, 62nd AES Convention (Brussels, Belgium), Audio Engineering Society (AES)

Andrew (1997), *A brief Introduction to MIDI* (Imperial College of Science Technology and Medicine)

Andrew W. (1996), *Reinforcement Learning: A Survey*. Journal of Artificial Intelligence Research. 4: 237–285

Baudouin, O., (2007). A Reconstruction of Stria. *." Computer Music Journal 31*, 3(3), pp.75-81.

Charles, D. and Thomas A, J (1997), *Computer Music: Synthesis, Composition and Performance*

Chowning, J (1973), The Synthesis of Complex Audio Spectra by Means of Frequency Modulation. *Journal of the Audio Engineering*, 7(21), pp.26-34.

Collins, K., 2008. *Game Sound*. Cambridge, MA: MIT Press.

D. Herremans, C.H., Chuan, E. Chew (2017), *A Functional Taxonomy of Music Generation Systems.* ACM Computing Surveys

Dahan, K (2007), *Surface Tensions: Dynamics of Stria*

Dean, J (2015). *Tensorflow: Large-Scale Machine Learning On Heterogeneous Distributed Systems*

Deng, L.; Yu, D. (2014), Deep Learning: Methods and Applications, *Foundations and Trends in Signal Processing*. 7 (3–4): 1–199

Dixon, S (2000), *On the Computer Recognition of Solo Piano Music* (Vienna, Austria) Austrian Research Institute for Artificial Intelligence

Dunlap, Richard A, (1997), *The Golden Ratio and Fibonacci Numbers*, World Scientific

Eck, D (2010) *Learning Features from Music Audio with Deep Belief Networks*

Eimert, Herbert (1972), "How Electronic Music Began", *Musical Times*, 113 (1550) (April):347–349.

Evans, G., Blackledge, J. and Yardley, P (2001) *Analytic Methods For Partial Differential Equations*

Fahy, F., 2004. *Advanced Applications in Acoustics, Noise and Vibration*

Forrest (1994), *The A-Z Of Analogue Synthesisers*. Devon: Susurreal Publishing

Habibi Aghdam, H. and Jahani Heravi, E (2017) *Guide To Convolutional Neural Network*

Hass, J (2017) *Introduction to Computer Music: Volume One*

Heideman, (1984) *Gauss and the history of the fast Fourier transform*

Hinton G (2009), Deep belief networks

Holmes, Thom (2008) *Early Computer Music*. *Electronic and experimental music: technology, music, and culture*

Jono Buchanan (2011), *Subtractive Synths explained*

Karlheinz Stockhausen Gesang Der Jünglinge. 1960. [film]

Karplus, Kevin; Strong, Alex (1983), Digital Synthesis of Plucked String and Drum Timbres. Computer Music Journal. MIT Press. 7 (2): 43–55.

Kent, Steven L. (2001), *The Ultimate History of Video Games: The Story Behind the Craze that Touched our Lives and Changed the World* (Roseville, California) Prima Publishing

Laura Zattra (2016) *Stria By John Chowning, The Ultimate Analysis*

Lemmetty, Sami (1999), *Phonetics and Theory of Speech Production*

Liu, J (2015), *Temporal Embedding in Convolutional Neural Networks for Robust Learning of Abstract Snippets*

Lorelei Koss (2017) *Visual arts, design, and differential equations*. Journal of Mathematics and the Arts 11:3, 29-158.

Lucas, Edouard (1891) *Theorie des nombres*

Manning, Peter (1993), *Computer and Electronic Music*

Marchand, Lagrange (2001), *Real-Time Additive Synthesis Of Sound By Taking Advantage Of Psychoacoustics*

Milano D (1975), Bob Moog: From Theremin to Synthesizer. Contemporary Keyboard

Miranda, E (2013), *Readings in Music and Artificial Intelligence*

Moore, E (1965), *Cramming more components onto integrated circuits*

Moorer, J (2008) *Signal Processing Aspects of Computer Music – A Survey*. *Computer Music Journal*, pp.4-37.

Mustapha, O (2016)*, Text-to-Speech Synthesis Using Concatenative Approach*

Nikol (2016), *An Introduction to Karplus-Strong Physical Modeling Synthesis*

Nilsson, Nils (1998), *Artificial Intelligence: A New Synthesis*

Osipenko (2019), *Towards Data Science*

Pacha, Alexander (2019), *Self-Learning Optical Music Recognition* (PhD). TU Wien, Austria

Palombini, Carlos (1993), Machine Songs V: Pierre Schaeffer: From Research into Noises to Experimental Music. *Computer Music Journal* 17, no. 3 14–19

Perrin, Robert, (2014), *Normal modes of a small gamelan gong.* The Journal of the Acoustical Society of America

Polotti (2008), *Sound To Sense, Sense To Sound*. (Berlin, Logos)

Publishing

Rebelo (2012), *Optical music recognition: state-of-the-art and open issues.* International Journal of Multimedia Information Retrieval, Vol. 1, Issue 3, 173-190

Reid (2000), *An Introduction To Additive Synthesis*

Rowe, R. (1992*), Interactive Music Systems: Machine Listening and Composing*. The MIT Press, Cambridge.

Russ, M (2009), *Sound Synthesis and Sampling*

Sarah (2015), *Google Open-Sources The Machine Learning Tech Behind Google Photos Search, Smart Reply And More*

Schantz, H (1982), *The History Of OCR, Optical Character Recognition*. Recognition Technologies Users Association

Smith, J (2008) *Introduction To Digital Filters: With Audio Applications*. W3K Publishing

Souppouris (2016), "*Google's 'Magenta' project will see if AIs can truly make art"*

Szendro, P (2001), *Pink-Noise Behaviour of Biosystems*. European Biophysics Journal. 30(3): 227–231

Wendy Carlos (1968), *Switched On Bach* [CD-ROM] Colombia Records