CITY, UNIVERSITY OF LONDON

DOCTORAL THESIS

Modelling Metrical Flux: An Adaptive Frequency Neural Network for Expressive Rhythmic Perception and Prediction

Author:
Andrew J. ELMSLEY
(né Lambert)

Supervisors:
Dr. Tillman WEYDE
Dr. Newton ARMSTRONG

A thesis submitted in fulfillment of the requirements for the degree of Doctor of Philosophy

in the

Music Informatics Research Group
Department of Computer Science

August 2017

Contents

List of Figures

List of Tables

Acknowledgements

Firstly I'd like to convey my thanks to my two brilliant supervisors, Tillman Weyde and Newton Armstrong, who took one naïve, out-of-their-depth postgraduate and forged him into the balding, exhausted wreck of a researcher you see today. Throughout the process they've been more than my mentors, but my confidants and friends. Thank you.

I'd like to thank my examiners, Elaine Chew and Gregory Slabaugh. Elaine's research on music cognition, performance, and mathematics has been a constant inspiration. Greg also examined my MPhil-PhD transfer, so has had a big influence on my journey. Hopefully I've influenced him a bit too, by introducing him to Yeasayer.

This research would not be possible without my three-year studentship from City, University of London. I'd like to thank the university and the staff at the Graduate School, Ken Grattan, Naomi Hammond, and Pam Parker, for their outstanding financial support.

The staff in Computer Science are exemplary and a pleasure to interact with. Thanks to the senior tutors for research, Stephanie Wilson and Evangelia Kalyvianaki; to the academic support staff, Naina Bloom, Nathalie Chatelain, Mark Firman, Paula Green, David Mallo-Ferrer, and Gill Smith, for your help on numerous occasions; and to Cagatay Turkay for his outstanding job in organising the PhDs' teaching schedule.

Thanks to all my fellow researchers at the MIRG: Emmanouil Benetos, Srikanth Cherla, Andreas Jansson, Reinier de Valk, and Daniel Wolff. Your support, wisdom, and friendship has been invaluable to me over the last few years. Many thanks also to the MIRG interns Alvaro Correia, Julien Krywyk, and Jean-Baptiste Rémy.

As an interdisciplinary student, the Music department was my second home, and the people were all so friendly and insightful. I'd like to thank Laudan Nooshin for welcoming me, and to any staff member or researcher who had to sit through one of my overly technical talks. Thanks also to the members of the City University Experimental Ensemble for all the strange noises.

Thanks to all my friends at City, who have all shaped this work in some form or other, Andre, Muhammad, Nathan, Niklas, Remi, Rob, and Sam. Thanks especially to Karin for her constant inspiration to improve my musical knowledge.

A special thanks goes to Ryan and Valerio, particularly in the final stages of my research, for keeping me moving forward and inspired about our next adventure.

Thank you to all my Brightonian friends for putting up with my one-track mind, and I wish the best of luck to the remaining PhDs in this group: Anna and Ed. I'd like to give a special mention to Rose for helping with proof-reading, and Chris, Kaile, Mark, Neil, and Tom for giving me a much needed musical outlet.

Thanks to all my family for all their support throughout this period. They may not understand what I'm up to, but they've always been proud of me, and I appreciate that immensely.

Finally the biggest thanks of all goes to my fantastic wife Helen, without whom I would be lost. Thank you for allowing me to wallow, for the reminders to work, for the distractions, and for keeping me grounded.

Declaration of Authorship

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other University. This dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration, except where specifically indicated in the text. This dissertation contains less than 65,000 words including appendices, bibliography, footnotes, tables and equations and has less than 150 figures.

Andrew J. ELMSLEY
August 2017

CITY, UNIVERSITY OF LONDON

Abstract

Department of Computer Science

Doctor of Philosophy

Modelling Metrical Flux: An Adaptive Frequency Neural Network for Expressive Rhythmic Perception and Prediction

by Andrew J. ELMSLEY

Beat induction is the perceptual and cognitive process by which humans listen to music and perceive a steady pulse. Computationally modelling beat induction is important for many Music Information Retrieval (MIR) methods and is in general an open problem, especially when processing expressive timing, e.g. tempo changes or rubato.

A neuro-cognitive model has been proposed, the Gradient Frequency Neural Network (GFNN), which can model the perception of pulse and metre. GFNNs have been applied successfully to a range of 'difficult' music perception problems such as polyrhythms and syncopation.

This thesis explores the use of GFNNs for expressive rhythm perception and modelling, addressing the current gap in knowledge for how to deal with varying tempo and expressive timing in automated and interactive music systems. The cannonical oscillators contained in a GFNN have entrainment properties, allowing phase shifts and resulting in changes to the observed frequencies. This makes them good candidates for solving the expressive timing problem.

It is found that modelling a metrical perception with GFNNs can improve a machine learning music model. However, it is also discovered that GFNNs perform poorly when dealing with tempo changes in the stimulus.

Therefore, a novel Adaptive Frequency Neural Network (AFNN) is introduced; extending the GFNN with a Hebbian learning rule on oscillator frequencies. Two new adaptive behaviours (attraction and elasticity) increase entrainment in the oscillators, and increase the computational efficiency of the model by allowing for a great reduction in the size of the network.

The AFNN is evaluated over a series of experiments on sets of symbolic and audio rhythms both from the literature and created specifically for this research. Where previous work with GFNNs has focused on frequency and amplitude responses, this thesis considers phase information as critical for pulse perception. Evaluating the time-based output, it was found that AFNNs behave differently to GFNNs: responses to symbolic stimuli with both steady and varying pulses are significantly improved, and on audio data the AFNNs performance matches the GFNN, despite its lower density.

The thesis argues that AFNNs could replace the linear filtering methods commonly used in beat tracking and tempo estimation systems, and lead to more accurate methods.

List of Abbreviations

AFNN	**A**daptive **F**requency **N**eural **N**etwork
ANN	**A**rtificial **N**eural **N**etwork
bpm	**B**eats **p**er **M**inute
BLSTM	**B**idirectional **L**ong **S**hort-**T**erm **M**emory Network
CogMIR	**Cog**nitive **M**usic **I**nformatics **R**esearch
CEC	**C**onstant **E**rror **C**arousel
CS	**C**omputer **S**cience
CSD	**C**omplex **S**pectral **D**ifference
CSEMP	**C**omputer **S**ystem(s) for **E**xpressive **M**usic **P**erformance
FHNO	**F**itz**h**ugh-**N**agumo **O**scillator
GFNN	**G**radient **F**requency **N**eural **N**etwork
GTTM	**G**enerative **T**heory (of) **T**onal **M**usic
IMA	**I**nner **M**etrical **A**nalysis
IO	**I**nput / **O**utput
IOI	**I**nter-**o**nset-**i**nterval
LD-GFNN	**L**ow **D**ensity - **G**radient **F**requency **N**eural **N**etwork
LSTM	**L**ong **S**hort-**T**erm **M**emory Network
MAZ	Chopin **Maz**urka Dataset
MIR	**M**usic **I**nformation **R**etrieval
MIREX	**MIR** **E**valuation e**X**change
MSE	**M**ean **S**quared **E**rror
MUME	**Mu**sic **Me**tacreation
NARX	**N**onlinear **A**uto-**R**egression model with e**X**tra inputs
ODF	**O**nset **D**etection **F**unction
opo	**O**scillators **p**er **O**ctave
PCC	**P**earson product-moment **C**orrelation **C**oefficient
RCI	**R**hythmic **C**omplexity **I**ndex
RNN	**R**ecurrent **N**eural **N**etwork
SPECS	**S**tandardised **P**rocedure for **E**valuating **C**reative **S**ystems
VDPO	**V**an **d**er **P**ol **O**scillator
WPO	**W**eighted **P**hase **O**utput

List of Symbols

c	connection matrix	
d	a small change (delta)	
f	a function	
l	length	
M	set of metres	
m	a local metre	
N	number of oscillators	
o	an onset / number of onsets	
p	a piece / number of pulse events	
SW	spectral weight	
t	time	
x	input	
$x(t)$	time-varying input	
y	output	
$y(t)$	time-varying output	
W	metric weight	
z	complex valued variable	
$\bar{z}$	complex conjugate	
$\|z\|, r$	magnitude	
$arg(z)$	phase	rad
α	linear dampening parameter	
β	amplitude compressing parameter	
δ	frequency detuning parameter	
ϵ_f	frequency adaptation parameter	
ϵ_h	frequency elasticity parameter	
ε	nonlinearity parameter	
$\lambda, \mu, \epsilon_c, \kappa$	canonical Hebbian learning parameters	
ω	angular frequency	$\mathrm{rad\,s^{-1}}$
ω_0	initial angular frequency	$\mathrm{rad\,s^{-1}}$
σ	a thresholding function	
φ	phase	rad
Φ	weighted phase output	rad
ζ	noise	

Electronic technology has liberated musical time and changed musical aesthetics. In the past, musical time was considered as a linear medium that was subdivided according to ratios and intervals of a more-or-less steady meter. However, the possibilities of envelope control and the creation of liquid or cloud-like sound morphologies suggests a view of rhythm not as a fixed set of intervals on a time grid, but rather as a continuously flowing, undulating, and malleable temporal substrate upon which events can be scattered, sprinkled, sprayed, or stirred at will. In this view, composition is not a matter of filling or dividing time, but rather of generating time.

Curtis Roads, 2014

For my wife…

Part I

Chapter 1

Introduction

This thesis explores a new model of machine perception of expressively timed rhythms, based on a cognitive model of human perception. This introduction outlines key concepts, defines specific research questions and objectives, and details the structure of the thesis. Section 1.1 introduces the key phenomenon that is modelled in this thesis; *metrical flux*: a changing dynamic feedback loop of metre *perception*, expectational *prediction*, and rhythmic *production*. Section 1.2 sets this phenomenon within the context of computer science research, and in doing so defines the problem space addressed in this thesis. Section 1.3 motivates the method of the enquiry through a description of *entrainment*: a synchronisation process between oscillations. Section 1.4 details the specific research questions and objectives tackled in this thesis, and Section 1.5 gives an overview of the contributions made. Finally, a list of the author's publications and academic achievements is given and the structure of the remaining thesis is outlined.

1.1 Metrical Flux

When we listen to or perform music, a fundamental necessity is to understand how the music is organised in time (Honing, 2012). Musical time is often thought of in terms of two related concepts: the *pulse* and the *metre* of the music. The pulse is the periodic structure we perceive within the music

that we can tap along to. According to Lerdahl and Jackendoff (1983), the metre extends the pulse to a multi-level hierarchical structure. Lower metrical levels divide the pulse into smaller periods and higher levels extend the pulse into bars, phrases, and even higher order forms.

This gives the impression that rhythm is all about dividing or combining periods together, perfectly filling time with rhythmic events. However, in performance this is rarely the case; musicians have been shown to deviate from this abstract clock-quantified pulse in subtly complex ways, often employing this as an expressive device (Räsänen et al., 2015; Clarke, 2001).

Examining expressive qualities of music performance has been ongoing since the Ancient Greeks (Gabrielsson and Lindström, 2010). Today if a performance is too well-timed it is often viewed as being 'robotic', lacking in expressive temporal variation (Kirke and Miranda, 2009). Some genres of music, marches for instance, are designed to induce a strong beat perception. However, it is well known that humans can successfully identify metre and follow the tempo of more expressive rhythms (Epstein, 1995). One recent study on human beat induction found that subjects were able to adapt to relatively large fluctuations in tempo resulting from performances of piano music in various genres (Rankin, Large and Fink, 2009). Skilled performers are able to accurately reproduce a variation from one performance to the next (Todd, 1989a), and listeners are also able to perceive meaning in the deviations from the implied metrical structure (Epstein, 1995; Clarke, 1999).

Electronic music pioneer Curtis Roads mused that the composer has the power to provide a subjective experience of time to the listener, via their perception of rhythmic events (Roads, 2014). Roads considers mainly computer music, where a composer has direct control over the timing of these events, but it is quite possible to extend this view on to every genre of music performed by human or machine.

Listening to and/or performing music forms a dynamic feedback loop of pulse and metre *perception*, expectational *prediction*, and possibly rhythmic *production* in the case of a performing musical agent.

As the performer expressively varies the tempo, the perceived metrical structure is perturbed. Even when the larger scale of the metrical structure remains consistent (e.g. time signature, strong and weak beats), which is often the case, the listener's perception of musical time is affected, along with any expectation of rhythmical events. The endogenous sense of pulse and metre is always in flux throughout the listening process. In this thesis, this is what is referred to as *metrical flux*.

1.2 Expressive Music Modelling

Automatically processing an audio signal to determine pulse event onset times (beat tracking) is a mature field, but it is by no means a solved problem. Analysis of beat tracking failures has shown that beat trackers have great problems with varying tempo and expressive timing (Grosche, Müller and Sapp, 2010; Holzapfel et al., 2012).

Creating formal systems for music theory and composition has a long history, and connectionist machine learning models are a well established approach. Todd's (1989) neural network model, for instance, was trained to predict melody and rhythm. In the network, the problem of melody modelling was simplified by removing timbre and velocity elements, and discretising the time dimension into metrically windowed samples.

More recently the term Music Metacreation (MUME) has emerged to describe contemporary computational approaches to automatic music tasks (Eigenfeldt et al., 2013). MUME systems utilise artificial intelligence, artificial life, and machine learning techniques to develop software that autonomously creates music. Such software is said to be a metacreation if it behaves in a way that would be considered creative if performed by humans (Whitelaw, 2004).

It is still rare for generative music systems to produce temporal variations in their timing and tempo outputs, but a generative system that outputs an quantised symbolic rhythm could always have that rhythm 'played'

by a computer system for expressive music performance (CSEMP; Kirke and Miranda, 2009).

Some holistic approaches have been made, most notably from IRCAM in *Omax* (Assayag et al., 2006) and *ImproteK* (Nika et al., 2014). These systems are both generative improvisation systems, designed to play with human musicians. *Omax*'s design is to ignore the pulse entirely by restructuring the audio input. *ImproteK* uses a beat-tracker to detect tempo, which is then fixed for the remainder of the improvisation.

Sometimes the application of expressive articulation is left to human performers. One example of this is Eigenfeldt's *An Unnatural Selection* (2015) in which human musicians played machine generated phrases, side-stepping the need for any expression to be generated by the system itself.

One MUME goal is the creation of an intelligent musical agent that could perform alongside a human performer as an equal. This is a difficult task, and if it is ever to be achieved, the expressive timing problem must be overcome.

1.3 From Rhythmic Expression to Entrained Oscillation

When Dutch physicist Huygens first built the pendulum clock in 1657, he noted a curious phenomenon: when two pendulum clocks are placed on a connecting surface, the pendulums' oscillations synchronise with each other. As one pendulum swings in one direction, it exerts a force on the board, which in turn affects the phase of the second pendulum, bringing the two oscillations closer in phase. Over time this mutual interaction leads to a synchronised frequency and phase. He termed this phenomenon *entrainment* (Huygens, 1673) and it has since been studied in a variety of disciplines such as mathematics and chemistry (Kuramoto, 1984; Strogatz, 2001; Pantaleone, 2002).

An entrainment process may be mutual or one-sided, which denotes a difference in the way the oscillators interact. A mutual process is one such as two connected pendulums; they each interact with one another to synchronise frequency and phase. A one-sided entrainment process consists of a master oscillator and a slave oscillator: the master influences the slave, but there is no way for the slave to influence the master. Thus only the slave oscillation adapts its frequency and phase. An example of this is the human sleeping cycle which is entrained to the sun's daily rise and set period. We have no influence on how fast the earth spins on its axis, and so this circadian rhythm is a slave oscillator in a one-sided entrainment process.

Connecting two oscillators together like Huygens' pendulums is termed *coupling* and can apply to populations of more than two oscillatory processes. What is often observed as one oscillation may transpire to be a nested population of oscillators, which collectively exhibit a synchronised oscillation. An example of this in biology is the human heartbeat, which is controlled by networks of pacemaker cells (Michaels, Matyas and Jalife, 1987).

In general terms, entrainment occurs whenever temporally structured events are coordinated through interaction, so that two or more periodic signals are coupled in a stable relationship. From this description it is clear to see how entrainment seems particularly relevant to describing the act of musicking (Small, 2011). For instance, an area of music where entrainment has been applied is in the study of how pulse and metre is inferred. When we listen to music, there are entrainment processes taking place that form an endogenous perception of time and temporal structure (the metre) in the piece.

The process through which humans (and other animals, see Patel et al., 2009) perform beat induction is one of entrainment. When we tap our foot to music we are able to synchronise our actions to an external rhythm in such a way that a foot tap coincides with the pulse of that rhythm. By doing so we are entraining to that rhythm. In addition, when we play music, either as a group or solo activity, the entrained periodicity is generated via

a concurrent production process, and is mutual between the participating players (Huron, 2006).

According to Large (2010), the endogenous sense of pulse and metre arises from patterns of neural oscillation activity. Our nervous system literally resonates to rhythm, just as a hollow body resonates to the harmonics of a stimulus. Large, Almonte and Velasco (2010) introduced a canonical oscillator model to study this phenomenon in the brain. The resonant response of the oscillators creates rhythm-harmonic frequency resonances, which can be interpreted as a perception of pulse and metre. The model has been applied successfully to a range of music perception problems including those with syncopated and polyrhythmic stimuli (Angelis et al., 2013; Velasco and Large, 2011). The cannonical model also exhibits entrainment properties, allowing it to shift its phase and resulting in changes to its observed frequency. This makes nonlinear resonance a good candidate for solving the expressive timing problem.

1.4 Research Goal

The main objective of this research was to develop methods for improving the modelling and processing of rhythm, pulse, and metre to address the current gap in knowledge for how to deal with varying tempo and expressive timing in automated and interactive music systems.

This thesis explores a machine learning approach to expressive rhythm perception, addressing all of the aspects above, with a basis in cognitive models of metre perception. The systems studied and developed here operate in continuous time, meaning there is no prior or external knowledge of tempo or metre beyond a single time-series input; there is no filling or dividing time.

1.4.1 Research Questions

This thesis has been driven by the following research questions:

1. Can a neural resonance based cognitive model of human metre perception (GFNN; Large, 2010) improve machine learning music models of melody? This question is explored in Chapter 3.

2. Can GFNNs form a machine learning music model of expressive rhythm production? This question is explored in Chapter 4.

3. How well does a GFNN capture tempo change? This is explored in Chapter 5.

4. Can a similar neural resonance based cognitive model improve the GFNN's response to tempo change and expressive timing? Such a model is developed and evaluated in Chapter 6.

5. How would such a model compare with previous perceptual models and on real-world audio datasets? This question is answered in Chapters 6 and 7.

1.5 Research Contributions

The following are the key contributions of the work presented in this thesis:

1. Development and evaluation of a novel combination of a GFNN with a Recurrent Neural Network (RNN) as two hidden layers within one holistic system.

2. The first evaluation of a GFNN with audio data as input and the first expressive timing study with a GFNN.

3. The first analysis of a GFNN's performance when dealing with changing tempo.

4. Proposal of a new phase-based evaluation metric for pulse perception: *weighted phase output* (WPO).

5. Extension of the GFNN to a novel neural network model, the Adaptive Frequency Neural Network (AFNN), modelling changing periodicities in metrical structures with a new Hebbian learning rule.

6. An open source Python implementation of the GFNN and AFNN models, provided on the author's GitHub repository[1].

7. An evaluation of GFNNs and AFNNs on expressive audio data from standard beat tracking datasets.

1.6 Publications

Parts of this thesis have been published in the following research papers written during the course of the PhD:

1. Andrew J. Elmsley, Tillman Weyde and Newton Armstrong (2017). 'Generating Time: Rhythmic Perception, Prediction and Production with Recurrent Neural Networks'. In: *Journal of Creative Music Systems* 1.2. DOI: 10.5920/JCMS.2017.04

2. Andrew J. Lambert, Tillman Weyde and Newton Armstrong (2016e). 'Metrical Flux: Towards Rhythm Generation in Continuous Time'. In: *4th International Workshop on Musical Metacreation, held at the Seventh International Conference on Computational Creativity, ICCC 2016*. Paris, France

3. Andrew J. Lambert, Tillman Weyde and Newton Armstrong (2016a). 'Adaptive Frequency Neural Networks for Dynamic Pulse and Metre Perception'. In: *Proceedings of the 17th International Society for Music Information Retrieval Conference (ISMIR 2016)*. New York, NY, pp. 60–6

4. Andrew J. Lambert, Tillman Weyde and Newton Armstrong (2015b). 'Perceiving and Predicting Expressive Rhythm with Recurrent Neural Networks'. In: *12th Sound & Music Computing Conference*. Maynooth, Ireland, pp. 265–72

5. Andrew Lambert and Florian Krebs (2015). 'The Second International Workshop on Cross-disciplinary and Multicultural Perspectives on

[1] https://github.com/andyr0id/PyGFNN

Musical Rhythm and Improvisation'. In: *Computer Music Journal* 39.2, pp. 97–100

6. Andrew Lambert, Tillman Weyde and Newton Armstrong (2014b). 'Studying the Effect of Metre Perception on Rhythm and Melody Modelling with LSTMs'. In: *Proceedings of the Tenth Artificial Intelligence and Interactive Digital Entertainment Conference*. Raleigh, NC, pp. 18–24

7. Andrew Lambert, Tillman Weyde and Newton Armstrong (2014a). 'Beyond the Beat: Towards Metre, Rhythm and Melody Modelling with Hybrid Oscillator Networks'. In: *Proceedings of the Joint 40th International Computer Music Conference and 11th Sound & Music Computing Conference*. Athens, Greece, pp. 485–90

1.6.1 Awards

1. Cognitive Music Informatics Research Symposium (2016). *Best Poster: Adaptivity in Oscillator-based Pulse and Metre Perception*.

2. Worshipful Company of Information Technologists (2016). *Outstanding Information Technology Student Award, Silver Prize*.

3. City University Graduate Symposium (2016). *Best Paper: An Adaptive Oscillator Neural Network for Beat Perception in Music*.

4. New York University Abu Dhabi (2014). *Travel bursary: NYUAD Rhythm Workshop*.

1.6.2 Other Academic Activities

The author has made several academic contributions during the course of the PhD, listed below:

1. Andrew J. Lambert (2014a). *A Fractal Depth for Interactive Music Systems*. Music Research Seminar. London, UK. (Talk).

2. Andrew J. Lambert (2014c). *MUME Methodologies: Presentation, Promotion and Appraisal*. 3rd International Workshop on Musical Metacreation (MUME 2014). Raleigh, NC. (Panel Chair).

3. Andrew J. Lambert (2014d). *Towards Metre, Rhythm and Melody Modelling with Hybrid Oscillator Networks*. City Informatics Research Symposium. London, UK. (Talk).

4. Andrew J. Lambert (2014b). *Beyond the Beat: Towards an Expressive Depth in Generative Music*. NYUAD Rhythm Workshop. Abu Dhabi, UAE. (Talk).

5. Andrew J. Lambert, Tillman Weyde and Newton Armstrong (2014c). *Deep Rhythms: Towards Structured Meter Perception, Learning and Generation with Deep Recurrent Oscillator Networks*. DMRN+8. London, UK. (Poster).

6. David Coleman (2014). *2014 Christmas Lectures - Sparks will fly: How to hack your home*. London, UK. (TV appearance).

7. Andrew J. Lambert, Tillman Weyde and Newton Armstrong (2015a). *Generating Time: An Expressive Depth for Rhythmic Perception, Prediction and Production with Recurrent Neural Networks*. Study Day on Computer Simulation of Musical Creativity. Huddersfield, UK. (Talk).

8. Andrew J. Lambert, Tillman Weyde and Newton Armstrong (2015c). *Rhythmic Perception, Prediction and Production with Recurrent Neural Networks*. UVA Music Cognition Group. Amsterdam, Netherlands. (Talk).

9. Andrew J. Lambert, Tillman Weyde and Newton Armstrong (2015d). *Tracking Expressive Timing with Gradient Frequency Neural Networks*. City University Graduate Symposium. London, UK. (Poster).

10. Andrew J. Lambert (2015). *Machine Perception and Generation of Metre and Rhythm*. Music Research Seminar. London, UK. (Talk).

11. Andrew J. Lambert, Tillman Weyde and Newton Armstrong (2016d). *An Adaptive Oscillator Neural Network for Beat Perception in Music*. City University Graduate Symposium. London, UK. (Talk).

12. Andrew J. Lambert, Tillman Weyde and Newton Armstrong (2016b). *Adaptive Frequency Neural Networks for Dynamic Pulse and Metre Perception*. Workshop on Auditory Neuroscience, Cognition and Modelling. London, UK. (Poster).

13. Andrew J. Lambert (2016b). *Creative Music Systems: Current Capacities and Future Prospects*. 1st Conference on Computer Simulation of Musical Creativity. Huddersfield, UK. (Talk).

14. Andrew J. Lambert (2016a). *Creative Music Systems: Bridging the Divide Between Academia and Industry?* 1st Conference on Computer Simulation of Musical Creativity. Huddersfield, UK. (Panelist).

15. Andrew J. Lambert, Tillman Weyde and Newton Armstrong (2016c). *Adaptivity in Oscillator-based Pulse and Metre Perception*. CogMIR. New York, NY. (Poster).

16. Andrew J. Elmsley (2016). *Modelling Metrical Flux: Adaptive Oscillator Networks for Expressive Rhythmic Perception and Prediction*. Queen Mary University of London Cognitive Science Group. London, UK. (Talk).

1.7 Conventions in Figures

Throughout this thesis, abstract symbols are used to represent architectural components of neural networks. Table 1.1 details the various symbols used.

1.8 Thesis Outline

The thesis is structured in two parts. Part I provides some background information and explores existing systems and neural network architectures

A nonlinear activation function

A weighted connection

A network layer

An oscillator

A frequency gradient

An LSTM block

TABLE 1.1: A list of symbols used in figures.

with new experiments. Chapter 2 provides a comprehensive literature review. Chapter 3 studies the problem of metre perception and melody learning in musical signals, proposing a multi-layered GFNN-RNN approach. Chapter 4 presents a machine learning study of the modelling and processing of expressive audio rhythms. Chapter 5 presents the results of an experiment with GFNNs and dynamic tempos.

Part II introduces and explores a new neural network model and evaluates this model over several new experiments. Chapter 6 introduces the AFNN, a novel variation on the GFNN, and the experiment in Chapter 5 is repeated and compared with the previous result. Chapter 7 further evaluates AFNNs with more realistic, human generated audio data. Chapter 8 draws the thesis together, discussing the contributions of the AFNN and the benefits of incorporating the model into an expressive timing rhythm generator and interactive system. Finally, Chapter 9 concludes the thesis and suggests future avenues of research.

Chapter 2

Related Work

2.1 Introduction

This thesis draws together work from many disciplines, each with its own long history of research. In this chapter the previous work is summarised and the gaps in current knowledge are identified. Section 2.2 explores the music theory and analysis literature on pulse and metre. Section 2.3 reviews the dynamical systems and neuroscientific literature on entrainment and oscillation, focussing on theories relating to perception of music. Developments and open problems within beat tracking in Music Information Retrieval (MIR) are described in Section 2.4, and in Section 2.5 the focus is brought to expressive timing and the current modelling approaches in computer science. Section 2.7 summarises computational music modelling and system evaluation approaches, focussing on connectionist machine learning approaches. Finally, Section 2.8 points to the areas this thesis addresses.

2.2 Rhythm, Pulse and Metre

2.2.1 Generative Theory of Tonal Music

In 1983, Lerdahl and Jackendoff laid out their *Generative Theory of Tonal Music* (GTTM), a detailed grammar of the inferred hierarchies listeners perceive when they listen to and understand a piece of music (Lerdahl and

Jackendoff, 1983a). The theory is termed *generative* in the sense of generative linguistics (Chomsky, 1957) whereby a finite set of formal grammars generate an infinite set of grammatical statements. Their novel psychological approach laid the foundations for much of the recent literature on the analysis, cognition, and perception of music.

Central to GTTM is the notion of hierarchical structures in music which are not present in the music itself, but perceived and constructed by the listener. The theory is in fact explicitly limited to perceived structures which are hierarchical in nature. Here a hierarchical structure is defined as a structure formed of discrete components that can be divided into smaller parts and grouped into larger parts in a tree-like manner.

Lerdahl and Jackendoff define four such hierarchies in tonal music (Lerdahl and Jackendoff, 1983b):

1. *grouping structure*: the segmentation of music into various phrases
2. *metrical structure*: the hierarchy of periodic beats the listener infers in the music
3. *time-span reduction*: the importance of pitch events in relation to rhythmic phrases
4. *prolongational reduction*: melodic stability in terms of perceived tension and relaxation

These four elements of GTTM are interrelated and have complex interactions, however this thesis is mainly concerned with *grouping structure* and *metrical structure*, considering the other GTTM structures in relation to these.

2.2.2 Rhythm

Arriving at a complete definition of the complex and multi-faceted notion of 'rhythm' could be a PhD thesis in itself. The concept pervades through all the above GTTM hierarchies.

Even though Lerdahl and Jackendoff stress that rhythmic induction must not be over-simplified, it can be first incorporated into a grouping structure, which can then be subsumed into the other hierarchies.

In GTTM, the most basic definition of rhythm is any sequence of events in time. To create a valid rhythm at least two events with durations must be made, but rhythms can be arbitrarily long and are limited only by the cognitive abilities of the perceiver. These events may optionally have associated pitches, in which case one would refer to the combined structure as a melody.

Long-form rhythms may be perceptually grouped into motifs, themes, phrases and so on, referred to as *grouping structure* in GTTM. Pulse and metre are two examples of such higher-level rhythmic structures formed on top of rhythm, and are discussed in the next sections.

2.2.3 Pulse

A natural and often subconscious behaviour when we listen to music is that we tap our feet, nod our heads, or dance along to it. By doing so, we are reducing the music we hear into a singular periodic signal. This signal can sometimes be present in the music, but is often only implied by the heard musical events and is constructed psychologically in the listener's mind. This process is known as *beat induction*; it is still an elusive psychological phenomenon that is under active research (Madison, 2009; London, 2012), and has been claimed to be a fundamental musical trait (Honing, 2012). Although there are sometimes differences, humans often choose a common period to tap to, Lerdahl and Jackendoff (1983) explain this selection as a *preference rule*.

This preferred period goes by many names. In general it is referred to as the *beat*, but this term is problematic as there is often confusion and ambiguity surrounding it: a beat can also refer to a singular rhythmic event or a metrically inferred event. The technical term used in GTTM is borrowed from the Renaissance, *tactus*, but in this thesis a term is used that has recently grown in popularity in music theory: *pulse* (Grondin, 2008).

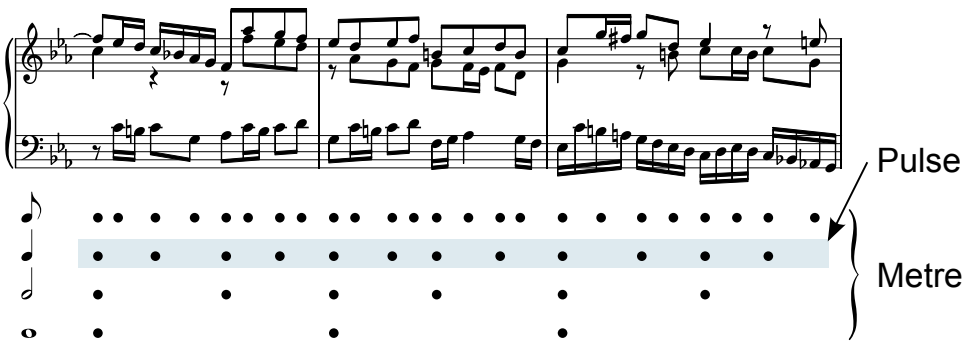


FIGURE 2.1: A metrical analysis of a musical fragment, showing a metrical hierarchy of 'strong' and 'weak' beats.

In this work, the pulse of music is defined as the series of psychological rhythmic events, or beats, that are inferred by the listener. Pulse is also related to tempo, which is defined as the frequency of the pulse, often expressed in beats per minute (bpm). During performance there may be temporal fluctuations in the pulse (see Section 2.5), but the pulse is generally periodic in nature.

2.2.4 Metre

In Section 2.2.3 pulse was defined as a series of psychological rhythmic events inferred by a musical signal. There is usually more than one candidate for it, so within the GTTM model a selection process is required to obtain the pulse. In fact, there are several candidates existing in a hierarchical relationship to the pulse, and possibly others in metrically ambiguous rhythms. These candidates are referred to in GTTM as *metrical levels* and together they form a hierarchical metrical structure (see Figure 2.1).

Each metrical level is associated with its own period, which divides a higher level into a certain number of parts. GTTM is restricted to two or three beat divisions, but in general terms, the period can be divided by any integer. The levels can be referred to by their musical note equivalent, for example a level containing eight beats per bar would be referred to as the quaver level (or eighth note level). It is important to note here that in GTTM beats on metrical levels do not have a duration as musical notes do, but exist only as points in time. Still, it is useful to discuss each level using

the names of their corresponding musical note durations.

The beats at any given level can be perceived as 'strong' and 'weak'. If a beat on a particular level is perceived as strong, then it also appears one level higher, which creates the aforementioned hierarchy of beats. The strongest pulse event in a given bar is known as the *downbeat*. Figure 2.1 illustrates four metrical levels, from crotchet (quarter note) to semibreve (whole note). Theoretically, larger measures, phrases, periods, and even higher order forms are possible in this hierarchy.

2.2.5 Inner Metric Analysis

Metrical structure analysis in GTTM provides a good basis for theoretical grammars and notations of metre and beat saliency. However, it does not adequately describe hierarchical metrical levels with respect to metric stability and change.

In notated music, the time signature suggests that metrical inferences are constant throughout the piece, or at least throughout the bars in which that time signature is in effect. However, the experience of many musicians indicates that the degree to which any metre is being expressed in the music can change throughout a piece. This is known as *metricity* (Volk, 2003). Metric hierarchies can vary, shift, conflict, and complement as the piece moves forward, which leads to changes in the perceived metrical structure. This is what Krebs (1999) refers to as *metrical dissonance*, an example of which can be seen in Cohn's (2001) complex hemiolas, where 3:2 pulse ratios can create complex metrical dynamics throughout a piece. This is not to claim that GTTM does not acknowledge metrical dissonance, indeed metrical dissonance links back to GTTM's time-span reduction and prolongational reduction elements. Nevertheless GTTM does lack the formal means to describe these metrical shifts and to distinguish pieces based on their metrical dissonance.

Inner Metric Analysis (IMA) (Nestke and Noll, 2001; Volk, 2008) forms a structural description of a piece of music in which an importance value, or 'metrical weight', is placed on each note in the piece. This metrical weight

is similar to GTTMs dot notation, where more dots denote stronger beats, but it is sensitive to changes in the metrical perspective and so provides a means to analyse shifting metrical hierarchies in a piece of music.

IMA takes note onset events as the primary indicator of metre, and ignores other aspects often said to be important for metre perception, such as harmony and velocity. The *inner* part of the name relates to this; it is the metric structure inferred by the onsets alone, ignoring the other metrical information available in the notated score. For example, elements such as the time signature are denoted as *outer* structures in that they are placed upon the music and may not arise from the music itself. This makes IMA a perceptual model. Despite the basis on a musical score, it concerns only rhythmic events as observed by a listener. With IMA, metrical dissonance can be expressed as a relationship between inner and outer metrical structures. At the two extremes, when the inner and outer structures concur the metre is coherent, and when they do not the metre is dissonant.

IMA works with the notion of a *local* metre, as defined by Mazzola and Zahorka (1993). A local metre is a series of at least three onsets that are equally spaced in time, creating at least two inter-onset-intervals (IOIs) of equal length. These three onsets can occur anywhere within the piece and do not have to be adjacent. The metrical weight is then calculated by collecting all the local metres in a piece, and then iterating through each event, counting how many local metres each event appears within. A local metre with more events within it is considered to be more stable and as such will contribute more weight than a shorter local metre.

$$W_{l,p}(o) = \sum_{m \in M(l): o \in m} l_m^p \tag{2.1}$$

Equation 2.1 shows the general metric weight, W of an onset, o, where $M(l)$ is the set of all local metres (m) of length at least l in a piece, and l_m is the length of local metre m. p is a parameter which controls how the weight varies with longer and shorter local metres; the higher the value of p, the

greater the contribution of long local metres to the metrical weight. By increasing p or l in Eq. 2.1, one can analyse the longer, and therefore more perceptually stable, local metres. However, these metrical weights are still only effective as long as the local metre is active, and therefore offer a limited, local perspective.

Nestke and Noll introduced another technique of IMA to provide a relation on onsets to a local metre even if the local metre is not occurring during that beat. Such a weighting is called a *spectral weight* and leads from Krebs' argument that once a local metre has been established, one can still relate new material to that level (Krebs, 1999). A spectral weight therefore extends any local metre into the future (and even the past), offering a more global perspective on the metric shifts in the piece. Furthermore, spectral weights consider not only onsets, but also rests within the metrical grid.

$$SW_{l,p}(t) = \sum_{m \in M(l): t \in ext(m)} l_m^p \tag{2.2}$$

Equation 2.2 shows the spectral weight, SW, of an onset or silence, t, where l, l_m, p and $M(l)$ are the same as in Eq. 2.1, and $ext(m)$ is a local metre that has been extended in time to fill the whole piece. Thus, spectral weights can provide a more global perspective on the metrical weight of an onset. The comparison between these local (metrical weight) and global (spectral weight) perspectives offers a way to analyse how metrical dissonance is constructed within a piece.

IMA has been empirically shown to be suitable for the description of the metricity of compositions and how the structural aspects of a composition are transferred to listeners (Volk, 2003).

2.3 Oscillation, Entrainment, and Expectation

2.3.1 Oscillator Entrainment Models

Entrainment has been a particularly useful concept in the field of biology where it has helped to explain natural biological rhythms such as circadian rhythms and some animal organisational and communicational behaviours (Strogatz and Stewart, 1993; Ancona and Chong, 1996; Clayton, Sager and Will, 2005). Swarms of fireflies, for example, are able to flash their abdomens in synchrony. Knoester and McKinley (2011) conducted an artificial evolution experiment to study this behaviour and found that a phase shifting interaction had evolved that was similar to Huygens' (1673) first recorded observations of entrainment.

Kuramoto (1984) formalised entrainment in oscillators in a mathematical model centred around a generalised oscillator as a basic functional unit.

$$\frac{d\varphi_i}{dt} = \omega_i + \zeta_i + \frac{K}{N}\sum_{j=1}^{N}\sin(\varphi_j - \varphi_i) \tag{2.3}$$

Equation 2.3 shows this model, where $\frac{d\varphi_i}{dt}$ is the change in phase of the ith oscillator, ω_i is its natural frequency, ζ_i is a noise term, φ_i, φ_j are the phase values of the i^{th} and j^{th} oscillator, and K is a coupling constant. The model describes a population of oscillators that are globally coupled via a sine interaction function. The output of the interaction function is greatest when the oscillators are in an anti-phase relationship (when they differ by $\frac{\pi}{2}$), and is weakest when the phases are identical, or differ by π. This way, the population is drawn to a mutual phase-locked synchronised state.

Whilst the Kuramoto model is a somewhat elegant encapsulation of the entrainment process, other models exist which combine periodic behaviour and innate entrainment properties. One such example is the Van der Pol oscillator (VDPO), which has its origins in electronics and has been used to model biological rhythms (Van der Pol and Mark, 1928; Camacho, Rand

and Howland, 2004).

$$\frac{d^2x}{dt^2} - \varepsilon(1 - x^2)\frac{dx}{dt} + x = y \tag{2.4}$$

Equation 2.4 shows the second order differential equation that defines a forced VDPO, where x is the output of the oscillation. It is a relaxation oscillator, which means that energy builds up slowly, and is then released quickly. The single parameter, ε, is termed the *nonlinearity* or *damping* coefficient and controls this energy flow and therefore the frequency. As ε approaches 0, the oscillation becomes more sinusoidal, as can be seen in Figure 2.2. When $\varepsilon > 0$ the model obeys a limit cycle, which means that energy will steadily grow or deplete in the model until it reaches a stable amplitude. y is a forcing term. VDPOs can be *unforced* by setting $y = 0$, or *forced* by providing a periodic driving term, $y = f(t)$. When this occurs, the VDPO entrains to the forcing frequency. Complex rhythmic dynamics can be created by coupling a population of VDPOs with a forcing term (Camacho, Rand and Howland, 2004; Lambert, 2012).

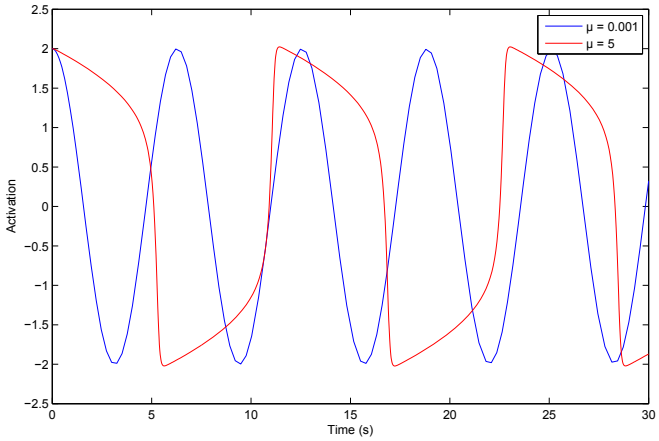


FIGURE 2.2: Two unforced ($y = 0$) VDPOs with $\varepsilon = 0.001$ and $\varepsilon = 5$ respectively.

2.3.2 Expectation and Attentional Dynamics

Jones (1976) was among the first to propose an entrainment theory for the way we perceive, attend, and memorise temporal events. The psychological theory addresses how humans are able to track, attend, and order temporal events by embracing the notion of time in the definition of stimulus structure. Jones posits that rhythmic patterns such as music and speech potentially entrain a hierarchy of attentional oscillations, forming an *attentional rhythm*. These attentional rhythms inform an expectation of when events are likely to occur, so that we are able to focus our attention at the time of the next expected event. In doing so, expectation influences how a temporal pattern is perceived and memorised. Thus, entrainment assumes an organisational role for temporal patterns and offers a prediction for future events, by extending the entrained period into the future.

Large and Jones (1999) extended this initial theory with the theory of *attentional dynamics*. The aim of attentional dynamics was to explain how listeners respond to systematic change in rhythmic events while retaining a general sense of their structure. Temporal and structural modulations occur as the pattern reveals itself, but we are still able to perceive, attend, and memorise these rhythms. Similar to Jones (1976), the model uses entrained attending oscillators which target attentional energy at expected points in time. The oscillators interact in various ways to enable attentional tracking of events with complex rhythms. Large and Jones validated the dynamic attending model by conducting several listening tests and comparing the results with simulated predictions of the model. They found that the model's predictions matched that of the human listeners, providing an indication that the model may be capturing some aspects of human behaviour.

2.3.3 Entrainment and Music

Ethnomusicologists are increasingly becoming aware of the importance of entrainment processes in understanding music making and music perception as a culturally interactive process (Clayton, Sager and Will, 2005). Ensembles are able to mutually synchronise with one another using complex visual and audio cues (Vera, Chew and Healey, 2013). The theory of attentional dynamics (see Section 2.3.2) was motivated in part by questions about music perception, and has an entrainment model at its core.

Large and Kolen (1994), Large (1995), and McAuley (1995) argue that the perception of metrical structure is a dynamic entrainment process between an external musical stimulus and internal processing mechanisms. They put forward differing oscillator models which entrain their frequency and phase to a single periodic component of a rhythmic stimulus, and thus infer the pulse of that stimulus.

So far the focus has been placed on entrainment as a process in which two periodic signals are brought into frequency and phase synchrony, but many relationships are possible in entrained signals. A 1:1 ratio of frequencies, exact synchronisation, is only one case of entrainment; other ratios such as 1:2, 3:2, 2:3 etcetera can still be said to be entrained as long as the ratio between the two signals remains reasonably consistent.

Large and Kolen's oscillator model hints at the ability to connect networks of oscillators, to entrain themselves at different ratios to the pulse. The network's response pattern can be interpreted as a hierarchical metrical structure.

Nonlinear Resonance

The phenomenon of nonlinear resonance (Large and Kolen, 1994) has been applied to metre perception and categorisation tasks. Large, Almonte and Velasco (2010) have introduced the Gradient Frequency Neural Network (GFNN), which is a network of oscillators whose natural frequencies are distributed across a spectrum. When a GFNN is stimulated by a signal,

the oscillators resonate nonlinearly, producing larger amplitude responses at certain frequencies along the spectrum. This nonlinear resonance can account for pattern completion, the perception of the missing fundamental, tonal relationships and the perception of metre.

When the frequencies in a GFNN are distributed within a rhythmic range, resonances occur at integer ratios to the pulse. These resonances can be interpreted as a hierarchical metrical structure. Rhythmic studies with GFNNs include rhythm categorisation (Bååth, Lagerstedt and Gärdenfors, 2013), beat induction in syncopated rhythms (Velasco and Large, 2011) and polyrhythmic analysis (Angelis et al., 2013).

Equation 2.5 shows the differential equation that defines a Hopf normal form oscillator with its higher order terms fully expanded. This form is referred to as the canonical model, and was derived from a model of neural oscillation in excitatory and inhibitory neural populations (Large, Almonte and Velasco, 2010). z is a complex valued output, $\bar{z}$ is its complex conjugate, and ω is the driving frequency in radians per second. α is a linear damping parameter and also a bifurcation parameter: when $\alpha < 0$ the model behaves as a damped oscillator, and when $\alpha > 0$ the model oscillates spontaneously. β_1 and β_2 are amplitude compressing parameters, which increase stability in the model. δ_1 and δ_2 are frequency detuning parameters, and ε controls the amount on nonlinearity in the system. Coupling to a stimulus is also nonlinear and consists of a passive part, $P(\varepsilon, x(t))$, and an active part, $A(\varepsilon, z)$, controlled by a coupling parameter k, producing nonlinear resonances.

$$\frac{dz}{dt} = z(\alpha + i\omega + (\beta_1 + i\delta_1)|z|^2 + \frac{(\beta_2 + i\delta_2)\varepsilon|z|^4}{1 - \varepsilon|z|^2}) + kP(\epsilon, x(t))A(\epsilon, z) \quad (2.5)$$

$$P(\epsilon, x(t)) = \frac{x}{1 - \sqrt{\varepsilon}x} \quad (2.6)$$

$$A(\epsilon, z) = \frac{1}{1 - \sqrt{\varepsilon}\bar{z}} \quad (2.7)$$

By setting the oscillator parameters to certain values, a wide variety of behaviours not encountered in linear models can be observed (see Large,

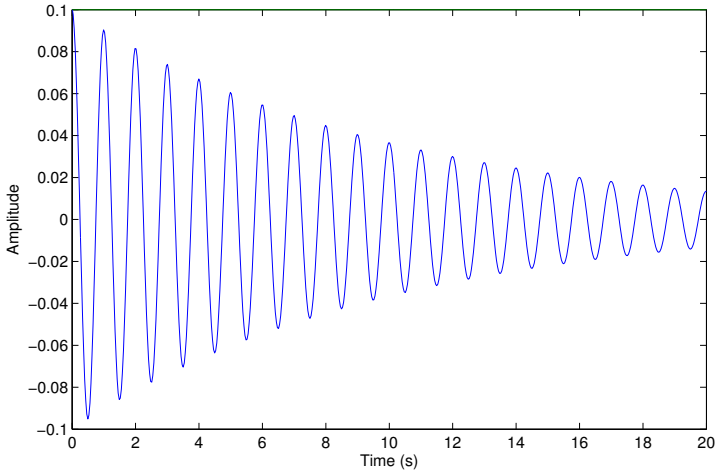


FIGURE 2.3: A cannonical oscillator without stimulus, and with the following parameters, $\omega = 2\pi, \alpha = -0.1, \beta_1 = 0, \beta_2 = -0.1, \delta_1 = 0, \delta_2 = 0, \varepsilon = 0.5, c = 0, x(t) = 0$

2010). In general, the model maintains an oscillation according to its parameters, and entrains to and resonates with an external stimulus. Figure 2.3 shows the waveform of a simplified canonical model (with the parameters β_1, δ_1 and δ_2 set to 0); it is a sinusoid-like waveform whose amplitude is gradually dampened over time. This gradual dampening of the amplitude allows the oscillator to maintain a temporal memory of previous stimulation.

GFNNs typically consist of a number of canonical oscillators, and the frequencies are usually logarithmically distributed to match the nature of octaves in pitch and rhythm. Performing a Fourier transform on the GFNN output reveals that there is energy at many frequencies in the spectrum, including the pulse (Figure 2.4). Often this energy is located at integer ratios to the pulse, implying a perception of the metrical structure.

Furthermore, oscillators within a network can be connected to one another with a connection matrix as is shown in Eq. 2.8,

$$\frac{dz}{dt} = f(z, x(t)) + \sum_{i \neq j} c_{ij} \frac{z_j}{1 - \sqrt{\epsilon} z_j} \cdot \frac{1}{1 - \sqrt{\epsilon} \bar{z}_i} \tag{2.8}$$

where $f(z, x(t))$ is the right-hand side of Eq. 2.5, c_{ji} is a complex number representing phase and magnitude of a connection between the i^{th} and j^{th} oscillator, z_j is the complex state of the j^{th} oscillator, and $\bar{z}_i$ is the complex

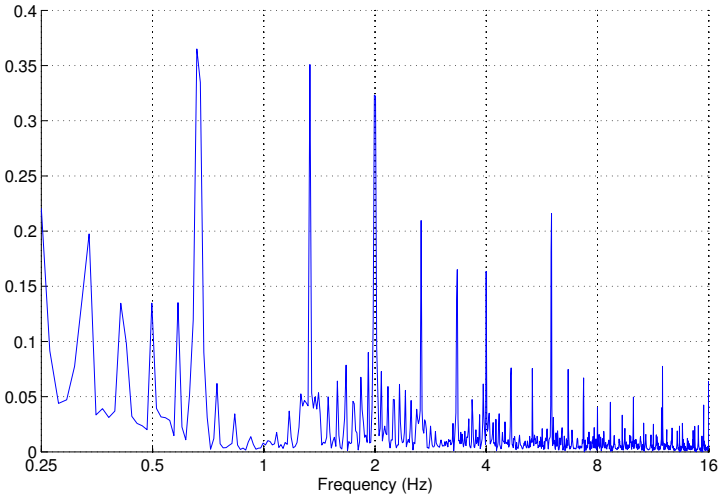


FIGURE 2.4: An example magnitude spectrum of a summed GFNN output.

conjugate of the i^{th} oscillator.

Velasco and Large (2011) connected two GFNN networks in a pulse detection experiment for syncopated rhythms. The two networks modelled the sensory and motor cortices respectively. In the first network, the oscillators were set to a bifurcation point between damped and spontaneous oscillation ($\alpha = 0, \beta_1 = -1, \beta_2 = -0.25, \delta_1 = \delta_2 = 0$ and $\varepsilon = 1$). The second network was tuned to exhibit double limit cycle bifurcation behaviour ($\alpha = 0.3, \beta_1 = 1, \beta_2 = -1, \delta_1 = \delta_2 = 0$ and $\varepsilon = 1$), allowing for greater memory and threshold properties. The first network was stimulated by a rhythmic stimulus, and the second was driven by the first. The two networks were also internally connected in integer ratio relationships such as 1:3 and 1:2. The results showed that the predictions of the model match human performance, implying that the brain may be adding frequency information to a signal to infer pulse and metre.

The internal connections were hand-coded for Velasco and Large's experiment, however Hebbian learning can be incorporated on these connections. Hebbian learning is a correlation-based unsupervised learning observed in neural networks (Kempter, Gerstner and Hemmen, 1999). In a similar way to Hoppensteadt and Izhikevich's (1996) model, resonance relationships between oscillators can form strong bonds.

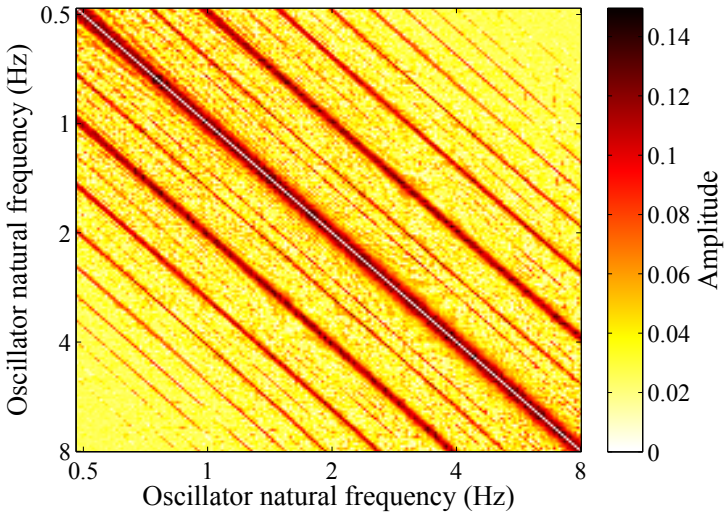


FIGURE 2.5: Amplitudes of a GFNN connection matrix, showing connections formed at high-order integer ratios.

The GFNN's Hebbian rule is shown in Eq. 2.9 and 2.10,

$$\frac{dc_{ij}}{dt} = c_{ij}(\lambda + \mu_1|c_{ij}|^2 + \frac{\epsilon_c\mu_2|c_{ij}|^4}{1-\epsilon_c|c_{ij}|^2}) + f(z_i, z_j) \tag{2.9}$$

$$f(z_i, z_j) = \kappa\frac{z_i}{1-\sqrt{\epsilon_c}z_i}\cdot\frac{z_j}{1-\sqrt{\epsilon_c}\bar{z_j}}\cdot\frac{1}{1-\sqrt{\epsilon_c}z_j} \tag{2.10}$$

where λ, μ_1, μ_2, ϵ_c and κ are all canonical Hebbian learning parameters. The Hebbian rule is similar in form to the canonical oscillator model itself (see Eq. 2.5), but without a driving frequency of its own. λ acts as a bifurcation parameter, while μ_1, μ_2, and ϵ_c affect the amount of nonlinearity in the matrix. z_i and z_j are the complex states of the i^{th} and j^{th} oscillators, and $\bar{z_j}$ is the complex conjugate of the j^{th} oscillator. κ is a coupling coefficient controlling the strength of the interactions between these oscillators and the learning rule.

Figure 2.5 shows a connection matrix after Hebbian learning has taken place. The Hebbian parameters are set to the following: $\lambda = .001, \mu_1 = -1, \mu_2 = -50, \epsilon_c = 16, \kappa = 1$. In this example the oscillators have learned connections to one another in the absence of any stimulus due to the oscillators operating in their limit cycle behaviour, meaning that they oscillate spontaneously at a stable amplitude. The connections can then learn ratios

of 1:1, 2:1 and 1:2, 3:1 and 1:3, 3:2 and 2:3, 4:3 and 3:4, and even higher order integer ratios.

Adding a connection matrix to a GFNN and activating Hebbian learning can strengthen and stabilise the entrainment and nonlinear resonance phenomena, reducing noise in the network from non-resonant frequencies. It also has the effect of reducing the amount of time required for resonances to appear in the network, which can sometimes take several seconds for frequencies in the rhythmic range (<=16Hz). Changes to the nonlinear resonance patterns in the GFNN over time and the learned connection matrix enables a similar analytical method to IMA (see Section 2.2.5), but is open for use with both symbolic and audio data.

2.4 Beat Tracking

In Music Information Retrieval (MIR), automatically processing an audio signal to determine pulse event onset times is known as *beat tracking*. It falls into a branch of MIR known as *automatic rhythm description* (Gouyon and Dixon, 2005). Beat tracking is useful for many MIR applications, such as tempo induction, which describes the rate of the pulse (Gouyon et al., 2006); rhythm categorisation, which attempts to identify and group rhythmic patterns (Bååth, Lagerstedt and Gärdenfors, 2013; Dixon, Gouyon and Widmer, 2004); downbeat tracking and structural segmentation, which aim to meaningfully split the audio into its temporal fragments such as bars and phrases (Levy, Sandler and Casey, 2006; Krebs, Böck and Widmer, 2013); and automatic transcription, which aims to convert audio data into a symbolic format (Klapuri, 2004).

Davies and Plumbley (2007) list four desirable properties for a beat tracking system:

1. both audio and symbolic data can be processed
2. no *a priori* knowledge of the input, such as genre information, is required

3. the system is efficient and can operate in realtime where necessary

4. changes in tempo can be followed and pulse fluctuation due to expression can be tracked

Beat tracking is still an active research topic, with many approaches taken. Yet, writing in 2007, Davies and Plumbley state that a beat tracking system meeting all of these requirements is non-existent, despite the long history of research dating back to 1990 (Allen and Dannenberg, 1990).

Dixon (2001) and Goto (2001) have both created agent-based beat trackers. Dixon's system was designed with time-varying tempo and rhythmic expression in mind. Several agents predict beat locations and the agent which predicts the beat most accurately is then used as the output of the system. Goto's system performs best when the audio input remains at a fixed tempo and is in a 4/4 metre, but is able to accurately track beats in realtime, at three metrical levels.

Scheirer's (1998) system takes a more perceptual approach by using linear comb filters, which operate on principles similar to Large and Kolen's (1994) nonlinear resonance model. The comb filter's state is able to represent the rhythmic content directly, and can track tempo changes by only considering one metrical level. Klapuri, Eronen and Astola's (2006) more recent system builds from Scheirer's design by also using comb filters, but extends the model to three metrical levels. Both Scheirer's and Klapuri, Eronen and Astola's (2006) systems perform well but struggle with complex rhythms such as polyrhythmic or syncopated stimuli.

Taking a machine learning approach, Böck and Schedl (2011) use a particular type of Recurrent Neural Network (RNN) called a Long Short-Term Memory Network (LSTM, see Section 2.7.2). The LSTM predicts a frame-by-frame beat activation function and does not rely on a separate onset detection step, but instead takes spectral features as input. The LSTM model used here is known as a bidirectional LSTM (BLSTM), and deals with input both forward and backward in time simultaneously. This increases the accuracy of the model as it has data from the future, but it also means that the

system is not ready for real-time input. Böck and Schedl propose two beat tracking systems: one which is better at steady tempos, and one which is able to track tempo changes. The only difference between the two systems is the post-processing step where the peaks in the BLSTM output are used for predicting beats.

The MIR Evaluation eXchange (MIREX)[1] project runs a beat tracking task each year, which evaluates several submitted systems against various datasets. This provides an easy way to determine what the current state-of-the-art of beat tracking systems is. In MIREX 2016 the best performing beat tracker was Böck and Schedl's BLSTM system, with probabalistic extraction of beat times (Korzeniowski, Böck and Widmer, 2014).

2.4.1 Beat Tracking with Nonlinear Oscillators

In Section 2.3.3, several entraining oscillator models were introduced which have been used in cognitive science, psychology, and neuroscience to model human beat induction and metre perception. Attempts have also been made to apply these models practically in a beat tracker.

Large (1995) used an early version of the nonlinear resonance model to track beats in performed piano music. The pianists played a selection of monophonic children's songs and a number of monophonic improvisations, and a single nonlinear oscillator was tasked with inferring a pulse at any metrical level. The oscillator performed fairly well, with most of the errors occurring due to the oscillator synchronising to the pulse in an anti-phase relationship. However, in cases when the performer was making heavy use of *rubato* (see Section 2.5), the beat tracker failed. Large suggests that this problem may be overcome by using networks of oscillators instead of the singule one used here. If the oscillator had been attempting to track a different metrical level then no rubato would have been encountered.

Large and Kolen's (1994) nonlinear resonance model uses an oscillator model of their own design. Eck and Schmidhuber (2002), by comparison, used an oscillator model from neurobiology named a FitzHugh-Nagumo

[1] http://www.music-ir.org/mirex/

oscillator (FHNO) (FitzHugh, 1961; Nagumo, Arimoto and Yoshizawa, 1962) and applied it to the task of beat induction. FHNOs are relaxation oscillators with natural entrainment properties, much like the VDPO discussed in Section 2.3.3. After somewhat successfully tracking the downbeats in a series of rhythmic tests devised by Povel and Essens (1985), Eck (2001) then connected several oscillators in a network. Eck found that there were several advantages to the networked approach, including increased stability and the ability to track multiple periodic elements in parallel.

Eck, Gasser and Port (2000) used a network of FHNOs in a robotic system that taps beats with a robotic arm. This embodied approach utilises a feedback loop with the robot's actuators, which provided an additional periodic input to the oscillator network, and was able to further stabilise the system. This experiment provides further evidence for the potential of entrained dynamic models such as oscillators for beat induction tasks, and is similar in principle to Velasco and Large's (2011) two-network model of the sensory and motor cortices.

2.4.2 Where Beat Trackers Fail

As mentioned in Chapter 1, Section 1.2, beat tracking in MIR is generally considered an open problem. Many systems have been proposed over the years, and the current state-of-the-art beat trackers do a relativity good job of finding the pulse in music with a strong beat and a steady tempo, yet we are still far from matching the human level of beat induction. There has been a recent surge in new beat-tracking systems (see for example, Davies and Plumbley, 2007; Dixon, 2007; Böck and Schedl, 2011), but little improvement over Klapuri, Eronen and Astola's (2006) system. This has lead Holzapfel et al. (2012) to hypothesise that the development of suitable beat tracking datasets with suitable features had become stagnant and unfit for purpose.

Holzapfel et al. claim that datasets do not contain enough excerpts

where beat trackers fail. Instead these challenging cases are treated as outliers and often ignored in discussions of results. This is not to say that studies of musical properties which make beat trackers fail have not been undertaken. Dixon (2001) has conducted one such study and proposed two measures to gauge the beat tracking difficulty of a given signal: the rhythmic complexity, and the amount of expressive variations in tempo and timing in performance.

Rhythmic complexity can be estimated via the *Rhythmic Complexity Index* (RCI) metric, which measures rhythmic syncopation as a proportion of pulse events without onset events, and onset events which do not fall on pulse events.

$$RCI = \frac{p_u + o_u}{p + o_u} \tag{2.11}$$

Equation 2.11 shows the formula to calculable RCI where p is the total number of pulse events, p_u is the number of pulse events without matching onset events, and o_u is the number of onset events without matching pulse events. RCI is a number between 0 and 1, with higher values indicating higher rhythmic complexity.

For a measure of timing variability, Dixon proposes the use of the standard deviation of the inter-beat intervals as a simple indicator. This can be expressed relative to the average beat interval to give a number that is comparable across excerpts in different tempos.

Grosche et al. have also performed an in-depth analysis of beat tracking failures on the Chopin Mazurka dataset[2] (MAZ) (Grosche, Müller and Sapp, 2010). MAZ is a collection of audio recordings comprising on average 50 performances of each of Chopin's Mazurkas. Grosche and Müller tested three beat tracking algorithms on a MAZ subset and looked at consistent failures in the algorithms' output with the assumption that these consistent failures would indicate some musical properties that the algorithms were struggling with. They found that properties such as expressive timing and ornamental flourishes were contributing to the beat trackers' failures.

[2]http://www.mazurka.org.uk/

To contribute a solution to the problem, Holzapfel et al. (2012) selected excerpts for a new beat tracking dataset by a selective sampling approach. Rather than comparing one beat tracker's output to some ground truth annotation, several beat trackers' outputs were compared against each other. If there was a large amount of mutual disagreement between predicted beat locations, the track was assumed to be difficult for current algorithms, and was selected for beat annotation and inclusion in the new dataset. This resulted in a new annotated dataset, now publicly available as the SMC dataset[3].

The SMC excerpts are also tagged with a selection of signal property descriptors. This allows for an overview of what contributes to an excerpt's difficulty. There are several timbral descriptors such as a lack of transient sounds, quiet accompaniment and wide dynamic range, but most of the descriptors refer to temporal aspects of the music, such as slow or varying tempo, ornamentation, and syncopation. Over half of the dataset is tagged with the most prominent tag: expressive timing.

2.5 Expressive Timing

From the beat tracking literature it is clear that being able to track expressive timing variations in performed music is one area in which there is much room for improvement (see Section 2.4.2). This is especially true if one is attempting to achieve a more human-like performance from the beat tracking algorithms. This has been attempted in many cases, most notably in Dixon (2001) and Dixon and Goebl's (2002) work, which lead to the Beatroot system (Dixon, 2007). However, Beatroot does not perform well on today's standard datasets, scoring poorly on the SMC dataset in recent MIREX results. Solving this issue in MIR could also lead to better segmentation techniques, such as phrase extraction where tempo and loudness curves are key indicators (Chuan and Chew, 2007; Cheng and Chew, 2008; Stowell and Chew, 2012).

[3] http://smc.inescporto.pt/research/data-2/

According to Gabrielsson and Lindström (2010), the examination of the expressive qualities of music has been ongoing since the Ancient Greeks, with empirical research starting around one century ago. The research field looks at what emotional meanings can be expressed in music, and what musical structures can contribute to the perception of such emotions in the listener. These structures can be made up of multi-faceted musical parameters such as dynamics, tempo, articulation, and timbre. Often these aspects have complex interactions, and a change in one can influence a perception of another (Chew, 2016). In this thesis, the focus is placed on the temporal aspects of expression.

Performers have been shown to express the metrical structure of a piece of music by tending to slow down at the end of metrical groupings. The amount a performer slows down correlates to the importance of the metrical level boundary (Clarke, 2001). It is well known that humans can successfully identify metre and follow the tempo based off such an expressive rhythm (Epstein, 1995). Rankin, Large and Fink (2009) conducted a study on human beat induction and found that we are able to adapt to relatively large fluctuations in tempo resulting from performances of piano music in various genres. The participants could successfully find a pulse at the crotchet or quaver metrical level. Skilled performers are able to accurately reproduce a variation from one performance to the next (Todd, 1989a), and listeners are also able to perceive meaning in the deviations from the implied metrical structure (Epstein, 1995; Clarke, 1999).

Chew and Callender (2013) have proposed an analytical approach that simultaneously considers tempo and log(tempo) changes in terms of score time and performance time. log(tempo) analysis supports the ability to proportionally and relatively describe changes in much the same way as pitch. For example, Large's nonlinear resonance (see Section 2.3.3) can be considered to be a log(tempo) representation, since the distribution of frequencies are usually logarithmic.

By linking tempo markings on the score and its rendering in performance, one is able to consider tempo change in a similar manner to IMA's

metrical dissonance (see Section 2.2.5). However, Chew and Callender's concept of score time and performance time is a step beyond Nestke and Noll's inner and outer meters, in that score time and performance time can help inform one another.

However, computer systems for expressive music performance (CSEMPs) have received little attention from both academia and the industry at large. According to Kirke and Miranda (2009), the introduction of built-in sequencers into synthesizers in the early 1980s contributed to a new, perfectly periodic timing, which sounded robotic to the ear. Rather than look for ways to make this timing model more human-like, artists embraced the robotic style to produce new genres of music such as synth pop and electronic dance music, which soon dominated the popular music scene.

One of the most common expressive devices when performing music is the use of rubato to subtly vary the tempo over a phrase or an entire piece. Todd (1989) produced a model of rubato implemented in Lisp which is able to predict durations of events for use in synthesis. The original model was based on Lerdahl and Jackendoff's ideas on time span reduction in GTTM (see Section 2.2.1). However, this was deemed psychologically implausible as it places too high a demand on a performer's short-term memory. In a similar way to IMA's spectral weight (see Section 2.2.5), the model considers all events regardless of time differences.

Todd's improved model incorporates a hierarchic model for timing units from a piece-wise global scale to beat-wise local scale. It works by looking at a score and forming an internal representation via GTTM's grouping structures. This internal representation is then used in a mapping function, outputting a duration structure as a list of numbers. Even though the model makes predictions about timing and rubato, it forms an analytical theory of performance rather than a prescriptive theory.

Today, research into CSEMPs is a small but important field within Computer Music. Widmer and Goebl (2004) have published an overview of existing computational models, and Kirke and Miranda (2009) have produced

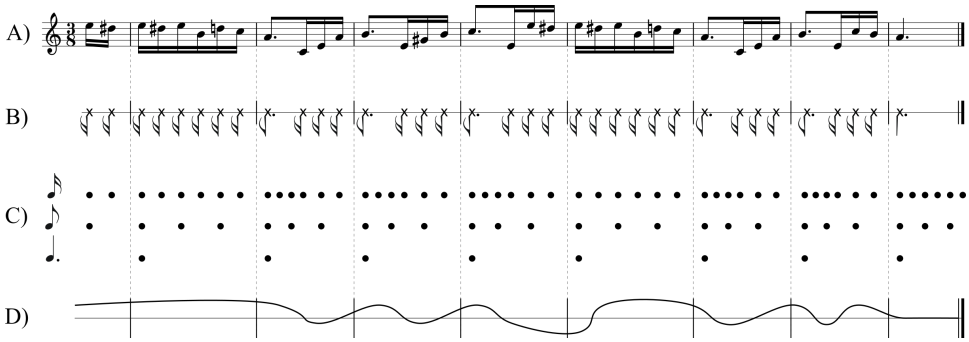


FIGURE 2.6: A simplified excerpt from Beethoven's *Für Elise*, showing (A) the score, (B) the rhythm, (C) the metrical structure, and (D) performed tempo.

a survey of available CSEMPs and an outline of a framework that most (but not all) CSEMPs adhere to. In this framework, the main kernel of the system is its ability to generate expressive performances, and is referred to as *performance knowledge*. *Performance context* and *music / analysis* processes feed into the kernel, along with an *adaptation process*, which can either be an automatic or manual way of controlling the performance knowledge. The adaptation process may also rely on *performance examples*, a corpus of music from which the adaptation process can learn. The performance knowledge kernel outputs audio or symbolic data via an *instrument model*, which is then fed back to the adaptation process.

2.6 Understanding Metrical Flux

In Chapter 1, Section 1.1 the notion of metrical flux was defined as a constant change in a listener's perception of musical time and expectation of rhythmical events. In the above sections some more background literature was discussed, unpacking the various aspects of metrical flux: rhythm, pulse, metre and expressive timing. In this section, these aspects are put together through an example of metrical flux in action.

Figure 2.6 shows a simplified excerpt of the opening bars from Ludwig van Beethoven's famous composition, *Für Elise*. The first row, (A), shows the main melody line in music notation, where we learn that the piece is in a 3/8 time signature and is in an A minor key. In row (B) the melody line

has been reduced to a rhythm-only representation; that is, an event and duration. One can see that the rhythm consists of mainly semi-quaver (16th note) durations, and is punctuated by dotted quavers (8th notes) occurring on the downbeats of bars 3, 4, 5, 7, and 8. A final dotted crotched completes the phrase. Furthermore, there is an anacrusis of one quaver in bar 1.

Row (C) displays the metrical structure of the piece. The pulse has not been highlighted as there are two equally valid choices: the quaver level or the dotted crotchet level. Musicians will generally choose their preferred metrical level to tap along to as the pulse. The anacrusis has offset the downbeats, and so the piece begins on a weakly perceived beat. This syncopation continues throughout the phrase, with the aforementioned dotted crotchets providing a sense of pace and higher-level structure.

When playing this piece on a piano, for instance, it is extremely common for the performer to add expressive flourishes in dynamics and tempo, to accentuate sub-phrases in the metrical structure. Row (D) indicates one such performance with an abstract tempo curve. Here the tempo remains fairly steady in the first two bars, before slowing at the sub-phase boundaries in bars 3, 4, and 5. In bar 5 there is a more pronounced ritardando, which extends into the next sub-phrase, before resuming the same tempo curve pattern as in earlier bars.

This combination of rhythm, pulse, metre, and expressive timing affects a listener's perception of musical time. Any future predictions of rhythmic events by the listener are also affected. The general aim of this thesis is to model this process of metrical flux.

2.7 Music Metacreation

Creating formal systems for music composition has a long history dating back around one thousand years when Guido d'Arezzo invented a way to automatically convert text into melodic phrases. There are numerous works throughout history that employ formalism in their creation, including Bach's *The Art of Fugue*, Schönberg's twelve-tone compositions, and

Cope's EMI compositions (Nierhaus, 2009; Cope, 1992). Boden and Edmonds (2009) place such systems within the broad field of Generative Art, stating that these generative techniques could include any rule-based system.

Over the years the notion has taken on many names such as automated composition, algorithmic composition and generative music (Collins and Brown, 2009), but recently the term *music metacreation* (MUME) has emerged to describe contemporary computational approaches to the field (Eigenfeldt et al., 2013). MUME as a term stems from Whitelaw's notion of metacreation; that is, the use of artificial intelligence, artificial life, and machine learning techniques to develop software that autonomously creates (Whitelaw, 2004). Such software is said to be a metacreation if it behaves in a way that would be considered creative if performed by humans.

MUME is not concerned with *if* a computer can be creative, as that question has already been answered with a resounding *yes* with systems such as Colton's *The Painting Fool* (Colton, 2012) and the aforementioned EMI which received critical acclaim. Instead MUME examines *how* these systems are built and evaluated, and how artists and scientists can collaborate within the interdisciplinary field (Eigenfeldt et al., 2014).

Eigenfeldt et al. (2014) claim that there are two general approaches in MUME research: the cognitive approach which uses models based on human cognitive theories; and the *black box* approach, which does not seek to mimic human processes, but to produce some kind of new machine creativity. The latter approach is the route most taken within MUME practitioners; however, there have been relatively few attempts taking a perceptual and cognitive approach to music generation (Maxwell et al., 2012).

2.7.1 Evaluating MUME Systems

When considering metacreation software, validating the work both in terms of the computational system and the output it creates is still a challenge for the community at large. A system may generate the same output no matter how many times it executes, or it may produce wildly varying outputs on

each execution. However, both these systems may be perceived as equally 'creative'. The way these systems and their outputs can be compared is an ongoing problem facing the MUME community.

According to Jordanous (2011), this is a widespread issue for all computational creativity research. Evaluation of creative systems is not being carried out in a systematic or standardised manner, with many researchers even failing to detail how the presented creative system was evaluated. She argues for a standardised approach, which still allows for flexibly dealing with the inherently different types of outputs for such systems: a Standardised Procedure for Evaluating Creative Systems (SPECS) (Jordanous, 2012).

The SPECS methodology consists of three steps:

1. Identify a definition of creativity that the system is aiming to satisfy
2. Clearly define what standards are used to define said creativity
3. Test the system against said standards and report the results

Step 1 is encouraged to incorporate both a general and a domain-specific definition of creativity, with the latter showing how the former is manifested. SPECS provides several aspects of creativity to examine a creative system within step 2, which helps to identify the aspects of the system that can be considered to be 'creative', and how the system's creativity could be improved. Finally, step 3 is left open to each researcher's specific problem area, with encouraged emphasis on aspects of creativity that are more important to the investigation. By following the SPECS methodology, researchers are able to systematically evaluate both the output of their systems and the system itself.

Eigenfeldt et al. (2013) have also contributed towards a solution to the evaluation problem by proposing a MUME taxonomy to facilitate discussions around measuring metacreative systems and works. The taxonomy is based around the agency or autonomy of the system in question, since in MUME the computational system is an active creative agent. By focussing

on the system's autonomy, one is able to distinguish between the composer's (system designer's) influence on the system and the performance elements, which may change from execution to execution. This is not to say that the taxonomy only works for interactive or online systems; offline systems such as score composition systems also have a degree of interactivity and influence of the creator in their process. For instance a system may generate its own structures completely autonomously, or rely on user inputs to guide the generative process.

Rather than restrict the taxonomy to only online or offline systems, as some researchers have done in their definitions (see Collins, 2008; Pearce, Meredith and Wiggins, 2002), the MUME taxonomy places the metacreation of the system on a gradient through the following seven levels of creativity:

1. *Independence*: there is some process on a gesture that is beyond the control of the composer
2. *Compositionality*: the system determines relationships between two or more gestures
3. *Generativity*: the system creates new musical gestures
4. *Proactivity*: the system decides when to initiate a new gesture
5. *Adaptability*: the system's behaviour changes over time via interaction with itself or other agents
6. *Versatility*: the system determines its own content or gestural style
7. *Volition*: the system decides for itself what, when, and how to compose/perform

The levels are intended to facilitate a comparison between MUME systems, by assigning the highest possible level to each system based on their process. A system on level 4, for example, may exhibit properties of the lower levels, but it cannot exhibit behaviours from level 5 and above. Furthermore, a system on level 5 does not imply either objective nor subjective

superiority to a level 3 system, but solely provides terms to examine each system. According to Eigenfeldt et al., no existing system has yet been classified at level 7, meaning that as composers become close to creating such systems, more levels may be needed to adequately describe the differences.

Both SPECS and the MUME taxonomy provide methodologies for evaluating MUME systems that go some way towards solving the evaluation problem. By utilising both systems, research into creativity can rigorously and systematically be examined and compared to other such systems.

2.7.2 Neural Network Music Models

Todd (1989) and Mozer (1994) were among the first to utilise a connectionist machine learning approach to MUME. One of the major advantages of this approach is that it replaces rule-based systems, which can be brittle, lack novelty, and tend not to deal with unexpected inputs very well. Instead, the structures of existing musical examples are learned by the network and generalisations are made from these learned structures to compose new pieces.

Both Todd's and Mozer's systems are recurrent networks that are trained to predict melody. They take as input the current musical context as a pitch class and note onset marker and predict the same parameters at the next time step. In this way the problem of melody modelling is simplified by removing timbre and velocity elements, and discretising the time dimension into windowed samples.

Whilst Todd and Mozer were mainly concerned with predicting pitch sequences over time, Gasser, Eck and Port (1999) have taken a connectionist approach to perceive and produce rhythms that conform to particular metres. Their neural network model *SONOR* is a self-organising network of adaptive oscillators that uses Hebbian learning to prefer patterns similar to those it has been exposed to in a learning phase. A single input / output (IO) node operates in two modes, perception and production. In the perception mode, the IO node is excited by patterns of strong and weak beats, conforming to a specific metre. Hebbian learning is used to create connections between the oscillators in the network. Once these connections have

been learned, the network can be switched to production mode, reproducing patterns that match the metre of the stimuli.

Recurrent neural networks (RNNs) such as those used in the above systems can be good at learning temporal patterns.

Recently work on modelling symbolic melodies as sequences of information tuples (multiple viewpoints; see Conklin and Witten, 1995; Pearce, 2005), was further extended by Cherla et al. (2013), and Cherla, Weyde and Garcez (2014). It was demonstrated in the latter that a set of six connectionist architectures could perform on par with, or better than state-of-the-art n-gram models previously evaluated in an identical setting on a musical pitch prediction task. Among the connectionist architectures, those relying on recurrent connections to model temporal information showed greater prediction accuracy. This work eventually led to the proposal of a new model known as the Recurrent Temporal Discriminative Restricted Boltzmann Machine (Cherla et al., 2015), which was found to outperform the rest of the five connectionist models, and also the n-grams considered in their study.

However, as noted by Todd (1989) and Mozer (1994), RNNs often lack global coherence due to the lack of long-term memory. This results in sequences with good local structures, but long-term dependencies are often lost. One way of tackling this problem is to introduce a series of time lags into the network input, so that past values of the input are presented to the network along with present values.

$$y(t) = f(y(t-1), ..., y(t-l)) \tag{2.12}$$

Equation 2.12 shows a simple time-series predictor where y represents a variable to be modelled, t is time and l is the number of lag steps in time. Kalos (2006) used a model of this type known as a Nonlinear Auto-Regression model with eXtra inputs (NARX) to generate music data in symbolic MIDI format. One advantage of this method is that it performs well on polyphonic music, but the time lag method still does not capture long-term

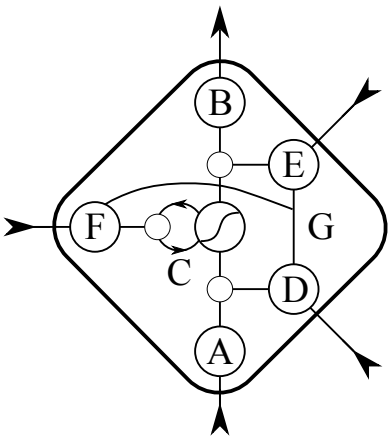


FIGURE 2.7: A single LSTM memory block showing (A) input, (B) output, (C) CEC, (D) input gate, (E) output gate, (F) forget gate and (G) peephole connections.

structure very successfully.

Long Short-Term Memory (LSTM) networks were specifically designed to overcome the problem of modelling long-term structures. Introduced by Hochreiter and Schmidhuber (1997), they noted that whilst RNNs could theoretically learn infinitely long patterns, in practice this was difficult due to the *vanishing gradient problem*. This is where the gradient of the total output error with respect to previous inputs quickly vanishes as the time lags between relevant inputs and errors increase. It can take as little as five time steps for this problem to occur in an RNN (Gers and Schmidhuber, 2001). A self-connected node known as the Constant Error Carousel (CEC) ensures constant error flow back through time, meaning that LSTMs can bridge time lags in excess of 1000 time steps (Hochreiter and Schmidhuber, 1997).

A simplified diagram of an LSTM memory block can be seen in Figure 2.7. The input and output gates control how information flows into and out of the CEC, and the forget gate controls when the CEC is reset. The input, output, and forget gates are connected via *peepholes*. For a full specification of the LSTM model the reader is referred to Hochreiter and Schmidhuber (1997) and Gers, Schmidhuber and Cummins (2000).

LSTMs have already had some success in music applications. Eck (2002) trained LSTMs which were able to improvise blues chord progressions,

Franklin (2006) found that LSTMs can learn long songs and generate new improvisations when given new harmonic inputs, and more recently Coca, Correa and Zhao (2013) used LSTMs to generate melodies that fit within user specified parameters. LSTMs continue to be used within deep learning (Bengio, 2009) systems, such as Sturm et al.'s (2016) automatic transcription and composition systems.

2.8 Conclusions

This chapter summarised the literature relevant to this thesis. The concepts of pulse and metre were unpacked from music-theoretical, music-analytical, and neuroscientific perspectives. It was shown that there is a current open problem in MIR when it comes to automatically tracking expressively timed rhythms. Beat trackers and current CSEMPs may benefit from modelling metre and metrical change as nonlinear resonance. The entrainment properties of the oscillators may be able to directly model tempo fluctuations in these cases. Implementing this into an LSTM melody model may increase the rhythmic accuracy of the system, and allow for continuous-time, non-metrically quantised output.

Chapter 3

Metre and Melody Modelling

3.1 Introduction

This chapter studies the problem of metre perception and melody learning in musical signals. A multi-layered neural network model is presented, consisting of a nonlinear oscillator network and a recurrent neural network (RNN).

The network consists of two different neural network models, connected as hidden layers within one system. The first is a Gradient Frequency Neural Network (GFNN; Large, Almonte and Velasco, 2010), a type of nonlinear oscillator network. It acts as an entrained resonant filter to the musical signal and serves as a metre perception layer. It 'perceives' metre by resonating nonlinearly to the inherent periodicities within the signal, creating a hierarchy of strong and weak periods. The second layer is a Long Short-Term Memory network (LSTM; Hochreiter and Schmidhuber, 1997), a RNN, which is able to learn the kind of long-term temporal structures required in music signal prediction (Eck, 2002).

The aim is to support melody and rhythm modelling in an LSTM by using the GFNN for metre perception. The investigation questions whether a music prediction task produces better results when utilising this model of metrical structure. The network is evaluated in different configurations and with different note representations on a melody prediction task. It is

shown that this network outperforms previous approaches of single layer recurrent neural networks in a melody and rhythm prediction task.

3.1.1 Contributions

This is the first time an oscillator network (GFNN) has been combined with an RNN (LSTM). In this thesis, the combined model is referred to as a GFNN-LSTM.

This chapter explores the following hypothesis: the GFNN-LSTM can make better musical predictions since it is enabled to make use of the relatively long temporal resonance in the GFNN output, and therefore model more coherent long-term structures with the LSTM. A system such as this could be used in a multitude of analytic and generative scenarios, including live performance applications.

Part of this chapter has been published in (Lambert, Weyde and Armstrong, 2014a) and (Lambert, Weyde and Armstrong, 2014b).

3.2 Models

3.2.1 GFNN

The GFNN consisted of 128 canonical oscillators defined by a simplified form of the equation show in Eq. 2.5. The following defines the simplification, derived by setting β_1, δ_1, and δ_2 to zero, and expanding $P(\varepsilon, x(t))$ and $A(\varepsilon, z)$:

$$\frac{dz}{dt} = z(\alpha + i\omega + \frac{\beta\varepsilon|z|^4}{1-\varepsilon|z|^2}) + \frac{x}{1-\sqrt{\varepsilon}x}.\frac{1}{1-\sqrt{\varepsilon}\bar{z}} \tag{3.1}$$

The parameters and variables in Eq. 3.1 are as described in Chapter 2 Section 2.3.3, except β, which is the equivalent of β_2 in Eq. 2.5.

For all experiments in this chapter, parameter values were fixed as follows: $\alpha = -0.1, \beta = -0.1, \varepsilon = 0.5$. This gives a sinusoid-like oscillation whose amplitude is gradually dampened over time (see Figure 2.3).

The oscillator frequencies in the network were logarithmically distributed from 0.25Hz to 16Hz. The GFNN was stimulated by rhythmic

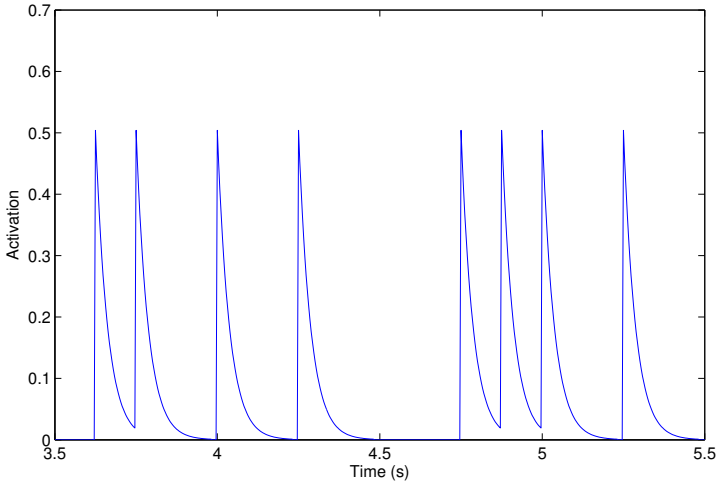


FIGURE 3.1: Example note onset time-series data.

time-series data in the form of a decay envelope on note onsets, synthesised from the symbolic data. All sequences in the corpus were synthesised at a tempo of 120bpm (2Hz), meaning that the metrical periodicities in the GFNN ranged from a demisemiquaver (32nd note) to a breve (double whole note).

3.2.2 LSTM

All experiments used the standard LSTM model with peephole connections enabled and the number of hidden LSTM blocks fixed at 10, with full recurrent connections. The number of blocks was chosen empirically as it provided reasonable prediction accuracy with plenty of potential for improvement, whilst minimising the computational complexity of the LSTM. Training was done by backpropagation through time (Werbos, 1990) using RProp$^-$ (Igel and Hüsken, 2000). During training, k-fold cross-validation (Kohavi, 1995) was used. In k-fold cross validation, the dataset is divided into k equal parts, or *folds*. A single fold is retained as the test data for testing the model, and the remaining k - 1 folds are used as training data. The cross-validation process is then repeated k times, with each of the k folds used exactly once as the test data. For our experiments k was fixed at 4. A maximum of 2500 training epochs was set per fold, but never reached as the training was halted early if no validation error improvement

was made in 20 epochs. An evaluation on the training data was also done, and found a mean percentage increase across all metrics of no more than 4.4%, indicating a good generalisation without over-fitting.

3.3 Experiments

3.3.1 Experimental Setup

The following experiments operate on monophonic symbolic music data. A corpus of 100 German folk songs from the Essen Folksong Collection (Schaffrath, 1995) was used.

All experiments were conducted in two steps, implementing the GFNN in MATLAB[1] using the fourth order Runge-Kutta integration method to solve the differential equations. The fourth order Runge-Kutta method provides a more accurate integration than the simpler Euler method by calculating a weighted average of four smaller integration steps. The LSTM in Python using the PyBrain[2] library.

3.3.2 Experiment 1: Pitch Prediction

The first experiment was designed to investigate the effect of adding metrical data from the GFNN to a pitch prediction task. Three LSTMs were designed, all of which were tasked with predicting pitch in the form of time-series data.

The absolute pitch values were abstracted to their relative scale degrees to keep the model simple in these initial experiments. Accidentals were encoded by adding or subtracting 0.5 from the scale degree and rests were encoded as 0 values. First, scale degree numbers, their onsets and offsets were inserted into the data stream and then the data was re-sampled using the zero-order hold method, such that one sample corresponded to a demisemiquaver. An example data stream can be seen in Figure 3.2.

[1] http://www.mathworks.co.uk/
[2] http://pybrain.org/

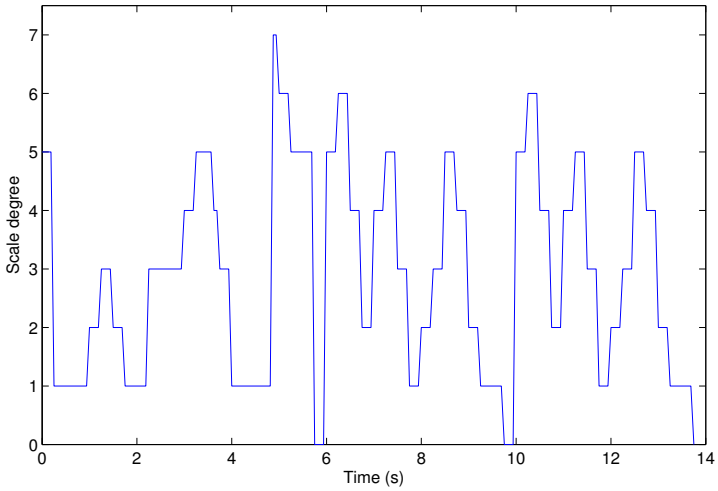


FIGURE 3.2: Example scale degree time-series data.

Since melody modelling with LSTMs alone has been well studied (see Eck and Schmidhuber, 2002; Franklin, 2006; Sturm et al., 2016), the first network (LSTM1a) was designed as a baseline to measure the impact of adding the GFNN as an input. Thus LSTM1a took no input from the GFNN, consisting solely of single input containing the time-series scale degree data from the corpus. Two further networks were constructed, one with 128 inputs (one for each oscillator in the GFNN; termed LSTM1b), and one with 8 inputs consisting of a filtered GFNN output (LSTM1c). LSTM1a, LSTM1b, and LSTM1c are illustrated in Figures 3.3 and 3.4.

As shown in Figure 2.4, a GFNN signal has relatively few resonant peaks of energy, therefore many oscillators would be irrelevant to the LSTM. Thus, it was hypothesised that the filtered output would make learning easier. The input to LSTM1c was filtered to retain the strongest resonant oscillations in the GFNN. The signal was averaged over the corpus and the oscillators with the greatest amplitude response over the final 25% of the piece were found. A spread of frequencies was ensured by ignoring frequencies if another near frequency was already included. The selected oscillators were then used for all sequences.

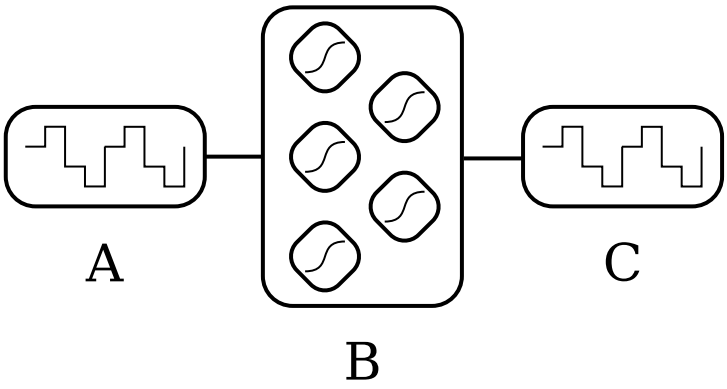


FIGURE 3.3: Network diagram for LSTM1a showing (A) scale degree sequence, (B) LSTM, and (C) scale degree prediction.

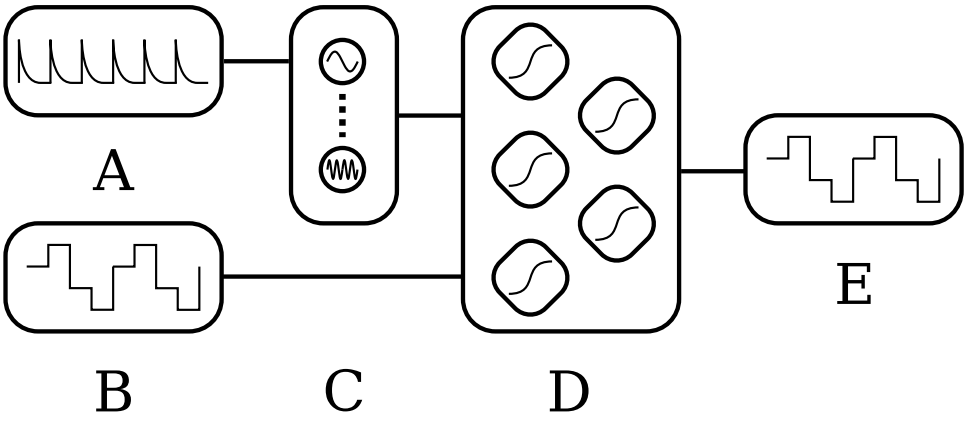


FIGURE 3.4: Network diagram for LSTM1b and LSTM1c showing (A) note onset sequence, (B) scale degree sequence, (C) GFNN, (D) LSTM, and (E) scale degree prediction.

Results

Networks were evaluated by activating each of them with the sequences in the corpus (ground truth). The networks were activated with the ground truth throughout the sequence, and for the last 75% of inputs the network output was compared to the target data.

The results have been evaluated using several metrics. Firstly the mean squared error (MSE) is reported, which is what the networks were optimised for during training. This provides a view of how close the output was to the target, with a lower number meaning higher accuracy. The next three results refer to the position of pitch changes using standard precision, recall, and F-measure. The following equations define how these metrics are calculated:

Network	MSE	Precision	Recall	F-measure	Accuracy
LSTM1a	0.75836	0.12154	0.34366	0.17425	0.67107
LSTM1b	0.74115	0.18644	**0.78908**	0.29838	0.47756
LSTM1c	**0.68866**	**0.22852**	0.70196	**0.34137**	**0.69459**

TABLE 3.1: Results of the pitch only experiment.

$$precision = \frac{correctly\,predicted\,onsets}{all\,predicted\,onsets} \tag{3.2}$$

$$recall = \frac{correctly\,predicted\,onsets}{ground\,truth\,onsets} \tag{3.3}$$

$$F = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{3.4}$$

Finally a pitch only metric is used, named “Accuracy". This has been calculated as a proportion of samples where the output scale degree matches the target value, where again higher is better. Output values were rounded to the nearest half before this comparison was made.

Pitch and rhythm are highly related, but have been singled out here to more fully understand the GFNNs effect on the network. The MSE and accuracy metrics represent timing and value, whereas the onset metrics of precision, recall, and F-measure represent timing only.

Table 3.1 shows the results tested against the validation data. The values shown are the mean values calculated over the 4 folds in the cross-validation.

The results show that the filtered input from the GFNN (LSTM1c) performed the best at predicting pitch and rhythm. However, there is a striking imbalance between the precision and recall scores for all networks, suggesting a chaotic output from the LSTMs, with too many events being triggered. This led to results that were not impressive overall, with pitch prediction improved, but rhythmic prediction performing poorly.

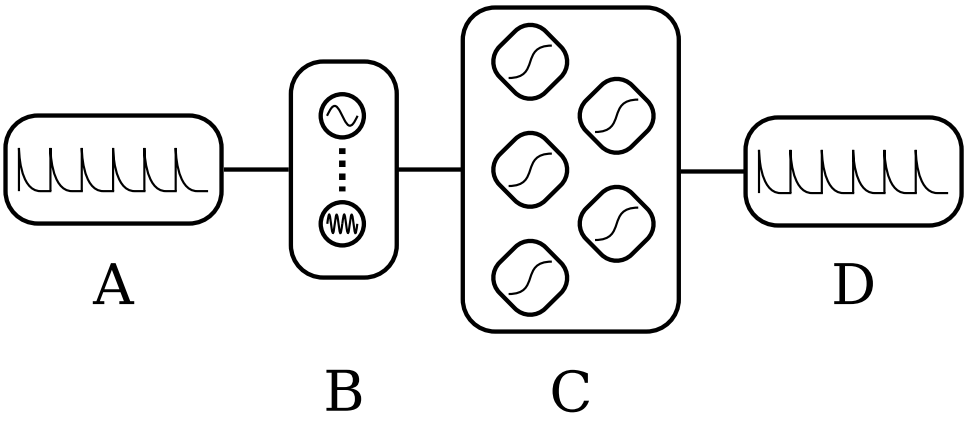


FIGURE 3.5: Network diagram for LSTM2a and LSTM2b showing (A) note onset sequence, (B) GFNN, (C) LSTM, and (D) note onset prediction.

Network	MSE	PCC	Precision	Recall	F-measure
LSTM2a	**0.01277**	**0.79400**	**0.82362**	**0.82769**	**0.82265**
LSTM2b	0.01380	0.77395	0.79411	0.81157	0.79564

TABLE 3.2: Results of the onset only experiment.

3.3.3 Experiment 2: Onset Prediction

The next experiment was designed to investigate if the GFNN did indeed contain useful rhythmic information for the LSTM to learn. A simpler task was designed where the LSTM had to predict the onset pattern used to stimulate the GFNN from the GFNN data only.

Two networks were created for this task: LSTM2a and LSTM2b. LSTM2a had a full GFNN input, and LSTM2b had the same filtered input from the previous experiment. Both networks had one output and were trained to reproduce the GFNN stimulus seen in Figure 3.1. A network diagram can be seen in Figure 3.5.

Results

Table 3.2 shows the results when the networks are tested against the validation data.

All networks were evaluated as in experiment 1, except there is no

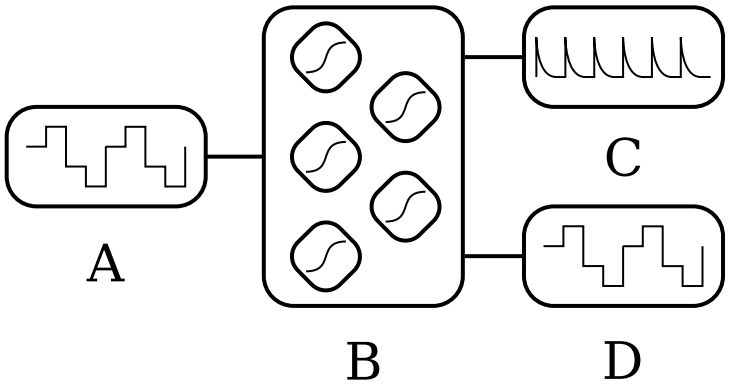


FIGURE 3.6: Network diagram for LSTM3a showing (A) scale degree sequence, (B) LSTM, (C) note onset prediction, and (D) scale degree prediction.

longer an accuracy metric and instead the Pearson product-moment correlation coefficient (PCC) is included. This gives a relative rather than absolute measure of how close the target and output signals match, with higher values representing closer matches. LSTM2a performed the best at this task in all metrics, however it is clear from the results that both LSTM2a and LSTM2b perform the tasks well.

The fact that LSTM2a outperformed LSTM2b shows that the LSTM network was able to train itself to ignore the noise produced by the GFNN. It also shows that the GFNN data contains useful information in the weaker resonances that the filtering process removed. The filtering process may have been too aggressive in this respect. However, having noted this, LSTM2b did not completely fail at the task, therefore a more permissive filtering technique may still produce better results than even LSTM2a.

3.3.4 Experiment 3: Onset and Pitch Prediction

Experiment 2 showed that the GFNN output can be used to reconstruct onsets. Experiment 3 was designed to investigate if tasking the network to directly predict the onsets could aid the prediction of pitch data. Therefore the tasks from experiments 1 and 2 were combined, resulting in LSTMs with two outputs: one for pitch and one for onsets.

Three LSTMs were constructed to conduct this experiment, following the same pattern as experiment 1: no GFNN input, full GFNN input, and

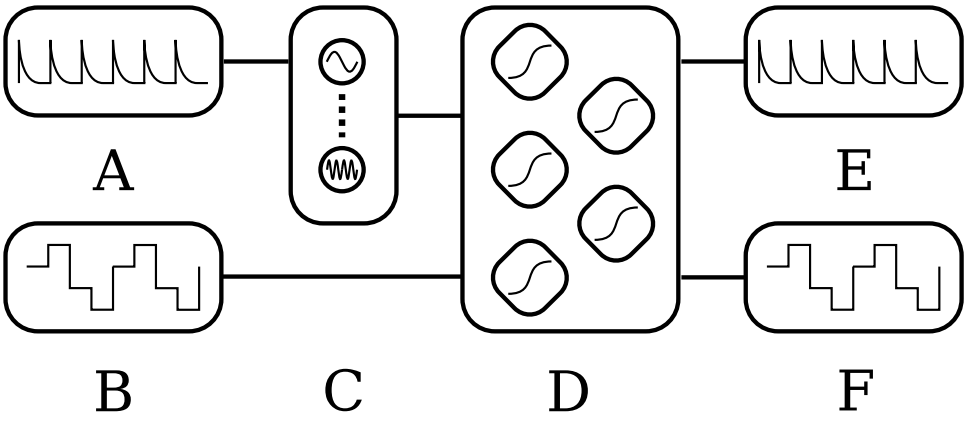


FIGURE 3.7: Network diagram for LSTM3b and LSTM3c showing (A) note onset sequence, (B) scale degree sequence, (C) GFNN, (D) LSTM, (E) note onset prediction, and (F) scale degree prediction.

Network	MSE	PCC	Precision	Recall	F-measure	Accuracy
LSTM3a	**7.26251**	0.23253	0.35655	0.06368	0.10233	0.64459
LSTM3b	7.34243	0.58499	**0.71622**	0.60717	0.65110	0.58371
LSTM3c	7.32129	**0.62905**	0.70480	**0.76750**	**0.72589**	**0.65755**

TABLE 3.3: Results of the pitch and onset experiment.

filtered input. Network diagrams can be seen in Figures 3.6 and 3.7.

Results

All networks were evaluated in the same way as experiments 1 and 2. The MSE metric was calculated for both outputs, PCC, precision, recall, and F-measure were only calculated for the onset pattern output, and accuracy was calculated only for the pitch output. Table 3.3 shows the results against the validation data.

The results show that LSTM3c was the best overall network. Whilst LSTM3a did score a better MSE, it scored very poorly on the onset prediction task. This shows that MSE may not be the best optimisation target during training. This may be due to the fact that the MSE is dominated by the scale degree error, as it has a greater numerical range than the onset error.

In experiment 1, all LSTMs suffered from poor precision scores. Judging by the onset scores for LSTM3b and LSTM3c, providing a GFNN input and directly modelling onsets through a predictive output led to a great improvement here.

In experiment 2, the fully connected LSTM2a outperformed the filtered LSTM2b on onset prediction, whereas in this experiment the reverse is true. This could be due to the increased complexity of the problem. The introduction of pitch modelling may have prevented the LSTM learning from the GFNN data as effectively, so that the filtering process was beneficial. Data from experiment 1 suggests that an improved filtering method may further improve results. Increasing the number of hidden LSTM blocks may also improve results for both LSTM3b and LSTM3c, as the network would be able to model more nonlinearities.

The accuracy scores for all networks are somewhat worse in this experiment when compared to experiment 1. However, the improved onset prediction indicates that LSTM3b and LSTM3c are more stable. More work is needed to investigate the behaviour of the pitch prediction to sequence accuracy and stability.

LSTM3c outperformed LSTM3a on the pitch prediction task, whilst also predicting stable onset patterns. This provides evidence that melody models can be improved by modelling metre.

3.4 Conclusions

In this chapter, a multi-layered network consisting of a metre perception layer (GFNN) and a temporal prediction layer (LSTM) was presented. The GFNN output, with its strong and weak nonlinear resonances at frequencies related to the pulse, can be interpreted as a perception of metre. The results show that providing this data from the GFNN helped to improve melody prediction with an LSTM. This supports the hypothesis that the LSTM is able to make use of the relatively long temporal resonance in the GFNN output, and therefore model more coherent long-term structures.

In all cases GFNNs improved the performance of pitch and onset prediction. Given the improvements to the onset prediction, modelling pitch and onsets can be seen to be the best overall approach. Additionally, the best results were achieved by filtering the GFNN output. However, experiment 2 showed that there is important information in the full GFNN signal which is lost through the filtering method adopted here. In addition, this filtering method may not be a good solution when dealing with varying tempos or expressive timing, as it introduces an assumption of a metrically homogeneous corpus. Thus, two tasks for future work are to develop filtering that improves performance and supports tempo variation as well as exploring representations and learning methods that combine stable onset prediction with sequence accuracy.

Both Eck and Schmidhuber's (Eck, 2002) and Coca et al.'s (Coca, Correa and Zhao, 2013) LSTMs either operate on note-by-note data, or quantised time-series data. By inputting metrical data, our system can be extended to work with real time data, as opposed to the metrically quantised data used in this chapter. These initial experiments give some indication that better melody models can be created by modelling metrical structures.

By using an oscillator network to track the metrical structure of performance data, a move towards real-time processing of audio signals can be made. Furthermore, the loop in the GFNN-LSTM can be closed, creating an expressive, metrically aware, and generative real-time model.

Chapter 4

Expressive Rhythm Modelling

4.1 Introduction

As previously discussed in Chapter 2, human musical performance rarely contains perfectly periodic metre. Musicians employ many expressive devices including deviate from the ideal pulse in subtly complex ways (Räsänen et al., 2015; Clarke, 2001). However, automatically tracking varying tempo and expressive timing is still an open problem within MIR (Grosche, Müller and Sapp, 2010; Holzapfel et al., 2012).

In this chapter, this issue is addressed through a machine learning study of the modelling and processing of expressive rhythms. The GFNN-LSTM model developed in Chapter 3 is adapted for use with an audio signal in a time-series prediction task. GFNNs have been applied successfully to a range of music perception problems including those with syncopated and polyrhythmic stimuli (see Angelis et al., 2013; Velasco and Large, 2011). The GFNN's entrainment properties allow each oscillator to phase shift, resulting in changes to their observed frequencies. This makes them good candidates for solving the expressive timing problem.

4.1.1 Contributions

This chapter presents the first study of a GFNN with audio data and the first expressive timing study with a GFNN.

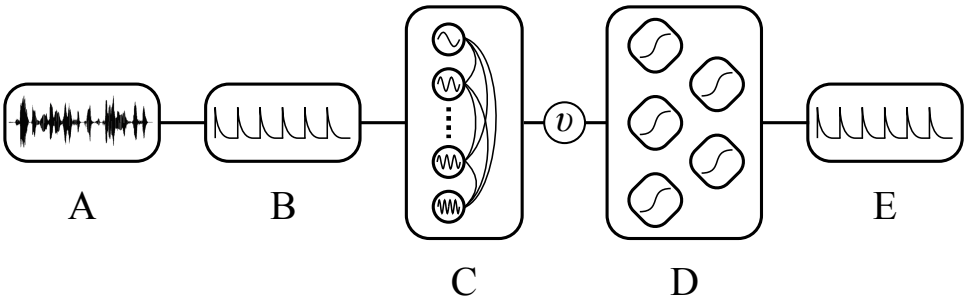


FIGURE 4.1: An overview of the GFNN-LSTM system showing (A) audio input, (B) mid-level representation, (C) GFNN, (D) LSTM, and (E) rhythm prediction output. The variable ν can be a mean field function or full connectivity.

The following hypothesis is explored: a GFNN-LSTM is enabled to make use of the entrainment properties in the GFNN output, and therefore model rhythmic expectation with the LSTM, and thus the system makes better rhythmic predictions. A system such as this could be used in a multitude of analytic and generative scenarios, including live performance applications.

Part of this chapter has been published in (Lambert, Weyde and Armstrong, 2015b) and (Elmsley, Weyde and Armstrong, 2017).

4.2 Models

4.2.1 Overview

The results from the Chapter 3 gave some indication that better melody models can be created by modelling metrical structures with a GFNN.

The system presented here is a significant step beyond this. For the first time audio data is incorporated, which opens the system up for a much wider set of live and off-line applications, but comes with its own set of new problems to solve. Unlike the metrically quantised data used in the experiments in Chapter 3, this data contains varying tempos, and is sampled at an arbitrary sample rate. Furthermore, this system experiments with enabling Hebbian learning within the GFNN in the hope this will enable stronger metric hierarchies and faster entrainment responses to emerge from the nonlinear resonance.

The aim of the experiment detailed below was to train a GFNN-LSTM to predict expressive rhythmic events. The system takes audio data as input and outputs an event activation function. It operates in a number of stages which are detailed below, and a schematic is provided in Figure 4.1.

A subset of the Chopin Mazurka dataset[1] (MAZ) was used. MAZ is a collection of audio recordings comprising on average 50 performances of each of Chopin's Mazurkas. The pieces are all expressively performed by various performers and vary in tempo and dynamics throughout each performance. However, the pieces are all within the same genre and are all performed on the piano, making drawing conclusions about the rhythmic aspects more valid. A subset of 50 excerpts, each 40 seconds long, was made by randomly choosing annotated excerpts of full pieces and slicing 40 seconds worth of data.

4.2.2 Mid-level representation

When processing audio data for rhythmic events, it is common to first transform the audio signal into a more rhythmically meaningful representation from which these events can be inferred. This representation could be extracted note onsets in binary form, or a continuous function that exhibits peaks at likely onset locations (Scheirer, 1998). These functions are called *onset detection functions* and their outputs are known as *mid-level representations*.

Since expressively rich audio is used here, an onset detection function which is sensitive both to sharp and soft attack events is needed. From Bello et al.'s (2005) tutorial on onset detection in music signals, the complex spectral difference (CSD) onset detection function was selected. This detection function emphasises note onsets by analysing the extent to which the spectral properties of the signal at the onset of musical events are changing. The function operates in the complex domain of a frequency spectrum where note onsets are predicted to occur as a result of significant changes in the magnitude and/or phase spectra. By considering both magnitude and

[1] http://www.mazurka.org.uk/

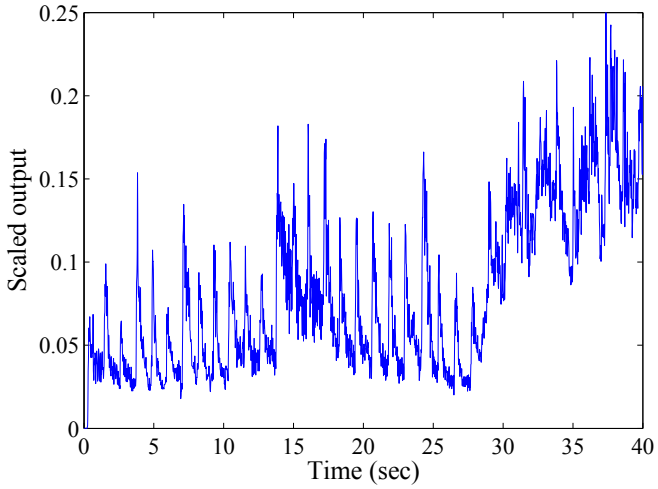


FIGURE 4.2: An example complex spectral difference output.

phase spectra, CSD can capture soft changes in pitch and hard rhythmic events.

Figure 4.2 displays an example output of CSD. Here the output range has been scaled to a 0 to 0.25 scale for input into the GFNN. This continuous function output can be converted into binary onset data by using suitable threshold levels for peak picking. A sample rate of 86.025Hz was used, which has been found to yield accurate detection results (Davies and Plumbley, 2007).

4.2.3 GFNN layer

The GFNN was implemented in MATLAB using the GrFNN Toolbox (Large et al., 2014). It consisted of 192 oscillators, logarithmically distributed with natural frequencies in a rhythmic range of 0.5Hz to 8Hz. The GFNN was stimulated by rhythmic time-series data in the form of the mid-level representation the audio data.

Two parameter sets were selected for the oscillators themselves, obtained from the examples in the GrFNN Toolbox. These different parameters affect the way the oscillators behave. The first parameter set puts the oscillator at the bifurcation point between damped and spontaneous oscillation. This is hereby termed 'critical mode', as the oscillator resonates with

input, but the amplitude slowly decays over time in the absence of input: $\alpha = 0, \beta_1 = \beta_2 = -1, \delta_1 = \delta_2 = 0, \epsilon = 1$. By setting $\delta_1 = 1$, the second parameter set is defined: 'detune mode'. δ_1 affects the imaginary plane only, which is the oscillator's inhibitor. Since the driving frequency parameter (ω) is also in the imaginary plane, δ_1 allows the oscillator to change its natural frequency more freely, especially in response to strong stimuli. As a result, this could allow for improved tracking of tempo changes.

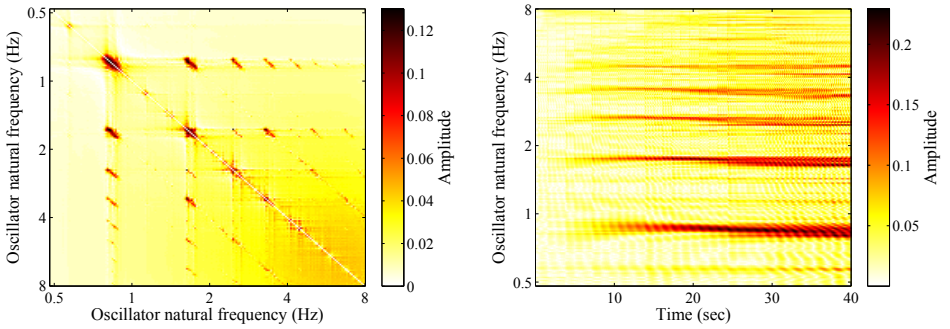
Hebbian Learning

Three different approaches to performing the Hebbian learning in the GFNN layer have also been selected. The baseline system simply has no connectivity between oscillators and therefore no learning activated at all. This is included so that the effect (if any) that learning in the GFNN layer has on the overall predictions of the system can be measured.

The first Hebbian approach is to activate online learning with the following parameters: $\lambda = 0, \mu_1 = -1, \mu_2 = -50, \epsilon_c = 4$ and $\kappa = 1$. Under these parameters, the network should learn connections between related frequencies as they resonate to the stimulus. The second approach is to once again activate the online learning, but starting from an initial state in the connection matrix. The reasoning for including this approach is explained below.

Figure 4.3a shows an example connection matrix that is learned from one particular excerpt. Taken together, the behaviour of the GFNN over time and the learned connection matrix enables a similar analytical method to Inner Metrical Analysis (IMA) (Nestke and Noll, 2001; see Chapter 2 Section 2.2.5), but is a continuous-time model, whereas IMA uses discrete, metrically quantised time steps.

Figure 4.3a shows that high order hierarchical relationships have been learned by the oscillators. However, these relationships are only valid for the particular excerpt with which they have been learned: they are localised to specific fixed frequencies rather than being a generalisation. This has both positive and negative aspects. On the positive side, the connection



(A) Connection matrix amplitudes. (B) Oscillator amplitudes over time.

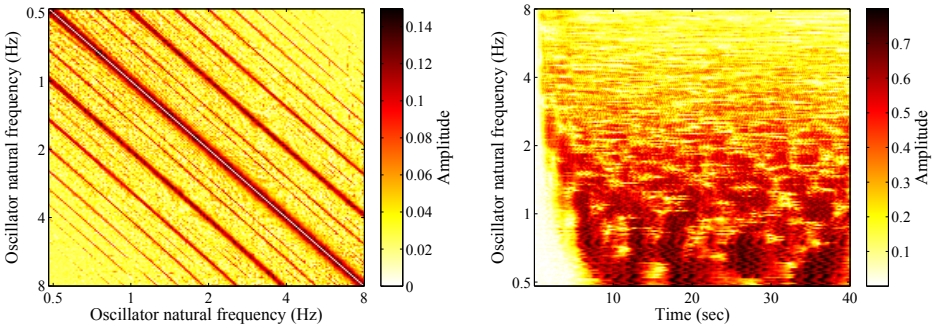
FIGURE 4.3: GFNN connection matrix and oscillator amplitudes with online learning.

matrix can be used as a way of analysing the frequency responses of the network. However, applying this connection matrix in a prediction task would not be that useful, as any rhythm outside this particular tempo with different local metres would not exhibit predictable behaviour.

By activating the learning rule when the oscillators are set to operate in limit cycle mode (a spontaneous oscillation in the absence of input), the internal connections can be learned in the absence of any stimulus. The resulting connection matrix is shown in Figure 4.4a. This provides a much more general state for the connection matrix to be in and potentially overcomes the limitations of the fixed frequency connections learned in Figure 4.3a.

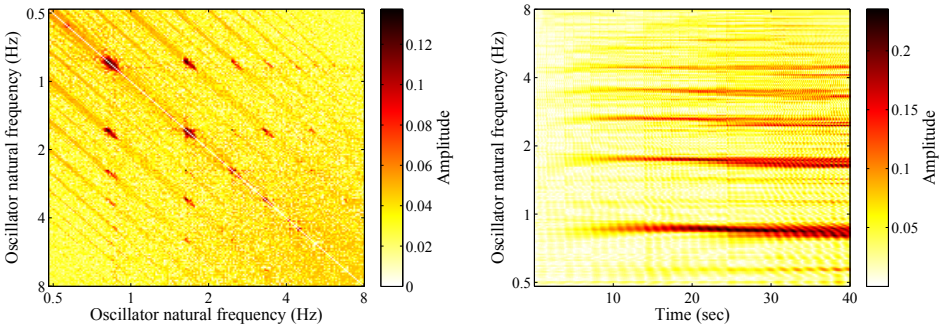
However, in the network response (Figure 4.4b) it can be seen that fixing the connections at this state results in a much noisier output of the GFNN. Resonances do build up very quickly, but the resulting oscillator output does not resemble the structured hierarchy found in Figure 4.3b. Essentially there are too many connections in the GFNN, leading to a *cascade* effect where a strong resonant response to the stimulus is transferred down the frequency gradient in a wave. This amounts to a GFNN output which is too noisy to be used for any subsequent machine learning tasks.

This can be counteracted by keeping online learning activated and also setting the initial connectivity state with that learned in limit cycle mode. The resulting connection matrix can be seen in Figure 4.5a. The matrix exhibits strong local connections at frequencies specific to the excerpt, but



(A) Connection matrix amplitudes. (B) Oscillator amplitudes over time.

FIGURE 4.4: GFNN connection matrix learned in limit cycle mode and oscillator amplitudes with fixed connections.



(A) Connection matrix amplitudes. (B) Oscillator amplitudes over time.

FIGURE 4.5: GFNN connection matrix and oscillator amplitudes with online learning and an initial state from Figure 4.4a.

more general high order connections are still present in the matrix. The amplitude response of the network (Figure 4.5b) shows a clear hierarchy of frequencies whilst also displaying a fast resonance response and less noise. This is the third approach to Hebbian learning taken in this chapter, which is termed *InitOnline*.

In some initial experimentation it was found that with Hebbian learning activated, the differential equations that drive the connection matrix can become unstable and result in an infinite magnitude. To ensure greater stability in the system, the connections in the connection matrix were limited to have a magnitude less than $\frac{1}{\sqrt{\epsilon_c}}$ (0.5 in these experiments). All stimuli was also rescaled to be in the range $0 <= x(t) <= 0.25$.

4.2.4 LSTM layer

The LSTM was implemented in Python using the PyBrain library (Schaul et al., 2010). For each variation of the GFNN, two LSTM topologies were trained. The first had 192 linear inputs, one for each oscillator in the GFNN, which took the real part of each oscillator's output. This is termed the *Full* LSTM. The real part of the canonical oscillation is a representation of excitatory neural population; by discarding the imaginary part, a meaningful representation of the oscillation is still retained, the simplicity of the input to the LSTM is increased (Large, Herrera and Velasco, 2015). The second topology took only one linear input, which consisted of the mean field of the real-valued GFNN. The mean field reduces the dimensionality of the input whilst retaining frequency information within the signal. This is termed the *Mean* LSTM.

All networks used the standard LSTM model with peephole connections enabled. The number of hidden LSTM blocks in the hidden layer was fixed at 10, with full recurrent connections. The number of blocks was chosen based on previous results which found it to provide reasonable prediction accuracy, whilst minimising the computational complexity of the LSTM (see 3.2.2).

All networks had one single linear output, which serves as a rhythmic event predictor. The target data used was the output of the onset detection algorithm, where the samples were shifted so that the network was predicting what should happen next. The input and target data was normalised before training.

As in Chapter 3, training was done by backpropagation through time (Werbos, 1990) using RProp$^-$ (Igel and Hüsken, 2000). k-fold cross-validation was used (Kohavi, 1995), where k was fixed at 5. A maximum of 350 training epochs was set per fold. Training stopped when the total error had not improved for 20 epochs, or when this limit was reached, whichever came sooner.

4.3 Results

4.3.1 Evaluation

This experiment was designed to discover whether the GFNN-LSTM is able to make good predictions in terms of the rhythm. Therefore the system was evaluated on its ability to predict expressively timed rhythmic events, whilst varying the parameters of the GFNN and connectivity. An explicit evaluation of the system's production of expressive timing is not made here, but an implicit evaluation of the tracking and representation of expressive timing is made, as it is reasonable to assume that a meaningful internal representation of metrical structure is needed for accurate predictions.

The results have been evaluated using several metrics. The first three results refer to the binary prediction of rhythmic events of pitch changes using the standard information retrieval metrics of precision, recall, and F-measure, where higher values are better. Events are predicted using a gradient threshold of the output data. The threshold looks for peaks in the signal by tracking gradient changes from positive to negative. When this gradient change occurs, an onset has taken place and is recorded as such.

These events were subject to a tolerance window of $\pm$58.1ms. This means that an onset can occur within this time window and still be deemed a true positive. At the sample rate used in this experiment, this equates to 5 samples either side of an event. It was also insured that neither the target nor the output can have onsets faster than a rate of 16Hz, which is largely considered to be the limit of where rhythm starts to be perceived as pitch (Large, 2010). These are limitations to the evaluation method, but since the prediction of rhythmic structures is of primary interest, not the production of expressive micro-timing, they are acceptable concessions.

The mean squared error (MSE) and the Pearson product-moment correlation coefficient (PCC) of the output signals are also provided, which provide overall similarity measures.

For all metrics the first 5 seconds of output by the network are ignored, making the evaluation only on the final 35 seconds of predictions.

4.3.2 Results

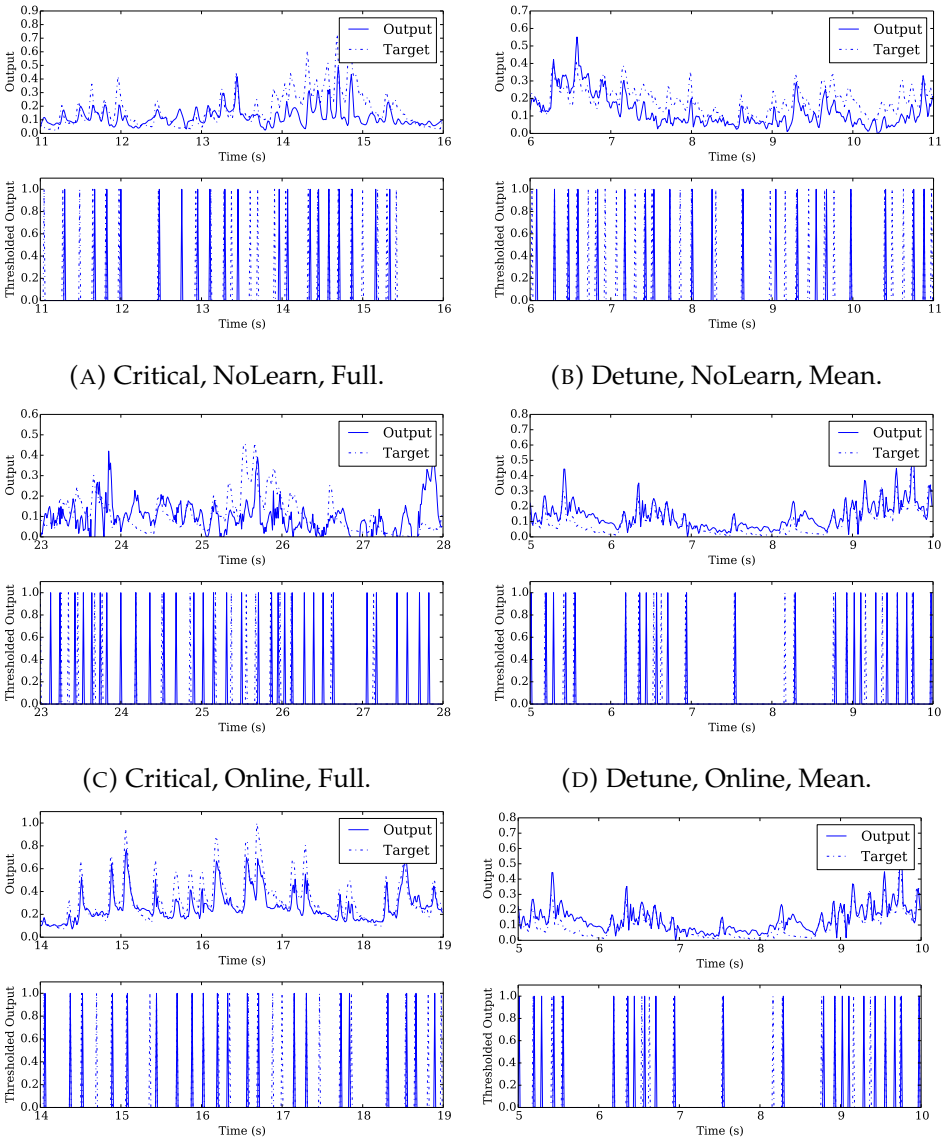
Table 4.1 and Table 4.2 display the results of the experiment, Figure 4.6 shows example outputs from various trained networks over time. The top figures show the continuous output set against the training data, whereas the bottom figures show extracted events after the gradient-based threshold detailed in the previous section has been applied.

Learning	LSTM	Precision	Recall	F-measure	MSE	PCC
None	Full	0.6114 (0.035)	0.6182 (0.034)	0.6059 (0.021)	0.0295 (0.003)	0.5296 (0.078)
None	Mean	0.6878 (0.100)	0.6883 (0.067)	0.6823 (0.081)	0.0294 (0.004)	0.6880 (0.184)
Online	Full	0.5637 (0.043)	0.6185 (0.076)	0.5798 (0.042)	**0.0276 (0.004)**	0.4326 (0.117)
Online	Mean	0.6862 (0.039)	0.6401 (0.050)	0.6548 (0.042)	0.0277 (0.001)	0.6600 (0.071)
InitOnline	Full	0.5982 (0.055)	0.6230 (0.041)	0.6000 (0.018)	0.0287 (0.001)	0.4711 (0.050)
InitOnline	Mean	**0.7032 (0.031)**	**0.6979 (0.041)**	**0.6958 (0.036)**	0.0300 (0.001)	**0.7363 (0.054)**

TABLE 4.1: Critical oscillation mode results. The values show the mean results calculated on the validation data. The value in brackets denotes the standard deviation.

Learning	LSTM	Precision	Recall	F-measure	MSE	PCC
None	Full	0.5972 (0.027)	0.6508 (0.036)	0.6161 (0.027)	0.0299 (0.003)	0.5088 (0.065)
None	Mean	0.7208 (0.058)	0.6891 (0.069)	0.6959 (0.057)	0.0306 (0.004)	**0.7609 (0.093)**
Online	Full	0.5831 (0.044)	0.6443 (0.067)	0.6020 (0.015)	0.0308 (0.002)	0.4978 (0.051)
Online	Mean	0.6943 (0.028)	0.6911 (0.045)	0.6866 (0.034)	0.0291 (0.004)	0.6855 (0.062)
InitOnline	Full	0.5666 (0.023)	0.6787 (0.033)	0.6114 (0.013)	**0.0286 (0.002)**	0.6341 (0.036)
InitOnline	Mean	**0.7239 (0.013)**	**0.7178 (0.061)**	**0.7142 (0.033)**	0.0295 (0.003)	0.7123 (0.062)

TABLE 4.2: Detune oscillation mode results. The values are as in Table 4.1.



(A) Critical, NoLearn, Full.

(B) Detune, NoLearn, Mean.

(C) Critical, Online, Full.

(D) Detune, Online, Mean.

(E) Critical, InitOnline, Mean.

(F) Detune, InitOnline, Mean.

FIGURE 4.6: Example outputs from various trained networks over time.

These numerical metrics and visual figures provide some indication of how well the system is capturing the rhythmic structures. However, this information may be better understood by listening to the predicted rhythms. To this end, the reader is invited to visit this chapter's accompanying website[2], where a collection of audio examples has been assembled for each network's target and output data.

[2]http://andyroid.co.uk/research/gfnn_lstm_rhythm_prediction

4.3.3 Discussion

The best overall GFNN-LSTM for expressive rhythm prediction incorporates detune oscillators, online learning with initial generic connections in the GFNN layer, and mean field connections.

From the results it can be seen that the mean field networks always outperformed the GFNN-LSTM with a full connection. This could be due to the mean field being able to capture the most resonant frequencies, whilst filtering out the noise of some less resonant frequencies. The resulting signal to the LSTM would therefore be more relevant for predicting rhythmic events. However, this may also be due to the limited number of LSTM blocks in each network forming a bottleneck in the fully connected networks. Increasing number of hidden LSTM blocks may mitigate this limitation.

One downside of the mean field networks is that drastically reducing the dimensionality in this way could cause either over or under-fitting. It can be seen in the results that whilst performance improved in all cases using the mean field, the standard deviation also increased. This means there was a greater range of performances between the folds and could possibly indicate some networks being trained to local optima. During training it was observed that the mean field networks took many more epochs for errors to converge. This could possibly be addressed by using sub-band mean fields, or some other method to reduce the dimensionality between layers.

In all cases, the detune oscillators outperformed the critical oscillators. In most cases the standard deviation was also decreased by using detune oscillators. This can be attributed to the greater amount of change in the imaginary part of the oscillator (inhibitory neural population). Tempo changes can be tracked as an entrainment process between a local population of oscillators in the network. Where there is a local area of strong resonance the oscillators will take on frequencies very near to one another. As the stimulus frequency changes, this local area will be able to follow it, moving the local resonance area along the frequency gradient.

When the results of this experiment were initially calculated, higher F-measures were observed than are reported here. Upon closer inspection a large discrepancy between the precision and recall scores was found, indicating a large number of false positives. It is important to choose a sensible threshold to control the number of events being output by the system. An overwhelming number of emitted events will result in a high recall and low precision. The recall value may be so high as to dominate the F-measure, falsely inflating it. Thus threshold values were chosen that optimise the balance between precision and recall values, resulting in a fairer, if a little lower, F-measures. This gives an indication that evaluating on F-measure alone does not always give the best overall score.

It is interesting to note that applying online learning to the network did improve the overall MSE of the signal, but the F-measure actually performed worse in all cases. Perhaps an adaptive threshold may be the solution to this problem, as the GFNN signal changes in response to previous inputs and the connections begin to form.

In Chapter 3, the best GFNN-LSTM achieved a rhythm prediction mean F-measure of 82.2%. Comparing this with the 71.4% mean achieved here may at first seem a little underwhelming. However, these new results represent a significant change in the signal input, and reflect the added difficulty of the task. In Chapter 3 symbolic data was used at a fixed tempo and without expressive variation, whereas this study is undertaken on audio data performed in an expressive way. The overall best single fold (Detune oscillators, InitOnline connections, and Mean input) was achieving an F-measure of 77.2%, which is extremely promising.

This approach to rhythm prediction is a novel one, which makes a comparative statement difficult to make. However, similarities can be drawn between this system and an MIR beat tracker; both are processing audio systems to extract rhythmic predictions. The best MIREX beat tracker in 2016 scored an F-measure of 73.9% (see Korzeniowski, Böck and Widmer, 2014) on the same dataset used above. This system has a similar design to the GFNN-LSTM: spectrogram change information is input into an LSTM

which, is trained to predict beat events. These predictions are processed with a bank of resonating comb filters to help smooth the output. Whilst a direct comparison cannot be made, as expressive rhythm events are predicted not pulse events, a comparison with this system is helpful for two reasons. Firstly it shows a similar method is producing state-of-the-art results in a field where comparisons are easier to make, and secondly it hints that the system is performing well on this dataset.

4.4 Conclusions

This chapter detailed a multi-layered recurrent neural network model for expressively timed rhythmic perception and prediction. The model consists of a perception layer, provided by a GFNN, and a prediction layer provided by an LSTM. The GFNN-LSTM was evaluated on a dataset selected for its expressive timing qualities and it was found to perform at a comparable standard to a previous experiment undertaken on symbolic data.

The system's performance is comparable to state-of-the-art beat tracking systems. For the purposes of rhythm generation, the F-measure results reported here are already in a good range. Greater values may lead to too predictable and repetitive rhythms, lacking in the novelty expected in human expressive music. On the other hand, lower values may make the generated rhythms too random and irregular, so that they may even not be perceived as rhythmic at all. To make any firm conclusions on this, formal listening tests would need to be conducted, based on the generated rhythms from the system. This is left for future work.

Another interesting avenue for future analysis is to explicitly evaluate the system's production of expressive timing. To achieve this, the tolerance window can be removed and time differences between the target and output events with a steady idealised pulse can be analysed.

By using an oscillator network to track the metrical structure of expressively timed audio data, processing the metrical structures of audio signals is enabled in realtime.

Chapter 5

Perceiving Dynamic Pulse with GFNNs

5.1 Introduction

Previous work on utilising GFNNs in an MIR context has shown promising results for computationally difficult rhythms such as syncopated rhythms where the pulse frequency may be completely absent from the signal's spectrum (Velasco and Large, 2011; Large, Herrera and Velasco, 2015), and polyrhythms where there is more than one pulse candidate (Angelis et al., 2013).

In previous chapters, GFNNs were used as part of a machine learning signal processing chain to perform rhythm and melody prediction. In Chapter 4, an expressive rhythm prediction experiment showed comparable accuracy to the state-of-the-art beat trackers. However, it was not clear from this experiment how well the GFNN was capturing changing metrical structures when the pulse frequency fluctuates.

The exact contribution of the GFNN to the holistic GFNN-LSTM model is unknown, as an explicit evaluation of that layer was not done in previous chapters. To address this, the investigation of this thesis will now move away from predicting rhythm, as was the case in previous chapters, towards analysing the *pulse* prediction capabilities of the GFNN alone. The

hypothesis behind this move is that any improvements that can be made in the GFNN layer will have a direct and measurable impact of the accuracy of the LSTM layer's predictions.

In this chapter, the results of an experiment with GFNNs are presented, partially reproducing the results from Velasco and Large's (2011) last major MIR application of a GFNN, and Large, Herrera and Velasco's (2015) more recent neuroscientific contribution. Also included is a new class of rhythms where tempos are changing.

There is one major difference between the results presented here and those presented in the past. Previous studies have placed a focus on the frequencies contained in the GFNN's output, often reporting the results in the form of a magnitude spectrum, and thus omitting phase information. When dealing with pulse and metre perception, phase is an integral part as it constitutes the difference between entraining to on-beats, off-beats, or something in-between. This is especially true when dealing with changing tempo. Therefore in this chapter a greater evaluation focus is placed on phase accuracy. A new quantitative evaluation metric is introduced here, named the Weighted Phase Output (WPO), which enables direct comparison between different GFNNs of different dimensions, stimulated by different rhythm datasets.

5.1.1 Contributions

This chapter contributes the first analysis of a GFNN's performance when dealing with changing tempo, and an extended evaluation of syncopated rhythms previously studied in the literature. In the literature, the evaluation of GFNNs' pulse finding predictions in terms of phase has never been attempted. To achieve this, a new way to evaluate GFNNs is introduced, based on a new type of GFNN output named here as *weighted phase output* (WPO).

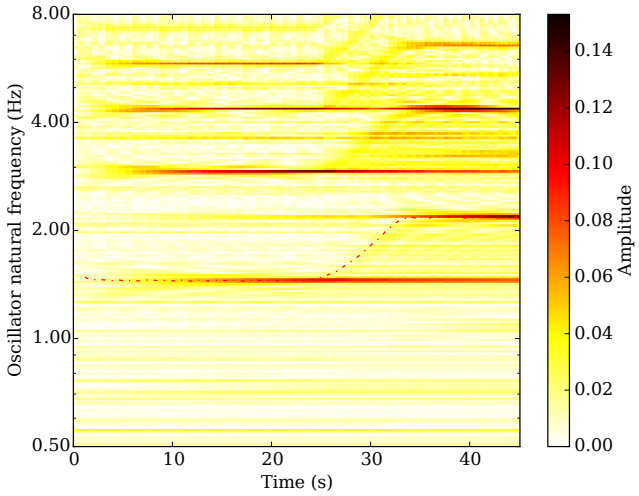


FIGURE 5.1: Amplitudes of oscillators over time. The dashed line shows stimulus frequency. The stimulus itself is shown in Figure 5.2. There is an accelerando after approximately 25s.

This chapter also uses the open source PyGFNN[1] library for all its experiments, a newly developed python library containing a GFNN and Runge-Kutta integrators implemented, maintained, and released by the author.

Part of this chapter has been published in (Lambert, Weyde and Armstrong, 2016a).

5.2 Phase Based Evaluation

Thus far in the literature, evaluation of GFNNs has not considered phase information. The phase of oscillations is an important output of a GFNN; in relation to pulse it constitutes the difference between predicting at the correct pulse times, or in the worst-case predicting the off-beats. In music with many off-beat events this evaluation may miss important aspects of the music.

Phase and frequency are interlinked in that frequency can be expressed as a rate of phase change and indeed the canonical oscillators' entrainment properties are brought about by phase shifts. Since the state of a canonical oscillator is represented by a complex number, both amplitude and phase

[1] https://github.com/andyr0id/PyGFNN

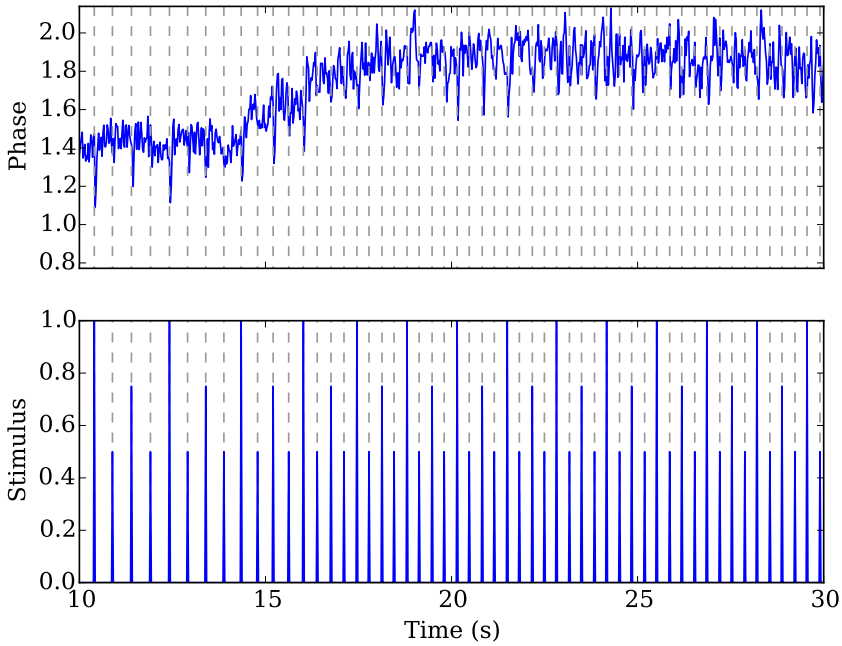


FIGURE 5.2: WPO of the GFNN over time. The stimulus is the same as Figure 5.1.

can be calculated instantaneously by taking the magnitude ($r = |z|$), and angle ($\varphi = arg(z)$) respectively. Eq. 5.1 defines WPO (Φ), and Figure 5.2 shows the WPO over time.

$$\Phi = \sum_{i=0}^{N} r_i \varphi_i \tag{5.1}$$

In Figure 5.2, it can be seen that the WPO signal is a fairly complex signal, this is due to the high-dimensionality and frequency spread of the oscillators in a GFNN. However, the signal does contain some clear indicators of pulse perception; the dips in the signal do tend to occur at pulse positions.

To achieve a clearer quantitative measure of how much the WPO is matching the pulse, one further step is required: a comparison of WPO with a ground truth signal. To create the ground truth, a phase signal similar to an inverted beat-pointer model (Whiteley, Cemgil and Godsill, 2006) is used.

While a beat-pointer model linearly falls from 1 to 0 over the duration of one beat, the inverted signal rises from 0 to 1 to represent phase growing

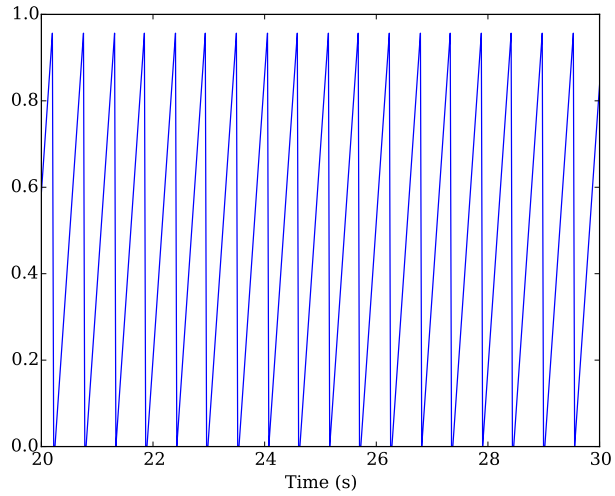


FIGURE 5.3: An example of the inverted beat-pointer data used as the correlation target in WPO correlation.

from 0 to 2π in an oscillation. An example can be seen in Figure 5.3. The entrainment behaviour of the canonical oscillators will cause phase shifts in the network, therefore the phase output should align to the phase of the input.

To make a quantitative comparison the Pearson product-moment correlation coefficient (PCC) of the two signals is calculated. This gives a relative, linear, mean-free measure of how close the target and output signals match. A value of 1 represents a perfect correlation, whereas -1 indicates an anti-phase relationship. Since the GFNN operates on more than one metrical level, high levels of correlation cannot be expected, and even a small positive correlation would be indicative of a good frequency and phase response, as some of the signal represents other metrical levels.

5.3 Experiment

5.3.1 Method

A pulse detection experiment was performed to evaluate the GFNN output with WPO. The aim of the experiment was to evaluate the GFNN's ability to track changing tempo.

For comparison with previously published results, two of the same rhythms used by Velasco and Large for use in this experiment have been selected. The first is an isochronous pulse and the second is the more difficult 'son clave' rhythm. These rhythms were supplemented by rhythms from the more recent Large, Herrera and Velasco (2015) paper. The rhythms are in varying levels of complexity (1-4), varied by manipulating the number of events falling on on-beats and off-beats. A level 1 rhythm contains one off-beat event, level 2 contains two off-beat events and so forth. Two level 1 patterns, two level 2 patterns, two level 3 patterns, and four level 4 patterns were used.

To test dynamic pulses, two new stimulus rhythms were included exhibiting accelerando and ritardando behaviour.

All these rhythms were tested at 20 different tempos, selected randomly from a range 80-160bpm. None of the networks tested had any internal connections activated, fixed or otherwise ($c_{ij} = 0$). An experiment to study the effect of connections was left for future work.

Similar oscillator parameters to Velasco and Large's (2011) experiment were chosen ($\alpha = 0, \beta_1 = \beta_2 = -1, \delta_1 = \delta_2 = 0$ and $\varepsilon = 1$). This is known as the *critical* parameter regime, poised between damped and spontaneous oscillation. Velasco and Large's GFNN density of 48opo was retained, but the number of octaves was reduced to 4 (0.5-8Hz, logarithmically distributed), rather than the 6 octaves (0.25-16Hz) used in their study. This equated to 193 oscillators in total. This reduction did not affect the results and is more in line with Large's later GFNN frequency ranges (see Large, Herrera and Velasco, 2015).

In summary, the experiment consisted of 5 stimulus categories, 20 tempos per category and 3 networks. There are two initial evaluations, one for comparison with previous work with GFNNs, and the second is testing dynamic pulses with accelerando and ritardando. The experiment used the open-source *PyGFNN* python library created by the author, which contains a GFNN implementation.

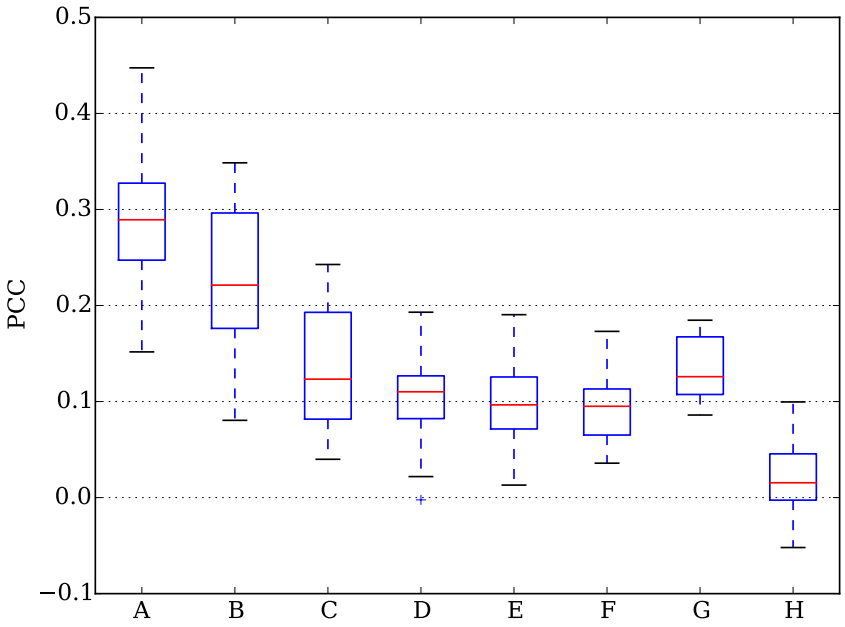


FIGURE 5.4: Box and Whisker plots of the PCC results. Rhythms are as follows: A) Isochronous, B-E) Large, Herrera and Velasco (2015) levels 1-4, F) Accelerando, G) Ritardando, and H) Son Clave. Boxes represent the first and third quartiles, the red band is the median, and whiskers represent maximum and minimum non-outliers.

5.4 Results

Figure 5.4 shows the results for the pulse detection experiment described above in the form of a box plot.

The GFNN is found to be effective for tracking and predicting pulse in isochronous rhythms (A), as it shows a good correlation against the inverted beat-pointer. This suggests relevant resonances have enough strength to dominate any interference from other oscillators in the network.

The first Large, Herrera and Velasco (2015) rhythm level (B) also performs well, though it does not have as stronger correlation as (A), which is to be expected for non-isochronous rhythms. Level 2 (C) shows the median correlation fall by close to 0.1, however the upper quartile and maximum bounds are still in a good range. Levels 3 and 4 (D and E) perform around the same with medians hovering around the 0.1 range. This is not evidence of a strong correlation and suggests that the extra syncopation present in

these rhythms is causing the oscillators to phase shift and predict off-beat pulse events rather than on-beats.

Both the tempo change rhythms accelerando (F) and ritardando (G) also perform poorly. Since these rhythms are isochronous except for the tempo change this indicates that interference is taking place in the network, causing the WPO to become noisy. An example of this effect can be seen in Figure 5.1, a memory of the previous resonance still persists and causes the new growing resonance to be lost in the signal. Interestingly the ritardando rhythms do perform better than accelerandos. This may be due to the oscillator model having a frequency-dependant damping rate. It was found that higher frequency oscillators lose amplitude faster than those with lower frequency. This mitigates the interference effect slightly, causing a small increase in performance.

In the Son Clave results (H) the network performs poorly, with only a small positive correlation being reported. A poorer result here was expected due to the difficulty of this rhythm, but a result this low indicates that the GFNN is not capturing the metrical structure of this rhythm effectively.

5.5 Conclusions

In this chapter the GFNN's ability to capture metrical structure under changing tempo conditions was examined. Where previous work with GFNNs focused on frequency and amplitude responses, the outputs were evaluated here on their WPO, considering that phase information is critical for pulse detection tasks. The experiment partially reproduced Velasco and Large's (2011) and Large, Herrera and Velasco's (2015) studies for comparison, and added two new rhythm categories for dynamic pulses.

In terms of phase prediction, it was found that GFNNs are able to capture metrical structure in isochronous and simple steady rhythms fairly well. However, when rhythms became more difficult or there was tempo change, interference and anti-phase entrainment became an issue.

Part II

Chapter 6

Adaptive Frequency Neural Networks

6.1 Introduction

Beat induction, the means by which humans listen to music and perceive a steady pulse, is achieved via a perceptual and cognitive process. Computationally modelling this phenomenon is an open problem, especially when processing expressive shaping of the music such as tempo change.

In Chapter 4, a hypothesis was held that the GFNN's entrainment properties, the ability for each oscillator to phase shift, would make them good candidates for solving the expressive timing problem. However, in Chapter 5 it was found that GFNN's respond poorly to tempo change rhythms due to problems of interference with other resonant frequencies. This makes GFNNs, in their current guise, difficult to use for beat tracking expressively timed rhythms.

In order to improve upon the output of the GFNN for expressively timed rhythms, the general level of interference must be drastically reduced. This could be done through extra dampening of the oscillators, so that the amplitude of resonant frequencies that are no longer relevant can die away faster. However, this can result in some instabilities within the network and some long-term memory within the network will be lost.

Another approach would be to reduce the number of oscillators in the network, thus clearing up any potential irrelevant oscillators. However, when doing this one must ensure that the oscillators can still resonant to a wide range of frequencies or else the network will not be able to provide any meaningful information. The oscillators must therefore be modified so that they are able to entrain to a greater range of frequencies.

This chapter presents a novel variation on the GFNN: the Adaptive Frequency Neural Network (AFNN) which achieves this entrainment basin increase. In an AFNN, an additional Hebbian learning rule is applied to the oscillator frequencies within the network. The frequencies adapt to the stimulus through an attraction to local areas of resonance. A secondary elasticity rule attracts the oscillator frequencies back to their original values. These two new interacting adaptive rules are both weak forces, and allow for a large reduction of the network's density. This minimises interference whilst also maintaining a frequency spread across the gradient.

The results of an experiment comparing GFNNs with AFNNs are presented within this chapter, partially reproducing the results from Velasco and Large's (2011) last major MIR application of a GFNN, and Large, Herrera and Velasco's (2015) more recent neuroscientific contribution. As in Chapter 5, a greater focus on phase accuracy is made in the evaluation than shown in the aforementioned works. The results show that AFNNs can produce a better response to stimuli with both steady and varying pulses compared with GFNNs.

6.1.1 Contributions

This chapter contributes a novel neural network model: the AFNN. The AFNN is evaluated with a comparative, quantitative study of previous GFNN models in the literature, in tandem with a new set of dynamic rhythms. An open source Python implementation is provided in the author's GFNN library[1].

[1] https://github.com/andyr0id/PyGFNN

Part of this chapter has been published in (Lambert, Weyde and Armstrong, 2016a), and presented at the Cognitive Music Informatics Research seminar (CogMIR) in 2016, for which it won best poster presentation.

6.2 The Interference Problem

Chapter 5 introduced the weighted phase output (WPO) of a GFNN. Figure 5.2 showed an example of WPO over time. Even though the amplitude response to the same stimulus shows a clear corresponding metrical hierarchy (see Figure 5.1), the phase response remained noisy. This is due to the high density of oscillators required in a GFNN.

Velasco and Large used 289 oscillators per layer in their experiment, a density of 48 oscillators per octave (opo). These high densities are often used in GFNNs to capture a wide range of frequencies, but can cause interference in the network. The term *interference* is used here to mean interacting signals amplifying or cancelling each other when summed.

Since each oscillator can only entrain to a narrow range of frequencies, the use of a lower density not only increases the likelihood of missing a relevant frequency, but it also stops local frequency populations from reinforcing one another. An example of this can be seen in Figure 6.1, where frequency information is not being captured as successfully in a low density GFNN (LD-GFNN), with a resolution of 4opo.

In Chapter 4, this issue was addressed by using only the real part of the oscillator as a single meanfield output. This retained a meaningful representation of the oscillation, but ultimately removed important information.

A selective filter could also be applied, by comparing each oscillator with the mean amplitude of the GFNN, and only retaining resonating oscillators. However, using a selective filter is not an ideal solution to the interference problem as it requires an additional, non real-time, processing step which cannot be easily incorporated into an online machine learning chain. In addition, new frequencies would not be selected until they begin

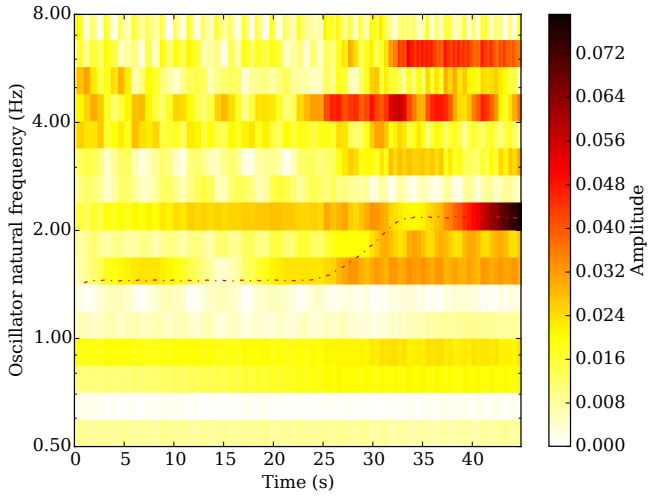


FIGURE 6.1: LD-GFNN (4opo) output. The dashed line shows stimulus frequency.

to resonate above the selection threshold, meaning that new resonances in changing tempos may be missed.

6.3 Adaptive Frequency Neural Networks

The AFNN model attempts to address both the interference within high density GFNNs, and improve the GFNNs ability to track changing frequencies, by introducing a Hebbian learning rule on the frequencies in the network. The rule is an adapted form of the general model introduced by Righetti, Buchli and Ijspeert (2006) shown in Eq. 6.1:

$$\frac{d\omega}{dt} = -\frac{\epsilon}{r}x(t)sin(\varphi) \tag{6.1}$$

Their method depends on an external driving stimulus ($x(t)$) and the state of the oscillator (r, amplitude; φ, phase), driving the frequency (ω) towards the frequency of the stimulus. The frequency adaptation happens on a slower time scale than the rest of the system and is influenced by the choice of ϵ, which can be thought of as a force scaling parameter. Since ϵ is divided by r higher amplitudes are affected less by the rule (Eq. 6.1).

This method differs from other adaptive models such as McAuley's

(1995) phase-resetting model by maintaining a biological plausibility ascribed to Hebbian learning (Kempter, Gerstner and Hemmen, 1999). It is a general method that has been proven to be valid for limit cycles of any form and in any dimension, including the Hopf oscillators which form the basis of GFNNs (see Righetti, Buchli and Ijspeert, 2006).

In an AFNN, Eq. 6.1 is modified to also include a linear elasticity, shown in Eq. 6.2.

$$\frac{d\omega}{dt} = -\frac{\epsilon_f}{r}x(t)sin(\varphi) - \frac{\epsilon_h}{r}(\frac{\omega - \omega_0}{\omega_0}) \tag{6.2}$$

The elastic force is an implementation of Hooke's Law, which describes a force that strengthens with displacement. The rule is introduced to ensure the AFNN retains a spread of frequencies (and thus metrical structure) across the gradient. The force is relative to natural frequency, and can be scaled through the ϵ_h parameter.

By balancing the adaptive (ϵ_f) and elastic (ϵ_h) parameters, the oscillator frequency is able to entrain to a greater range of frequencies, whilst also returning to its natural frequency (ω_0) when the stimulus is removed.

Figure 6.2 shows the frequencies adapting over time in the AFNN under sinusoidal input. Frequency is on the y-axis and time is on the x-axis, and each line represents on oscillator. The red dashed line shows the stimulus frequency, which in this example is 2.5Hz. The oscillator closest to the stimulus frequency quickly adapts its own frequency to the stimulus, resonances and stabilises for the duration of the simulation. Other high and low frequency oscillators find their own harmonic resonances and also remain stable.

There are two interesting harmonics where the dynamics of the two adaptive rules can be observed. Firstly an unstable resonance in the 2Hz range can be observed. The adaptive rule is attracting this oscillator to the stimulus, and the elastic rule is keeping it from straying too far from its natural frequency. This result in a frequency oscillation centring around 2Hz. The same pattern can be observed in a more extreme form in the oscillator around 3Hz. This oscillator is caught between the stimulus frequency and

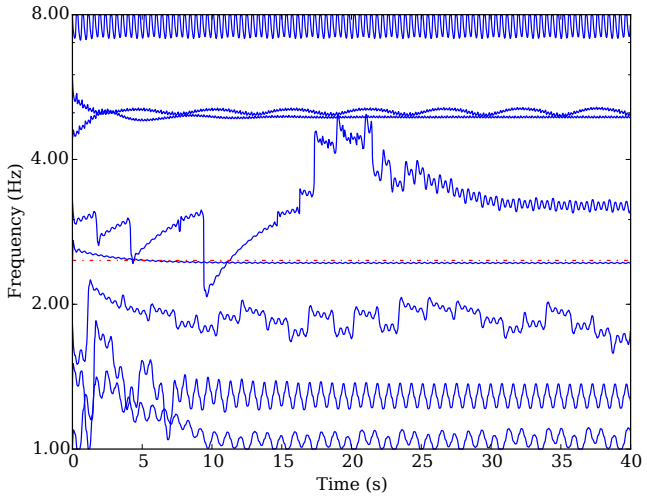


FIGURE 6.2: AFNN frequencies adapting to a sinusoidal stimulus. The dashed line shows stimulus frequency.

the 5Hz resonance shared by two other oscillators. The adaptive rule can be seen attractting the oscillator first to the stimulus, where it loses stability, then to the 5Hz harmonic, where it loses stability again, before eventually finding a stable resonance at an interesting harmonic ratio.

It can be seen that the AFNN preserves the architecture of the GFNN; the main difference is the frequency learning procedure. Figure 6.3 shows the WPO of an AFNN stimulated with the same stimulus as in Figure 5.2. One can observe that a reduced level of interference is apparent.

6.4 Experiment

6.4.1 Method

A pulse detection experiment was conducted to test the performance of the AFNN on both steady and tempo change rhythms. The experimental setup was the same as was presented in Chapter 5: 5 stimulus categories, and 20 tempos per category. In this chapter three different networks are compared: GFNNs, low density GFNNs and AFNNs.

The AFNN uses the same oscillator parameters and distribution as the GFNN, but the density is reduced to 4opo, 16 oscillators in total. ϵ_f and

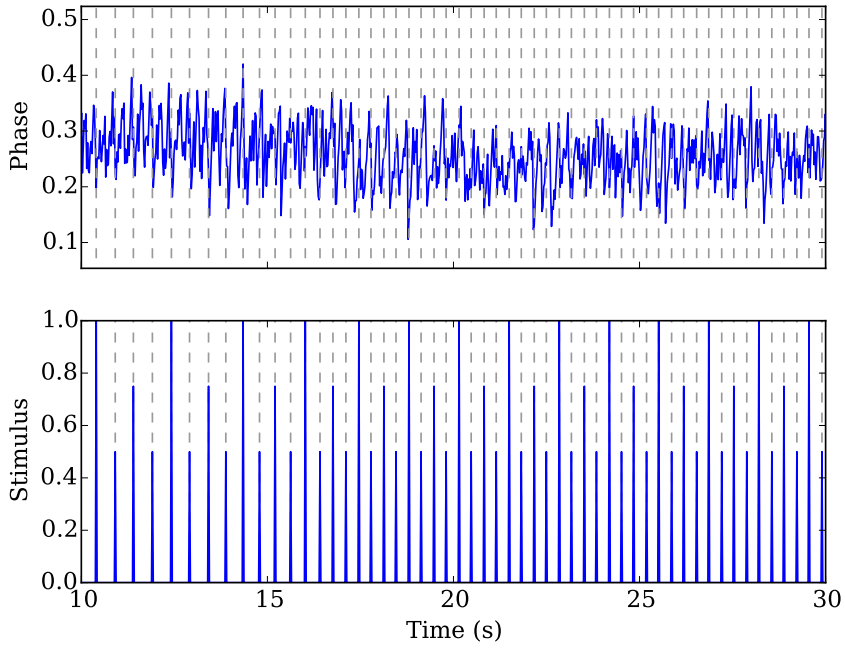


FIGURE 6.3: WPO of the AFNN over time. Reduced interference can be seen compared with Figure 5.2.

ϵ_h were hand-tuned to the values of 1.0 and 0.3 respectively. For comparison with the AFNN, a low density GFNN is also included, with the same density as the AFNN but no adaptive frequencies.

In implementation, the adaptive rule in integrated in simultaneously to the main oscillation deferential equation, within one fourth order Rung-Kutta solver.

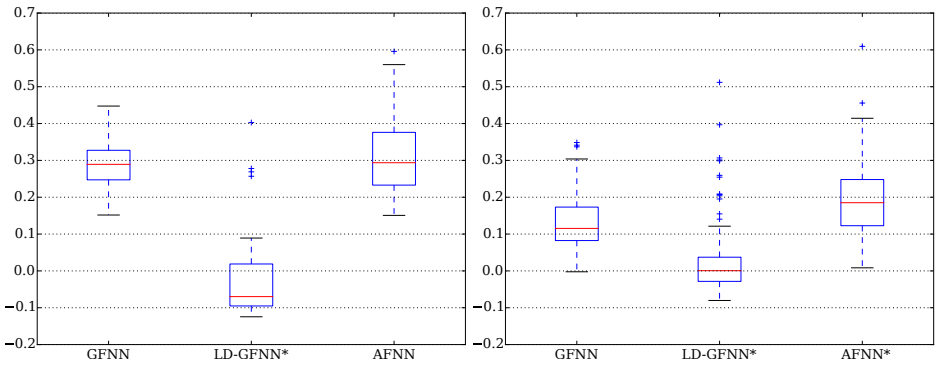
The evaluation method is the same as presented in Chapter 5.

6.4.2 Results

Figure 6.4 shows the results for the pulse detection experiment described above in the form of Box and Whisker plots. A significance test has been performed using a Wilcoxon signed rank test, due to the non-normal distribution of the results. An asterisk (*) denotes a statistical significance of $p < 0.05$, compared against the GFNN result.

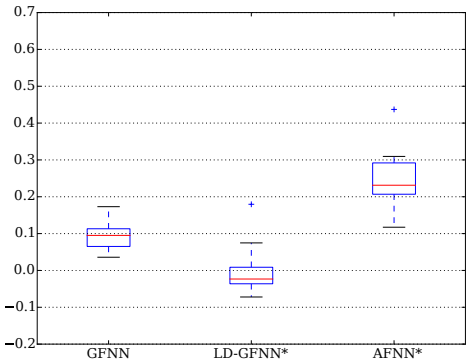
Figure 6.4a shows that the low density GFNN (B) performs significantly worse than the GFNN (A), showing little positive correlation and some negative correlation. This indicates the importance of having a dense GFNN.

The outliers seen can be explained by the randomised tempos: sometimes by chance the tempo falls into an entrainment basin of one or more oscillators. Despite its low density, the AFNN (C) fairs as well as the GFNN (A), showing a matching correlation to the target signal, especially in the upper quartile and maximum bounds. Exploring more values for ϵ_f and ϵ_h may yield even better results.

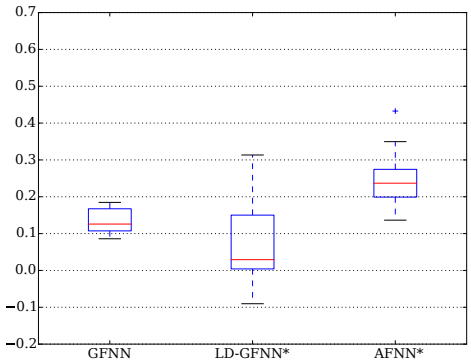


(A) Isochronous

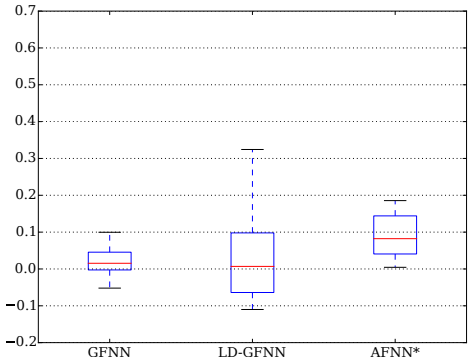
(B) Levels 1-4



(C) Accel.



(D) Rit.



(E) Son Clave

FIGURE 6.4: Box and Whisker plots of the PCC results. Boxes are as in Figure 5.4. *Denotes significance in a Wilcoxon signed rank test ($p < 0.05$), compared with (A).

In the Large, Herrera and Velasco (2015) rhythms (Figure 6.4b) a similar pattern is observed: the low density GFNN is more or less random in

its response, with the high number of outliers denoting high performance on that particular random tempo. The AFNN shows a significant improvement over the GFNN.

The Accelerando and Ritardando rhythm results (Figure 6.4c and 6.4d) show that the AFNN's response is a significant improvement over the GFNN, but still has low minimum values. This may be due to the fact that the adaptive rule depends on the amplitude of the oscillator, and therefore a frequency change may not be picked up straight away. Changing the oscillator model parameters to introduce more amplitude damping may help here. Nevertheless the AFNN model still performs significantly better than the GFNN, with a much lower oscillator density.

In the son clave results (Figure 6.4e) all networks perform poorly. A poorer result in comparison to the other rhythms was expected due to the difficulty of this rhythm. However, a significant improvement can be seen in the AFNN, which may be due to the reduced interference in the network.

6.5 Conclusions

In this chapter, a novel Adaptive Frequency Neural Network model (AFNN) was proposed. AFNNs extend GFNNs with a Hebbian learning rule on the oscillator frequencies, attracting them to local areas of resonance. Where previous work with GFNNs focused on frequency and amplitude responses, the outputs were evaluated on their weighted phase response, considering that phase information is critical for pulse detection tasks. An experiment was conducted, partially reproducing Velasco and Large's (2011) and Large, Herrera and Velasco's (2015) studies for comparison adding two new rhythm categories for dynamic pulses. When compared with GFNNs an improved response was shown by AFNNs to rhythmic stimuli with both steady and varying pulse frequencies.

AFNNs allow for a large reduction in the density of the network, which can improve the way the model can be used in tandem with other machine learning models, such as neural networks or classifiers. Furthermore the

system functions fully online for use in real time. In the future this possibility could be explored by implementing a complete beat-tracking system with an AFNN at its core.

There is a lot of exploration to do with regard to the GFNN/AFNN parameters, including the testing values for the adaptive frequency rule, oscillator models and internal connectivity. The outcome of this exploration may improve the results presented in this chapter.

The mode-locking to high order integer ratios, nonlinear response, and internal connectivity set GFNNs apart from many linear filtering methods such as the resonating comb filters and Kalman filters used in many signal prediction tasks. Coupled with frequency adaptation, the AFNN model provides very interesting prospects for applications in MIR and further afield.

Before the model can be used in a complete beat-tracking system an evaluation with more realistic MIR datasets must be performed.

Chapter 7

Perceiving Performed Expression with AFNNs

7.1 Introduction

In Chapter 6 a novel Adaptive Frequency Neural Network (AFNN) was detailed, which lowers the dimensionality of Large, Almonte and Velasco's (2010) Gradient Frequency Neural Network (GFNN) model, allowing the network to better track tempo changes. The network was quantitatively evaluated using a dataset of symbolic rhythms with simulated tempo changes. By doing so it was shown that AFNNs can match the performance of GFNNs on isochronous rhythms, whilst also significantly improving performance on dynamic tempos. However, since the evaluated rhythms were all idealised, synthesised, and symbolic, this did not give an accurate picture of how such networks will perform with real-world data.

This chapter further evaluates AFNNs with more realistic, human-performed datasets. Two audio datasets were selected for their expressive timing properties. Once again the AFNN was evaluated on its ability to perceive the pulse, and compared with the GFNN. An extensive grid search was performed on the AFNN to optimise the oscillator and adaptivity parameters.

Dealing with audio data creates additional complexities because it is much noisier than symbolic datasets. For this reason the mid-level representation approach presented in Chapter 4 is improved by utilising Böck and Widmer's (2013) SuperFlux onset detection function.

The results show that both systems perform poorly in comparison to the results presented in Chapter 6; however, the AFNN performed just as well as a GFNN, despite the lower dimensionality, and in some cases the AFNN outperformed the GFNN.

7.1.1 Contributions

This chapter contributes a detailed evaluation and analysis of GFNNs and AFNNs on expressive audio data.

7.2 Improving the Mid-level Representation

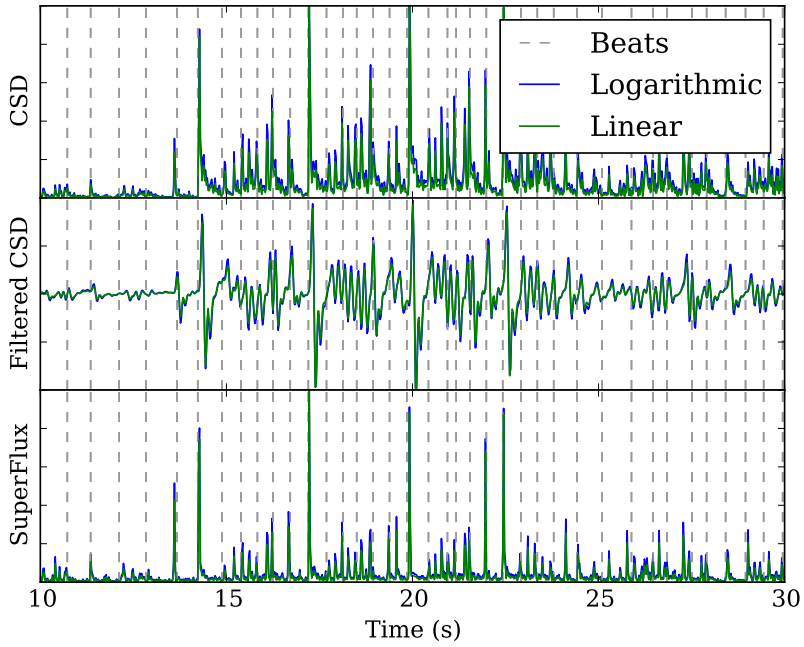
In Chapter 4 the complex spectral difference (CSD) mid-level representation was used to transform an audio signal into a more rhythmically meaningful representation. Figure 7.1a and 7.1b display example CSD outputs from the Mazurka (MAZ) and SMC datasets. The top plots show CSD, scaled both linearly and logarithmically.

From the above figures it can be seen that CSD can be noisy in the release phase of an onset, possibly due to phase modulation on instruments during a sustained note. It is also common to observe a low frequency modulation of the signal, which can produce a baseline wander.

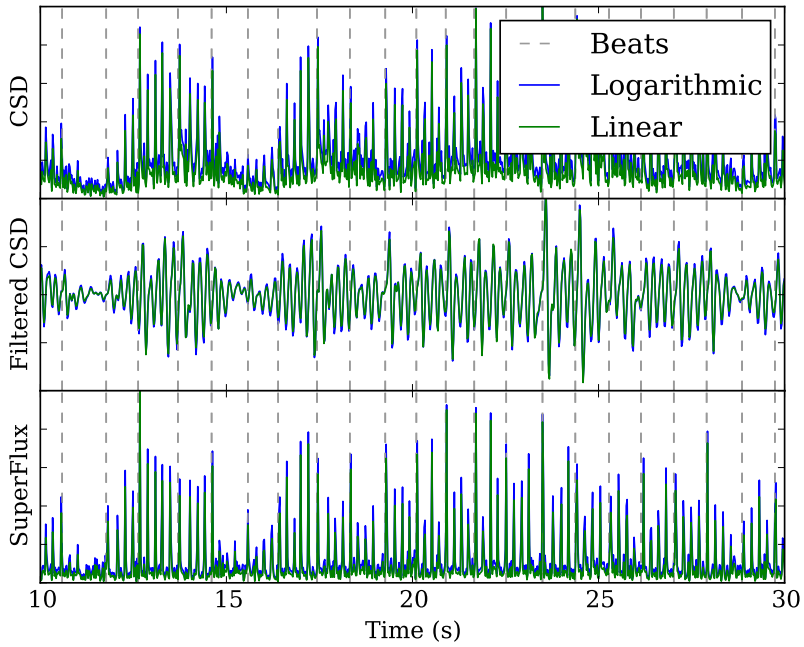
The GFNN responds best to peaky signals. Log-scaling the CSD signal can increase the general peakiness by squashing the high amplitude peaks and amplifying the low amplitude peaks. However, this also has an amplifying effect on the high and low frequency noise within the signal.

To reduce the high frequency noise in the release phase and the low frequency modulating noise, a band-pass filter can be applied. The middle plots show the result of a band-pass filter application: a more waveform-like signal rather than an the envelope-like CSD. Applying log-scaling can

also help to increase peakiness, but it is not as effective as on the CSD directly.



(A) A comparison of mid-level representations of an excerpt of Mazurka 06-1, performance ID 9048-01.



(B) A comparison of mid-level representations of an excerpt of SMC_003.

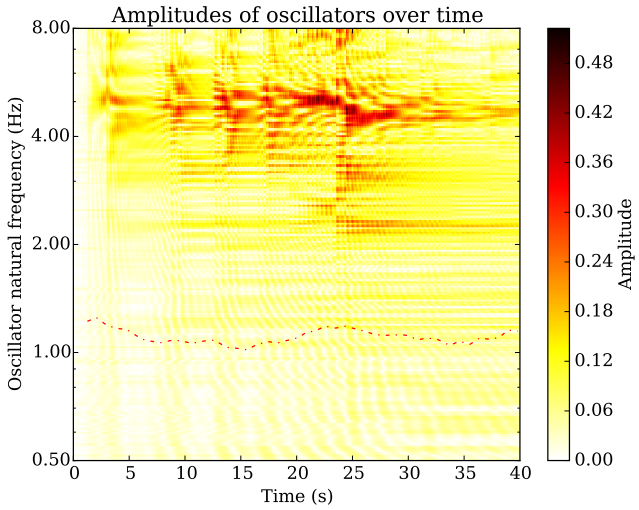
FIGURE 7.1: Mid-level representation comparisons from two different excerpt examples.

The bottom plots show the output of an alternative onset detection function, known as SuperFlux (Böck and Widmer, 2013). SuperFlux is able to reduce the noisiness of CSD by tracking spectral trajectories with a maximum filter. Thus the effects of vibrato are reduced and number of false positives can also be reduced. In the lower plot of Figure 7.1a the effect can clearly be seen just before the 15 second mark and again around 20 seconds, the release phase noise has been greatly reduced. Furthermore, SuperFlux also reduces the low-frequency modulation, which can clearly be seen in the lower Figure 7.1b. In general, the noise floor of the signal is reduced and there is very little low frequency content.

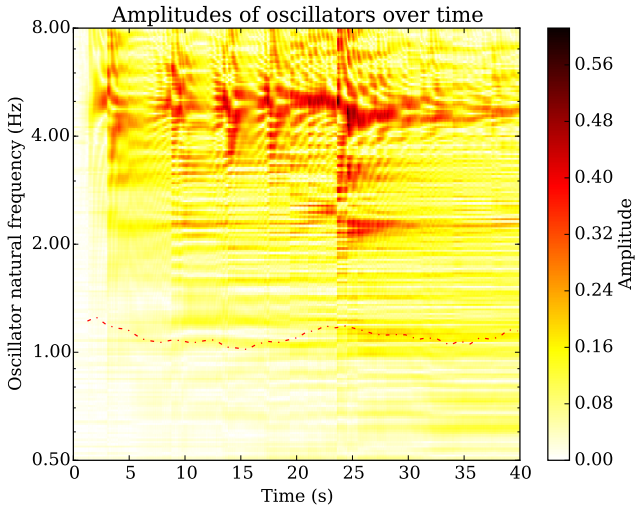
Figure 7.2 shows the output of a standard GFNN (critical oscillators, 48opo) to the different onset detectors. The red dashed line shows the pulse. In this example the pulse is two octave metrical levels below the strongest resonant frequency in the network, due to the nature of the rhythm in this excerpt. Figure 7.2a shows the CSD result, from which a clear resonance at this aforementioned pulse-harmonic frequency can be seen, as can the effect of the noisiness in the higher frequencies. An unstable pulse-harmonic is also resonating at just above 2Hz.

Figure 7.2b shows the bandpass filtered CSD result. In can be observed that resonances are in general much stronger in the same areas as the non-filtered CSD, but the sinusoidal-like signal causes a wider band of resonance in the network. This would make extracting the relevant frequency information more difficult.

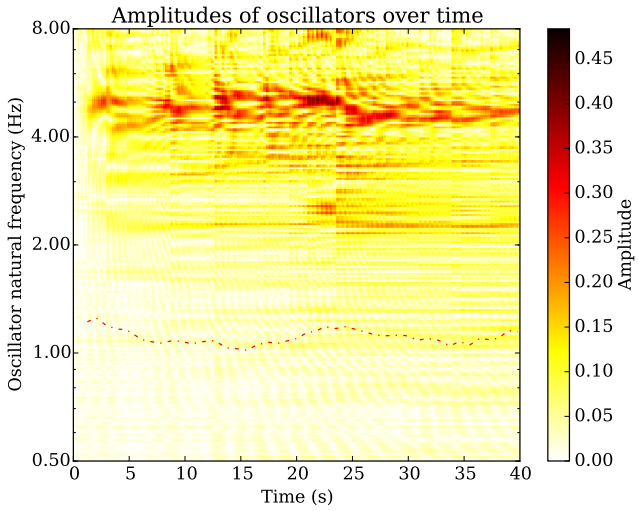
In Figure 7.2c, which is the output from the SuperFlux algorithm, a similar pattern can be seen to the others, but the resonance band is more focused on the higher harmonic and the 2Hz harmonic appears more stable throughout the excerpt. This provides evidence that the SuperFlux onset detection function is more suitable for use with GFNNs than CSD.



(A) GFNN response to unfiltered, log-scaled CSD.



(B) GFNN response to filtered, log-scaled CSD.



(C) GFNN response to unfiltered, log-scaled SuperFlux.

FIGURE 7.2: GFNN responses to CSD, using different onset detectors and filters. The red dashed line represents the pulse frequency.

7.3 Experiment

7.3.1 Method

A pulse detection experiment was conducted to evaluate the performance of AFNNs on discovering the pulse in realistic performed audio data.

Just as in Chapter 5 and 6, the AFNN's result is compared with a GFNN through the quantitative weighted phase output (WPO) correlation.

The AFNNs and GFNNs were evaluated on two datasets, the first of these is the same Mazurka dataset (MAZ) used in Chapter 4. A subset of 80 excerpts have been selected, 40 seconds each in duration, ensuring an even spread among mazurkas and performers. The second dataset used is known as the SMC dataset (SMC; Holzapfel et al., 2012), a set of excerpts from tracks identified as being difficult for the current state-of-the-art beat trackers, partly due to the degree of expressive timing in the excerpts.

SMC excerpts have been previously tagged with classifications as to why the annotation was difficult. In the experiments presented in this chapter, an SMC subset of 80 tracks have been selected, based on two tags that refer to dynamic tempos: 'expressive timing', and 'gradual tempo change'. For a full list of tags and the frequencies that they occur in the SMC dataset, see (Holzapfel et al., 2012).

The two datasets provide different things to look out for in the evaluation result. Both datasets are polyphonic and audio-based, but MAZ is mono-timbral (piano) and metrically homogeneous (3 beats in a bar) whereas SMC can be multi-timbral and contains music from many genres in many different metres. This means that the SMC excerpts are much more difficult to process compared with MAZ.

In the experiment a total of 228 different models were tested in a grid search. This included three different oscillator models: the damped oscillators used in Chapter 3, the detune oscillators used in Chapter 4, and the critical oscillators used in Chapter 4-6. The GFNNs were fixed at 48opo, but the AFNNs were tested at 12, 6, and 3opo. Five values of the AFNN's

Dataset	Oscillator	Network	WPO	Abs. WPO
MAZ	Damped	GFNN	**0.03239** (0.07448)	**0.05950** (0.05529)
		$AFNN_1$	0.02859 (0.07381)	0.05820 (0.05364)
	Critical	GFNN	-0.01543 (0.04597)	0.03749 (0.03075)
		$AFNN_2$	-0.01117 (0.04935)	0.03946 (0.03168)
	Detune	GFNN	-0.01832 (0.05001)	0.04144 (0.03346)
		$AFNN_3$	-0.01272 (0.04839)	0.03965 (0.03051)
SMC	Damped	GFNN	**0.02683** (0.06259)	0.04781 (0.04849)
		$AFNN_4$	0.02025 (0.06952)	**0.05446** (0.04772)
	Critical	GFNN	-0.03155 (0.06043)	0.05026 (0.04606)
		$AFNN_5$	-0.02596 (0.06183)	0.04846 (0.04636)
	Detune	GFNN	-0.03462 (0.05648)	0.05073 (0.0426)
		$AFNN_6$	-0.03120 (0.06039)	0.04981 (0.04625)

TABLE 7.1: Results of the grid search. The values show the mean results. The value in brackets denotes the standard deviation.

adaptive (ε_f) and elastic (ε_h) parameters were also tested: 1.0, 0.65, 0.3, 0.2, and 0.1. The continuous-valued output of the SuperFlux function was used as the input to all networks, at a sample rate of 86.15 Hz.

7.3.2 Results

Table 7.1 shows the results of the grid search, comprising the GFNN result and the best of all the tested AFNNs. The first two numbers are the WPO correlation and their standard deviation. For reasons that will be explained below, an absolute WPO correlation was also calculated . The additional AFNN parameters are listed in Table 7.2.

The results show that the AFNNs did not perform better than GFNNs in this experiment. However, as seen in Table 7.3, no significant differences were found between any of the models.

Both GFNNs and AFNNs show WPO correlations much lower than those observed in Chapter 5 and 6. This was expected due to the extra

Dataset	Network	Oscillator	ϵ_f	ϵ_h	opo
MAZ	$AFNN_1$	Damped	0.2	0.3	6
	$AFNN_2$	Critical	1.0	0.65	12
	$AFNN_3$	Detune	0.65	1.0	12
SMC	$AFNN_4$	Damped	0.65	0.2	3
	$AFNN_5$	Critical	1.0	0.2	12
	$AFNN_6$	Detune	0.2	0.1	12

TABLE 7.2: AFNN parameters for Table 7.1.

Dataset	Oscillator	p values	
		WPO	Abs. WPO
MAZ	Damped	0.35567	0.46352
	Critical	0.2218	0.36854
	Detune	0.18055	0.97271
SMC	Damped	0.33813	0.47251
	Critical	0.26517	0.65652
	Detune	0.38169	0.81833

TABLE 7.3: p-values returned from a Wilcoxon signed rank test between the GFNN and AFNN results. No significant differences were found in any of the models ($p >> 0.05$).

noise that audio data introduces into the input and the fact that the audio excerpts are expressively performed; however, it was not expected to affect the result this severely. In the future, both models' response to audio data must be improved, this is discussed in Chapter 8, Section 8.4 but is out of the scope of this thesis.

Despite the lower correlations several notable findings can still be observed. Critical and detune oscillators tend towards negative correlations, indicating that they are more likely to entrain to off-beats rather than on-beats. This may be due to the complex syncopated rhythms often found in both the MAZ and SMC excerpts, which may be exacerbated by tempo change. In order to fully understand this, a detailed qualitative study of the network's dynamics should be performed. The tend towards negative

correlations could be corrected for with a simple a phase offset on the network's output. By offsetting the phase value, an anti-phase relationship can be moved to a phase-locked relationship.

The absolute WPO correlation (abs. WPO) in Table 7.1 indicates what the mean correlation could be if all the negative phases were corrected with an offset. The results do show the expected improvement over the standard WPO, caused by negative values being reflected into a positive value and therefore adjusting the mean. These improvements can more than double the mean in cases where the network tended more towards the off-beat. This suggests that implementing an automatic phase offset, or stronger self-driven phase locking may prove a fruitful endeavour.

Damped oscillators perform the best across both datasets. As the name suggests, the damped oscillator mode greatly increases amplitude damping over time. Both the frequency adaptation rule and WPO rely on the oscillator's amplitude: greater amplitudes mean more stable frequencies, and greater contribution to WPO. It therefore stands to reason that in datasets where there is more tempo change, such as the ones investigated here, the network would require more damping. There is a trade-off, however, between long-term memory and forgetfulness in the network, especially when considering long-term structures. This is discussed more in Chapter 8, Section 8.3.

While GFNNs have outperformed AFNNs with damped oscillators in MAZ, AFNNs have achieved a comparable (not significantly different) score with 6opo compared with the 48opo of the GFNN. This represents an 8x efficiency increase. In SMC, AFNNs outperform GFNNs with 3opo, which is a 16x increase in efficiency compared with the GFNN. Computationally speaking then, an AFNNs still represents the better model choice.

When considering the AFNN's adaptive (ε_f) and elastic (ε_h) parameters in Table 7.2, it seems that MAZ requires a balance between ε_f and ε_h. SMC has a less clear relationship, but ε_f does tend to be stronger than ε_h. These differences in balancing adaptivity and elasticity may be due to the nature of the dataset's respective contents. MAZ's excerpts are all solo piano and

tend to be performed with gentle tempo curves, but the tempo is relatively stable. One can imagine the gentle pull and push of the adaptive forces describing this behaviour well. SMC on the other hand contains data from various genres and sometimes has more extreme tempo changes, step-wise jumps, and gaps where there is no pulse. This not only makes SMC the more difficult dataset to predict but also suggests that there is not one set of parameters that will fit all excerpts. Again, a qualitative analysis of the network's dynamics in this case would be beneficial.

7.4 Conclusions

In this chapter, an evaluation of AFNNs with more realistic, human generated data was conducted. Two audio datasets were selected for their expressive timing properties and the AFNN was evaluated on its ability to perceive the pulse.

An extensive grid search showed that both systems perform poorly in comparison to the results presented in Chapter 6; however, the AFNN performed just as well as a GFNN, with an up to 16x increase in efficiency and no significant differences between the models. In some cases the AFNN outperformed the GFNN. While this makes the case for an AFNN as a more efficient GFNN, the evidence for tracking dynamic pulse with AFNNs is not strong.

Closer inspection via a qualitative analysis of the output of the AFNN could reveal potential opportunities for improvement but this is beyond the scope of this thesis.

Chapter 8

Discussion

8.1 Introduction

In this chapter, the results of all the experiments are drawn together and related back to the problem task and the wider literature.

The aim of this research was to improve computational models for expressive rhythm perception and prediction in automated and interactive music systems. A cognitive machine learning approach was taken, utilising the existing GFNN and LSTM models in Chapters 3 - 5, before introducing and evaluating a proposed novel model: the AFNN in Chapters 6 and 7.

In several simulated experiments with the models, it was found that modelling metrical perception with GFNNs improved the rhythmic predictions in an RNN music model. Furthermore, it was discovered that adding a frequency adaptation rule to the GFNN (termed an AFNN) further improved the oscillator network's response to tempo change, both in resonance clarity through an improved signal correlation, and network efficiency through a reduced dimensionality.

The AFNN was evaluated over a series of experiments on sets of symbolic and audio rhythms, both from the literature and created specifically

for this research. When evaluating the time-based output of symbolic stimuli with both steady and varying pulse frequencies, AFNNs showed significantly improved responses and entrainment correlation in the pulse frequency. On two datasets of audio data, there was no significant difference in the performance of AFNNs against GFNNs. The AFNNs matched the performance of GFNNs, despite their lower oscillator density. 48opo GFNNs were reduced to 3opo AFNNs in the best case, 12opo AFNNs in the worst case.

8.2 Expectational and Probabilistic Prediction

Chapters 3 and 4 utilised a two layer neural network model for rhythmic and melodic modelling. The models were different in scope, with Chapter 3 focusing on steady-tempo symbolic melody prediction, and Chapter 4 modelling expressively timed rhythmic predictions. The topologies of the two models were similar in that they both incorporated a continuous-time oscillator network paired with an RNN, and they both modelled time in series. In essence, the model incorporated two different layers of prediction: *expectational* and *probabilistic*. An expectational prediction dictates when one *could* take action; a probabilistic prediction dictates when one *should* do so.

The GFNN captures expectational prediction. The summed activations of the oscillators in the network at any point in time show an expectancy of an event occurring at that time. This is achieved through the neural oscillation periods synchronising at certain ratios. Much like GTTM's dot notation, which denotes musical downbeat within a metrical structure, when several oscillators are aligned in their activation, that denotes a 'stronger' beat. This not only harks back to Huron's (2006) concept of musical anticipation, but also Large and Jones's (1999) theory of attentional dynamics: a focusing of attention at the time of the next expected event. Attentional dynamics was the precursor to Large and Kolen's (1994) nonlinear resonance theory, upon which the GFNN is based.

In certain ways, the AFNN can be seen as another layer upon the attentional dynamics theory. Rather than interpreting the network as a relatively small bank of oscillators with adapting frequencies, one can instead view the changing frequencies within the network as attentional energy across a wider spread, focusing in on relevant periodicities as well as time windows. Unlike Large and Jones the AFNNs adaptation is completely Hebbian-based and so also retains its biological plausibility.

The LSTM part of the networks used in Chapters 3 and 4 model a different kind of prediction: probabilistic. Reading expectational resonance information from the oscillator network, the LSTM performs a complex nonlinear calculation of its own, then transforms the output into a probabilistic rhythm activation. This is an important step to take when creating a predictive or generative model, as it forms a higher level of control over the low-level expectational output of the GFNN. The LSTM captures subtle information about how to read and interpret the GFNN's signal. Higher level musical features such as genre or performance style can be learned by the model, producing an output that is not only a series of beats, but a set of informed musical actions. Cognitively speaking this relates back to Conklin and Witten's (1995) and Cherla, Weyde and Garcez's (2014) multiple viewpoint prediction models.

What sets the GFNN-LSTM apart from the models used in the aforementioned works is the ability to move away from discrete-time models to continuous-time, which gives the ability to model expressively timed rhythms. An avenue of exploration in the future would be to explore how multiple-viewpoint models and continuous-time expectational models could be further incorporated into one another. This would allow for a continuous-time multiple viewpoint prediction rather than the discrete timing used in the current model.

8.3 Oscillator Network Comparisons

Large (1995) and Eck (2002) have previously theorised that a connected oscillator network alone could deal with rubato and tempo change within a beat tracking system, but the results from Chapter 7 showed no evidence to support that theory. When evaluated on their time-domain phase output and correlated against a beat-phase signal, the results showed little positive correlations.

However, this thesis did not take into account the two layer, multi oscillator models used in Velasco and Large (2011) and Large, Herrera and Velasco (2015). In these studies, a layer of *critical* oscillators was paired with a layer of 'limit-cycle' oscillators to represent the sensory and motor cortices respectively. This simplification step was introduced to focus on the primary research goals investigating a specific set of behaviours: namely expressive time perception and frequency adaptation. To do this, only one layer GFNNs and AFNNs were considered. Extending the study to multiple oscillator layers and models was deemed overly complex. This still retained a fair enquiry as comparisons made between the two models were always on a like-for-like basis. The comparative studies in this thesis confirm Velasco and Large's findings that GFNNs formed of critical oscillators do show beat induction properties, but the energy level of the pulse frequency is fairly low. Even so, the results of Chapter 4 show that even this output from the GFNN can be useful in an RNN prediction model.

It is reasonable to assume that the AFNN will indeed extend to multiple layers and retain the same benefits. The adaptive rule was taken and modified from Righetti, Buchli and Ijspeert (2006), where it has been proven to work with similar limit-cycle oscillators to those used in the Velasco and Large's motor cortex models.

Another simplification made in the experiments presented here is the eventual exclusion of internal connections, fixed or learned, from the oscillator network. In Chapter 4, three connectivity states were investigated: no connections, online connections, and online connections with an initial

state. It was found that the third option produced the best results. However, following a personal communication from Large it was revealed that, since the connection matrix is complex valued, phase offsets are introduced between oscillators. This has the side effect of inter-oscillator entrainment becoming more complex to model. One cannot rely on the phase output of the oscillator itself without taking into account the various phase offsets in the connection matrix. That, in turn, would affect the weighted phase output of the system. For these reasons, the network evaluations from Chapter 5 onwards did not activate the connection matrix.

One of the side effects of this decision was that lower frequencies in the network took longer to resonate. This can be seen in Figure 7.2, where the strongest resonances showed a two-octave error against the pulse. Oscillators with lower frequencies operate on longer time scales to the higher frequencies, meaning that resonances took longer to build and required more energy. Internal connections in the GFNN can help to mitigate this effect and so one must be take another look at the connection matrix. This could also mimic GTTM's preference rule, by being tuned to prefer resonance in certain bands. For example 2Hz, or 120bpm, is a typical tempo used in many kinds of music. It is the beginning of the *allegro* tempo range and is often the default setting of many electronic sequencers. Tuning the network to these well-defined bands (and their harmonics) would provide an immediate resonance boost.

If the connection matrix were to be reactivated, then the engineering case for an AFNN would be made clearer as the algorithm is the most computationally expensive part of the system at $\mathcal{O}(n^2)$, where n is the number of oscillators in the network. On the hardware used in this research, this meant that GFNNs with a connection matrix were not able to operate in realtime. However, AFNNs were able to reduce n back to faster-than-realtime speeds.

GFNNs have been criticised in the past for their plausibility as a cognitive model of beat induction. One of the major criticisms is the length of

time a clear resonance takes to build in the network, which is several seconds on average. Since humans can seem to induce a beat after two or three events, the canonical oscillator model within a GFNN can be called into question. Unfortunately, AFNNs may make matters worse in this regard, as the frequency adaptation rule operates on a long time scale to the oscillation itself (Righetti, Buchli and Ijspeert, 2006), and therefore may take a while to discover stable resonances. However, whilst resonance energy does build slowly over time within a GFNN, oscillator entrainment does happen relatively quickly, usually within four events. Perhaps another Hebbian rule, similar to the way the connection matrix learns, can observe the entraining phase shifts in the network in such a way to boost resonance gain. One avenue for exploration here could be the Kuramoto model (Kuramoto, 1984), which utilises phase change as a coupling mechanism.

This may also prove to be beneficial for cases of expressive timing. In expressively timed rhythms the sense of pulse and metre is constantly shifting. This can cause oscillators in the network to lose energy, or in the worst case become completely desynchronised. A learning rule that observes phase change could help oscillators when entrainment shifts occur, by giving them a boost of additional amplitude.

8.4 Improving the AFNN's Adaptivity to Audio

The results presented in Chapter 6 showed significantly improved WPO correlations with an AFNN against a GFNN, but when the networks were evaluated on audio data in Chapter 7, there was no significant difference observed. Why is it that the audio data did not show the same improvement?

One reason may be due to the frequency adaptation rule itself. In Chapter 7, steps were taken to improve upon the onset detection function used in Chapter 4. The SuperFlux algorithm was used to reduce the noise floor and low frequencies from the signal, whilst also increasing the peakiness of the detected onsets. This produced a clearer, cleaner response in the GFNN

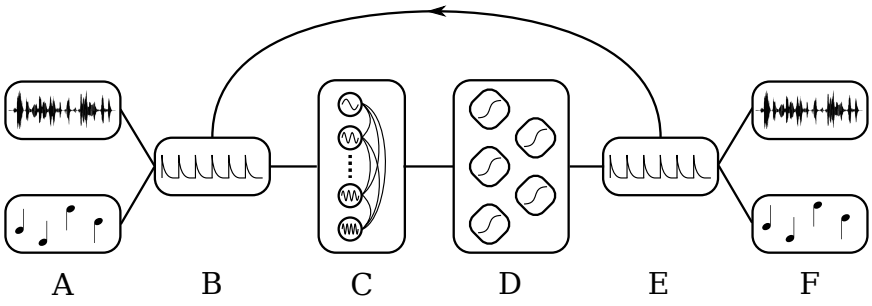


FIGURE 8.1: An overview of the proposed model showing (A) audio or symbolic input, (B) time-series rhythm representation, (C) AFNN, (D) LSTM, (E) time-series rhythm prediction, and (F) audio or symbolic output. An internal feedback loop connects E and B.

(see Figure 7.1 and 7.2). However, the ODF still contained a lot of low level noise. The AFNN's frequency adaptation rule is very sensitive to noise in the input signal. Clean spiky signals such as those used in Chapter 6 or pure sinusoidal signals work well. Consider the adaptive side of the rule:

$$-\frac{\epsilon_f}{r}x(t)sin(\varphi) \tag{8.1}$$

this takes the input signal $x(t)$ and directly multiplies it against the sin of the phase φ. This means that any noise in the signal is amplified as the phase rotates, and is at its peak when $\varphi = \pi/2$ or $\varphi = 3\pi/2$. Since the input signal is continuous, this could be mitigated with an adaptive threshold on the signal:

$$-\frac{\epsilon_f}{r}\sigma(x(t))sin(\varphi) \tag{8.2}$$

where σ is an adaptive threshold activation function, such as a cubic spline interpolation (Scardapane et al., 2016).

Passing an activation in this way could negate all low-level noise, and provide a cleaner signal to the adaptive rule, whilst maintaining a continuous signal to the oscillators themselves.

8.5 AFNNs for Continuous Time Rhythm Generation

In this thesis, two generative models have already been presented. Chapter 3 presented a GFNN-LSTM model to predict both pitch and rhythm in

metrically-quantised time-series data of folk melodies; and in Chapter 4 this system was extended to expressive rhythm predictions. It would be trivial to close the loop in this system, creating a feedback between input and output. This would allow indefinite, self-driven generation of new rhythmic structures which could be evaluated for their novelty.

It remains a rarity for generative music systems to produce expressive variations in their output. A more common approach is for a generative system to output an abstract symbolic rhythm to be 'played' by a computer system for expressive music performance (CSEMP). Systems such as *Omax* (Assayag et al., 2006) and *ImproteK* (Nika et al., 2014) can be said to be holistic generative improvisation systems, designed to be played with a human musician. *Omax*'s design is to ignore the pulse entirely by restructuring the audio input. *ImproteK* uses a beat-tracker to detect tempo, which is then fixed for the remainder of the improvisation.

Sometimes the application of expressive articulation is left to human performers. For example, in Eigenfeldt's *An Unnatural Selection* (2015), musical phrases were generated by a genetic algorithm in score form, which were then sight read by eight human musicians. The musicians played these generated phrases, side-stepping the need for expressive articulation to be generated by the system itself.

The systems presented in this thesis use audio or symbolic data as input, and the output is a new rhythm prediction signal. The rhythm output can be easily used to produce a new audio or symbolic signal and exciting the network with untrained data will produce novel outputs. This application of a GFNN-LSTM is an expressive rhythm generative system.

Rather than separate rhythm generation into two distinct event creation and expressive playback phases, the GFNN-LSTM represents a holistic approach based on cognitive models of metre perception. The system outputs in continuous time, meaning there is no prior or external knowledge of tempo or metre beyond a single time-series input.

Figure 8.1 shows an overview of the proposed system. There is a singular input (A), which can be symbolic or audio data. This is converted into

a time-series data signal (B), retaining only rhythmic onsets. A relatively high sample rate should be chosen to minimise any metric quantisation and retain timing variance. The sample rate of 86Hz used in this thesis would make a good choice and is supported by previous findings (Davies and Plumbley, 2007).

In (C) the GFNNs from Chapter 3 and 4 are replaced with the AFNN. AFNNs address the interference within GFNNs, and improves the network's ability to track changing frequencies.

Before the AFNN stage, the model could still be described as a discrete time model. However, integrating the AFNN's system of differential equations through a time-step forms a continuous time model, from which values can be sampled at discrete time points. The resonances formed in (C) are then used as inputs to an LSTM (D). The LSTM's prediction (E) is used to render a new audio or symbolic rhythm (F) and can be combined with a pitch output to generate a complete melody. A feedback loop connects (E) to (B) so the system can operate autonomously or as part of an ensemble. Alternatively, a feedback loop can be made linking (F) to (A), which may be more straightforward in an ensemble setting.

In terms of Kirke and Miranda's (2009) CSEMP framework, the LSTM fulfils several roles. It is the central kernel where performance knowledge, context, and to a certain extent the instrument model can be captured, all leaning from a corpus of performance examples. The AFNN mainly fulfils the role of the music analysis and adaptation processes, but also contains elements of performance knowledge and context.

One variant approach would be to follow *SONOR* (Gasser, Eck and Port, 1999) by using the AFNN for both metrical structure learning and generation (see Chapter 2 Section 2.3.3). The input or learning phase could be achieved by enabling the internal connectivity leaning. After input is learned or stabilised, one could switch the oscillators to limit cycle mode and fix the internal connections to enable a self-oscillating output. One might also use the frequency adaptation of an AFNN to be more adaptive than SONOR's sinusoids. The AFNN output could be taken as the system

output, as is the case with SONOR, but this would only provide the expectational prediction layer. For a full probabilistic prediction, the LSTM or similar model must be incorporated.

8.5.1 Generative Evaluation

This thesis presents evidence to suggest that the model outlined above is viable as an online interactive generative rhythm system. However, in future work the generative outputs of such a system must still be evaluated and validated.

When considering generative software, validating the work both in terms of the computational system and the output it creates is still a challenge for the community at large (Jordanous, 2011).

Adopting Jordanous' (2012) Standardised Procedure for Evaluating Creative Systems (SPECS) methodology the following statements about the above system as a generative system can be made:

1. The system is aiming to satisfy a definition of creativity as producing novel rhythmic patterns, expressively timed to be in line with human performers' renditions of that same style of music.

2. The standards used to define said creativity are the annotated onset times of a selected dataset.

3. The system has been tested against these standards through quantitative and statistical metrics such as F-measure (see Chapter 4), which considers the generated rhythms precision and recall. Further qualitative evaluation can take the form of an online listening test.

In a similar vein to a previous expressive rhythm prediction experiment (see Chapter 4), the LSTM layer could be trained to predict rhythm onsets based off the AFNN's input. Once trained, the system would then be capable of generating new rhythms rendered in a similar expressive feel to the training corpus. This would exhibit all of the features up to level 5 in Eigenfeldt et al. (2013) MUME taxonomy:

1. *Independence*: the pitch content and rhythmic output's timing would be beyond the control of the composer.

2. *Compositionality*: the system would determine relationships between inputs and rhythm/pitch outputs.

3. *Generativity*: the system would create new musical gestures.

4. *Proactivity*: the system would decide when to initiate a new gesture, reacting to the input.

5. *Adaptability*: the system's behaviour would change over time via the AFNNs internal frequency adaptation, or via the external feedback loop, which incorporates other agents' input.

The system would not exhibit versatility, as the gestural style would be determined by the training data, and so it would not be able to produce rhythms in any other style. Volition would also not be exhibited, as the system would be driven by an external force.

8.6 Other Potential Applications for AFNNs

This thesis has been driven by an investigation into how to deal with varying tempo and expressive timing in automated and interactive music systems. The AFNN model was developed in order to improve upon Large, Almonte and Velasco's (2010) GFNN for expressive timing cases, and also to improve the integration of an oscillator network into a connectionist machine learning music model. However, other applications besides rhythm modelling and generation do exist in which the AFNN could prove useful.

Another use for AFNNs would be as an analytical tool. Ethnomusicologists are increasingly becoming aware of the importance of entrainment processes as an approach to understanding music making and music perception as a culturally interactive process (Clayton, Sager and Will, 2005). The entrainment and frequency-adding properties of nonlinear resonance can combine with the frequency adaptation in the AFNN to produce a new

kind of spectrogram-like representation for continuous-time rhythmic decomposition. A computational musicologist could look at the resonance patterns, and how they change over time, as a metrical analysis similar to IMA (Nestke and Noll, 2001; Volk, 2008). Uniquely, the AFNN has the ability to look at the frequency change derivatives, enabling the study expressive tempo change properties such as rubato and groove. Such an analysis can provide information on the dynamics of tempo and timing for which there is no established way of extraction from decomposition techniques such as Fourier analysis.

In terms of MUME categories, the AFNN is certainly classified as a cognitive approach. It is a biologically plausible model of the way a human neurological process may work. However, this thesis is not a cognitive investigation in itself, but an application of the model for practical engineering solutions. To use the model to develop and interrogate new cognitive theories of behaviour and understanding is left to the cognitive scientists and computational neuroscientists, such as those currently working with Large. This would require a wholly different methodology than that adopted in this thesis.

The field of MIR could see several uses for the AFNN. This thesis, and indeed much of the literature on GFNNs in general, has focused on AFNNs for rhythmic perception, but it is possible to extend the model to frequencies above 16Hz, in the audio range. There have been attempts in the past, most notably in Large (2010), but nothing that has yet had an impact on the field of machine listening in MIR. Example applications for the use of GFNNs/AFNNs include: chord recognition, timbre recognition, source separation, f0-estimation, or simply just another audio analysis technique.

In this thesis, a case has been made for the AFNN as a rhythm generator. Taking the generated frequencies into the audio domain may produce an interesting harmonic oscillator synthesiser. The author, as a researcher and musician, would be interested to hear what the changing audio-harmonic resonances would sound like when sonified.

8.7 On Deep Learning

Throughout this thesis the term *deep learning* has been avoided, however, the recent advances in connectionist machine learning attributed to this notion cannot be ignored. Deep learning models were first explored by Hinton, Osindero and Teh (2006) and later the term was coined by Bengio (2009). Deep learning referrs to any machine learning system with many layers of nonlinear transformations.

A deep learning neural network, commonly referred to as a deep network, is an example of a deep learning system, such as an LSTM with three or more hidden layers of 200 nodes each. Such deep networks have made improvements in the state-of-the-art of several pattern recognition tasks, such as image classification (Krizhevsky, Sutskever and Hinton, 2012), speech recognition (Dahl et al., 2012), and face recognition (Taigman et al., 2014). Each transformation layer in the system is purported to represent different abstractions of the input. For example, an image classifier may take pixel values as its input, extract edges in the first layer, collections of edges in the second, objects in the third, to form a classification in the output layer. Deep learning systems can take raw data and learn their own feature extractions for the task at hand. 'Shallower' networks must use pre-extracted features, often reduced in dimensionality, which requires extra human design steps. They have even been applied to music generation tasks (Sturm et al., 2016), including within Google's Magenta project[1].

One criticism of deep networks is their tendency to be easily fooled into mislabelling images with high confidence as something they are clearly not (Nguyen, Yosinski and Clune, 2015). This is a common problem and may point to a wider issue with machines learning their own features. Researchers are yet to understand exactly what the extracted features really relate to, or even if the learned features are anything more than quirks within the learned dataset. Unexpected inputs such as those used in Nguyen, Yosinski and Clune (2015) can output unexpected and invalid results.

[1] `https://magenta.tensorflow.org/`

The neural networks in this thesis cannot be classified as deep networks. The GFNNs used here and in the literature are not 'deep' enough to qualify, and they cannot take raw data as input (i.e. PCM samples). In the GFNN literature there has not been an instance of a network with more than two layers and the highest reported density has been 128opo (768 total oscillators; Angelis et al., 2013). LSTMs are being used widely for deep networks, but the LSTMs presented in this thesis are again far too small to be called 'deep', as they consist of only one hidden layer with 10 nodes.

Another difference between this work and the deep learning literature is that the oscillator networks cannot be trained with supervised methods such as backpropagation. GFNNs and AFNNs do perform a feature extraction, and this extraction is learned in terms of connectivity, but this is not trained and then fixed like a standard RNN, it is always an online process. Indeed, it was observed in this research that when GFNNs do have fixed connections this can create unexpected behaviour, including noisy resonance cascades down the frequency gradient (see Chapter 4 Section 4.2.3). The features extracted can be likened more closely to a comb filter – a digital signal processing technique that emphasises bands of frequencies – although GFNNs are more closely linked to biological systems.

Deep networks are computationally very expensive to train, requiring large matrix calculations. This has led to the release of library abstractions such as Theano[2] and TensorFlow[3] to perform the calculation on GPUs, which are optimised for such tasks and therefore can speed up the routine by a significant amount. The dynamical, continuous time nature of the oscillator models make it difficult to implement in the graph-like paradigm required by these libraries, but doing so would be worthwhile.

Despite the apparent shallowness of the networks presented in this thesis, the results are often comparable to similar state-of-the-art systems. This shows that performance improvements can be made without creating deeper networks, but instead through making better models.

[2] `http://deeplearning.net/software/theano/`
[3] `https://tensorflow.org/`

Chapter 9

Conclusions

9.1 Thesis Summary

A performing musical agent encapsulates a dynamic feedback loop of pulse and metre perception, expectational event prediction, and rhythmic production. When this occurs under changing tempo conditions the perceived metrical structure is perturbed, the listener's perception of musical time is affected, and their internal sense of pulse and metre is in a state of flux. In this thesis this process was referred to as *metrical flux*, and the research undertaken was concerned with modelling this phenomenon for the purposes of improving automatic and interactive music systems.

A cognitive machine learning approach was taken, utilising the existing Gradient Frequency Neural Network (GFNN; Large, Almonte and Velasco, 2010) and Long Short-Term Memory recurrent neural network (LSTM; Hochreiter and Schmidhuber, 1997) to first identify if nonlinear resonance patterns could be useful for melody modelling tasks (see Chapter 3), before attempting to predict expressive rhythm with these systems (see Chapter 4), and evaluating the GFNN's response to changing tempo (see Chapter 5).

A novel model was introduced, termed the AFNN (see Chapter 6) which introduced a new learning rule to the frequencies of the oscillators in a GFNN. This allowed for a great dimensionality reduction in the network

and was evaluated against the GFNN on pulse predicting tasks involving symbolic and real-world audio recordings (see Chapter 7).

9.2 Outcomes

At the beginning of this thesis several research questions were posed which can now be addressed.

1. Can a GFNN improve machine learning music models of melody?

To answer this question an experiment was performed comparing the pitch and rhythm prediction performance of a standard LSTM model against an LSTM with an added GFNN layer (GFNN-LSTM). This is reported in Chapter 3. It was found that providing the metrical resonance data from the GFNN did indeed help to improve melody prediction with an LSTM. Since the symbolic melodies were metrically homogeneous and all set at the same tempo, additional performance increases could be made by filtering the oscillators in the GFNN down to only the most resonant. Such a step would not be possible with changing tempo or differing genres or performance styles. Despite the positive result here, it is not possible to say in the general case whether GFNNs will improve any machine learning model. However, since LSTMs perform well in time-series modelling tasks including music, and are therefore widely used, this is still a useful result to report.

2. Can GFNNs form a machine learning music model of expressive rhythm production?

In Chapter 4 an experiment was performed to investigate this question. Following the same GFNN-LSTM network topology from Chapter 3, the network was trained to predict rhythm onsets, but this time on an audio dataset of expressively timed piano music. The use of different oscillator models and internal connectivity states was investigated, and a new meanfield filtering method was introduced. The best performing system

achieved a mean prediction F-measure of 77.2% across the validation set. According to the literature review conducted during this research, no similar system was found that had ever been quantitatively evaluated and reported before. Therefore it was not possible to directly compare the result to existing systems. However, a comparison was drawn to the state-of-the-art LSTM beat tracking systems using the same dataset and found the results similar. As with the previous question, a general claim about the use of GFNNs with any machine learning model cannot be made.

3. How well does a GFNN capture tempo change?

In order to answer this question the GFNNs were evaluated on a synthesised dataset of symbolic rhythms with changing tempos and the results compared to other rhythms found in the literature. A new time-based evaluation method was introduced, designed specifically to examine the phase correlation of the GFNN to the pulse. In Chapter 5 it was reported that GFNNs captured metrical structure in isochronous and simple steady rhythms fairly well. However, when rhythms became more difficult or where there was tempo change the correlations against the pulse became very low.

4. Can a similar neural resonance based cognitive model improve the GFNN's response to tempo change and expressive timing?

To investigate this question a novel model was proposed, the AFNN, that uses a Hebbian learning rule to directly alter frequencies of oscillators in a GFNN. The rule combined notions of attraction to resonance areas and elasticity back to the natural frequency to ensure a spread of frequencies across the network was still maintained. In Chapter 6 the new model was evaluated under the same conditions as the previous experiment. It was found that AFNNs had a significantly improved response to tempo changes, and matched the performance of GFNNs on steady tempos.

5. How would such a model compare with previous perceptual models and on real-world audio datasets?

Two audio datasets were chosen to investigate this question. The first was the same piano dataset used in Chapter 4, and the second was a beat-tracking dataset specifically chosen for its high amount of expressively timed excerpts. The audio data was transformed into a continuous onset detection function and run through both GFNNs and AFNNs before being evaluated for their pulse correlations. Despite the promising results shown in AFNNs reported in Chapter 6, there was no significant difference in the evaluation of metric scores between the GFNN and the AFNN shown in Chapter 7. Therefore it was found that AFNNs can provide a more computationally efficient solution compared with GFNNs. A discussion of why the performance was poor on this audio data was conducted in Chapter 8, Section 8.4, and a potential solution was put forward.

9.3 Limitations

The work presented in this thesis has several limitations.

Chapters 3 and 4 presented experimental neural network architectures for perceiving and predicting rhythm, melody and expressive timing. The results presented are based on very limited datasets of symbolic and audio data consisting of 100 and 80 training examples respectively. These numbers were chosen to provide a dataset that is as coherent as possible, but that also provides enough data to perform reliable cross-validation. For instance, a subset of German folk songs were used in Chapter 3, as one cannot assume that LSTMs are able to differentiate between culturally-specific rhythms and melodies. Furthermore, the topologies were fixed and comparable to previous work, but no direct system comparison was made between the GFNN-LSTMs presented here and previous attempts in the literature. This was due to a lack of available information in the literature, rendering it near impossible to reproduce an exact system. Since performing these experiments, more researchers have begun to share more details about their experiments. Most notably with Sturm et al.'s (2016) work. It

is hoped that enough information has been provided in this thesis to reproduce all experiments exactly.

It important to note also that all rhythm and melody modelling was reduced to a monophonic line for pitch and rhythm prediction, regardless of the input data polyphony. This was done to simplify the network architecture and evaluation routine. In Chapter 4, some qualitative examples are also provided, however this still represents a significant limitation in the applicability of the results.

In Chapter 7, it was found that GFNNs and AFNNs were extremely sensitive (and not very resilient) to noisy input from an onset detection function (ODF). Steps were taken to improve the GFNN/AFNN output by choosing a different ODF to Chapter 4, yet a comparison of ODFs was not provided and would have been useful for future researchers.

9.4 Future Work

Apart from addressing the limitations above, several recommended avenues of research are left open for further exploration. Firstly, more work should be undertaken to improve the AFNN's response to audio data. The results presented in Chapter 7 were not as expected. While they make the case for an AFNN as a more efficient GFNN, the evidence for tracking changing tempo with AFNNs is not strong. Since that is that case, a qualitative analysis of the output of the AFNN should be done in more detail. In a similar manner to Grosche, Müller and Sapp's (2010) study on beat-tracking errors, one may look closely at examples from MAZ and SMC where the AFNN performs particularly well or poorly. Comparisons between oscillator models and GFNN models can be made to gain insight into the behaviour of the networks. A qualitative study such as this can improve the frequency adaptation rule proposed in this thesis. One improvement has been suggested here for investigation (see Chapter 8, Section 8.4).

Work should also be done to extend the AFNN to multi-layer oscillator networks with both inter-oscillator and inter-layer connections. Velasco

and Large (2011) provide evidence that doing so could improve the pulse responses reported in this thesis. However, this would not be a trivial undertaking; as discussed in Chapter 8, Section 8.3 the connection matrix introduced phase offsets into the oscillators which need to be accounted for.

Once these improvements have been made, the next obvious step for creating an expressively timed interactive music system would be to repeat experiment from Chapter 4 with the improved AFNN model. The replacement of the GFNN with the AFNN could make the metrical signal clearer, and therefore simplify the modelling required by the LSTM layer. Also rather than using a real meanfield output, as was the case in Chapter 4, one could adopt the weighted phase output (WPO) proposed in Chapter 5. Perhaps the dimensionality of the AFNN would be reduced enough not to require WPO or meanfield reductions at all.

This generative system may be incorporated into a multiple viewpoint prediction model (Conklin and Witten, 1995), with the added improvement of being continuously timed, rather than discretely timed.

After this has been built and quantitatively evaluated, the system can be qualitatively evaluated as a generative system, following the steps discussed in Chapter 8, Section 8.5.

Other applications of the AFNN should be explored too. There is still a clear need for a beat-tracking system that deals well with expressively timed rhythm. Incorporating an AFNN into a beat tracker could be a worthwhile endeavour. Indeed there are several potential MIR applications, discussed in Chapter 8, Section 8.6.

For any researchers wishing to take these tasks on, an open source Python implementation of the GFNN and AFNN models is provided on the author's GitHub repository[1].

[1] `https://github.com/andyr0id/PyGFNN`

9.5 Personal Experience

My PhD research began with two fundamental questions: *Can musical behaviour arise out of synchronised oscillator networks? Can such a system be used to explore the biological root of musical creativity?* As you can tell it has been quite a journey to get from these conjectural questions to the (hopefully) more rigorous Computer Science (CS) thesis you have just read. However, it has not been a complete shift, you can still see the traces of my original interests scattered throughout the work.

My fascination with nonlinear dynamics, oscillation, and synchronisation began while reading my Masters in *Creative Systems* at the University of Sussex. My first attempt at using neural networks to generate rhythm was with a Continuous-time Recurrent Neural Network (CTRNN; Funahashi and Nakamura, 1993) drum machine. The CTRNN was artificially evolved with a fitness function that rewarded drum patterns fitting certain distributions. Hilariously, the evolved behaviour was simply the drums being hit one after the other, steadily increasing in tempo. Was this evolved expressive timing? Certainly not.

It was during an Adaptive Systems lecture that I was first introduced to the Kuramoto model of oscillator synchronisation (Kuramoto, 1984), and this inspired me to create *Crickets*: a self-synchronising oscillator network for rhythm generation (Lambert, 2012). Unlike my CTRNN drummer, Crickets had much more modest aims and allowed a user to interact with the network to create some rudimentary rhythms.

At the start of the PhD I was coming from a cybernetics and dynamical systems perspective. I was convinced that a continuous-time rhythm generation system, facilitated by synchronised nonlinear oscillators like in *Crickets*, could provide some insight into the biology of music.

I was already aware of Eck's (2001) oscillator networks, but the real early breakthrough came when I read Large's *Neurodynamics of Music* (2010), and first discovered the GFNN. Here was a network of oscillators, canonicalised

from neurological models, that entrained and resonated in harmonic patterns to rhythm and pitch. It was almost exactly the system I had envisioned.

Shortly after, I identified a knowledge gap in modelling and generating expressively timed rhythms. I hypothesised that the continuous-time nature of the oscillators, along with their entrainment properties, may allow expressive timing to be directly captured in the model as phase shifting resonance, much like the Kuramoto model. For me this was the perfect place to make my CS contributions, and I became more interested in modelling musical timing as a continuous flux. The AFNN was a natural extension to this idea, using a Kuramoto-like frequency adaptation rule I was able to extend the entrainment basin of the canonical oscillator, and directly represent changing tempo in the model.

My personal aim was to bring this around full-circle, back to rhythm generation, within the time frame of my PhD. Unfortunately I was not able to achieve this. The additional work in setting up and evaluating an experiment of this nature with full scientific rigour would have required at least another year to complete. Removing the word 'generation' from my thesis title to focus on the perception and prediction aspects of the model was very difficult for me, but ultimately it enabled me to write a stronger, more coherent thesis. In my final year I was happy to receive a positive peer-review and acceptance at the MUME workshop outlining the plan for what the generative system may become (Lambert, Weyde and Armstrong, 2016e).

Looking back, I am more than happy with what I have achieved during these last three and a half years. I have had some amazing experiences, met some fantastic and inspiring people, and had the opportunity to work on something that I truly love.

9.6 Final Thoughts

The opening notion of 'generating time' (Roads, 2014) clearly refers to more stochastic methods of rhythmic generation, relating to similar ideas in granular synthesis. The contextualised interpretation of this quote would lean more towards the ability to generate ametric structures without a clearly discernible pulse. Here it has been taken in a different direction: towards envisioning musical time as a dynamic feedback loop of metre perception, expectational prediction, and rhythm production. This loop was termed *metrical flux*.

Throughout this thesis systems were described that holistically capture and model metrical flux. Even though these systems operate in continuous time, meaning there are no assumptions of tempo or metre, the model is limited to a single time-series representation of rhythm and an indicator of tempo change. In reality, musical time, rhythm, and time-varying expression is a complex, multi-faceted behaviour. Modelling just the rhythmic onsets will only get us so far before incorporating ideas of other musical expectations, such as pitch, dynamics, and desired emotional response, which all exist in a state of interactive flux. This thesis takes some steps along this path, but there is much more work to be done if we are ever to create an intelligent musical agent that could perform alongside a human performer as an equal.

Bibliography

Allen, Paul E. and Roger B. Dannenberg (1990). 'Tracking musical beats in real time'. In: *Proceedings of the 1990 International Computer Music Conference*. San Fancisco, CA, pp. 140–3.

Ancona, Deborah and Chee-Leong Chong (1996). 'Entrainment: Pace, Cycle, and Rhythm in Organizational Behavior'. In: *Research in Organizational Behavior*. Ed. by B. M. Staw and L. L. Cummings. Vol. 18. US: Elsevier Science/JAI Press, pp. 251–284. ISBN: 1-55938-938-9.

Angelis, Vassilis et al. (2013). 'Testing a Computational Model of Rhythm Perception Using Polyrhythmic Stimuli'. In: *Journal of New Music Research* 42.1, pp. 47–60. ISSN: 0929-8215. DOI: 10.1080/09298215.2012.718791.

Assayag, Gérard et al. (2006). 'Omax Brothers: A Dynamic Topology of Agents for Improvization Learning'. In: *Proceedings of the 1st ACM Workshop on Audio and Music Computing Multimedia*. Santa Barbara, CA: Association for Computing Machinery (ACM), pp. 125–32. DOI: 10.1145/1178723.1178742.

Bello, Juan Pablo et al. (2005). 'A Tutorial on Onset Detection in Music Signals'. In: *IEEE Transactions on Speech and Audio Processing* 13.5, pp. 1035–47. DOI: TSA.2005.851998.

Bengio, Yoshua (2009). 'Learning Deep Architectures for AI'. In: *Foundations and trends® in Machine Learning* 2.1, pp. 1–127. DOI: 10.1561/2200000006.

Boden, Margaret A and Ernest A Edmonds (2009). 'What is Generative Art?' In: *Digital Creativity* 20.1-2, pp. 21–46. DOI: 10.1080/14626260902867915.

Bååth, Rasmus, Erik Lagerstedt and Peter Gärdenfors (2013). 'An Oscillator Model of Categorical Rhythm Perception'. In: *Proceedings of the 35th Annual Conference of the Cognitive Science Society*. Ed. by M Knauff et al. Austin, TX: Cognitive Science Society, pp. 1803–8.

Böck, Sebastian and Markus Schedl (2011). 'Enhanced Beat Tracking with Context-Aware Neural Networks'. In: *Proceedings of the 14th International Conference on Digital Audio Effects (DAFx-11)*. Paris, France, pp. 135–9.

Böck, Sebastian and Gerhard Widmer (2013). 'Maximum Filter Vibrato Suppression for Onset Detection'. In: *Proceedings of the 16th International Conference on Digital Audio Effects (DAFx-13)*. Maynooth, Ireland.

Camacho, Erika, Richard Rand and Howard Howland (2004). 'Dynamics of Two van der Pol Oscillators Coupled via a Bath'. In: *International Journal of Solids and Structures* 41.8, pp. 2133–43. DOI: 10.1016/j.ijsolstr.2003.11.035.

Cheng, Eric and Elaine Chew (2008). 'Quantitative Analysis of Phrasing Strategies in Expressive Performance: Computational Methods and Analysis of Performances of Unaccompanied Bach for Solo Violin'. In: *Journal of New Music Research* 37.4, pp. 325–38. ISSN: 0929-8215. DOI: 10.1080/09298210802711660.

Cherla, Srikanth, Tillman Weyde and Artur S d'Avila Garcez (2014). 'Multiple Viewpiont Melodic Prediction with Fixed-Context Neural Networks.' In: *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR 2014)*. Taipei, Taiwan, pp. 101–106.

Cherla, Srikanth et al. (2013). 'A Distributed Model For Multiple-Viewpoint Melodic Prediction.' In: *Proceedings of the 14th International Society for Music Information Retrieval Conference (ISMIR 2013)*. Curitiba, Brazil, pp. 15–20.

Cherla, Srikanth et al. (2015). 'Discriminative Learning and Inference in the Recurrent Temporal RBM for Melody Modelling'. In: *Proceeding of*

the 2015 International Joint Conference on Neural Networks (IJCNN). IEEE, pp. 1–8. DOI: 10.1109/IJCNN.2015.7280691.

Chew, Elaine (2016). 'Playing with the Edge'. In: *Music Perception: An Interdisciplinary Journal* 33.3, pp. 344–66. ISSN: 0730-7829, 1533-8312. DOI: 10.1525/mp.2016.33.3.344.

Chew, Elaine and Clifton Callender (2013). 'Conceptual and Experiential Representations of Tempo: Effects on Expressive Performance Comparisons'. In: *Mathematics and Computation in Music*. Ed. by Jason Yust, Jonathan Wild and John Ashley Burgoyne. Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 76–87. ISBN: 978-3-642-39356-3 978-3-642-39357-0. DOI: 10.1007/978-3-642-39357-0_6.

Chomsky, Noam (1957). *Syntactic Structures*. Berlin: Mouton de Gruyter.

Chuan, Ching-Hua and Elaine Chew (2007). 'A Dynamic Programming Approach to the Extraction of Phrase Boundaries from Tempo Variations in Expressive Performances'. In: *Proceedings of the 8th International Society for Music Information Retrieval Conference (ISMIR 2007)*. Vienna, Austria, pp. 305–308.

Clarke, Eric F (1999). 'Rhythm and Timing in Music'. In: *The psychology of music*. Ed. by Diana Deutsch. Second Edition. San Diego: Academic Press, pp. 473–500.

Clarke, Eric F. (2001). 'Generative Principles in Music Performance'. In: *Generative Processes in Music: The Psychology of Performance, Improvisation, and Composition*. Ed. by John Sloboda. Oxford: Oxford University Press.

Clayton, Martin, Rebecca Sager and Udo Will (2005). 'In Time with the Music: The Concept of Entrainment and its Significance for Ethnomusicology'. In: *European Meetings in Ethnomusicology* 11, pp. 3–142. ISSN: 1582-5841.

Coca, A.E., D.C. Correa and Liang Zhao (2013). 'Computer-aided Music Composition with LSTM Neural Network and Chaotic Inspiration'. In: *The 2013 International Joint Conference on Neural Networks (IJCNN)*. Dallas, TX: IEEE, pp. 1–7. DOI: 10.1109/IJCNN.2013.6706747.

Cohn, Richard (2001). 'Complex Hemiolas, Ski-Hill Graphs and Metric Spaces'. English. In: *Music Analysis* 20.3, pp. 295–326. ISSN: 02625245. DOI: 10.1111/1468-2249.00141.

Coleman, David (2014). *2014 Christmas Lectures - Sparks will fly: How to hack your home*. London, UK. (TV appearance).

Collins, Nick (2008). 'The Analysis of Generative Music Programs'. In: *Organised Sound* 13.3, pp. 237–48. DOI: 10.1017/S1355771808000332.

Collins, Nick and Andrew R. Brown (2009). 'Generative Music Editorial'. In: *Contemporary Music Review* 28.1, pp. 1–4. ISSN: 0749-4467. DOI: 10.1080/07494460802663967.

Colton, Simon (2012). 'The Painting Fool: Stories from Building an Automated Painter'. In: *Computers and Creativity*. Ed. by John McCormack and d'Inverno Mark. Springer, pp. 3–38.

Conklin, Darrell and Ian H Witten (1995). 'Multiple Viewpoint Systems for Music Prediction'. In: *Journal of New Music Research* 24.1, pp. 51–73. ISSN: 09298215. DOI: 10.1080/09298219508570672.

Cope, David (1992). 'Computer Modeling of Musical Intelligence in EMI'. In: *Computer Music Journal* 16.2, pp. 69–83. DOI: 10.2307/3680717.

Dahl, George E et al. (2012). 'Context-dependent Pre-trained Deep Neural Networks for Large-vocabulary Speech Recognition'. In: *IEEE Transactions on Audio, Speech, and Language Processing* 20.1, pp. 30–42. ISSN: 15587916. DOI: 10.1109/TASL.2011.2134090.

Davies, Matthew E.P. and Mark D. Plumbley (2007). 'Context-Dependent Beat Tracking of Musical Audio'. In: *IEEE Transactions on Audio, Speech, and Language Processing* 15.3, pp. 1009–20. ISSN: 1558-7916. DOI: 10.1109/TASL.2006.885257.

Dixon, Simon (2001a). 'An Empirical Comparison of Tempo Trackers'. In: *Proceedings of the 8th Brazilian Symposium on Computer Music*. Fortaleza, Brazil, pp. 832–40.

— (2001b). 'Automatic Extraction of Tempo and Beat from Expressive Performances'. In: *Journal of New Music Research* 30.1, pp. 39–58. ISSN: 09298215. DOI: 10.1076/jnmr.30.1.39.7119.

— (2007). 'Evaluation of the Audio Beat Tracking System Beatroot'. In: *Journal of New Music Research* 36.1, pp. 39–50. ISSN: 09298215. DOI: 10.1080/09298210701653310.

Dixon, Simon and Werner Goebl (2002). 'Pinpointing the Beat: Tapping to Expressive Performances'. In: *Proceedings of the International Conference on Music Perception and Cognition*. Sydney, Australia, pp. 617–20.

Dixon, Simon, Fabien Gouyon and Gerhard Widmer (2004). 'Towards Characterisation of Music via Rhythmic Patterns'. In: *Proceedings of the 5th International Society for Music Information Retrieval Conference (ISMIR 2004)*. Barcelona, Spain, pp. 509–17.

Eck, D. and J. Schmidhuber (2002). 'Finding Temporal Structure in Music: Blues Improvisation with LSTM Recurrent Networks'. In: *Proceedings of the 2002 12th IEEE Workshop on Neural Networks for Signal Processing*. New York, NY, pp. 747–56. DOI: 10.1109/NNSP.2002.1030094.

Eck, Douglas (2001). 'A Network of Relaxation Oscillators that Finds Downbeats in Rhythms'. In: *Artificial Neural Networks — ICANN 2001*. Ed. by Georg Dorffner, Horst Bischof and Kurt Hornik. Lecture Notes in Computer Science 2130. Springer Berlin Heidelberg, pp. 1239–47. ISBN: 978-3-540-42486-4 978-3-540-44668-2.

— (2002). 'Finding Downbeats with a Relaxation Oscillator'. In: *Psychological Research* 66.1, pp. 18–25. ISSN: 0340-0727, 1430-2772. DOI: 10.1007/s004260100070.

Eck, Douglas, Michael Gasser and Robert Port (2000). 'Dynamics and Embodiment in Beat Induction'. In: *Rhythm Perception and Production*. Ed. by Peter Desain and Luke Windsor. Lisse, The Netherlands: Swets & Zeitlinger, pp. 157–70.

Eigenfeldt, Arne (2015). 'Generative Music for Live Musicians: An Unnatural Selection'. In: *Proceedings of the Sixth International Conference on Computational Creativity*. Park City, UT.

Eigenfeldt, Arne et al. (2013). 'Towards a Taxonomy of Musical Metacreation: Reflections on the First Musical Metacreation Weekend'. In: *Proceedings of the Ninth Artificial Intelligence and Interactive Digital Entertainment Conference (AIIDE'13)*. Boston, MA, pp. 40–7.

Eigenfeldt, Arne et al. (2014). 'Video, Music and Sound Metacreation'. In: *xCoAx 2014: Proceedings of the Second Conference on Computation, Communication, Aesthetics and X*. Porto, Portugal.

Elmsley, Andrew J. (2016). *Modelling Metrical Flux: Adaptive Oscillator Networks for Expressive Rhythmic Perception and Prediction*. Queen Mary University of London Cognitive Science Group. London, UK. (Talk).

Elmsley, Andrew J., Tillman Weyde and Newton Armstrong (2017). 'Generating Time: Rhythmic Perception, Prediction and Production with Recurrent Neural Networks'. In: *Journal of Creative Music Systems* 1.2. DOI: 10.5920/JCMS.2017.04.

Epstein, David (1995). *Shaping Time: Music, the Brain, and Performance*. New York: Schirmer.

FitzHugh, Richard (1961). 'Impulses and Physiological States in Theoretical Models of Nerve Membrane'. In: *Biophysical Journal* 1.6, pp. 445–66. ISSN: 00063495. DOI: 10.1016/S0006-3495(61)86902-6.

Franklin, Judy A. (2006). 'Recurrent Neural Networks for Music Computation'. In: *INFORMS Journal on Computing* 18.3, pp. 321–338. ISSN: 1091-9856. DOI: 10.1287/ijoc.1050.0131.

Funahashi, Ken-ichi and Yuichi Nakamura (1993). 'Approximation of Dynamical Systems by Continuous Time Recurrent Neural Networks'. In: *Neural Networks* 6.6, pp. 801–06. ISSN: 0893-6080. DOI: 10.1016/S0893-6080(05)80125-X.

Gabrielsson, Alf and Erik Lindström (2010). 'The Role of Structure in the Musical Expression of Emotions'. In: *Handbook of music and emotion: Theory, research, applications*. Ed. by Patrik N. Juslin. Oxford: Oxford University Press, pp. 367–400.

Gasser, Michael, Douglas Eck and Robert Port (1999). 'Meter as Mechanism: A Neural Network Model that Learns Metrical Patterns'. In:

Connection Science 11.2, pp. 187–216. ISSN: 0954-0091. DOI: 10.1080/095400999116331.

Gers, F.A. and J. Schmidhuber (2001). 'LSTM Recurrent Networks Learn Simple Context-free and Context-sensitive Languages'. In: *IEEE Transactions on Neural Networks* 12.6, pp. 1333–40. ISSN: 1045-9227. DOI: 10.1109/72.963769.

Gers, Felix A., Jürgen Schmidhuber and Fred Cummins (2000). 'Learning to Forget: Continual Prediction with LSTM'. In: *Neural Computation* 12.10, pp. 2451–71. ISSN: 0899-7667. DOI: 10.1162/089976600300015015.

Goto, Masataka (2001). 'An Audio-based Real-time Beat Tracking System for Music with or without Drum-sounds'. In: *Journal of New Music Research* 30.2, pp. 159–171. ISSN: 09298215. DOI: 10.1076/jnmr.30.2.159.7114.

Gouyon, Fabien and Simon Dixon (2005). 'A Review of Automatic Rhythm Description Systems'. In: *Computer Music Journal* 29.1, pp. 34–54. ISSN: 01489267. DOI: 10.1162/comj.2005.29.1.34.

Gouyon, Fabien et al. (2006). 'An Experimental Comparison of Audio Tempo Induction Algorithms'. In: *IEEE Transactions on Audio, Speech and Language Processing* 14.5, pp. 1832–44. ISSN: 15587916. DOI: 10.1109/TSA.2005.858509.

Grondin, Simon (2008). *Psychology of Time*. Bingley: Emerald Group Publishing. ISBN: 978-0-08-046977-5.

Grosche, Peter, Meinard Müller and Craig Stuart Sapp (2010). 'What Makes Beat Tracking Difficult? A Case Study on Chopin Mazurkas'. In: *Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR 2011)*. Utrecht, Netherlands, pp. 649–54.

Hinton, Geoffrey E, Simon Osindero and Yee-Whye Teh (2006). 'A Fast Learning Algorithm for Deep Belief Nets'. In: *Neural computation* 18.7, pp. 1527–54. DOI: https://doi.org/10.1162/neco.2006.18.7.1527.

Hochreiter, S and J Schmidhuber (1997). 'Long Short-Term Memory'. In: *Neural Computation* 9.8, pp. 1735–80. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735.

Holzapfel, Andre et al. (2012). 'Selective Sampling for Beat Tracking Evaluation'. In: *IEEE Transactions on Audio, Speech, and Language Processing* 20.9, pp. 2539–48. ISSN: 1558-7916. DOI: 10.1109/TASL.2012.2205244.

Honing, Henkjan (2012). 'Without it no Music: Beat Induction as a Fundamental Musical Trait'. In: *Annals of the New York Academy of Sciences* 1252.1, pp. 85–91. ISSN: 00778923. DOI: 10.1111/j.1749-6632.2011.06402.x.

Hoppensteadt, Frank C and Eugene M Izhikevich (1996). 'Synaptic Organizations and Dynamical Properties of Weakly Connected Neural Oscillators II. Learning Phase Information'. In: *Biological Cybernetics* 75.2, pp. 129–35. ISSN: 03401200. DOI: 10.1007/s004220050280.

Huron, David Brian (2006). *Sweet Anticipation: Music and the Psychology of Expectation*. Cambridge, MA: MIT Press.

Huygens, Christiaan (1673). *Horologium oscillatorium, sive de motu Pendulorum ad Horologia aptato demonstrationes geometricae*. Paris: Muguet.

Igel, Christian and Michael Hüsken (2000). 'Improving the Rprop Learning Algorithm'. In: *Proceedings of the Second International ICSC Symposium on Neural Computation (NC 2000)*. Berlin, Germany: ICSC Academic Press, pp. 115–21.

Jones, Mari R. (1976). 'Time, our Lost Dimension: Toward a New Theory of Perception, Attention, and Memory'. In: *Psychological Review* 83.5, pp. 323–55. ISSN: 0033-295X. DOI: 10.1037/0033-295X.83.5.323.

Jordanous, Anna (2011). 'Evaluating Evaluation: Assessing Progress in Computational Creativity Research'. In: *Proceedings of the Second International Conference on Computational Creativity (ICCC-11)*. Mexico City, Mexico, pp. 102–7.

— (2012). 'A Standardised Procedure for Evaluating Creative Systems: Computational Creativity Evaluation Based on what it is to be Creative'. In: *Cognitive Computation* 4.3, pp. 246–79. ISSN: 18669956. DOI: 10.1007/s12559-012-9156-1.

Kalos, A. (2006). 'Modeling MIDI Music as Multivariate Time Series'. In: *IEEE Congress on Evolutionary Computation, 2006 (CEC 2006)*. Vancouver, Canada, pp. 2058–64. DOI: 10.1109/CEC.2006.1688560.

Kempter, Richard, Wulfram Gerstner and J Leo van Hemmen (1999). 'Hebbian Learning and Spiking Neurons'. In: *Physical Review E* 59.4, pp. 4498–514. ISSN: 1063651X. DOI: 10.1103/PhysRevE.59.4498.

Kirke, Alexis and Eduardo Reck Miranda (2009). 'A Survey of Computer Systems for Expressive Music Performance'. In: *ACM Computing Surveys* 42.1, 3:1–3:41. ISSN: 0360-0300. DOI: 10.1145/1592451.1592454.

Klapuri, Anssi P. (2004). 'Automatic Music Transcription as We Know it Today'. In: *Journal of New Music Research* 33.3, pp. 269–82. ISSN: 09298215. DOI: 10.1080/0929821042000317840.

Klapuri, Anssi P., Antti J. Eronen and Jaakko T. Astola (2006). 'Analysis of the Meter of Acoustic Musical Signals'. In: *IEEE Transactions on Audio, Speech, and Language Processing* 14.1, pp. 342–55. ISSN: 1558-7916. DOI: 10.1109/TSA.2005.854090.

Knoester, David B and Philip K McKinley (2011). 'Evolving virtual fireflies'. In: *Advances in Artificial Life: Darwin Meets von Neumann*. Ed. by George Kampis, István Karsai and Eörs Szathmáry. Vol. 5778. Lecture Notes in Computer Science. Springer, pp. 474–81. ISBN: 978-3-642-21313-7.

Kohavi, Ron (1995). 'A Study of Cross-validation and Bootstrap for Accuracy Estimation and Model Selection'. In: *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI)*. Vol. 2. Montreal, Canada, pp. 1137–45.

Korzeniowski, Filip, Sebastian Böck and Gerhard Widmer (2014). 'Probabilistic Extraction of Beat Positions from a Beat Activation Function.' In: *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR 2014)*. Taipei, Taiwan, pp. 513–18.

Krebs, Florian, Sebastian Böck and Gerhard Widmer (2013). 'Rhythmic Pattern Modeling for Beat and Downbeat Tracking in Musical Audio'. In: *Proceedings of the 14th International Society for Music Information Retrieval Conference (ISMIR 2013)*. Curitiba, Brazil, pp. 227–32.

Krebs, Harald (1999). *Fantasy pieces: Metrical Dissonance in the Music of Robert Schumann*. Oxford: Oxford University Press.

Krizhevsky, Alex, Ilya Sutskever and Geoffrey E Hinton (2012). 'Imagenet Classification with Deep Convolutional Neural Networks'. In: *Advances in Neural Information Processing Systems 25 (NIPS 2012)*. Stateline, NV, pp. 1097–105.

Kuramoto, Yoshiki (1984). *Chemical oscillations, waves and turbulence*. Berlin: Springer.

Lambert, Andrew (2012). 'A Stigmergic Model for Oscillator Synchronisation and its Application in Music Systems'. In: *Proceedings of the International Computer Music Conference*. Ljubljana, Slovenia, pp. 247–52.

Lambert, Andrew and Florian Krebs (2015). 'The Second International Workshop on Cross-disciplinary and Multicultural Perspectives on Musical Rhythm and Improvisation'. In: *Computer Music Journal* 39.2, pp. 97–100.

Lambert, Andrew, Tillman Weyde and Newton Armstrong (2014a). 'Beyond the Beat: Towards Metre, Rhythm and Melody Modelling with Hybrid Oscillator Networks'. In: *Proceedings of the Joint 40th International Computer Music Conference and 11th Sound & Music Computing Conference*. Athens, Greece, pp. 485–90.

— (2014b). 'Studying the Effect of Metre Perception on Rhythm and Melody Modelling with LSTMs'. In: *Proceedings of the Tenth Artificial Intelligence and Interactive Digital Entertainment Conference*. Raleigh, NC, pp. 18–24.

Lambert, Andrew J. (2014a). *A Fractal Depth for Interactive Music Systems*. Music Research Seminar. London, UK. (Talk).

— (2014b). *Beyond the Beat: Towards an Expressive Depth in Generative Music*. NYUAD Rhythm Workshop. Abu Dhabi, UAE. (Talk).

— (2014c). *MUME Methodologies: Presentation, Promotion and Appraisal*. 3rd International Workshop on Musical Metacreation (MUME 2014). Raleigh, NC. (Panel Chair).

— (2014d). *Towards Metre, Rhythm and Melody Modelling with Hybrid Oscillator Networks*. City Informatics Research Symposium. London, UK. (Talk).

— (2015). *Machine Perception and Generation of Metre and Rhythm*. Music Research Seminar. London, UK. (Talk).

— (2016a). *Creative Music Systems: Bridging the Divide Between Academia and Industry?* 1st Conference on Computer Simulation of Musical Creativity. Huddersfield, UK. (Panelist).

— (2016b). *Creative Music Systems: Current Capacities and Future Prospects*. 1st Conference on Computer Simulation of Musical Creativity. Huddersfield, UK. (Talk).

Lambert, Andrew J., Tillman Weyde and Newton Armstrong (2014c). *Deep Rhythms: Towards Structured Meter Perception, Learning and Generation with Deep Recurrent Oscillator Networks*. DMRN+8. London, UK. (Poster).

— (2015a). *Generating Time: An Expressive Depth for Rhythmic Perception, Prediction and Production with Recurrent Neural Networks*. Study Day on Computer Simulation of Musical Creativity. Huddersfield, UK. (Talk).

— (2015b). 'Perceiving and Predicting Expressive Rhythm with Recurrent Neural Networks'. In: *12th Sound & Music Computing Conference*. Maynooth, Ireland, pp. 265–72.

— (2015c). *Rhythmic Perception, Prediction and Production with Recurrent Neural Networks*. UVA Music Cognition Group. Amsterdam, Netherlands. (Talk).

— (2015d). *Tracking Expressive Timing with Gradient Frequency Neural Networks*. City University Graduate Symposium. London, UK. (Poster).

— (2016a). 'Adaptive Frequency Neural Networks for Dynamic Pulse and Metre Perception'. In: *Proceedings of the 17th International Society for Music Information Retrieval Conference (ISMIR 2016)*. New York, NY, pp. 60–6.

Lambert, Andrew J., Tillman Weyde and Newton Armstrong (2016b). *Adaptive Frequency Neural Networks for Dynamic Pulse and Metre Perception*. Workshop on Auditory Neuroscience, Cognition and Modelling. London, UK. (Poster).

— (2016c). *Adaptivity in Oscillator-based Pulse and Metre Perception*. CogMIR. New York, NY. (Poster).

— (2016d). *An Adaptive Oscillator Neural Network for Beat Perception in Music*. City University Graduate Symposium. London, UK. (Talk).

— (2016e). 'Metrical Flux: Towards Rhythm Generation in Continuous Time'. In: *4th International Workshop on Musical Metacreation, held at the Seventh International Conference on Computational Creativity, ICCC 2016*. Paris, France.

Large, E. W. et al. (2014). *GrFNN Toolbox 1.0: Matlab Tools for Simulating Signal Processing, Plasticity and Pattern Formation in Gradient Frequency Neural Networks*.

Large, Edward W. (1995). 'Beat Tracking with a Nonlinear Oscillator'. In: *Working Notes of the IJCAI-95 Workshop on Artificial Intelligence and Music*. Montreal, Quebec, pp. 24–31.

— (2010). 'Neurodynamics of Music'. en. In: *Music Perception*. Ed. by Mari R. Jones, Richard R. Fay and Arthur N. Popper. Springer Handbook of Auditory Research 36. Springer New York, pp. 201–31. ISBN: 978-1-4419-6113-6 978-1-4419-6114-3.

Large, Edward W., Felix V. Almonte and Marc J. Velasco (2010). 'A Canonical Model for Gradient Frequency Neural Networks'. In: *Physica D: Nonlinear Phenomena* 239.12, pp. 905–11. ISSN: 0167-2789. DOI: 10.1016/j.physd.2009.11.015.

Large, Edward W., Jorge A. Herrera and Marc J. Velasco (2015). 'Neural Networks for Beat Perception in Musical Rhythm'. In: *Frontiers in Systems Neuroscience* 9.159. ISSN: 1662-5137. DOI: 10.3389/fnsys.2015.00159.

Large, Edward W. and Mari R. Jones (1999). 'The Dynamics of Attending: How People Track Time-varying Events'. In: *Psychological Review* 106.1, pp. 119–59. ISSN: 0033-295X. DOI: 10.1037/0033-295X.106.1.119.

Large, Edward W. and John F. Kolen (1994). 'Resonance and the Perception of Musical Meter'. In: *Connection Science* 6.2-3, pp. 177–208. ISSN: 0954-0091. DOI: 10.1080/09540099408915723.

Lerdahl, Fred and Ray Jackendoff (1983a). *A Generative Theory of Tonal Music*. Cambridge, MA: MIT press.

— (1983b). 'An Overview of Hierarchical Structure in Music'. In: *Music Perception* 1.2, pp. 229–52. ISSN: 07307829. DOI: 10.2307/40285257.

Levy, Mark, Mark B Sandler and Michael A Casey (2006). 'Extraction of High-Level Musical Structure From Audio Data and Its Application to Thumbnail Generation'. In: *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*. Toulouse, France, pp. 13–6. ISBN: 1-4244-0469-X. DOI: 10.1109/ICASSP.2006.1661200.

London, Justin (2012). *Hearing in Time: Psychological Aspects of Musical Meter*. Oxford University Press. ISBN: 978-0-19-974437-4.

Madison, Guy (2009). 'An Auditory Illusion of Infinite Tempo Change Based on Multiple Temporal Levels'. In: *PLoS ONE* 4.12, e8151. ISSN: 19326203. DOI: 10.1371/journal.pone.0008151.

Maxwell, James B et al. (2012). 'MusiCOG: A Cognitive Architecture for Music Learning and Generation'. In: *Proceedings of the Sound and Music Computing Conference*. Copenhagen, Denmark.

Mazzola, Guerino and Oliver Zahorka (1993). *Geometry and Logic of Musical Performance I, II, III*. SNSF Research Reports. Zürich: Universität Zürich.

McAuley, J. Devin (1995). 'Perception of Time as Phase: Toward an Adaptive-oscillator Model of Rhythmic Pattern Processing'. PhD thesis. Indiana University Bloomington.

Michaels, Donald C, Edward P Matyas and Jose Jalife (1987). 'Mechanisms of Sinoatrial Pacemaker Synchronization: A New Hypothesis.' In: *Circulation Research* 61.5, pp. 704–14. ISSN: 00097330. DOI: 10.1161/01.RES.61.5.704.

Mozer, Michael C. (1994). 'Neural Network Music Composition by Prediction: Exploring the Benefits of Psychoacoustic Constraints and Multiscale Processing'. In: *Connection Science* 6.2-3, pp. 247–80. ISSN: 09540091. DOI: 10.1080/09540099408915726.

Nagumo, Jinichi, S Arimoto and S Yoshizawa (1962). 'An Active Pulse Transmission Line Simulating Nerve Axon'. In: *Proceedings of the IRE* 50.10, pp. 2061–70. DOI: 10.1109/JRPROC.1962.288235.

Nestke, Andreas and Thomas Noll (2001). 'Inner Metric Analysis'. In: *Harmonic Analysis and Tone Systems*. Ed. by Ján Haluška. Berlin: Tatra Mountains Mathematical Publications, pp. 91–111.

Nguyen, Anh, Jason Yosinski and Jeff Clune (2015). 'Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images'. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, pp. 427–36. DOI: 10.1109/CVPR.2015.7298640.

Nierhaus, Gerhard (2009). 'Algorithmic Composition: Paradigms of Automated Music Generation'. In: *Computer Music Journal* 34.3, pp. 70–4. DOI: 10.1162/comj_r_00008.

Nika, Jérôme et al. (2014). 'Planning Human-Computer Improvisation'. In: *Joint 40th International Computer Music Conference and 11th Sound & Music Computing Conference*. Athens, Greece, pp. 1290–97.

Pantaleone, James (2002). 'Synchronization of Metronomes'. In: *American Journal of Physics* 70.10, pp. 992–1000. ISSN: 0002-9505, 1943-2909. DOI: 10.1119/1.1501118.

Patel, Aniruddh D et al. (2009). 'Experimental Evidence for Synchronization to a Musical Beat in a Nonhuman Animal'. In: *Current Biology* 19.10, pp. 827–30. DOI: 10.1016/j.cub.2009.05.023.

Pearce, Marcus, David Meredith and Geraint Wiggins (2002). 'Motivations and Methodologies for Automation of the Compositional Process'. In: *Musicae Scientiae* 6.2, pp. 119–47. DOI: 10.1177/102986490200600203.

Pearce, Marcus Thomas (2005). 'The Construction and Evaluation of Statistical Models of Melodic Structure in Music Perception and Composition'. PhD thesis. City University London.

Pol, Balth van der and Jan van der Mark (1928). 'The Heartbeat Considered as a Relaxation Oscillation, and an Electrical Model of the Heart'. In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 6.38, pp. 763–75. DOI: 10.1080/14786441108564652.

Povel, Dirk-Jan and Peter Essens (1985). 'Perception of Temporal Patterns'. In: *Music Perception: An Interdisciplinary Journal* 2.4, pp. 411–40. DOI: 10.2307/40285311.

Rankin, Summer K., Edward W. Large and Philip W. Fink (2009). 'Fractal Tempo Fluctuation and Pulse Prediction'. In: *Music Perception* 26.5, pp. 401–13. DOI: 10.1525/mp.2009.26.5.401.

Righetti, Ludovic, Jonas Buchli and Auke J. Ijspeert (2006). 'Dynamic Hebbian Learning in Adaptive Frequency Oscillators'. In: *Physica D: Nonlinear Phenomena* 216.2, pp. 269–81. DOI: 10.1016/j.physd.2006.02.009.

Roads, Curtis (2014). 'Rhythmic Processes in Electronic Music'. In: *Joint 40th International Computer Music Conference and 11th Sound & Music Computing conference*. Athens, Greece, pp. 27–31.

Räsänen, Esa et al. (2015). 'Fluctuations of Hi-Hat Timing and Dynamics in a Virtuoso Drum Track of a Popular Music Recording'. In: *PLOS ONE* 10.6, e0127902. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0127902.

Scardapane, Simone et al. (2016). 'Learning Activation Functions from Data Using Cubic Spline Interpolation'. In: *arXiv:1605.05509 [cs, stat]*. arXiv: 1605.05509.

Schaffrath, Helmut (1995). *The Essen Folksong Collection in Kern Format*.

Schaul, Tom et al. (2010). 'PyBrain'. In: *Journal of Machine Learning Research* 11.Feb, pp. 743–46.

Scheirer, Eric D. (1998). 'Tempo and Beat Analysis of Acoustic Musical Signals'. In: *The Journal of the Acoustical Society of America* 103.1, pp. 588–601. ISSN: 0001-4966. DOI: 10.1121/1.421129.

Small, Christopher (2011). *Musicking: The Meanings of Performing and Listening*. Wesleyan University Press.

Stowell, Dan and Elaine Chew (2012). 'Maximum a Posteriori Estimation of Piecewise Arcs in Tempo Time-Series'. In: *From Sounds to Music and Emotions*. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp. 387–399. DOI: 10.1007/978-3-642-41248-6_22.

Strogatz, Steven H. (2001). *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology and Chemistry*. Boulder, CO: Perseus.

Strogatz, Steven H. and Ian Stewart (1993). 'Coupled Oscillators and Biological Synchronization'. In: *Scientific American* 269.6, pp. 102–9. DOI: 10.1038/scientificamerican1293-102.

Sturm, Bob L et al. (2016). 'Music Transcription Modelling and Composition using Deep Learning'. In: *arXiv preprint arXiv:1604.08723*.

Taigman, Yaniv et al. (2014). 'Deepface: Closing the Gap to Human-level Performance in Face Verification'. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Columbus, OH, pp. 1701–8. DOI: 10.1109/cvpr.2014.220.

Todd, Neil (1989a). 'A Computational Model of Rubato'. In: *Contemporary Music Review* 3.1, pp. 69–88. ISSN: 0749-4467. DOI: 10.1080/07494468900640061.

Todd, Peter M. (1989b). 'A Connectionist Approach to Algorithmic Composition'. In: *Computer Music Journal* 13.4, pp. 27–43. DOI: 10.2307/3679551.

Velasco, Marc J. and Edward W. Large (2011). 'Pulse Detection in Syncopated Rhythms using Neural Oscillators'. In: *12th International Society for Music Information Retrieval Conference*. Miami, FL, pp. 185–90.

Vera, Bogdan, Elaine Chew and Patrick GT Healey (2013). 'A Study of Ensemble Synchronisation Under Restricted Line of Sight'. In: *Proceedings*

of the 14th International Society for Music Information Retrieval Conference (ISMIR 2013). Curitiba, Brazil, pp. 293–98.

Volk, Anja (2003). 'The Empirical Evaluation of a Mathematical Model for Inner Metric Analysis'. In: *Proceedings of the 5th Triennal ESCOM Conference*. Hanover, Germany, pp. 467–70.

— (2008). 'Persistence and Change: Local and Global Components of Metre Induction using Inner Metric Analysis'. In: *Journal of Mathematics and Music* 2.2, pp. 99–115. DOI: 10.1080/17459730802312399.

Werbos, Paul J. (1990). 'Backpropagation through Time: What it does and How to do it'. In: *Proceedings of the IEEE* 78.10, pp. 1550–60. DOI: 10.1109/5.58337.

Whitelaw, Mitchell (2004). *Metacreation: Art and Artificial Life*. Cambridge, MA: MIT Press.

Whiteley, Nick, Ali T. Cemgil and Simon J. Godsill (2006). 'Bayesian Modelling of Temporal Structure in Musical Audio'. In: *Proceedings of the 7th International Society for Music Information Retrieval Conference*. Victoria, Canada, pp. 29–34.

Widmer, Gerhard and Werner Goebl (2004). 'Computational Models of Expressive Music Performance: The State of the Art'. In: *Journal of New Music Research* 33.3, pp. 203–16. DOI: 10.1080/0929821042000317804.