Metrical Flux: Towards Rhythm Generation in Continuous Time

Andrew J. Lambert, Tillman Weyde and **Newton Armstrong**
City University London

Abstract

This work-in-progress report describes our approach to expressive rhythm generation. So far, music generation systems have mostly focused on discrete time modelling. Since musical performance and perception unfolds in time, we see continuous time modelling as a more realistic approach. Here we present work towards a continuous time rhythm generation system.

In our model, two neural networks are combined within one integrated system. A novel Adaptive Frequency Neural Network (AFNN) models the perception of changing periodicities in metrical structures by entraining and resonating nonlinearly with a rhythmic input. A Recurrent Neural Network models longer-term temporal relations based on the AFNN's response and generates rhythmic events.

We outline an experiment and evaluation method to validate the model and invite the MUME community's feedback.

Introduction

> "Composition is not a matter of filling or dividing time, but rather of generating time." (Roads 2014)

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 rhythm we perceive within the music that we can tap along to. 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, phases and even higher order forms (Lerdahl and Jackendoff 1983).

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 the pulse in subtly complex ways, and sometimes employ this as an expressive device (Räsänen et al. 2015; Clarke 1988).

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 dynamics (Kirke and Miranda 2009).

In the above quote, Roads muses that the composer has the power to provide a subjective experience of time to the listener, via their perception of rhythmic events. Roads (2014) 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.

As the performer expressively varies the temporal dynamics, the perceived metrical structure is perturbed. Even when the outer metrical structure remains consistent, which is often the case, the listener's perception of musical time is affected, along with any expectation of rhythmical events. Thus, any endogenous sense of pulse and metre is always in flux throughout the listening process.

Our research explores a machine learning approach to expressive rhythm generation addressing all of the aspects above. Rather than separate rhythm generation into two distinct event creation and expressive playback phases, we are attempting a holistic approach based on cognitive models of metre perception. Our system outputs in continuous time, meaning there is no prior or external knowledge of tempo or metre beyond a single time-series input. In order to achieve this continuous generative output, we also propose methods for improving the modelling and processing of rhythm, pulse and metre in computer science, which can still struggle with varying tempo and expressive timing.

Our proposed generative model incorporates a novel variation on the Gradient Frequency Neural Network (GFNN; Large 2010), which we have named an Adaptive Frequency Neural Network (AFNN). An AFNN is an oscillating neural network model based on the neuro-cognitive model of nonlinear resonance and models the way the nervous system resonates to auditory rhythms. AFNNs model the perception of changing periodicities in metrical structures by applying a Hebbian learning rule to the oscillator frequencies in the network. We have found that, compared with GFNNs, AFNNs can produce a better response to stimuli with both steady and varying pulses.

The AFNN is paired with a Recurrent Neural Network (RNN), which models the longer-term temporal relations of the AFNN's response and can be trained to generate new rhythmic events through a prediction task.

The rest of this paper is structured as follows: Section provides a very brief and non-exhaustive overview of some relevant literature, Section details our proposed model, and finally Section outlines a rhythm generation experiment we are planning to undertake to validate and evaluate the proposed model.

Background

Metrical Flux

The Generative Theory of Tonal Music (Lerdahl and Jackendoff 1983) was one of the first and most influential attempts to create formal models of the hierarchical structures inferred by listeners when listening to music. One such hierarchy is *metrical structure*, which are layers of beats existing in a hierarchically layered relationship with the rhythm. Each metrical level is associated with its own period, which divides the previous level's period into a certain number of parts.

Humans often choose a common, comfortable metrical level to tap along to, which is commonly referred to as 'the beat'. This this is a problematic term since a beat can also refer to a singular rhythmic event or a metrically inferred event. To avoid that ambiguity, we use the term 'pulse' (Grondin 2008).

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 based off 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).

Automatically processing an audio signal to determine pulse events is known as *beat tracking* and has a long history of research dating back to 1990 (Allen and Dannenberg 1990). Some early work by Large used a single nonlinear oscillator to track beats in performed piano music (Large 1995). More recently, Böck et al. (Böck, Krebs, and Widmer 2015) used resonating comb filters with a type of RNN called a Long Short-Term Memory Network (LSTM) to achieve a state-of-the-art beat tracking result. 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).

Generative Rhythmic Expression

Todd (1989b) and Mozer (1994) were among the first to utilise a machine learning approach to music generation. This approach has some advantages over rule-based systems, which can be strict, lack novelty, and not deal with unexpected inputs very well. Instead, the structure of existing musical examples are learned by the system and generalisations are made from these learned structures to compose new pieces.

Both Todd and Mozer's systems are RNNs that are trained to predict melody and rhythm. 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 metrically windowed samples.

It is rare for generative music systems to produce temporal variations in their output, but a generative system that outputs an abstract symbolic rhythm could always have that rhythm 'played' by a computer system for expressive music performance (CSEMP). Research into CSEMPs is a small but important field within Computer Music. In general, CSEMPs have received relatively little attention from both academia and the industry at large (Kirke and Miranda 2009). Widmer and Goebl (2004) have published an overview of existing computational models, and Kirke and Miranda (2009) have produced a survey of available CSEMPs.

Some research has been done on rhythmic expression modelling, such as Todd's computational model of rubato (Todd 1989a), which is one of the most common expressive devices when performing music. Todd's model incorporates a hierarchic model for timing units from a piecewise global scale to beat-wise local scale. An internal representation is formed and 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.

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 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. One example of this is Eigenfeldt's *An Unnatural Selection* (2015). In a form of improvisation, 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 this to be generated by the system itself.

Proposed Model

System Overview

Our system takes a holistic approach to rhythm generation and expressive timing; it is a rhythm generating system which includes an expressive timing model its output.

Our previous work used a similar model utilising GFNNs. We trained the system to predict both pitch and rhythm in metrically-quantised time-series data of folk melodies (Lambert, Weyde, and Armstrong 2014). In an-

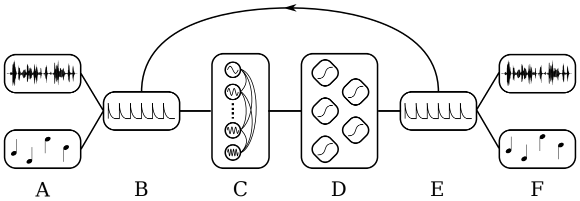

Figure 1: An overview of our proposed model showing (A) audio or symbolic input, (B) time-series rhythm representation, (C) AFNN, (D) RNN, (E) time-series rhythm prediction, and (F) audio or symbolic output. An internal feedback loop connects E and B.

other study, an expressive rhythm prediction experiment using a piano corpus showed encouraging rhythm prediction accuracy (Lambert, Weyde, and Armstrong 2015).

Figure 1 shows an overview of the model. There is a singular input (A), which could be symbolic or audio data. This is converted into a into a time-series data signal (B), retaining only rhythmic onsets. We choose a relatively high sample rate (86.15Hz) to minimise any metric quantisation and retain timing variance.

An AFNN (C), a bank of nonlinear resonating oscillators, is stimulated by the signal and forms the core pulse and metre modelling in the system. AFNNs are a novel contribution and are described in more detail below. Before the AFNN stage, the model could still be described as a discrete time model, but since the AFNN is a system of differential equations integrated through a time-step, we are using a continuous time model from which we sample values at discrete time points.

The resonances formed in (C) are then used as inputs to an RNN (D). RNNs have excellent time-series prediction properties and have been trained to predict expressively-timed rhythmic onsets. For our system we choose the LSTM network, due to its ability to learn long-term dependencies (Hochreiter and Schmidhuber 1997), and its prior success at generating musical structures (Eck and Schmidhuber 2002). The RNN'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.

Adaptive Frequency Neural Network

The system described relies on signal processing and machine learning models that have been tried and tested for decades, even within music generation software. One novel contribution of our research has been the creation and inclusion of the AFNN, which is a variation on the GFNN.

The neuro-cognitive model of nonlinear resonance models the way the nervous system resonates to auditory rhythms by representing a population of neurons as a canonical nonlinear oscillator (Large 2010). A GFNN consists of a number of canonical oscillators distributed across a frequency spectrum, and has been shown to predict beat induction behaviour from humans (Large, Herrera, and Velasco 2015). Figure 2 shows the amplitude of oscillators over time in a GFNN when stimulated by in isochronous rhythm. The resonant response of the network adds rhythm-harmonic frequency information to the signal, and the GFNN's entrainment properties allow each oscillator to phase shift, resulting in deviations from their natural frequencies. Canonical oscillators will resonate to an external stimulus that contains frequencies at integer ratio relationships to its natural frequency. This sets nonlinear resonance apart from many linear filtering methods such as the resonating comb filters used in (Klapuri, Eronen, and Astola 2006) and Kalman filters (Kalman 1960). This makes GFNNs good candidates for modelling the perception of temporal dynamics in music.

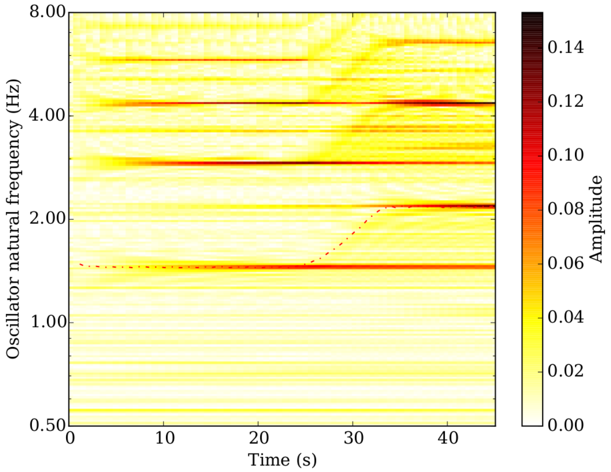

Figure 2: GFNN amplitude output over time. The dashed line shows stimulus frequency.

However, we have found that GFNNs can sometimes become noisy, especially when the pulse frequency fluctuates (Lambert, Weyde, and Armstrong 2014; 2015). This can be observed in Figure 2; at approximately 25s a tempo change occurs but a resonant memory of the previous stimulation persists. This persistent resonance causes an interference effect on the output of the network.

In our previous work, we have addressed this issue by taking an average of the oscillators as a single 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, this is not an ideal solution as new frequencies would not be selected until they begin to resonate above the selection threshold, meaning that new resonances in changing tempos may be missed.

The AFNN attempts to address both the interference within GFNNs, and improve the GFNNs ability to track changing frequencies, by introducing a Hebbian learning rule on oscillator frequencies. This rule is an adapted form of the general model introduced by Righetti, Buchli, and Ijspeert (2006). Their method depends on an external driving

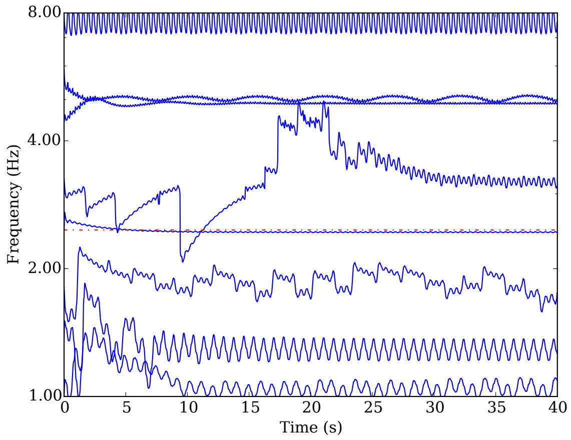

Figure 3: AFNN frequencies adapting to a sinusoidal stimulus.

stimulus ($x(t)$), and the state of the oscillator in terms of amplitude (r) and phase (φ), driving the frequency (ω) toward the frequency of the stimulus. The frequency adaptation is influenced by the choice of a force scaling parameter, ϵ. ϵ also scales with r, meaning that higher amplitudes (stronger resonances) are affected less by the rule.

This method differs from other adaptive oscillator models such as McAuley's phase-resetting model (McAuley 1995) by maintaining a Hebbian biological plausibility. It is also a general method that has been proven to be valid for limit cycles of any form.For this proof, see (Righetti, Buchli, and Ijspeert 2006).

We have adapted this rule to also include a linear elasticity. The elastic force is an implementation of Hooke's Law, which describes a force that strengthens with displacement. We have introduced this rule to ensure the AFNN retains a spread of frequencies (and thus metrical structure) across the gradient. The elastic force is relative to natural frequency, and can be scaled through its own parameter. Eq. (1) shows the final adaptive rule:

$$\frac{d\omega}{dt} = -\frac{\epsilon_f}{r}x(t)sin(\varphi) - \frac{\epsilon_h}{r}(\frac{\omega - \omega_0}{\omega_0}) \tag{1}$$

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 3 shows the frequencies adapting over time in the AFNN under sinusoidal input. These two new interacting adaptive rules allow for a great reduction in the number of oscillators in the network (compared with a GFNN), which minimises oscillator interference whilst also maintaining a frequency spread across the gradient. We have conducted initial experiments with AFNNs and observed improved response from AFNNs with 16 oscillators, compared with the 289 oscillators used in (Large, Herrera, and Velasco 2015).

Evaluation

At the time of writing this paper, we feel we have gathered sufficient evidence to suggest that the model outlined above is viable as an online interactive generative rhythm system. We are now in the process of planning an experiment to validate and evaluate the generative outputs of the system.

The RNN requires training before it can be used in production mode. In a similar vein to a previous expressive rhythm prediction experiment (Lambert, Weyde, and Armstrong 2015), the RNN layer will be trained to predict rhythm onsets based off the AFNN's input. Once trained, the RNN will then be capable of generating new rhythms rendered in a similar expressive feel to the training corpus.

We have selected a symbolic corpus of monophonic Jazz solos as the target data for this, named the *Jazzomat* dataset (Frieler et al. 2013). This dataset is a comprehensive and representative database of jazz solos, transcribed from audio releases.

There are several benefits to choosing this dataset. Firstly it is monophonic, meaning that we avoid having to assign voices to the output rhythm. Secondly, the pieces are categorised by several rhythmic classes in terms style, genre and 'rhythm feel'. We can train several models, segmenting the training into these categories in addition to a combined version. Finally, using an annotated symbolic corpus means we can statistically compare features of our generated rhythms with the corpus, as well as perform phrase segmentation to create meaningful extracts for the training. Exploring other datasets is left for future work.

The RNN layer will be trained with cross-validation on extracted excerpts from the dataset. The internal feedback loop will be disabled during this training process. An auxiliary output of the experiment will be an analysis of how well the AFNN and RNN can capture and model the dataset.

Once training is complete we can evaluate the models with quantitative and statistical metrics such as F-measure, and select the best performing networks to be put into 'production mode'. In production mode the feedback loop will be reactivated, and the network will be seeded with an initial rhythmic pattern. The feedback loop will cause the system to generate new rhythms which will be recorded ready for the qualitative evaluation stage.

The tentative design of the qualitative evaluation takes the form of an online listening test. Both the generated rhythms and test phrases extracted from the dataset (unseen during training) will be rendered to audio. Users will be presented with three patterns at a time, which will be a mixture of generated and ground-truth data. The listeners will be asked to make similarity and quality judgements on two or more generated and ground-truth rhythms. We will gather some information about the listeners such as music qualifications and tastes, but the listening test will be anonymous and online-only.

This evaluation will give us some insight as to the validity of the model. Through the relative similarity scores we can identify if our generated rhythms sit within the same perceptual space as those from the dataset. If there is a discrepancy we can try to explain it through the quantitative measures.

In future we hope to gather further qualitative data through an interactive experiment involving the system and human musicians.

Conclusions

In this paper we have proposed a system and evaluation method for continuous-time rhythm generation. This work is still in progress and as such we invite the MUME community's feedback on any part of our process.

Acknowledgements

Andrew Lambert is supported by a PhD studentship from City University London.

References

Allen, P. E., and Dannenberg, R. B. 1990. Tracking musical beats in real time. In *Proceedings of the 1990 International Computer Music Conference*, 140–3.

Assayag, G.; Bloch, G.; Chemillier, M.; Cont, A.; and Dubnov, S. 2006. Omax brothers: a dynamic yopology of agents for improvization learning. In *Proceedings of the 1st ACM workshop on Audio and music computing multimedia*, 125–32.

Böck, S.; Krebs, F.; and Widmer, G. 2015. Accurate tempo estimation based on recurrent neural networks and resonating comb filters. In *Proceedings of the 16th International Society for Music Information Retrieval Conference*, 625–31.

Clarke, E. F. 1988. Generative principles in music performance.

Clarke, E. F. 1999. Rhythm and timing in music. *The psychology of music* 2:473–500.

Eck, D., and Schmidhuber, J. 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*, 747–56.

Eigenfeldt, A. 2015. Generative Music for Live Musicians: An Unnatural Selection. In *Proceedings of the Sixth International Conference on Computational Creativity*.

Epstein, D. 1995. *Shaping time: Music, the brain, and performance*. Schirmer Books New York.

Frieler, K.; Abeßer, J.; Zaddach, W.-G.; and Pfleiderer, M. 2013. Introducing the Jazzomat project and the MeloPy library. In *Proceedings of the Third International Workshop on Folk Music Analysis*, 76–8.

Gabrielsson, A., and Lindström, E. 2010. The role of structure in the musical expression of emotions. In *Handbook of music and emotion: Theory, research, applications*. 367–400.

Grondin, S. 2008. *Psychology of Time*. Emerald Group Publishing.

Grosche, P.; Müller, M.; and Sapp, C. S. 2010. What Makes Beat Tracking Difficult? A Case Study on Chopin Mazurkas. In *in Proceedings of the 11th International Society for Music Information Retrieval Conference*, 649–54.

Hochreiter, S., and Schmidhuber, J. 1997. Long Short-Term Memory. *Neural Computation* 9(8):1735–80.

Holzapfel, A.; Davies, M. E.; Zapata, J. R.; Oliveira, J. L.; and Gouyon, F. 2012. Selective Sampling for Beat Tracking Evaluation. *IEEE Transactions on Audio, Speech, and Language Processing* 20(9):2539–48.

Honing, H. 2012. Without it no music: beat induction as a fundamental musical trait. *Annals of the New York Academy of Sciences* 1252(1):85–91.

Kalman, R. E. 1960. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering* 82(1):35–45.

Kirke, A., and Miranda, E. R. 2009. A Survey of Computer Systems for Expressive Music Performance. *ACM Comput. Surv.* 42(1):3:1–3:41.

Klapuri, A. P.; Eronen, A. J.; and Astola, J. T. 2006. Analysis of the meter of acoustic musical signals. *IEEE Transactions on Audio, Speech, and Language Processing* 14(1):342–55.

Lambert, A.; Weyde, T.; and Armstrong, N. 2014. Studying the Effect of Metre Perception on Rhythm and Melody Modelling with LSTMs. In *Tenth Artificial Intelligence and Interactive Digital Entertainment Conference*, 18–24.

Lambert, A. J.; Weyde, T.; and Armstrong, N. 2015. Perceiving and Predicting Expressive Rhythm with Recurrent Neural Networks. In *12th Sound & Music Computing Conference*.

Large, E. W.; Herrera, J. A.; and Velasco, M. J. 2015. Neural networks for beat perception in musical rhythm. *Frontiers in Systems Neuroscience* 9(159).

Large, E. W. 1995. Beat tracking with a nonlinear oscillator. In *Working Notes of the IJCAI-95 Workshop on Artificial Intelligence and Music*, 24–31.

Large, E. W. 2010. Neurodynamics of Music. In Jones, M. R.; Fay, R. R.; and Popper, A. N., eds., *Music Perception*, number 36 in Springer Handbook of Auditory Research. Springer New York. 201–31.

Lerdahl, F., and Jackendoff, R. 1983. *A generative theory of tonal music*. Cambridge, Mass.: MIT press.

McAuley, J. D. 1995. *Perception of time as phase: Toward an adaptive-oscillator model of rhythmic pattern processing*. Ph.D. Dissertation, Indiana University Bloomington.

Mozer, M. C. 1994. Neural network music composition by prediction: Exploring the benefits of psychoacoustic constraints and multi-scale processing. *Connection Science* 6(2-3):247–80.

Nika, J.; Echeveste, J.; Chemillier, M.; and Giavitto, J.-L. 2014. Planning Human-Computer Improvisation. In *Joint 40th International Computer Music Conference and 11th Sound & Music Computing conference*, 330.

Rankin, S. K.; Large, E. W.; and Fink, P. W. 2009. Fractal Tempo Fluctuation and Pulse Prediction. *Music Perception: An Interdisciplinary Journal* 26(5):401–13.

Righetti, L.; Buchli, J.; and Ijspeert, A. J. 2006. Dynamic hebbian learning in adaptive frequency oscillators. *Physica D: Nonlinear Phenomena* 216(2):269–81.

Roads, C. 2014. Rhythmic Processes in Electronic Music. In *Joint 40th International Computer Music Conference and 11th Sound & Music Computing conference*.

Räsänen, E.; Pulkkinen, O.; Virtanen, T.; Zollner, M.; and Hennig, H. 2015. Fluctuations of Hi-Hat Timing and Dynamics in a Virtuoso Drum Track of a Popular Music Recording. *PLOS ONE* 10(6).

Todd, N. 1989a. A computational model of rubato. *Contemporary Music Review* 3(1):69–88.

Todd, P. M. 1989b. A Connectionist Approach to Algorithmic Composition. *Computer Music Journal* 13(4):27–43.

Widmer, G., and Goebl, W. 2004. Computational models of expressive music performance: The state of the art. *Journal of New Music Research* 33(3):203–16.