Investigating the Musical Affordances of Continuous Time Recurrent Neural Networks

Steffan Ianigro
Architecture, Design and Planning,
University of Sydney,
Darlington NSW 2008, Australia
steffanianigro@gmail.com

Oliver Bown
Art and Design,
University of New South Wales,
Paddington NSW 2021, Australia
o.bown@unsw.edu.au

Abstract

This paper investigates the musical affordances of Continuous Time Recurrent Neural Networks (CTRNNs) as an evolvable low-level algorithm for the exploration of sound. Our research will be divided into two parts. Firstly, we will conduct various studies that provide insight into CTRNN behaviours, identifying aspects that could prove creatively valuable to musicians. We expect to find that this system will exhibit musical behaviours reminiscent of conventional audio processing methods such as amplitude modulation and additive synthesis, as well as producing interesting temporal structures. In the second part of this paper, we will discuss how these interesting behaviours can be harnessed by musicians. Specifically we investigate how evolutionary search can be used to exploit the compact low-level structure of CTRNNs and explore their potential for audio diversity beyond the capabilities of more traditional methods of audio exploration.

Introduction

Evolutionary Algorithms (EAs) have been widely explored as tools for musical composition, demonstrated in the survey by Husbands et al. (2007). EAs are highly abstract biological models and provide an effective search heuristic for solving complex problems. Of particular interest to the authors are EAs used as creative tools for the exploration of audio synthesis algorithms such as described in (Yee-King and Roth 2008; Dahlstedt 2007). Despite the success of these systems, the question of what audio representations maximise both exploitability and variety is still a debated issue. For example, McCormack (2008) emphasises the potential for creativity afforded by searching low-level structures for creative artefacts, such as manipulating pixels of an image in search of interesting artworks. However, McCormack also identifies the futility of searching these low-level representations, as although they may be capable of extensive diversity, artefacts of any interest will take an impractical amount of time to find. This example refers to brute force random search, but the same notion is true when evolving low-level audio representations, such as manipulating individual samples of an audio waveform to synthesise interesting sounds. We could evolve almost any audio possibility, but if the genetic representation is too broad, the vastness of the system's search space would render many of these creative prospects unreachable.

In pursuit of more explorable audio representations, many authors of EAs embed higher-level software structures within their creations that they think will yield interesting results (McCormack 2008), constraining the system's creative search space within manageable limits. For example, when evolving the parameters of a commercial synthesiser, a high-level of abstraction or a meta relationship exists between the parameters that are being manipulated by the EA (genotype) and the resulting audio produced by the structure of the device (phenotype). As a result, the system's output is constrained by the capabilities of its components as individuals produced by the system will exhibit strong traits of the underlying formalised structures that created them. This means that the outputs of the system will be of a specific 'class', defined by the audio representation or parameterisations selected by the system's creator (McCormack 2008). This reduction of the audio search space means that the system is more manageable to explore and thus creatively useful, a solution that may prove sufficient if a user just wants to explore permutations of an existing system, but what if a user seeks audio with greater spectral complexity or variety beyond the capabilities of the plethora of music-making devices at their disposal?

Within this paper, we propose evolving Continuous Time Recurrent Neural Networks (CTRNNs) as an alternative, providing a low-level audio representation with a compact explorable genotype structure, capable of exhibiting complex dynamics that could afford interesting sonic possibilities that are otherwise hard to achieve using more conventional synthesis approaches. However, discovery of these complex dynamics can be problematic, as although there is much research on CTRNNs covering a range of domains, little is know about their behaviours when used as audio synthesis mechanisms, raising questions such as: how do CTRNN parameter changes translate to the audio domain?; do CTRNNs behave similarly to more conventional audio synthesis mechanisms?; and how can users effectively discover their scope of audio possibilities?

We aim to address these questions through two empirical investigations. In the first part of this paper, we will conduct various CTRNN studies in an attempt to discover and understand behaviours that may prove creatively valuable to a musician. We have identified four interesting dynamics: the introduction of temporal and pitch structures; a strong relationship between CTRNN inputs and outputs; amplitude modulation characteristics; and additive synthesis capabilities. An online interactive appendix of selected figures from these studies can be found at `www.plecto.io/ICCC2016appendix`. In the second part of this paper, we discuss how evolution can be used to discover and shape CTRNN behaviours, allowing musicians to harness their idiosyncratic dynamics. Specifically, we ask questions about what types of EA structures will afford effective creative search of their parametric space and propose future research directions for implementing CTRNNs as evolvable structures for audio exploration.

Background

Related Work

Artificial Neutral Networks (ANNs) have been used for many different functions in music, from beat tracking algorithms (Lambert, Weyde, and Armstrong 2015) to artificial composers that can extract stylistic regularities from existing pieces and produce novel musical compositions based on these learnt musical structures (Mozer 1994). Bown and Lexer (2006) offer another application, proposing the use of CTRNNs to create autonomous software agents that exhibit musicality. Bown and Lexer also outline the possibility of using CTRNNs as audio synthesis algorithms, a prospect which inspired this research.

A notable example of similar work is discussed by Ohya (1995), who describes a system that trains a Recurrent Neural Network to match an existing piece of audio. The network structure can then be manipulated to synthesise variants of the original sound. Eldridge (2005) provides another example, exploring the use of Continuous Time Neural Models for audio synthesis. In previous work (Ianigro and Bown 2016), we propose a system that allows users to interactively evolve CTRNNs to produce aesthetically desirable outputs for use in their artistic practices. In this paper we build on our system, *Plecto*, and further investigate the behaviour of CTRNNs within the audio signal domain.

CTRNNs

CTRNNs are nonlinear continuous dynamical systems that can exhibit complex temporal behaviours (Beer 1995). They are well suited to produce audio output as various configurations result in smooth oscillations that resemble audio waveforms. They are an interconnected network of computer-modelled neurons, typically of a type called the *leaky integrator*. The internal state of each neuron is determined by the differential equation (1),

$$\tau_i(dy_i/dt) = -y_i + \sum W_{ij}\sigma(g_j(y_j - b_j)) + I_i \qquad (1)$$

where τ_i is the time constant, g_i is the gain and b_i is the bias for neuron i. I_i is any external input for neuron i and W_{ij} is the weight of the connection between neuron i and neuron j. σ is a $tanh$ non-linear transfer function (Bown and Lexer 2006).

The behaviour of a neuron is defined by three parameters - gain, bias and time constant - and each neuron input has a weight parameter that governs its strength over the neuron's other inputs (Bown and Lexer 2006). CTRNNs are continuous, recurrent, and due to their complex dynamics, they are often trained using an EA. For this research, we adopt a fully connected CTRNN, meaning that the neurons in the hidden layer are all connected and the input layer has a full set of connections to the hidden layer. Each hidden neuron also has a self connection, enhancing its behavioural complexity. The output or activation of each neuron is calculated using a $tanh$ transfer function, providing outputs between -1 and 1 for use as samples in an audio wavetable (the CTRNN output is the activation of a selected hidden neuron).

Evolutionary Search

Many EAs are based on Darwinian theory, with evolutionary change a result of the fittest of each generation surviving and passing on the traits that made them fit (Husbands et al. 2007). These algorithms provide a powerful method for searching a problem space, optimising candidates until the best solution is found. There are two main type of EAs: target based EAs which evaluate individuals according to a criterion that is encoded into the system, and interactive EAs that incorporate human evaluation as their selective pressure. The latter is often used for creative applications, with the user assuming the role of a 'pigeon breeder', acting as a selective pressure in an artificial environment (Bown 2009). This is an appealing prospect as it is difficult to define explicit fitness functions for audio phenotypes that can identify subjective creatively desirable traits (Tokui and Iba 2000). There are also many other types of EAs for creative exploration, such as the ecosystem model described in (McCormack 2001) and the use of artificial immune systems such as discussed in (Abreu, Caetano, and Penha 2016).

Evolution of Neural Networks

The growing area of neuroevolution refers to the optimisation of neural networks using EAs (Stanley and Miikkulainen 2002). This is an effective approach when training CTRNNs; unlike methods such as back-propagation in which network weights are adjusted to minimise the network error, multiple features of the network can be evolved at one time. The definition of an EA's performance criterion is also more flexible than the definition of an energy or error function (Floreano, Dürr, and Mattiussi 2008). There is a variety of work in this area, such as (Jónsson, Hoover, and Risi 2015; Hoover and Stanley 2007), describing systems that evolve neural networks in pursuit of creative artefacts. In this paper, we adopt a similar method of network optimisation, using an EA to manipulate gain, bias, time constant and weight parameters of the CTRNN that provide a compact genotype capable of producing extensive diversity. We will use an Artificial Immune System (AIS), a type of evolutionary optimisation algorithm called *opt-aiNet* (Timmis and Edmonds 2004) to achieve this.

Musical Behaviours of CTRNNs

In this section, we will conduct CTRNN studies by randomly generating network configurations, feeding various types of audio inputs into these networks and observing the results through audio analysis. This process will provide an insight into behaviours that are common within the search spaces of CTRNNs, such as if CTRNNs have dynamics similar to more conventional audio generation algorithms and how consistent these behaviours are, informing our expectations of the evolvability of audio CTRNNs.

Generation of Temporal and Pitch Structures

A contributing factor to the complexity that CTRNNs are able to produce is the presence of neuron self connections (Beer 1995). A strong self connection can dominate the neuron input, saturating the neuron by locking it in a certain state (emitting a constant output). This behaviour is analogous to an internal switch that can influence the behaviour of the rest of the network, creating interesting temporal dynamics that afford many creative possibilities such as described in (Bown and Lexer 2006). To investigate the audio synthesis implications of this behaviour, we adjusted the hidden neuron self connection weight of a CTRNN with one input neuron and one hidden neuron whilst feeding in the audio sample notated in Figure 1. This experiment produced some interesting results, as once the hidden neuron weight passed a certain threshold, the CTRNN alternated between states of saturation (outputs a constant value of -1 or 1) and oscillation. This is evident in Figure 2, showing the original unprocessed audio which was used as the input for the CTRNN (top) and the processed CTRNN output (bottom). In the case of this exploration, the CTRNN primarily responded to the amplitude fluctuations caused by the kick drum. However, as its hidden neuron self connection is adjusted, the degree of saturation changes, exhibiting graceful degradation of the behaviour, almost like tuning the sensitivity of a conventional signal gate used in audio production.

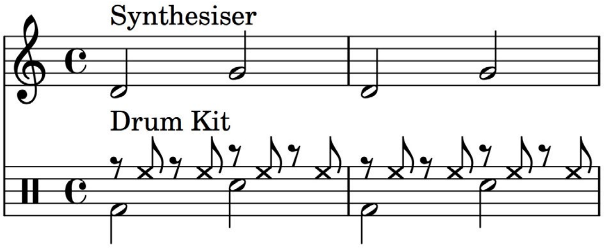

Figure 1: Notated version of the CTRNN input used to produce Figures 2, 3 and 4. The Drum Kit consists of a bass drum (open note head) and a hi-hat (cross note head). The synthesiser has a timbre very similar to a sine wave.

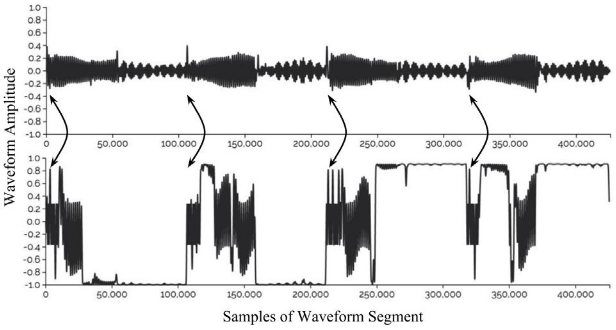

Figure 2: Comparison of unprocessed input notated in Fig. 1 (top) to its processed counterpart (bottom), showing how certain input values saturate the CTRNN's hidden neuron and others do not.

We further tested the consistency of this behaviour by adopting another input for the CTRNN that exhibits a more sporadic melody absent of any percussion. We were able achieve a similar result by tweaking the CTRNN's hidden neuron self connection weight parameter until we observed similar saturation fluctuations. Furthermore, we found that if sinusoid inputs are adopted instead of more complex audio samples, the results are less interesting as the neuron saturates and remains so, emphasising that the continuous flipping of neurons is caused by amplitude fluctuations in the neuron input. But what are the musical implications of this behaviour in larger CTRNN structures?

In Figure 3, we can see the output of a larger CTRNN configuration (one input neuron and three hidden neurons) with an input of a small audio sample consisting of a two note melody and rhythmic accompaniment (notated in Fig. 1). The CTRNN's original input melody is evident in part one (each bar consisting of notes D and G), however, in part two we can see the introduction of new melodic and rhythmic content (notes D, C, A, G and F). The timbre of the CTRNN's input also varies and the synthesiser's percussive accompaniment is removed. Through further randomisation of the CTRNN's parameters, we found another CTRNN configuration that produces similar behaviour (output notated in Fig. 4), identifying that this introduction of temporal and pitch structures is not a one off occurrence but can occur in various forms within the CTRNN search space. These larger CTRNN outputs have similarities to the neuron saturation behaviour described earlier. For example, if we compare both Figures 3 and 4 to their original input material (Fig. 1), we can see that these introduced temporal and pitch structures coincide with the rhythmic events in the CTRNN's input material, such as when the hi-hat symbol is struck. Therefore, it appears that CTRNN input amplitude fluctuations are flipping neuron states within the network, shifting the musical structure of the CTRNN's output. This is a more complex manifestation of the behaviour seen in Figure 2 and has many creative implications, affording a means to generate temporal and pitch structures. Furthermore, the complex neuron interactions within CTRNNs can produce unexpected outputs, exhibiting agency or Musical Metacreativity (Eigenfeldt et al. 2013) during the compositional process.

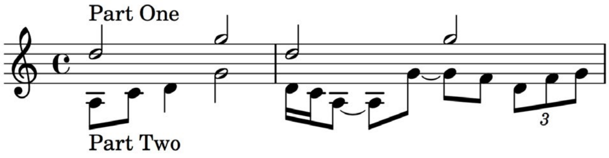

Figure 3: Simplified notation of CTRNN's output (input notated in Fig. 1). We transposed the melody up one octave for legibility and the frequencies produced by the CTRNN are converted to their closest equal tempered note values.

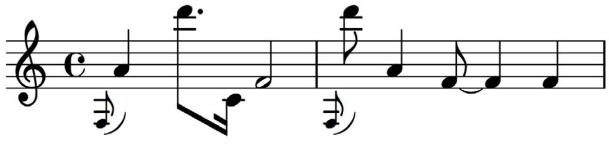

Figure 4: Simplified notation of CTRNN's output (input notated in Fig. 1). We transposed the melody up one octave for legibility and the frequencies produced by the CTRNN are converted to their closest equal tempered note values.

Strong Input/Output Relationship

The strong relationship we have observed between CTRNN inputs and outputs highlight that CTRNN behaviour can be similar to that of a modular synthesiser or digital signal processing (DSP) effects module, altering their input structure towards creatively exciting directions. Further evidence of this dynamic can be seen in the pitch structure of both Figures 3 and 4, with the additional note values exhibiting similar pitch structures to sections of a harmonic series based the CTRNN's input melody. For example, in Figure 3, when the note D of the input melody is playing, we also hear a counter melody consisting of A, C and D, which are the 6th, 7rd and 8th overtones of a harmonic series with a fundamental of D. This DSP effect-like dynamic could afford some interesting possibilities for the use of CTRNNs as building blocks within a larger, modular system, a possibility we will further discuss in the final section of this paper.

Amplitude Modulation

Through further analysis of the CTRNN configurations that produced Figures 3 and 4, we noticed some CTRNN behaviour similar to that produced by an amplitude modulation synthesis algorithm. This form of audio modulation follows a general rule that if two signals are multiplied, two partials result (called sidebands), one at the sum of the two frequencies and one at the difference (Puckette 2007). We can see evidence of this behaviour in both Figures 5 and 6, displaying spectrogram outputs of the same CTRNNs that produced Figures 3 and 4, except a sinusoid waveform oscillating at 523Hz was used as their inputs instead of the more complex input notated in Figure 1. Sideband structures are evident around the CTRNN input frequencies in both figures at ratios typical of amplitude modulation. This behaviour is interesting as we can see CTRNNs are not only an evolvable structure capable of generating interesting rhythmic and pitch structures, but afford possibilities for timbral variation. Puckette (2007) also identifies that amplitude modulation can be used as an octave divider, offering a possible explanation for the overtone structures that appear below their fundamental frequencies in Figures 3 and 4.

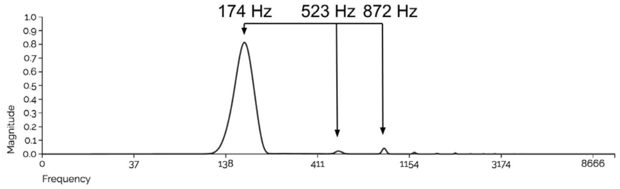

Figure 5: Sidebands that correspond to the multiplication of the sinusoid CTRNN input oscillating at 523Hz and a frequency of 349Hz. Frequency values are approximate (+-3Hz).

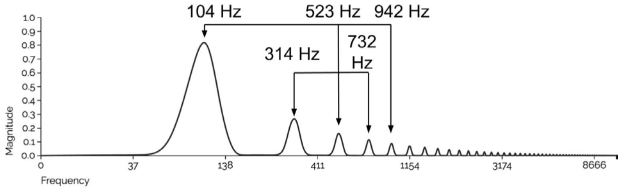

Figure 6: Sidebands that correspond to the multiplication of the sinusoid CTRNN input oscillating at 523Hz and a frequency of 419Hz as well as multiplication with a frequency of 209Hz. Frequency values are approximate (+-3Hz).

Additive Synthesis

Neurons within CTRNNs can also oscillate at fixed frequencies independent of their input. In larger CTRNNs, a dynamic analogous to additive synthesis (Puckette 2007) can result in which neuron oscillations are summed with either other neuron or CTRNN input oscillations to produce a more complex audio waveform. Figure 7 shows this relationship, exhibiting the CTRNN's sinusoid input oscillating at 523Hz (top) which appears to be summed with a lower frequency (caused by neuron oscillations at 86Hz (+-3Hz) within the CTRNN), producing a multi-phonic CTRNN output (bottom). These summed frequencies in the CTRNN's output can also change independently of each other, evident in Figure 8. At the top, we can see a spectrogram produced by the same CTRNN that produced Figure 7 when fed a sinusoid input oscillating at 523Hz. The bottom also shows a spectrogram produced by this CTRNN except we used a sinusoid oscillating at 1000Hz as its input. We can see the presence of the same neuron oscillations at about 86Hz in both spectrograms, however the other dominant oscillations present in the CTRNN outputs vary in regard to the CTRNN's input frequency. It is worth noting that if the CTRNN's sinusoid input oscillates at a rate below about 375Hz, we lose this additive synthesis behaviour, demonstrating the non-linear dynamics of CTRNNs.

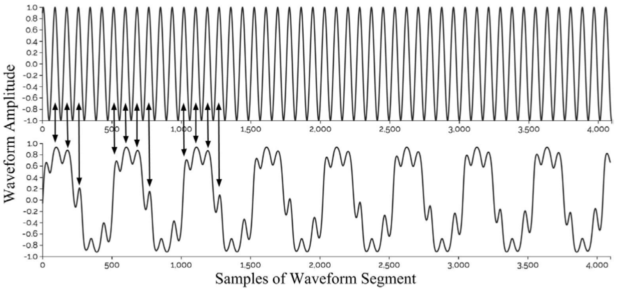

Figure 7: Comparison of a sinusoid oscillating at 523Hz (top) to the CTRNN's output it produced (bottom), demonstrating CTRNN additive synthesis capabilities.

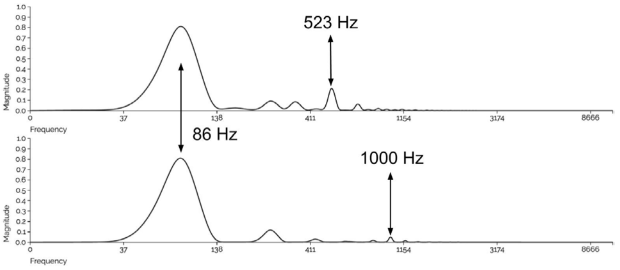

Figure 8: Comparison of spectrograms produced by a CTRNN with different sinusoid inputs (top: 523Hz, bottom: 1000Hz) showing the independent relationship between the CTRNN neuron oscillations (at about 86Hz) and the CTRNN's input.

Through these studies, we have found the occurrence of multiple musically interesting audio synthesis characteristics of CTRNNs. The variety of behaviours we have observed also hint at the generality or scope for audio variety that CTRNNs are capable of producing. However, their complex dynamics raise many questions about how to discover and utilise these creative possibilities within the workflows of musicians. In the next section, we will discuss a means to explore the creative possibilities of CTRNNs using an EA.

Evolution of CTRNNs

Many different EA designs exist that have been used for creative search. Within this section, we adopt a model based on the *opt-aiNet* algorithm conceived by de Castro and Timmis (de Castro and Timmis 2002), a multimodal optimisation algorithm inspired by some of the evolutionary properties of the human immune system (de Franca, Von Zuben, and de Castro 2005). This is an appealing model for our purpose as it can maintain many candidate solutions to a problem, providing not only the global optimum but also many of the local optima in a search space (Timmis and Edmonds 2004). This method has also shown promise for use as a sound matching utility (Abreu, Caetano, and Penha 2016), a use case we adopt within this section.

In order to both measure how effective the *opt-aiNet* EA is for searching the creative possibilities of CTRNNs, as well as further understanding the creative capabilities of CTRNNs, we conduct a sound matching experiment in which CTRNNs (with one input neuron, ten hidden neurons and a constant input value of 0) are evolved towards five different drone-like audio targets. These selected audio sample targets cover a range of timbral profiles, which if successfully matched, will identify that the low-level functionality of CTRNNs affords a varied creative search space of audio possibilities that is explorable by an EA. Additionally, we will discuss another use for the *opt-aiNet* EA structure as a Novelty Search (NS) algorithm which rewards candidates that are unique in some way compared to existing individuals (Lehman and Stanley 2008). This is an interesting prospect for exploring the sound possibilities of CTRNNs without needing a predefined target.

opt-aiNet EA Design

The *opt-aiNet* algorithm follows a general structure outlined below. Differing from more conventional EA structures, this model incorporates sub-populations, each locally optimised with the fittest individual of each sub-population added to the main population for global evaluation during each algorithm iteration. These sub-populations are generated by cloning and mutating each member of the global population, with mutation rates inversely proportionate to the parent individual's fitness. This EA model also discourages convergence on a specific area of the search space using a population suppression mechanism. Once the population stagnates (the difference between average fitness errors over time is below a predefined threshold), individuals of the global population are compared using a distance metric and individuals with a close similarity are removed (higher fitness individuals are maintained). A number of randomly generated individuals are then introduced into the population (its size can vary dynamically) to facilitate thorough exploration of the EA's search space (Timmis and Edmonds 2004).

1. Randomly initialise the population.
2. While the stopping criterion is not met, continue, else save the global population to a database.
   - I Calculate the fitness of each individual in the global population.
   - II Generate a number of clones for each individual, creating sub populations.
   - III Mutate each clone inversely proportionate to its parent's fitness (fitter individuals are mutated less).
   - IV Determine the fitness of individuals within each sub population including the parent individual and remove all but the fittest, which replaces the parent cell in the global population.
   - V Calculate the average distance from the algorithm target and if the population stagnates, continue to steps 3 else go back to step 2.
3. Re-calculate the fitness of each individual in the global population after the fittest individuals of the sub-populations replace their parents.
4. Determine the highest affinity individuals (similar phenotype) and perform population suppression to avoid redundancy whilst maintaining the fittest individuals.

5. Introduce a number of randomly generated individuals and go back to step 2.

The global population is initiated with 10 individuals and 10 clones are produced for each individual. The threshold dictating the chance of mutation for each parameter is calculated according to (2)

$$a = (1/\beta)\exp(-f*) \qquad (2)$$

where β is a parameter that controls the decay of the inverse exponential function and $f*$ is the fitness of the parent individual normalised within the interval of $[0..1]$. The mutated parameter value is calculated according to (3)

$$C' = c + aN(0, 1) \qquad (3)$$

where c is a parameter value of a parent cell, C' is the mutated parameter value, a is calculated according to (2) and $N(0, 1)$ is a Gaussian random variable with a mean of 0 and standard deviation of 1.

Fitness Function

Within this experiment, we use a multi-objective fitness function to compare CTRNN audio outputs with the EA's target audio sample. Much work exists on reducing timbral profiles to comparable dimensions for the measurement of timbral similarity such as (Carpentier et al. 2010; Abreu, Caetano, and Penha 2016), with spectral features like *Spectral Centroid* and *Spectral Spread* being commonly used metrics. Another method for measuring timbre similarly is by comparing Mel-Frequency Cepstral Coefficients (MFCCs) of two audio samples, such as discussed in (Yee-King 2011; Aucouturier and Pachet 2004). Extracting MFCCs is a single, tested descriptor for musical timbre, therefore we have adopted this measure as one of the objectives in the EA's fitness function. MFCCs are pitch independent therefore we also use the dominant frequency present in the audio spectrum as the other fitness objective. These measures are calculated from a frequency domain description of the audio being analysed, produced by applying a Fast Fourier Transform (FFT) to small windows of the audio (4096 frames with an overlap of 2048 samples) after a Hamming windowing function is applied. As we are dealing with drone-like audio samples that do not change much over time, the amplitudes of the frequency bins produced are averaged to reduce noise in the spectrum, providing a spectral description of the most consistent frequencies in the analysed audio. The dominant frequency of the audio is calculated by identifying the frequency bin with the highest magnitude and the MFCCs are calculated as described in (Yee-King 2011): the FFT magnitudes are passed through a 42 component Mel filter bank spaced in the range of 20 to 22,050Hz, the 42 outputs of which are then transformed using a Discrete Cosine Transform and all 42 coefficients are kept. The similarity error between dominant frequencies is the absolute value of their difference. The similarity error between MFCCs is calculated using a Dynamic Time Warping (DTW) Algorithm (Muda, Begam, and Elamvazuthi 2010) with a Euclidean distance metric. Individuals are ranked according to each fitness objective and these individual ranks are summed to measure the overall fitness of the individual.

Results

For each of the five different EA targets, we ran the algorithm for 100 iterations and as seen in Figure 9, the population commonly converges before 95 iterations. The six fittest individuals within the EA's population are then saved to a database once the algorithm stopping criterion is met. The best of these candidate CTRNNs can be heard and compared with their targets in the online appendix for this paper at `www.plecto.io/ICCC2016appendix`.

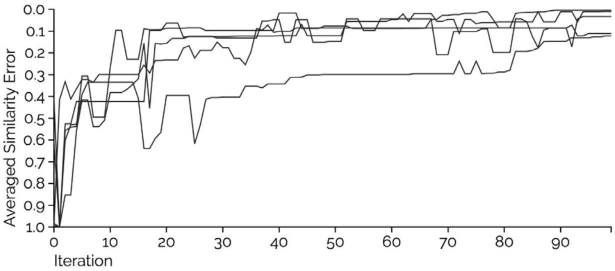

Figure 9: Graph of averaged MFCC and dominant frequency similarity errors (normalised within the interval of $[0..1]$) for each of the five algorithm runs. The best ranked individual of each algorithm iteration is displayed.

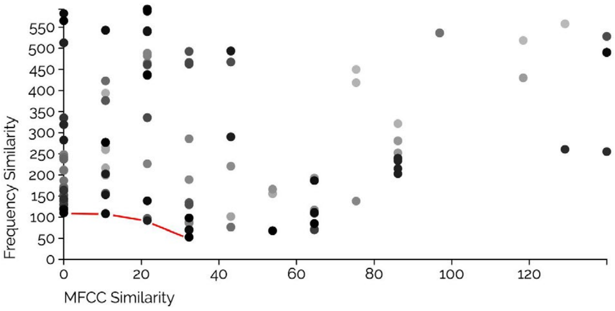

Figure 10: Pareto front of EA (bottom left hand corner) when evolving CTRNNs towards the 'Glass' audio sample. The colour of each individual communicates the EA's iteration in which the individual was produced (darkest are the final iterations).

Among these saved individuals are Pareto optimal candidates, meaning the performance of one of the individual's objectives cannot be improved without adversely affecting another objective (Van Veldhuizen and Lamont 1998). For example, Figure 10 depicts a zoomed in view of the EA's population phenotype space produced when evolving CTRNNs towards the 'Glass' audio sample. In the bottom left hand corner, we can see four Pareto optimal individuals which could all be considered to have an optimal similarity error, forming the EA's Pareto front. There are however often only between one and three individuals from each algorithm run that can be considered Pareto optimal, as there is often a high correlation between the MFCC and the dominant frequency similarity errors when comparing CTRNN outputs with the EA's target audio.

After previewing the EA's outputs, we found that the individuals that sounded most similar to their targets were Pareto optimal solutions with the smallest similarity error between MFCCs. Furthermore, when listening and comparing the various candidate CTRNN outputs to their targets, it is evident that this EA structure is effective when evolving CTRNNs to match simple audio samples but has difficulty replicating more complex sounds, such as multiphonic spectral profiles. For example, when simple audio targets are used such as the 'Glass' or 'Clarinet' audio samples, the resulting CTRNN outputs exhibit strong aural similarities to their target. This contrasts to the CTRNN outputs produced when using more spectrally complex EA targets such as denser, multi-phonic timbres. The attempt to match the 'Cello' audio sample is an example, as spectral aspects of the original recording were lost in the CTRNN outputs even though some of the pitch and general timbral characteristics were present. Matching the 'Complex Synth' audio sample resulted in similar behaviour, with the CTRNN outputs exhibiting only certain aspects of the original audio's spectral structure.

These results highlight that this EA still needs further work. For instance, it may be interesting to adopt a NEAT (NeuroEvolution of Augmented Topologies) method (Stanley and Miikkulainen 2002), meaning that the topology or structure of the CTRNN is manipulated by the EA as well as its parameters as opposed to just the gain, bias, time constant and weight parameters which formulate the genotype for the EA used in this experiment. This approach could provide a means to dynamically increase the complexity of the CTRNN's output by growing the network, removing CTRNN structural limitations when matching complex sounds. Furthermore, additional fitness objectives could be added to the EA's multi-objective fitness function to capture a greater variety of audio characteristics such as information about the change of audio over time, allowing the EA to match more dynamically varied targets such as percussive sounds. Additionally, when dealing with more complex targets, the EA's similarity errors seldom align with aural comparisons of candidate CTRNN outputs and their targets. This suggests that the extraction of MFCCs as a timbral measure either needs to be further refined or supported by other timbral comparison metrics. Nevertheless, these experiments have shown that a simple CTRNN structure can produce a range of timbres and although we have not been able to fully replicate complex sounds, we feel the EA is a good starting point in constructing an effective algorithm for the discovery of CTRNN behaviours.

Future Directions

From our observations within this paper, we believe that CTRNNs could prove valuable as a compositional aid for the discovery of interesting sounds, with their low-level functionality and compact genotype structure affording an explorable algorithm capable of extensive audio diversity. One goal of this research is to achieve a system that enables rapid user exploration of CTRNN audio possibilities. However, although evolving CTRNNs using the *opt-aiNet* algorithm showed promise, the process is time-consuming and will not be feasible in the creation of an engaging system that allows rapid user exploration of audio CTRNNs.

As discussed earlier, another interesting use case for the *opt-aiNet* algorithm could be for NS as we now have tested metrics for audio comparison which can be used to differentiate potential novel CTRNN candidates from existing individuals. This approach removes the need to define an explicit objective for the algorithm, simply rewarding novel finds. Therefore, an interesting design possibility could be to create a large population of small unique CTRNN modules using this method, which can be rapidly assembled by users to build more complex audio structures. This process will take advantage of the DSP effect-like dynamic that CTRNNs possess, with each CTRNN module imparting its various characteristics at each stage of a larger modular system's audio chain.

Additionally, in (Ianigro and Bown 2016), a system is described that evolves CTRNNs using an interactive EA, allowing users to select and evolve CTRNN configurations they find interesting for use within their artistic practices. The paper also identifies difficulties that arise when interactively evolving CTRNN structures, with their vast search spaces creating user fatigue and ineffective discovery of the CTRNN search space. However, if this interactive evolutionary approach is instead used to evolve combinations of higher-level CTRNN modules, a more effective system for the discovery of sound may be achieved. We aim to explore this possibility through the development of *Plecto*, a distributed composition tool that allows users to explore the creative potential of CTRNNs. The progress of this system can monitored by visiting `www.plecto.io`.

Conclusion

Through this research, we conclude that CTRNNs are an effective evolvable synthesis mechanism, affording a compact genotype structure which can be manipulated to achieve vast audio diversity. We have conducted various CTRNN experiments and identified four basic musical dynamics that we believe could be conducive to interesting musical discovery: the introduction of temporal and pitch structures; a strong relationship between CTRNN inputs and outputs; amplitude modulation characteristics; and additive synthesis capabilities. We have also discussed how neuroevolution can be used to manipulate CTRNNs as a means to navigate their creative search space. However, as our current EA design can be slow when discovering ideal candidates to a creative problem, we also discuss future system designs that facilitate flexible, open ended discovery of CTRNN behaviours. Specifically, we discuss a hierarchical system, which at its base level adopts a NS adaption of the *opt-aiNet* EA to discover many small CTRNN modules, each exhibiting unique behaviours that exist within the CTRNN creative search space. At its top level, users can interactively evolve combinations of these CTRNN modules to discover audio complexity that is specific to their creative needs. In our next phase of research, we will implement this system and conduct user studies to further investigate how the low-level dynamics of CTRNNs can be utilised as an effective creative tool that fits into the creative workflows of musicians.

References

Abreu, J.; Caetano, M.; and Penha, R. 2016. Computer-aided musical orchestration using an artificial immune system. In *Evolutionary and Biologically Inspired Music, Sound, Art and Design*. Springer. 1–16.

Aucouturier, J.-J., and Pachet, F. 2004. Tools and architecture for the evaluation of similarity measures: Case study of timbre similarity. In *ISMIR*.

Beer, R. D. 1995. On the dynamics of small continuous-time recurrent neural networks. *Adaptive Behavior* 3(4):469–509.

Bown, O., and Lexer, S. 2006. Continuous-time recurrent neural networks for generative and interactive musical performance. In *Applications of Evolutionary Computing*. Springer. 652–663.

Bown, O. 2009. Ecosystem models for real-time generative music: A methodology and framework. In *International Computer Music Conference (Gary Scavone 16 August 2009 to 21 August 2009)*, 537–540. The International Computer Music Association.

Carpentier, G.; Tardieu, D.; Harvey, J.; Assayag, G.; and Saint-James, E. 2010. Predicting timbre features of instrument sound combinations: Application to automatic orchestration. *Journal of New Music Research* 39(1):47–61.

Dahlstedt, P. 2007. Evolution in creative sound design. In *Evolutionary computer music*. Springer. 79–99.

de Castro, L. N., and Timmis, J. 2002. An artificial immune network for multimodal function optimization. In *Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on*, volume 1, 699–704. IEEE.

de Franca, F. O.; Von Zuben, F. J.; and de Castro, L. N. 2005. An artificial immune network for multimodal function optimization on dynamic environments. In *Proceedings of the 7th annual conference on Genetic and evolutionary computation*, 289–296. ACM.

Eigenfeldt, A.; Bown, O.; Pasquier, P.; and Martin, A. 2013. Towards a taxonomy of musical metacreation: Reflections on the first musical metacreation weekend. In *Proceedings of the Artificial Intelligence and Interactive Digital Entertainment (AIIDE'13) Conference, Boston*.

Eldridge, A. 2005. Neural oscillator synthesis: Generating adaptive signals with a continuous-time neural model.

Floreano, D.; Dürr, P.; and Mattiussi, C. 2008. Neuroevolution: from architectures to learning. *Evolutionary Intelligence* 1(1):47–62.

Hoover, A. K., and Stanley, K. O. 2007. Neat drummer: Interactive evolutionary computation for drum pattern generation. Technical report, Technical Report TR-2007-03.

Husbands, P.; Copley, P.; Eldridge, A.; and Mandelis, J. 2007. An introduction to evolutionary computing for musicians. In *Evolutionary computer music*. Springer. 1–27.

Ianigro, S., and Bown, O. 2016. Plecto: A low-level interactive genetic algorithm for the evolution of audio. In *Evolutionary and Biologically Inspired Music, Sound, Art and Design*. Springer. 63–78.

Jónsson, B. .; Hoover, A. K.; and Risi, S. 2015. Interactively evolving compositional sound synthesis networks. In *Proceedings of the 2015 on Genetic and Evolutionary Computation Conference*, 321–328. ACM.

Lambert, A. J.; Weyde, T.; and Armstrong, N. 2015. Perceiving and predicting expressive rhythm with recurrent neural networks.

Lehman, J., and Stanley, K. O. 2008. Exploiting open-endedness to solve problems through the search for novelty. In *ALIFE*, 329–336.

McCormack, J. 2001. Eden: An evolutionary sonic ecosystem. In *Advances in Artificial Life*. Springer. 133–142.

McCormack, J. 2008. Facing the future: Evolutionary possibilities for human-machine creativity. In *The Art of Artificial Evolution*. Springer. 417–451.

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):247–280.

Muda, L.; Begam, M.; and Elamvazuthi, I. 2010. Voice recognition algorithms using mel frequency cepstral coefficient (mfcc) and dynamic time warping (dtw) techniques. *arXiv preprint arXiv:1003.4083*.

Ohya, K. 1995. A sound synthesis by recurrent neural network. In *Proceedings of the 1995 International Computer Music Conference*, 420–423.

Puckette, M. 2007. The theory and technique of electronic music.

Stanley, K. O., and Miikkulainen, R. 2002. Evolving neural networks through augmenting topologies. *Evolutionary computation* 10(2):99–127.

Timmis, J., and Edmonds, C. 2004. A comment on opt-ainet: An immune network algorithm for optimisation. In *Genetic and Evolutionary Computation–GECCO 2004*, 308–317. Springer.

Tokui, N., and Iba, H. 2000. Music composition with interactive evolutionary computation. In *Proceedings of the 3rd international conference on generative art*, volume 17, 215–226.

Van Veldhuizen, D. A., and Lamont, G. B. 1998. Evolutionary computation and convergence to a pareto front. In *Late breaking papers at the genetic programming 1998 conference*, 221–228. Citeseer.

Yee-King, M., and Roth, M. 2008. Synthbot: An unsupervised software synthesizer programmer. In *Proceedings of the International Computer Music Conference, Ireland*.

Yee-King, M. J. 2011. *Automatic sound synthesizer programming: techniques and applications*. University of Sussex.