Interactively Evolving Compositional Sound Synthesis Networks

Björn Þór Jónsson
IT University of Copenhagen
Copenhagen, Denmark
bangsi@bthj.is

Amy K. Hoover
University of Malta
Msida, Malta MSD 2080
amy.hoover@gmail.com

Sebastian Risi
IT University of Copenhagen
Copenhagen, Denmark
sebr@itu.dk

ABSTRACT

While the success of electronic music often relies on the uniqueness and quality of selected timbres, many musicians struggle with complicated and expensive equipment and techniques to create their desired sounds. Instead, this paper presents a technique for producing novel timbres that are evolved by the musician through interactive evolutionary computation. Each timbre is produced by an *oscillator*, which is represented by a special type of artificial neural network (ANN) called a *compositional pattern producing network* (CPPN). While traditional ANNs compute only sigmoid functions at their hidden nodes, CPPNs can theoretically compute *any* function and can build on those present in traditional synthesizers (e.g. square, sawtooth, triangle, and sine waves functions) to produce completely novel timbres. Evolved with NeuroEvolution of Augmenting Topologies (NEAT), the aim of this paper is to explore the space of potential sounds that can be generated through such *compositional sound synthesis networks* (CSSNs). To study the effect of evolution on subjective appreciation, participants in a listener study ranked evolved timbres by personal preference, resulting in preferences skewed toward the first and last generations. In the long run, the CSSN's ability to generate a variety of different and rich timbre opens up the intriguing possibility of evolving a complete CSSN-encoded synthesizer.

1. INTRODUCTION

While many electronic musicians strive to produce novel timbres for their pieces [9, 18], traditional sound synthesis requires extensive expertise [4]. Inspired by analog oscillators, most digital synthesizers produce periodic combinations of sine, sawtooth, square, and triangle waves [14] that are controlled and manipulated by the user. Because setting synthesizer parameters is complicated, amateur musicians typically default to expert-determined settings to find compelling sounds, thereby limiting the possible novelty of discovered timbres.

GECCO '15, July 11 - 15, 2015, Madrid, Spain

DOI: http://dx.doi.org/10.1145/2739480.2754796

To alleviate expertise required by the user and also permit discovery of novel sounds, the approach in this paper is helps users evolve the waveforms through interactive evolutionary computation (IEC) [30]. Each timbre is represented by a special type of artificial neural network called a *compositional pattern producing network* (CPPN; 28), which unlike traditional artificial neural networks can compute any function at its hidden nodes. CPPNs can not only compute the standard sine, sawtooth, square, and triangle waves present in analog synthesizers, but can also add new functions to theoretically produce novel timbres. The sound producing CPPNs introduced in this paper, called *compositional sound synthesis networks* (CSSNs) are then evolved through NeuroEvolution of Augmenting Topologies (NEAT). Each CSSN can complexify over generations (i.e. add and mutate existing functions and weights), which can result in a more personalized CSSN space. By evolving CSSN topologies, users can find and explore a rich space of potential sounds.

Experiments in this paper are designed to explore the perceived quality of evolved timbres produced by CSSNs. In a listener study participants are asked to rank five related timbres from different generations in the same evolutionary run. Results indicate that preference is dependent on the particular user, but that first and later generations are preferred. Section 2 describes previous approaches to sound synthesis with evolutionary computation while section 3 discuss generating timbres with CSSNs. Sections 4 describe the experiment and listener study. Results are presented in 5 and discussed in sections 6 and 7.

2. BACKGROUND

This section introduces previous approaches in evolving sounds and provides background on CPPNs, NEAT, and wavetable oscillators, the foundations of CSSNs.

2.1 Evolutionary Computation and Sound Generation

Timbre choice often influences musical mood and a listener's perception of quality, and is therefore an important consideration for composers and electronic music producers. While most digital audio workstations provide sound banks from which composers can choose, enhancing these sounds is is often left to the user. Because producing desirable sounds is a complex process requiring the optimization of potentially hundreds of parameters [3], several approaches reproduce existing sounds electronically by evolving and optimizing these parameters. For instance, with a given synthesizer, parameters can be evolved to transform a synthesized waveform into target waveforms [12, 20, 16]. Similarly, other methods evolve these waveforms directly [17, 19, 1, 7]. While these approaches can potentially help musicians to create synthesized sounds for music production, they rely on matching existing sounds rather than exploring the space of entirely new sounds.

Interactive evolutionary computation (IEC) is a popular method for evolving art and music [5, 30]. Instead of relying on fitness functions designed by developers, users make their own aesthetic decisions by rating candidate individuals. With IEC users can evolve music and sounds [11, 3, 15], images [26], and visuals for video games [24, 8, 10, 23]. For example in MutaSynth [3], timbres are encoded as arrays of floats where each gene represents a synthesizer parameter. Users interactively evolve these sounds by both listening to the generated sound and inspecting a visual representation. Parameters are similarly evolved for the CSound synthesis engine in Johnson [15].

In most approaches, users can explore a sound space that is produced by an external synthesizer whose parameters are interactively evolved. Instead, the approach in this paper allows users to interactively evolve wave forms that are themselves generated by a composition of functions rather than strings of parameter settings.

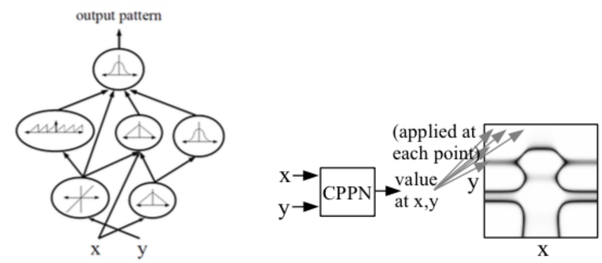
2.2 CPPNs and NEAT

Because timbres are in essence patterns generated from a composition of different functions, in this paper they are represented by compositional pattern producing networks (CPPNs; 28), which are a special type of artificial neural network (ANN) that can compute any activation function at its hidden nodes. In addition to the sigmoid activation functions present in traditional ANNs, CPPNs (shown in Figure 1a) also incorporate sine, linear, and Gaussian activation functions that can bias toward certain types of regularities (e.g. repetition, repetition with variation, bilateral symmetry, etc.). Additionally, CPPNs are typically applied across a broader range of possible inputs like coordinates of a two-dimensional space, therefore allowing them to produce e.g. two-dimensional images (Figure 1b).

In this paper, CPPNs are evolved with *NeuroEvolution of Augmenting Topologies* (NEAT; 27), which was originally designed to solve complex control and decision tasks. NEAT begins evolution with a minimal network topology and complexifies through evolution by adding and mutating connections and nodes to the evolving network. CPPNs evolved in this way have generated picture patterns in Picbreeder [26], musical voices in MaestroGenesis [11], flowers in Petalz [24, 23], and weapon visuals in the video game Galactic Arms Race [8]. Inspired by the quality of results from these approaches, this paper explores the space of one-dimensional sound waves created by CPPNs and searched with NEAT. By representing timbres through hidden node activation functions inherited from analog sound synthesis, CPPNs can in principle reproduce and elaborate on traditional waveforms to produce novel timbres.

2.3 Constructing Wavetables

The patterns produced by CPPNs are the basis to create playable wavetable oscillators in this paper. Wavetables are popular in digital sound synthesis and are in essence an array of N values, where each value represents the oscillator's amplitude at a particular point in the cycle (Figure 2).



(a) CPPN (b) CPPN to Image

Figure 1: Compositional Pattern Producing Networks. (a) CPPNs can use a variety of different functions like sigmoids, Gaussian, sine and many others in contrast to more traditional ANNs with sigmoid activation functions (b) The CPPN example in this figure inputs two arguments x and y that are interpreted as coordinates in two-dimensional space. Applying the CPPN to all the coordinates and drawing them with an ink intensity determined by its output results in a two-dimensional image.

While wavetables are sometimes described as "the sequence of numbers that represent a single cycle of a regular, periodic waveform" [22] others describe it as a technique for stepping through a series of periodic waveforms (in a table) for the duration of each note played [13]. Like analog oscillators, this paper represents wavetables as a single, arbitrary waveform that are playable in a periodic fashion.

The waveforms for building wavetable oscillators can come from a variety of sources. For instance, some are pre-processed and pre-recorded. A recent, notable example of a such a synthesizer is *Animoog*, whose "diverse library of timbres is derived from analog waveforms captured from classic Moog oscillators, both vintage and modern, and run through ...high-end outboard and analog signal processors" [21], thus delivering some of the desired qualities of analog hardware synthesizers to the more affordable digital domain.

Some audio synthesis applications help users create wavetable oscillators by allowing users to draw arbitrary waveforms freely by hand. Others employ techniques from vector graphics to shape the periodic waveforms, such as Bézier curves such as the *DIN Is Noise* synthesizer [25]. These methods challenge the ability to apply visual imagination to the process of shaping interesting timbres. Instead of designing waveforms directly, the new approach presented in this paper lets the user interactively evolve such waveforms, which are encoded by CPPNs. These waveforms are then automatically converted into playable wavetable oscillators.

While there appear to be no previous applications of CPPN-NEAT to create wavetable oscillators (CSSNs), a previous approach evolves soundtrack patterns with CPPN-NEAT [31] and guitar effect pedals [6].

3. APPROACH

The approach presented here focuses on the CSSN representation for sound synthesis and provides an initial investigation into the space of timbres evolved through NEAT. To facilitate the exploration of the space of CSSN timbres, populations of waveforms are presented to the user and evolved through IEC.

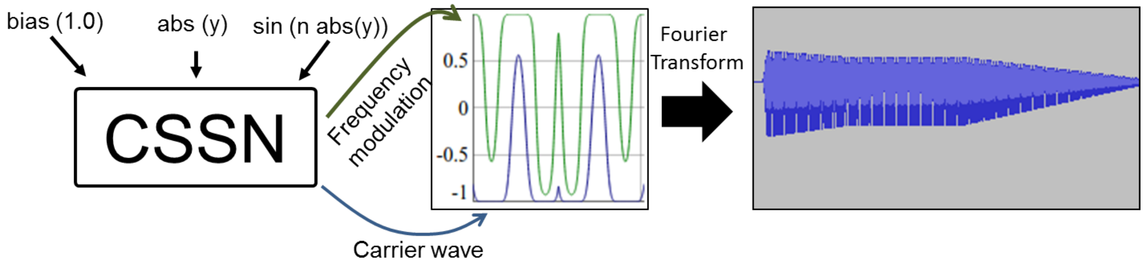


Figure 3: Compositional Sound Synthesis Network. The CSSN has three inputs and two outputs. It receives the absolute values of numbers in a sequence that represents a single cycle of a waveform, and periodic inputs as sine waves produced from multiples of the numbers in that same sequence. The two CSSN outputs allow the assembly of a simple *frequency modulation* synthesizer, in which the modulator output can affect the oscillation frequency of the carrier oscillator output.

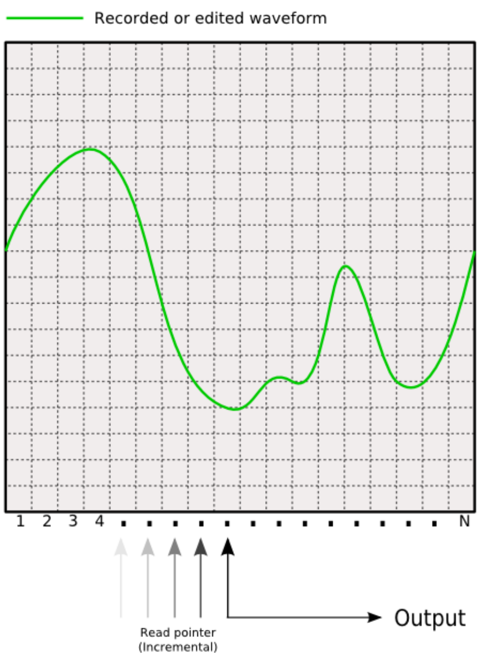


Figure 2: Wavetable Oscillator. This figure shows the basic structure of a wavetable oscilator. Retrieved from http://upload.wikimedia.org/wikibooks/en/4/4b/SSWavetable.png.

The hope of this novel approach is that the generated waveforms reveal interesting timbres that can be serve as unique building blocks for the assembly of novel synthesizers while potentially offering a viable alternative to previous wavetable construction approaches. Additionally, IEC approaches have the unique potential to provide continual improvement in the perceived quality of additional musical voices [11].

The next sections explain the underlying CPPN-based representation in more detail, followed by a description of the interactive framework called *Breedesizer*.

3.1 CPPN Waveform Synthesis

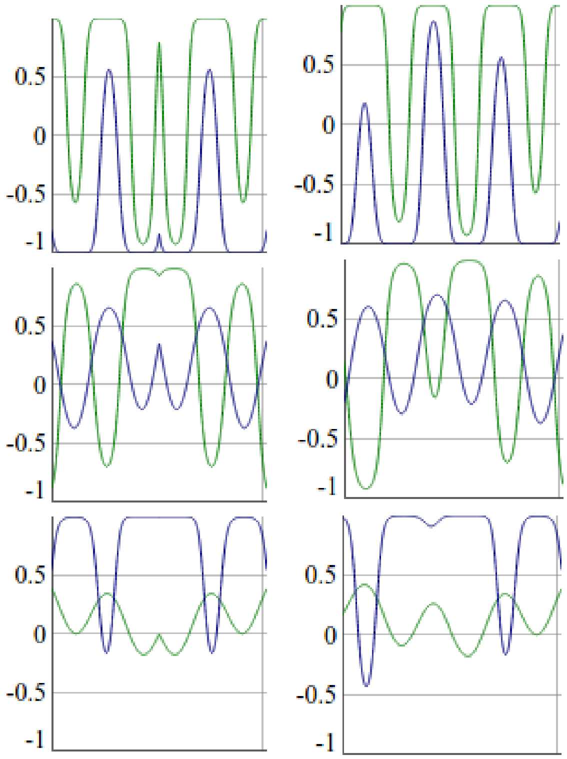
A crucial aspect of generating timbres in this paper is the representation, which is based on CPPNs. CPPNs are particularly suited to produce structured waveforms because of their ability to generate patterns with important regularities such as symmetry and repetition [28, 26]. A fundamental difference between the timbre-encoding CPPNs in this paper and the CPPNs employed in Picbreeder (Figure 1) is that they work with one-dimensional instead of two-dimensional inputs to the networks.

The novel approach presented here is called a *compositional sound synthesis network* (CSSN), an extension of CPPNs to generate playable wavetable oscillators. The current CSSN implements sine, cosine, and inverse tangent activation functions, but further exploration will likely benefit from including a wider variety of functions in the set, such as those representing the classic analog waveforms (e.g. Sawtooth, Square and Triangle).

An overview of the approach is shown in Figure 3. The CSSN is queried with one-dimensional input y that represent each sample of the waveform to be generated. The numbers are in a sequence that represents a single cycle of a periodic waveform between -1 and 1. To ensure continuity between the ends of the waveforms, the absolute value of y is provided as input instead of y directly. To produce repetition in CSSN outputs, $sin(n \times abs(y))$ is fed into an additional input node, with an adjustable integer value n to allow different frequencies of repetition. Removing the application of the absolute value function on the y values passed to the sine wave calculation can result in discontinuous waveforms that may produce unpleasant or interesting timbres.

Each CSSN has two outputs to allow for the assembly of a simple *frequency modulation* (FM) [2] synthesizer; one output modulates the oscillation frequency of the carrier oscillator produced by the second output. The modulated FM synthesis produces a richer timbre than the unmodulated carrier oscillator. Figure 4 shows example waveforms created by different CSSNs with and without symmetry enabled. The underlying encoding is able to elaborate on the repeating sine input, the most fundamental building block of sounds in this paper, and produce personalized oscillators.

To create playable wavetable oscillators from the arbitrary waveforms produced by the CSSN, their signal data is decomposed to its constituent frequencies by a Fourier transform. The results of this process is a table of coefficients in a Fourier series, which represent the partials of a periodic waveform. The FM synthesizer are constructed in con-



(a) Symmetric (b) Non-symmetric

Figure 4: Example Timbres Produced by Compositional Sound Synthesis Networks (CSSNs). This figure shows two waveforms produced by the same CSSN, either with inputs restricted to absolute values (a), or with those restrictions lifted from the periodic input node (b). The green curves represent the carrier wave and the blue curves represent the frequency modulation wave.

nection with each wavetable oscillator under consideration. This is accomplished by forming two periodic waveforms as outputs from the evolved CPPN, where each is used to construct its own wavetable oscillator; one is acting as a modulator of the timbre produced by the other carrier output.

Using the Web Audio API[1], those Fourier coefficients are transformed into a periodic wave, or a wavetable, which can have arbitrary harmonic content. The bigger the table (or sample size), the closer the approximation by the Fourier transform, and the size is adjustable in the user interface with a default of 1024.

More waveforms could easily be added as additional outputs to the CSSN and be used as building blocks of other audio synthesis techniques. The simple FM synthesis approach described here provides a glimpse into the possibilities of using this technique to arrange more complex synthesizers.

3.2 The IEC Application: Breedesizer

Breedesizer is written in HTML and JavaScript and is readily accessible in common web browsers without additional setup overheads of traditional software plug-ins. The program is publicly available at http://bthj.is/breedesizer.

[1]http://webaudioapi.com/

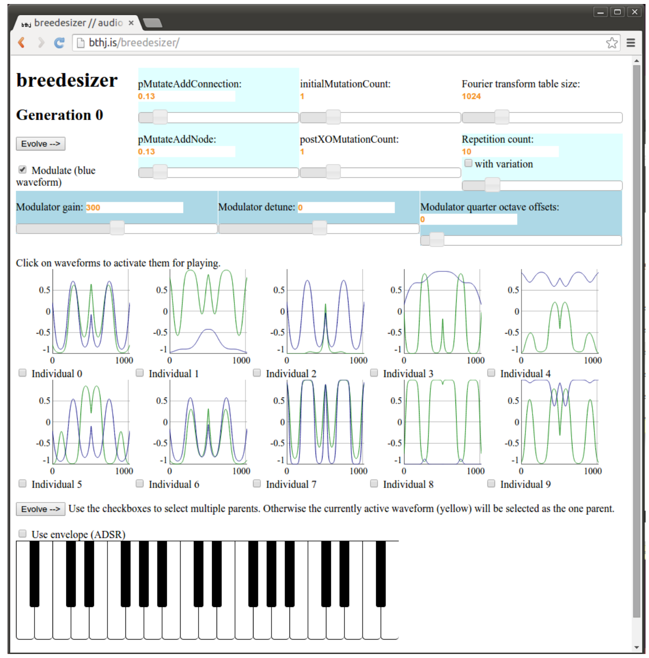


Figure 5: Breedesizer User Interface. Breedesizer allows the user to evolve CSSN-encoded musical timbres through IEC. A variety of different sliders can adjust mutation rates and the frequency of the periodic input to the CSSN or enable/disable FM modulation. Ten individuals are displayed at once to the user. Each waveform can be listened to by clicking on it or playing the virtual keyboard.

Breedesizer is built on the CPPN-NEAT implementation of the Worldwide Infrastructure for Neuroevolution (WIN)[2] solution, which allows them to run in a web browser.

A screenshot of the Breedesizer user interface is shown in Figure 5. The interface presents one population of ten waveforms at a time, which are produced by the CSSN-encoding described in Section 3.1. The ten waveforms at the beginning of each IEC session are created randomly. Future generations are based on mating parents selected from the current population and mutating their offspring. Mutations in the CSSNs modify the connections weights or add nodes and new connections with a low probability.

Each waveform presented can be enabled for playback by clicking on it. An on-screen musical keyboard can be clicked with a pointing device to play different notes with the selected oscillator. The keys of a desktop computer's QWERTY keyboard can also be clicked to play different notes.

In lessen user fatigue, each waveform instantly plays one short note when clicked (C3 at 130.813 Hz), for the duration of one quarter note at 120 BPM, 500 ms. The continuous playing of a selected MIDI song while auditioning the timbres of the different oscillators is still under development.

Mutation probabilities are adjustable in the user interface with two sliders, one for the probability of adding a connection and another for adding a node. A slider is also present for adjusting how many times to mutate each individual.

[2]http://winark.org/, a proposed framework for turning any evolutionary domain into an online interactive platform [29]. The modules are compiled with the Component[3]

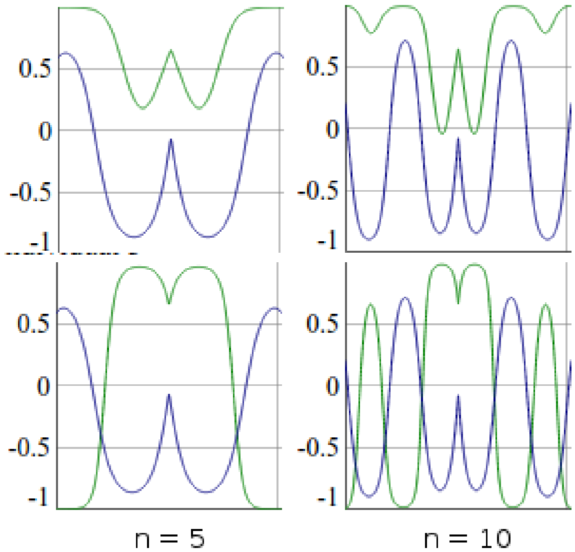


Figure 6: Varying the Periodic CSSN Input. By varying the periodic frequency input into the CSSN different but functionally related waveforms can be created.

Initially, this value is set to five to produce a large variety of different timbres for the first generation, and then reduced to a value of one.

The frequency of the periodic wave input is adjustable with a slider in the user interface, updating the waveform drawings in real time as the slider values are changed. This option allows the user to hear different timbre textures from each CSSN. An example of two waveform variations produced by the same CSSN with different periodic input frequencies is shown in Figure 6.

Three additional sliders are present in the user interface, for adjusting parameters to the FM synthesis configuration. The first sliders change the amplitude of the modulator, controlling its effect on the carrier wave. The second slider enables the detuning of the modulator wave, which is per default set to the same frequency as the carrier. Slightly detuning the modulator frequency produces an audible and often interesting phase change between the oscillators, with a loop between an in-phase and out-of-phase state. The third slider changes the modulator frequencies by multiples of the carrier frequency, where each slider step corresponds to $\frac{carrier\_frequency}{4}$.

The presence of those sliders can be considered as contrary to the primary aim of the presented IEC environment, which is to enable non-experts to discover interesting synthesized sounds; the adjustments of the optional parameters does benefit from familiarity with the fundamentals of frequency modulation synthesis. However, those knowledge-demanding sliders are present in the user interface nonetheless, as the optional addition of FM synthesis (via a checkbox) from the output of each CSSN, is considered an important example of the possible uses for CPPN-NEAT beyond the mere formation of waveforms to base wavetable oscillators on. Additionally, the adjustment of those parameters is important for the appreciation of the expressive power that FM synthesis has to offer.

To provide a familiar synthesizer characteristic and allowing played notes to start and stop less abruptly, an ADSR envelope (Attack-Decay-Sustain-Release) can optionally be enabled while playing the musical keyboard.

4. EXPERIMENT

The main focus of this paper is to explore whether IEC allows an effective exploration of the search space induced by CSSN-encoded timbres. Additionally, the hope is that the quality of timbres should increase as evolution progresses. This section describes the IEC experiment in which the authors evolved timbres interactively and the listener study designed to elucidate its results.

4.1 Experimental Parameters

The CSSN probability of adding a new connection or a new node is set to 0.13 and the probability of changing a weight is 0.7. These parameters were found through prior experimentation. Population size is 10 per generation. The activation functions for each hidden neuron in the CSSNs are chosen from the canonical set of Gaussian, Bipolar sigmoid, sine and linear, each with a 0.25 probability of being added. The ADSR envelope is enabled for the study presented here, resulting in timbres that start and end smoothly by ramping up the volume at the beginning and then bringing it down at the end of the waveform.

4.2 Interactively Evolving Timbres

To study whether the CSSN can create a variety of novel timbres through IEC the authors evolved waveforms for 45 generations through the interface described in Section 3.1. The experiment is designed to provide insight into what type of sounds the CSSN can generate and what are the challenges faced when evolving timbres in this way.

While the particular run of interactive evolution reported in this paper is anecdotal, the hope is that it can nevertheless provide deeper insights into the particular interactions between IEC and CSSN-generated timbres. Most runs exhibited similar dynamics and features in the evolved sounds.

4.3 Listener Study

To explore whether timbres can increase in perceived quality over time, 26 users ranked five clips of Beethoven's "Für Elise" played by related timbres from different generations in the same evolutionary lineage. The survey was administration through SurveyMonkey (www.surveymonkey.com) and advertised through the ISMIR mailing list and colleagues of the authors. The survey reads:

> Below are five clips of Beethoven's Für Elise. Each one is played by a different instrument or timbre. Each clip differs only in the instrument being heard. Please rank your preferences from Most Favorite to Least Favorite.

5. RESULTS

This section presents the results of interactively evolving timbres by the authors and the insights from the listener study. The CSSN-generated sounds, which also serve as the basis for the listener study, are available through the survey on SurveyMonkey: https://www.surveymonkey.com/s/TZX6DJ6. The corresponding generations for the wavefiles in the survey are C=0, B=3, E=20, A=36, and D=45.

Generation	Mean	Std. Dev
0	3.15	1.08
3	2.34	1.26
20	2.07	1.19
36	3.38	1.16
45	3.92	1.09

Table 1: Average Scoring of Survey Participants.

5.1 Evolving Timbres

The initial population of waveforms, presented in the Breedesizer interface, already provides an interesting variety of timbres. The miscellaneity of waveforms initially presented can be attributed to the high number of mutations performed on the individuals created from the population seeds, but the interestingness of the timbres can be attributed to the representational properties of CPPNs.

Figure 7 shows a lineage of timbre evolution of generations 0, 3, 20, 36, and 45. Evolution through the first generations commonly produces more variety of timbres with a similar perceived quality. Individuals that emerge around and after generation four can produce less pleasing sounds than their predecessors and continued evolution can show little or no improvement for the immediately following generations. It may seem like the initial production of variety has stopped, as monotonic timbres of inferior quality are produced over several successive generations.

This period of apparent deterioration, or stagnation at best, gives the user performing the IEC little incentive to continue the evolution, but if he or she perseveres, interesting individuals can emerge again after generation 20. The increase in variety and perceived quality is commonly maintained beyond that point and the timbres can be perceived as fuller, with lesser resemblance of a tin can sound. This increase in richness of the produced sounds can be attributed to subtle nuances that begin to appear in the waveforms, as especially apparent in the blue waveform of Figure 7d.

The pattern of perceived timbre quality during the IEC process is reflected in the user study, discussed next.

5.2 Listener Study Results

The results from the 26-person listener study are shown in Figure 8 and summarized in Table 1. Participants commonly preferred the sounds from the first generation to those from the immediately following generations. However, starting from generation 20 the sounds consistently increase in perceived listener appreciation. Importantly, generation 45 is judged significantly higher than generations 0 to 20 ($p < 0.05$; Student's t-test). Thus the participants felt some overall progression from the initial generation to the last, even though the timbres first decreased in perceived quality.

Interestingly, while similar experiments in evolving musical accompaniments through IEC showed a consistent increase in perceived listener appreciation [11], results from the timbre evolution and listener study are less clear. For instance, first generations in timbre evolution show little or no improvement and the sounds can sometimes first deteriorate in perceived quality before they improve.

The main result is that the CSSN can eventually generate rich and full timbres through IEC that are appreciated by the survey participants. Future work will facilitate and accelerate this process.

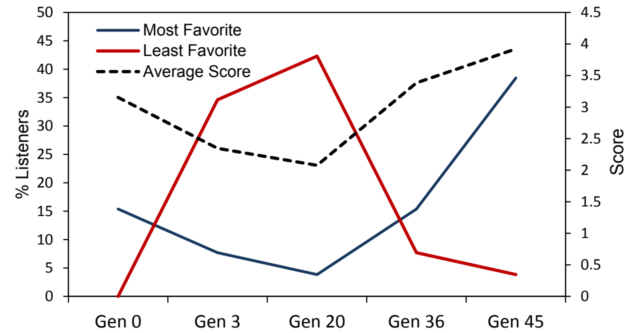


Figure 8: Perceived Quality by Survey Participants. This figure shows the average rating and the percentage of listeners rating the song either as their most or least favorite. The main result is that the participants commonly preferred the sounds from the first generation to those from the immediately following generations, and sounds produced beyond generation 20 are favored still more.

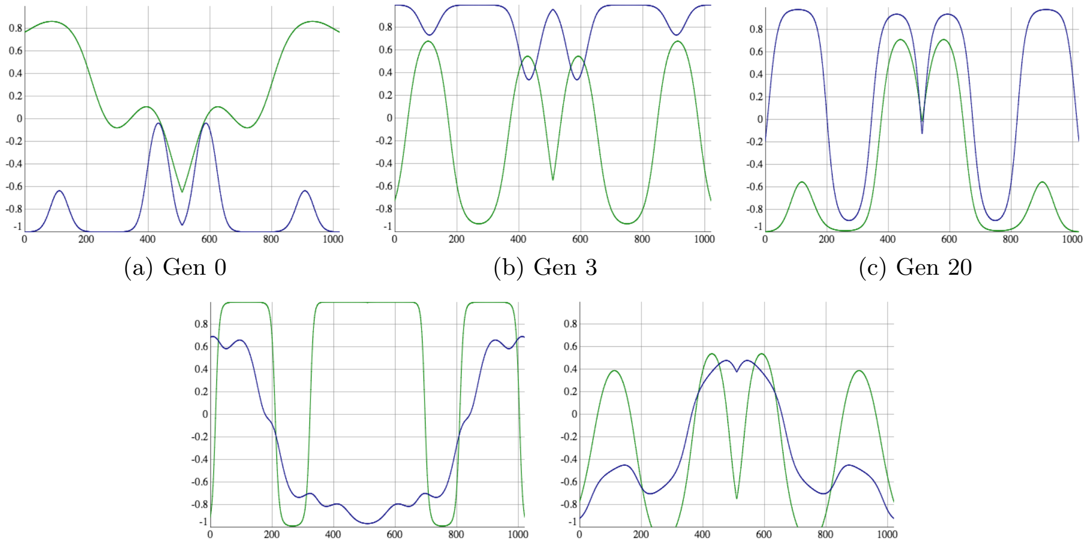
6. DISCUSSION AND FUTURE WORK

The representational properties of CPPNs to form periodic waves enables the presented CSSN approach to produce a variety of timbres. In this way, CSSNs offer a viable alternative to more traditional wavetable construction methods such as free-form drawing or shaping Bézier curves by hand. Future work aims to compare the spaces of timbres that can be produced with these approaches to further explore the structure of the presented search space.

Furthermore, collaborative effort in the evolutionary process, in which users can branch off each others work in a fashion similar to that offered by Picbreeder [26], could help overcome the lack of incentive given by the initial periods of evolutionary stagnating. Fatigue sets soon in, especially when dealing with sound, and a user with "fresh ears" could be more likely to have the extra patience needed to steer the evolution in an interesting direction.

Future work will also focus on leveraging the CSSN's ability to create a wider variety of different sounds. Exploration in the space of single waveforms output from each CSSN can be of limited interest, while using multi-waveforms to construct more complex synthesis configurations may prove to be more interesting and fruitful.The CSSN approach can easily be extended to output more waveforms, which would be different but still functionally related. The additional FM synthesis output in this paper showed how such an addition can make the produced timbres more interesting. Other audio synthesis techniques are based on the combination of many periodic waveforms, both for timbre and control, and they could provide interesting search spaces for exploration. Additionally, CSSNs could output waveforms for each entry in a wavetable, and one additional waveform to define a curve guiding the sweeps through the series of periodic waveforms in this table, similar to wavetable synthesis.

Eventually, whole synthesizer configurations (i.e. where oscillator outputs flow through filters and are affected by control signals) could be evolved. Instead of specifying activation functions, each network node would designate one component of the synthesizer while the weights on connections between nodes would define the strength of interaction between the components.



(a) Gen 0 (b) Gen 3 (c) Gen 20

(d) Gen 36 (e) Gen 45

Figure 7: Evolutionary Timbre Sequence. One lineage of timbre waveforms evolved through successive generations. The blue curves represent the carrier wave, or timbre as it would be heard without FM synthesis, and the green curves represent the frequency modulation wave, which is in effect when FM synthesis is enabled. Only one parent is selected for each generation in this evolution example.

The CSSN presented in this paper enables a variety of different applications, thus opening up a new research direction in sound synthesis.

7. CONCLUSION

This paper presented a novel method for sound synthesis, called compositional sound synthesis networks (CSSNs). The CSSN representation is explored with a new application called Breedesizer that allows users to interactively evolve CSSN-encoded sounds. The benefit of the CSSN representation is that it enables the generation of arbitrary waveforms, thereby offering a viable alternative to traditional methods of sound synthesis. Users can elaborate on CSSN-generated timbres through Breedesizer, allowing an amateur composer to generate novel and rich sounds. Future work will focus on extending the approach presented in this paper to evolve CSSNs that output a multitude of different waveforms and eventually complete synthesizers.

8. ACKNOWLEDGMENTS

The research is supported in part by the FP7 ICT project C2Learn (project no: 318480). Special thanks to Paul A. Szerlip who assisted with the CPPN-NEAT implementation of the Worldwide Infrastructure for Neuroevolution (WIN).

References

[1] Caetano, M. F., Manzolli, J., and Zuben, F. J. V. Interactive control of evolution applied to sound synthesis. In *FLAIRS Conference* (2005), 51–56.

[2] Chowning, J. M. The synthesis of complex audio spectra by means of frequency modulation. *Computer Music Journal* (1977), 46–54.

[3] Dahlstedt, P. Creating and Exploring Huge Parameter Spaces: Interactive Evolution as a Tool for Sound Generation. In *Proceedings of the 2001 International Computer Music Conference* (2001), 235–242.

[4] Dahlstedt, P. Evolution in creative sound design. In *Evolutionary Computer Music*. Springer, 2007, 79–99.

[5] Dawkins, R. *The Blind Watchmaker*. Longman, Essex, U.K., 1986.

[6] DeBoer, S. M., and Stanley, K. O. Systems and methods for inducing effects in a signal, Dec. 3 2013. US Patent 8,600,068.

[7] Garcia, R. A. Growing sound synthesizers using evolutionary methods. In *Proceedings ALMMA 2001: Artificial Life Models for Musical Applications Workshop,(ECAL 2001)* (2001).

[8] Hastings, E. J., Guha, R. K., and Stanley, K. O. Evolving Content in the Galactic Arms Race Video Game. In *Proceedings of the IEEE Symposium on Computational Intelligence and Games (CIG 2009)*, IEEE (2009), 241–248.

[9] Hill, B. Breaking down the breakdown: The use of timbres in contemporary dance music sub-genres. In *Proceedings of the Australasian Computer Music Conference (ACMC 2005)* (2005).

[10] Hoover, A. K., Cachia, W., Liapis, A., and Yannakakis, G. N. Audioinspace: Exploring the creative fusion of generative audio, visuals and gameplay. In *Evolutionary and Biologically Inspired Music, Sound, Art and Design* (2015), 101–112.

[11] Hoover, A. K., Szerlip, P. A., and Stanley, K. O. Functional scaffolding for composing additional musical voices. *Computer Music Journal 38*, 4 (2014), 80–99.

[12] Horner, A., Beauchamp, J., and Haken, L. Machine tongues xvi: Genetic algorithms and their application to fm matching synthesis. *Computer Music Journal* (1993), 17–29.

[13] Howell, S. Origins of wavetable, s&s and vector synthesis. http://www.soundonsound.com/sos/feb06/articles/qa0206_1.htm. Accessed: January 21 2015.

[14] Jenkins, M. *Analog Synthesizers : Understanding, Performing, Buying–From the Legacy of Moog to Software Synthesis*. Taylor and Francis, 2009.

[15] Johnson, C. G. Exploring sound-space with interactive genetic algorithms. *Leonardo 36*, 1 (2003), 51–54.

[16] Lai, Y., Jeng, S.-K., Liu, D.-T., and Liu, Y.-C. Automated optimization of parameters for fm sound synthesis with genetic algorithms. In *International Workshop on Computer Music and Audio Technology* (2006).

[17] Macret, M., and Pasquier, P. Automatic design of sound synthesizers as pure data patches using coevolutionary mixed- typed cartesian genetic programming. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2014)* (2014).

[18] Manning, P. *Electronic and Computer Music*. Oxford University Press, 2004.

[19] Manzolli, J., Jr., A. M., Fornari, J., and Damiani, F. The evolutionary sound synthesis method. In *Proceedings of the Ninth ACM International Conference on Multimedia*, ACM (2001), 585–587.

[20] McDermott, J., Griffith, N. J. L., and O'Neill, M. Evolutionary computation applied to the control of sound synthesis. In *Applications of Evolutionary Computing*, vol. 4974 of *Lecture Notes in Computer Science* (2008).

[21] Moog Music Inc. Animoog (product page). http://www.moogmusic.com/products/apps/animoog-0. Accessed: January 22 2015.

[22] Reid, G. Thor demystified 11: The wavetable oscillator part 1. https://www.propellerheads.se/blog/tutorials/thor-demystified-11-the-wavetable-oscillator-part-1/. Accessed: January 21 2015.

[23] Risi, S., Lehman, J., D'Ambrosio, D., Hall, R., and Stanley, K. Petalz: Search-based procedural content generation for the casual gamer. *Computational Intelligence and AI in Games, IEEE Transactions on PP*, 99 (2015), 1–1.

[24] Risi, S., Lehman, J., D'Ambrosio, D. B., Hall, R., and Stanley, K. O. Combining search-based procedural content generation and social gaming in the petalz video game. In *Proceedings of the Artificial Intelligence and Interactive Digital Entertainment Conference (AIIDE 2012)*, AAAI Press (Menlo Park, CA, 2012).

[25] Sampath, J. DIN Is Noise synthesizer. http://dinisnoise.org/. Accessed: January 21 2015.

[26] Secretan, J., Beato, N., D'Ambrosio, D. B., Campbell, A. R. A., Folsom-Kovarik, J. T., and Stanley, K. O. Picbreeder: A case study in collaborative evolutionary exploration of design space. *Evolutionary Computation 19*, 3 (2011), 373–403.

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

[28] Stanley, K. O. Compositional pattern producing networks: A novel abstraction of development. *Genetic Programming and Evolvable Machines 8*, 2 (2007), 131–162.

[29] Szerlip, P., and Stanley, K. O. Steps toward a modular library for turning any evolutionary domain into an online interactive platform. In *ALIFE 14: The Fourteenth Conference on the Synthesis and Simulation of Living Systems*, vol. 14 (2014), 726–733.

[30] Takagi, H. Interactive evolutionary computation: fusion of the capabilities of ec optimization and human evaluation. *Proceedings of the IEEE 89*, 9 (Sep 2001), 1275–1296.

[31] Ye, J., and Chen, S. SoundBreeder with MultiNEAT. http://web.cs.swarthmore.edu/~meeden/cs81/s14/papers/AndyLucas.pdf. Accessed: January 19 2015.