# Neurythmic: A Rhythm Creation Tool Based on Central Pattern Generators

Dan Bennett
Department of Computer Science
University of Bristol
db15237@bristol.ac.uk

Anne Roudaut
Department of Computer Science
University of Bristol
roudauta@gmail.com

Peter Bennett
Department of Computer Science
University of Bristol
peter.bennett@bristol.ac.uk

## ABSTRACT

We describe the development of Neurythmic: an interactive system for the creation and performance of fluid, expressive musical rhythms using Central Pattern Generators (CPGs). CPGs are neural networks which generate adaptive rhythmic signals. They simulate structures in animals which underly behaviours such as heartbeat, gut peristalsis and complex motor control. [7, 6].

Neurythmic is the first such system to use CPGs for interactive rhythm creation. We discuss how Neurythmic uses the entrainment behaviour of these networks to support the creation of rhythms which avoid the rigidity of grid-based approaches.

As well as discussing the development, design and evaluation of Neurythmic, we discuss relevant properties of the CPG networks used (Matsuoka's Neural Oscillator), and describe methods for their control. Evaluation with expert and professional musicians shows that Neurythmic is a versatile tool, adapting well to a range of quite different musical approaches.

## Author Keywords

NIME, proceedings, Central Pattern Generator, rhythm, sequencing, instrument, design

## CCS Concepts

•**Applied computing** → **Sound and music computing;**
•**Computer systems organization** → *Neural networks;*

## 1. INTRODUCTION

Recent years have seen growth in the popularity of realtime performance of musical rhythms with computer sequencers. Advances in technology and interface design over the decades have allowed musicians to take interactive and improvisatory approaches to electronic music sequencing; analogous to performance with conventional instruments.

There are, however, tradeoffs when adapting interfaces to such realtime, improvisatory approaches to sequencing: generally such interfaces limit expressivity and promote less nuanced, more grid-bound rhythmic articulation, using familiar forms of variation. Musical forms have, of course, developed around these tradeoffs, and thriving musical cultures exist exploiting the mechanical qualities of electronic rhythm sequencing. Nonetheless, there is clearly value in overcoming these limitations and delivering new modes of expression - not to supplant existing modes, but rather to augment and enrich them. Neurythmic aims to address this situation, using rhythm-generating neural networks called Central Pattern Generators (CPG)[6] to support an exploratory approach to realtime rhythm creation without sacrificing rhythmic nuance.

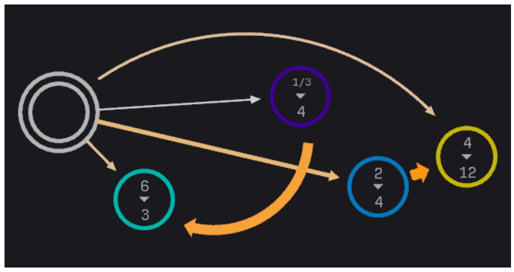


**Figure 1: Neurythmic's main user interface - a simple 5 voice network (described in 4.1)**

Neurythmic is the first system to use CPGs for interactive musical rhythm creation. CPGs are neural networks which generate adaptive rhythmic signals. They model those structures in animals which underly behaviours such as heartbeat and gut peristalsis [7]. It is suggested that CPGs may also be central to complex motor behaviour in vertibrates [6]. In Neurythmic we use the adaptive rhythm-generating properties of CPGs to support rhythmic nuance and variation in improvisatory, realtime music making, offering an alternative to the grid-aligned rhythms characteristic of modern rhythm sequencers.

CPGs allow the creation of rhythms which retain coherence without relying on grids, thanks to their capacity to "entrain". By "entrain" we mean roughly that a CPG node can bring its own oscillation into synchronisation with that of an input signal. In Neurythmic that input signal is the output of another CPG node, with each node forming one part of an overall rhythm.

Rhythmic interest arises since the entrainment behaviour of a node is variable. This behaviour depends on the strength of the signal connection, the frequency of the CPG node, and the frequency of the incoming signal. Varying these parameters results in different qualities of rhythmic synchronisation and phase offset. In musical terms, this means different musical rhythms, repeating with different patterns of variation.

A view of the interface for Neurythmic is shown in fig 1, and a video can be seen at `https://youtu.be/uukpx8qVDoc`. Each circle represents a node in the CPG network and arrows represent connections between nodes. To guide this system toward the desired musical results the user can ad-



*NIME'18,* June 3-6, 2018, Blacksburg, Virginia, USA.

just each node's cycle frequency (top number - as division of the main tempo), as well as adjust the strength of connections between nodes (by moving them closer together or further apart). The user can also add and remove nodes and connections, and change the sounds assigned to each node.

In Neurythmic, nodes generate a note at the beginning of each cycle. Nodes each have a "natural" cycle frequency. This cycle is affected by its connection to other nodes, entraining into a pattern of rhythmic synchronisation with the node's inputs. The quality of this synchronisation is determined by connection weights and the relative frequencies of the input signals.

For simple networks, with low connection weights, user control of rhythm is quite direct, with parameters giving direct control over rhythmic divisions, and beat placement. As complexity is added to the network, user controls determine the output less directly and more generative properties emerge, with the user guiding the system by feel towards the intended outcome.

Below we discuss the development and evaluation of Neurythmic. After reviewing related work, we discuss technical aspects of the CPG networks we used, describing relevant behaviours and the approaches we developed to control them. From there we describe the design of Neurythmic itself, arrived at by a user-centred design approach with expert and professional musicians. We then describe the evaluation of Neurythmic, via a structured creative task and a guided retrospective interview with those same musicians. The results of our evaluation indicate that users were able to quickly adapt to Neurythmic's unusual representation of musical rhythm, describing it as "intuitive", and that they found the musical results appealing, particularly valuing the unquantised quality of the rhythms. Finally we discuss future directions for research using CPGs for musical creativity.

## 2. RELATED WORK

While there has is little previous research on musical applications of CPGs, the general literature on them is extensive, ranging from robotic arm control [14] to the efficient coordination of traffic signals [5]. The type of CPG used by Neurythmic - the half-centre variant of Matsuoka's Neural Oscillator (MNO) [9] was chosen largely due to the volume of literature available to draw upon. In particular results described in [11] were useful in developing control systems, and for identifying zones of stable behaviour.

Excepting a brief paragraph in [8], the only research to date on musical creativity with CPGs seems to be that of Alice Eldridge. Her paper on the subject [4] focuses on the use of MNO at audio frequencies, for the synthesis of complex timbres. Her analysis focuses on the entrainment behaviour of 2 oscillators with a single feed-forward signal connection, and on the frequency spectrum of an oscillator without input. We take a step beyond this, describing patterns of rhythmic behaviour in sub-entrainment zones, developing schemes for the generation of rhythm using MNO, and developing control systems and user interfaces to support musical interaction.

Neurythmic's broad approach has precedent in approaches which generate of rhythm using multiple synchronised loops. Examples of this include polyrhythmic variants of the popular Euclidean Sequencing approach [13], as well as systems such as Gears [2] and the 'polyrhythmic' mode of the popular Octatrack hardware sequencer [1]. Like these systems, Neurythmic indexes musical events to independently cycling loop generators. Such systems tend to ensure coherence of rhythmic results by using low-number integer ratios between the frequencies of their loops - something Neurythmic can replicate since CPGs without input will cycle reliably at a fixed frequency. This approach guarantees repetition of the overall pattern within a small number of bars. While this offers an advance in rhythmic complexity and possibility over a standard linear step-sequencer interface, it is restricted to a particular polyrhythmic class of rhythms, and does not escape the rigidity of the rhythmic grid. Neurythmic moves beyond this by relying on the more flexible entrainment property of CPGs for synchronisation.

## 3. NEURYTHMIC'S CPG

Neurythmic uses a network of CPG units of a type called Matsuoka's Neural Oscillator (MNO). There are various forms of MNO, taking in different topologies and numbers of neurons [9, 10]. Each variant is described by a set of non-linear differential equations, describing the behaviour of a set of neurons, connected by inhibitory synapses, and capable, as a group, of giving rise to oscillatory behaviour. The MNO model used by Neurythmic is the "half centre" variant, modelling just two neurons in a relationship of mutual inhibition (fig.2). This model takes one or more excitatory inputs and a summed inhibitory input. In the equations for the oscillator, presented in (fig.3.) $[x]^+$ stands for the function $max(x, 0)$, and $[x]^-$ for the function $max(-x, 0)$.

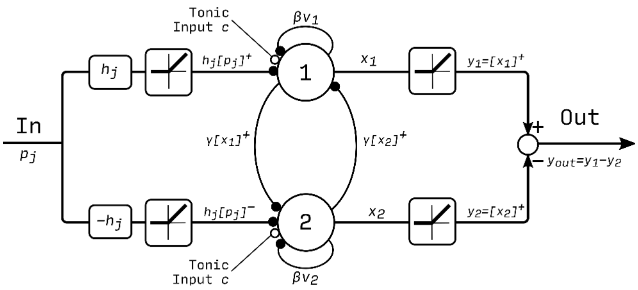


**Figure 2: Two neuron, half-centre Matsuoka Oscillator (redrawn from [14])**

$$\tau_1 \dot{x}_1 = c - x_1 - \beta v_1 - \gamma [x_2]^+ - \sum_j h_j [p_j]^+, \qquad (1)$$

$$\tau_2 \dot{v}_1 = [x_1]^+ - v_1, \qquad (2)$$

$$\tau_1 \dot{x}_2 = c - x_2 - \beta v_2 - \gamma [x_1]^+ - \sum_j h_j [p_j]^-, \qquad (3)$$

$$\tau_2 \dot{v}_2 = [x_2]^+ - v_2, \qquad (4)$$

$$y_{out} = [x_1]^+ - [x_2]^+. \qquad (5)$$

**Figure 3: Equations for the Matsuoka Oscillator**

Subscript numbers indicate the neurons to which parameters belong. $x$ is the membrane potential of the neuron, $v$ represents its state of adaptation. $\tau$ is the time constant, determining the reaction time. $\beta$ is the constant determining adaptation time of both neurons and $\gamma$ the constant determining degree of adaptation of the neurons to one-another. $c$ is the "tonic", excitatory input, determining the amplitude of oscillation. $p$ is the set of $j$ inhibitory external inputs to the neuron, its positive part affecting neuron 1, its negative part neuron 2 $h$ denotes the weights of each of these inputs. Finally, $y_{out}$ is the output of the system

## 3.1 Network Behaviour

Tuning MNO networks towards specific behaviours is a complex business and Matsuoka himself has noted the lack of guidelines for even very basic questions in this area [11]. It is difficult to mathematically describe the rhythmic behaviour of CPG networks once they grow beyond a few nodes and edges - the point at which we have found their behaviour becomes musically interesting - with the result that documented approaches to tuning tend to proceed by heuristics such as swarm optimisation [12, 15].

Accordingly, in the absence of guidance, we selected our equation parameters by finding rough operating ranges via hands on experimentation, then refining by simulation and measurement.

### *3.1.1 Behaviour in sub-entrainment zones*

Once incoming signal rises above a certain threshold, a MNO node will fully entrain to its incoming signal - the node will reliably cycle once for every cycle of the incoming signal. It will do so regardless of the node's own “natural” frequency. This results in quite simple patterns of repetition that, in themselves have little novel musical interest. If the input signal is below this threshold (“sub-entrainment”), then more complex behaviour is demonstrated.

Below entrainment levels, the node's cycle is only partially synchronised to the incoming signal. This results in rhythmic patterns which may repeat only over multiples of the parent signal, or alternatively which may repeat within one parent cycle, but with multiple cycles of the child node, varying in length across the parent cycle. We thus found that even in simple networks (e.g. a root node, a few direct child nodes and feedforward connections) it was not difficult to generate complex and musically interesting results using sub-entrainment linkages.

Specific rhythmic behaviour varies considerably according to signal weight, and to the frequency relationship between input signal and node cycle. Within this variation, however, a broad pattern recurs, with zones of behaviour indexed to increasing weight. This pattern is illustrated by a representative example in fig. 4 and described below.

As shown in that figure, at very low weights, the child node retains in high degree the signature of its natural frequency. The child cycle lengthens with increased input weight, but the pattern resets at the start of the parent cycle, creating a swung, asymmetrical but coherent rhythm. Above this, the pattern acquires greater variation. We can see there is some clustering, continuing the trajectory of phases in the first zone, but overall more complexity is exhibited, and less rhythmic coherence. As weights increase further, another zone appears: rhythms abruptly cohere again. Here however, as weight increases the phase of the middle cycle begins to converge towards the phase relationship the system will assume above the entrainment threshold. Finally we reach that entrainment threshold with one cycle of the child node for each cycle of the incoming signal, at a consistent phase offset from its parent.

Not the entire range of this behaviour will be musically interesting - certainly not in this raw, isolated, two voice, context. However, these results demonstrate the rhythmic diversity possible by variation of just a single parameter in simplest possible network. In larger networks CPGs become highly versatile rhythm generators capable of an incredibly wide range of musical characters.

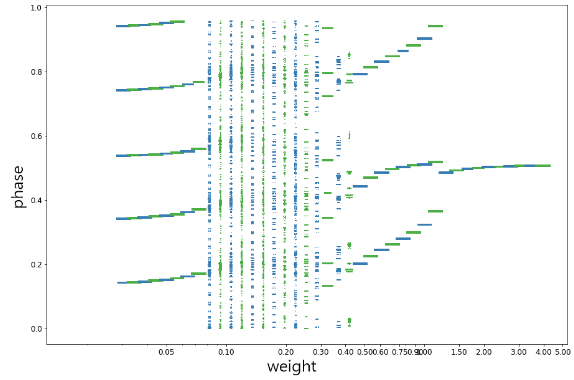


**Figure 4: Phase relationships for 2 node, feed forward network, frequency ratio 1:5, weight increasing in discrete steps, 28 cycles of the parent cycle plotted per weight step. The size of marker indicates frequency of recurrence of phase value**

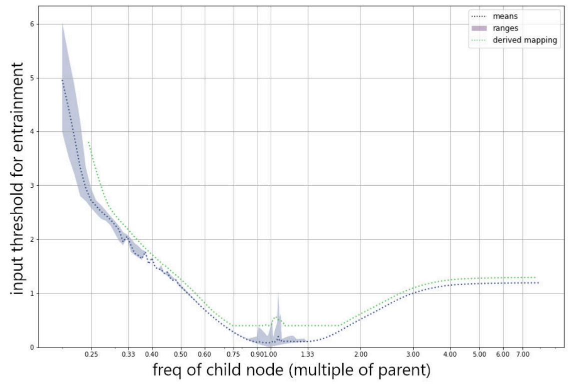


**Figure 5: Entrainment threshold and derived control-mapping curves for input from a single MNO**

### *3.1.2 Input Signal Response Curve*

As noted in [4], the threshold value at which a node entrains to signal input is affected by the frequency ratio between node's cycle and the input signal. When input signal is close to a node's “natural” frequency, very low input weights are required for entrainment. At lower or higher input frequencies, the threshold rises. We found that the curve of this threshold varies with certain equation parameters, being affected by the ratio of $\tau_1 : \tau_2$ and by the values of $\beta$ and $\gamma$.

To help users predict behaviour when setting connection weights we created a control mapping for connection weights based on this curve (see fig 5). We began by measuring the entrainment threshold curve for our chosen equation parameters. Though these measurements come from a single input feed-forward model we found that, with averaging and the enforcement of a minimum cutoff, we could derive a control curve which improved the feel and subjective controllability of the system for larger networks, as confirmed both by our own use and user evaluation.

### *3.1.3 Accurate Control of Frequency*

As described in [11], the time constants, $\tau_1$ and $\tau_2$, can be used to control frequency in isolation, so long as $\tau_1$ remains a constant multiple, $M$, of $\tau_2$. The same paper presents a result approximating the frequency of an unconnected, half-centre MNO, from its equation parameters. While this approximation, proved too inaccurate for our purposes, we noted in testing that the error of the approximation was to a high degree constant for given fixed values of $\beta$, $\gamma$, and $\tau_1/\tau_2$. As such, by including this error as a constant in the equation, we were able to relate oscillator frequency to parameter values with a high degree of accuracy. The improved estimation equation is shown in fig. 6. As well as incorporating the error constant $E$ it includes another $S$ for the sample rate of the simulation and is rearranged to give the required value of $\tau_1$ for a desired oscillation frequency.

$$\tau_1 \approx \frac{ES}{\omega}\sqrt{\frac{(\beta + \beta M - M\gamma}{M\gamma}}$$
$$\text{(where } M = \frac{\tau_2}{\tau_1},\ S = \text{sample rate},\ E = \text{error constant)} \tag{6}$$

**Figure 6: Improved approximation of $\tau_1$ for 2 neuron Matsuoka Oscillator**

This approach requires that a calibration process to be run (simulating n cycles of an isolated oscillator and measuring the error) when $\beta$, $\gamma$, or $\tau_1/\tau_2$ is changed. Since these variables need not be changed under normal operation of Neurythmic, this can be done once at startup.

# 4. NEURYTHMIC DESIGN

## 4.1 Overview

In this subsection we give a walkthrough of the use of the system and an overview of its features. In the next we discuss the process by which we arrived at this design.

A representative screenshot from the main user interface for Neurythmic is illustrated in figs.1 & 7. The system generates musical rhythms based on the behaviour of a network of MNO nodes (described in section 3). Each node in the network acts as a separate musical voice, playing a sound at the beginning of each of its cycles (triggered by the output signal's first peak after positive zero-crossing), the sound specified by the user via a simple percussive synthesizer interface.

Users build up a network, starting with a central "root" node (the white, double circle), whose frequency sets the tempo of the system. Users may add new nodes as children of an existing node - receiving an input connection from the parent - and all nodes but the root node can be deleted.

As well as building their network, users may also manipulate the behaviour of individual nodes. To begin with a node's "natural frequency" can be changed, set as a division of the root tempo. With no input signal this *"natural"* frequency is also the node's *actual* cycle frequency. With one or more input signals the cycle pattern is more complex due to entrainment behaviours (see 3 above).

The user can manipulate certain timing characteristics of a node directly. The rhythmic output of the node can be shifted in time (handled internally by a controllable delay between signal output and note generation), and may also be caused to conform, to a greater or lesser degree, to a user-defined, fixed rhythmic grid. This uses a simple variable quantiser system, which moves the output of the node towards the closest of a set of user-defined grid-points. The degree to which notes are caused to conform to the grid is set by a user control, variable from zero effect, to full grid-alignment. Since this quantisation is applied per-node, the user is able to freely mix conventional, rigidly-constrained electronic rhythms with elements exhibiting some or all of

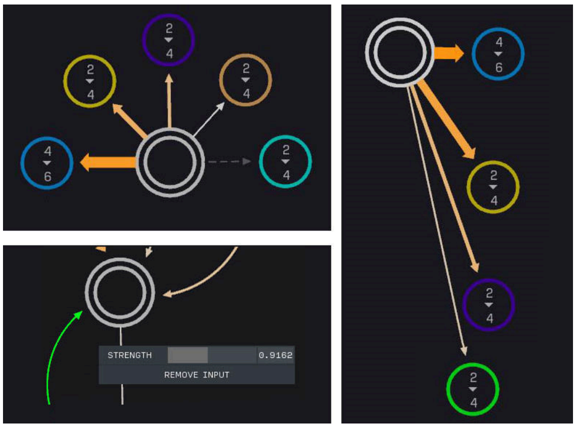


**Figure 7: Connections are scaled both by individual connection controls, and by distance. Top left: connections at same distance, but with different weights. Right: connections with the same weight but scaled by their different distances. Bottom Left: adjusting an individual connection (that highlighted in green).**

the microrhythmic nuance of the underlying CPG network.

The user can also manipulate the connections between nodes. Two mechanisms are provided for this: first, weight can be set directly on individual connections by right clicking the connection to reveal a menu (fig 7, bottom left). Second, weights can be scaled by moving nodes on the screen and thus changing their proximity to one another. Closing the distance between connected nodes increases the weight of their connection, while moving them apart reduces the weight. Since moving a node affects all of its incoming and outgoing connections, it allows for a single-gesture change to affect the system as a whole, or a significant subset of it. Supporting this further the user can group select nodes and move them together. This two-part control scheme - setting weight at connection level and scaling it via node-proximity - is designed to support both broad "gestural" improvisation and fine tuning.

Parameters for the network are fed back to the user via the display (see 1), both numerically via menus, and by more quickly readable means. The weight of a connection is shown by changes in colour, intensity and thickness of the connecting arrow. Weak connections are drawn in thin white lines, and strong connections in thick, orange lines. Connections with zero weight, carrying no signal, display as grey dotted lines (see fig.7 for illustration), allowing users to temporarily disable connections which they may wish to later bring back into play. In addition, the tempo multiplier for a node's "natural frequency" is displayed at the top of the node. The number of grid-divisions per bar that the node's output is constrained to, is displayed at the bottom.

## 4.2 Design Process

We arrived at the design described, by taking the characteristics described in section 3, and developing a system around them via a user-centred design approach. We moved through three stages of prototyping and development, collaborating formally with four musicians. The first stage - informal prototype development - relied on our own experiences with the networks and on feedback from friends and colleagues. This led to the development of a first formal prototype, which we evaluated formally with our four musician collaborators. Feedback from that testing led to the third stage of development. This "final" version was evaluated formally with those same musicians. The results of that evaluation are presented in section 5.

This first full prototype was tested formally with 4 expert and professional musicians. We used a simultaneous think-aloud protocol, introducing the system to the users and inviting them to play freely. We then interviewed the users about their experience, encouraging them to discuss it in the context of their own practice, and to imagine features that might be added to allow them to use the system in their own work. This process allowed us to identify usability issues and missing features, as well as possible use cases for the system which we had not previous identified. These insights fed into the third stage of development: the design presented in this paper and described in the previous subsection.

The first formal prototype was close in certain ways to our final design, having an interactive draggable network representation, control of frequency per node and the same note-per-cycle triggering scheme. However, some significant features and refinements were only added in response to user testing. Most notably we added the quantiser function and an improved interface for moving rhythmic output. We found that the system's capacity to surprise and to generate rhythms with nuance came at the cost that it was easy to accidentally create undesireable results, or for the musicians to become lost. By adding the quantiser we aimed to support exploration within controlled limits. This provided a means to ensure that basic structural elements of the rhythm remained predictable while other elements could retain the underlying network's less grid-bound rhythmic nuance.

Other features added or refined after user evaluation included the use of control curves to map user input to connection weights as described in section 3.1.2. We also added a separate audio mixer, with channels colour coded to the node representations, tweaked various gui elements to reduce number of clicks required for actions. We worked to improve readability of the network by increasing the size of all network elements, and by adding a display to each node showing its frequency division and the number of bar-divisions it is constrained to.

## 4.3 Implementation

Neurythmic was implemented in C++ using the OpenFrameworks multimedia framework for graphics and UI and the Tonic library for audio synthesis. Matsuoka's Oscillator was simulated using the 4th order Runge Kutta method.

# 5. EVALUATION

## 5.1 Methodology

The final version of Neurythmic described in this paper was formally evaluated with 3 of the 4 musicians involved in earlier stages of the process (one musician moved out of the area and was unavailable for testing). All four participants are men in their 30s. One is a professional composer, sound-designer, and academic. One is a double bassist, composer and academic. One is an experimental electronic musician and sound-recordist. The participant who dropped out before evaluation is a singer-songwriter who previously worked as a touring session guitarist. All are respected artists whose work has been performed, published and hosted by respected institutions including BBC national radio, London's Cafe Oto, and the Deutsches Symphonie-Orchester.

We provided participants with an intentionally restricted, percussive sound palette - simple FM synth presets - aiming to focus them on the system's rhythmic behaviour. We allowed each participant a 15 minute session of free play to re-familiarise themselves with the system and ask questions, then asked the participant to perform a structured creative task. Immediately afterwards we audio-recorded an interview about the task and the user's impressions of Neurythmic. Finally the interviews were transcribed and subjected to qualitative analysis using the thematic analysis technique described in [3].

The structured creative task asked participants to imagine they were commissioned by a television producer to create music for a new project. The style and details of the project were of the participants' own imagining. The participant was asked to begin developing material that they could take into a first meeting with the producer. This task was designed to guide participants towards focused creative use of the system in their own style.

Each interview began by asking how the participant felt the task had gone. This prompted quite open, reflective responses. Particularly in early stages of the interview, we worked to sustain this open reflection. We tried to support participants in monologue following their own priorities for the use of such a system, rather than our own preconceptions. To this end we used an interview strategy focused on open questions, the paraphrasing of participant's responses, and requests for rewording and clarification.

Alongside this prompted-monologue approach we kept a set of scripted questions to be turned to in later stages of the interview and when monologue halted and could not be revived. These covered particular evaluation topics, but were phrased so as to avoid leading responses, e.g. "Would you use the system in your own practice", "How would you describe the musical output of the system". These scripted questions were only used if the topics had not already been covered by the participants of their own volition.

## 5.2 Results

The results are presented under themes used in thematic analysis of the interview transcripts

### *5.2.1 Success in performing specified musical task*

All participants reported finding material they were satisfied with, but their levels of satisfaction varied. Two out of the three reported that they were able to work towards an intended goal which fitted their personal musical style, and both stated that they would like to use Neurythmic in their own work. One user in particular reported that he came to the task with a specific idea and achieved it without trouble: "that's what I wanted to do. And it felt very achievable".

### *5.2.2 Musical Output*

Despite being restricted by the limitations of the simple synthesizer provided, participants used the system to pursue quite different sonic ends. These ranged from drone-like approaches to more "fluid" takes on conventional drum-sequencing tropes.

One participant focused on "groove" and compared the experience favourably to sequencing rhythm in a DAW. He described the musical results favourably, calling the rhythm "unlocked" and "detailed, slightly unhinged". He also valued the moments of "discovery" made possible by the system's generative properties, saying it felt as though the network were "coming alive".

One participant used the system in a way we had not anticipated: as what he called a "surface generator". He used a very high tempo and varied the network structure to move fluidly between clear rhythm and a more chaotic timbral surface. He compared this favourably to the way he uses granular synthesis for his sound design work.

The third participants had more trouble and stated that the sounds available from the percussive synthesizer were not suited to his preferred approach.

All users commented positively on the unquantised nature of the rhythms in this final round, and despite the addition of the quantiser system, two of the three participants spent most of their time with the quantiser either turned off, or turned quite low.

### 5.2.3 Ease of use

In several respects Neurythmic diverges substantially from conventional rhythm sequencers. The network representation and significance of connection weight has little parallel in common approaches to rhythm sequencing. Further, while the Neurythmic interface uses familiar time divisions for setting rhythm, these divisions do not function straightforwardly: the node's frequency *influences* behaviour rather than *defines* it, since input signals also influence results.

Nonetheless, all participants indicated that they developed an intuitive understanding of system, even if they did not feel they understood precisely how e.g. the entrainment mechanism worked. One participant expressed this by saying "you don't look beneath the hood - you just see it as a kind of clock sync [...] it's pretty common sense"

All participants used the spatial representation of the network to group their system into distinct sub-ensembles - something we had also found useful in our own use of the system. Two participants addressed this unprovoked, stating that they found the spatial layout useful, giving them an on-screen map of musical qualities.

### 5.2.4 Issues and Improvements

The issues that came up most frequency were the system's limited sound palette, and lack of connectivity to other synthesizers. Both of these features were intentional, aiming to focus evaluation on rhythm generation, rather than on sound design and other aspects of musical creation. Connectivity to other systems via MIDI or OSC is a priority for future development.

Only one participant approached the system with continuous live performance in mind, and in doing so noted aspects of system design which made performance problematic. Neurythmic currently introduces new nodes with timbre, pitch and volume preconfigured - this makes introducing a node during performance disruptive. This highlights addition and removal of nodes as a potential performance mechanism we had not considered - we had imagined building and manipulating networks as separate activities.

All users noted that a multitouch interface would be an improvement, feeling that the ability to manipulate multiple nodes at once would make the system more performable. One user also suggested a VR interface, and the ability to represent and mould networks in 3D.

## 6. FUTURE DIRECTIONS

CPGs show great promise for use in rhythm sequencing systems. They offer a new paradigm for rhythmic interaction between units, and generate rhythms which avoid the stiffness of conventional electronic sequencing. Despite the unfamiliarity of the mechanism it seems that, appropriately presented, CPG based systems can be made accessible to users after only a little exposure.

Neurythmic takes one particular approach to the use of CPGs for musical creativity. It is not difficult to think of ways in which this approach could be developed further. Beyond augmenting the existing system with MIDI, Multitouch and VR interfaces, we might consider driving step sequencer grids from nodes to allow direct specification of multiple notes per cycle.

Nor is it difficult to imagine other musical situations in which CPGs might be applied. CPGs might be used for example to create gestural interaction interfaces which allow users to guide rhythms by physical motion. One approach to this might be to synchronise a node to user input from a force feedback joystick, whose feedback is driven by output from the network.

Given the significance of CPGs in rhythmic behaviour in animals, from heartbeat to complex motor function, the use of CPGs for the creation of musical instruments and interfaces is hugely underexplored. We hope this paper might provoke more research in this direction.

## 7. REFERENCES


[1] Elektron - oktatrack performance sequencer, 2017.

[2] U. Brandstatter, M. Brandstatter, and C. Sommerer. Gears for audio-visual composition. In *Proceedings of the Audio Mostly 2016*, pages 170–177. ACM, 2016.

[3] V. Braun and V. Clarke. Using thematic analysis in psychology. *Qualitative Research in Psychology*, 3(2):77–101, 2006.

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

[5] F. Fang, W. L. Xu, K. Lin, F. Alam, and J. Potgieter. Matsuoka neuronal oscillator for traffic signal control using agent-based simulation. *Procedia Computer Science*, 19:389–395, 2013.

[6] S. L. Hooper. *Central Pattern Generators*. John Wiley & Sons, Ltd, 2001.

[7] P. S. Katz and R. M. Harris-Warrick. Neuromodulation of the crab pyloric central pattern generator by serotonergic/cholinergic proprioceptive afferents. *Journal of Neuroscience*, 10(5):1495–1512, 1990.

[8] H. Kerlleñevich, P. E. Riera, and M. C. Eguia. Santiago-a real-time biological neural network environment for generative music creation. In *Applications of Evolutionary Computation*, pages 344–353. Springer, 2011.

[9] K. Matsuoka. Sustained oscillations generated by mutually inhibiting neurons with adaptation. *Biological cybernetics*, 52(6):367–376, 1985.

[10] K. Matsuoka. Mechanisms of frequency and pattern control in the neural rhythm generators. *Biological cybernetics*, 56(5-6):345–353, 1987.

[11] K. Matsuoka. Analysis of a neural oscillator. *Biological cybernetics*, 104(4-5):297–304, 2011.

[12] N. Shafii, S. Aslani, O. M. Nezami, and S. Shiry. Evolution of biped walking using truncated fourier series and particle swarm optimization. In *Robot Soccer World Cup*, pages 344–354. Springer, 2009.

[13] G. T. Toussaint et al. The euclidean algorithm generates traditional musical rhythms. In *Proceedings of BRIDGES: Mathematical Connections in Art, Music and Science*, pages 47–56, 2005.

[14] M. M. Williamson. Oscillators and crank turning: exploiting natural dynamics with a humanoid robot arm. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 361(1811):2207–2223, 2003.

[15] E. Yazdi and A. T. Haghighat. Evolution of biped walking using neural oscillators controller and harmony search algorithm optimizer. *arXiv preprint arXiv:1006.4553*, 2010.