Eduardo R. Miranda,* Larry Bull,† François Gueguen,* and Ivan S. Uroukov†

*Interdisciplinary Centre for Computer Music Research (ICCMR)
University of Plymouth
Plymouth, PL4 8AA UK
eduardo.miranda@plymouth.ac.uk
†School of Computer Science
University of the West of England
Bristol, BS16 1QY UK
{larry.bull, ivan.uroukov}@uwe.ac.uk

Computer Music Meets Unconventional Computing: Towards Sound Synthesis with In Vitro Neuronal Networks

We are interested in exploring ways in which unconventional modes of computation may provide new directions for future developments in computer music. Research into unconventional computing (Calude, Casti, and Dinneen 1998) addresses computational paradigms other than the standard von Neumann architecture, which has prevailed in computing since the 1940s. This article presents an investigation into the feasibility of synthesizing sounds with hybrid wetware-silicon devices using in vitro neuronal networks.

The field of computer music has evolved in tandem with the field of computer science. Computers have been programmed to play music as early as the beginning of the 1950s, when the CSIR Mk1 computer in Australia was programmed to play popular musical melodies (Doornbusch 2005). The *Illiac Suite for String Quartet*, written in the USA in the late 1950s by composer Lejaren Hiller and mathematician Leonard Isaacson (Hiller and Isaacson 1959), is often cited as the first piece of music involving materials generated by a computer. (The fourth movement was generated using a Markov chain.) Currently, the computer is ubiquitous in many aspects of music, ranging from software for musical composition and production, to systems for distribution of music on the Internet. Therefore, it is likely that future developments in computer science will continue to have an impact on music.

New computational paradigms based on or inspired by the principles of information processing in physical, chemical, and biological systems are promising new venues for the development of new types of computers, and these new ideas may eventually supersede classical paradigms. For instance, it has been reported that reaction-diffusion chemical computers have been capable of performing a number of complex computational tasks, including the design of logical circuits (Steinbock, Toth, and Showalter 1996) and image processing (Kuhnert, Agladze, and Krinsky 1989).

In short, unconventional computing takes computation (or part of it) into the real world, thereby harnessing the immense parallelism and non-algorithmic openness of physical systems. There has been a growing interest in research into the development of hybrid wetware-silicon devices for non-linear computations using cultured brain cells (DeMarse et al. 2001; Potter et al. 2004; Bontorin et al. 2007; Bull and Uroukov 2007; Novellino et al. 2007). The ambition is to harness the intricate dynamics of in vitro neuronal networks to perform computations. Researchers have already mastered techniques to culture tens of thousands of brain cells (neurons and glia) in vitro, on the scale of a few square millimeters, in a mini Petri-like dish with embedded electrodes. Such a system is referred to as a multi-electrode array (MEA) device (see Figure 1). The electrodes can detect neural activity of aggregates of cells and stimulate those cells with electrical pulses (see Figure 2). An MEA can record extra-cellular neural signals fast enough to detect the firing of thousands of nearby neurons as micro-voltage spikes. Thus, the activity of multiple aggregates of neurons can be observed in parallel, and neuronal network phenomena can be studied. Supplying

Computer Music Journal, 33:1, pp. 9–18, Spring 2009
© 2009 Massachusetts Institute of Technology.

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/comj.2009.33.1.9 by guest on 23 March 2021

Figure 1. A typical MEA used to stimulate and record electrical activity of cultured brain cells on the surface of an array of electrodes. (Image printed with permission of Multichannel Systems [www.multichannelsystems.com].)

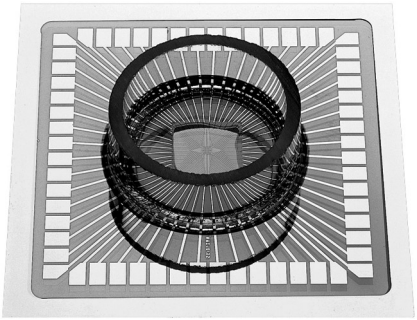
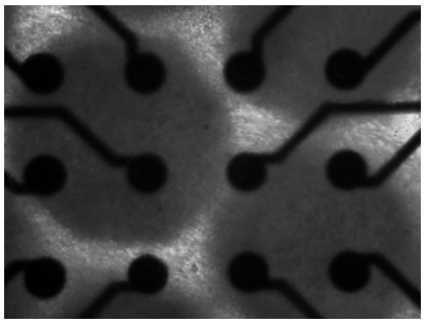


Figure 2. Phase-contrast microscopy showing aggregates of cultured cells on an MEA device.



electrical stimulation through the multiple electrodes typically induces widespread neural activity.

In vitro cultures of brain cells display a strong disposition to form synapses, even more so if subjected to electrical stimulation. It is well known that in vitro cells spontaneously branch out, even if left to themselves without external input other than nutrients in the dish. They establish connections with their neighbors and begin to communicate within days, demonstrating an inherent bias to form networks. Dissociated neurons begin to form connections within a few hours, and an elaborate and spontaneously active living neuronal network is established within a few days. After one month in culture, the development of these networks becomes relatively stable and is characterized by spontaneous bursts of activity (Kamioka et al. 1996). Potter and DeMarse (2001) have developed methods to maintain cultures of brain cells for a number of months, allowing for long-term continuous observations of their behavior.

Research into hybrid wetware-silicon devices with in vitro neuronal networks has made continual progress in recent years. DeMarse et al. (2001) reported the development of a neurally controlled virtual animal—or Animat—using dissociated cortical neurons from rats cultured on an MEA device. Distributed patterns of neural activity (also referred to as *spike trains*) controlled the behavior of the Animat in a virtual, computer-simulated environment. The Animat provided electrical feedback about its movement within its virtual environment to the cells on the MEA device. Changes in the Animat's behavior were studied together with the neural processes that produced those changes in an attempt to understand how information was encoded and processed by the cultured neurons. Potter et al. (2004) described a similar study, but they used physical robots instead of simulations of animals. In this case, different patterns of spike trains triggered specific robotic movements; for example, step forward, turn right, and so forth. The robot was fitted with light sensors and returned brightness information to the MEA as it got closer to the light source. The researchers monitored the activity of the neurons for new signals and emerging neuronal connections. Steve Potter's group also collaborated with artists at SymbioticA in Australia to create an art installation (Potter et al. 2004). They connected an MEA device with cultured neurons in their laboratory in Atlanta to a robotic drawing arm in Perth, Australia. A video camera relayed the drawing process to Atlanta, comparing the image in progress with a photograph of a person. The comparison generated a feedback signal for the cells on the MEA device. The dynamics of in vitro neuronal networks represent a source of very rich temporal behavior, and we are interested in exploiting this behavior to make music.

This article is organized as follows. It begins by introducing the basics of culturing brain cells. Then, it briefly presents the procedures that we have established to stimulate the in vitro neuronal networks and record their behavior, followed by an introduction to a technique that we have developed to sonify their behavior. Next, we report on the initial results from our research into the development of techniques to steer the behavior of the networks. The aim is to exert some form of controllability and repeatability in the system, which is the next

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/comj.2009.33.1.9 by guest on 23 March 2021

step toward placing effective renderings of in vitro neuronal networks technology into controllable sound synthesizers.

Culturing Brain Cells

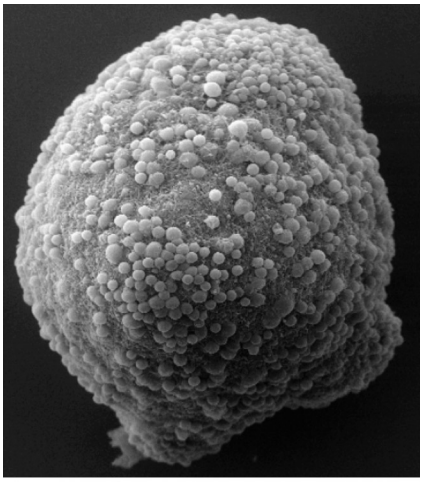
The majority of research into in vitro neuronal networks exploits monolayer (two-dimensional) cultures of rat cortical cells, where the cells grow across the surface of an MEA device (Shahaf and Marom 2001). However, it is now possible to obtain brain cells (neurons and glia) from seven-day-old hen embryos in ovo and maintain them in vitro for relatively long periods of time (typically several months). We have recently described how the maturation of spontaneous spiking behavior in cultures of such cells is typically very similar to that reported in rat cortical cells (Uroukov et al. 2006a). Hence, we have opted for using three-dimensional neuronal cell cultures from hen embryos, rather than monolayer cultures of rat cortical cells. The behavior of 3D cultures is potentially more akin to in vivo networks than are 2D monolayer cultures, and therefore exponentially richer in behavior (Seeds 1971).

Figure 3 shows a typical hen embryo aggregate neuronal culture, also referred to as a *spheroid*. In our experiments, spheroids are grown in culture in an incubator for 21 days. They are then placed into an MEA device in such a way that at least two electrodes make connections into the neuronal network inside the spheroid. The achievement of an appropriate connection is ascertained through recording of the constant spontaneous spiking activity within the spheroid on a given electrode. Next, one electrode is arbitrarily designated as the input by which to apply electrical stimulation and the other as the output from which to record the effects of the stimulation on the spheroid's spiking behavior. (See Uroukov et al. 2006b for more information on the protocols for culturing cells and placement into an MEA device.)

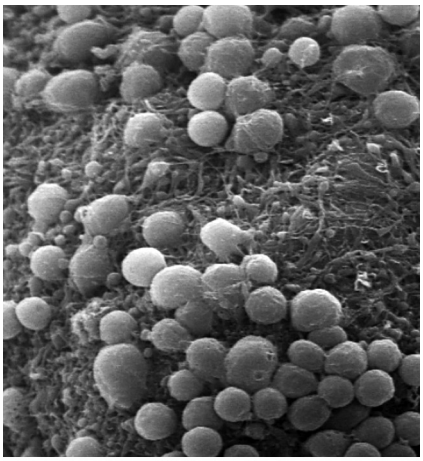
Stimulation and Recording

We used 3D MEA devices supplied by Multi Channel Systems MCS GmbH, Germany. The MEA dish

Figure 3. A typical aggregate neuronal culture from a hen embryo, viewed on a scanning electron microscope: magnified (a) 350 times and (b) 2,000 times.



(a)



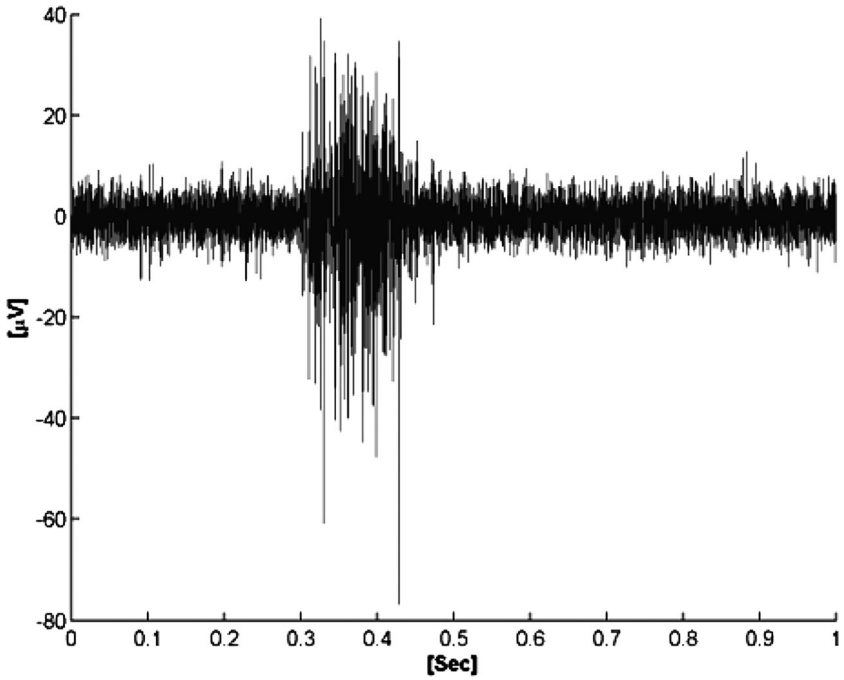
(b)

surface was modified with 10 μg/ml aqueous solution of Polymer Ethylene Imine (PEI) under sterile conditions. The molecular weight of PEI varied between 0.610 and 1.010 according to product specifications. After the modification, the dish was washed twice with demineralized (DEMI) water before the spheroid was placed in it.

Stimulation at the input electrode consisted of a train of biphasic pulses of 300 mV each, occurring once every 300 msec. This induced change in the stream of spikes at the output electrode, which was recorded and saved into a file. The sampling frequency of the output electrode was set to 25 kHz, and the spikes were detected by threshold depending

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/comj.2009.33.1.9 by guest on 23 March 2021

Figure 4. The first second of a stimulation session, showing the spheroid's activity plotted as microvolts (μV) against time. Induced spikes of higher amplitudes took place between 300 msec and 420 msec.



upon the standard deviation and the offset of noise (Uroukov and Bull 2008). Each simulation session lasted for 60 sec, with a 600-sec rest between them. We observed an increase in the spiking behavior after each session, which is an indication that such stimulations seem to foster self-organization within the spheroid. The neuronal networks form in such a way that external stimulation causes significant excitation within the structure.

The resulting neural activity for each session was saved in separate files. Figure 4 plots an excerpt lasting for 1 sec of typical neuronal activity from one of the sessions. Note that the network continually fires spontaneously. The noticeable spikes of higher amplitude indicate concerted increases of firing activity by groups of neurons, which are most probably due to response to input stimuli.

Sonification

We developed and tested a number of sonification methods using different synthesis techniques, including FM, AM, subtractive synthesis, additive synthesis, and granular synthesis (Miranda 2002). Here, we introduce the method that was chosen as the most successful, which combines aspects of granular synthesis and additive synthesis. This choice was informed by three criteria: ability to hear the behavior of the data, simplicity of the synthesizer architecture, and interestingness of the results for use in musical compositions.

The synthesizer is an additive synthesizer with nine sinusoidal oscillators, which required three input values to generate a tone: frequency (*freq*), amplitude (*amp*), and duration (*dur*). The data produced *freq* and *amp* values for the first oscillator only. The values for the other oscillators are relative to the values of the first oscillator (e.g., $freq_{osc2} = freq_{osc1} \times 0.7$, $freq_{osc3} = freq_{osc1} \times 0.6$). The synthesizer was implemented in Csound (Boulanger 2000), and we wrote an application in C++ to generate the respective Csound score files from the data files.

Initially, we synthesized a separate tone for every data point in the recorded neural activity (e.g., every successive *y*-axis value in Figure 4). However, this produced excessively long sounds. To address this problem, a data compression-technique was developed that still preserved the behavior we

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/comj.2009.33.1.9 by guest on 23 March 2021

wanted to sonify, namely, patterns of neural activity and induced spikes. For clarity, we first describe the method whereby we produced a tone for every datum. Then, we present the method using data compression.

In the case of synthesis of one tone per datum, each datum yielded three values for the synthesizer: frequency (*freq*), amplitude (*amp*) and duration (*dur*). The frequency value is calculated in Hz as $freq = (datum \times \varphi) + \alpha$. We set $\alpha = 440$ as an arbitrary reference to 440 Hz; changes to this value produce sonifications in different frequency registers. The variable φ is a scaling factor that accounts for the range of values in the data file. This scaling factor needs to be variable, because the range of μV values produced by the spheroids can vary with different experimental conditions. For the sonifications described in this article, we used $\varphi = 20$.

The synthesizer's amplitude parameter is a number between 0 and 10. The amplitude is calculated as $amp = 2 \times \log_{10}(abs(datum) + 0.01) + 4.5$, which produces a value between 0.5 and 9.5. To avoid negative amplitudes, we take the absolute value of the datum. Then, 0.01 is added to avoid taking a logarithm of 0, which cannot be computed. We later decided to multiply the result of the logarithm by 2 to increase the interval between the amplitudes. Because $\log_{10}(0.01) = -2$, if we multiply this result by 2, then the minimum possible outcome would be equal to –4. We add 4.5 to the result, because our goal is to assign a positive amplitude value to every datum, even a datum of 0 μV.

The duration of the sound is calculated in seconds; it is proportional to the absolute value of the datum, which is divided by a constant *c*: $dur = \frac{abs(datum)}{c} + t$. In the case of the present example, $c = 100$. The higher the value of *c*, the more "granular-like" (Roads 1988; Miranda 1995) the results. We add *t* to the result to account for excessively short or possibly null durations (e.g., $t = 0.05$).

In the case of sonification of compressed data, the compression algorithm begins by creating a set with the value of a datum; this will be the first sample of the data. Then, it feeds in the second sample, the third, and so on. The value of each incoming sample is compared with the value of the first sample to check if they are close to each other according to a given distance threshold Δ. If the difference between them is lower than Δ, then the incoming datum is stored in the set. Otherwise, the values of all data stored in the set are averaged and used to generate a tone. Then, a new set is created whose first value is the value of the datum that prompted the creation of the last tone, and so forth. In this case, the frequency of a tone is calculated as $freq = (\frac{(set\_average - n) \times 900}{x - n}) + 100$, where *n* is the minimum value found in the data file that is being sonified, and *x* is the maximum value. (The values of *n* and *x* do not necessarily need to be the minimum and maximum values in the data file; they can be set arbitrarily, with the condition that $n < x$.) The result is scaled to the range 100–1,000 Hz. The amplitude is calculated as with the case of one tone per datum, as described previously, with the only difference being that the datum is replaced by the set average. The duration is also calculated as described previously, except that we introduce a bandwidth defined by minimum and maximum duration thresholds. If the calculated duration of a tone falls outside the bandwidth, we set the actual duration to the threshold value.

Figure 5 shows the cochleogram of an excerpt of a sonification, where one can clearly observe sonic activity corresponding to induced spiking activity. Figure 6 zooms into the details of the first 500 msec of a sonification, where one can observe variations in amplitude and the granular characteristic of the sound.

Toward Control through Machine Learning

Sonification of data is an interesting practice in its own right, with applications in science (Baier, Hermann, and Stephani 2007) and sonic arts (Kabisch, Kuester, and Penny 2005). However, this project does not stop here. The natural progression is to move on toward sound synthesis, where control and repeatability of some sort are often desired. It is at this stage of the project that the potential of performing nonlinear computations with hybrid wetware-silicon devices begins to be explored.

In this section, we report the first results of our research into the development of protocols to control

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/comj.2009.33.1.9 by guest on 23 March 2021

Figure 5. Cochleogram of an excerpt of a sonification where spikes of higher amplitude can be heard between 2 sec and 4.2 sec.

Figure 6. The first 500 msec of a sonification where one can observe a sequence of five very short tones at different amplitudes. In this case, their durations have been assigned a pre-established value of 100 msec each.

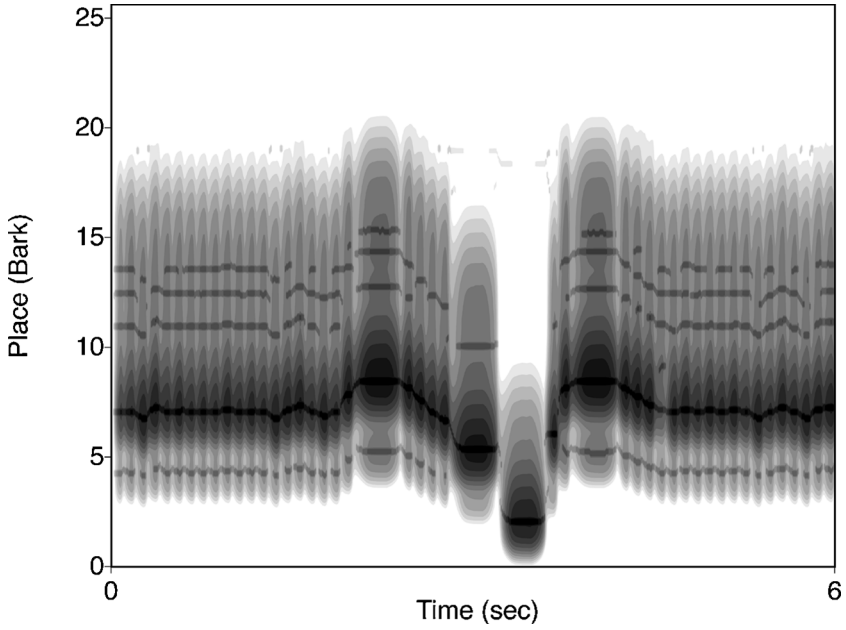


Figure 5

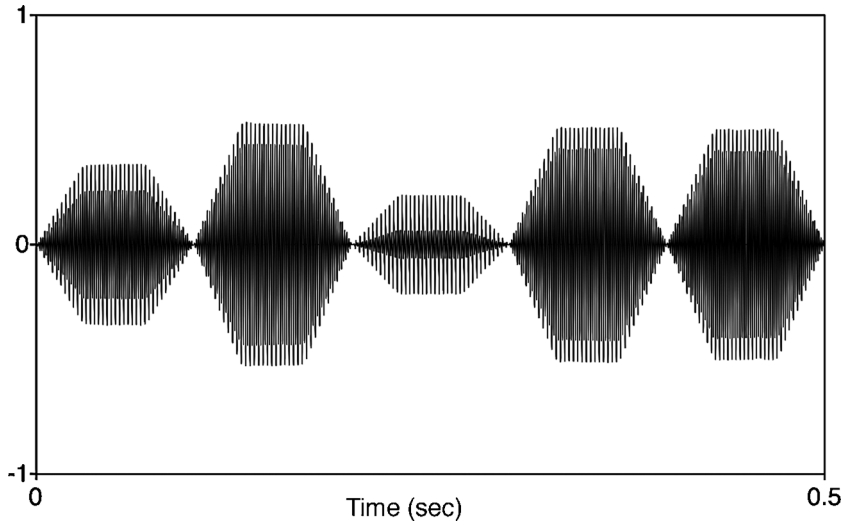


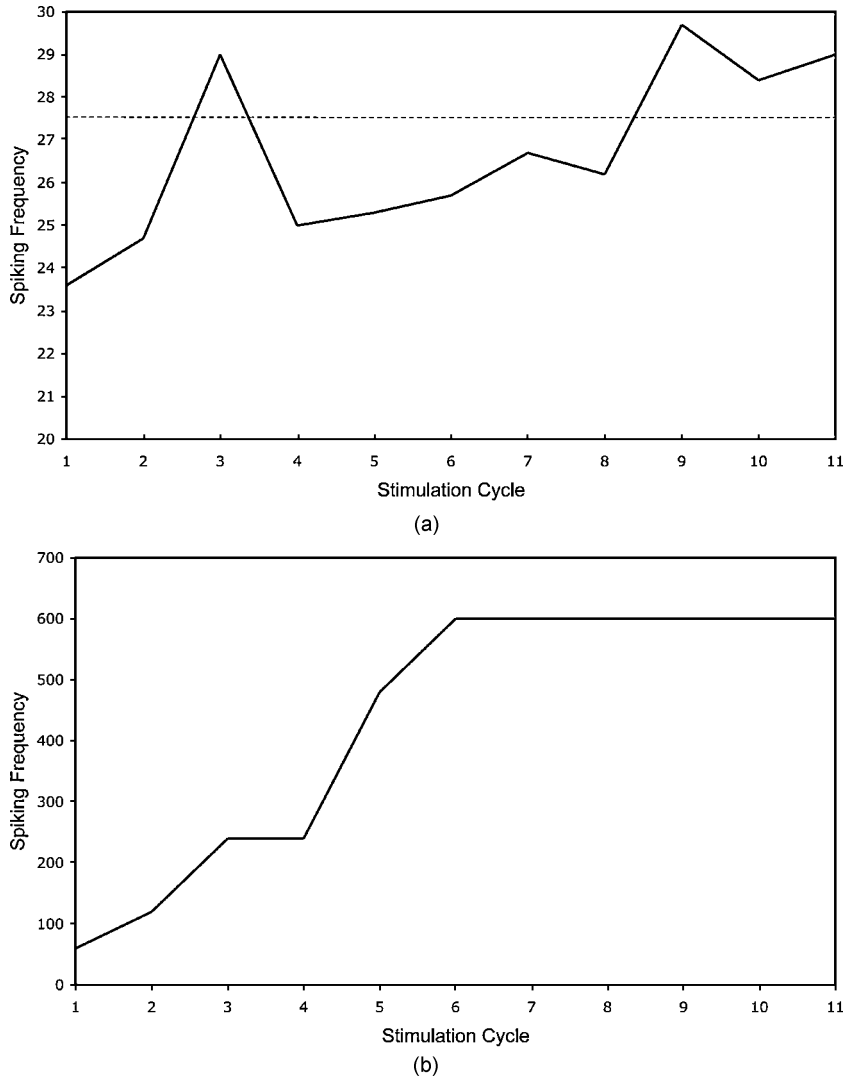
Figure 6

spiking behavior (in vitro wetware computing) through machine learning (in silicon computing). Currently, we are looking into the possibility of producing spike trains with controllable temporal structures.

Shahaf and Marom (2001) have demonstrated stimulus-response learning behavior in cultured rat neurons, where a required response for a given input was obtained from a pre-determined electrode. We have made good progress in obtaining

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/comj.2009.33.1.9 by guest on 23 March 2021

Figure 7. Example learning behavior under XCS control, showing (a) spiking frequency response becoming repeatedly higher than the target indicated by the dashed line and (b) how XCS altered the stimulus application time to achieve this.



similar behavior with cultured neurons from hen embryos. We adopted an evolutionary computing reinforcement learning approach that employs a form of Holland's Learning Classifier System, known as XCS, to create generalizations over a state-action space (Holland 1986; Wilson 1995). Such systems learn production rules of the general form IF <state(s)> AND <action> THEN Reward = *x*.

The average spontaneous spiking frequency of an output electrode of the MEA device is ascertained over a window lasting for 300 sec. The standard deviation of the spikes is detected over the window. The task of the XCS controller is to cause the chosen electrode to reply to the simple stimulus described herein with a spiking frequency of the spontaneous mean plus two standard deviations; a significant increase in typical spiking frequency is required.

The inputs to the XCS are (1) the spiking frequency, averaged over the last three seconds, and (2) the time length of the stimulus. The first number is presented as a fraction of the maximum spiking frequency observed during the 300 sec of spontaneous behavior. The second number is presented as a fraction of the maximum allowed

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/comj.2009.33.1.9 by guest on 23 March 2021

stimulation time of 600 sec. The XCS returns one of three actions: to double, halve, or maintain the current stimulation time. A reward of 500 is given if the spiking frequency increases over the last stimulation period compared to that immediately prior, and a reward of 1,000 is given if the target spiking frequency, or one greater, is achieved. We allow a 300-sec rest period between applications of the stimulus and truncate the maximum duration of stimulation to 600 sec. Thus, 300 sec after the last stimulation period, the XCS controller is given the last recorded spiking frequency of the neuronal network under stimulation as a three-point running average, as well as the amount of time for which the stimulus was applied that caused the response. It then adjusts or maintains the stimulus duration for the next cycle. A stimulation period of 60 sec is used for the initial cycle. Hence, the XCS is presented with an input consisting of two real numbers scaled between 0.0 and 1.0. The condition part of the classifier is encoded as un-ordered pairs of real numbers in the range [0, 1], one pair for each input. A pair is considered to match the corresponding input value if one element of the pair is smaller or equal to the target, and the other is larger or equal. The action of the classifier results in an integer value.

The control problem that we face here is not trivial. Nevertheless, we have succeeded in steering the behavior of the in vitro neuronal networks with the XCS controller in at least a third of our experiments so far. Figure 7 shows an example where the XCS controller was able to cause the required spiking response to the stimulus. As can be seen, and as was typical here, the XCS controller achieves this by increasing the duration for which the stimulation is applied. However, there were cases where no significant change in spiking appeared to have occurred regardless of how the XCS adjusted the stimulation. In other cases, the average spiking response decreased during the experiment regardless of stimulation.

Concluding Discussion

Sound synthesis with in vitro neural networks still is in its infancy, but so is the field of unconventional computing with in vitro neural networks. This article reported an initial investigation into the feasibility of such an approach. To the best of our knowledge, this is the first time that this modality of unconventional computation has been investigated in the context of computer music.

We introduced a technique to sonify the behavior of in vitro neuronal networks, which proved the concept: it works (in the sense that we can hear the behavior of the data), and it is simple. We acknowledge that one of our three criteria for evaluating potential sonification techniques (see the first paragraph of the section, "Sonification") is rather subjective and biased by the aesthetic judgment of the authors, one of whom is a composer. Nevertheless, we have decided to adopt this subjective criterion, because this technology is being developed primarily for making music, and we believe that composers should ultimately decide upon the nature of the technology and sounds with which they make music.

An important property of a sound-synthesis technique is its ability to produce different types of sounds with a certain degree of predictable control. Controlling behavior and understanding how information is coded and processed by the cultured neurons are two "holy grails" of research into in vitro neuronal networks. We introduced an evolutionary computing reinforcement-learning approach to tackle the problem of control and demonstrated how stimulus-response learning behavior in cultured neurons from hen embryos can be achieved.

Our ability to steer the behavior of in vitro neuronal networks in a third of cases is very encouraging, because it demonstrates that the problem is tractable. From a control engineering perspective, it would be highly desirable to be able to fully control the in vitro neural networks, especially if one intends to exploit this technology to build new digital musical instruments (Miranda and Wanderley 2006). However, our current inability to exert full control or predict the behavior of these networks does not prohibit musical application of this technology. On the contrary, the technology still allows a number of attractive possibilities for musicians interested in ideas such as those of John Cage (1961), who welcomed indeterminism in the creative process. Moreover, a less conservative approach to

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/comj.2009.33.1.9 by guest on 23 March 2021

music technology is definitely needed here to take advantage of one of the major strengths of unconventional computing, namely, nonlinear computation.

Acknowledgments

This work was partially supported by EPSRC grant GR/T11029. Mr. Gueguen, from the Department of Informatics of Université de Bretagne Occidentale, France, thanks ICCMR for the opportunity to work on this project during his placement at the University of Plymouth.

References

Baier, G., T. Hermann, and U. Stephani. 2007. "Event-Based Sonification of EEG Rhythms in Real Time." *Clinical Neurophysiology* 118(6):1377–1386.

Bontorin, G., et al. 2007. "A Real-Time Closed-Loop Setup for Hybrid Neural Networks." *Proceedings of 29th Annual Conference of the IEEE Engineering in Medicine and Biology Society*. Piscataway, New Jersey: Institute for Electrical and Electronics Engineers, pp. 3004–3007.

Boulanger, R. C., ed. 2000. *The Csound Book*. Cambridge, Massachusetts: MIT Press.

Bull, L., and I. S. Uroukov. 2007. "Initial Results from the use of Learning Classifier Systems to Control In Vitro Neuronal Networks." *Proceedings of the 9th Genetic and Evolutionary Computation Conference – GECCO'07*. New York: Association for Computing Machinery, pp. 369–376.

Cage, J. 1961. *Silence: Lectures and Writings.* Middletown, Connecticut: Wesleyan University Press.

Calude, C., J. L. Casti, and M. J. Dinneen, eds. 1998. *Unconventional Models of Computation: Proceedings of the 1st International Conference – UMC'98*. New York: Springer.

DeMarse, T., et al. 2001. "The Neurally Controlled Animat: Biological Brains Acting with Simulated Bodies." *Autonomous Robots* 11(3):305–310.

Doornbusch, P. 2005. *The Music of the CSIRAC: Australia's First Computer Music*. Victoria, Australia: Common Ground.

Hiller, L. A., and L. M. Isaacson. 1959. *Experimental Music*. New York: McGraw-Hill.

Holland, J. H. 1986. "Escaping Brittleness: The Possibilities of General-Purpose Learning Algorithms Applied to Parallel Rule-Based Systems." In R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, eds. *Machine Learning: An Artificial Intelligence Approach*. San Francisco, California: Morgan Kaufmann, pp. 48–78.

Kabisch, E., F. Kuester, and S. Penny. 2005. "Sonic Panoramas: Experiments with Interactive Landscape Image Sonification." *Proceedings of the 2005 International Conference on Augmented Tele-Existence*. New York: Association for Computing Machinery, pp. 156–163.

Kamioka, H., et al. 1996. "Spontaneous Periodic Synchronized Bursting During Formation of Mature Patterns of Connections in Cortical Cultures." *Neuroscience Letters* 206(1–2):109–112.

Kuhnert, L., K. I. Agladze, and V. I. Krinsky. 1989. "Image Processing Using Light Sensitive Chemical Waves." *Nature* 337:244–247.

Miranda, E. R. 2002. *Computer Sound Design: Synthesis Techniques and Programming*. Oxford: Elsevier/Focal Press.

Miranda, E. R. 1995. "Granular Synthesis of Sounds by means of a Cellular Automaton." *Leonardo* 28(4):297–300.

Miranda, E. R., and M. M. Wanderley. 2006. *New Digital Musical Instruments: Control and Interaction Beyond the Keyboard*. Middleton, Wisconsin: A-R Editions.

Novellino, A., et al. 2007. "Connecting Neurons to a Mobile Robot: An In Vitro Bidirectional Neural Interface." *Computational Intelligence and Neuroscience* Vol. 2007: Article ID 12725. Available online at www.hindawi.com/journals/cin/.

Potter, A. M., et al. 2004. "Hybrots: Hybrids of Living Neurons and Robots for Studying Neural Computation." *Proceedings of Brain Inspired Cognitive Systems*. Stirling, UK. Available online at www.cs.stir.ac.uk/~lss/BICS2004/CD/CDBISprog.html.

Potter, S. M., and T. B. DeMarse. 2001. "A New Approach to Neural Cell Culture for Long-Term Studies." *Journal of Neuroscience Methods* 111:49–62.

Roads, C. 1988. "Introduction to Granular Synthesis." *Computer Music Journal* 12(2):11–13.

Seeds, N. W. 1971. "Biochemical Differentiation in Reaggregating Brain Cell Culture." *Proceedings of the National Academy of Sciences* 68(8):1858–1861.

Shahaf, G., and S. Marom. 2001. "Learning in Networks of Cortical Neurons." *Journal of Neuroscience* 21:8782–8788.

Steinbock, O., A. Toth, and K. Showalter. 1996. "Chemical Wave Logic Gates." *Journal of Physical Chemistry* 100:18970–18975.

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/comj.2009.33.1.9 by guest on 23 March 2021

Uroukov, I., and L. Bull. 2008. "On the Effect of Long-Term Electrical Stimulation on 3-Dimensional Cell Cultures: Hen Embryo Brain Spheroids." *Medical Devices: Evidence and Research* 1:1–12.

Uroukov, I., et al. 2006a. "MEA Recordings of the Spontaneous Behaviour of Hen Embryo Brain Spheroids."*Proceedings of the 5th International Meeting on Substrate-Integrated Microelectrode Arrays*. Reutlingen, Germany: BIOPRO, pp. 232–234.

Uroukov, I., et al. 2006b. "Electrophysiological Measurements in 3-Dimensional In Vivo–Mimetic Organotypic Cell Cultures: Preliminary Studies with Hen Embryo Brain Spheroids." *Neuroscience Letters* 404:33–38.

Wilson, S. W. 1995. "Classifier Fitness Based on Accuracy." *Evolutionary Computing* 3:149–175.

Downloaded from http://www.mitpressjournals.org/doi/pdf/10.1162/comj.2009.33.1.9 by guest on 23 March 2021