A Nonlinear Approach to Generate Creative Data using Physarum polycephalum-based Memristors

S Venkatesh

E Braund

E Miranda *School of Art, Design and Architecture*

A Nonlinear Approach to Generate Creative Data using *Physarum polycephalum*-based Memristors

Satvik Venkatesh *[1], Edward Braund †[1], and Eduardo Reck Miranda ‡[1]

[1]Interdisciplinary Centre for Computer Music Research (ICCMR), Plymouth University, Plymouth, UK, PL48AA.

Abstract

This chapter presents a generic biocomputing system developed to generate data for creative pieces such as music and art. It harnesses the nonlinear behaviour of *Physarum polycephalum*-based memristors, which is in contrast to stochastic processes that are often explored in creative systems. Within this chapter, we explain the advantages of using biomemristors for such applications and discuss a compact and portable biocomputer called *PhyBox*. It harnesses biomemristors as processing units and highlights the need for non-digital ways of representing, processing, and storing data. The system generates new creative pieces that are inspired by seed data that is input by the user. It allows the user to determine the *degree of similarity* between the output and the pre-existing creative data by controlling the *nonlinearity* of *Physarum polycephalum*-based memristors. The mapping procedure considers the resistor's behaviour to be ideal and reproduces the pre-existing data if a resistor is connected instead of a memristor. Our system is generic because it does not depend on the type of creative piece that is being processed. The chapter presents results from testing the system under different scenarios and provides ways for creative practitioners to adopt Unconventional Computing technologies in their works.

4.1 Introduction

Stochastic models are common in systems that generate creative works like music, digital art and text (Cope, 2005; Roads, 1996; Xenakis, 1992; Simonton,

*Email: satvik.venkatesh@students.plymouth.ac.uk
†Email: edward.braund@plymouth.ac.uk
‡Email: eduardo.miranda@plymouth.ac.uk

2003; Conroy and O'leary, 2001). A stochastic process aims to induce randomness, that is the output of the system cannot be predicted (Ross, 1996). One of the earliest stochastic models is the Bernoulli process, which was implemented by flipping a coin with *heads* and *tails* opposing to two different values. Hedges (1978) describes music that was composed by the rolling of dice, in which each number represented a musical note. The use of Markov chains is a well-established methodology adopted by many creative systems (Pinkerton, 1956; Brooks et al., 1957). A Markov chain presents a conditional probability distribution of future states based on the present state of the system (Ross, 1996). The probability distributions in Markov chains are generally calculated based on certain rules or pre-existing data. This allows the system to generate creative pieces that are *inspired* by some content. Cope (1996) built a system to generate music in the style of composers like Bach and Mozart. Programmes have used artificial neural networks to develop several subroutines for music composition (Miranda, 2001). Evolutionary computation and genetic algorithms also use stochastic processes to create generations of creative pieces. This chapter proposes the use of *nonlinear* behaviour of biomemristors as opposed to stochastic processes for creative applications.

Unconventional Computing (UC) aims to develop new computer architectures for data processing and storage by adopting physical, biological and chemical systems (Adamatzky, 2010). Conventional creative applications are primarily based on discrete data. The fundamental unit of information is the binary digit, that is either 0 or 1. This chapter presents a hardware-wetware architecture that uses two different fundamental units for information. The pre-existing aspect of the system is implemented by using a conventional computer, but the *nonlinearity* is realised by a *Physarum polycephalum*-based biomemristor, which replaces the stochastic element. Alongside the advancement of UC technologies, new approaches towards data representations are bound to be realised. The relationship between pre-existing creative data and the generated creative product needs to visualised in non-digital ways. The chapter establishes a link between the pre-existing data and output by using metrics that allow the user to control the *nonlinearity* of *Physarum polycephalum*-based memristors.

In this context, the term nonlinear behaviour is derived from memristors being nonlinear electronic components. As shown in figure 4.1, when a voltage change is applied across a memristor, there is a spike in current. *Memristance* is formulated by equation 4.1.

$$M = \frac{d\phi}{dq} \tag{4.1}$$

where M is the memristance, ϕ is the magnetic flux and q is the charge. The current flowing through the memristor is not constant. For a positive change in voltage, the magnitude of current declines over time because of an increase in memristance. Memristance depends on the physical history of the memristor, that is the amount of charge that has flowed through it (Johnsen, 2012). Chua (1971) hypothesised the memristor as a fundamental circuit element because its behaviour cannot be replicated by a combination of the other three circuit elements— resistor, capacitor and inductor. Memristance shares the same unit as resistance and if M is constant, it is identical to resistance (Tour and He, 2008). Just how probability distributions are used to effect the future states in Markov chains, this chapter uses memristors as processing units. It aims to harness the spiking property of *Physarum polycephalum*-based memristors to create a bridge between the pre-existing creative data and generated output.

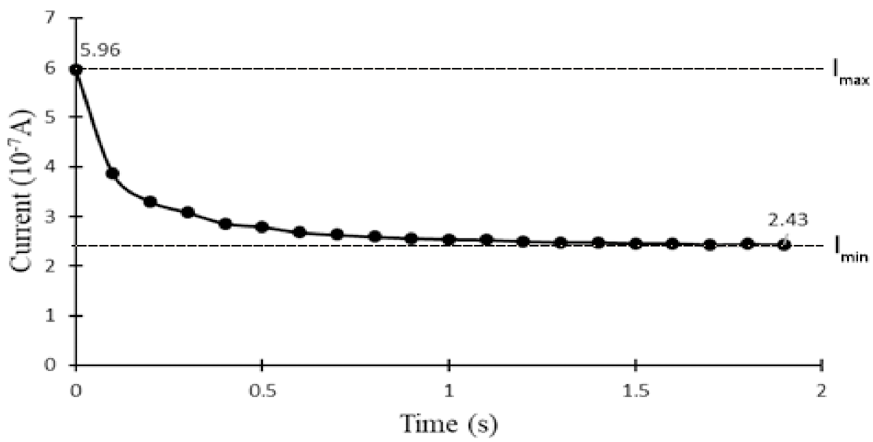


Figure 4.1: Spiking response of *Physarum polycephalum* for a positive change in voltage. The relevance of I_{max} and I_{min} is explained later in the chapter.

Physarum polycephalum is plasmodial slime mould which is easy and inexpensive to obtain. Its ability to act as a biomemristor was demonstrated in Gale et al. (2015). Initially, it was grown in small Petri dishes with electrodes retrofitted inside them. Braund (2017) conducted several experiments to harness it as a memristive component. Compact receptacles to contain the organism were designed and fabricated by using a 3-D printer (Braund and Miranda, 2017*b*). They conveniently incorporated the organism into electronic circuits. In this chapter, each biomemristor is implemented as a processing unit and it realises input in the form of voltages. The current flowing in the circuit is a function of the *Physarum polycephalum*'s memristance. Therefore, the magnitude of current is considered to be the organism's output. Each creative system handles data by breaking it down into attributes. For instance, an image can be described by red, green and blue (RGB) values for each pixel. Therefore, digital art can be generated by assigning three memristors to each of the respective colours.

The approach of employing nonlinearity to generate creative data can be evaluated by comparing the behaviours of linear and nonlinear electronic components. Figure 4.2 illustrates an overview of the system. The processing unit is connected serially between the input and output. Hence, this chapter addresses the comparison between a memristor and resistor. Creative data needs to be translated into voltages to serve as input for the memristor; current needs to be translated back to creative data to generate the final output. A mapping procedure to perform these translations is demonstrated in this chapter. The resistor, being a linear component is considered to be *ideal*. Therefore, when the memristor is replaced by a resistor the system should produce the pre-existing creative data itself. This demonstrates that the generation of creative data is purely because of the nonlinearity of the biomemristor and no other factor.

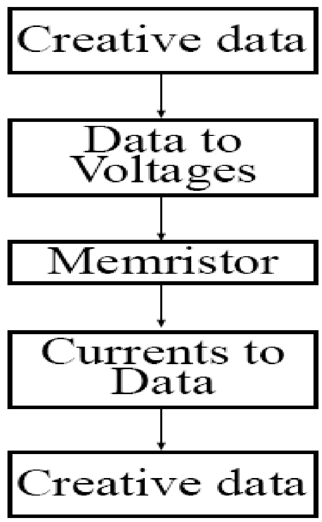


Figure 4.2: Overview of functions carried out by the system.

The system encompasses two user-adjustable parameters— Dwell time and measurement offset (Braund and Miranda, 2017*a*). Dwell time specifies how long the voltage is applied across the memristor. Measurement offset determines the time at which the current is sensed. The objective of these parameters is to control the *degree of similarity* of the output with regards to the pre-existing creative data. A detailed discussion of how these parameters effect the output is presented in this chapter. As mentioned earlier, it drifts away from conventional methods of data representation and proposes non-digital ways to control creative pieces generated by the system.

4.2 Why Biomemristors?

UC technologies have a history of being accessible only through well-resourced laboratories that use sophisticated instruments. Creative applications that adopt biocomputing will be useful to artists who are non-experts in the engineering of the system. Music composers, digital artists and poets to name but three can incorporate UC technologies in their creative processes, only if the framework is feasible to use. *PhyBox* (as shown in figure 4.3) is a stand-alone and portable system for *Physarum polycephalum*-based memristors to be incorporated as processing units. It was developed at Interdisciplinary Centre for Computer Music Research (ICCMR), Plymouth University, UK.

PhyBox uses a Raspberry Pi as its main processing unit. At present, it provides a framework to allow four biomemristors to process data. It uses breakout boards to apply voltages and read currents. MCP4725 is used as the digital to analogue converter (DAC) and ADS1115 is used as the analogue to digital converter (ADC). Both these breakout boards are manufactured by Adafruit. The Raspberry Pi and these boards are inexpensive and widely available. MCP4725 is used to source voltages across the biomemristor. Current is sensed in the circuit with the help of a shunt resistor placed in series with each memristor. ADS1115 measures the voltage across the shunt. Figure 4.1 shows a spike recorded using the PhyBox. This system runs a multi-threaded Python programme. It simultaneously processes data for four biomemristors. All threads carry the same function— source voltage across the memristor and measure the current flowing through it. An additional thread or programme runs in the background to implement the mapping procedure, that is to translate creative data into voltages and current readings into the creative data. The number of processing units are easily changeable by adding or deleting threads. Hence, PhyBox can be visualised as a microkernel that handles different types of data through biomemristors. The microkernel is encapsulated into a box that is fabricated by using a 3-D printer. The dimensions of PhyBox is $135 \times 105 \times 60$ *cm* and thus, making it a portable module. Pre-existing creative data can be fed into the system by either using a memory stick or the means of any real-time transport protocol. Receptacles that contain *Physarum polycephalum* are detachable from the PhyBox. They are clamped onto the system as shown in figure 4.3. This gives the user flexibility to easily replace them. The number of processing units depends on the type of creative data that is generated by the system. As mentioned earlier, digital art can be defined by three parameters— red, green and blue. Music can be defined by four parameters— pitch, velocity, time between note-ons (rhythm) and duration. The time taken by PhyBox to generate a creative piece depends on the dwell time specified by the user. For example, if there are 80 elements of creative data that needs to be processed and a dwell of 1.5 seconds is specified, then the system would take 120 s to generate the output. Since the biomemristors are easily detachable, linear electronic components like resistors can be used for certain attributes. For instance in music, you could have the rhythmic structure resembling the pre-existing creative data and allow the biomemristors to alter only the pitch. PhyBox harnesses the nonlinear behaviour of *Physarum polycephalum*. Hence, the accuracy while sourcing voltages and measuring currents is crucial. Several tests were conducted and readings were verified with a Keithley programmable electrometer. It proved to be an efficient framework for *Physarum polycephalum*-based memristors and figure 4.1 shows a spike recorded by it. The breakout boards can source a maximum voltage of 5V and *Physarum polycephalum* produces negative currents in some cases. Considering these factors, the operating range for *Physarum polycephalum* was decided on $0{-}3V$.

Braund and Miranda (2017*a*) conducted experiments to verify the mem-

Figure 4.3: PhyBox. A micro-kernel to use *Physarum polycephalum*-based memristors as processing units.

ristive behaviour of *Physarum polycephalum*. It produces pinched hysteresis curves that look similar to that of an ideal memristor, but not the same. The current–voltage (I–V) profile of an ideal memristor for an alternating current is a pinched hysteresis loop which passes through the origin (Leon, 2015). However, each biomemristor produces a unique I–V profile and would generate different deviations from the pre-existing creative data. Furthermore, the I–V profile of a biomemristor does not remain constant throughout its life cycle. It depends on its age and external factors such as light, temperature and humidity. Braund and Miranda (2017*a*) have also studied methods of altering a *Physarum polycephalum*-based memristors hysteresis profile. Here, the cell was treated with solutions such as calcium chloride ($CaCl_2$), which produced hysteresis curves that were considerably different from the original ones (Braund, 2017). This opens up opportunities to effect the output by using external parameters. Hence, the creative piece that is generated with the help of PhyBox cannot be predicted, but can definitely be controlled. This proves its potential to substitute stochastic processes in creative applications. Incorporating *Physarum polycephalum* as the processing unit for an attribute of data uses only one information bit. Each attribute of creative data can be generated solely by the nonlinear property of the organism. Implementing the same with a conventional computer would require many more information bits.

Snider (2008) studied the spiking behaviour of chemical memristors that are composed of metal oxides. The spikes are much shorter in duration. They attain a more stable memristance after 40 ms as opposed to 1000 ms in *Physarum polycephalum*-based memristors. Chemical memristors are generally not affordable out of laboratories and a hardware framework to record their spikes would be expensive to acquire. On the contrary, Stephenson and Stempen (1994) discuss simple ways to obtain *Physarum polycephalum* and explain that it is also naturally available (Adamatzky, 2010). Chemical memristors produce ideal curves and do not change their behaviour with time. This would lead to higher possibilities of repetition in the generated output. These facts explain the advantages of using biomemristors over chemically manufactured memristors for creative applications.

4.3 Mapping Procedure

This section discusses the mapping procedure of the system and considers the resistor's behaviour to be ideal. It assumes the operating range of the biomemristors to be $0-3V$.

4.3.1 Events to Voltages

The system aims to generate a creative piece which is inspired by pre-existing creative data. Therefore, the data needs to be sorted based on a specific criterion. In this chapter, we have used *popularity* as the criterion to sort data. If a specific event in the data occurs more number of times than another, then it is assigned greater priority. If two events have occurred equal number of times, then the more recent event is assigned greater priority. Table 4.1 sorts different types of creative data based on *popularity*.

Pitch	**No. of occurrences**	**Priority**	**Assigned voltage (V)**
C4	9	1	0.375
A4	6	2	1.125
G5	4	3	1.875
C5	2	4	2.625

(a) Sorting of events in musical data of the attribute *pitch*.

Colour Value	**No. of occurrences**	**Priority**	**Assigned voltage (V)**
150	15	1	0.3
250	10	2	0.9
0	8	3	1.5
60	6	4	2.1
120	1	5	2.7

(b) Sorting of events in a 24-bit image of the different shades of *red*. It considers the value of red in the range $0-255$.

Table 4.1: Sorting of events in different types of creative data.

In order to develop a relation between sorted data and the spiking behaviour memristors, we observe the rate at which the memristor is changing its memristance. Figure 4.1 depicts that the memristor attains a more stable memristance after a certain period of time. Greater changes in voltage cause greater magnitudes of spikes. Hence, we devise a relationship between the stability of memristance and the popularity of events. Events that have higher number of occurrences are assigned lower voltages and events that have lower number of occurrences are assigned higher voltages. All the events that are detected by the system need to be *quantised* into definite voltage ranges between 0 and 3V. The translation of events into voltages follows the *midrise quantisation*, a technique followed by many communication systems. It is a type of uniform quantisation that divides a particular range into equal intervals. Further information on midrise quantisation can be found in Bosi and Goldberg (2003). Figure 4.4 illustrates the mapping of events into voltages. It allocates a voltage range for each event. The mean of each range is the voltage that is assigned to the respective event.

In this system, voltages are assigned dynamically. Each time an event occurs in the creative data, the table of assigned voltages is updated. Let us consider the first 7 pitches occurring in the musical composition *Fur Elise* by Beethoven. The sequence of pitches are *E5* $\rightarrow$ *D#5* $\rightarrow$ *E5* $\rightarrow$ *D#5* $\rightarrow$ *E5* $\rightarrow$ *B4* $\rightarrow$ *D5*. Table 4.2 shows how voltages are updated. Note that the more recent event is give higher priority if the events have occurred equal number of times.

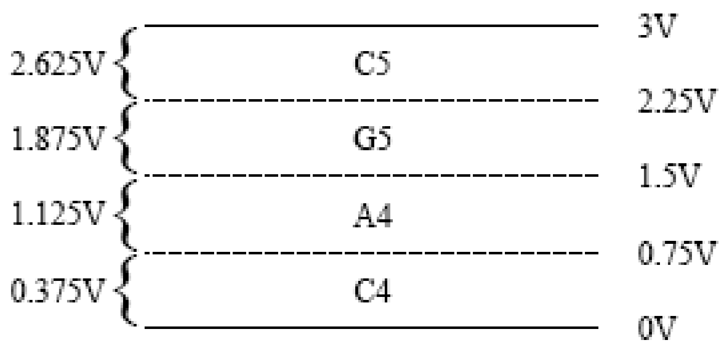
The equation to calculate the assigned voltage is given below.

$$v_{\mathrm{a}} = \frac{3}{N}(p - \frac{1}{2})Volts \tag{4.2}$$

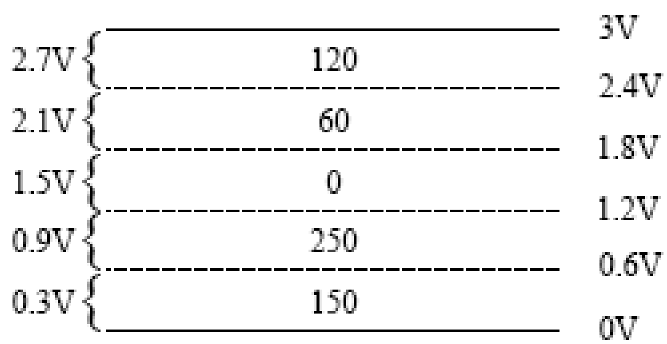
where v_a is the assigned voltage, N is the number of events and p is the priority of the event. Every *pitch* occurring in a musical piece can be considered to be an individual event. However, events like *colour value* need to be rounded. This is done to handle redundancy of creative data. A pixel with RGB values {200,50,20} is not very different from one with values {199,50,20}. Similarly, musical events have high redundancy in velocity and time-related events. Therefore, for testing Phybox with musical data, we rounded velocity to the nearest 5 and time-related events to the nearest 30 ms.

4.3.2 Currents to Events

A similar procedure is followed to translate current readings into creative data. While mapping creative events to voltages, the voltage range is already known, that is $0{-}3V$. This is not the case while mapping current to events. Hence, we need to obtain a current range to implement the mapping. For each spike, the system records the maximum and minimum current value, I_{max} and I_{min} respectively. The current flowing through the circuit is recorded every 100 ms. Obtaining current readings at a faster rate leads to poor performance of the *system clock* of the Raspberry Pi. The system was tested with different sampling rates and 10Hz was chosen empirically. However, experiments need



(a) Mapping of four pitch events in musical data to voltages.



(b) Mapping of different shades of red in a 24-bit image to voltages.

Figure 4.4: Mapping of events to voltages.

to be conducted to find an optimal sampling rate. A positive change in voltage produces a positive spike as shown in figure 4.1 and a negative change in voltage produces a negative spike as shown in figure 4.5.

After obtaining current readings, we record the minimum and maximum memristance that can be exhibited by the memristor. The minimum and maximum memristance are calculated by the following equations.

$$M_{\text{min}} = \frac{v_{\text{a}}}{I_{\text{max}}} \tag{4.3}$$

$$M_{\text{max}} = \frac{v_{\text{a}}}{I_{\text{min}}} \tag{4.4}$$

Pitch	No. of occurrences	Priority	Assigned voltage (V)
E5	1	1	1.5

(a) Assigned voltages after first event.

Pitch	No. of occurrences	Priority	Assigned voltage (V)
$D^{\#}5$	1	1	0.75
E5	1	2	2.25

(b) Assigned voltages after second event.

Pitch	No. of occurrences	Priority	Assigned voltage (V)
E5	2	1	0.75
$D^{\#}5$	1	2	2.25

(c) Assigned voltages after third event.

Pitch	No. of occurrences	Priority	Assigned voltage (V)
$D^{\#}5$	2	1	0.75
E5	2	2	2.25

(d) Assigned voltages after fourth event.

Pitch	No. of occurrences	Priority	Assigned voltage (V)
E5	3	1	0.75
$D^{\#}5$	2	2	2.25

(e) Assigned voltages after fifth event.

Pitch	No. of occurrences	Priority	Assigned voltage (V)
E5	3	1	0.5
$D^{\#}5$	2	2	1.5
B4	1	3	2.5

(f) Assigned voltages after sixth event.

Pitch	No. of occurrences	Priority	Assigned voltage (V)
E5	3	1	0.375
$D^{\#}5$	2	2	1.125
D5	1	3	1.875
B4	1	4	2.625

(g) Assigned voltages after seventh event.

Table 4.2: Assigned voltages updated dynamically.

where M_{min} is the minimum memristance, M_{max} is the maximum memristance and v_a is the assigned voltage. The current range *[I_1,I_2]* is obtained by the following equations. It takes the table of assigned voltages into account.

$$I_1 = \frac{min(V_{\text{a}})}{M_{\text{max}}} \tag{4.5}$$

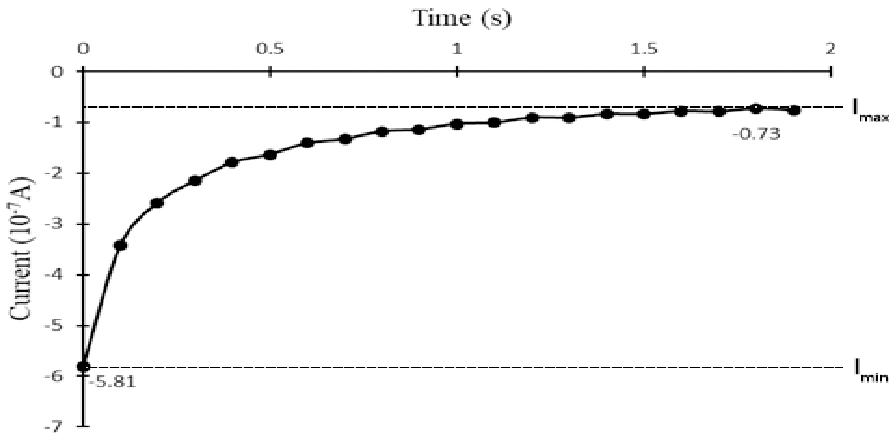


Figure 4.5: Calculation of I_{max} and I_{min} for a negative spike.

$$I_2 = \frac{max(V_\mathrm{a})}{M_\mathrm{min}} \tag{4.6}$$

where V_a is the set of all assigned voltages. Current is translated to creative data by using a similar procedure. Figure 4.6 shows the translation of current readings into pitches. It considers the table of assigned voltages in figure 4.4a. Lower values of current are mapped to more popular events and higher values of current are mapped to less popular events. Note that the direction of current is taken into consideration. For instance, $-5mA$ is considered to be lower than $2mA$.

4.3.3 Resistor as Ideal Behaviour

This section discusses the output of the Phybox for connecting a resistor instead of a biomemristor. The current flowing through a given resistor is constant for a given voltage. Hence, the current flowing in the circuit can be calculated by using Ohm's law. Let us consider the example of a circuit that carries a resistance of $500k\Omega$. The resistor's output for pitches in Fur Elise is illustrated in figure 4.7. The output is the same as the original song. It is due to the fact that M_{min} and M_{min} are equal for a resistor. Dwell time and offset percentage do not effect the output. Furthermore, the resistance does not effect the output. Hence, for any given resistance and combination of user-adjustable parameters, the output is always the original creative piece. This proves that a creative piece generated by using *Physarum polycephalum*-based memristors is solely based on nonlinear behaviour. Several tests were conducted with PhyBox to verify the ideal behaviour of resistors. $200k\Omega$, $500k\Omega$ or $700k\Omega$ resistors were connected to test the circuit with varying resistances. The output produced by the system consistently matched the input. Tests to confirm this statement are described

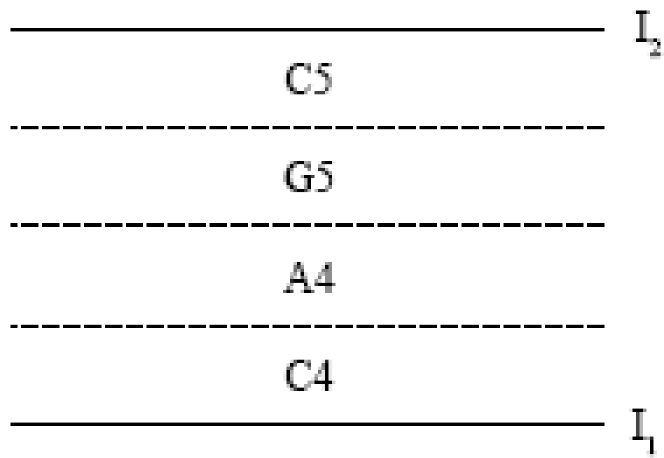


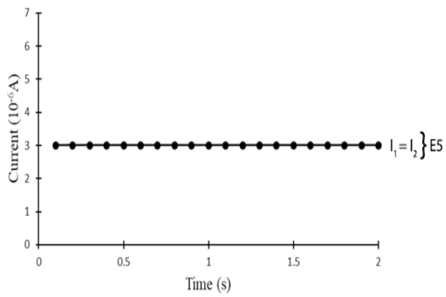
Figure 4.6: Translation of current readings into pitches.

in section 4.4.

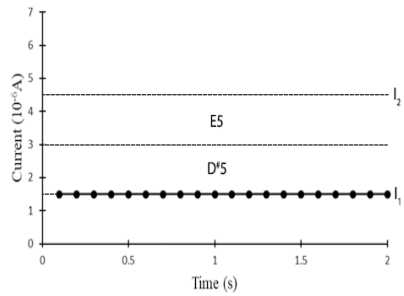
The linear behaviour of resistors can be creatively incorporated into such systems. In digital art, one could retain the shades of red, green or blue as in the original picture. In music, the rhythmic structure of the original song could be retained. Such features give the user more control over the output generated by the system. Fur Elise was fed into the Phybox and $500k\Omega$ resistors were connected for time between note-ons and duration. The output produced by the system retained the rhythmic structure of the song and produced variations only in pitch and velocity.

4.3.4 User-adjustable Parameters

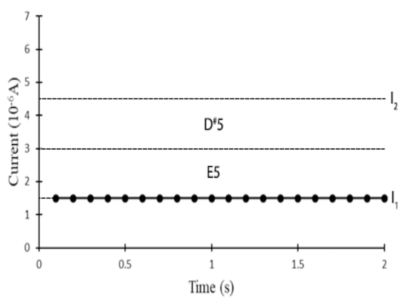
As mentioned earlier, the behaviour of *Physarum polycephalum* is controlled by two parameters — dwell time and measurement offset. A longer dwell time gives the memristor time to attain a more stable memristance and vice-versa. For example, if the dwell time is 2 s and measurement offset is 25%, then current is sensed 0.5 s after the voltage has been applied across the memristor. This section explains how these parameters effect the output produced by the memristors. Measurement offset provides a way controlling the *degree of similarity* with the original creative piece. When current readings are closer to the mean of I_1 and I_2, the memristor exhibits behaviour that is more linear. Therefore, current readings that are closer to the mean produce outputs that are more similar to the input. Readings that are farther from the mean produce outputs that are less similar to the input. Figure 4.8 shows how the output of biomemristors vary for different values measurement offset. 5% is expected to produce outputs



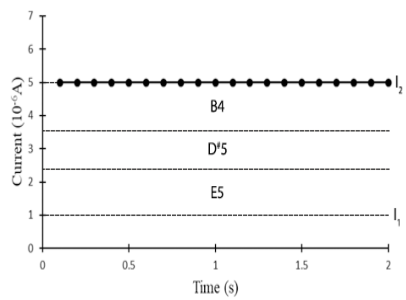
(a) Resistor's response for first pitch in Fur Elise.



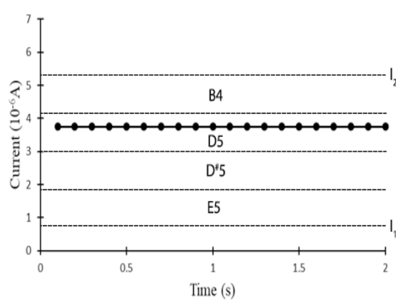
(b) Resistor's response for second and fourth pitch in Fur Elise.



(c) Resistor's response for third and fifth pitch in Fur Elise.



(d) Resistor's response for sixth pitch in Fur Elise.



(e) Resistor's response for seventh pitch in Fur Elise.

Figure 4.7: Resistor's response for pitches in Fur Elise. Solid line is the resistor reading calculated by using Ohm's law. The output generated is *E5* $\rightarrow$ *D*$^{\#}$*5* $\rightarrow$ *E5* $\rightarrow$ *D*$^{\#}$*5* $\rightarrow$ *E5* $\rightarrow$ *B4* $\rightarrow$ *D5*.

that are more similar to the input when compared to 50% and 95%. Tests to confirm these statements are described in section 4.4.

This chapter predicts that the nonlinearity realised by measurement offsets of 0% and 95% are slightly different. Both have similar degree of dissimilarity

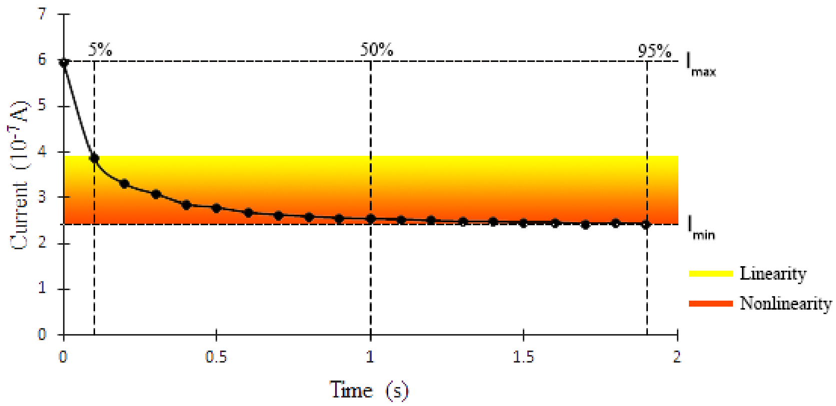


Figure 4.8

with the input, but they deviate from the input differently. On one hand, when the input creative piece induces a positive spike, 0% offset tends to produce an event that is less popular. 95% offset tends to produce an event that is more popular. On the other hand, when the input piece induces a negative spike, 0% offset tends produce an event of higher popularity. 95% offset tends to produce an event of lower popularity. These statements need to be analysed with the help of data mining techniques and a clearer distinction between 0% and 95% offsets needs to be devised.

As mentioned earlier, a longer dwell time gives the memristor more time to attain a more stable memristance. In other words, it has longer time to recover from the spike. On one hand, for a dwell time of 0.5 s, two consecutively occurring positive spikes are more likely to produce currents of higher values when compared to a dwell time of 2 s. On the other hand, for a dwell time of 0.5 s, two consecutively occurring negative spikes are more likely to produce currents of lower values. However, a clear relation of dwell time on the output is still not deduced. This due to the fact that the mapping procedure calculates I_{max} and I_{min} for each event. Hence, the effect of dwell time is likely to be cancelled out.

4.3.5 Hardware Limitations

The operating range for *Physarum polycephalum* is $0-3V$. PhyBox uses a 16-bit ADC and a shunt resistance of $100k\Omega$. By using Ohm's law, the current resolution of the system was calculated to be $1.25nA$. Greater values of memristance would require the ADC to make more accurate measurements because currents will be of a smaller magnitude. While performing experiments, *Physarum polycephalum*-based memristors consistently had a memristance of lower than $7M\Omega$. Let N be the number of discrete events that are fed into the PhyBox. There exists a value N_{max} such that for $N > N_{\max}$, the resistor would no longer produce the input musical piece and lose its ideal behaviour. While conducting experiments with Phybox, N_{max} was never reached. Creative data with greater values of N needs to be fed into the system in order to obtain a practical value of N_{max}. N_{max} can be increased by improving the resolution of the ADC or increasing the shunt resistance.

4.4 Creative Systems

The examples of creative systems discussed so far were based on discrete events. This section explores the use of transitions, which is analogous to Markov chains presenting probability distribution of future states based on the current state. A transition is defined as the occurrence of two consecutive events. If a creative piece contains N events, then it will contain *N-1* transitions. In these systems, the mapping procedure is implemented for transitions instead of individual events. Therefore, if a resistor is used as the processing unit, the output would be identical to the input, excluding the first event. This is because there is no transition that exists after feeding in the first event.

The system maintains an independent translation table for each event. Table 4.3 shows the assignment of voltages for Fur Elise. On occurrence of each event, only relevant transitions are considered for assignment of voltages. For instance in Fur Elise, after the occurrence of the sixth event, the relevant transitions are {E5 $\rightarrow$ D$^{\#}$5} and {E5 $\rightarrow$ B4}. The transition {D$^{\#}$5 $\rightarrow$ E5} will not be considered because the sixth event is *E5* and not *D$^{\#}$5*. Equation 4.2 can be used to calculate the table of assigned voltages, but N would stand for the number of relevant transitions and not the number of events.

While mapping currents to events, I_1 and I_2 can be similarly obtained by using equations 4.3−4.6. The range $[I_1, I_2]$ is divided into only transitions that are relevant to the input. For example in Fur Elise, on the occurrence of the sixth event, the current range is divided into two regions and not three.

Creative systems that are based on transitions require more number of events to populate the translation tables. If the musical sequence $C4 \rightarrow D4 \rightarrow E4 \rightarrow F4 \rightarrow G4 \rightarrow A4 \rightarrow B4 \rightarrow C5$ is fed into the system, then it would generate a replica of the input. This is because each event has only one transition that is relevant to it. It allows us to feed in huge amount of data and allow the system to generate new creative pieces. It also requires many more events to reach N_{max}.

Tests were conducted to evaluate the effect of user-adjustable parameters on the musical outputs produced by the biomemristors. Two samples A and B of four biomemristors each were taken. They were tested with two musical compositions— *Fur Elise*, by Beethoven and *Gavotte en rondeau* by Bach. Dwell time was kept constant at 2 s and measurement offset was varied between 5%, 50% and 95%. Output produced by the PhyBox was evaluated by calculating its deviation *dev* from the input melody. If the input transition has a priority i and the output transition has a priority k, *dev* is defined as $\mid i - k \mid$. The *degree of similarity* was quantified as Avg_{dev} and was calculated by equation 4.7.

Transition	No. of occurrences	Priority	Assigned voltage (V)
E5 → $D^{\#}5$	1	1	1.5

(a) Assigned voltages after second event. Translation table for pitch E5.

Transition	No. of occurrences	Priority	Assigned voltage (V)
$D^{\#}5$ → E5	1	1	1.5

(b) Assigned voltages after third event. Translation table for pitch $D^{\#}5$.

Transition	No. of occurrences	Priority	Assigned voltage (V)
E5 → $D^{\#}5$	2	1	1.5

(c) Assigned voltages after fourth event. Translation table for pitch E5.

Transition	No. of occurrences	Priority	Assigned voltage (V)
$D^{\#}5$ → E5	2	1	1.5

(d) Assigned voltages after fifth event. Translation table for pitch $D^{\#}5$.

Transition	No. of occurrences	Priority	Assigned voltage (V)
E5 → $D^{\#}5$	2	1	0.75
E5 → B4	1	2	2.25

(e) Assigned voltages after sixth event. Translation table for pitch E5.

Transition	No. of occurrences	Priority	Assigned voltage (V)
B4 → D5	1	1	1.5

(f) Assigned voltages after seventh event. Translation table for pitch B4.

Table 4.3: Translation tables updated after the occurrence of the second event.

$$Avg_{\mathrm{dev}} = \frac{\mid i - k \mid}{N - 1} \tag{4.7}$$

where N is the number of notes in the song.

The tests confirmed that 5% had maximum similarity with the input melody and 50% and 95% showed much higher deviation. This allows the user to specify adjustable parameters as per the requirements. It also confirms the behaviour predicted in figure 4.8. The output produced by a specific combination of user-adjustable parameters was always different. For instance, tests no. 1 and 4 had the same input parameters, but their outputs are different and showed similar degree of similarity with the input melody.

Tests were also conducted with two samples C and D of only resistors. Sample C contained $100k\Omega$, $500k\Omega$, $200k\Omega$ and $100k\Omega$. Sample D contained $700k\Omega$, $500k\Omega$, $200k\Omega$ and $100k\Omega$. The results are tabulated in table 4.5.

Avg_{dev} for all attributes was consistently 0 and thus, proves that creative pieces were purely created by the memristor's nonlinearity. Tests were also conducted by connecting memristors for pitch and velocity and $200k\Omega$ resistors for time between note-ons and duration. Results are shown in table 4.6.

The output produced by PhyBox in table 4.6 maintained the rhythmic structure of the original input melody. This gives the user flexibility to allow the

Test	Sample	Input	N	DT	MO	$\textbf{Avg}_{\textbf{dev1}}$	$\textbf{Avg}_{\textbf{dev2}}$	$\textbf{Avg}_{\textbf{dev3}}$	$\textbf{Avg}_{\textbf{dev4}}$
1	A	Fur Elise	296	2	5	0.39	0.88	0.35	0.77
2	A	Fur Elise	296	2	50	1.33	2.40	0.65	2.27
3	A	Fur Elise	296	2	95	1.49	2.52	0.65	2.32
4	B	Fur Elise	296	2	5	0.43	0.96	0.34	0.78
5	B	Fur Elise	296	2	50	1.44	2.52	0.59	2.23
6	B	Fur Elise	296	2	95	1.41	2.55	0.6	2.25
7	A	Gavotte en rondeau	45	2	5	0.3	0.43	0.43	0.36
8	A	Gavotte en rondeau	45	2	50	0.98	1.27	0.56	0.56
9	A	Gavotte en rondeau	45	2	95	1.07	1.38	0.61	0.59
10	B	Gavotte en rondeau	45	2	5	0.31	0.56	0.31	0.38
11	B	Gavotte en rondeau	45	2	50	1.05	1.56	0.31	0.36
12	B	Gavotte en rondeau	45	2	95	0.98	1.27	0.56	0.5

Table 4.4: *N*, DT and MO stand for number of events, dwell time and measurement offset respectively. Avg_{dev1}, Avg_{dev2}, Avg_{dev3} and Avg_{dev4} stand for the Avg_{dev} of pitch, velocity, time between note-ons and duration respectively.

Test	Sample	Input	N	DT	MO	$\textbf{Avg}_{\textbf{dev1}}$	$\textbf{Avg}_{\textbf{dev2}}$	$\textbf{Avg}_{\textbf{dev3}}$	$\textbf{Avg}_{\textbf{dev4}}$
1	C	Fur Elise	296	2	50	0	0	0	0
2	C	Fur Elise	296	2	50	0	0	0	0
3	D	Govette en rondeau	45	2	50	0	0	0	0
4	D	Govette en rondeau	45	2	50	0	0	0	0

Table 4.5: Output of PhyBox with resistors used for all four attributes.

Test	Sample	Input	N	DT	MO	$\textbf{Avg}_{\textbf{dev1}}$	$\textbf{Avg}_{\textbf{dev2}}$	$\textbf{Avg}_{\textbf{dev3}}$	$\textbf{Avg}_{\textbf{dev4}}$
1	B	Fur Elise	296	2	50	1.38	2.61	0	0
2	B	Govette en rondeau	45	2	50	0.97	1.43	0	0

Table 4.6: Output of PhyBox with memristors for pitch and velocity and resistors for time between note-ons and duration.

output to have some attributes that are identical to the input. The mapping procedure is solely based on popularity and does not depend on the type of creative data. Therefore, these tests confirmed the potential of PhyBox to act as a generic framework for creative applications.

4.5 Concluding Discussions

This chapter presented a generic biocomputing system for creative applications. It harnesses the spiking property of memristors to develop such applications. The advantages of using nonlinearity over random or pseudo-random processes have been discussed. It stressed on the importance of using non-digital ways to represent data. It defined the relationship between pre-existing creative data and the generated output with the help of *nonlinear* parameters that control the behaviour of *Physarum polycephalum.*

The appropriateness of using biomemristors as opposed to chemical memristors was discussed. Biomemristors acted as inexpensive and accessible processing units for the system. Furthermore, the memristance of *Physarum polycephalum* depends on environmental conditions like light, temperature and humidity. The hysteresis curve produced by *Physarum polycephalum* varies over time. Hence, it generated interesting variations that can be useful for creative applications. The chapter gave a brief introduction to PhyBox, a micro-kernel to harness *Physarum polycephalum*-based memristors as processing units. It allowed four biomemristors to simultaneously process data. PhyBox is a convenient and portable system that can be easily accessed by creative practitioners. It belongs to the field of UC and hence, several approaches are still at an elementary stage. In terms of information bits, Phybox only uses one memristor to process each attribute of data. This is much lower than conventional computers that implement stochastic processes. The processing time depended on the dwell time specified by the user. Based on the experiments conducted in this chapter, it took 2 s to process each event in the creative data.

The mapping procedure took linear electronic components into consideration. It assumed the resistor's output to be ideal. The procedure was developed such that the system produced a replica of the input if resistors were connected instead of memristors. The value of resistance did effect the output. This proved that the creative pieces generated by PhyBox were purely because of the nonlinear behaviour of *Physarum polycephalum.* The idea to translate events to voltages was derived from midread quantisation, a technique adopted by many communication systems. Current readings were translated back to events by a similar procedure.

The chapter discussed the use of transitions as opposed to individual events. This concept was analogous to n-order Markov chains that is widely adopted by creative systems. It demonstrated that sequences of different lengths can be incorporated by PhyBox and the mapping procedure had to be implemented accordingly. The effect of measurement on the output generated by PhyBox was discussed in detail. The system was tested with two musical compositions, *Fur Elise* and *Gavotte en rondeau.* 5% offset produced outputs that were most similar to the input. 95% offset generated creative pieces that had greater dissimilarity with the input. The output generated by a particular combination of user-adjustable parameters did not repeat itself, but showed similar degree of similarity with the original piece in successive tests. It was also demonstrated that the behaviour of each biomemristor is unique. Tests were conducted with combinations of resistors and memristors. Resistors were connected for two attributes and memristors were connected for the other two attributes. Attributes processed by the resistors exactly reproduced the input and memristors produced variations for their respective attributes. This enabled the user to alter only specific attributes of data.

Most examples in this chapter were based on musical systems. However, it demonstrated the mapping procedure to be independent of the type of creative data. It can be incorporated into systems that generate digital art or even text. It was flexible to increase or decrease the number of biomemristors that were processing data. This allowed the developer to alter the system according to the specifications of the type of creative data. It proved to be a generic mapping procedure that accepted creative events and generated data by employing the nonlinear behaviour of biomemristors.

4.6 Future Work

This chapter explored the use of nonlinear behaviour of biomemristors as opposed to random or pseudo-random processes implemented in conventional creative systems. Biocomputation for such systems is relatively new and has wide scope of further development. The chapter demonstrated a reliable and portable biocomputing framework, but several challenges might have to be addressed for the large scale deployment for general creative practitioners. The advantages of using nonlinearity need to be further understood by adopting it in different types of systems. All experiments in this chapter were demonstrated through music. However, practical implementations and tests of PhyBox with other types of creative data need to conducted. The role of nonlinear parameters needs to be clearly explained to the user. A stronger and clearer relation between the input and the output needs to be derived.

This chapter conducted experiments for measurement offsets of only 5%, 50% and 95%. Tests with other values of measurement offset need to be conducted to understand the effect of this parameter in greater depth. The region between 0 s and 0.1 s in the spiking response of biomemristors needs to researched. Braund and Miranda (2017*a*) suggested that 2 s is an optimal dwell time for the operation of memristors. But the mapping procedure proposed in this chapter might require a much shorter dwell time. This chapter predicts that the system could operate with a dwell time of 1 s. This is due to the fact that 50% and 95% offsets produced creative pieces of almost same degree of dissimilarity. Furthermore, their positions in figure 4.8 attribute to similar current values. This would increase the processing time by at least two times. This chapter did not find a notable impact of dwell time on the output generated by the PhyBox. Hence, this parameter could be kept constant by finding an optimal dwell time. On the whole, the prospects of biocomputing in creative computing seem bright with a wide spectrum of unexplored approaches which might change ways of data representation, storage, processing and generation.

4.7 References

Adamatzky, A. (2010), *Physarum Machines: Computers from Slime Mould*, Vol. 74 of *A*, World Scientific.

Bosi, M. and Goldberg, R. E. (2003), *Introduction to Digital Audio Coding and Standards*, Kluwer Academic Publishers.

Braund, E. (2017), Unconventional Computing and Music: An Investigation into Harnessing Physarum polycephalum, PhD thesis, University of Plymouth.

Braund, E. and Miranda, E. R. (2017*a*), An approach to building musical bioprocessors with physarum polycephalum memristors, *in* E. Miranda, ed., 'Guide to Unconventional Computing for Music', Springer, pp. 219–244.

Braund, E. and Miranda, E. R. (2017*b*), 'On building practical biocomputers for real-world applications: Receptacles for culturing slime mould memristors and component standardisation', *Journal of Bionic Engineering* **14**(1), 151–162.

Brooks, F. P., Hopkins, A. L., Neumann, P. G. and Wright, W. V. (1957), 'An experiment in musical composition', *IRE Transactions on Electronic Computers* **EC-6**(3), 175–182.

Chua, L. (1971), 'Memristor – the missing circuit element', *IEEE Transactions on Circuit Theory* **18**(5), 507–519.

Conroy, J. M. and O'leary, D. P. (2001), Text summarization via hidden markov models, *in* 'Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval', ACM, pp. 406–407.

Cope, D. (1996), *Experiments in Musical Intelligence*, A-R Editions, Inc.

Cope, D. (2005), *Computer Models of Musical Creativity*, MIT Press Cambridge.

Gale, E., Adamatzky, A. and de Lacy Costello, B. (2015), 'Slime mould memristors', *BioNanoScience* **5**(1), 1–8.

Hedges, S. A. (1978), 'Dice music in the eighteenth century', *Music & Letters* **59**(2), 180–187.

Johnsen, G. K. (2012), 'An introduction to the memristor – a valuable circuit element in bioelectricity and bioimpedance', *Journal of Electrical Bioimpedance* **3**(1), 20–28.

Leon, C. (2015), 'Everything you wish to know about memristors but are afraid to ask', *Radioengineering* **24**(2), 319.

Miranda, E. (2001), *Composing Music with Computers*, Music Technology Series, Focal Press.

Pinkerton, R. C. (1956), 'Information theory and melody', *Scientific American* **194**(2), 77–87.

Roads, C. (1996), *The Computer Music Tutorial*, MIT press.

Ross, S. M. (1996), *Stochastic processes*, 2 edn, JohnWiley & Sons, Inc.

Simonton, D. K. (2003), 'Scientific creativity as constrained stochastic behavior: The integration of product, person, and process perspectives.', *Psychological Bulletin* **129**(4), 475.

Snider, G. S. (2008), Spike-timing-dependent learning in memristive nanodevices, *in* 'Proceedings of the 2008 IEEE International Symposium on Nanoscale Architectures', IEEE Computer Society, pp. 85–92.

Stephenson, S. L. and Stempen, H. (1994), *Myxomycetes: Handbook of Slime Molds*, Timber Press, Inc.

Tour, J. M. and He, T. (2008), 'Electronics: The fourth element', *Nature* **453**(7191), 42.

Xenakis, I. (1992), *Formalized Music: Thought and Mathematics in Composition*, number 6, Pendragon Press.