

Towards Harmonic Extensions of Pulsed Melodic Affective Processing – Further Musical Structures for Increasing Transparency in Emotional Computation

Alexis J. Kirke and Eduardo R. Miranda

*Interdisciplinary Centre for Computer Music Research, School of Humanities, Music and Performing Arts, University of Plymouth, Drake Circus, Plymouth, PL4 8AA, UK,
E-mail: Alexis.Kirke@Plymouth.ac.uk*

Received: January 21, 2013. Accepted: March 11, 2013.

Pulsed Melodic Affective Processing (PMAP) is a method for the processing of artificial emotions in affective computing. PMAP is a data stream which can be listened to, as well as computed with. The affective state is represented by numbers which are analogues of musical features, rather than by a binary stream. Previous affective computation has been done with emotion category indices, or real numbers representing positivity of emotion, etc. PMAP data can be generated directly by sound and rhythms (e.g. heart rates or key-press speeds) and turned directly into music with minimal transformation. This is because PMAP data is music and computations done with PMAP data are computations done with music. Why is this important? Because PMAP is constructed so that the emotion which its data represents at the computational level, will be similar to the emotion which a person “listening” to the PMAP melody hears. So PMAP can be used to calculate “feelings” and the result data will “sound like” the feelings calculated. Harmonic PMAP (PMAPh) is an extension of PMAP allowing harmonies to be used in calculations.

Keywords: Communications, Human-Computer Interaction, Music, Affective Computing, Boolean Logic, Neural Networks, Emotions, Multi-Agent Systems, Robotics, Harmony, Computation

1 INTRODUCTION

Previous work on unconventional computing and music has focused on using unconventional computation methods as engines for new modes of musical

expression. For example using *in vitro* neural networks [1] or slime molds [2] to drive a sound synthesizer. The research has not focused on studying the computational properties of the underlying unconventional computation, but on examining the feasibility of the sound generation for compositional purposes, and of applying some control to the generation. In the current paper, music is utilized not as an output of unconventional computation but as a method of unconventional computation. It has been said that almost any set of symbols with some structural interaction rules can be used to compute. However music holds a unique place as a form of emotional expression [3]. Thus if the symbols and rules of music can be used in computations related to affective (emotional) processes, then the possibility arises of a symbol set which expresses the results of its own computational result.

This paper is based on the idea of using melodies as a tool for affective computation in artificial systems. Such an idea is not so unusual when one considers the data stream in spiking neural networks. Spiking neural networks (SNNs) have been studied both as artificial entities and as part of biological neural networks in the brain. These are networks of biological or artificial neurons whose internal signals are made up of spike or pulse trains that propagate through the network in time. [4] has actually developed a back-propagation algorithm for artificial SNNs. Back-propagation is one of the key machine learning algorithms used to develop neural networks which can respond intelligently. In SNN's it is usually the spike rate (or "tempo") that encodes information, not the spike height, though there has been some biological work suggesting spike height, in addition to spike rate, may encode information in SNNs in the brain [5,6]. It is interesting to note that a series of timed pulses with differing heights can be naturally encoded by one of the most common musical representations used in computers: MIDI (the musical instrument digital interface). In its simplest form MIDI encodes a melody, which consists of a note timing and note pitch height. In this paper it is argued that melodies can be viewed as functional as well as recreational – they can fulfill the function of encoding an artificial emotional state in a form which can be used in affective computation directly expressible to human beings (or indeed to other machines). The basis of the data stream used in this paper for processing is a pulse stream in which the pulse rate encodes a tempo, and the pulse height encodes a pitch. This is extended here to include the use of harmonies to encode and compute affectivity.

1.1 Uses and Novelty of PMAP

Before explaining the motivations behind PMAP in more detail, an overview of the useful functionality will be given. Similarly the novel functionality which PMAP provides will be summarized. In terms of functionality PMAP provides a method for the processing of artificial emotions, which is useful in affective computing – for example combining emotional readings for input or output, making decisions based on that data or providing an artificial agent

with simulated emotions to improve their computation abilities. In terms of novelty, PMAph is novel in that it is a data stream which can be listened to, as well as computed with. The affective state is represented by numbers which are analogues of musical features, rather than by a binary stream of 1s and 0s. Previous work on affective computation has been done with normal data carrying techniques – e.g. emotion category index, a real number representing positivity of emotion, etc.

This element of PMAph provides an extra utility – PMAph data can be generated directly by sound and turn directly into sound. Thus rhythms such as heart rates or key-press speeds can be directly turned into PMAph data; and PMAph data can be directly turned into music with minimal transformation. This is because PMAph data *is* music and computations done with PMAph data are computations done with music. Why is this important? Because PMAph is constructed so that the emotion which a PMAph data stream represents in the computation engine, will be similar to the emotion that a person “listening” to PMAph-equivalent melody would be. So PMAph can be used to calculate “feelings” and the result data will “sound like” the feelings calculated. This is will be clarified over the course of this paper.

In the previous subsection it was described how this paper is motivated by similarities between MIDI-type structures and the pulsed-processing [7] computation found in artificial and biological systems. It is further motivated by three other key elements which will now be examined: (i) the increasing prevalence of the simulation and communication of affective states by artificial and human agents/nodes; (ii) the view of music as the “language of emotions”; (iii) the concept of audio-display of non-audio data.

1.2 Affective Processing and Communication

It has been shown that affective states (emotions) play a vital role in human cognitive processing and expression [8]:

1. Universal and Enhanced Communication – two people who speak different languages are still able to communicate basic states such as happy, sad, angry, and fearful.
2. Internal Behavioral modification - a person’s internal emotional state will affect the planning paths they take. For example affectivity can reduce the number of possible strategies in certain situations – if there is a snake in the grass, fear will cause you to only use navigation strategies that allow you to look down and walk quietly. Also pre- and de-emphasising certain responses: for example if a tiger is chasing you, fear will make you keep running and not get distracted by a beautiful sunset, or a pebble in your path.
3. Robust response – in extreme situations the affective reactions can bypass more complex cortical responses allowing for a quicker reaction,

or allowing the person to respond to emergencies when not able to think clearly – for example very tired, or in severe pain, and so forth.

As a result, affective state processing has been incorporated into robotics and multi-agent systems [9]. A further reason in relation to point (1) above and Human-Computer Interaction studies is that emotion may help machines to interact with and model humans more seamlessly and accurately [10]. So representation of simulation affective states is an active area of research.

The dimensional approach to specifying emotional state is one common approach. It utilizes an n-dimensional space made up of emotion “factors”. Any emotion can be plotted as some combination of these factors. For example, in many emotional music systems [11] two dimensions are used: Valence and Arousal. In that model, emotions are plotted on a graph (see Figure 1) with the first dimension being how positive or negative the emotion is (Valence), and the second dimension being how intense the physical arousal of the emotion is (Arousal). For example “Happy” is high valence, high arousal affective state, and “Stressed” is low valence high arousal state.

1.3 Music and Emotion

There have been a number of questionnaire studies done which support the argument that music communicates emotions [12]. Previous research [13] has suggested that a main indicator of valence is musical key mode. A major key mode implies higher valence, minor key mode implies lower valence. For example the overture of *The Marriage of Figaro* opera by Mozart is in a

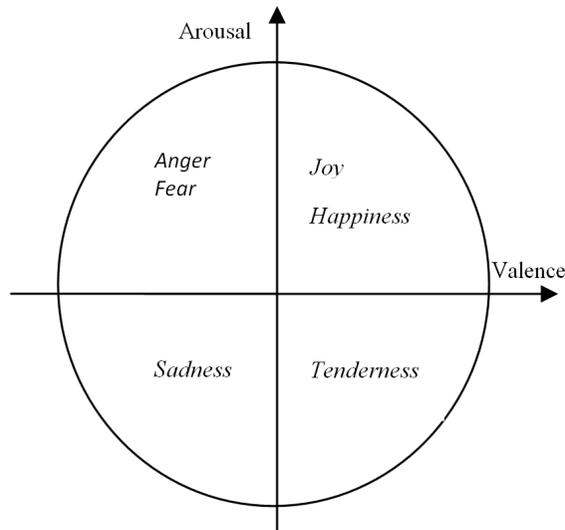


FIGURE 1
The Valence/Arousal Model of Emotion

major key; whereas Beethoven's melancholic "Moonlight" Sonata movement is in a minor key. It has also been shown that tempo is a prime indicator of arousal, with high tempo indicating higher arousal, and low tempo - low arousal. For example: compare Mozart's fast overture above with Debussy's major key but low tempo opening to "Girl with the Flaxen Hair". The Debussy piano-piece opening has a relaxed feel – i.e. a low arousal despite a high valence.

1.4 Sonification

Sonification [14] involves representing non-musical data in audio form to aid its understanding. Common forms of sonification include Geiger Counters and Heart Rate monitors. Sonification research has included tools for using music to debug programs [15], sonify activity in computer networks [16] and to give insight into stock market movements [17]. In the past, sonification has been used as an extra module attached to the output of the system under question.

A key aim of PMAP is to allow sonification in affective systems at any point in the processing path within the system. For example between two neurons in an artificial neural network, or between two agents in a multi-agent system, or between two processing modules within a single agent. The aim is to give the engineer or user quicker and more intuitive insight into what is occurring within the communication or processing path in simulated emotion systems by actually using simple music itself for processing and communication.

There are already systems which can take the underlying binary data and protocols in a network and map them onto musical features [18]. However PMAP is the only data processing model currently which is its own sonification and requires no significant mapping for sonifying. This is because PMAP data is limited to use in affective communications and processing, where music can be both data and sonification simultaneously. PMAP is not a new sonification algorithm, it is a new data representation and processing approach which is already in a sonified form.

This means that no conversion is needed between the actually processing / communication stream and the listening user - except perhaps downsampling. It also allows the utilization of such musical features as harmony and timing synchronization to be incorporated into the monitoring, when multiple modules / agents are being monitored simultaneously.

2 PMAP REPRESENTATION OF AFFECTIVE STATE

In the original monophonic PMAP [19] the data stream representing affective state is a stream of pulses. The pulses are transmitted at a variable rate. This can be compared to the variable rate of pulses in biological neural networks

in the brain, with such pulse rates being considered as encoding information. In PMAP this pulse rate specifically encodes a representation of the arousal of an affective state. A higher pulse rate is essentially a series of events at a high tempo (hence high arousal); whereas a lower pulse rate is a series of events at a low tempo (hence low arousal).

Additionally, the PMAP pulses can have variable heights with 10 possible levels. For example 10 different voltage levels for a low level stream, or 10 different integer values for a stream embedded in some sort of data structure. The purpose of pulse height is to represent the valence of an affective state, as follows. Each level represents one of the musical notes C,D,Eb,E,F,G,Ab,A,Bb,B. For example 1mV could be C, 2mV be D, 3mV be Eb, etc. We will simply use integers here to represent the notes (i.e. 1 for C, 2 for D, 3 for Eb, etc). These note values are designed to represent a valence (positivity or negativity of emotion). This is because, in the key of C, pulse streams made up of only the notes C,D,E,F,G,A,B are the notes of the key C major, and so will be heard as having a major key mode – i.e. positive valence. Whereas streams made up of C,D,Eb,F,G,Ab,Bb are the notes of the key C minor, and so will be heard as having a minor key mode – i.e. negative valence.

For example a PMAP stream of say [C,C,Eb,F,D,Eb,F,G,Ab,C] (i.e. [1,1,3,5,3,4,5,6,7]) would be principally negative valence because it is mainly minor key mode. Whereas [C,C,E,F,D,E,F,G,A,C] (i.e. [1,1,4,5,2,4,5,6,8]) would be seen as principally positive valence. And the arousal of the pulse stream would be encoded in the rate at which the pulses were transmitted. So if [1,1,3,5,3,4,5,6,7] was transmitted at a high rate, it would be high arousal and high valence – i.e. a stream representing ‘happy’ (see Figure 1). Where as if [1,1,4,5,2,4,5,6,8] was transmitted at a low pulse rate then it will be low arousal and low valence – i.e. a stream representing ‘sad’.

Note that [1,1,3,5,3,4,5,6,7] and [3,1,3,5,1,7,6,4,5] both represent high valence (i.e. are both major key melodies in C). This ambiguity has a potential extra use. If there are two modules or elements both with the same affective state, the different note groups which make up that state representation can be unique to the object generating them. This allows other objects, and human listeners, to identify where the affective data is coming from.

In non-simulated systems the PMAP data would be a stream of pulses. In fact in the second example below, a pulse-based data stream (MIDI) is used directly. However in performing the analysis on PMAP for simulation, it is sometimes convenient to utilize a parametric form to represent the data stream form. The parametric form represents a stream by a Tempo-value variable and a Key-mode-value variable. The Tempo-value is a real number varying between 0 (minimum pulse rate) and 1 (maximum pulse rate). The Key-mode-value is an integer varying between -3 (maximally minor) and 3 (maximally major).

3 PREVIOUS PMAP RESULTS

Monophonic PMAP has been applied and tested in a number of simulations. As there is no room here to go into detail, these systems and their results will be briefly described. They are:

- a. A security team multi-robot system [19, 20, 21]
- b. A musical neural network to detect textual emotion [19, 20, 21]
- c. A stock market algorithmic trading and analysis approach [19, 22]
- d. A multi-layered affective protocol for robust communication [20]

The security robot team simulation assumed robots with two levels of intelligence: a higher level more advanced cognitive function and a lower level basic affective functionality. The lower level functionality could take over if the higher level ceased to work. A new type of logic gate was designed to use to build the lower level: musical logic gates. PMAP equivalents of AND, OR and NOT were defined, inspired by Fuzzy Logic. Brief descriptions of these are given later in this paper. It was shown that using a circuit of such gates, PMAP could provide basic fuzzy search and destroy functionality for an affective robot team. It was also found that the state of a three robot team was human audible by tapping into parts of the PMAP processing stream.

As well as designing musical logic gates, a form of musical artificial neuron was defined. A simple two layer PMAP neural network was implemented using the MATLAB MIDI toolbox. The network was trained by gradient descent to recognise when a piece of text was happy and when it was sad. The tune output by the network exhibited a tendency towards “sad” music features for sad text, and “happy” music features for happy text. The stock market algorithmic trading and analysis system involved defining a generative affective melody for a stock market based on its trading imbalance and trading rate. This affective melody was then used as input for a PMAP algorithmic trading system. The basic system was shown to make better profits than random in a simulated simplified stock market.

The multi-layered affective protocol in (d) above utilized ideas from human expressive performance of music which supports that there are two levels of communication in a human’s musical performance, both of which can separately communicate emotion. They are the compositional level and the expressive performance level. By applying computer expressive performance simulations to a PMAP data stream, it was shown that the affective state of the data stream could be communicated between computer nodes even when the underlying PMAP data was greatly distorted by simulated communications errors. A key element of this approach to PMAP comms robustness was that the expressive performance overlay would “sound like” its affective content when listened to by a person, in the same way that the underlying PMAP data would “sound like” the affectivity it contained.

4 PMAPH: PROPOSED PMAP HARMONIC EXTENSIONS

An addition is now proposed to this previous work – the utilization of harmony in PMAP. Before doing this, the basics of PMAP logic and connectionism will be re-iterated from previous papers.

4.1 Music Gates

Three possible PMAP gates were examined in [19] based on AND, OR and NOT logic gates. The PMAP versions of these are respectively: MAND, MOR and MNOT (pronounced “emm-not”), MAND, and MOR. So for a given stream, the PMAP-value can be written as $m_i = [k_i, t_i]$ with key-value k_i and tempo-value t_i . The definitions of the musical gates are (for two streams m_1 and m_2):

$$\text{MNOT}(m) = [-k, 1-t] \quad (1)$$

$$m_1 \text{ MAND } m_2 = [\text{minimum}(k_1, k_2), \text{minimum}(t_1, t_2)] \quad (2)$$

$$m_1 \text{ MOR } m_2 = [\text{maximum}(k_1, k_2), \text{maximum}(t_1, t_2)] \quad (3)$$

These use a similar approach to Fuzzy Logic [23]. MNOT is the simplest – it simply inverts the key mode and tempo – minor becomes major and fast becomes slow, and vice versa. The best way to get some insight into what the affective function of the music gates is it to utilize music “truth tables”, which will be called Affect Tables here. In these, four representative state-labels – based on the PMAP-value system - are used to represent the four quadrants of the PMAP-value table: “Sad” for $[-3, 0]$, “Stressed” for $[-3, 1]$, “Relaxed” for $[3, 0]$, and “Happy” for $[3, 1]$. Table 1 shows the music tables for MAND and MNOT. The columns marked KT-Value refer to the PMAP-value system.

Taking the MAND of two melodies, the low tempos and minor keys will dominate the output. Taking the MOR of two melodies, then the high tempos and major keys will dominate the output. Another perspective: the MAND of the melodies from Moonlight Sonata (minor key) and the Marriage of Figaro Overture (major key), the result would be mainly influenced by Moonlight Sonata. However if they are MOR’d, then the Marriage of Figaro Overture key mode would dominate. The MNOT of Marriage of Figaro Overture would be a minor key version. The MNOT of Moonlight Sonata would be a faster major key version. It is also possible to construct more complex music functions. For example MXOR (pronounced “mex-or”):

$$\begin{aligned} m_1 \text{ MXOR } m_2 = & (m_1 \text{ MAND } \text{MNOT}(m_2)) \\ & \text{MOR } (\text{MNOT}(m_1) \text{ MAND } m_2) \end{aligned} \quad (4)$$

		MAND			MNOT				
State Label 1	State Label 2	KT-value 1	KT-value 2	MAND value	State Label	State Label	KT-value	MNOT value	State Label
Sad	Sad	-3,0	-3,0	-3,0	Sad	Sad	-3,0	3,1	Happy
Sad	Stressed	-3,0	-3,1	-3,0	Sad	Stressed	-3,1	3,0	Relaxed
Sad	Relaxed	-3,0	3,0	-3,0	Sad	Relaxed	3,0	-3,1	Stressed
Sad	Happy	-3,0	3,1	-3,0	Sad	Happy	3,1	-3,0	Sad
Stressed	Stressed	-3,1	-3,1	-3,1	Stressed				
Stressed	Relaxed	-3,1	3,0	-3,0	Sad				
Stressed	Happy	-3,1	3,1	-3,1	Stressed				
Relaxed	Relaxed	3,0	3,0	3,0	Relaxed				
Relaxed	Happy	3,0	3,1	3,0	Relaxed				
Happy	Happy	3,1	3,1	3,1	Happy				

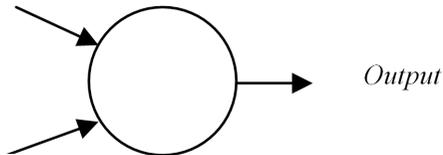
TABLE 1
Music Tables for MAND and MNOT

4.2 Musical Neural Networks

The use of timed pulses in Spiking Neural Networks supports an investigation into using PMAP pulses in Artificial Neural Networks. In particular in a neural network application in which emotion and rhythm are core elements. A form of learning artificial neural network which uses PMAP is now given. These artificial networks take as input, and use as their processing data, pulsed melodies. A musical neuron (muron – pronounced MEW-RON) is shown in Figure 2.

The muron in this example has two inputs, though a muron can have more than this. Each input is a PMAP melody, and the output is a PMAP melody. The weights on the input w_1 and w_2 are each two element vectors which

$$w_1 = [R_1, D_1]$$



$$w_2 = [R_2, D_2]$$

FIGURE 2
A Muron with two inputs.

define a key mode transposition, and a tempo change. A positive R_k will make the input tune more major, and a negative one will make it more minor. Similarly a positive D_t will increase the tempo of the tune, and a negative D_t will reduce the tempo. The muron combines input tunes by superposing the spikes in time – i.e. overlaying them. Any notes which occur at the same time are combined into a single note with the highest pitch being retained. This retaining rule is fairly arbitrary but some form of non-random decision should be made in this scenario (future work will examine if the “high retain” rule adds any significant bias). Murons can be combined into networks, called musical neural networks, abbreviated to “MNN”. The learning of a muron involves setting the weights to give the desired output tunes for the given input tunes. Applications for which PMAP is most efficiently used are those that naturally utilize temporal or affective data (or for which internal and external sonification is particularly important).

4.3 HARMONIC EXTENSIONS FOR PMAP

One key element of music which is not shared necessarily by other areas such as visual display, is the ability for two musical streams to combine in a single stream which has its own gestalt meaning: i.e. a harmony. This allows the potential for auditory display of multiple streams of data. For example a major PMAP stream at a medium tempo would tend to communicate happiness, whereas a minor PMAP stream at a medium tempo would tend to communicate sadness. If the two are synchronized and combined together as a harmony this will probably cause it to sound quite unpleasant, since 4 of the pitches in a major key clash harmonically with 3 of the pitches in a minor key. This harmonic “clashing”, known as dissonance – indicates that the two streams have very different keys and thus - by the original PMAP definition - very different valences.

In Monophonic PMAP the harmonic effects of multiple streams are only detectable by humans who listen to the streams simultaneously. This property was discussed in [20] as being useful for a human commander tapping into the processing of the robot team. An extension is now proposed to PMAP, called Harmonic PMAP, or PMAPh (pronounced “pee-maff”). In PMAPh the arousal of a data stream is still defined by its tempo. However PMAPh also allows for polyphonic data streams, with up to two simultaneous notes occurring at any one time in the data stream. In polyphonic PMAPh data streams the valence of a polyphonic data stream when there are simultaneous pitches is based on the consonance as calculated using Table 2. Consonance in PMAPh is equivalent to Key-value in monophonic PMAP. A low consonance means that the two simultaneous notes clash harmonically (are dissonant), and a high consonance means they sound pleasant together. In monophonic PMAP a high Key-value means more of the melody notes are in a major

Note MIDI Difference in same octave	Name	PMAPh Consonance
0, 4, 7	unison / 3 rd / 5 th	3
5, 9	4 th / 6 th	1
2,3,8,10	2 nd / minor 2 nd / minor 6 th / minor 7 th	-1
1, 6, 11	minor 2 nd / minor 5 th / major 7 th	-3

TABLE 2

Reference table for PMAPh valence

(more positive sounding) key, whereas a low Key-value means more of the notes are in a minor (less positive sounding) key.

So the unison, 3rd and 5th intervals have the highest valence, and the semitone, tritone and major 7th intervals have the lowest valence. The values in the first two columns of this table come from a Genetic Algorithm harmony system developed for a system for emotional expression by algorithmic harmony composition [25]. This table allows the musical logic gates and musical neural networks defined in PMAPh to be used in PMAPh with polyphonic inputs.

The implementation of a musical neuron in [19] and [21] was originally defined as selecting the highest of two pitches when two pitches occurred simultaneously in time on the inputs. In PMAPh it is defined as an interval made up of the highest and lowest simultaneous pitches. It means that a unitary musical neuron (i.e. with unitary weights) can be used to combine two monophonic PMAPh streams into a polyphonic PMAPh stream.

To examine the differences between PMAPh and PMAPh consider the MAND music gate. In a PMAPh MAND gate both input melodies have to be major key for the output to have a major key. The tempo output is the minimum tempo of the two inputs. It implies that both inputs have to have high valence for the output to have high valence, and both inputs have to have high arousal for the output to have high arousal. In PMAPh when the stream is not monophonic, its valence is positively correlated to the consonance defined in Table 2. Table 3 shows the functioning of a MAND gate in PMAPh for just the most consonant and least consonant intervals. (Note that because simple harmonies are more to do with pitch than with rhythm, the following discussion does not go into detail about tempo in PMAPh.)

Figure 3 gives an example application that differentiates PMAPh from monophonic PMAPh.

The circles in Figure 3 are unitary musical neurons (i.e. with weights that do not change the input tunes features). Suppose the 4 inputs are monophonic and *time-quantized*. The time quantized element is important here, as it increases the likelihood that two pitches in two separate melodies could occur at the same time, thus creating harmonies. Suppose first that the tunes on I_1 ,

MAND (maxima only)					
Input Interval 1	Input Interval 2	Consonance 1	Consonance 2	MAND value	Valence
1, 6, 11	1, 6, 11	-3	-3	-3	Low
1, 6, 11	0, 4, 7	-3	3	-3	Low
0, 4, 7	1, 6, 11	3	-3	-3	Low
0, 4, 7	0, 4, 7	3	3	3	High

TABLE 3
MAND table for Maximum Consonances

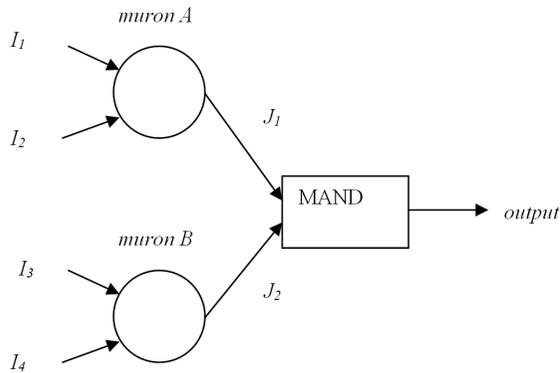


FIGURE 3
A PMAPh demonstration circuit

I_2 , I_3 and I_4 are melodies which have no notes that coincide in time. Then the output of the circuit in Figure 3 will be the same for monophonic PMAP and for PMAPh by the definition of the musical neuron discussed above. However the more notes that coincide in time, the more different the outputs for PMAP and PMAPh will be.

Suppose that all notes coincide in time. In monophonic PMAP the outputs of murons A and B (J_1 and J_2) will work as “pitch gates”, only allowing through the highest of the two pitches, without changing the timing. Then the MAND gate will work as described in previous PMAP work [19] and Table 3 – i.e. if the keymodes of J_1 and J_2 are both major the output key will be major, but if one or more is minor, the output will have a minor key mode. In PMAPh however the murons will output harmonies to J_1 and J_2 , where J_1 contains the intervals made up by simultaneous notes in I_1 and I_2 ; and J_2 made up of I_3 and I_4 notes. In PMAPh the tempo behavior is the same as in PMAP, and since all notes in I_1 and I_2 (and I_3 and I_4) are simultaneous, they must have the same tempo. Hence the final output of the circuit in Figure 3 will have the same tempo as the initial inputs.

However the valence / consonance behavior of a MAND gate in PMAPh is defined using Tables 2 and 3, in a way analogous to the MAND gate in monophonic PMAPh is for Key-mode. In PMAPh the MAND gate gives a very consonant (“pleasant sounding”) output harmony when the two input harmonies are consonant, and a dissonant (“clashing”) output when the two input harmonies are dissonant.

So what is the overall effect of this circuit when the four input melodies are synchronized? If the music table is calculated for Figure 3, it is seen that the output is consonant (pleasant sounding) when the inputs are all the same key-mode, or are pair-wise the same by muron. Affectively this means that the output will be high valence when all the inputs have the same valence, or inputs for both murons have individually the same valence. To describe this more iteratively, suppose all inputs initially have the same valence – the output valence will be high. Now suppose one input deviates from the others, then the output valence will go low. However suppose an input on the same muron as the deviating muron also deviates from the original valence – then the output will go high again.

This is a very different functionality to the monophonic PMAPh version and allows for selective detection of non-pairwise valence deviations. If each input came from a robot and the robots were paired into two teams, then the system would help to monitor that both robot teams were matched in how positive or negative they were “feeling”. The more their valences deviated pair-wise – the lower the output valence would become. For the security robots application [19] this could be used to detect if paired robots were moving too far apart – since being close together would increase the likelihood that they were experiencing the same environment and thus had similar valences. As with monophonic PMAPh this state detection output can be used for further calculations, but furthermore if it is probed and heard by a human being, it will be aurally significant: the more the paired robots deviate from each other, the more “clashing” the PMAPh stream harmonies sound to a human ear. Thus the PMAPh stream can be used for audio display as well as computation.

To demonstrate this in action, a new simulation from the multi-robot system in [19] was run. Note that the PMAPh approach is *not* being used here for the robots to communicate with each other. It is being used to allow each individual robot to process affective information internally. The PMAPh equations used in [19] are shown in equations 5 and 6, but will not be explained in detail here.

$$\text{WEAPON} = \text{DetectOther MAND MNOT}(\text{FriendFlag}) \quad (5)$$

$$\text{MOTOR} = \text{WEAPON MOR MNOT}(\text{DetectOther}) \quad (6)$$

The short melodies which robots use as the building blocks for their PMAPh calculations are called “identives”. The identives used in [19] have been

updated here so as to maximize the harmonic effects. This is because identities only interact harmonically in interesting ways if the “key” notes are in similar positions. Thus if two identities are such that the main notes identifying the differences between C major and C minor never coincide in time, then there will rarely be significant dissonance (or consonance).

The first 500 notes of the WEAPON object are shown in Figure 4 at the end of this paper. Tempo is fixed in this example so as to emphasize the harmonic effects. Security robots 1, 2, 3, and 4 are shown respectively in the figure rows. The simulation is run so that Robots 1 and 2 start off close to each other in the environment, as do robots 3 and 4. Figure 5 shows the consonance of the outputs of the two neurons and the resulting MAND output. In Figure 5, also at the end of the paper, three sections of the MAND output have been marked up. These are the areas with the lowest valence (i.e. highest dissonance). Anyone tapping in to this PMAP stream and listening to it will hear a more dissonant tune. The three sections of the Robot’s PMAP WEAPON stream which cause these periods have been marked up as well in Figure 4. A major key stream in WEAPON causes the robot’s weapon to fire, whereas a minor key stream means the weapon is not firing. It can be seen that during the first marked-up period in Figure 4, Robot 4 is firing and Robot 3 is not. During the second period Robot 1 is firing and Robot 2 is not. During the third period Robot 4 is firing and Robot 3 is not. Thus if the robots are considered as being paired off into teams of 1 and 2, and 3 and 4, then the output of the PMAPh circuit in Figure 3 becomes dissonant when one or more of the two robot teams are mismatched in their weapon behaviour, as discussed earlier.

5 CONCLUSIONS AND FUTURE WORK

This paper has given an overview of the concept of Pulsed Melodic Affective Processing, a complementary approach in which computational efficiency and power are more balanced with understandability to humans; particularly where computation addresses rhythmic and simulated emotion processing. Previous work on PMAP has demonstrated the utilization of musical logic gates, and musical artificial neurons.

A key contribution of PMAP is at the research interface between non-standard computation and human-computer interaction. In normal circuit and network sonification, a probe needs to be placed at the node we desire to sonify, and that data then needs to be fed into a sonification algorithm to be converted into meaningful sounds for the user. However if, in the case of affective circuits and networks, the underlying data uses the PMAP representation, then no sonification algorithm is needed. The data is already in the form of a melody which represents the affective state of the data – in other words the data representation is its own sonification. There are systems which allow the sonification of network data through separate data sonification

algorithms. These systems will take the underlying binary data and protocols, map them onto features, and then play these features. However PMAP is the only data processing and transmission model currently which is its own sonification and requires no significant mapping. This is because PMAP is currently limited to use in affective communications and processing, and such affective states can be represented and computed in many cases by musical data anyway.

Because of music's special ability to turn two separate streams into a single harmonic gestalt with its own affective implications, it is argued that extensions of PMAP into polyphony should be investigated. An extension of PMAP to PMAPh has been proposed, together with an example that demonstrated some significant functionality differences.

There are a significant number of issues to be further addressed with PMAP in general. For example the actual application of the music gates discussed depends on the level at which they are to be utilized. The underlying data of PMAP (putting aside for a moment the PMAP-value representation used for simplicity earlier) is a stream of pulses of different heights and pulse rates. At the digital circuit level this can be compared to VLSI hardware spiking neural network systems [26] or VLSI pulse computation systems. As has been mentioned, a key difference is that the pulse height varies in PMAP, and that specific pulse heights must be distinguished for computation to be done. But assuming this can be achieved, then the gates would be feasible in hardware. It is probable that each music gate would need to be constructed from multiple VLSI elements due to the detection and comparison of pulse heights necessary.

The other way of applying at a low level, but not in hardware, would be through the use of a virtual/simulated machine. So the underlying hardware would use standard logic gates or perhaps standard spiking neurons. The idea of a virtual/simulated machine may at first seem contradictory, but one only needs to think back twenty years when the idea of the Java Virtual Machine would have been unfeasible given current processing speeds then. In 5-10 years current hardware speeds may be achievable by emulation; and should PMAP-type approaches prove useful enough, would provide one possible implementation.

Other issues to be further examined include: is the rebalance between efficiency and understanding useful and practical, and also just how practical is sonification - can sonification more advanced than Geiger counters, heart rate monitors, etc. really be useful and adopted? The valence/arousal coding provides simplicity, but is it sufficiently expressive while remaining simple? Similarly it needs to be considered if a different representation than tempo/key mode would be better for processing or transparency. Furthermore PMAP also has a close relationship to Fuzzy Logic and Spiking Neural Networks - so perhaps it can adapted based on lessons learned in these disciplines.

REFERENCES

- [1] Miranda, E., Adamatzky, A. and Jones, J. (2011) Sounds Synthesis with Slime Mould of Physarum Polycephalum, *Journal of Bionic Engineering* 8, pp. 107-113 Elsevier
- [2] Miranda, E. R., Bull, L., Gueguen, F., Uroukov, I. S. (2009). Computer Music Meets Unconventional Computing: Towards Sound Synthesis with In Vitro Neuronal Networks, *Computer Music Journal*, Vol. 33, No. 1, pp- 9-18.
- [3] Juslin, P. and Sloboda, J. (2001) Music and Emotion: Theory and Research. Oxford University Press, Oxford, UK.
- [4] Bohte, S., Kok, J., La Poutre, H. (1997) Spike-prop: Error-backpropagation in multi-layer networks of spiking neurons. *Neurocomputing*, 48(1-4), 17-37.
- [5] Perrinet, L., Samuelides, M., Thorpe, S. (2004) Coding static natural images using spiking event times: do neurons cooperate? *IEEE Trans Neural Networks* 15 1164-75.
- [6] Chen N, Yu J, Qian H, Ge R, Wang J-H. (2010) Axons Amplify Somatic Incomplete Spikes into Uniform Amplitudes in Mouse Cortical Pyramidal Neurons. *PLoS ONE* 5(7): e11868.
- [7] Miller, J., Young, W. (1996) Simple Pulse Asynchronous State Machines, *Proceedings of International Symposium on Circuits and Systems*, IEEE
- [8] Malatesa, L., Karpouzis, K., Raouzaoui, A. (2009) Affective intelligence: the human face of AI, In *Artificial intelligence*, Springer-Verlag.
- [9] Banik, S., Watanabe, K., Habib, M., Izumi, K. (2008) Affection Based Multi-robot Team Work, In *Lecture Notes in Electrical Engineering*, pp. 355--375.
- [10] Picard, R. (2003) Affective Computing: Challenges, *International Journal of Human-Computer Studies*, Vol. 59, No. 1-2, pp. 55--64.
- [11] Kirke, A., Miranda, E. (2011) Emergent construction of melodic pitch and hierarchy through agents communicating emotion without melodic intelligence, In *Proceedings of 2011 International Computer Music Conference (ICMC 2011)*, International Computer Music Association.
- [12] Juslin, P., Laukka, P. (2004) Expression, perception, and induction of musical emotion: a review and a questionnaire study of everyday listening. *Journal of New Music Research*, vol. 33, pp. 216--237.
- [13] Juslin, P. (2005) From Mimesis to Catharsis: expression, perception and induction of emotion in music, In *Music Communication*, pp. 85--116, Oxford University Press.
- [14] Cohen, J. (1994) Monitoring Background Activities, In *Auditory Display: Sonification, Audification, and Auditory Interfaces*, pp. 499--531.
- [15] Vickers, P., Alty, J. (2003) Siren songs and swan songs debugging with music. *Communications of the ACM*, Vol. 46, No. 7, pp. 87--92.
- [16] Gilfix, M., Couch, A.L. (2000) Peep (the network auralizer): Monitoring your network with sound. *Proceedings of the 14th System Administration Conference (LISA 2000)*, 109--117, New Orleans, Louisiana, USA, 3--8 December 2000. The USENIX Association.
- [17] Kirke, A., Miranda, E. (2012) Application of Pulsed Melodic Affective Processing to Stock Market Algorithmic Trading and Analysis, *Proceedings of 9th International Symposium on Computer Music Modeling and Retrieval (CMMR2012)*, London
- [18] Chafe, C., Leistikow, L. (2001) Levels of Temporal Resolution in Sonification of Network Performance, *Proceedings of 2001 International Conference on Auditory Display*, Finland.
- [19] Kirke, A., Miranda, E. (In Press) "Pulsed Melodic Processing – the Use of Melodies in Affective Computations for Increased Processing Transparency". *Music and Human-Computer Interaction*, S. Holland, K. Wilkie, P. Mulholland and A. Seago (Eds.), London: Springer.
- [20] Kirke, A., Miranda, E. (Submitted) Pulsed Melodic Affective Processing – Musical Structures for Increasing Transparency in Emotional Computation, *Transactions of the Society for Modelling and Simulation International*, SAGE

- [21] Kirke, A. and Miranda, E. (2012). "Pulsed Melodic Affective Processing – Using Music for Natural Affective Computation and Increased Processing Transparency", *Neural, Parallel, and Scientific Computations* 20:227–240, Dynamic Publishers, USA.
- [22] Kirke, A. and Miranda, E. R. (2012). Application of Pulsed Melodic Affective Processing to Stock Market Algorithmic Trading and Analysis. *Proceedings of 9th International Symposium on Computer Music Modeling and Retrieval (CMMR2012)*, London (UK).
- [23] Marinos, P. (1969) Fuzzy Logic and Its Application to Switching Systems, *IEEE Transactions on Computers C*, Vol. 18, No. 4, pp. 343--348.
- [24] Haykin, S.,: *Neural Networks a Comprehensive Foundation*, Prentice Hall (1994).
- [25] Kirke, A, Davies, G. (2013) Open Outcry. *Invited Talk, The Royal Institution*, London, UK
- [26] Indiveri, G.; Chicca, E.; Douglas, R.: A VLSI array of low-power spiking neurons and bistable synapses with spike-timing dependent plasticity, *IEEE Transactions on Neural Networks*, Vol. 17, No. 1, pp. 211--221 (2006).

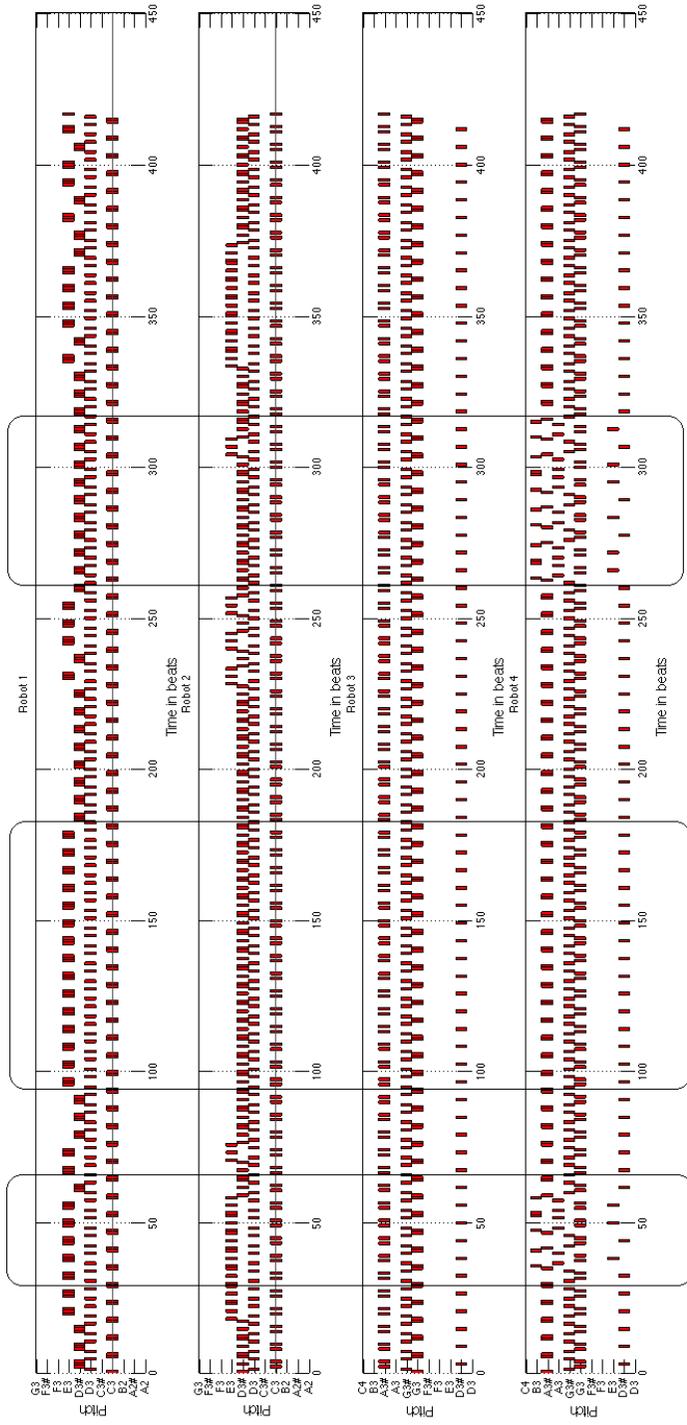


FIGURE 4
MMAP streams in Robots 1 – 4 in Test Run

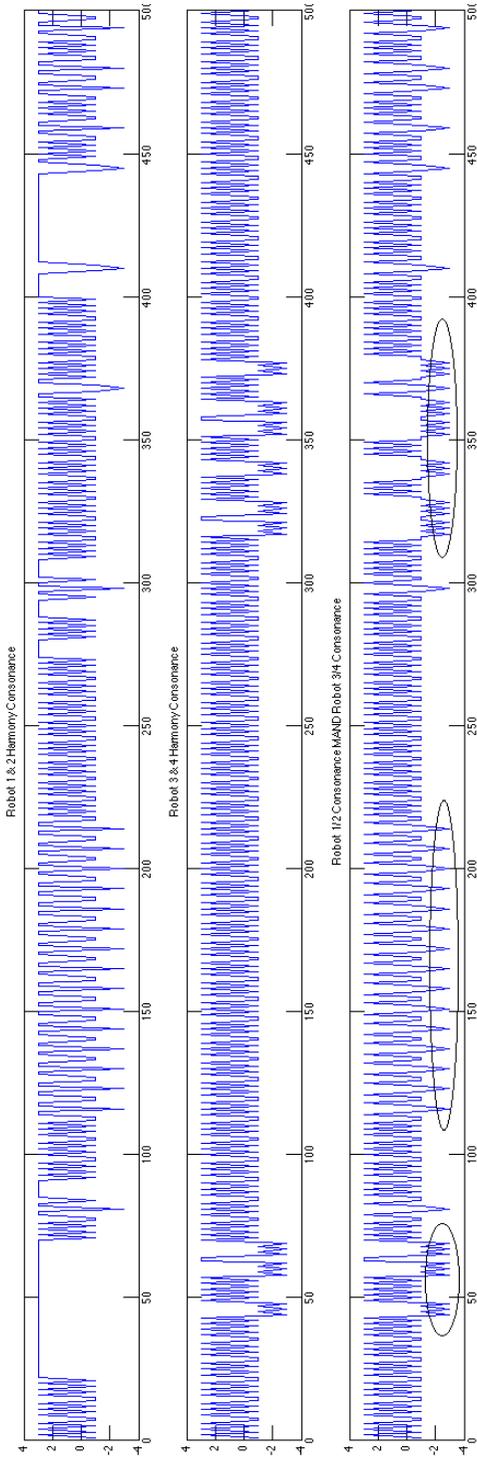


FIGURE 5
Consonance of Robots 1 / 2 and 3 / 4, with MAND of the two Consonances