

2014-05

Pulsed Melodic Affective Processing: Musical structures for increasing transparency in emotional computation

Kirke, Alexis

<http://hdl.handle.net/10026.1/6621>

10.1177/0037549714531060

SIMULATION

SAGE Publications

All content in PEARL is protected by copyright law. Author manuscripts are made available in accordance with publisher policies. Please cite only the published version using the details provided on the item record or document. In the absence of an open licence (e.g. Creative Commons), permissions for further reuse of content should be sought from the publisher or author.

Pulsed Melodic Affective Processing –Musical Structures for Increasing Transparency in Emotional Computation

Alexis Kirke¹, Eduardo Miranda¹

¹ Interdisciplinary Centre for Computer Music Research, School of Humanities, Music and Performing Arts, University of Plymouth, Drake Circus, Plymouth, UK
{Alexis.Kirke, Eduardo.Miranda}@Plymouth.ac.uk

Abstract. Pulsed Melodic Affective Processing (PMAP) is a method for the processing of artificial emotions in affective computing. PMAP is a data stream designed to 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 various emotional dimensions. PMAP data can be generated directly by sound (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. This is 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. Thus PMAP can be used to calculate “feelings” and the result data will “sound like” the feelings calculated. PMAP can be compared to neural spike streams, but ones in which pulse heights and rates encode affective information. This paper illustrates PMAP in a range of simulations. In a multi-agent simulation, initial results support that an affective multi-robot security system could use PMAP to provide a basic control mechanism for ‘search-and-destroy’. Results of fitting a musical neural network with gradient descent to help solve a text emotional detection problem are also presented. The paper concludes by discussing how PMAP may be applicable in the stock markets, using a simplified order-book simulation.

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

1 Introduction

This paper is an investigation into the use of melodies as a tool for affective computation and communication in artificial systems, through a connectionist architecture, a simulation of a robot security team, and a stock market tool. Such an idea is not so unusual when one considers the data stream in spiking neural networks (SNNs). 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. Bohte et al. [1] have developed a back-propagation algorithm for artificial SNNs. Back-propagation is one of the key machine learning algorithms used to develop neural networks that can respond intelligently. It is an established practice for scientists to listen to amplified neural spike trains via loudspeakers as a method of navigating the location of an electrode in the brain [2], and 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: the Musical Instrument Digital Interface (MIDI) [3]. In its simplest form MIDI encodes a melody, which consists of note timing and note pitch information. In this paper we argue that melodies can be viewed as functional and recreational – they can fulfill the function of encoding an artificial emotional state, in a form which can be used in affective computation tasks 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 tempo, and the pulse height encodes pitch.

1.1 Uses and Novelty of PMAP

Before explaining the motivations behind PMAP in more detail, an overview of its functionality will be given. Similarly the novelty of that functionality will be summarized. PMAP provides a method for the processing of artificial emotions that 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, PMAP is novel in that it is a data stream which can be listened to, as well as computed with. Affective state is represented by numbers which are analogues of musical features, rather than by a discrete binary stream. 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.

The encoding of PMAP is designed to provide extra utility – PMAP data can be generated directly by sound and turned directly into sound. Thus, rhythms such as heart rates or key-press speeds can be directly turned into PMAP data; and PMAP data can be directly turned 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 a PMAP data stream represents in the computation engine will be similar to the emotion that a person “listening” to the PMAP-equivalent melody would be. So PMAP can be used to calculate “feelings” and the resulting data will “sound like” the feelings calculated. This is will be clarified over the course of this paper.

Due to the novelty of the PMAP approach, the structure of this paper involves providing multiple examples of the ability of melodies to be used in machine learning and processing. This does not follow the normal approach taken with machine learning, communications or unconventional computation for validation and comparison. For example, the musical neural network demonstration does not include creating a formal description of the network and then rigorously demonstrating it in comparison to previous machine learning methods. This is for two reasons: lack of

space and lack of comparable approaches. It is felt that such a novel approach needs to be shown to be at least relevant in multiple applications, hence there is insufficient room to develop and demonstrate validations for all of the three demonstration areas presented later. Also, there is no basis for comparison. Musical neural network methodologies are almost certainly less efficient than non-melody based computation equivalent. The same can be said of the other examples demonstrated in the paper. The positive argument is that they, and the other PMAP approaches, provide an HCI advantage in addition to their computational ability. There are no other computation approaches that do this, hence no meaningful comparisons are possible without controlled listener evaluation results to determine how well the PMAP streams represent the elements of the affective computations. However before doing these, it is first necessary to investigate if affective melodies are indeed useable in multiple affective applications.

In the previous subsection it was described how this paper is motivated by similarities between MIDI-type structures and the pulsed-processing [4] 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 [5]:

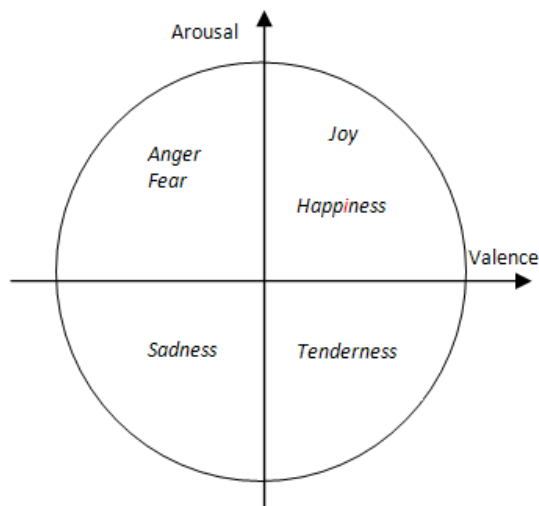
1. Universal and Enhanced Communication: two people who speak different languages are still able to communicate basic states such as happy, sad, angry, and fearful.
2. Internal Behavioral Modification: a person’s internal emotional state will affect the planning paths they take. For example, affectivity 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 such that, 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, etc.
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, in severe pain, and so on.

As a result, affective state processing has been incorporated into robotics and multi-agent systems [6]. Multi-agent systems (MAS) are groups of agents where each agent is a digital entity which can interact with other agents to solve problems as a group, though not necessarily in an explicitly co-ordinated way. What often separates agent-

based approaches from normal object-oriented or modular systems is their emergent behaviour [7]. The solution of the problem tackled by the agents is often generated in an unexpected way due to their complex interactional dynamics, though individual agents may not be that complex.

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 [8]. So representation of simulating 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 [9] two dimensions are used: Valence and Arousal. In this model, emotions can be 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 a high valence, high arousal affective state, and “Stressed” is a low valence high arousal state.

Figure 1: The Valence/Arousal Model of Emotion, from [9]



1.3 Music and Emotion

There have been a number of questionnaire studies done which support the argument that music communicates emotions [10]. Previous research [11] 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 galloping “William Tell Overture” by G. Rossini opens in a major key and is a happy piece – i.e. higher valence. Whereas the first movement of L.V. Beethoven’s Symphony No. 5 is mostly

in a minor key, and although it can be played at the same speed as the William Tell Overture, feels much more brooding and low valence. This is significantly because of its mostly minor key mode. 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: Beethoven's first movement above is often played Allegro (fast). Compare this to his famous piano piece "Moonlight Sonata" – also minor key, but marked Adagio for slow. The piano-piece has a melancholic feel – it is low arousal because of its low tempo as well as low valence.

1.4 Sonification

Sonification [12] 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 [13], sonify activity in computer networks [14] and to give insight into stock market movements [15]. 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 [16]. 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 for 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 (though these capabilities are not examined here).

2 PMAP Representation of Affective State

In PMAP 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 12 possible levels. For example 12 different voltage levels for a low level stream, or 12 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,Db,D,Eb,E,F,Gb,G,Ab,A,Bb,B. For example 1mV could be C, 2mV be Db, 4mV be Eb, etc. We will simply use integers here to represent the notes (i.e. 1 for C, 2 for Db, 4 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,Bb,Eb,C,D,F,Eb,Ab,G,C] (i.e. [1,11,4,1,3,6,4,9,8,1]) would be principally negative valence because it is mainly minor key mode. Whereas [C,B,E,C,D,F,E,A,G,C] (i.e. [1,12,5,1,3,6,5,10,8,1]) 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 [1,12,5,1,3,6,5,10,8,1] was transmitted at a high rate, it would be high arousal and high valence – i.e. a stream representing ‘happy’ (see Figure 1); and at a low rate it would be low arousal and high valence – i.e. a stream representing ‘relaxed’ or ‘tender’ (Figure 1). Whereas if [1,11,4,1,3,6,4,9,8,1] was transmitted at a low pulse rate then it will be low arousal and low valence – i.e. a stream representing ‘sad’.

Note that [1,12,5,1,3,6,5,10,8,1] and [3,12,1,5,1,1,5,8,10,6] 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 first example below, a pulse-based data stream (MIDI) is used directly. However in performing the analysis on PMAP for simulation in the second simulation, it would be convenient to utilize a parametric form to represent the data stream form. The parametric form represents a stream with 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).

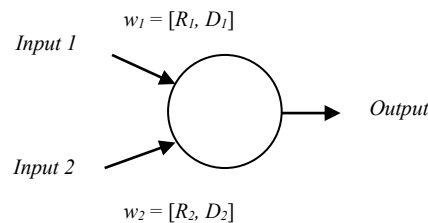
3 Musical Neural Network Example

This first example of the use of PMAP will focus on how PMAP streams can represent non-musical data as part of a machine learning algorithm. It will not be used to demonstrate the sonification abilities of PMAP explicitly but to show that PMAP

can be used for non-musical computations. The example will utilize a form of simple artificial neural network (ANN). ANNs are computational models inspired by the function and structure of neural networks in the biological brain. They are a connected collection of artificial neurons that processes information through an input layer and produce the results of the processing through an output layer. An ANN is usually an adaptive system that changes its behaviour during a learning phase. Many adaption methods utilize a method known as gradient descent [17]. This learning is used to develop a model linking the inputs and outputs so as to create a desired response. In recent years, there has also been work in making the neurons more realistic so they take spike trains, similar to those found in the brain, as input signals. As has been mentioned, these are known as spiking neural networks, and learning algorithms have been developed for SNNs as well. The use of timed pulses in spiking neural networks supports an investigation into PMAP pulses in artificial neural networks. In particular, a neural network application in which emotion and rhythm are core elements. One such example is now presented.

A form of learning artificial neural network which uses PMAP is first described. 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 two-element vectors which define a key mode transposition, and a tempo change respectively. A positive R_k will transpose more input tune notes into a major key mode, and a negative one will transpose more input notes into a minor key mode. 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 superimposing 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).

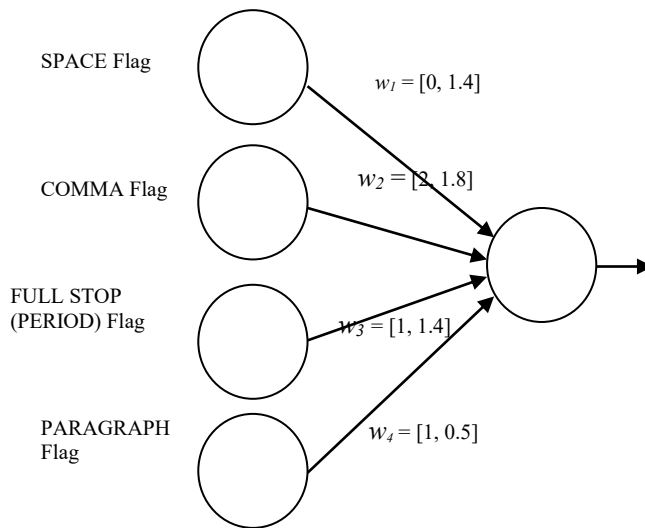
Figure 2: A Muron with two inputs with weight vectors w_1 and w_2 respectively



One such system will now be proposed for the estimation of affective content of real-time typing. The system is inspired by research by the authors on analysing QWERTY keyboard typing. This approach is based on the way that piano keyboard playing can be computer-analyzed to estimate the emotional communication of the piano player [18]. It has been found by researchers that the mood a musical performer is trying to communicate effects not only their basic playing tempo, but also the structure of the hierarchical patterns of the musical timing of their performance [19]. Similarly we propose that a person's mood will affect not only their typing rate [18], but also their relative word rate and paragraph rate, and so forth.

In [18] a real-time system was developed to analyse local tempo of typing and estimate affective state. The MNN/PMAP version demonstrated in this paper is not real-time, and does not take into account base typing speed: it focuses on relative rates of offline pre-typed data. These simplifications are for the sake of expedient simulation and experiments. However it does implicitly analyse hierarchies of tempo patterns, which the system in [18] did not.

Figure 3: Four input MNN for Offline Text Affective Analysis with final learned weight values



The proposed architecture for the emotion estimation is shown in Figure 3. It has 2 layers known as the Input and Output layers. The input layer has four neurons – which generate notes. The idea of these 4 inputs is they represent 4 levels of the timing hierarchy in language. The lowest level is letters, whose rate is not measured in the demo. These letters make up words, which are usually separated by a space. The words make up phrases. In an ideal system the syntax hierarchy would be used to define phrases. However for simplification here, an approximation is made using commas. This will reduce the accuracy of the results but allows for a simpler

demonstration of the learning capacity of the network. So, phrases will be defined here as being punctuated by commas. These phrases make up sentences (separated by full stops), and sentences make up paragraphs (separated by a paragraph end). So the tempo of the tunes output from these four neurons represent the relative word-rate, phrase-rate, sentence-rate and paragraph rate of the text. Note that for data from an internet based messenger application, the paragraph rate will represent the rate at which messages are sent. Every time a space character is detected, then a note is output by the SPACE Flag. If a comma is detected then a musical note is output by the COMMA Flag, if a full stop/period then the FULL STOP (PERIOD) Flag generates a note, and if an end of paragraph is detected then a note is output by the PARAGRAPH Flag.

The “carrier melodies” used in the input layer are a series of constantly rising pitches. The precise pitches in these melodies are not important – it is having a variety of pitches at a neutral tempo, so that they can be transformed through different affective states. The desired output of the MNN will be a tune which represents an affective estimate of the text content. A happy tune means the text structure is happy, like wise a sad tune means the text is sad. Normally, neural networks are trained using a number of methods, most commonly some variation of gradient descent, a type of algorithm which attempts to change the network parameters so as to lower the difference between the actual output and the desired output. A gradient descent algorithm is used here. w_1, w_2, w_3, w_4 are all initialised to $[0,1]$ = [Key mode sub-weight, Tempo sub-weight]. Thus, initially the weights have no effect on the key mode, and multiply tempo by 1, i.e., they have no effect over-all. The final learned weights are also shown in Figure 3. Note, in this simulation actual tunes are used. In fact the Matlab MIDI toolbox is used.

To train the neural network, rather than using live typing, a series of pre-typed documents were sourced from the internet. This is possible because it is not the character typing rate but the relative rates in the text hierarchy that are being utilized. The documents are a record of relative typing rates. The documents in the training set were selected from internet posted personal or news stories which were clearly summarised as sad or happy stories. 15 sad and 15 happy stories were sampled. The happy and sad tunes are defined respectively as the targets: a tempo of 90 BPM and a major key mode, and a tempo of 30 BPM and a minor key mode.

At each step the learning algorithm selects a training document. Then it selects one of $w_1, w_2, w_3,$ or w_4 . Then the algorithm selects either the key mode or the tempo sub-weight. It then performs a single one-step gradient descent based on whether the document is defined as Happy or Sad (and thus whether the required output tune is meant to be Happy or Sad). The size of the one step is defined by a learning rate, set separately for tempo and for key mode. The key mode was estimated using a modified key finding algorithm [20] which gave a value of 3 for maximally major and -3 for maximally minor. The tempo was measured in beats per minute. Before training, the initial average error rate across the 30 documents was calculated. The initial average error was 3.4 for key mode, and 30 for tempo.

After the 1920 step iterations of learning the average errors reduced to 1.2 for key mode, and 14.1 for tempo. These results are described in more detail in Table 1, split by valence (happy or sad). Note that these are in-sample errors for a small population of 30 documents. However what is interesting is that there is clearly a significant error

reduction due to gradient descent. This shows that it is possible to fit the parameters of a musical combination unit (a muron) so as to combine musical inputs and give an affectively representative musical output, and address a non-musical problem. As a practical example, this system could be embedded as music into messenger software to give the user affective indications through sound.

It can be seen in Table 1 that the mean tempo error for Happy documents (target 90 BPM) is 28.2 BPM. This large error is due to an issue similar to linear separability in normal ANNs [17], though it is beyond the scope of this paper to go into the details of the separability problem. One way of understanding it is to consider that the muron is approximately adding tempos linearly. So when it tries to learn two tempos it will focus on one more than the other – in this case the Sad tempo. In standard ANNs, the linear separability problem can be overcome by adding another layer of neurons after the input layer. The difficulty that arises then is that gradient descent becomes more complex. This problem has been solved in standard ANNs using the Back Propagation algorithm mentioned earlier. Hence adding a hidden layer of murons may well help to reduce the Happy error significantly if some form of Back Propagation can be developed for musical neural networks, in the same way it has been developed for spiking neural networks.

Table 1: Mean Error of MNN after 1920 iterations of gradient descent

| | Key Target | Mean Error | Key Tempo (BPM) | Target | Mean Tempo Error (BPM) |
|-------------------|------------|------------|-----------------|--------|------------------------|
| Happy Docs | 3 | 0.8 | 90 | | 28.2 |
| Sad Docs | -3 | 1.6 | 30 | | 0 |

So having demonstrated the use of music streams to model a non-musical problem, what benefits can the use of PMAP give us for this particular application? A key benefit of PMAP is the insight it can give to the internal functioning of an affective circuit, using a simple sonic approach. To gain insight into the internal functioning of the above MNN one simply places a sonic probe at a point in the network one wishes to analyse, and the results can be auralized. In this case the situation is simpler as the neural network only has two layers, so analysis would be simple even without PMAP. Therefore, as was mentioned at the beginning of the section, the above example is used primarily to demonstrate the way that PMAP streams can represent and adapt to non-musical data. However as was discussed earlier, having more than 2 layers in a musical neural network may be helpful. It has been found that understanding the functioning of the middle layer in standard 3 layer neural networks is not always simple [17]. So if a 3 layer PMAP approach could be developed, as we hope to demonstrate in future work, then the extra transparency of the PMAP auto-sonification may prove to be more helpful.

4 Multi-Agent Simulation

Another simple application is now introduced. A software multi-agent system (MAS) is used here to model a multi-robot system [21]. It provides a method for examining the interactions in the initial design of a robot team, without the money or time-investment needed to test with hardware. The below describes a multi-robot

security system being simulated as a software MAS. Like many software multi-agent simulations it is highly simplified in its functionality compared to an actual physical system.

Why would a multi-robot security system need an affective state? One function of affective states in biological systems is that they provide an additional motivation to action when the organism is damaged or in an extreme state [22]. For example an injured person will still try to defend themselves or escape if attacked such that they are unable to think clearly in a rational way. An affective subsystem for a robot who is a member of a security team is now examined; one that can ‘kick in’ or over-ride if the primary decision making functions are damaged or deadlocked. A group of mobile security robots with built-in weapons are placed in a potentially hostile environment and required to search the environment for intruders, and, upon finding intruders, to move towards them and fire on them. The PMAP affective sub-system shown below is designed to keep friendly robots apart (so as to maximize the coverage of the space), to make them move towards intruders, and to make them fire when intruders are detected. To achieve this, a simple circuit of PMAP gates – shown in Figure 4 - is used. These gates are also introduced below.

Note that the PMAP 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. It is assumed that the robot has two layers of processing: a more complex symbolic layer used when the robot is fully functional, and, in case that layer is damaged, a simpler parallel lower-level layer. The use of an affective processing “back-up” layer echoes that found in biological organisms, as mentioned earlier. It also provides for a continuous or fuzzy response to input data, whereas simply using a low-level logic layer may be constrained to basic on / off processing. Finally, it is useful for a robot security system to be able to provide knowledge of its affective state processing: the PMAP streams, as opposed to simple real-numbered representations of robot emotional state, can be made audible to give a user quick, simple and eyes-free insight into the function of the various elements of the robots’ internal modules – perhaps at the design or maintenance stage. The audibility of PMAP could also be of use during live operation, for example if the team’s human commander is in the field and needs to keep hands and eyes free to deal with intruders. The commander can have the PMAP streams of the security robots’ affective states sent to a radio ear-piece. This would allow eyes-free monitoring of the team state. Normally the provision of such eyes-free insight would require a sonification algorithm to be applied to the area of the robot which the user wished to analyse. However PMAP streams, by their very nature, encode that information as music already.

4.1 Music Gates

Three possible PMAP gates will now be examined based on AND, OR and NOT logic gates. The PMAP versions of these are respectively: MAND, MOR and MNOT (pronounced “emm-not”). 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 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 2 shows the music tables for MOR and MNOT.

Table 2: Music Tables for MOR and MNOT

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

Taking the MAND of two melodies, low tempos and minor keys will dominate the output. Taking the MOR of two melodies, high tempos and major keys will dominate the output. To give another perspective: the MAND of the melodies from Beethoven’s Moonlight Sonata (minor key) and the William Tell Overture (major key), the result would be mainly influenced by Moonlight Sonata. However if they are MOR’d, then the William Tell Overture key mode would dominate. The MNOT of the William Tell 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”):

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

The actual application of these music gates 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 above) 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 [24] 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. 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.

As mentioned, PMAP gates function in ways similar to fuzzy logic. To analyze a fuzzy logic circuit in an eyes-free way would normally require a probe to be inserted at points in the Logic circuit and that probe information to then be translated into sound through a sonification algorithm. However circuits built from the above Music Gates can be analyzed by simply listening to the data stream. At any point in the circuit an audio probe can be inserted to give a sense of the affective data at that junction in an audible way.

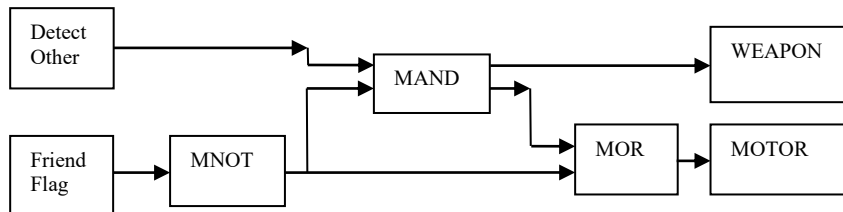
4.2 MAS Simulation of a Multi-Robot System

The modules for the PMAP affective subsystem are shown in Figure 4: “DetectOther”, “FriendFlag”, “MOTOR”, and “WEAPON”. “DetectOther” emits a regular minor mode melody; then every time another agent (human or robot) is detected within firing range, a major-key mode melody is emitted. This is because detecting another agent means that the robots are not spread out enough if it is a friendly, or it is an enemy if not. “FriendFlag” emits a regular minor key mode melody except for one condition. Other authorized ‘friends’ are identifiable (visually or by RFI) and when an agent is detected within range - if it is an authorized friendly – this module emits a major key mode melody. The “MOTOR” unit, when it receives a major key note, moves the robot forward one step. When it receives a minor key note, it moves the robot back one step. The “WEAPON” unit, when it receives a minor key note, fires one round. The weapon and motor system is written symbolically in equations (5) and (6):

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

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

Figure 4: Affective Subsystem for Security Multi-robot System



Calculating (5) and (6), using Equations (1) and (2) from earlier, gives the theoretical results in Table 3. Only five rows of the table are shown as the other states will not occur in the real-world situation. The 5 rows have the following interpretations: (a) if alone continue to patrol and explore; (b) If a distant intruder is detected move towards it fast and start firing slowly; (c) If a distant friendly robot is detected move away so as to patrol a different area of the space; (d) If enemy is close-by move slowly (to stay in its vicinity) and fire fast; (e) If a close friend is detected move away. This should mainly happen (because of row c) when the robot team are initially deployed and they are bunched together, hence slow movement to prevent collision.

To test in an MAS, four security robots are used, implementing the PMAP-value processing described earlier, rather than having actual melodies within the processing system. The security robots using the PMAP affective sub-system are called “F-Robots” (friendly robots). The movement space is limited by a border and when an F-Robot hits this border, it moves back a step and tries another movement. Their movements include a perturbation system which adds a random nudge to the robot movement, on top of the affectively-controlled movement described earlier. The simulation space is 50 units by 50 units. An F-Robot can move by up to 8 units at a time, backwards or forwards. Its range (for firing and for detection by others) is 10 units. Its PMAP minimum tempo is 100 beats per minute (BPM), and its maximum is 200 BPM. These are encoded as a tempo value of 0.5 and 1 respectively. Stationary unauthorized intruders are placed at fixed positions (10,10), (20,20) and (30,30).

The F-robots are placed at initial positions (10,5), (20,5), (30,5), (40,5), (50,5), i.e., they start at the bottom of the space. The system is run for 2000 movement cycles – in each movement cycle each of the 4 F-Robots can move. 30 simulations were run and the average distance of the F-Robots to the immobile intruders was calculated. Also the average distances between F-Robots was calculated. These were done with a detection range of 10 and a range of 0. A range of 0 effectively switches off the musical processing. The results are shown in Table 4. It can be seen that the affective subsystem keeps the F-Robots apart encouraging them to search different parts of the space. In fact it increases the average distance between them by 72%. Similarly the music logic system increases the likelihood of the F-Robots moving towards intruders. The average distance between the F-Robots and the enemies decreases by 21% thanks to the melodic subsystem. These results are fairly robust, with coefficients of variation between 4% and 2% respectively across the results. Figures 5 and 6 show two simulation runs, with each F-Robots’ trace represented by a different colour, and each fixed intruder shown by an “X”.

It was found that the WEAPON firing rate had a very strong tendency to be higher as enemies were closer. The maximum tempo of Robot 1’s firing (just under maximum tempo 1) or firing rate is achieved when the distance is at its minimum. Similarly the minimum firing rate occurs at distance 10 (the detection range) in most cases. In fact the correlation between the two is -0.98 which is very high. This shows that PMAP allows similar flexibility to fuzzy logic, in that the gun rate is controlled fuzzily from minimum to maximum.

Table 3: Theoretical Effects of Affective Subsystem

| <i>Detect Other</i> | <i>Friend Flag</i> | <i>Detect Other-Value</i> | <i>Friend Flag-Value</i> | <i>MNOT (Friend Flag)</i> | <i>MAND Detect Other</i> | <i>WEAPON</i> | <i>MNOT (Detect Other)</i> | <i>MOR WEAPON</i> | <i>MOTOR</i> |
|---------------------|--------------------|---------------------------|--------------------------|---------------------------|--------------------------|---------------|----------------------------|-------------------|---------------|
| Sad | Sad | -3,0 | -3,0 | 3,1 | -3,0 | inactive | 3,1 | 3,1 | Fast forwards |
| Relaxed | Sad | 3,0 | -3,0 | 3,1 | 3,0 | Firing | -3,1 | 3,1 | Fast forwards |
| Relaxed | Relaxed | 3,0 | 3,0 | -3,1 | -3,0 | Inactive | -3,1 | -3,0 | Slow back |
| Happy | Stressed | 3,1 | -3,1 | 3,0 | 3,0 | Firing | -3,0 | 3,0 | Slow forwards |
| Happy | Happy | 3,1 | 3,1 | -3,0 | -3,0 | inactive | -3,0 | -3,0 | Slow back |

Table 4: Results for Robot Affective Subsystem

| Range | Avg Distance between F-Robots | Std Deviation | Average Distance of F-Robots from an Intruder | Std Deviation |
|--------------|--------------------------------------|----------------------|--|----------------------|
| 0 | 7.6 | 0.5 | 30.4 | 0.3 |
| 10 | 13.1 | 0.5 | 25.2 | 0.4 |

How might a user utilize the PMAP streams to learn about the robot’s behavior sonically? Suppose the user wants to analyze the behavior of the lower MOR gate shown in Figure 4. Perhaps they want to re-design the robot affective system and want to test the MOR gate gives them the result they want based on certain inputs. Or it may be because they think there is a fault in the system because it is damaged, and want to test this part of the circuit. In a PMAP system the user could insert an audio probe and listen to the output of the MOR gate. As has been mentioned, in this particular simulation the PMAP-value model is being used. Hence, unlike in the previous musical neural network simulation, for convenience it is real-number representations of the musical state which are being transmitted through the circuit. However these can easily be turned into sound in this simulation because the two numbers being transmitted represent Key mode and Tempo. Thus if each of the 4 robots is assigned a distinctive motif and it is modulated with any tempo and key-value readings from within the circuit, a good sense of what someone using a music probe would hear in a real PMAP version of the robot circuit can be simulated.

Motives designed to identify a module, agent, etc. will be called “Identive”. The identives for the 4 robots were selected as:

1. [1,2,3,5,3,2,1] = C,Db,D,E,D,Db,C
2. [3,5,6,7,6,5,3] = Db,Eb,F,Gb,F,Eb,Db
3. [6,7,9,1,9,7,6] = F,Gb,Ab,C,Ab,Gb,F
4. [7,9,1,6,1,9,7] = Gb,Ab,C,F,C,Ab,Gb

Figure 5: Simulation of security robots without Pulsed Melodic Affective Processing

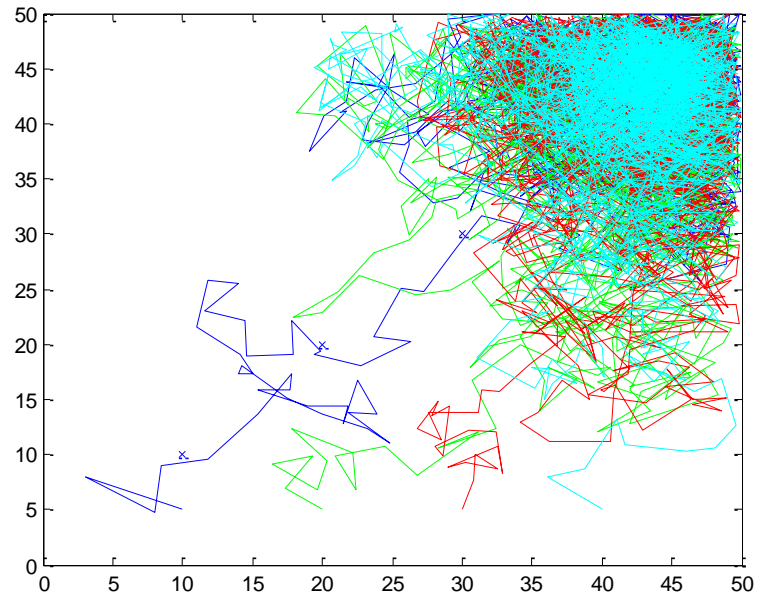
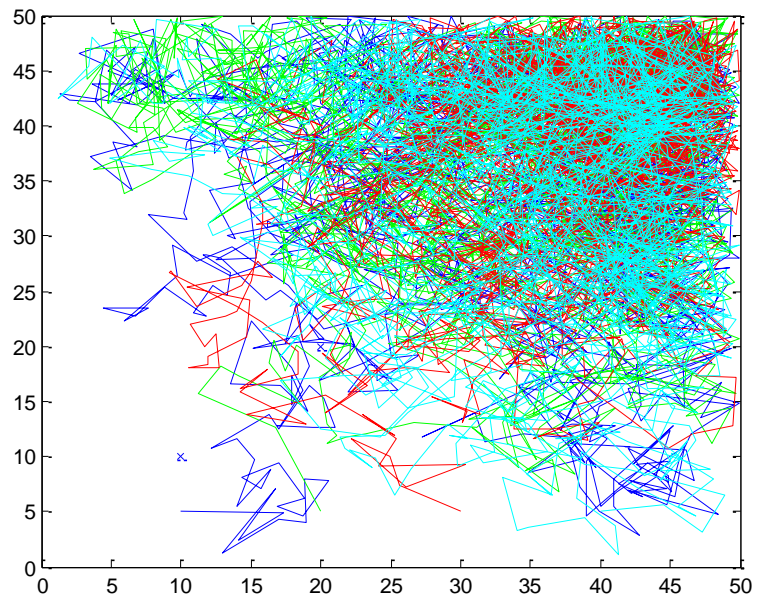
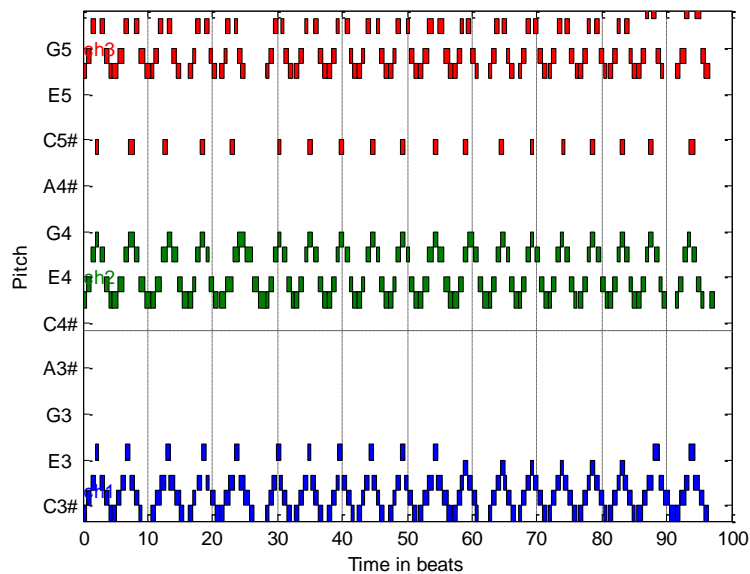


Figure 6: Simulation of security robots with PMAP system and range of 10 units, showing a better search dispersion as a result of the PMAP compared to Fig.5



Placing a simulated audio probe at the output of the MAND gate in Figure 4, involves transforming these motifs based on the PMAP-values of tempo and key-mode found on the MAND output, into musical motifs. Figure 7 shows the first 400 notes of robots 1 to 3's MAND output in the simulation, in piano roll notation. For plotting clarity, the different MAND units have been octave transformed (the lowest is robot 1, the highest robot 3). It was found that the octave separation used for visual clarity in Figure 7 actually helped with aural perception from the simulated audio probe. It was found that more than 3 robots were not really individually perceivable when listened to together. It was also found that transforming the tempo minimums and maximums to between 100 and 200 beats per minute and quantizing by 0.25 beats seemed to make changes more perceivable as well.

Figure 7: A plot of 500 notes in the MAND gate output of robots 1 to 3 (octave separated).



The tempo changes, which are visible in all three PMAP data streams in the Figure, were found to be independently audible in informal listening tests by the authors. So the output of the MAND gate for all three robots could be heard by directly listening to the processing stream. What could also be heard was that the top two data streams (robot's 2 and 3) were more in synchronisation than the bottom one (robot 1). The key mode was slightly harder to discern and required more concentration. This MAND gate output also drives the weapon module. Listening to the audio output it became clear from the start that some of the robots were firing and some were not. This is audible as there was dissonance created by the different key modes (major key mode means weapon firing, minor means not firing). Listening more closely, the point at which robot 1 stopped firing (around beat 58 in Figure 7) was audible. More clearly audible was the point at which robot 3 started firing (around beat 85). Thus the state

of the robot team’s weapons, and the individual robots, was to a degree discernible from their data stream. A fuzzy logic system could have been used to design this robot system, and then the streams of fuzzy data converted into sound using an external sonification algorithm. The key difference here is that the data stream is being heard not sonified – the data stream is its own sonification.

This is the main contribution of the PMAP approach to the field of sonification research: PMAP is the first data representation for processing which is its own sonification. In the non-simulated version of this circuit, if a user wanted to investigate the behaviour of the circuit at different points, for example the output of the MNOT gate or the MOR gate, they could simply place their probe there and hear the data stream directly without the need for a sonification algorithm. Note that this has been demonstrated here on a relatively simple circuit; as affective circuits grow increasingly complex PMAP’s utility can grow as gaining insight into a circuit’s inner functionality becomes more of an issue without a meaningful probing approach.

Of course the complexity of real-life problems in security and military robots goes far beyond the highly simplified examples presented in this paper and requires large state spaces with exponential number of transitions between them. Such systems are usually based on formal systems that allow formal verification, i.e., the robot will behave as expected in all conditions, and on methods for providing bounded computation and achieving tractability. Furthermore military robots, especially weapon systems, are sometimes time-critical applications, which require extremely fast response times. Thus the above PMAP simulation can only be viewed as a very initial demonstration of a potential application of PMAP in multi-robot systems. However as processing speeds increase, and the tools of affective computing expand in their sophistication, it would seem that further work on developing PMAP could lead to tractable solutions for hardware multi-robot systems.

An extension of the above robot system is to incorporate rhythmic biosignals from modern human-worn security suits [25][26]. For example if “BioSignal” is a tune generating module whose tempo is a heart rate reading from a security body suit, and whose key mode is based on EEG valence readings from the reader, then the MOTOR system could become:

$$MOTOR = WEAPON MOR MNOT(DetectOther) MOR MNOT(BioSignal) \quad (7)$$

The music table for (7) would show that if a (human) friend is detected whose biosignal indicates positive valence, then the F-robot will move away from the friend to patrol a different area. If the friendly human’s biosignal is negative then the robot will move towards them to aid them.

5 Affective Market Mapping

An example of PMAP will now be given in an area where sonification has been more extensively studied: the stock market. The key difference in the approach below to previous studies, e.g. [31, 32], is that although it can be used purely as a form of market sonification, this sonification’s musical notes can potentially be used directly to make calculations about the stock market. For example, in a simple form of algorithmic trading approach which will be described.

There are 3 elements which suggest PMAP may have potential in the stock markets: a simple market-state mapping (described below), the incorporation of trader, client and news article “sentiment” into what is an art as well as a science, and a natural sonification for eyes-free HCI in busy environments. The Affective Market Mapping (AMM) involves mapping stock movements onto a PMAP representation. Such a mapping would allow PMAP processing to interact with stock market data and be used for algorithmic trading. One mapping that was initially considered was a risk / return mapping – letting risk be mapped onto tempo, and return be mapped onto key mode. Thus a higher risk would be represented by a more highly aroused affective state, and a high return by a more positive affective state. However this does not give an intuitively helpful result. For example it might imply that a high arousal high valence stock (high risk / high return) is “happy”. However this entirely depends on the risk profile of the investor / trader. So a more flexible approach – and one that is simpler to implement - for the AMM is:

1. Key mode is proportional to Market Imbalance.
2. Tempo is proportional to Number of Trades per Second.

These can refer to a single stock, a group of stocks, or a whole index. Consider a single stock S . The Market Imbalance Z in a time period dT is the total number of shares of buying interest in the market during dT minus the total number of shares of selling interest during dT . This information is not publically available, but can be approximated. For example it can be approximated as in [33]: the total number of buy-initiated sales minus the total number of sell-initiated trades (normalized by the Average Daily Volume for S); with a trade is defined as buy-initiated if it happens on an uptick in the market price of stock S , and sell-initiated if it happens on a downtick (the “tick algorithm”). If there are as many buyers as sellers in stock S then it is balanced and its market imbalance Z will be 0. If there are a large number of buyers and not enough sellers (e.g. in the case where positive news has been released about the stock) the imbalance will become positive.

To generate a melody from a stock, simply take a default stream of non-key notes at a constant or uniformly random rate; and every time there is a trade, add a major key note for a buy-initiated trade and a minor key note for a sell-initiated trade. So for example, if a stock is being sold off rapidly due to bad news, it will have a negative market imbalance and a high trading rate – which will be represented in PMAP as a minor key and high tempo. To western listeners this represents low valence and high arousal, often labelled as “angry” or “fearful”. Stocks trading up rapidly on good news will have a major key and high tempo (“happy”), stocks trading up slowly in a generally positive market will have a low tempo and high valence (“relaxed”). The resulting stream from the AMM matches in the PMAP encoding what many would consider their affective view of the stock, and as such would sound like that to many as well.

5.1 Simulation

To examine a simple processing usage of the AMM and PMAP a basic algorithmic trading system will be implemented. Algorithmic trading has become extremely

prominent in the markets in the last few years [34]. The field of behavioral finance has highlighted the importance of emotions in finance and markets [35]. However we are not aware of any such work which focuses on affectivity. To examine this approach, a simple stock market order book simulation has been developed. The market contains a single stock whose initial price is \$100. Orders arrive at the market at a constant rate of one every 10 minutes. The stock has an Average Daily Volume of around 40000 shares. Each trade can be a buy or sell order with a probability p of being a buy order and $1-p$ of being a sell-order. The order book can contain up to 30 buy orders and 30 sell orders. Each order is uniformly randomly sized. The market price $p(t)$ evolves based on whether an order is a buy or sell order, the order size, and a price volatility parameter.

$$p(t) = p(t-1) + \text{priceDriftFactor} \cdot \text{orderSize} \cdot \text{orderPrice} / \text{ADV} - \text{volatility} + 2 \cdot r \cdot \text{volatility} \quad (8)$$

The level at which a simulated order is priced is the market price $p(t)$ with a certain deviation of % size defined by a parameter priceFluctFactor . Once the book has filled up with arriving orders, new orders overwrite the oldest ones. Although in the simulation it is known precisely whether the order is a buy or sell order, the tick algorithm is still used to estimate the order side for the AMM. The accuracy of this estimation will depend on the size of the random price fluctuations in orders and the market price volatility – i.e., the higher the volatility and fluctuation parameters, the less accuracy the tick algorithm will exhibit. For the simulation detailed here volatility was set to 0.02, priceDriftFactor to 0.005 and priceFluctFactor to 0.001. This led to the tick algorithm being on average, about 75% accurate. (In other words about 75% of orders were correctly classified.) If volatility is increased to 0.005, the accuracy drops to around 60%.

To see how this model functions with the AMM, consider the prices of a month's worth of trading shown on the left hand side of Figure 8, where maximum order size is 1000 shares. This month is a "neutral" month – in other words, the probability of a buy-order is equal to the probability of a sell-order. The right hand side of Figure 8 shows a month where there is a constant probability of 70% of a sell-order arriving, and of 30% of a buy-order arriving. The left side of Figure 9 shows a measure representing valence calculated for this "selling month", calculated using the AMM. Note that in the following discussions valence and arousal are used instead of key mode and tempo. This is equivalent to the parametric version of PMAP used in the earlier MAS, but differs in presentation. Such a presentation allows the reader to more simply see the affective relationships in the stock market data. It also avoids the complication of explicitly constructing and analysing a melodic stream process. Higher valences are approximately concurrent with a more major key mode, and the lower valences to a more clearly minor key mode.

The first thing to observe is that the valence in the left hand side of Figure 9 is usually negative, with a mean valence of -0.34. There are 5 sections where it goes above 0, but this is consistent with the existence of local maxima in the globally falling stock price in the right hand side of Figure 8.

Figure 8: Stock Price in dollars in a “Neutral” Month; and in a “Selling Month” (x-axis is simulation time steps)

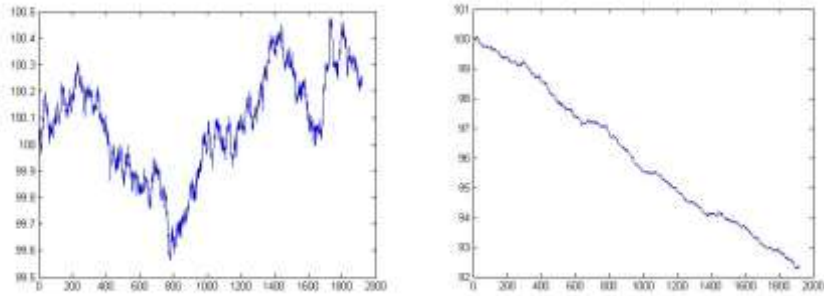


Figure 9: Valence of Stock Price in the “Selling” Month; and Price during “Event” (x-axis is simulation time steps)

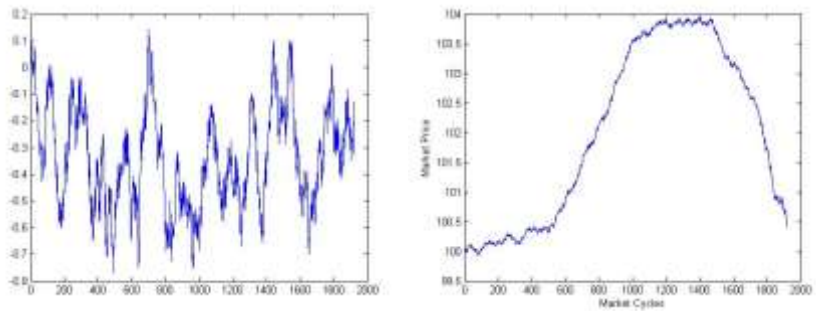
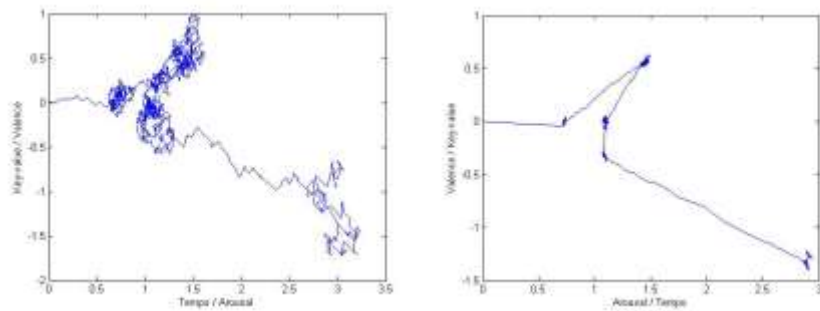


Figure 10: Affective Market Model of stock event; and mean Affective Market Model averaged over 50 stock events



The right hand of Figure 9 shows a market event which begins with a relative relaxed trading in the stock just above \$100, followed by a rapid rise in the stock price due to an increase in buy order probability. This is followed by another period of

stable price trading just below \$104, then for some reason the stock starts to fall with increasing rapidity back to just above \$100. This is done by setting the buy probabilities to 0.5, 0.75, 0.5, 0.25 respectively; and setting average order amounts to 1000, 2000, 1500, and then during the selling period to 1500 and then 4000. It is much clearer to see patterns of behaviour if valence is plotted against arousal as in Figure 10. Looking now at how this is reflected in the Affective Market Model, we can observe the left hand of Figure 10. To clarify this further an averaged version is shown in the right hand side of Figure 10, averaged over 50 runs.

The stock begins at the far left of the diagram with a low arousal and neutral valence due to the slow build of the order book (which starts from empty). One can then observe at least 5 “emotional regimes” that the market moves through, as the arousal/valence line is followed by eye moving from the far left to the far right of the diagram:

1. “Relaxed” – after the arousal builds up there is a regime around 0.02 arousal at the left of the diagram.
2. “Joyful”/“Excited” – this is the region of maximum valence / key-value and with significantly increase arousal / tempo, during which the stock price is rising more rapidly.
3. “Happy” – The market rise is slowing down as it approaches \$104
4. “Sad” – The market starts to go down slowly.
5. “Angry”/“Fearful” – At around \$102.50 the stock begins to fall rapidly.

An interesting element to observe concerning these regimes is that they are audible since if sound is played with the relevant key-value and tempo the music will (for western listeners) have the affective communication (approximately) of: “Relaxed”, “Excited”, “Happy”, “Sad”, and “Fearful” [36].

To examine how the AMM might be used in algorithmic trading, consider a simple rule:

If keymode > trigger then buy stock quantity proportional to tempo
If keymode <-trigger then sell short the stock with quantity proportional to tempo

To simplify an experiment with a rule like this, the valence-arousal space will be utilized to approximately represent the keymode-tempo space:

If valence > triggerV then buy stock quantity proportional to tempo
If valence <-triggerV then sell short the stock with quantity proportional to tempo

Using this rule and the above market model, with a valence trigger value of 0.1, trading simulations were run. When the trigger kicked in a stock quantity of $50 \times \text{Arousal}$ was traded. So an arousal of 0 would lead to a trade of 0 shares, an arousal of 2 would lead to a trade of 100 shares. The results are shown in Table 5, each cell gives the average profit from 50 experiments. Results are shown with the arousal-based trading sizes as well.

Table 5: Profits for the Strategy

| triggerV | Arousal-based Trade Size? | Trigger Strategy Profit | Random Strategy Profit |
|----------|---------------------------|-------------------------|------------------------|
| 0.6 | Y | \$4,515 | \$671 |
| 0.6 | N | \$9,109 | -\$244 |
| 0.4 | Y | \$21,618 | \$17 |
| 0.4 | N | \$18,333 | \$76 |
| 0.1 | Y | \$15,609 | \$1,843 |
| 0.1 | N | \$18,235 | -\$103 |

The Random strategy trades with approximately the same frequency as the Trigger strategy but at randomized times and random order sizes. It can be seen that the Trigger strategy outperforms the Random strategy, and that a full valence / arousal strategy (where trade size is based on arousal) outperforms a valence-only strategy. Another interesting element of the arousal-based order size is that order sizes will tend to be closer to the immediate market volumes, which may tend to reduce transaction costs. Note that algorithmic strategies such as the above could be embedded in music logic circuits and musical neural networks, allowing them to interact with other PMAP functionality such as sentiment analysis of news text feeds.

In theory the above stock market methodologies could all have been derived purely based on valence and arousal, without mentioning tempo and key mode. However PMAP is designed to simplify the sonification of internal processing. So this work is designed to show another area where PMAP can be applied, rather than to address specifically how the sonification of internal processing has particular benefits in stock market computations. There is also a benefit which stands out here in the use of PMAP – it incorporates a sonification of the market. The melodies provide a natural sonification of stock movements – a useful factor for traders whose eyes are already too busy [32]. One can also consider the harmonic relationship between two stocks, or between a stock and the market. There may be PMAP methods developable such that if stocks start to create cross-dissonance where once was consonance (e.g. one becomes more major as the other stays minor) then this indicates a potential divergence in any correlated behaviour. The incorporation of harmonies into PMAP has already been investigated in relation to MAS [37].

6 Conclusions and Future Work

Through various simulations, this paper has introduced the concept of pulsed melodic affective processing (PMAP), a complementary approach in which computational efficiency and power are balanced with understandability to humans (HCI); particularly where computation addresses rhythmic and simulated emotion processing.

Normally, to propose a new computational model, one needs to develop the model and validate it by solving a real-life problem, or at least by providing a framework for solving such a problem. Then, one needs to evaluate the solution and compare it with other approaches for solving the same problem. In other words, in the case of PMAP,

what problems can be solved using music streams, how efficient the solutions are, and what are their limitations? However, PMAP is not argued to be an efficient and universal form of Turing computation. It is argued to be the first form of computation designed from an HCI perspective. Furthermore, it is designed for affective computation. As a result, validation showing that it is a form of universal computation has not been attempted. Though some form of more formal demonstration of breadth and accuracy in affective computation is clearly desirable, there is no universal method of doing this in affective computing. Hence, as a first step this paper has been dedicated to showing the approach to be potentially applicable in a series of broad but highly simplified scenarios. The next stage of validation would be to perform listening tests to confirm that the affective state represented in the virtual “circuitry” is similar to that for western listeners. This is beyond the scope of an already extensive report, and is part of our planned future work.

In this paper, music gates and murons have been introduced; as well as potential applications for this technology in multi-agent/robot systems, text analysis and stock markets. The tasks are not the most efficient or accurate solutions, but have been a demonstration of a sound-based unified approach addressing HCI and affective processing. In the multi-robot security system, PMAP provided a low-level affective processing which could continue to function if higher systems were damaged. This processing was shown to provide a basic functionality of firing on hostiles and spreading the robots more evenly around the patrol area. It further provided implicit methods for eyes-free monitoring of the robot teams’ behaviours by a human commander / controller.

In the case of the text emotion analysis system, PMAP enabled a music-based machine learning algorithm that provided more direct input / output routes for analysis. The input was a series of rhythms that represented the hierarchical “rhythms” of the text, and the output was music that was shown to encode approximately the affective content of the text in the majority of cases. A non-musical alternative would have been to have as input instead a series of numbers representing the rhythms of the text at different hierarchical levels, and then have as output a valence measure for positivity of text. This involves converting rhythms into numbers on the input, and then converting numbers into emotion on the output. PMAP not only provides an “already sonified” output, but is the natural representation system for such a problem because of the rhythmic nature of input. Whether it is the most efficient approach to machine learning is another question altogether; but as has been repeatedly stated, PMAP is a compromise between computational efficiency and transparency.

The final application demonstration was in an area familiar to sonification research, the stock market. Inspired by insights from behavioural finance, an affective transformation was defined for an order-driven market and demonstrated in a simulation, including a simple algorithmic trading system. Although the algorithmic trading system simulation did not directly use musical notes or parameters (as was the case in the previous two applications), the valence-arousal space used in the simulation was equivalent to the keymode-tempo space of parametric PMAP, and the actual Affective Market Mapping was initially defined in PMAP terms of keymode and tempo.

A key contribution of PMAP is to sonification research. 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 limited to use in affective communications and processing, and such affective states can be represented in many cases by musical data anyway.

There are a significant number of issues to be further addressed with PMAP, a key one being that – now initial results have been obtained in a number of application area simulations – how can a more formal verification be achieved; and one which incorporates HCI testing? Others 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 be better for processing or transparency. 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. Furthermore, most low level processing in standard computation is done in hardware – so issues of how PMAP hardware is built need to be investigated.

Christensen S, Oppacher F. An analysis of Koza's computational effort statistic for genetic programming. In: Foster JA, Lutton E, Miller J, Ryan C, Tettamanzi AG, editors. Genetic programming. EuroGP 2002: Proceedings of the 5th European Conference on Genetic Programming; 2002 Apr 3-5; Kinsdale, Ireland. Berlin: Springer; 2002. p. 182-91.

References

1. Bohte S, Kok J, La Poutre H. Spike-prop: Error-back propagation in multi-layer networks of spiking neurons. *Neurocomputing* 1997, 48(1-4):17-37.
2. Chang M, Wang G, Berger J. Sonification and vizualisation of neural data. In: Proceedings of the International Conference on Auditory Display; 2010 June 9-15; Washington D.C., USA.
3. Huber D. *The MIDI Manual*. Carmel: SAMS; 1991.
4. Miller J, Young W, Simple Pulse Asynchronous State Machines. In: Proceedings of IEEE International Symposium on Circuits and Systems; 1996 May 13-15; Atlanta, USA.
5. Malatesa L, Karpouzis K, Raouzaoui A. Affective intelligence: the human face of AI. In: Bramer E, editors. *Artificial intelligence LNAI 5640*. Berlin: Springer; 2009. p. 53.
6. Banik S, Watanabe K, Habib M, Izumi K. Affection Based Multi-robot Team Work. In: Mukhopadhyay SC, Huang RYM, editors. *Sensors: Advancements in Modeling, Design Issues, Fabrication and Practical Applications LNEE*. Springer; 2008. pp. 355-375.

7. Dahlstedt P, McBurney P. Musical agents: Toward Computer-Aided Music Composition Using Autonomous Software Agents. *Leonardo* 2006, 39:469-470
8. Picard R. Affective Computing: Challenges, *International Journal of Human-Computer Studies* 2003, 59(1-2): 55-64
9. Kirke A, Miranda E. Emergent construction of melodic pitch and hierarchy through agents communicating emotion without melodic intelligence, In: *Proceedings of International Computer Music Conference ICMC*; 2011 July 31-Aug 5; Huddersfield, England.
10. Juslin P, Laukka P. Expression, perception, and induction of musical emotion: a review and a questionnaire study of everyday listening. *Journal of New Music Research* 2004, 33: 216-237
11. Juslin P. From Mimesis to Catharsis: expression, perception and induction of emotion in music. In: Miell D, MacDonald R, Hargreaves DJ, editors. *Music Communication*. Oxford University Press; 2005. p. 85-116.
12. Cohen J. Monitoring Background Activities. In: *Auditory Display: Sonification, Audification, and Auditory Interfaces*. 1994. p. 499-531.
13. Vickers P, Alty J. Siren songs and swan songs debugging with music. *Communications of the ACM* 2003, 46(7):87-92.
14. Gilfix M, Couch AL. Peep (the network auralizer): Monitoring your network with sound. In: *Proceedings of 14th System Administration Conference LISA*; 2000 3-8 Dec, New Orleans, USA; p. 109-117.
15. Nesbitt KV, Barrass S. Evaluation of a multimodal sonification and visualization of depth of market stock data. In: *Proceedings of the 2002 International Conference on Auditory Display*; 2002 2-5 July; Kyoto, Japan.
16. Chafe C, Leistikow L. Levels of Temporal Resolution in Sonification of Network Performance. In: *Proceedings of 7th International Conference on Auditory Display*; 2001 29 July-1 Aug, Helsinki, Finland.
17. Haykin S. *Neural Networks a Comprehensive Foundation*. Prentice Hall; 1994.
18. Kirke A, Bonnot M, Miranda E. Towards using expressive performance algorithms for typist emotion detection, In: *Proceedings of International Computer Music Conference ICMC*; 2011 July 31-Aug 5; Huddersfield, England.
19. Bresin R, Friberg A. Emotional Coloring of Computer-Controlled Music Performances, *Computer Music Journal* 2002, 24(4): 44-63.
20. Krumhansl C, Kessler E. Tracing the dynamic changes in perceived tonal organization in a spatial representation of musical keys, *Psychological Review* 1982, 89(4):334-368.
21. Kirke, A. *Application of Intermediate Multi-Agent Systems to Integrated Algorithmic Composition and Expressive Performance of Music* Ph.D. Thesis, Plymouth University; 2011.
22. Cosmides L, Tooby J. Evolutionary Psychology and the Emotions, In *Handbook of Emotions*, pp. 91--115 (2000).
23. Marinos P. Fuzzy Logic and Its Application to Switching Systems, *IEEE transactions on computers* C 1969, 18(4):343-348.
24. 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* 2006, 17(1):211-221.
25. Stanford V. Biosignals Offer Potential for Direct Interfaces and Health Monitoring. *Pervasive Computing* 2004, 3(1):99-103.
26. Kotchetkov I, Hwang B, Appelboom G, Kellner C, Connolly, ES. Brain-computer Interfaces: Security, Neurosurgical, and Ethical Perspective. *Neurosurgical Focus* 2010, 28(5).
27. Cowley J. *Communications and Networking: An Introduction*, Springer; 2006.
28. RFC 793 - Specification of Internet Transmission Control Program, September 1981 Version

29. Widmer G, Goebel W Computational Models of Expressive Music Performance: The State of the Art. *Journal of New Music Research* 2004, 33:203-216.
30. Kirke A, Miranda E, editors. *Guide to Computing for Expressive Music Performance*. Springer; 2012.
31. Worrall D. The use of sonic articulation in identifying correlation in capital market trading data, In: *Proceedings of the 15th International Conference on Auditory Display*; 2009 18-22 May; Copenhagen, Denmark.
32. Ciardi FC, At FCEM. sMAX A Multimodal Toolkit for Stock Market Data Sonification, In: *Proceedings of ICAD 04 -Tenth Meeting of the International Conference on Auditory Display*; 2004 6-9 July; Sydney, Australia.
33. Kissell R, Glantz M. *Optimal Trading Strategies*. Amacom; 2003.
34. Pole A. Algorithmic Trading and Statistical Arbitrage. *Trading* 2007, 2(1): 79-87.
35. Davies GB, De Servigny A. *Behavioral Investment Management: An Efficient Alternative to Modern Portfolio Theory*, McGraw-Hill (2012).
36. Livingstone SR, Muhlberger R, Brown AR, Loch A. Controlling Musical Emotionality: An Affective Computational Architecture for Influencing Musical Emotions. *Digital Creativity* 2007, 18(1):43-53.
37. Kirke A, Miranda E. Towards Harmonic Extensions of Pulsed Melodic Affective Processing – Further Musical Structures for Increasing Transparency in Emotional Computation. *International Journal of Unconventional Computation* 2014 (In Press), Old City Publishing.

Bios. corresponding author. and phone number

Alexis Kirke (corresponding author) is permanent Research Fellow, Interdisciplinary Centre for Computer Music Research, Plymouth University, Plymouth, UK.
Tel. +44 1752 586227

Eduardo Miranda is Professor of Computer Music at Plymouth University, Plymouth, UK and leads the Interdisciplinary Centre for Computer Music Research at Plymouth University.

Received:31 May 2012

Revised:19 Dec 2013

Revised: 26 Aug 2013

Revised: 27 Feb 2014

Accepted: 7 Mar 2014