Pulsed Melodic Processing – the Use of Melodies in Affective Computations for increased Processing Transparency

Alexis Kirke[1], Eduardo Miranda[1]

[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 Processing (PMP) is a computation protocol that utilizes musically-based pulse sets ("melodies") for processing – capable of representing the arousal and valence of affective states. Affective processing and affective input/output are key tools in artificial intelligence and computing. In the designing of processing elements (e.g. bits, bytes, floats, etc.), engineers have primarily focused on the processing efficiency and power. They then go on to investigate ways of making them perceivable by the user/engineer. However Human-Computer Interaction research - and the increasing pervasiveness of computation in our daily lives – supports a complementary approach in which computational efficiency and power are more balanced with understandability to the user/engineer. PMP allows a user to tap into the processing path to hear a sample of what is going on in that affective computation, as well as providing a simpler way to interface with affective input/output systems. This requires the developing of new approaches to processing and interfacing PMP-based modules. In this chapter we introduce PMP and examine the approach using three example: a military robot team simulation with an affective subsystem, a text affective-content estimation system, and a stock market tool.

Keywords: Human-Computer Interaction, Music, Affective, Boolean Logic, Neural Networks

1 Introduction

This chapter proposes the use of music as a processing tool for affective computation in artificial systems. It has been shown that affective states (emotions) play a vital role in human cognitive processing and expression (Malatesa et al. 2009):

1. Universal and Enhanced Communication – two people who speak different languages can more easily 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 it can reduce the number of

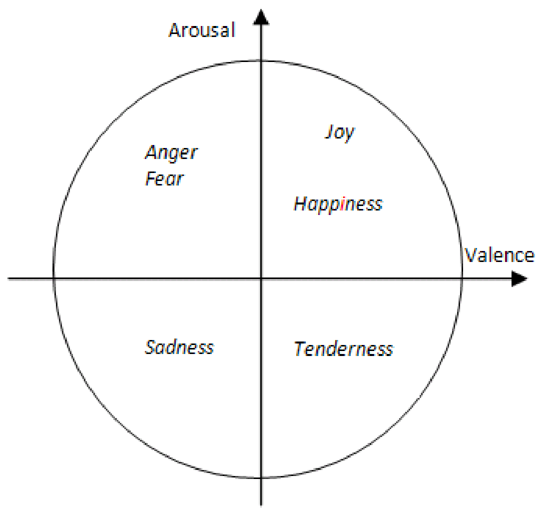
possible strategies in certain situations – if there is a snake in the grass, fear will cause you to only use navigation strategies that allow you to look down and walk quietly. Also pre- and de-emphasising certain responses: for example if a tiger is chasing you, fear will make you keep running and 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 artificial intelligence processing and robotics (Banik et al. 2008). The issue of developing systems with affective intelligence which also provide for greater user-transparency is what is addressed in this chapter. Music has often been described as a language of emotions (Cooke 1959). There has been work on automated systems which communicate emotions through music (Livingstone et al. 2007) and which detect emotion embedded in music based on musical features (Kirke and Miranda 2011). Hence the general features which express emotion in western music are known.

Before introducing these, affective representation will be briefly discussed. The dimensional approach to specifying emotion 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 (Kirke and Miranda 2009) 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.

Fig. 1: The Valence/Arousal Model of Emotion



Previous research (Juslin 2005) 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 major key; whereas Beethoven's melancholy "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.

Affective Computing (Picard 2003) focuses on robot/computer affective input/output. Whereas an additional aim of PMP is to develop data streams that represent such affective states, and use these representations to internally process data and compute actions. The other aim of PMP is more related to Picard's work – to aid easier sonification of affective processing (Cohen 1994) for transparency in HCI, i.e. representing non-musical data in musical form to aid its understanding. Related sonification research has included tools for using music to debug programs [10].

2 PMP Representation of Affective State

Pulsed Melodic Processing (PMP) is a method of representing affective state using music. In PMP the data stream representing affective state is a series of pulses of 10 different levels with a varied pulse rate. This rate is called the "Tempo". The pulse levels can vary across 12 values. The important values are: 1,3,4,5,6,8,9,10,11,12 (for pitches C,D,Eb,E,F,G,Ab,A,Bb,B). These values represent a valence (positivity or negativity of emotion). Values 4, 9 & 11 represent negative valence (Eb, Ab, Bb are part of C minor) e.g. sad; and values 5, 10,& 12 represent positive valence (E, A, B are part of C major), e.g. happy. The other pitches are taken to be valence-neutral. For example a PMP stream of say [1,1,4,4,2,4,4,5,8,9] (which translates as C,C,Eb,Eb,C#,Eb,Eb,E,G,Ab) would be principally negative valence since most of the notes are in the minor key of C.

The pulse rate of a stream contains information about arousal. So [1,1,4,4,2,4,4,5,8,9] transmitted at maximum pulse rate, could represent maximum arousal and low valence, e.g. "Anger". Similarly [10,8,8,1,2,5,3,1] (which translates as A,G,G,C,D,E,C,C) transmitted at a quarter of the maximum pulse rate could be a positive valence, low arousal stream, e.g. "Relaxed" (because it is in the major key of C). If there are two modules or elements both with the same affective state, the different note groups which go together to 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 performing some of the initial analysis on PMP, it is convenient to utilize a parametric form, rather than the data stream form. The parametric form represents a stream by a Tempo-value variable and a Key-value variable. The Tempo-value is a real number varying between 0 (minimum pulse rate) and 1 (maximum pulse rate). The Key-value is an integer varying between -3 (maximally minor) and 3 (maximally major).

3 Musical Logic Gate Example

Three possible PMP gates will now be examined based on AND, OR and NOT logic gates. The PMP versions of these are respectively: MAND, MOR and MNOT (pronounced “emm-not”), MAND, and MOR. So for a given stream, the PMP-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):

MNOT$(m) = [-k,1-t]$ (1)
$m1$ MAND $m2 = [minimum(k1,k2), minimum(t1,t2)]$ (2)
$m1$ MOR $m2 = [maximum(k1,k2), maximum(t1,t2)]$ (3)

These use a similar approach to Fuzzy Logic (Marinos 1969). MNOT is the simplest – it simply reverses 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 are used to represent the four quadrants of the PMP-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.

Table 1: Music Tables for MAND and MNOT

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	-,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				

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 way of viewing this is that MAND requires all inputs to be “optimistic and hard-working” whereas MOR is able to “ignore” inputs which are “pessimistic and lazy”. Another perspective: the MAND of the melodies from Moonlight Sonata (minor key, low tempo) and the Marriage of Figaro Overture (major key, high tempo), the result would be mainly influenced by Moonlight Sonata. However if they are MOR’d, then the Marriage of Figaro Overture would dominate. The MNOT of Marriage of Figaro Overture would be a slow 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"):

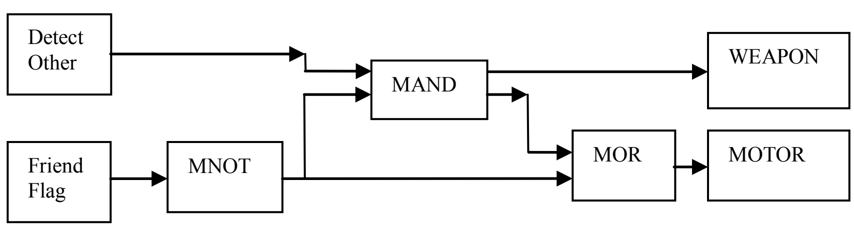
$$m_1 \text{ MXOR } m_2 = (m_1 \text{ MAND MNOT}(m_2)) \text{ MOR } (\text{MNOT}(m_1) \text{ MAND } m_2) \qquad (5)$$

The actual application of these music gates depends on the level at which they are to be utilized. The underlying data of PMP (putting aside the PMP-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 (Indiveri et al. 2006) or VLSI pulse computation systems. A key difference is that the pulse height varies in PMP, 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 machine. So the underlying hardware would use standard logic gates or perhaps standard spiking neurons. The idea of a virtual 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 PMP-type approaches prove useful enough, would provide a practical implementation.

A simple application is now examined. One function of affective states in biological systems is that they provide a back-up for when the organism is damaged or in more extreme states (Cosmides and Tooby 2000). For example an injured person who cannot think clearly, will still try to get to safety or shelter. An affective subsystem for a robot who is a member of a military team is now examined; one that can kick in or over-ride if the higher cognition functions are damaged or deadlocked. Figure 2 shows the system diagram. A group of mobile robots with built-in weapons are placed in a potentially hostile environment and required to search the environment for enemies; and upon finding enemies to move towards them and fire on them. The PMP affective sub-system in Figure 2 is designed to keep friendly robots apart (so as to maximize the coverage of the space), to make them move towards enemies, and to make them fire when enemies are detected.

Fig. 2: Affective Subsystem for Military Multi-robot System



The modules in Figure 2 are “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 friends are identifiable (visually or by RFI) - when an agent is detected within range, and if it is a friendly robot – this module emits a major key mode melody. “MOTOR” – this 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. “WEAPON” - this unit, when it receives a minor key note, fires one round. The weapon and motor system is written symbolically in equations (4) and (5):

WEAPON = *DetectOther* MAND MNOT*(FriendFlag)* (4)
MOTOR = WEAPON MOR MNOT*(DetectOther)* (5)

Table 2: Theoretical Effects of Affective Subsystem

Detect Other	***Friend Flag***	***Detect Other-Value***	***Friend Flag-Value***	**MNOT *(Friend Flag)***	**MAND *Detect Other***	**WEAPON**	**MNOT *(Detect Other)***	**MOR WEAPON**	**MOTOR**
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

Using Equations (1) and (2) gives the theoretical results in Table 2. The 5 rows have the following interpretations: (a) if alone continue to patrol and explore; (b) If a distant enemy 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 robot team are initially deployed and they are bunched together, hence slow movement to prevent collision.

To test in simulation, four friendly robots are used, implementing the PMP-value processing described earlier, rather than having actual melodies within the processing system. The robots using the PMP 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 of 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 PMP 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. The enemy robots 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 enemy robots was calculated. Also the average distances between F-Robots was calculated. These were done with a range of 10 and a range of 0. A range of 0 effective switches off the musical processing. The results are shown in Table 3. 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 enemy robots. 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 3 and 4 show two simulation runs, with each F-Robots' trace represented by a different colour, and each fixed enemy robot shown by an "X".

Table 3: Results for Robot Affective Subsystem

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

It was also found that the WEAPON firing rate had a very strong tendency to be higher as enemies were closer. Robot 1's tempo value when it is within range of an enemy and firing is shown in Figure 5. The x-axis is the distance from the closest enemy, and the y-axis is tempo. It can be seen that the maximum tempo (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 in most cases. In fact the correlation between the two is -0.98 which is very high. This shows that PMP allows the same flexibility as fuzzy logic, in that the gun rate is controlled fuzzily from minimum to maximum. The line is not straight and uniform because it is possible for robot 1 to be affected by its distance from other enemies and from other friendly robots.

Finally it is worth considering what these robots actually sound like as they move and change status. To allow this each of the 4 robots was assigned a distinctive motif, with constant tempo. Motives designed to identify a module, agent, etc. will be called "Identive". The identives for the 4 robots were:

1. [1,2,3,5,3,1] = C,D,Eb,F,Eb,D,C
2. [3,5,8,10,8,5,3] = Eb,F,G,Ab,G,F,Eb
3. [8,10,12,1,12,10,8] = G,Ab,Bb,C,Bb,Ab,G
4. [10,12,1,5,1,12,10] = Ab,Bb,C,G,C,Bb,Ab

Fig. 3: Simulation of Military Robots without Pulsed Melodic Processing

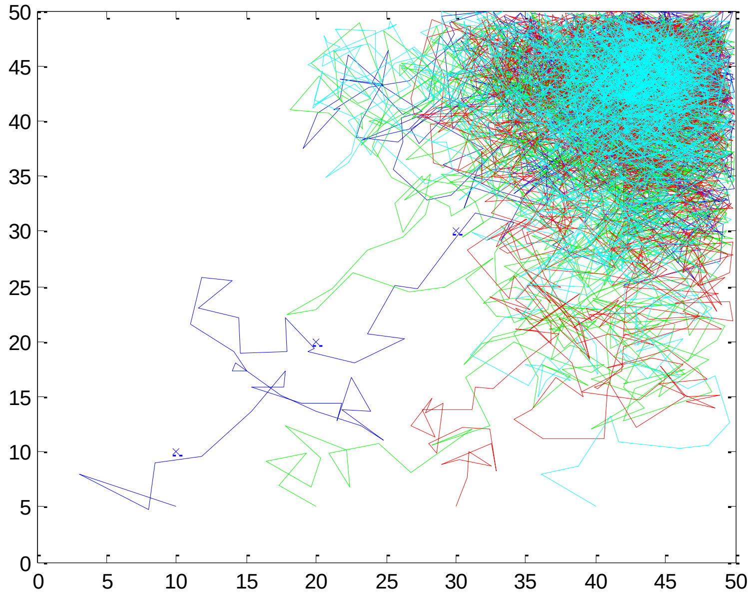


Fig. 4: Simulation of Military Robots with PMP system and Range of 10 units

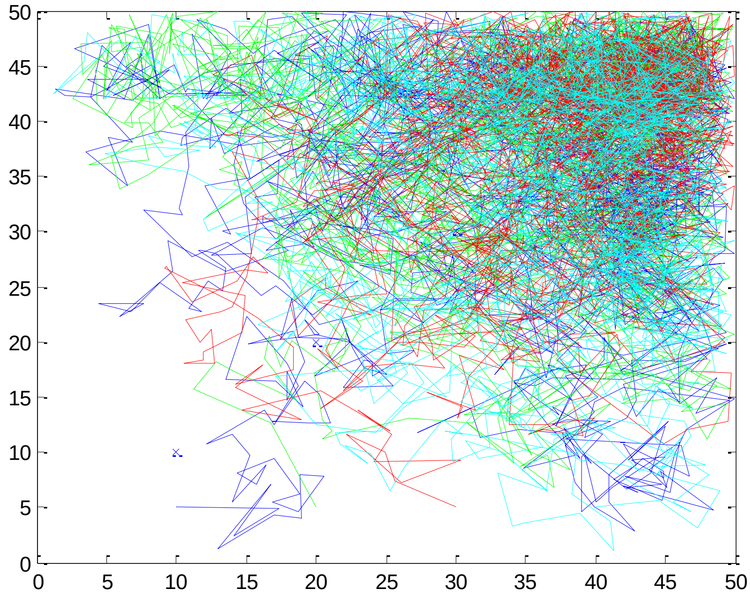


Figure 6 shows the first 500 notes of robots 1 to 3 in the simulation in piano roll notation. The octave separation used for the Figure 6 also helped with aural perception. (So this points towards octave independence in processing as being a useful feature.) It was found that more than 3 robots were not really perceivable. 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 seem to make changes more perceivable as well.

Fig. 5: Plot of distance of R1 from enemy when firing + weapon tempo value

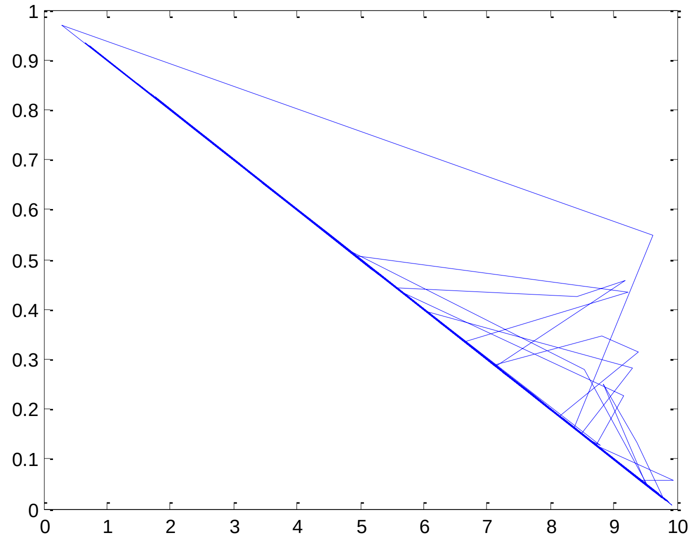
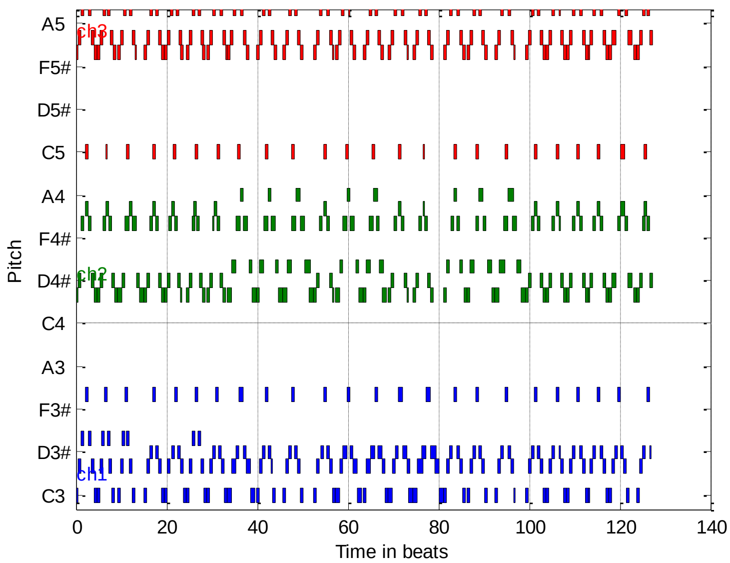


Fig. 6: A plot of 500 notes in the "motor" processing of robots 1 to 3 (octave separated).



An extension of this system is to incorporate rhythmic biosignals from modern military suits (Stanford 2004)(Kotchetkov et al. 2010). For example if "BioSignal" is a tune generating module whose tempo is a heart rate reading from a military body suit, and whose key mode is based on EEG valence readings, then the MOTOR system becomes:

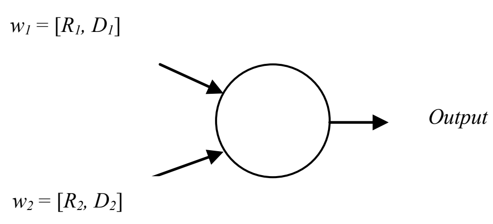
$$MOTOR = WEAPON\ MOR\ MNOT(DetectOther)\ MOR\ MNOT(BioSignal) \quad (6)$$

The music table for (6) 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.

4 Musical Neural Network Example

We will now look at a form of learning artificial neural network which uses PMP. 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 7. The muron in this example has two inputs, though it can have more than this. Each input is a PMP melody, and the output is a PMP 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. 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 PMP is most efficiently used are those that naturally utilize temporal or affective data (or for which internal or external sonification is particularly important).

Figure 7: A Muron with two inputs.



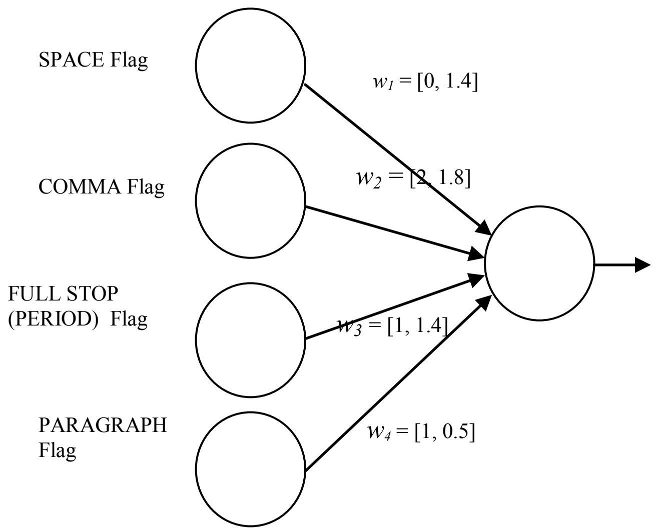
One such system will now be proposed for the estimation of affective content of real-time typing. The system is inspired by research by the authors on analysing

QWERTY keyboard typing, in a similar way that piano keyboard playing is analyzed to estimate the emotional communication of the piano player (Kirke et al. 2011). In this a real-time system was developed to analyse tempo of typing and estimate affective state. The MNN/PMP version demonstrated in this chapter is not real-time, and does not take into account base typing speed. This is to simplify simulation and experiments here. The proposed architecture for offline text emotion estimation is shown in Figure 8. It has 2 layers known as the Input and Output layers. The input layer has four murons – which generate notes. Every time a Space character is detected, then a note is output by the Space Flag. If a comma is detected then a note is output by the comma flag, if a full stop/period then the Period Flag generates a note, and if an end of paragraph is detected then a note is output by the Paragraph flag. 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, because offline pre-typed data is used. These letters make up words (which are usually separated by a space). The words make phrases (which are often separated by commas). 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 murons represent the relative word-rate, phrase-rate, sentence-rate and paragraph rate of the typist. (Note that for data from a messenger application, the paragraph rate will represent the rate at which messages are sent). It has been found by researchers that the mood a musical performer is trying to communicate effects not only their basic playing rate, but also the structure of the musical timing hierarchy of their performance (Kirke et al. 2011). Similarly we propose that a person's mood will affect not only their typing rate (Kirke et al. 2011), but also their relative word rate and paragraph rate, and so forth.

The input identives are built from a series of simple rising semitone melodies. The desired output of the MNN will be a tune which represents the affective estimate of the text content. A happy tune means the text structure is happy, sad means the text is sad. Normally Neural Networks are trained using a number of methods, most commonly some variation of gradient descent. A gradient descent algorithm will be used here. w_1, w_2, w_3, w_4 are all initialised to [0,1] = [Key mode sub-weight, Tempo sub-weight]. So initially the weights have no effect on the key mode, and multiply tempo by 1 – i.e. no effect. The final learned weights are also shown in Figure 8. Note, in this simulation actual tunes are used (rather than PMP-value parameterization used in the robot simulation). In fact the Matlab MIDI toolbox is used. The documents in the training set were selected from the internet and were 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 w1, w2, w3, or w4. 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, separately for tempo and for key mode. Before training, the initial average error rate across the 30 documents was calculated. The key mode was measured using a modified key finding algorithm (Krumhansl and Kessler 1982) which gave a value of 3 for maximally major and -3 for maximally minor. The tempo was measured in Beats per minute. The initial average error was 3.4 for key mode, and 30 for tempo.

Fig. 8: MNN for Offline Text Affective Analysis



After the 1920 iterations of learning the average errors reduced to 1.2 for key mode, and 14.1 for tempo. These results are described more specifically in Table 4 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. (Though this system could be embedded as music into messenger software to give the user affective indications through sound). It can be seen in Table 4 that the mean tempo error for Happy documents (target 90 BPM) is 28.2 BPM. This is due to an issue similar to linear non-separability in normal artificial neural networks (Haykin 1994). The Muron is approximately adding tempos linearly. So when it tries to approximate two tempos then it focuses on one more than the other – in this case the Sad tempo. Hence adding a hidden layer of murons may well help to reduce the Happy error significantly (though requiring some form of melodic Back Propagation).

Table 4: Mean Error of MNN after 1920 iterations of gradient descent

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

5 Affective Market Mapping

The Affective Market Mapping (AMM) involves mapping stock movements onto a PMP representation. One mapping that was initially considered was a risk / return mapping – letting risk be mapped onto arousal / tempo, and return be mapped onto valence / key mode. However this does not give an intuitively helpful result. For example it implies 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 calculated as in (Kissell and Glantz 2003) - 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. 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 have 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 PMP as a minor key and high tempo – previously labelled as "angry". Stocks trading up rapidly on good news will be "happy", stocks trading up slowly in a generally positive market will be "relaxed". The resulting PMP stream matches what many would consider their affective view of the stock.

For algorithmic trading, affective strategies can be investigated. An example might be "affective arbitrage". In this case the affective content of current news stories about a company could be automatically ascertained by text scanning algorithms (either using an MNN of the type in the previous section, or by keyword analysis that utilizes the various word-affective databases available). These could be compared to the affective state of the company's stocks, bonds etc. If there is a sufficient disparity, then an opportunity may exist for arbitrage. Suppose we define a measure of PMP called the "Positivity":

$$positivity(X) = keyValue(X) + tempoValue(X) \tag{7}$$

Then happy stocks would have a higher positivity than sad ones, and relaxed would have a slightly lower positivity than happy ones. An algorithmic trading rule could be:

If *positivity(newsStoriesAboutX MXOR stockX) > k* Then *Flag(stockX)* (8)

The MXOR function will give a low key/tempo (valence/arousal) output for valence and arousal as long as the news story and the stock's affectivity are similar enough. However if the news story becomes very emotionally positive while the stock is more negative, or vice versa, then the MXOR value will begin to increase. Data mining for "affective arbitrage" opportunities could be done by investigating various functions of stocks and seeing if they allow profits; for example rules such as:

positivity(Stock1 MAND stock2) > x
positivity(Stock1 MXOR Market) > y
positivity(Stock1) + postivity MNOT(Market) < z

could be investigated. Trader "feeling" about the market sentiment could also be incorporated. "I've got a good feeling", "Slow market", etc.

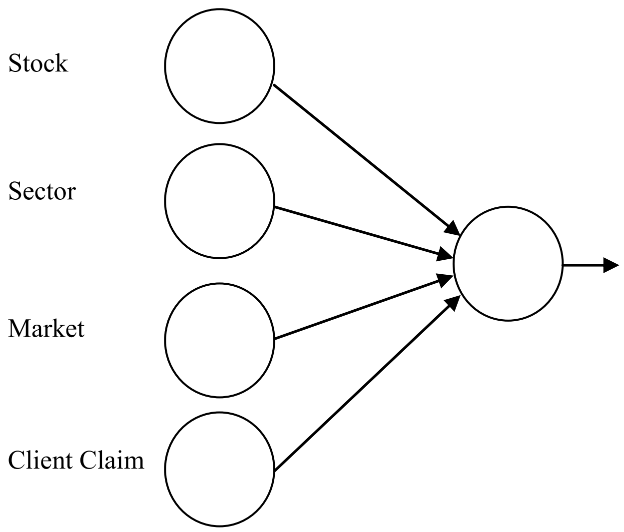
Emotions are considered to be particularly relevant in the field of behavioural finance (Subrahmanyam 2008). In behavioural finance a client's emotional reaction to stock movements may be different to the actual rational implications of the stock movements. Hence an investment professional needs to optimize a client's affective reactions as well as their portfolio. Figure 9 shows a possible approach to learning a client's behaviour biases for investing using PMP. In the past a client may have said they are happy to invest in stocks S1 to Sn. However in reality they may show different affective responses to the movements of these stocks over the time they are held. The MNN in Figure 9 is trained based on past client reactions. For example if they were happy about the performance of MSFT (Microsoft) which is part of the tech sector in the S&P 500 market, then that can be used as training date for the MNN. This can then be applied for all stocks the client has reported particular positive or negative reactions to. Any stocks they do not report on could be assumed to have a "relaxed" affect for the client. As data is collected it will gradually become clear to the MNN how the client will react. Then when the portfolio is rebalanced, any stocks which cause an excessive negative reaction can be optimized out.

The "Client Claim" input is based on any questionnaire a client was given when having over management of their portfolio to the investment professional. For example a new client may claim they "like" tech stocks, and dislike utility "stocks". Note – that it would probably be necessary to add a hidden layer to the MNN to achieve useful results.

The affective state of a portfolio is calculated as the average PMP values across all stocks in the portfolio. So a portfolio full of frequently trading stocks will have a higher tempo. A portfolio where stocks are being sold off will tend to have a minor key / low valence.

As well as considering the affectivity of a stock or a market, we can consider the affectivity of a trading strategy. A "happy" strategy is buying fast, an "angry" strategy is selling fast. For example, consider investment implementations: the Market Impact (Kissell and Glantz 2003) of a stock trade can be viewed as a form of affective influence – moving the key/valence of the stock's market towards that of the trader and thus incurring a cost. So minimizing market impact involves minimizing the effect of the trader's key/valence on the market's key/valence. Minimizing trading risk involves maximising tempo/arousal so the market does not have time to move against you. So minimizing these sorts of trading costs for a single stock involves maximizing tempo in your trading, while keeping the key/valence-influence minimal.

Fig. 9: MNN for Offline Learning Client Preferences



As well as the processing potential of PMP in the markets, it is interesting to note that the melodies provide a natural sonification of stock movements and processing – a useful factor for traders whose eyes are already too busy. One can also consider the harmonic relationship between two stocks, or between a stock and the market. If they start to create dissonance where once was consonance (e.g. one becomes more major as the other stays minor) then this indicates a divergence in any correlated behaviour.

So there are four elements which suggest PMP may have potential in the stock markets: the simple Market Mapping, the incorporation of trader, client and news article "feelings" into what is an art as well as a science, a unified framework for affectivity across short and long-term investments and trading implementation, and a natural sonification for eyes-free HCI in busy environments.

6 Conclusions

This chapter has introduced the concept of pulsed melodic processing, a complementary approach in which computational efficiency and power are more balanced with understandability to humans (HCI); and which can naturally address rhythmic and affective processing. As examples music gates and murons have been introduced; as well as potential applications for this technology in robotics, real-time text analysis and financial markets. This chapter is by necessity a summary of the research done, leaving out much of the detail and other application ideas; these include the use of biosignals, sonification experiments, ideas for implementing PMP in a high level language, programming by music, etc. However it demonstrates that music can be used to process affective functions either in a fixed way or via learning algorithms. The tasks are not the most efficient or accurate solutions, but have been a proof of concept of a sound-based unified approach addressing HCI and processing.

There are a significant number of issues to be further addressed with PMP, a key one being 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. PMP also has a close relationship to Fuzzy Logic and Spiking Neural Networks – so perhaps it can adapted based on lessons learned in these disciplines. And finally, most low level processing is done in hardware – so issues of how PMP hardware is built need to be investigated.

References

Banik, S., Watanabe, K., Habib, M., Izumi, K.: Affection Based Multi-robot Team Work, In Lecture Notes in Electrical Engineering, pp. 355--375 (2008).

Bresin, R., Friberg, A.: Emotional Coloring of Computer-Controlled Music Performances, Computer Music Journal, Vol. 24, No. 4, pp. 44--63 (2002).

Cohen, J.: Monitoring Background Activities, In Auditory Display: Sonification, Audification, and Auditory Interfaces, pp. 499--531 (1994).

Cooke, D.: The Language of Music. Oxford University Press (1959).

Cosmides, L., Tooby, J.: Evolutionary Psychology and the Emotions, In Handbook of Emotions, pp. 91--115 (2000).

Haykin, S.: Neural Networks: A Comprehensive Foundation, Prentice Hall (1994).

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).

Juslin, P.: From Mimesis to Catharsis: expression, perception and induction of emotion in music, In Music Communication, pp. 85--116, Oxford University Press (2005).

Kirke, A., Miranda, E.: A Survey of Computer Systems for Expressive Music Performance, ACM Surveys, Vol. 42, No. 1, pp. 1--41 (2009).

Kirke, A., Bonnot, M., Miranda, E.: Towards using expressive performance algorithms for typist emotion detection, In Proceedings of 2011 International Computer Music Conference (ICMC 2011), International Computer Music Association (2011).

Kirke, A., Miranda, E.: 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 (2011).

Kissell, R., Glantz, M.,: Optimal Trading Strategies, Amacom (2003).

Kotchetkov, I., Hwang, B., Appelboom, G., Kellner, C., Sander Connolly, E.: Brain-computer Interfaces: Military, Neurosurgical, and Ethical Perspective, Neurosurgical Focus, Vol. 28, No. 5 (2010).

Krumhansl, C., Kessler, E.: Tracing the dynamic changes in perceived tonal organization in a spatial representation of musical keys, Psychological Review, Vol. 89, No. 4, pp. 334--368 (1982).
Livingstone, S.R., Muhlberger, R., Brown, A.R., Loch, A.: Controlling Musical Emotionality: An Affective Computational Architecture for Influencing Musical Emotions. Digital Creativity, Vol. 18, No. 1, pp. 43--53 (2007).
Malatesa, L., Karpouzis, K., Raouzaiou, A.: Affective intelligence: the human face of AI, In Artificial intelligence, Springer-Verlag (2009).
Marinos, P.: Fuzzy Logic and Its Application to Switching Systems, IEEE transactions on computers C, Vol. 18, No. 4, pp. 343--348 (1969).
Picard, R.: Affective Computing: Challenges, International Journal of Human-Computer Studies, Vol. 59, No. 1-2, pp. 55--64 (2003).
Stanford, V.: Biosignals Offer Potential for Direct Interfaces and Health Monitoring, Pervasive Computing, Vol. 3, No. 1, pp. 99-103 (2004).
Subrahmanyam, A.: Behavioural Finance: A Review and Synthesis, European Financial Management, Vol. 14, No. 1, pp. 12--29 (2008).
Vickers, P., Alty, J.: Siren songs and swan songs debugging with music. Communications of the ACM, Vol. 46, No. 7, pp. 87--92 (2003).