PROJECT MUSE®

Application of musical Computing to Creating a dynamic Reconfigurable multilayered Chamber Orchestra Composition

Alexis Kirke

Leonardo Music Journal, Volume 29, 2019, pp. 55-61 (Article)

Published by The MIT Press

➔ For additional information about this article

<https://muse.jhu.edu/article/745358>

Application of Musical Computing to Creating a Dynamic Reconfigurable Multilayered Chamber Orchestra Composition

ALEXIS KIRKE

ABSTRACT

With increasing virtualization and the recognition that today's virtual computers are faster than hardware computers of 10 years ago, modes of computation are now limited only by the imagination. Pulsed Melodic Affective Processing (PMAP) is an unconventional computation protocol that makes affective computation more human-friendly by making it audible. Data sounds like the emotion it carries. PMAP has been demonstrated in nonmusical applications, e.g. quantum computer entanglement and stock market trading. This article presents a musical application and demonstration of PMAP: a dynamic reconfigurable score for acoustic orchestral performance, in which the orchestra acts as a PMAP halfadder to add two numbers.

ORCHESTRAS AND NONMUSICAL PROCESSES

This article presents a composition in which an orchestra carries out a nonmusical process, specifically a form of computation. Most orchestral projects involving nonmusical processes are musification (i.e. they use more traditional elements of sound such as melody, harmony and rhythm). The composition *Orchestral Processing Unit*, discussed below, is instead an instantiation of a computation. However, because most orchestral work related to nonmusical processes is musification, I briefly highlight examples of past work in that area.

Cullen and Coyle [1] used synthesized orchestration to sonify employee-related data in a database. Hinterberger [2] examined the sonification of EEG ("brainwave") readings with orchestra. An orchestral-sound project [3] sonified data from supernovae. Another sonification of physics data was the LHChamber Music project [4]. In this project, different instruments played data from different experiments. Eduardo Miranda [5] wrote a piece for symphonic orchestra and choir; as part of this piece he sonified the output of a simulated biological neural network. The Heart Chamber Orchestra [6] generated a live score using the heart rates of

12 musicians. The score appeared on computer screens in front of each musician.

The process used to generate the acoustic orchestral performance described in this article is computation. What is unusual about this is that the computation itself is a musical substrate. The motivation for and implementation of this substrate is now explained.

MUSICAL PROCESSES WITH NONMUSICAL FUNCTIONS

Virtual machines—such as the Java Virtual Machine [7] and machines that allow users to run a nonnative OS [8]—have existed for decades. Server virtualization is common; it sacrifices processing power in the virtualization process [9], but this is outweighed by the advantages.

The virtual machine referred to in this article does not reduce computation time. It is more in the vein of the graphical user interface (GUI) interaction paradigm, also known as Window, Icon, Mouse and Pointer (WIMP) interfaces. WIMP interfaces did not increase processing power when added. In fact, they reduced processing power [10] because of the requirements for bitmapped screens and windows. However, many tasks now would be unfeasibly slow for us without WIMP. Changing the mode of human-computer interaction opens up opportunities for increasing the usefulness of computers, even though it uses up processing power.

Given the recent developments in virtual computing, unconventional computing can now greatly expand its possible modes. Such modes are limited only by the imagination; hence the new field of unconventional virtual computation (UVC) [11]. Previous work using simulations to run unconventional computation involved simulating a hardware or wetware system, not making fundamental simulation the substrate [12].

One mode of UVC that could provide benefits is human-computer interaction by replacement (HCI by Replacement, or HBR). HBR [13] is an approach to unconventional virtual computing that combines computation with HCI, a complementary approach in which computational efficiency and

Alexis Kirke (composer, researcher), University of Plymouth, Interdisciplinary Centre for Computer Music Research, Room 003, Hepworth Building, Drake Circus, U.K. Email: alexis.kirke@plymouth.ac.uk. ORCID: 0000-0001-8783-6182.

See www.mitpressjournals.org/toc/lmj/29 for supplemental files associated with this issue.

power are balanced with the goal of making the user interface understandable to humans. Rather than working with ones and zeros in a simulated circuit, the user has the user-interface object itself; e.g. if the user wants data to be audible, they would replace the computation substrate with melodies. Some forms of HBR may not be implementable in hardware in the foreseeable future, but current hardware speeds could be matched by future virtual HBR machines.

My focus here is on forms of HBR in affective computation or in computation that has an affective interpretation. Research has shown that affective states (emotions) play a vital role in human cognitive processing and expression [14]. As a result, affective state processing has been incorporated into robotics and multiagent systems. Researchers have been actively exploring ways to represent and compute with affective states.

The dimensional approach to specifying emotional state is one common approach. In many emotional music systems [15], two dimensions are used: valence and arousal. In these models, emotions are plotted on a graph, 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.

To a degree, these affective states can be represented musically. A number of questionnaire studies support the argument that music communicates emotions, and previous research [16] has suggested that a main indicator of valence is musical key mode: A major key mode implies higher valence, while a minor key mode implies lower valence. For example, the overture to Mozart’s *The Marriage of Figaro* is in a major key, whereas the melancholic first movement of Beethoven’s Piano Sonata No. 14 (*Moonlight*) is in a minor key.

The research also underscores that tempo is a prime indicator of arousal, with a high tempo indicating high arousal and a low tempo, low arousal. For example, compare Mozart’s fast overture to *The Marriage of Figaro* with Debussy’s major-key-but-low-tempo opening to “Girl with the Flaxen Hair.” The Debussy piano-piece opening has a relaxed feel that is low-arousal despite having a high valence [17].

PMAP REPRESENTATION

In the HBR protocol Pulsed Melodic Affective Processing (PMAP) [18], the data stream representing affective state is a stream of pulses transmitted at a variable rate (cf. the variable rate of pulses in biological neural networks in the brain). The pulse rates encode information (neuroscientists often use audio probes to listen to neural spiking). In PMAP, this pulse rate specifically encodes a representation of the arousal of an affective state. A high pulse rate is essentially a series of events at a high tempo (hence high arousal), whereas a low pulse rate is a series of events at a low tempo (hence low arousal).

Additionally, the PMAP pulses can have variable heights, with 10 possible levels—for example, 10 different voltage levels for a low-level stream, or 10 different integer values for a stream embedded in some sort of data structure. The purpose

of pulse height is to represent the valence of an affective state using pitch and key mode [19].

PMAP provides a method for the processing of artificial emotions. PMAP data tempo can be generated directly from rhythmic data and be turned directly into rhythmic data or sound. Thus rhythms such as heart rates, key-press speeds or time-sliced photon-arrival counts can be turned directly into PMAP data, and PMAP data can be turned directly into music, with minimal transformation. PMAP has been applied and tested in a number of simulations, listed below. (More details are available in previous UVC and PMAP publications; sources are provided in the notes.)

- a. A security team multirobot system [20]
- b. A musical neural network to detect textual emotion [21]
- c. A stock market algorithmic trading and analysis approach [22]
- d. A system that keeps a photonic quantum computer in a state of maximum entanglement [23]

Due to space limitations, only (a) is briefly described in this article. The security robot team simulation, conducted by Kirke and Miranda, involved robots with two levels of intelligence: a higher-level, more advanced cognitive function and a lower-level basic affective functionality. The lower-level functionality could take over if the higher level ceased to work. A new type of logic gate was designed to use to build the lower level: a musical logic gate. PMAP equivalents of AND, OR and NOT were defined, inspired by fuzzy logic.

The PMAP versions of AND, OR and NOT are, respectively, MAND, MOR and MNOT (pronounced “emm-not”). Thus, for a given stream, a PMAP segment of data can be summarized 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):

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

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

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

Kirke and Miranda showed that, using a circuit of such gates, PMAP could provide basic fuzzy search-and-destroy functionality for an affective robot team. They also found that the state of a three-robot team could be made human-audible by tapping into parts of the PMAP processing stream and listening to key mode and tempo.

PMAP HALF-ADDER

The main focus of this article is the application of PMAP to a dynamic reconfigurable orchestral performance. *Orchestral Processing Unit* (OPU) is a chamber orchestra performance using the tempo encoding of PMAP. Because of an orchestra’s size and slow reaction time, and because a performance needs to have enjoyable coherent development, I decided that an orchestral performance using PMAP would have to focus on the PMAP process rather than having the PMAP process control something else. Furthermore, I desired to make the

Fig. 1. Circuit and truth table for a half-adder, 2019. (Circuit attribution: inductiveload [public domain], from Wikimedia Commons; truth table © Alexis Kirke.)

process a form of computation that would be understandable to the audience in a more intuitive way. Using an orchestra as an affective computation system would be too complex for an audience to understand and therefore experience. Thus, I implemented a more traditional form of computer circuit in orchestral PMAP: a two-bit addition system. Such a system, when looped through twice, allows the addition of two numbers between 0 and 3.

The orchestral PMAP adder is implemented using two passes through a half-adder. A half-adder is implemented in traditional computation, as seen in Fig. 1. S is the sum of binary numbers A and B. If the resulting sum is greater than 1 bit in size, it is set to 0 and the carry flag C is set to 1. A half-adder allows two numbers between 0 and 1 to be added, as shown in Fig. 1. To allow numbers between 0 and 2 to be added, two passes are needed through the PMAP half-adder.

The most musically controllable way of making a half-adder in PMAP is to focus on two tempo values and remove the pitch dimension. It is problematic to use the pitch dimension in a musical composition, since having an orchestra play concurrent major and minor lines will usually lead to unpleasant dissonance.

For OPU, a logic level is represented by a monophonic musical phrase. If the phrase is played with its original note lengths, it is considered to be logic 1; if played at half its original note length, it is considered to be logic 0. Then the AND gate in Fig. 1 can be replaced by a MAND gate. The XOR gate is replaced as follows. XOR is constructed in Boolean logic using AND, OR and NOT gates:

$$\text{Output} = (A \text{ AND } \text{NOT}(B)) \text{ OR } (\text{NOT}(A) \text{ AND } B) \quad (4)$$

Because the MAND is functionally a two-dimensional fuzzy AND gate, and because the tempo in OPU has only two states, a minimum and a maximum, and because in the extreme a fuzzy AND is identical to an AND gate, the AND gate can be replaced by a MAND gate. The same applies for the OR and NOT gates. Thus we have:

$$\text{Output} = (A \text{ MAND } \text{MNOT}(B)) \text{ MOR } (\text{MNOT}(A) \text{ MAND } B) \quad (5)$$

Fig. 2. PMAP half-adder, looped through twice, 2019. (© Alexis Kirke)

Equation (5) is then implemented in the orchestra, as shown in Fig. 2.

PROCEDURE IN PERFORMANCE

The processing is implemented as follows. Melodies—call them M_A and M_B —represent inputs A and B in Fig. 1. For M_A a logic 1-value is represented by M_A played with its defined note lengths. A 0-value is represented by M_B played with its note lengths at half the note length of M_A —i.e. the melody is played at twice the speed and takes half as long to play. Figure 3 shows the two possible states for one of the clarinets during a particular part of the calculation/performance. The upper line is logic 1 and the lower line (with twice the note lengths) is logic 0. (Only the first half of the line is shown.)

Each group of instruments in Fig. 2 has a paper score showing, at every point in the performance, the two possible phrases the group could be playing: the maximum-tempo phrase, and the same phrase played at half the tempo. The conductor indicates to players when they should play and whether they should play the top line or the bottom line.

Note that, due to the size of the orchestra and the time and style constraints, the conductor performs the carry calculations and storage; these are not included in Fig. 2. As will be seen below, the conductor is given a rule that if both

the trombone and viola represent a 1-bit, then the conductor should make a note of this. The existence or nonexistence of these notes affects which musical lines they indicate to the players later in the calculations.

To see how one of the two half-adder calculations are done, suppose the conductor wants to input a 1-value to the top input in the circuit in Fig. 2. The piano is the input, output and storage register of the adding process (it does not perform any processing). So the conductor indicates the piano should start playing M_A . Consider the top line of processing in Fig. 2, which can be written as:

$$\text{Horn} = \text{Piano} \quad (6)$$

$$\text{Trumpet} = \text{NOT}(\text{Horn}) \quad (7)$$

The conductor next indicates to the horn that it should play what the piano (the input register) is playing from its score. The horn plays M_A as well. The next stage is the trumpet playing the MNOT of M_A . The definition of MNOT is that the maximum tempo becomes the minimum tempo and vice versa, as shown in equation (1). So the conductor indicates to the trumpet that it should start to play M_A at double-note length (half the speed). The next musical gate along this line of Fig. 2 is the MAND. Based on equation (2), the MAND outputs the minimum tempo for all inputs, unless both of its inputs are maximum tempo. This MAND gate has inputs from the percussion and the trumpet. Given that the trumpet is playing minimum tempo, the MAND output—the trombone—is told by the conductor to play the phrase at a minimum tempo. The whole trombone score is shown in Fig. 4.

I gave the conductor a “verbal score” (reproduced in Table 1) to perform the calculation. The first column—the segment column—refers to markings on the conductor’s full score. The full score is too large to include, but an excerpt is shown

Fig. 4. Single-instrument score example—tenor trombone, 2019.
(© Alexis Kirke)

in Fig. 5. Segments S1 to S29 perform two XOR calculations. S31 is actually a special case for the conductor instruction set, as shown in Table 2. S31 involves combining carries to create the final sum. These carries are noted down by the conductor during the playing of Segments S4 and S19. The piano is also a special case. As the calculation/performance continues, the piano acts as a register—storing intermediate calculation results. The piano can store two bits of data, in the left and the right hand.

Looking at Table 1, the four bits are input at segments S1, S3, S16 and S18. S1 and S3 are the least and most significant bits for the first input; S16 and S18 are the least and most significant bits for the second input. After the conductor guides the orchestra through to S31, the piano plays the three output bits.

TABLE 1. Conductor verbal score.

Segment	Instrument Beginning	Top or Bottom?
S1	Piano	Composer will tell you top X or bottom X stave.
S2	Violin	Same stave as piano X in S1.
S3	Piano	Composer will tell you top X or bottom X stave.
S4	Flute (If violin and flute are both top, conductor make a note somewhere say “C1”)	Same stave as piano X in S3.
S5	[Piano STOP] Oboe	If flute playing top, then play bottom. If flute playing bottom, then play top.
S6	Viola	Top if violin and oboe is top; bottom otherwise.
S7	Piano	Composer will tell you top X or bottom X stave.
S8	Horn	Same stave as piano X in S7.

(Table continued on next page.)

TABLE 1. Conductor verbal score (continued).

Segment	Instrument Beginning	Top or Bottom?
S9	Piano	Composer will tell you top X or bottom X stave.
S10	Timpani	Same stave as piano X in S9.
S11	[Piano STOP] Trumpet	If horn playing top, then play bottom. If horn playing bottom, then play top.
S12	Trombones	Top if timpani and trumpet are top; bottom otherwise.
S13	Cello	Top UNLESS trombone and viola are bottom.
S14	Piano	Same stave as cello in S13.
S15	Stop all instruments EXCEPT piano	
S16	Piano Y	Pianist keeps playing X stave it was playing in S15 all the way through to S31 inclusive. Instructions below are for <i>additional</i> lines. Composer will tell you top Y or bottom Y stave.
S17	Violin	Same stave as piano Y in S16.
S18	Piano Y	Composer will tell you top B or bottom B stave.
S19	Flute / bass / clarinet 1 & 2 / oboe (If violin and flute are both top, conductor make a note somewhere say <u>"C2"</u>)	Same stave as piano Y in S18.
S20	Bassoon	If flute playing top, then play bottom. If flute playing bottom, then play top.
S21	Viola	Top if violin and bassoon are top, bottom otherwise.
S22	Piano Y	Composer will tell you top Y or bottom Y stave.
S23	Horn	Same stave as piano Y in S22.
S24	Piano Y	Composer will tell you top Y or bottom Y stave.
S25	Timpani	Same stave as piano Y in S24.
S26	[Piano STOP, except for X stave theme] Trumpet	If horn playing top, then play bottom. If horn playing bottom, then play top.
S27	Trombone / bass trombone	Top if timpani and trumpet are top; bottom otherwise.
S28	Cello	Top UNLESS trombone and viola are bottom, in which case bottom.
S29	Piano Y	Same stave as cello in S28.
S30	Stop all instruments EXCEPT piano	
S31	Piano X, Y, Z	LEFT HAND: Continue piano X one octave down. RIGHT HAND (See table below or): If C2 marked then following applies to piano Z, otherwise it applies to piano Y: If piano was playing piano Y top in S30 and C1 is marked, OR if piano was playing piano Y bottom in S30 and C1 is not marked, then play bottom; else play top.
S32	Stop all instruments.	

(© Alexis Kirke)

TABLE 2. Alternative conductor instructions for final calculation.

Continue Piano X in S30 one octave down, and:

If C2	and C1	and piano Y in S30 playing	then play
Not marked	Not marked	Bottom	Bottom on piano Y
Not marked	Not marked	Top	Top on piano Y
Not marked	Marked	Bottom	Top on piano Y
Not marked	Marked	Top	Bottom on piano Y
Marked	Not marked	Bottom	Bottom on piano Z
Marked	Not marked	Top	Top on piano Z
Marked	Marked	Bottom	Top on piano Z
Marked	Marked	Top	Bottom on piano Z

This table is an alternative to using the conductor instructions in S31. It may be easier to read and enact. It has the same effect as the instructions in S31 but explains them in a different way. (© Alexis Kirke)

Fig. 5. Conductor score excerpt, 2019. (© Alexis Kirke)

PERFORMING A CALCULATION IN CONCERT

The performance took place at the Peninsula Arts Contemporary Music Festival in Plymouth, U.K., performed by the Ten Tors Orchestra and conducted by Simon Ible. A recording of the PMAP movement is available online [24]. The two numbers added in the performance were 3 and 3. I sent these numbers to the conductor around 48 hours before the

performance (he had had the reconfigurable score for a few weeks before this). The binary values were 11 and 11, so what I actually sent to the conductor were the instructions to get the piano to play the top line of the score at each of its four input points. The result of the sum meant that two carries of 1 were stored by the conductor; this led him to instruct the pianist to play certain chords at the end whose constituent

lines represented binary 110—which is 6 in decimal, the result of $3 + 3$.

The orchestra succeeded in calculating accurately. I did not attend the rehearsals; I provided only scores and instructions. The accurate performance indicated the musicians' ability to interpret a computation with the conductor. The fact that the conductor asked for the calculation inputs to be provided 48 hours before the performance indicates that he wanted to limit the degree to which the musicians had to make live choices in responding to him during live performance. However, I never gave him the "hard" score—i.e. the actual fixed musical score for a $3 + 3$ calculation—but only the protocol and the inputs.

CONCLUSIONS

In this article, I introduced the unconventional virtual computation protocol PMAP. PMAP has been demonstrated previously to have nonmusical functionalities; OPU examined whether it could be used artistically. OPU instantiated two half-adders using tempo-based PMAP across orchestral instruments. I provided a protocol to the conductor in advance, describing how to control the orchestra live as a PMAP circuit, and I provided the inputs (3 and 3) 48 hours before the performance. The orchestra was able to perform the calculation, while playing well, and the resulting performance was judged favorably by observers.

References and Notes

- 1 E. Coyle and C. Cullen, "Orchestration within the Sonification of Basic Data Sets," in *Proceedings of the 10th Meeting of the International Conference on Auditory Display* (Sydney: ICAD, 2004).
- 2 T. Hinterberger, "Orchestral Sonification of Brain Signals and Its Application to Brain-Computer Interfaces and Performing Arts," in *Proceedings of the 2nd International Workshop on Interactive Sonification* (York: ISAD, 2007).
- 3 G. Brumfiel, "Dying Stars Write Their Own Swan Songs," National Public Radio (10 January 2014): www.npr.org/sections/thetwo-way/2014/01/10/261397236/dying-stars-write-their-own-swan-songs (accessed 1 November 2018).
- 4 Sophie Hetherington, "CERN Scientists Perform Their Data," CERN (5 April 2017): www.home.cern/about/updates/2014/10/cern-scientists-perform-their-data (accessed 1 November 2018).
- 5 E. Miranda, *Thinking Music* (Plymouth, U.K.: Plymouth Univ. Press, 2014) p. 92.
- 6 P. Votava and E. Berger, "The Heart Chamber Orchestra," *eContact!* **14**, No. 2 (2012).
- 7 S. Freund and J. Mitchell, "A Formal Framework for the Java Bytecode Language and Verifier," in *Proceedings of the 14th ACM SIGPLAN Conference* (New York: ACM, 1999) pp. 147–166.
- 8 "Understanding Full Virtualization, Paravirtualization, and Hardware Assist," VMware Inc. (2008): www.vmware.com/uk/techpapers/2007/understanding-full-virtualization-paravirtualizat-1008.html (accessed 25 July 2019).
- 9 W. Guohui and T. Ng, "The Impact of Virtualization on Network Performance of Amazon EC2 Data Center," in *Proceedings of 2010 IEEE INFOCOM* (San Diego: IEEE, 2010) pp. 1–9.
- 10 J. Garmon, "What Were the Original System Requirements for Windows 1.0?" (November 2010): www.jaygarmon.net/2010/11/what-were-original-system-requirements.html (accessed 1 November 2018).
- 11 A. Kirke et al., "A Hybrid Computer Case Study for Unconventional Virtual Computing," *International Journal of Unconventional Computing* **11**, Nos. 3–4, 205–226 (2015).
- 12 See, for example, L. Spector et al., "Finding a Better-Than-Classical Quantum AND/OR Algorithm Using Genetic Programming," *Proceedings of the 1999 Congress on Evolutionary Computation* (Washington, DC: IEEE, 1999) pp. 2239–2246.
- 13 Kirke et al. [11].
- 14 S. Banik et al., "Affection Based Multi-Robot Team Work," in S.C. Mukhopadhyay and R.Y.M. Huang, eds., *Sensors: Advancements in Modeling, Design Issues, Fabrication and Practical Applications* (Berlin: Springer, 2008) pp. 355–375.
- 15 A. Kirke and E. Miranda, "A Multi-Agent Emotional Society Whose Melodies Represent Its Emergent Social Hierarchy and Are Generated by Agent Communications," *Journal of Artificial Societies and Social Simulation* **18**, No. 2 (2015).
- 16 P. Juslin, "From Mimesis to Catharsis," in D. Miell, R. MacDonald and D. Hargreaves, eds., *Musical Communication* (Oxford: Oxford Univ. Press, 2005) pp. 85–116.
- 17 This is a rather coarse analysis; further details can be found in, for example, Kirke and Miranda [15].
- 18 A. Kirke and E. Miranda, "Pulsed Melodic Affective Processing," *Simulation* **90**, No. 5, 606–622 (2015).
- 19 For more details and examples that use key mode *and* tempo, see Kirke and Miranda [18].
- 20 Kirke and Miranda [18].
- 21 Kirke and Miranda [18].
- 22 A. Kirke and E. Miranda, "Application of Pulsed Melodic Affective Processing to Stock Market Algorithmic Trading and Analysis," *Proceedings of 9th International Symposium CMMR 2012* (London: Springer, 2012).
- 23 Kirke et al. [11].
- 24 A. Kirke, "Orchestral Processing Unit (premiere)," YouTube (1 June 2017): www.youtube.com/watch?v=oPFoDCSXRqI (accessed 1 November 2018). The PMAP movement starts at 6 min 15 sec.

Manuscript received 1 November 2018.

ALEXIS KIRKE is a senior research fellow at the University of Plymouth, England, in the Interdisciplinary Centre of Computer Music Research. He has received PhDs from the Faculty of Arts (2011) and the Faculty of Technology (1997) at the University of Plymouth.