Bioinformatic Response Data as a Compositional Driver

Robert Hamilton[*]

[*]Center for Computer Research in Music and Acoustics (CCRMA), Stanford University
rob@ccrma.stanford.edu

Abstract

This paper describes a software system using bioinformatic data recorded from a performer in real-time as a probabilistic driver for the composition and subsequent real-time generation of traditionally notated musical scores. To facilitate the generation and presentation of musical scores to a performer, the system makes use of a custom LilyPond output parser, a set of Java classes running within Cycling '74's MAX environment for data analysis and score generation, and an Atmel AT-Mega16 micro-processor capable of converting analog bioinformatic sensor data into Open Sound Control (OSC) messages.

1 Introduction

The mapping of fluctuations of a performer's physiological state during the performance of a piece of music to specific compositional parameters can be used to form an intimate relationship between a performer and the structure of a piece of music. Human physiological responses, voluntarily or involuntarily generated and measured with bioinformatic sensors, can be mapped in software to various compositional parameters. With the use of a real-time score generating and display system, physiological response re-interpreted as notated musical gesture can be displayed to the performer for realization. The mapping of predominately involuntary performer excitation levels to pre-composed musical events creates a hybrid improvisational and compositional form allowing both the composer and performer to have input into the final compositional structure.

Rather than use voluntarily generated physiological signals as active controls on the musical output, this system seeks instead to modify compositional content to react to involuntary physiological reaction. In this model, autonomic physiological data acts as a control signal while a performer's physical gesture retains its traditional role as an expressive component of performance. In essence, the compositional decisions made by the composer act as a deterministic filter for the autonomic control signals generated by the performer.

By varying the relationship between physiological reaction and resultant compositional output, different compositional forms can be created. For instance, when an inverse response mapping is applied, where strong sensor readings generate weak or relatively simple compositional structures, a performer's physiological state can potentially be coerced into a less excited state. Similarly, a performer in a stable or less excited state will be presented with more active musical cells, aiming to excite the performer into a more active state. When a direct mapping between physiological state and compositional form is applied, musical output mirrors physiological state, outputting musical cells that reinforce the current state. Additionally, more complex mapping relationships can be defined and implemented with relative ease.

2 Related work

While there exist numerous projects designed to generate musical construct from bioinformatic response, the majority seem to focus on not only conscious or active control by performers/subjects but also on the application of relatively direct mappings of bio-data to musical form. In this approach, control systems allow performers to use voluntary physiological gesture as a direct controller for musical gesture, turning the body into a sophisticated musical control interface (Knapp and Cook 2005; Knapp and Lusted 1990). Even when systems incorporate physiological biofeedback signals, many do so to create direct and controllable mappings between performer and performance.

Work by Dr. Geoffrey Wright and NeuroSonics on Brain Generated Music (BGM) addresses the use of EEG data as a musical driver to create more abstract representations of physiological data (Kurzweil 1999). Indeed, such an approach makes use of a confluence of voluntary and involuntary bioinformatic data, as well as the generation of bioinformatic feedback during a "performance", as subjects listen to music generated by their brain waves in real-time.

In the paradigm of real-time score generation and presentation systems, Kevin Baird's *No Clergy* project addresses many of the same generation and display issues faced here using a network server and web-browser for score display and a Ruby/Python backend (Baird 2005).

Development of the Java probabilistic composition classes used in this project began with the jChing compositional system (Hamilton 2005), designed to model John Cage's chance-based I-Ching compositional techniques (Pritchett 1996).

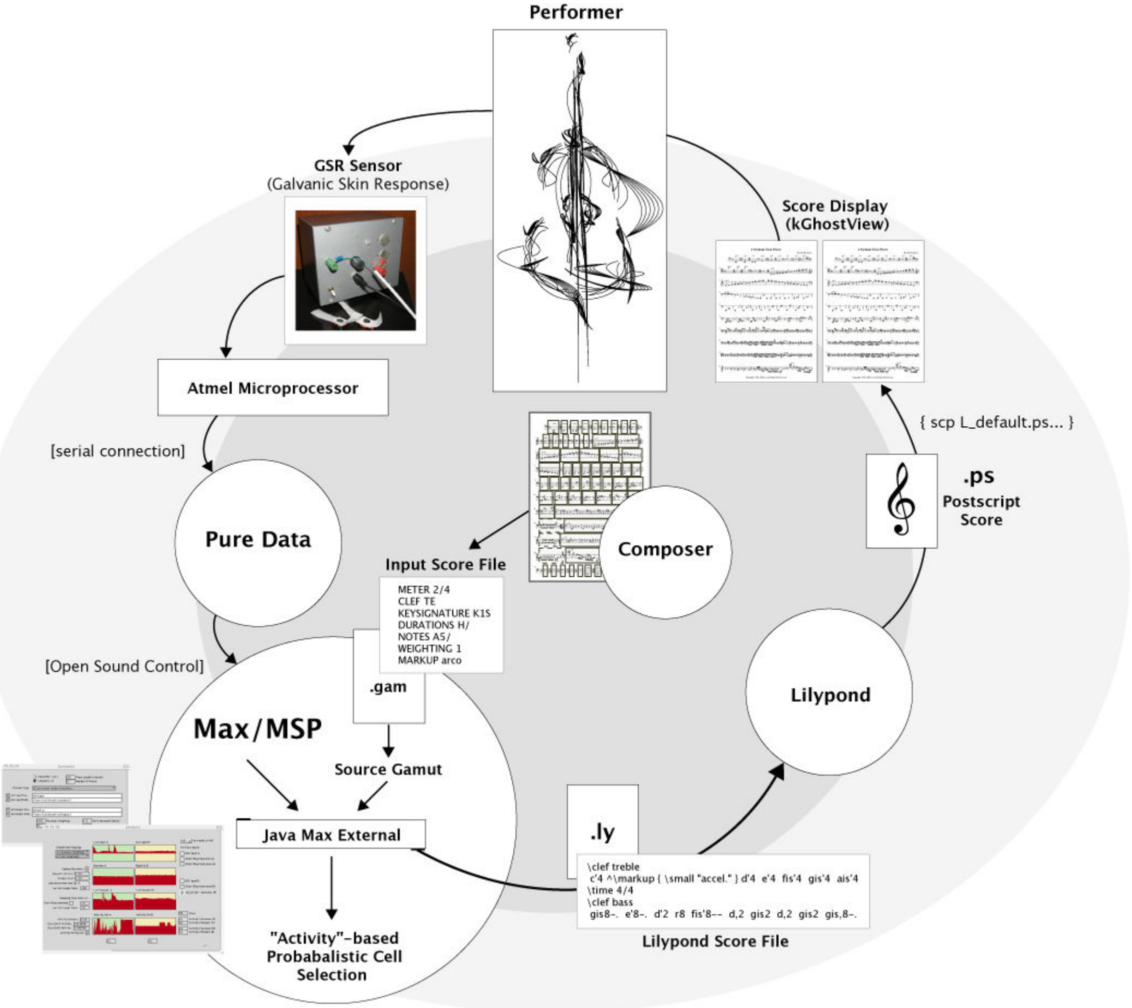
3 System Design

To provide for the collection and processing of incoming data streams as well as for the output of notated musical data, the integration of a number of existing software and hardware platforms was necessary. By standardizing data formats and making use of OSC for data transmission (Wright and Freed 1997), existing open-source software such as LilyPond (Nienhuys and Nieuwenhuizen 2003) Pure Data or PD (Puckette 1996), and GhostView (Thiesen) could be utilized alongside custom Java classes and patches within commercial software such as Max/MSP for data processing, analysis and display.

The basic counter-clockwise workflow of hardware and software data-exchange can be viewed in Figure 1. It should be noted that while the current sensor hardware tracks galvanic skin response, the use of software-based data normalization and the manner in which variance in physiological data is calculated with reference to a performer-specific "baseline" output level makes the implementation of additional sensors relatively simple.

3.1 Galvanic Skin Response (GSR)

For the purpose of system proof-of-concept and initial testing, a simple galvanic skin response (GSR) circuit was used to measure variance in skin conductivity during a musical performance. Galvanic skin response can be described as a measured fluctuation in the electrical resistance of the skin. Using a pair of electrodes usually connected to adjacent fingers, a small electrical current is passed through a subject's skin via one electrode and subsequently measured by another. By measuring changes in skin conductivity relative to applied stimuli, it has been proposed that not only can a subject's emotional or attentional reaction be measured but that the GSR can be considered relatively autonomic and not easily controlled by the subject (Greenfield and Steinback 1972). While a number of pre-recorded data streams of varying physiological data sources have been tested with the system (EKG, EEG), the relative simplicity of implementation of the GSR circuit in a real-time environment led to its use in initial testing and performance situations.



```
METER 2/4
CLEF TE
KEYSIGNATURE K1S
DURATIONS H/
NOTES A5/
WEIGHTING 1
MARKUP arco
```

```
\clef treble
 c'4 ^\markup { \small "accel." } d'4  e'4  fis'4  gis'4  ais'4
\time 4/4
\clef bass
 gis8-.  e'8-.  d'2  r8  fis'8--  d,2  gis2  d,2  gis2  gis,8-.
```

Figure 1. Hardware/Software System Workflow

3.2 Hardware

Performer GSR levels are monitored with the use of an analog GSR circuit connected to an ATMEL AT-MEGA16 microprocessor. The microprocessor, running custom C-code, formats the ADC converted voltage values for output using the OSC data protocol. The GSR circuit used in the initial testing and development of the system was designed and built by Jay Kadis of Stanford University's CCRMA (see Figure 2).

While a standard methodology for the measurement of GSR data calls for the attaching of conductive sensors to the fingers of a subject (to take advantage of the greater amount of resistance fluctuation in finger tissue), as musical instruments tend to be performed using the fingers and hands, to reduce data artifacts due to physical displacement of finger mounted GSR sensors during performance, a pair of sensors were instead attached to the performer's toes. Non-performance tests of GSR fluctuations comparing toe and finger placements showed similar results for either location.

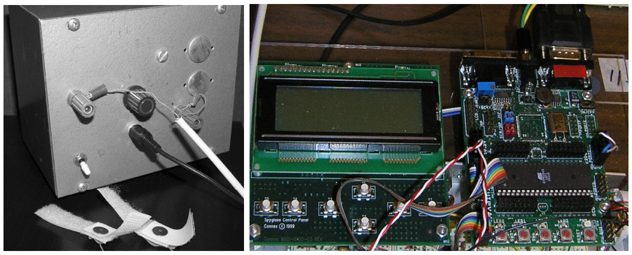
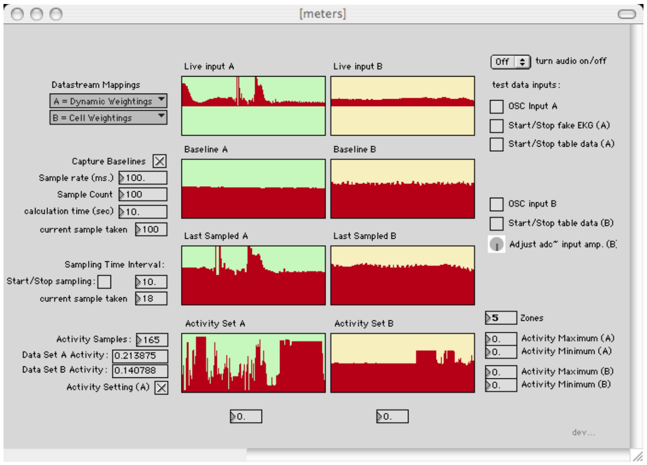


Figure 2. GSR circuit box with finger/toe sensors and Atmel microprocessor

3.3 Software workflow

OSC formatting and routing objects[1] running in a PD patch on a computer with a serial connection to the microprocessor receive a steady stream of voltage values converted by the microprocessor's ADC. The PD patch simply forwards these unprocessed converted voltage values over a local OSC connection to a Max/MSP patch for normalization, data processing and analysis.

The core of the system lies in a set of Java classes designed to take pre-composed musical cells as input, define probabilistic relationships between each cell's pitch "activity" (defined as an aggregate of semitone pitch-steps throughout the cell) and to output musical cells in the LilyPond (.ly) data format. These Java classes are instantiated within the Max/MSP environment, allowing for real-time interaction between the data streams and the classes, as well as a fully-featured system GUI for real-time control and data representation (see Figure 3), without a prohibitive development timeframe.



[meters]

Datastream Mappings

A = Dynamic Weightings

B = Cell Weightings

Capture Baselines ☒

Sample rate (ms.) 100.

Sample Count 100

calculation time (sec) 10.

current sample taken 100

Sampling Time Interval:

Start/Stop sampling: □ 10.

current sample taken 18

Activity Samples: 165

Data Set A Activity: 0.213875

Data Set B Activity: 0.140788

Activity Setting (A) ☒

Off turn audio on/off

test data inputs:

□ OSC Input A

□ Start/Stop fake EKG (A)

□ Start/Stop table data (A)

□ OSC input B

□ Start/Stop table data (B)

Adjust adc~ input amp. (B

5 Zones

0. Activity Maximum (A)

0. Activity Minimum (A)

0. Activity Maximum (B)

0. Activity Minimum (B)

Figure 3. Max/MSP GUI

By leveraging Max/MSP's ability to interact with the BSD Unix shell of an Apple computer running OS X (using the "shell" object), programmatic shell calls are made to both LilyPond (for postscript score compilation) and to SCP for data-transport from the processing machine to a locally-networked display terminal running a postscript viewing application such as kGhostView. This modular approach for processing and data display creates an extremely flexible workflow which can be adapted to run on a number of system platforms and software applications.

4 Data processing

Signal levels taken from the GSR sensors are recorded into sample buffers, creating a windowed data set representing a fluctuation of input signal over a given time frame. Both the frequency of sampling and the number of samples comprising a window are configurable using the Max/MSP GUI. Each windowed data set is first compared against a baseline data set – taken before the start of the performance with the performer in a relatively stable physiological state – and subsequently used to generate a single scaled "activity" value, representing the variation in amplitude of each input sample in relation to the amplitude of its previously recorded sample.

After establishing baseline values for input data, it is useful to establish relative maximum and minimum data values for computation. The subsequent range of possible or probable data values can be subsected into any number of "activity zones" by setting a "zone" value in the GUI. This effectively creates *n*-number of equally-sized ranges of activity for both the musical Gamut Squares as well as for the input GSR data sets. In this manner, increases or decreases in precision can be set to account for more or less active data streams.

[1] *OSC, OSCroute,* and *dumpOSCSerial* objects by Matt Wright et al.

4.1 Musical Activity

Pre-composed musical data cells comprised of musical note and articulation data called “Gamut Squares” are loaded into memory and used to create a detailed hierarchical musical data structure in Java (Hamilton 2005). In a manner similar to the aforementioned signal “activity” metric, the intervallic distance between adjacent notes in a given Gamut Square is used to calculate a melodic or interval-based activity cell value (see Figure 4).

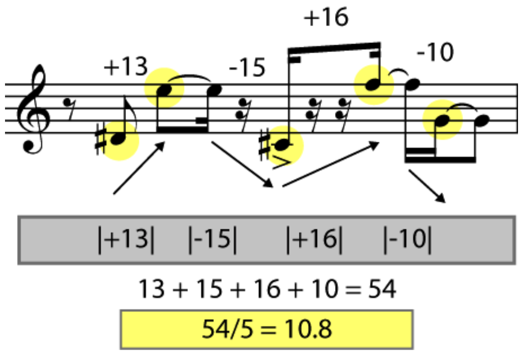


Figure 4. Melodic Activity as adjacent semitone distance

By tracking pitch change from note to note across the duration of a given musical phrase, we can calculate an activity value for the melodic content of each excerpt. The simplest method for calculating melodic activity ignores harmonic roles of notes – each note’s position within a classically defined harmonic structure – and instead concentrates on the vertical pitch motion from note to note. (in this context a note is defined as an independent note onset and offset).

Given Figure 4, if the distance from adjacent diatonic pitches is defined as a value of one, this example contains pitch-to-pitch stepwise motions of 13 (D#4->E5), 15 (E5->C#4), 16 (C#4->F5), and 10 (F5->G4) semitones. It should be noted that using this method of defining individual notes, rests are ignored and only note onsets are taken into account. In this manner, the raw melodic complexity value sums to 13+15+16+10 or 54 semitones. To account for varying numbers of notes from excerpt to excerpt, we can divide this value by the total number of note onsets found in the excerpt and receive an average level of melodic complexity for the given excerpt of 10.8 semitones.

The use of additional activity metrics, including rhythmic activity and harmonic activity are currently under investigation. Of particular interest is the adaptation of a rhythmic complexity metric such as one outlined by Pressing (1998) where each note’s position in a measure relative to the perceived beat can be weighed towards a cumulative phrase-level complexity value. By combining melodic, harmonic and rhythmic complexity calculations, it holds that a more accurate assessment can be made of the perceived complexity or activity of a given musical excerpt.

4.2 Note cell selection

Selection of musical data cells for output occurs by simply correlating GSR activity readings with musical activity values from respective zones. Cells from the desired activity zone are given a GUI-defined high probability of selection from the overall set of musical cells. Cells from other activity zones are given a correspondingly low probability of selection. Cells are then selected from this macro set of probability-scaled data cells and set into a structure for subsequent output. By selecting musical cells using probabilities rather than by directly selecting cells based on their activity levels the level to which a performer’s bioinformatic data can shape the composition is left inexact. In this manner, the composition can always embark on unexpected directions irregardless of its relationship to the performer.

4.3 Selection of note dynamics

While the current system implementation uses physiological data primarily to drive the selection of note cells, other compositional aspects such as dynamic and articulation can be mapped to data sources and selected probabilistically. In tests using pre-recorded or modeled data-streams (EEG, EKG), as well as tests using the live GSR stream, the windowed activity reading was mapped to musical dynamic selection on a note-by-note basis. Mappings are currently applied directly, where a greater activity reading leads to an increase in probabilistic weighting for dynamic values in corresponding activity zones. In this model, a louder dynamic, such as ***ff*** is regarded as having a greater activity than a softer dynamic, ***pp***.

4.4 Data formatting and output

When a user-defined threshold of beats of music has been reached, selected musical cells are converted into the LilyPond musical score data format using a custom-written LilyPond parser and output to a text-file. Using Unix shell calls invoked from Max/MSP, this .ly file is then processed by LilyPond into a standard .ps postscript file and moved to a directory being “watched” by a GhostView postscript viewer application such as kGhostView. Any change to the file’s modification date results in a refresh of the kGhostView display. The display is being presented on a computer monitor to the performer who is then able to perform the recently generated musical phrase. In recent performances with the system, it has been useful to generate two postscript output files for alternate sets of output data, and to use a vertically-aligned pair of kGhostView display windows to alternately update sections of the composition. In this manner, one window of display information can be updated while the performer is still playing the previously rendered and displayed window.

5 Performance practice

As an initial test of the system, a series of performances of probabilistically generated cell-based musical compositions driven by fluctuations in a performer's GSR were given in the Fall of 2005 at Stanford University's CCRMA. Cellist Colin Oldham performed a suite of compositions where short pre-composed phrases of music of varying complexity and pitch variance were dynamically selected and presented for performance based on the real-time windowed analysis of his fluctuating GSR levels. During these initial performances it became clear that while the basic nature of real-time composition necessitated a performer with excellent sight-reading abilities, the pre-composed nature of this cell-based compositional approach allowed the performer to study and practice the source material before performance, greatly reducing performance error due to surprise. Additionally, by viewing score data in two independently-refreshed windows, the performer was able to read ahead while performing less-challenging materials, again reducing possible performance error.

6 Conclusions

From early testing and performances it is clear that while the concept of physiological data as a compositional driver seems viable, great care must be given in choosing bioinformatic sensors so that fluctuations in body state are consistent and to an extent predictable within a given range. While the data generated by the GSR sensor shows evolution and gradual change over longer time periods (in the n-seconds range), GSR tracking failed to show adequate fluctuation following short-term musical events (in the n-milliseconds range) without extreme stimulation. State changes as measured by GSR seem to develop over longer periods of time rather than discretely measurable periods and might be a better match with other compositional parameters, such as part density in a multi-voiced work.

Future directions for the project include development and integration of additional data sensors, such as EKG, EEG or body-temperature sensors, which should provide a more consistently active state across shorter time frames. By combining a variety of sensors, a more accurate measurement of physiological state and its reaction to musical events should be possible. Additional development of small wired or wireless biosensors capable of transmitting data over standard protocols (i.e. USB, Bluetooth, wireless LAN) is currently under investigation.

Additional testing covering a range of instrumental performers and compositional excerpts is being planned. In doing so, more appropriately reactive mappings between various compositional constructs and performer state should become clear. Similarly, the implementation of additional musical activity metrics should provide a more comprehensive cognitive assessment of perceived activity in composed musical phrases.

7 Acknowledgments

Thanks to Jay Kadis for his GSR signal processing circuit. Thanks also to Jonathan Berger, Bill Verplank, Max Mathews, Colin Oldham and Carr Wilkerson.

References

Baird, K. 2005, No Clergy:Real-Time Generation and Modification of Music Notation. In *Proceedings of the 2005 SPARK Festival*, Minneapolis: University of Minneapolis.

Greenfield, N.S. and Steinback, R.A. 1972, *Handbook of psychophysiology*. New York: Holt, Rinehart & Winston.

Hamilton, R. 2005, The jChing: an Algorithmic Java-Based Compositional System. In *Proceedings of the 2005 International Computer Music Conference*, Barcelona: International Computer Music Association.

Knapp, R.B. and Cook, P.R. 2005, The Integral Music Controller: Introducing A Direct Emotional Interface To Gestural Control of Sound Synthesis. *Proceedings of the 2005 International Computer Music Conference,* Barcelona: International Computer Music Association.

Knapp, R.B. and Lusted, H.S. 1990, A Bioelectric Controller for Computer Music Application*s*, *Computer Music Journal*, 14 (1), 42-47.

Kurzweil, R. 1999, *The Age of Spiritual Machines: When Computers Exceed Human Intelligence*, New York: Viking Penguin Press.

Nienhuys, H. and Nieuwenhuizen, J. 2003, LilyPond, a system for automated music engraving. In *Proceedings of the XIV Colloquium on Musical Informatics*, Firenze, Italy.

Pressing, J. 1998, Cognitive complexity and the structure of musical patterns. In *Proceedings of the 4th Conference of the Australasian Cognitive Science Society*, Newcastle, Australia.

Pritchett, J. 1996, *The Music of John Cage*, New York: Cambridge University Press.

Puckette, M. Pure Data. 1996, *Proceedings of the International Computer Music Conference,* San Francisco: International Computer Music Association.

Thiesen, T., GhostView, http://www.cs.wisc.edu/~ghost/

Wright M. and Freed, A. 1997, Open Sound Control: A New Protocol for Communicating with Sound Synthesizers. In *Proceedings of the International Computer Music Conference,* Thessaloniki: International Computer Music Association.