Bioinformatic Feedback: performer bio-data as a driver for real-time composition

Robert Hamilton
Center for Computer Research in Music and Acoustics (CCRMA)
Department of Music, 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.

Keywords
Bioinformatics, composition, real-time score generation.

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 physiological sensors, can be mapped in software to 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 interpretation.

Continuing performance of this newly generated musical notation data can result in a bioinformatic feedback loop, within which the performer's physiological state reacts to a musical abstraction of the most recent physiological state. 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. When an inverse response mapping is applied, where strong sensor readings generate weak or relatively simple compositional structures, a performer's physiological state can 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. Conversely, when a direct mapping between physiological state and compositional form is applied, musical output mirrors physiological state, out-putting musical cells that reinforce the current state.

2. 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 platforms with was necessary. By standardizing data formats and making use of OSC [11] for data transmission, existing open-source software such as LilyPond [7], Pure Data (PD) [9], and GhostView [10] could be utilized alongside commercial software such as Max/MSP for data processing, analysis and display (see Figure 1).

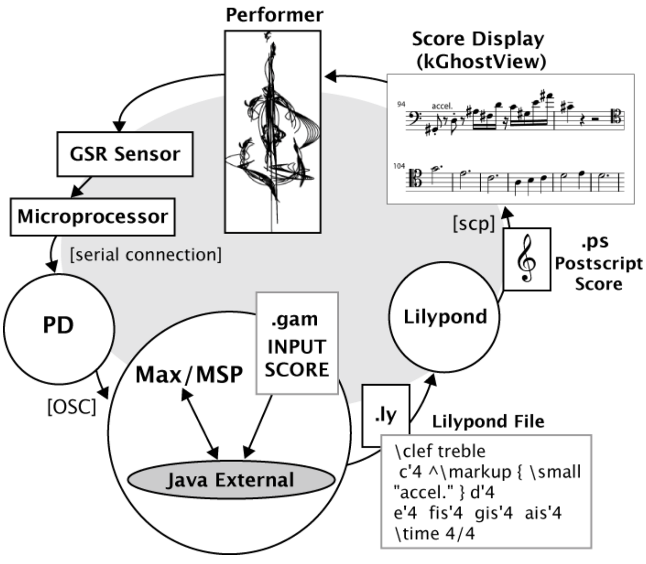

Figure 1. System Data-Flow

2.1 Galvanic skin response (GSR)

For the purpose of system proof-of-concept and initial testing, a simple Galvanic skin response (GSR) sensor was used to measure variances in skin conductance during a musical performance. Galvanic Skin Response can be described as a measured fluctuation in the electrical resistance of the skin. 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 [2]. 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 sensors in a real-time environment led to their use in initial testing and performance situations.

2.2 Hardware

GSR data is extracted from a performer with the use of a custom analog GSR circuit connected to an ATMEL AT-MEGA16 microprocessor over a serial connection. The microprocessor is running custom C-code capable of outputting voltage readings in the OSC data protocol as a scaled stream of values appropriate for analysis. 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.

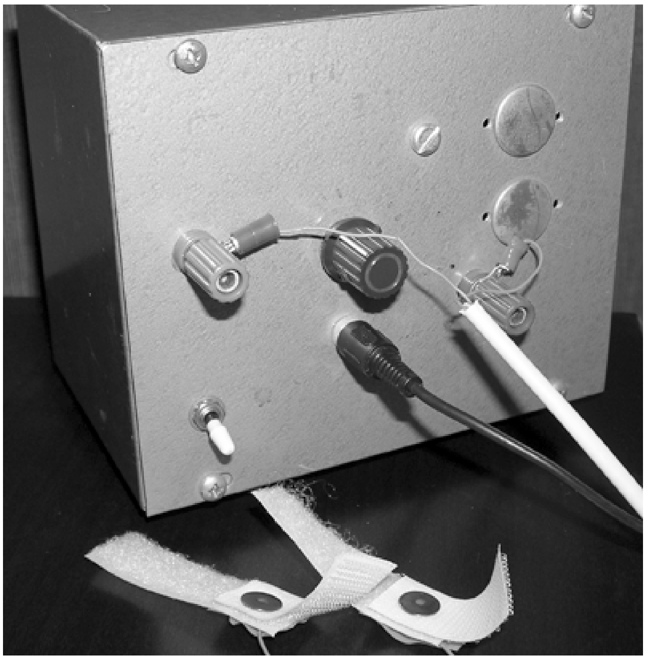

Figure 2. GSR circuit box and finger/toe sensors

Figure 3. Max/MSP GUI

2.3 Software workflow

OSC formatting and routing objects[1] running in a PD patch on a computer with a serial connection provide a steady data stream of resistance values converted from the analog realm to the digital realm by the microprocessor's ADC. In this case, the PD patch simply makes use of objects capable of reading serial data to act as a signal router, sending raw digital resistance values over a local OSC connection to a Max/MSP patch where data processing and analysis take place.

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 pitch-steps through a cell) and to output musical cells in the Lilypond (.ly). 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.

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 and to SCP for data-transport from the processing machine to a locally-networked display terminal running a postscript viewing application such as GhostView. 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.

2.4 Data processing

Signal levels taken from two GSR sensors attached to a performer's fingers or toes 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.

---

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

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. In a manner similar to the aforementioned signal "activity" metric, the fluctuation of pitch in a given Gamut Square is used to calculate a musical activity cell. A more detailed description of cell-based compositional techniques and the particular data formatting for cell input is available in [3].

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.

2.5 Pitch/rhythmic cell selection

Selection of musical data cells for output occurs by 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.

2.6 Dynamics selection

While the current implementation of the system 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 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***.

2.7 Data formatting and output

When a user-defined threshold of beats of music has been generated, the 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 application, refreshing the viewer. The viewer 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.

3. 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 Center for Computer Research in Music and Acoustics (CCRMA). Cellist Colin Oldham, a visiting scholar at CCRMA, 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 separately-refreshed windows, the performer was able to read ahead while performing less-challenging materials, again reducing possible performance error.

4. 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 [4][5]. 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 [6]. 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 [1] 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.

Additionally, initial development of the Java probabilistic composition classes used in this project began as the core to the *jChing* compositional system [3], itself designed as a software model of John Cage's influential *I-Ching* compositional techniques [8].

5. 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.

To better determine the extent to which compositional cells can affect performers of various instruments, 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 the Bioinformatic Feedback system should become clear.

6. ACKNOWLEDGMENTS

Many thanks to Jay Kadis for the generous donation of his GSR signal processing circuit. Thanks also to Jonathan Berger, Bill Verplank, Max Mathews, Colin Oldham and Carr Wilkerson for their suggestions and support.

7. REFERENCES

[1] Baird, K. No Clergy:Real-Time Generation and Modification of Music Notation. In *Proceedings of the 2005 SPARK Festival*, (Minneapolis, USA, 2005).

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

[3] Hamilton, R. The jChing: an Algorithmic Java-Based Compositional System. In *Proceedings of the 2005 International Computer Music Conference (ICMC 2005)* (Barcelona, Spain, September 5-9, 2005).

[4] Knapp, R.B. and Cook, P.R. The Integral Music Controller: Introducing A Direct Emotional Interface To Gestural Control of Sound Synthesis. *Proceedings of the 2005 International Computer Music Conference (ICMC 2005)* (Barcelona, Spain, September 5-9, 2005).

[5] Knapp, R.B. and Lusted, H.S. A Bioelectric Controller for Computer Music Applications, *Computer Music Journal*, 14 (1), 1990, pp. 42-47.

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

[7] Nienhuys, H. and Nieuwenhuizen, J. LilyPond, a system for automated music engraving. *Proceedings of the XIV Colloquium on Musical Informatics (XIV CIM 2003)*, Firenze, Italy, May 2003.

[8] Pritchett, J. *The Music of John Cage* (New York: Cambridge University Press, 1996)

[9] Puckette, M. Pure Data. *Proceedings of the International Computer Music Conference, (ICMC 1996)* (San Francisco, California, 1996) pages 269-272.

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

[11] Wright M. and Freed, A., Open Sound Control: A New Protocol for Communicating with Sound Synthesizers. In *Proceedings of the International Computer Music Conference (ICMC 1997)*, (Thessaloniki, Greece, 1997).