Design Concepts and Control Strategies for Interactive Improvisational Music Systems

THOMAS CIUFO

Special Studies/Music

Brown University

1 Young Orchard Ave, Providence, RI

USA

Thomas_Ciufo@Brown.edu http://www.brown.edu/Departments/Music/ciufo/

Abstract:

This paper will focus on two important and under explored areas pertinent to interactive, improvisational performance. First, an overview of basic control strategies will be outlined. Next, techniques for acquiring control data from a real-time audio stream, and approaches to mapping this data to interactive digital signal processing functions, will be explored. Examples from a current work by the author (*Eighth Nerve* for prepared electric guitar and computer) will be referenced. This piece uses a combination of audio analysis techniques and direct control from sensors (built into a custom hybrid guitar) to drive real-time, interactive sound transformations.

Key-Words: Interactive systems, improvisation, audio analysis, mapping, gestural control.

1 Introduction: The Demands of Improvisation

While there is a growing body of research surrounding interactive systems, [1] mapping, [2] gestural control, [3] controller design, [4, 5] and composed instruments [6, 7] this paper will explore strategies especially applicable to interactive, improvisational performance. As Robert Rowe states, “Interactive improvisation poses perhaps the greatest challenge for machine musicianship. Here the machine must contribute a convincing musical voice in a completely unstructured and unpredictable environment” [1]. While there are many approaches to improvisation (whether idiomatic or non-idiomatic [8]), it is widely acknowledged that improvisation places complex and unique demands on interactive computer music systems. As David Gamper points out, “The performer may want to be able to do anything possible at any possible time” [9]. This openness and spontaneity means that interactive systems must be designed to support expressive interaction and rapid changes in direction, and produce stimulating and unexpected output. Several control strategies and design approaches appropriate to these goals will be described.

2 Control Strategies: Overview

Before delving into the specifics of interactive systems design and composition, it is important to take a broad look at control strategies and the relationships they imply. Available control options cover the gamut from deterministic (such as simple control surfaces or recalling presets) to unpredictable (using chaos or random generators) with many hybrids in-between. Control data can be created in advance, generated in real-time by a performer, generated by an algorithm, or combine all of these approaches. Specific questions that will shape the overall design of the instrument and improvisation should include:

- What type of relationship with the instrument or system is desired? (Is the system an extension of the instrument/controller or does the system play the role of another performer/improviser, or some other role?)
- What level of control does the performer desire? (Is this going to be a democracy, dictatorship, or something in-between?)
- Should the control systems operate on a micro level, simply steer the overall direction of the improvisation or some combination of the two?
- How much explicit control is desirable and how much feedback or unpredictable stimulus should the system offer?
- If a gestural controller or hybrid instrument is involved, what are the primary control acquisition issues?
- How can a balance between extensibility and playability be achieved?

These questions have many possible answers and will vary from piece to piece or even within a given system or performance. Nevertheless, the contemplation of these questions will greatly influence the design, implementation, and sonic outcome of an interactive system. The primary continuum that I would like to highlight is between deterministic and indeterministic. Joel Chadabe describes this continuum as follows: “The functioning of any particular electronic musical instrument can be placed on a taxonomic marked by deterministic functioning at, let’s say, the left, and indeterministic functioning at the right. At the leftmost extreme of the line, a deterministic instrument is defined by the complete predictability of its output relative to a performer’s controls. At the rightmost extreme, an indeterministic instrument outputs a substantial amount of unpredictable information relative to a performer’s controls. In working with such an instrument, a performer shares control of the music with algorithms as virtual co-performers such that the instrument generates unpredictable information to which the performer reacts, the performer generates control information to which the instrument reacts, and the performer and instrument seem to engage in a conversation” [10]. Chadabe refers to this type of instrument as an ‘interactive instrument’.

As an improviser, this idea of interaction and shared control is very enticing, as it places me within the center of an ever unfolding, unpredictable, present moment. It allows me to build systems that take on a personality or behavior of their own. By improvising within these sound environments, I can interact with what I hear, and the outcome is only partially known, setting up more of a dialogue than a recitation. George Lewis discusses the notion of conversation or dialogue in reference to his Voyager system. “As I have observed elsewhere, interactivity has gradually become a metonym for information retrieval rather than dialogue, posing the danger of commodifying and ultimately reifying the encounter with technology” [11]. Throughout this paper, I will be approaching interaction from various points along the continuum between deterministic and indeterministic, between predictability and surprise.

2.1 Explicit or Manual Control

At times, explicit or direct control may be necessary or advantageous. Direct control can be appropriate for parameters such as main output volume, system on/off, or other parameters that need to be controlled with precision and/or predictability. In an improvisational setting, explicit controls may be of limited use as they often imply that the designer knows specifically what will need to happen ahead of time. Obviously some level of planning and design is necessary for interactive instruments; but for improvisation, a high degree of flexibility is desirable, so the notion of presets may be of limited use. When needed, manual control can be accomplished in a number of different ways.

- Use manual controls, such as MIDI faders, directly mapped to one or a few specific destinations.
- Use simple, precise sensors, such as potentiometers or switches, directly mapped to one or a few specific destinations.
- Recall or trigger pre-made control functions, such as presets or break point functions.

2.2 Generative or Indeterminate Control

Towards the other extreme are control systems that run on their own, with little or no influence from the performer or the rest of the system. Robert Rowe defines generative techniques as those where “the output of a compositional formalism is derived solely from the operation of the formalism itself, possibly aided by stored tables of material” [12]. Computers are adept at random and chaotic generation, making these techniques available for real-time interactive systems. These types of processes can be used at different levels within an interactive instrument. They can be used to animate micro level controls or higher level macro controls. By using generative algorithms at various points in a system, complex and highly unpredictable outcomes can be produced. Some of the generative techniques currently available include [13]:

- simple random number generators, non-repeating random, random walks, and colored noise generators
- generate and test, tendency masks, look up tables, and markov chains
- stochastic models, fractal patterns, and chaos generators

2.3 Machine Musicianship and Listening

In the following quote, Robert Rowe describes a primary motivation for machine musicianship. "Designing computer programs that will recognize and reason about human musical concepts enable the creation of applications for performance, education, and production that resonate with and reinforce the basic nature of human musicianship" [1]. While attempting to design and use machines to emulate human behavior raises many theoretical questions, [14, 15] the research and application of machine musicianship techniques provides many useful tools for the interactive instrument designer. A topic this broad is well beyond the scope of this paper, so I will highlight only a few key approaches pertinent to improvisation. These include the use of rule based and learning based systems (such as neural networks, genetic algorithms, and artificial agents) for listening to and analyzing live performance data. A wide range of musical and/or perceptual features can be analyzed, such as chord type, key, meter and beat tracking, event density, rest detection, pattern matching, and so on. Based on this analysis data, certain decisions can be made by the computational system that result in the playback, transformation, and/or generation of musical materials.

Many of the interactive systems for improvisation have used MIDI controllers or pitch-to-MIDI converters to deliver a greatly simplified representation to the software system [11]. Additionally, many systems operate on a note or phrase level and use western music materials such as pitch, melody, harmony, and rhythm as the primary building blocks. Because my primary interest lies in the nature of the sound itself, I have had to look (and listen) for other ways to design and interact with improvisational systems.

3 Audio Stream Analysis

Because my primary improvisational material is timbre changing over time, I have been designing systems that analyze the audio stream directly as it is performed. Wanderley discusses this in the context of sensor use in gestural controllers, stating, "Direct acquisition is performed by the use of different sensors to capture performer actions. As opposed to direct acquisition, indirect acquisition provides information about performer actions from the evolution of structural properties of the sound being produced by an instrument" [16]. This approach has several important innovators including Puckette, [17] Lippe, [18] Settle, [19] and Machover [20]. My attraction to audio stream analysis comes from my interest in making and using acoustic sounds in performance, combining the processing power of the computer with the richness and immediacy of the acoustic world. It seems appropriate to attempt to derive aspects of gesture directly from the performed sound.

3.1 What Can Be Analyzed?

The question of what can be analyzed is difficult to answer because the effectiveness of audio analysis is highly dependent on context and the specific demands placed on the analysis data. For example, achieving accurate and repeatable results from the real-time analysis of each instrument in a full orchestra would be extremely difficult, especially regarding pitch data [21]. By comparison, acquiring interesting and usable data from a solo instrument is much more realistic. Given my interest in shared control and dynamic systems, I only need the analysis to represent or reflect the various types of gestures in the audio stream, so complete accuracy is not required. In my current interactive improvisations, I am using the analyzer~ object by Tristan Jehan [22] running in Max/MSP. Analyzer~ estimates the following features:

- pitch
- loudness
- brightness
- noisiness
- onset detection
- bark scale decomposition
- sinusoidal decomposition

In certain instances, a patch may utilize two complete analyzer~ objects: one on the input signal and one somewhere near the output. This way the computer can have some idea of what is coming into the system while also keeping track of the systems output. This is similar to Rowe's use of a second analysis stage, which he refers to as the critic [12].

3.2 What To Do With All This Data?

What to do with this data is a difficult question that can have many interesting answers depending on the design and context of the system. I will start by discussing a few general strategies and finish with examples from a specific piece. One basic approach is to examine the analysis data to see what can be learned or at least inferred. Many of these techniques have been addressed elsewhere in the context of analyzing MIDI performance data [12, 23].

- Various inferences can be made from pitch analysis, such as interval measurement, rate of melodic change, and tonal center. Pitch data is of limited use in my work, as many of my sound sources have highly unstable pitch. In these cases, I am especially thoughtful about the mapping of these pitch inferences. In some cases, it may be useful to be rather crude and quantize unstable pitch into high, medium, and low ranges, or to simply measure the relative pitch deviation.
- Loudness is probably the easiest parameter to utilize because it relates clearly to the energy being presented to the system. It is also possible to detect rests and make inferences about articulation from changes in the loudness value.
- Brightness is another very useful parameter because it can track a wide range of sound sources and represents an important timbral variable, i.e., the spectral centroid of the given frame.
- Noisiness is also a useful timbral indicator, especially in conjunction with brightness. Noisiness can also be utilized to gate other data streams, for example bypassing the pitch data if the input signal is above a certain noisiness threshold.
- Onset detection can be used directly for event detection or segmentation, or timed to calculate event duration and rest status. Analysis data can be used in logic statements, such as: if time since last attack is more than x seconds and current loudness is less than y dB, then send 'rest' message.
- Sinusoidal decomposition can have many uses including general timbre analysis and inharmonicity measurement, but it is especially useful for resynthesis. In most cases, real-time additive resynthesis will be somewhat crude, but it can still be useful and expressive.

3.3 Data Conditioning

In most cases the analysis data will require further conditioning or logic structures that interpret the data. The most basic conditioning is probably averaging and scaling. It is often desirable to have both a continuous, instantaneous output and one that is averaged over a given period of time. This can be done using simple averaging, histograms, or other types of statistical analysis. It is also convenient to scale the analysis data to more compatible ranges, but care must be taken to keep the resolution appropriate to the destination. As mentioned previously, combinations of simple conditional statements and logic structures can be very useful. These include conditionals such as if-then-else, greater than, less than, equal, and other mathematical expressions that can be used to manipulate the analysis data.

3.4 Challenges of Audio Analysis

Even with careful application of the above-mentioned techniques, real-time audio stream analysis can be challenging. Depending on the complexity of the audio signal, particularly the level of inharmonicity and noise, the analysis data can be relatively error prone, most often in the area of pitch tracking. In many cases these so-called problems can be minimized by careful conditioning and mapping of the analysis data, or even designed into the system as another unpredictable feature. Another challenge is the inherent feedback encountered when using a microphone as the transducer for audio analysis. In most cases, the input microphone will also be listening to the output of the audio system, delayed in time and at a lower level than the intended input source. Techniques for reducing this feedback include using close source to microphone placement, using highly directional microphones, and carefully placing the speakers with dispersion patterns in mind. Contact transducers can often be used, but care must be exercised, as their audio quality is usually not ideal. One common technique involves the use of contact transducers for the analysis input while using open-air microphones to capture the signal for amplification and further DSP manipulations.

4 Application: Hybrid Systems

In this paper, I have presented several different control strategies, including the use of physical sensors and audio stream analysis. I am in agreement with Wanderley that hybrid systems using both of these control methods "can potentially outperform single acquisition systems" [16]. However, I would add a third element to the equations: chaotic or randomly generated control elements.

I am currently working on a hybrid approach for the improvisational performance piece-instrument-system *Eighth Nerve,* for prepared electric guitar and computer. This piece uses a custom electric guitar that has been fitted with sensors including two rotary potentiometers, two momentary switches, one toggle switch, and a pressure sensor that runs the length of the neck of the guitar. Work is also under way on a two-dimensional tilt sensor that will continuously report the angle of the guitar. This interactive system uses physical sensor data, along with an array of audio stream analysis data and randomly generated data to control a range of real-time DSP manipulations. These sound transformations are applied to the performed sounds of the prepared guitar, and include spectral domain filtering, multi-part looping and time expansion and compression, as well as simple additive resynthesis using a variety of waveforms. Additionally, the guitar sound is convolved against itself using random sampling of the performed sound. This system uses performer driven random generators and multiple nested feedback loops that further complicate the predictability of the sound output. The complete design, mapping, and DSP architecture are beyond the scope of this paper, but a few examples will illustrate the overall approach.

- The spectral filter/looper can initiate recording into one of four randomly chosen buffers, either via manual control (momentary switch) or by onset detection. These buffers use looping append recording to create unpredictable discontinuities. The playback rate is dynamically controlled using scaled input brightness or noisiness. One of seven spectral filter shapes is randomly chosen using onset detection with dynamic interpolation between the current and previous shape, driven by input amplitude, brightness, or noisiness.
- The convolution processor uses an amplitude threshold trigger to record into one of four randomly chosen short-term buffers. Each newly detected attack initiates playback of a randomly selected buffer with the playback speed derived from a combination of amplitude, brightness, noisiness, and pressure on the neck sensor. These buffered samples are convolved against the real-time guitar signal.
- The frequencies of the eight most prominent sine waves are analyzed and then used for a crude form of additive resynthesis. The amplitudes from the analysis are discarded and simple attack/decay envelopes are used for the resynthesis, with envelope times adjusted, based on the loudness of the input signal. The resynthesis waveform is changed periodically (based on onset detection) with rest detection used to change waveforms in-between gestures.
- System memory (looper and convolution buffers) can be incrementally cleared either manually (using repeated presses on a momentary switch) or using specific pitches as triggers. Spacialization location is chaotically driven with the speed modulated by the amplitude of the specific voice.
- Other system controls include a master volume (rotary pot) and two system wide feed-back/chaos controls (rotary pot with toggle) that collect many signal outputs, process them, and send them back to many signal inputs.
- Additional research is underway to implement a two-dimensional tilt sensor that will control different parameters depending on context dependent system parameters.

5 Summary and Future Work

I have attempted to present a general overview of interactive control strategies applicable to improvisational performance including manual control, algorithmic and random approaches, machine listening, and specifically audio stream analysis. Future work includes continued development of hybrid systems with a focus on more sophisticated interpretations of audio analysis data, including expanded use of artificial intelligence techniques. Real-time audio analysis can provided a wealth of useful information about performed sound gestures and is especially beneficial in an improvisational setting where the sonic direction is not known ahead of time.

References:

[1] R. Rowe, Machine Musicianship, MIT Press, 2001.

[2] M. Wanderley, A. Hunt, M. Paradis, The Importance of Parameter Mapping in Electronic Instrument Design, International Conference on New Interfaces for Musical Expression, 2002.

[3] M. Wanderley, M. Battier, Eds, Trends in Gestural Control of Music, IRCAM, 2000.

[4] D. Wessel, M. Wright, J. Schott, Intimate Musical Control of Computers with a Variety of Controllers and Gesture Mapping Metaphors, International Conference on New Interfaces for Musical Expression, 2002.

[5] P. Cook, Principles for Design of Computer Music Controllers, ACM CHI Workshop in New Interfaces for Musical Expression, 2001.

[6] C. Bahn, D. Truman, Interface: Electronic Chamber Ensemble, ACM CHI Workshop in New Interfaces for Musical Expression, 2001.

[7] N. Schnell, M. Battier, Introducing Composed Instruments, Technical and Musicological Implications, International Conference on New Interfaces for Musical Expression, 2002.

[8] D. Bailey, Improvisation: Its Nature and Practice in Music, Da Capo Press, 1993.

[9] D. Gamper with P. Oliveros, A Performer-Controlled Live Sound-Processing System: New Developments and Implementations of the Expanded Instrument System, Leonardo Music Journal Vol. 8, 1998.

[10] J. Chadabe, The Limitations of Mapping as a Structural Descriptive in Electronic Instruments, International Conference on New Interfaces for Musical Expression, 2002.

[11] G. Lewis, Too Many Notes: Computers, Complexity and Culture in *Voyager*, Leonardo Music Journal, Vol. 10, 2000.

[12] R. Rowe, Interactive Music Systems: Machine Listening and Composing, The MIT Press, 1993.

[13] C. Dodge, Computer Music: Synthesis, Composition, and Performance, Schirmer Books, 1997.

[14] N. Wiener, The Human use of Human Beings: Cybernetics and Society, Da Capo Press, 1950.

[15] K. Hayles, How We Became Posthuman: Virtual Bodies in Cybernetics, Literature, and Informatics, University of Chicago Press, 1999.

[16] M. Wanderley, Performer-Instrument Interaction: Applications to Gestural Control of Sound Synthesis, Dissertation, 2001.

[17] M. Puckette, T. Apel, Real-time Audio Analysis Tools for Pd and MSP, Proceedings of the International Computer Music Conference, International Computer Music Association, 1998.

[18] C. Lippe and M. Puckett, Getting the Acoustic Parameters from a Live Performance, 3rd International Conference for Music Perception and Cognition, 1994.

[19] Z. Settle, The Use of Real-Time Interactive Music Systems in Music Composition and Performance, Dissertation, 2001.

[20] T. Machover, Hyperinstruments - A Progress Report 1987 – 1991, MIT, 1992.

[21] T. Jehan, T. Machover, M. Fabio, Sparkler: An Audio-Driven Interactive Live Computer Performance for Symphony Orchestra, Proceedings of the International Computer Music Conference, 2002.

[22] T. Jehan, B. Schoner, An Audio-Driven Perceptually Meaningful Timbre Synthesizer, Proceedings of the International Computer Music Conference, 2001.

[23] T. Winkler, Composing Interactive Music: Techniques and Ideas Using Max, The MIT Press, 1998.