Published in *Proceedings of the 1991 International Computer Music Conference*, Montreal

Interactive Signal Processing for Acoustic Instruments

Todd Winkler
CCRMA
Stanford University

Abstract

As high-quality signal processors become more sophisticated and programmable, composers are increasingly interested in using them to transform the sounds of acoustic instruments during a live performance. This paper describes a real-time interactive program, created with MAX, that allows a performer to generate and control sound from a MIDI-controllable signal processor according to musical cues and performance gestures. The program analyzes incoming performance data and responds by sending continuous controller messages to various parameters, giving the performer immediate feedback and the ability to alter the processed sound.

MIDI signal processors have proliferated in the last few years, offering sophisticated algorithms that transform a sound in real-time. When used as integral parts of music compositions, they provide a common sound world for synthesized sounds and acoustic instruments to blend, while expanding the coloristic and expressive range of the instruments. Although these "effects boxes" use limited types of processes, their portability, price and programmability makes them attractive tools for use in concert performance. Increased flexibility is gained by using computers to send continuous controller messages, which allow for subtle, dynamic control over all the aspects of the processed sound. As the control of these processes becomes more flexible, it can also become more musical, by responding to a skilled musician, who controls how the parameters change over time.

For this purpose, I created FollowPlay, a program written in MAX (Puckette, 1988) specifically designed for interactive composition. It allows musicians to generate the sound source for processing while sending performance cues to a computer, using performance gestures to specify exactly how the processing will take place, thereby controlling as many aspects of modified sound as the composer desires. The program is specifically designed to respond to acoustic instruments played via a microphone to a pitch-to-MIDI converter (which detects pitch and dynamics over time), but will respond equally as well to MIDI instruments.

One aspect of the program is a collection of MAX patches, each designed to "interpret" incoming performance information. By looking at pitch, velocity, duration and time, the program is able to follow, analyze, store and compare various aspects of a performance. These are general tools for interactive composition (not just for interactive signal processing) and include modules that look at how the music is being played, detecting tempo, articulation, crescendo and diminuendo. Histograms store a specified number of events, capturing information about articulation, dynamics and phrasing, which allows the computer to look at whole sections of a piece. This enables the program to respond immediately to performance gestures, while also making decisions based on how a longer passages were played (Rowe, 1990). The signal processing is responsive to a particular performer or performance, empowering the musician with an extension of their sound sensitive to their musical nuance.

Analytical methods to detect tempo and dynamics are quickly becoming the tools of the trade for interactive composition, and many people have already shown numerous useful algorithms that play music based on a live performance. While most of the algorithms for generating complex music are not useful signal processing, many of the techniques are. For example, sequences of continuous controller information can be triggered by the performer, creating complex timbral changes. The program uses the score-following features of the EXPLODE object (Puckette,

1990) to capture gestures and trigger event. Signal processing information can be sent out immediately as incoming pitches are matched to a prerecorded score. The ubiquitous echo, for instance, can be used to create complex rhythmic canons, as the delay time changes with each successive note. Another technique, which uses two channels of pitch shift to create three-part counterpoint, was demonstrated in my composition "Three Oboes." By using the computer to change the intervals of transposition in time with specific notes, two melodies in contrary motion were heard along with the original solo line.

Other modules make decisions based on how and what is being played, without prior knowledge of the score, and are therefore usable for improvisation as well as a written score. But the mapping of musical gestures onto the various parameters of signal processing must be carefully planned. Just as the subject of a fugue must be thought out for its potential for future exploration and expansion, here too, the composer is challenged to find musical gestures that serve the dual purpose of creating melodic interest while generating a function applicable to signal processing. For example, a module called "Capture Gesture," uses EXPLODE to record the dynamics and rhythm of a 6-second violin phrase. Using velocity as break points, the MAX "line" object creates a function that is sent as continuous controller data to change reverberation time. By lengthening the overall time of the function, the apparent size of the hall changes continuously over the next two minutes of the piece, with the same proportions as the original phrase, thus enabling the performer to "phrase" the apparent size of the hall.

Several techniques can be combined at once using multiple signal processors either in parallel or chained together, the output of one device being the input to the next. This allows for a type of "signal processing orchestration," to be created, using a wide variety of ostinatos, arpeggios, trills, melodic lines, canons, and harmonies, all triggered by a single input.

Problems

Any MIDI-controllable signal processor can be used by the program if the parameters are able to receive continuous controller information. Unfortunately, there is no standardization among devices for how to respond to controller data (numbers 0 to 127), what parameters are controlled by what controller numbers (often this is user definable) or how many parameters can be changed at once. For example, the Yamaha DMP11 is wonderfully engineered for interactive signal processing: all parameters from flanging to EQ can be changed at any time with continuous controller data. By contrast, the Yamaha SPX1000 has much better sound quality, but it is limited to controlling only two parameters at a time. Other problems arise in that these devices receive controller data from 0 to 127, but many of the parameters are scaled from 0 to 16 (pan), -12 to +12 (pitch change) or in irregular increments (reverb time), making it necessary to create a library of translators for each device, which takes data from MAX and scales it to an appropriate number before sending it to the processor.

The problems with pitch detection of acoustic instruments are too numerous to go in to here. MAX offers the opportunity to build error detectors, and note filters can avoid such common problems as double triggers. Pitch-to-MIDI converters vary from adequate to inaccurate. Thankfully, new hardware and pitch detection algorithms will solve many of these problems, at least for some instruments.

Future

One major component that is missing from this program is the analysis of timbre. Several people have introduced real-time Fast Fourier Transform (FFT) analyses which can be used to capture the subtle nuances of timbral changes always present in acoustic music. This opens up a new world of interactive possibilities. Specific harmonics can be followed and mapped to filters, for example, or the strength of upper partials can be used to influence the speed of modulation. Spectral analysis can also be used to control sound synthesis via system exclusive information.

Several recent projects show promising developments in signal processing. New hardware, such as the IRCAM Musical Workstation (IMW) and several devices based around the Motorola DSP56000 allow unprecedented real-time control over sound processing. Many complex processing algorithms that previously took hours to crunch on a mainframe computer are already up and running in real-time on these systems. The increased flexibility in speed and programming enables composers to create unique processes specifically tailored to a musical need. As this technology becomes more available, composers will be able to create processes much more closely aligned with their musical concepts. At that point the standard choices offered these days of flanging, chorus, pitch-shift and reverb will seem like old hat. But this generation of processors still offers a wide range of musical possibilities, most notably the ability to process a sound using multiple effects and to use interactive signal processing to create essential materials of a composition.

References

Puckette, Miller, " The Patcher," Proceedings of the 1988 International Computer Music Conference.

Puckette, Miller, " EXPLODE: A User Interface for Sequencing and Score Following," Proceedings of the 1990 International Computer Music Conference.

Rowe, Robert , "Feature Classification and Related Response in a Real-time Interactive Music System," Proceedings of the 1990 International Computer Music Conference.