

RECENT ADVANCES IN THE COUPLING OF THE LANGUAGE MAX WITH THE MATHEWS/BOIE RADIO DRUM

W. Andrew Schloss
School of Music, University of Victoria
P.O. Box 1700, Victoria, B.C.
CANADA V8W 2Y2
(604) 721-7902
was@uvunix.bitnet or [uw-beaver, ubc-vision]!uvictr!was

ABSTRACT: This demonstration provides new examples of musical uses of the Radio Drum, an interface that is sensitive to location in three dimensions, while maintaining a high degree of temporal accuracy. Also, the language MAX allows one to make "virtual configurations" of the drum, so that a particular event or combination of events can have virtually any musical result. Thus, the language combined with the controller are ideal partners in realtime control of arbitrary synthesizers for a variety of performance situations.

INTRODUCTION

The Mathews/Boie Radio Drum, originally from AT&T Bell Laboratories, can offer performers a new level of control over synthesizer parameters [Mathews and Schloss, 1989]. Miller Puckette's MAX program, written at IRCAM, offers great flexibility and power when dealing with any sort of MIDI data, particularly in complex performance situations [Puckette, 1990]. Combined, these tools offer the performer or composer a musical interface of unparalleled control and flexibility. This convergence is a fortunate coincidence; the two projects grew independently. In this paper we will review the basic functionality of both components of the system, and then discuss some of the ways they are used in performance situations to maximum effectiveness.

In the scope of a musical process, at least three levels of control are possible; they could be called microscopic (timbral level), normal ("note" or instrument level), and macroscopic (process level). We will discuss these levels below as they relate to performance.

THE TOOLS

The Radio Drum itself could be called a "gesture sensor" that keeps constant track of the 3-dimensional spatial location of mallets in the following way: A small amount of wire is wrapped around the end of each of two sticks and acts as a conducting surface electrically driven by a radio frequency voltage source. The drum surface beneath the sticks receives the signal, and the x and y positions of the sticks are derived by determining the first moment of capacitance between the sticks and the surface below. The z position (height) is given by the reciprocal of the first moment. The accuracy in x and y is about .5 cm, over a surface of about 26 by 36 cm, for a total of about 4,000 possible values on the surface alone. The greatest accuracy in z is in the region from about 5 cm above the surface and closer. The response time is between 5 and 10 msec, but this latency can be lowered to below 5 msec. (For technical details see [Boie, et al 1989]).

Once we have the analog signal that represents a continuous determination of x , y , and z coordinates of the ends of the mallets, it can be digitized and analyzed by a dedicated processing unit which talks to the host computer over an RS-232 serial connection. Otherwise the analog signal can be sent straight to a digitizing card in the host computer such as Data Translations or Digidesign. Both methods are currently in use, and both allow a "continuous mode" in which position is continuously output, or a "whack mode" in which only detections of "strokes" are reported, as with a conventional drum. Using a card on the host computer is more general, as it is easier to program the DSP card in the computer than it is to write PROM code for the dedicated processor. However, it turns out that it is sufficient to program the PROM code at this low level only once for all performance possibilities. This is because as long as raw

position data are available at all times to the host computer, they can be used to make the most flexible high-level decisions that one could ever want. For a discussion of a future MIDI version of the drum, see the final paragraph.

The language MAX, a realtime, object-oriented graphical language, is what gives the drum its power. An external object (a code resource) is written as an addition to the standard collection of MAX objects to convert serial data from the drum into MIDI; this simple conversion puts all the power of MAX at our disposal. It is worth mentioning here that the ability to add external objects to MAX is a major step; the development of special objects is proceeding quickly in several locations. MAX is being used at CNMAT (UC Berkeley) and at IRCAM for realtime digital signal processing, as well as MIDI processing. A manual for writing these objects has been written by David Zicarelli [Zicarelli, 1990].

PERFORMANCE ISSUES

Using MAX to interpret the x , y , and z coordinates of the sticks' position, we can reconfigure or remap the surface of the drum in arbitrary ways. For example, the first thing we want to do is make "palettes," which are basically different ways of assigning different regions on the surface of the drum to different results. A convenient way to do this is to use table look-up, with the x and y divisions corresponding to "staircase" tables. For example, if we want to divide the drum into $3 \times 4 = 12$ rectangular regions, we make two tables (one for x , one for y), one with a three-value staircase, the other with four. If the values in the table are chosen correctly, the sum of the two values will give us the values 0 through 11, corresponding to a 12-valued grid on the surface.

Unfortunately, one quickly runs out of "real estate" on the drum surface—we need to be able to toggle between different configurations of the drum. Using MAX and a velocity-sensitive set of MIDI footswitches (like a MIDI organist's foot manual), a series of foot switches analogous to the "shift" and "control" keys on an alphanumeric keyboard are implemented. One can also make "locking" keys that are "sticky" only for high velocity values; otherwise they are only in effect only while being held down. Other multi-valued foot switches have been created that are configured with MAX, and greatly increase the performance bandwidth. Although such complex control could turn into a performer's nightmare, with a natural implementation and sufficient practice it is possible to gain an intuitive feeling for this kind of control.

A novel example of velocity mapping is seen in interactive aspects in performance. We want to be able to record MIDI coming from another player, manipulate it, and send it out in various ways. There are times when we want to operate on something that happened in the past. Following the example of David Wessel [Wessel, 1990] we can create a recorder that "reaches into the past." But here again, we have a bandwidth problem in that there is only so much one can do with two hands, and the performer wants to be able to continue playing while controlling the recording process with his or her feet. Using velocity to create a multi-valued switch, we can map velocity to "reach into the past," by marking phrase-boundaries automatically and then mapping velocity to the number of phrases in the past desired. Very complex mappings of x , y and z can be created that control how the data from the other performers are processed.

LEVELS OF CONTROL

The first level, the *timbral level*, is "microscopic," and demands continuous control of synthesis parameters in several simultaneous dimensions, something that conventional keyboards are not able to do, even when aftertouch, modulation wheel, etc. are added. With the three-dimensional position data coming out of the drum and the programmability of MAX, we can map position in space to any desired parameter, and develop an intuitive feeling for timbral control.

The second level, the *note level*, is "normal" and the most familiar. Here the space is used to control various aspects of the sound, but there is a one-to-one correspondence between striking the surface and hearing an attack. This mode is still much more flexible than other controllers for two reasons: there is no requirement to actually strike the surface; the drum is actually sending out continuous position data in three dimensions near the surface of the pad. This would allow us for example, to predict *ahead of time* the arrival-time of the stick. In certain situations,

for example when dealing with sounds with a slow attack, this could be used to significant advantage. With a MIDI keyboard, or most other controllers, there is no way to know ahead of time that something is about to happen. Using the spatial characteristics of the drum, we can predict the time of arrival and use this extra information to correct for timing problems. It is also possible to have several different sounds triggered as we pass through virtual "layers" above the surface. Of course, this freedom from dependence on physical impact to detect attacks also means that the performer can avoid any sound of impact whatsoever, which is crucial if the instrument is to be used in any context in which there may be significant portions of music that are pianissimo. It is untenable to try to play an electronic instrument that makes more sound acoustically than is being emitted electronically.

The third level, *control of a musical process*, is "macroscopic" and quite open. It helps here to have a completely abstract paradigm, completely configurable, to map to any aspect of the process that is desired. In another situation, it would make an excellent interface to a multimedia environment like Zicarelli's *Ovaltune* or Bernard Mont-Reynaud's *Improvisation*, for example. Another promising possibility is to use this system as a controller of "global" or acoustic parameters from live instruments in an ensemble, for example EQ and spatial location.

THE FUTURE

The Radio Drum in its current instantiation does not *directly* output MIDI data. We now send raw position data from it to the serial port of the Macintosh II, and then use MAX to turn these data into MIDI output for a chosen synthesis engine. This flexibility will hopefully not be lost in the commercial (MIDI) version of the drum, due to be manufactured in the not-too-distant future. The drum, as it works currently, is useless without a computer to convert its output into MIDI data for use with commercial synthesizers. Since any serious computer music performance situation will likely include a host computer to control MIDI devices, this is not a major problem. Only in more commercial situations would we expect performers to be using MIDI gear without the use of a computer in the system.

Our experience indicates that there is tremendous richness in this system for a variety of musical goals, and we believe that a great deal of music that will be created with it.

REFERENCES

R.A. Boie, L.W. Ruedisueli and E.R. Wagner "Gesture Sensing via Capacitive Moments" Work Project No. 311401-(2099,2399) AT&T Bell Laboratories, 1989.

Mathews, Max and A. Schloss "The Radio Drum as a Synthesizer Controller" presented at the ICMC 1989, Columbus Ohio.

Puckette, Miller "Amplifying Musical Nuance" J. Acoust. Soc. of Am. Suppl. 1, Vol. 87, Spring, 1990.

Wessel, David "Computer-Assisted Interactions Between Two Musicians" J. Acoust. Soc. of Am. Suppl. 1, Vol. 87, Spring, 1990.

Zicarelli, David "Writing External Objects for MAX" (As yet unpublished manual), 1990.