
Ajay Kapur
California Institute of the Arts, Valencia, CA, USA
University of Victoria, Victoria, BC, Canada
Simon Fraser University, Burnaby, BC, Canada
KarmetiK LLC, Reno, NV, USA

Arne Eigenfeldt
Simon Fraser University, Burnaby, BC, Canada

Curtis Bahn
Rensselear Polytechnic Institute, Troy, NY, USA

W. Andrew Schloss
University of Victoria, Victoria, BC, Canada

Collaborative Composition For Musical Robots

Abstract

The goal of this research is to collaborate with a number of different artists to explore the capabilities of robotic musical instruments to cultivate new music. This paper describes the challenges faced in using musical robotics in rehearsals and on the performance stage. It also describes the design of custom software frameworks and tools for the variety of composers and performers interacting with the new instruments. Details of how laboratory experiments and rehearsals moved to the concert hall in a variety of diverse performance scenarios are described. Finally, a paradigm for how to teach musical robotics as a multimedia composition course is discussed.

1 | Introduction

In December 2007, Toyota Motor Corporation announced its new line of robots that included a 5-foot “virtuoso violinist” that has 17 computer-controlled dexterous joints in each of its arms and hands. Quite an amazing demonstration of mechanical music and robotic design was portrayed as the machine performed Elgar’s “Pomp and Circumstance”. However, one must question the direction this corporation is taking this field. Are they truly interested in building new vehicles for artistic expression or is this a ploy to build super-fancy toys with an almost jester-like role at upper class functions and dinner parties?

The goal of many of the predecessors in the academic and artistic circles who build musical robotic systems has been to design new instruments to express new musical ideas, not attainable by audio speakers, or human performers. The *MahaDeviBot* [7], a 12-armed solenoid based robotic drummer is such an instrument. The goal of this work is to document the collaboration of a collection of artists, who have worked together to further explore the capabilities of this new instrument, setting a paradigm for how mechanical systems can be used to make meaningful music and the progress towards implementation of 21st Century musical instruments.

A number of different drumming robots have been designed in the academic and artistic communities. Researchers at Harvard University struggled to create an accurate robotic drum roll [5], while next door researchers at MIT developed Cog to control the number of times a stick can bounce [16]. Gil Weinberg developed Haile to explore human to robot interaction [15]. Mitsuo Kawato continues to develop hydraulic systems for humanoid drumming [1]. Many artists have presented a number of different pieces including Baginskyís ìThelxiapeiaî for modified rototom [2], MacMurtieís life size sculptures [9], Gordon Monohans ìMachine Matrixî [11], and Miles van Dorssenís ìCell Projectî including an 8 octave Xylophone, Bamboo Rattle, gong, high-hat and bells [3]. Eric Singer and Trimpin have also had significant contributions to the evolution of robotic drumming [12, 13]. We are greatly inspired by Eric Singerís RoboSonic Eclectic[1] and Trimpinís work with the Kronos Quartet 4-Cast Unpredictable[2] in which humans play along with musical robots.

This discusses challenges in collaborative composing for musical robotics in section 2. Section 3 describes custom tools built for different software packages for aiding in the composition process and for live performance. Section 4 describes performance scenarios for the robot on stage. Section 5 describes how musical robotics are being used in the class room to help open the minds of the upcoming computer musicians.

2 | Challenges

Our experimentation with robotic systems for musical performance brought many familiar yet new

[1] lemurbots.org/robosonic.pdf (Available February 2008)

[2] May 5th, 2007 at Alexander Kasser Theater

challenges to working with sensors. A set of allen wrenches, screw drivers, plyers, a calliper and a dremel are carted to each performance along with a box set of extra springs, screws, washers, and spare parts. Our first designs had frameworks made of wood. This obviously is too heavy a material, and using aluminum is ideal because of its sturdiness and light weight. However, we learned from our initial prototypes that welding anything would be a mistake. All parts should be completely modular to allow for changes in the future. Thus designing our robots out of 20/20 T-slotted aluminum was a perfect material to accomplish all our goals of sustainability, modularity, mobility and professional appearance.

One of the chief challenges in collaboration with composers and musicians was there is only one *MahaDeviBot*. The chief collaborators all live in geographically diverse locations (New York, Vancouver, Victoria, and Los Angeles). Thus, transportation methods had to be considered, including building a custom suitcase for travel on airplanes, and a collapsing the machine to "car-mode" for transfer between performance venues and rehearsal spaces. The *MahaDeviBot* is also very heavy which adds to the challenge.

Beside logistics, there are also performance challenges. We must tune the machines' instruments for every show, which would not be necessary if we were just triggering "perfect" samples. Also, because of the nature of any mechanical system, there are imperfections in event timings based on varying spring tension, speed and strength of previous strikes. However, this produces more realistic rhythms, as humans also have imperfections when actually "grooving".

Another challenge was that each collaborator uses different software to compose and perform. Thus, modular tools had to be designed, as described in Section 3.

By far the biggest challenge with solenoid-based robots controlled by MIDI commands is not only the intrinsic delay that any solenoid has, but the unavoidable *velocity-dependent* delay. That is to say, quieter notes take longer to play than loud notes. The reason for this is clear: the reception of a command to play a note (in MIDI at least) is the initiation of the process of striking the object. If the solenoid is moving slower, it will sound quieter, but it will also take longer to arrive at the object it is striking. This means that a constant velocity input will result in a well-behaved consistent delay, but more realistic performances with accents, etc. will have serious timing discrepancies, and accents will sound terribly out of rhythm.

On the Disklavier, Yamaha dealt with this problem by assigning a 500 msec delay to all MIDI input, and then internally correcting for the velocity-dependent delay; this works because the worst possible delay (at the lowest velocities) is less than 500 msec. This solution is elegant and effective when playing from a MIDI sequence, where the data that are sent to the piano are invisible. But for a live performer triggering the events, it is wrong not only for the audience, but especially for the performer himself, who needs to hear the instant result of his actions. On the Disklavier, this 500 msec delay can be turned off, but then you are left with the original problem. There is no solution to this problem without departing from MIDI and having some sophisticated sensing that can "predict" how fast the performer's hand is moving. Of course, pianists do this instinctively (anticipating quiet notes so that the hammer arrives at the string at the same time as loud notes), without even thinking about it. Learning from this, it is important to calibrate for velocity of *MahaDeviBot* before each performance.

3 | Tools

Each artist used a different software package to compose for the robot. However, a *MahaDeviBot* Toolkit had to be designed as a set of common tools to be used for composing and performing live. These included calibration, pre-delay functionality, virtual performer, and the ability to be networked.

3.1 | Calibration

Although communicating with the robot is rather straightforward via MIDI note-on messages, in practical terms, fine-tuning the musical details is slightly more problematic. Each drum has a specific velocity range, below which it will not strike, and above which it may double strike. These ranges change each time the robot is reassembled after moving. Therefore, a velocity range test patch was created in ChucK [14] and Max/MSP[3] that can determine these limits quickly and efficiently before each rehearsal or performance. The composition program would directly access this array and choose velocities within the range of each drum.

Similarly, each drum also has a physical limit as to how fast it can re-strike; this limit is also determined through a test patch used to inform the program regarding potential tempo limitations. For example, the frame drums have limits of approximately 108 BPM for three consecutive sixteenths (138 ms inter-onset times) while the tambourine and hand-drum can easily play the same three sixteenths at over 200 BPM (better than 75 ms inter-onset times). The composition programs (ex. Kinetic Engine (section 4.3)), directly accessing these limits, would then at-

[3] http://www.cycling74.com/ (February 2008)

tempt to limit consecutive notes for each drum at contentious tempi.

3.2 | Pre-Delay Functionality

Another issue involved in robotic drumming, is that some composers use samples of drums along with the *MahaDeviBot* drum strikes. In order to achieve the desired effect, a pre delay function had to be implemented. Commercial software such as Ableton Live[4] has pre-delay functionality built into each track, thus this was easy to implement in that framework, Code also needed to be written in ChucK and Max/MSP which sent to the robot MIDI messages milliseconds before the audio signal. It was found that 37 ms was the perfect pre-delay value.

3.3 | *MahaDeviBot* Virtual Performer

As the body of artists composing for *MahaDeviBot* increased, software for emulating its virtual performer had to be designed to allow for the robot to be in many locations at once.

One solution was the creation of a VST instrument plug-in that played samples of the robot. Others included making use of a sampler in programs like Ableton Live. More advanced techniques include a virtual performer ChucK class that can be initiated when the robot was elsewhere, or a similar concept in Max/MSP. The sample recordings were made with a fixed microphone above the robot, and unnormalized, so as to retain the large amplitude differences between the various instruments.

When the *MahaDeviBot* was eventually substituted for the virtual performer, the issue of pre-delay had to be addressed, particularly in the fixed media work (see 4.1). Shifting onset times was accomplished very easily in Logic, since the amount would be consistent for every note of that drum.

Velocity data had to be shifted as well, as the virtual performer could obviously play using the complete MIDI range (127 values), whereas the robot's velocity ranges were much more limited. Since the velocity relationship's between individual notes, as well as between drums, could be set using the virtual performer, and only the amplitude relationship between the live robot and the fixed media required alteration, velocity range manipulation, like that of onset times, was straightforward in Logic and other composition software.

3.4 | Networked Robotics

When the *MahaDeviBot* began to perform in concerts where multiple machines would determine what it had to play, it became cumbersome to move the MIDI cable from machine to machine. Thus, an Open Sound Control (OSC) [17] based server was made for the robot and all client machines would send messages accordingly via Ethernet cables. A central Ethernet router was placed on stage with the *MahaDeviBot* always taking slot one with IP 192.168.0.2. A decision was made not use wireless capabilities of the router, so as not to add more confusion and possible delay time to our stable system. One issue of concern was whether a close wireless network would allow rogue messages from hackers in the audience to hijack the *MahaDeviBot*. The server was designed in ChucK; however the client machines generally use Max/MSP to convert MIDI messages to send OSC.

04 | Performance Scenarios

4.1 | Fixed Media Piece

One work created for *MahaDeviBot* involved live performance by *ESitar* [8] interacting with fixed media triggered by a sequencer (Logic) that combined live processing and diffusion of soundscape recordings with predetermined MIDI data sent to the robot. Somewhat surprisingly, the *MahaDeviBot's* musical results were closer to that of a live performer than sequencer-driven sample playback, even when triggered from a sequencer. For example, extremely complex polyrhythms and interlocking cross-rhythms, unplayable by humans but perceptible via sample playback, sounded muddy and imprecise when performed by the robot; however, more simple rhythms acquired subtle variations that maintained listener interest. *MahaDeviBot* seems to have a personality - albeit one that changes each time it was reassembled - in its imperfections. For a composer interested in inhuman complexity - or even "robotic" perfection - *MahaDeviBot* is not a solution; however, for composers interested in human-like variations in playback - missing from so much sequencer-based music, *MahaDeviBot* is a welcome performer.

4.2 | *MahaDeviBot* for 2 *ESitars* and *EDilruba*

Another approach to robotic composition employs performance data from musicians playing amplified and sensor-extended instruments to inform the specification and general quality of drumming patterns. This experiment was administered using two different sensor-extended Sitar designs (Kapur and Bahn) as well as an extended Dilruba (Bahn), (a North Indian bowed string instrument which is a cross between a Sitar and a 'Cello). One program we called "butterfly," would listen to the sitar performer's thumb sensor, to determine when to start actuating different arms at very fast rates and small velocities (so small that the drums would not even be stuck). The fret being performed would deter-

[4] http://www.ableton.com/ (February 2008)

mine which of the 12 actuators would be activated. Another program "listened" to the performer using a simple energy calculation (RMS) and would strike a drum if a certain threshold was reached. Another program would take sensor data for both human performers to aid in decisions in which sequenced drum patterns should be performed [6].

01 | *MahaDeviBot Trio with Kapur & Bahn at NIME 2007.*

4.3 | *MahaDeviBot* for KinetiK Engine

Kinetic Engine [4] is a rhythm generating program using a multi-agent architecture to create complex ensemble rhythms. Previous versions involved playing percussion samples, and a great deal of effort was made into creating more realistic performance gestures by generating timbral variations through signal processing and sample switching. Not surprising, using *MahaDeviBot* alleviated many of these necessities, as the acoustic nature of the drums, coupled with subtle performance variations of the robot, created the necessary variations.

Even with the drum's limited velocity range - for example, a frame drum typically ranges from 35 - 85 within the MIDI velocity range - a great deal of perceived amplitude variation was possible within this fixed range. Although performing at the lowest end of the velocity range occasionally produced inaudible notes, these can be heard as being closer to human, rather than machine, inaccuracies.

4.4 | *MahaDeviBot* for Radiodrum

Andrew Schloss, composer and performer on the *Radiodrum [10]*, created a new piece for the robot called "*MahaDeviBot* Variations." There were two techniques that we used in "*MahaDeviBot* Variations" that were successful. The first was to trigger samples (live from the Radiodrum) at the same time as the robot. Even given the delay, this still had musical validity, especially if the samples were chosen carefully to complement the sound of the acoustic instrument. The second was that we experimented with processing the live sound as it was played by the robot. That is to say, there is a microphone on the acoustic drum or object, and just as you trigger the robot to play from the *Radiodrum*, you initiate and continuously control several parameters of DSP using the 3-dimensional sensing of the *Radiodrum*. This is a new and fertile area, combining robotic performance with DSP and DAFX in realtime. It is quite dramatic, since the performer's gesture and the robot's response are visible, and so are the subsequent gestures used to control the processing of the sound the robot makes.

4.5 | *MahaDeviBot* with a Live Band

One of the biggest challenges has been to try and get the *MahaDeviBot* to perform with a live band. In an initial attempt, the robot was invited to an International Jazz Festival: several factors, including limited set-up time, failing MIDI communication, and the complexity of coordinating many humans with the robot, proved overwhelming, to destroyed the event. However, with months of practice and more experience on logistical issues of robotic performance, the *MahaDeviBot* successfully became an active member in a live band. One major key to success was building a new instrument for the drummer of the band to control the robot, which determined patterns, builds, tempo changes, dynamics and rhythmic space and density.

02 | KarmetiK Collective playing with *MahaDeviBot November 2007.*

05 | Class Room Robotic Music

At California Institute of the Arts we ask, how can this emerging artform be taught in the classroom? In a classroom full of composers and computer musicians, we begin by a history lecture of various artists and scientists prevalent in the field. Each student chooses one artist to do a detailed report and presentation to the class. This is to help inform the students of various work that has already been done, and to learn how they can use it in their own work. The next phase of the class involves building software tools similar to the one described in section 3. Each student designs calibration software, pre delay functionality, and most importantly a virtual performer so that they can compose in their dorm rooms. At the end of this phase, each student is introduced to OSC and how to convert their messages to our client server model.

The final stage involves composing, rehearsal and performance. Each student takes very different approaches, utilizing their own aesthetics and abilities. Some choose to work with symphonic instruments, others with Indian instruments, others with only drums, other with new interfaces, others with live video software; all works are put together into a one night performance, showcasing their work.

06 | Conclusions

This research showed the evolution of a piece of technology that, through collaboration with many artists, took on life and became a successful tool for 21st Century composition. The authors see this is a huge paradigm shift from the common computer music mentality of one-(wo)man designing software/hardware for themselves and not fully exploring the capabilities through user testing and joint collaboration.

Future work will proceed in various directions. Work on using the *MahaDeviBot* as a means for teaching North Indian music is beginning to be explored at *CalArts*. As the robot gains popularity, more composers are beginning to integrate it into their work, including the venerable Ustad Aashish Khan. More robotic instruments are now in the design phase, trying to complete a full North Indian robotic music ensemble. This will present many more questions including: How do you make robotic instruments interact with themselves?

REFERENCES

[1] C. G. Atkeson, J. Hale, et. al., "Using Humanoid Robots to Study Human Behavior," *IEEE Intelligent Systems: Special Issue on Humanoid Robotics*, vol. 15, pp. 46-56, 2000.

[2] N. A. Baginsky, "The Three Sirens: A Self Learning Robotic Rock Band."

[3] M. V. Dorssen, "The Cell."

[4] A. Eigenfeldt, "The Creation of Evolutionary Rhythms Within a Multi-Agent Networked Drum Ensemble," ICMC, Copenhagen, Denmark, 2007.

[5] A. Z. Hajian, D. S. Sanchez, and R. D. Howe, "Drum Roll: Increasing Bandwidth Through Passive Impedance Modulation," IEEE Robotics and Automation Conference, Albuquerque, USA, 1997.

[6] A. Kapur and E. Singer, "A Retrieval Approach for Human/Robot Musical Performance," ISMIR, Victoria, Canada, 2006.

[7] A. Kapur, Trimpin, E. Singer, A. Suleman, and G. Tzanetakis, "A Comparison of Solenoid-Based Strategies for Robotic Drumming," ICMC, Copenhagen, Denmark, 2007.

[8] A. Kapur, G. Tzanetakis, and P. F. Driessen, "Audio-Based Gesture Extraction on the ESitar Controller," DAFX, Naples, Italy, 2004.

[9] C. MacMurtie, "Amorphic Robot Works."

[10] M. Mathews and A. Schloss, "The RadioDrum as a Synthesizer Controller," ICMC, 1989.

[11] G. Monahan, "Kinetic Sound Environments as a Mutation of the Audio System," Musicworks, Toronto, Canada, 1995.

[12] E. Singer, J. Feddersen, et. al., "LEMUR's Musical Robots," NIME, Hamamatsu, Japan, 2004.

[13] Trimpin, *Portfolio*. Seattle, Washington.

[14] G. Wang & P. R. Cook, "ChucK: A Concurrent, On-the-fly Audio Programming Language," ICMC, Singapore, 2003.

[15] G. Weinberg, et. al., "Haile - A Preceptual Robotic Percussionist," ICMC, Barcelona, Spain, 2005.

[16] M. M. Williamson, "Robot Arm Control Exploring Natural Dynamics," vol. Ph.D. Boston: MIT, 1999.

[17] M. Wright, A. Freed, and A. Momeni, "OpenSound Control: State of the Art 2003," NIME, Montreal, Canada, 2003.