Shared Control in Robotic Drum Performance

Written by: Regis Verdin

2/29/2016

Abstract

A system was created for enabling shared control of musical material between a robotic percussionist and one or more people. The system/composition was created using Max/MSP to work with the GTCMT robot Shimon. Shimon is traditionally a marimba player, but various percussion instruments were substituted here instead.

1 Introduction & Motivation

Shimon is a robotic marimba player in the Georgia Tech Center for Music Technology. The aim of this project was to explore the possibilities of having Shimon perform on other instruments than the marimba. It seemed that Shimon's physical form allowed for many more musical paradigms than were being expressed with the marimba alone, and that perhaps Shimon should be thought of as separate from the marimba altogether.

Shimon's marimba was replaced with a set of 4 percussion instruments: large tom, ride cymbal, music stand, and a flexible sheet of metal. The motivation here was that a human could present Shimon with any combination of percussive instruments, and Shimon could learn to play them. Using two microphones, Shimon is given auditory feedback on the characteristics of the instrument it is striking, and adjusts its playing accordingly. Shimon can also respond to timbral changes enacted by the human performer on the percussion instruments.

The patch in Max/MSP (which is functioning here as part of Shimon's "brain"), serves as a modular composition. The aesthetics were driven by a wish to demonstrate Shimon's listening capabilities, creating a slow build up that gradually introduces more complexity in the interaction between Shimon and the human performer.

In this paper we will look into the technical and aesthetic context for this kind of musical creation, the details of the tested performance (drum layout, etc.), and the construction of the Max patch, as well as considering future possibilities for this kind of work.

2 Background

The notion of embodied cognition was a starting point for this project, especially in its relation to robotic musicianship. The basic idea is that an agent's cognitive abilities are largely shaped by its environment and physical capabilities. In our case, Shimon is the agent. It occurred to me that Shimon was so closely tied to its marimbas that it was almost taken for granted that Shimon should always play marimbas. In that case, the goal here was to first re-contextualize the marimbas as part of a mutable environment rather than being a fixed part of Shimon's physical structure. Having done that, Shimon became extremely versatile as a percussionist, capable of not only striking with 8 independently controlled solenoid-driven mallets at different velocities, but also being able to re-position each of 4 arms along one axis of whatever percussion objects it would be presented with.

This ability to move Shimon's arms along the surface of a drum suggested that Shimon could be trained to recognize timbre. The composition

did not end up using Shimon's arm movements, but instead relied upon the human performer to manipulate the percussion instruments to change the timbre. This came within the context of exploring how humans can interact with robotic musicians, and the various interactions and communication dynamics that can be played out.

The musical possibilities of this context suggested the repertory of solo (unpitched) percussion compositions. I took inspiration especially from those by Iancu Dumitrescu and Iannis Xenakis, in the space for deep timbral listening that Dumitrescu often brings, and in the stochastic qualities of Xenakis's music.

3 Methodology

The performance system and composition were both implemented in Max/MSP. The patch begins by sending UDP messages to move Shimon's arms to predetermined locations. These locations were chosen based on the layout of the percussion instruments, so that the mallets on each arm could strike their assigned drum. As mentioned before, the drums (from left to right) were as follows: large tom, ride cymbal, music stand, and a suspended flexible sheet of metal. Microphones are also placed above the tom and sheet of metal.

The main function in the patch is a rhythmic scheduler, which allows a user to send a series of rhythmic values to the system in a simple form. For example, "4 4 3" could play a note for 4 quarter notes, the next for 4 quarter notes, and the last for 3 quarter notes. The rhythmic unit can also easily be changed to 8th notes or any other subdivision. When the scheduling subpatcher receives a new series of values, it appends them to the currently scheduled values. A series of rhythmic fragments was composed and added to a list. The program randomly chooses from a subset of these depending on which section is being played.

Two separate listener functions were also implemented, one for the tom and one for the sheet metal. The concept was to limit the machine listening to these two drums, to simplify testing. Spectral centroid analysis takes place in real-time on these two drums and triggers changes in the way Shimon plays. For example, if the human performer presses on the head of the tom, this increases the spectral centroid and triggers the start of the section using the cymbal and the music stand. When the performer bends the sheet of metal, the spectral centroid also changes, and the tempo is mapped to this feature later in the composition.

4 Results & Analysis

The system was used in several test performances and presentations, in a predictable fashion that was easy for the author to play along with. Shimon was demonstrated to respond to timbral changes applied by the human performer, for enacting sectional changes in the music and modulating the tempo in real-time. The composition was mostly focused on demonstrating Shimon's ability to play non-marimba percussion, and Shimon's ability to respond to timbral changes in the tom and sheet metal. The compositional depth was somewhat limited (simply using random numbers to pick rhythmic fragments) to focus on these, but could be expanded. It was successful in demonstrating Shimon's capabilities at playing a wide variety of percussion instruments, and was (subjectively) enjoyable to share control with Shimon in the performance.

5 Future Work & Recommendations

Having created the basic system for this kind of performance, future work can focus more on creating an interesting composition that demonstrates the wide possibilities of having Shimon listening for timbral changes on its drums. This interactions could be even more obvious. The system would work well as an installation, where multiple users could casually approach Shimon and learn to play along with it. A well chosen set of objects to place on the drum head would be appropriate here; performers could use these to modify the percussion timbres in a great number of ways.

Contact mics would ideally be placed on each

of the percussion instruments. Using an open air microphone was limiting as soon as other drums started playing, because their sounds would also be analyzed. Shimon's arms should also move to explore the possibilities of the drums before it. If a particular area of interest is detected using the spectral analysis, Shimon's head could also move in that direction to show its engagement with a human performer. The timbre was also limited by the beaters Shimon was using, so these should be changed to better suit each drum. A long term goal could be to augment Shimon's understanding of the environment with vision, in addition to audio, so it could also see the form of various percussion objects presented to it and react accordingly.

6 Conclusion

A musical performance system, involving shared control with a robot musician, was created in Max/MSP. In several test performances a human user was able to interact with the robot by manipulating the timbre of the percussion instruments the robot was playing. The robot responded to these sonic changes in a way that encouraged the human performer to explore the interaction dynamic.

7 Additional Resources

GitHub Code: https://github.com/regisverdin/ShimonShays.git

Demo Video: https://youtu.be/L4Y1ju0at4I