The Robo-Cajon: An Example of Live Performance with Musical Robotics

Austin Franklin
Mälardalen University
austin.franklin@mdu.se

ABSTRACT

The Robo-Cajon is a robotic musical performer capable of real-time improvisation and live performance with a human performer. It is a wooden cajon, or box-shaped percussion instrument played with the hands, mounted with two push/pull solenoids that receives input from a separate cajon mounted with piezo contact microphones performed by a human. Rhythmic data from the human performer is first classified using a Multi-Layer Perceptron (MLP) and a Hidden Markov Model (HMM) is used for prediction and generating rhythmic patterns based on this prior classification. The Robo-Cajon demonstrates a novel and lightweight approach to human-computer interaction and real-time applications for various machine learning algorithms. The Robo-Cajon was realized using Max and Arduino.

1. INTRODUCTION

Musical robotics is a field which explores the intersection and relationship between creativity and robotics. The potential for machines to express creativity is important to the field of robotics and AI, but it is especially relevant for machines that replicate human activities that are generally described as creative, such as music making [1]. Humans are especially good at predicting, synchronizing, and harmonizing amongst rapidly changing environments in real-time, while the challenges faced by current robotics architecture to detect, interpret, and respond quickly make these tasks difficult. Music making is a well-structured setting for determining a robots ability to perform these tasks, and the implications of high performance are applicable throughout the field of robotics [2].

The author presents *The Robo-Cajon*, a robotic musical performer capable of real-time improvisation and live performance with a human performer. [1] Capable of detecting and interpreting rhythms from a human performer on a separate cajon using a Multi-Layer Perceptron as described by J. Bullock and A. Momeni [3], *The Robo-Cajon* generates new rhythmic sequences in real-time using a Hidden Markov Model at a fixed tempo predetermined by the human performer in the Global Transport in Max. A Bluetooth enabled foot pedal is introduced that allows the human performer additional agency over robotic performer and can be used to retrain the MLP and HMM at any time.

[1] Demo video available at https://www.youtube.com/watch?v=m-J5rTQk3Uct=115s

2. RELATED WORK

Several basic types of "beater" or "mallet" based robotic instruments were explored and used as the mechanical basis for *The Robo-Cajon*, such as **Dartron** [4] and the **MahaDeviBot** [5] seen in Figure 1. As a proof of concept, the *The Robo-Cajon* was designed without the use of mallet based rotational motion in favor of the linear motion provided by the solenoid plunger itself.

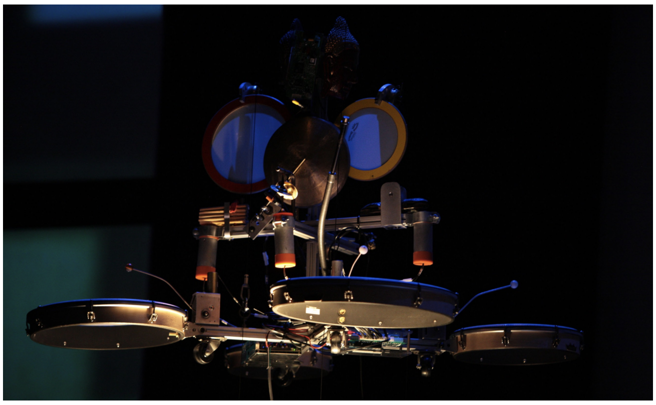

Figure 1. *MahaDeviBot*

In 1999 Trimpin created an installation entitled **Conloninpurple**, which is "a 7-octave instrument with wooden bars and metal resonators" [6]. Each bar is excited using a push/pull solenoid which strikes the bar when a specific MIDI message is received (Figure 2). It is this simple mechanical framework that allows for easy experimentation and portability with *The Robo-Cajon*. Routing MIDI messages to trigger a hardware controller is also inherently slower than feeding analog signals to solenoids via Arduino.

A number of systems included in the category of interactive music have been developed to enable human-machine improvisation. Examples utilizing software to interpret the gestures of human performers and produce synthesized sounds include **Cypher** [7], **Voyager** [8], and **The Continuator** [9]. Other systems interpret the gestures of human performers and produce sound using physical electro-mechanical instruments. These include **Haile** and **Shimon** [10] seen in Figure 3, both of which exhibit musical expression through the embodiment of musical actions [11]. *The Robo-Cajon* implements a software and hardware component and is positioned somewhere in between these two categories.

Figure 2. *Conloninpurple*

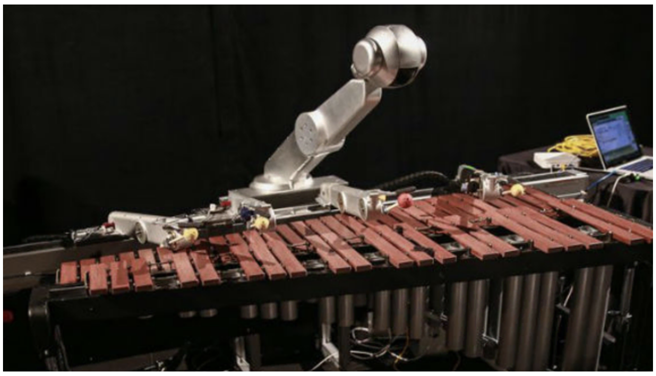

Figure 3. *Shimon*

3. INTERACTIVE DESIGN

The design for *The Robo-Cajon* is intentionally minimal (Figure 4) as a prototype design that is cost effective, using only two cajons and a Bluetooth enabled foot pedal. [2] The human performer first sets the tempo in Max using the Global Transport, performs on their cajon in that tempo, and uses the foot pedal to initialize training on the second cajon of all rhythmic material performed since the last foot pedal trigger. The addition of the footpedal originally made developing and testing *The Robo-Cajon* easier, and was decidedly kept due to its ease of use. The robotic performer begins improvising on this material instantly and can be retrained at any time by the human performer. This gives the human performer some degree of control over the robotic performer, allowing them to shape the overall phrasing and musical structure.

Both cajons are identical in size with the robotic performer being custom built using the dimensions from the human performer (12x12x17.25 inches). [3] The solenoids on the robotic performer are secured with a brace that is set approximately 1/2 inch from the front face. One solenoid is positioned at the center of the brace to produce a lower sounding pitch mimicking the bass sound of the human performer while the other is positioned near the edge to mimic the snare sound (Figure 4). This brace holds the

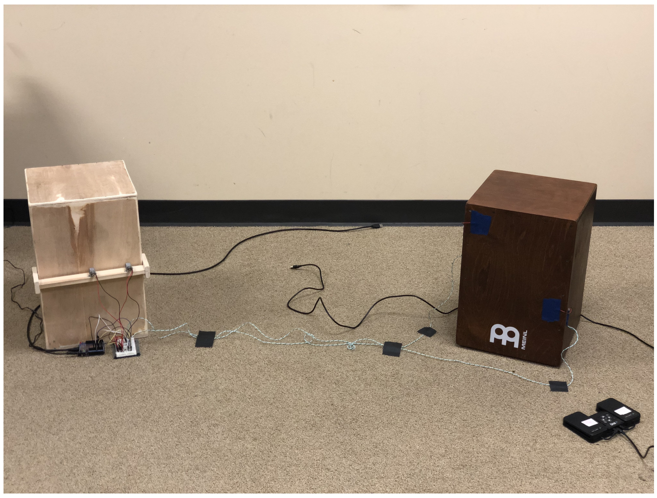

Figure 4. *The Robo-Cajon*

solenoids in this position and at this distance to insure that each strike from the plunger results in an optimal and consistent sound.

4. TECHNICAL DETAILS

The software component for *The Robo-Cajon* utilizes Arduino to read from analog pins and write to digital pins, and Max to interpret this data in a way that can be classified and used to generate new sequences. Each of these components are described in more detail below.

4.1 Arduino

The Robo-Cajon uses an Adafruit METRO 328 Arduino-Compatible board [4] to act as a bridge between both cajons and Max (Figure 5). It takes the analog pin values from two piezo contact microphones and sends them to Max using the 'serial' object. Concurrently, Arduino receives values back from Max and uses them to determine which digital pin to set as high or low. The state of these digital pins control the state of two 12V push/pull solenoids, where high equals extended and low equals retracted. The most efficient way to control the state of two digital pins simultaneously in Arduino is with a switch case:

```
switch(pinValue) {
     case 0;
          digitalWrite(solenoid1, LOW);
          break;
     case 1:
          digitalWrite(solenoid1, HIGH);
          break;
     case 2:
          digitalWrite(solenoid2, LOW);
          break;
     case 3:
          digitalWrite(solenoid2, HIGH);
          break;
}
```

[2] https://www.sweetwater.com/store/detail/Duo200–airturn-duo-200-transmitter-with-2-pedals?utm$_{s}ource = googleutm_{m}edium = organicpla$

[3] https://www.amazon.com/Meinl-Percussion-Internal-Europe-Baltic-BC1NTWR/dp/B00LJP78Q4/ref=psdc$_1$1972491$_t$2$_B$0799$BM6KN$

[4] https://www.adafruit.com/product/2488

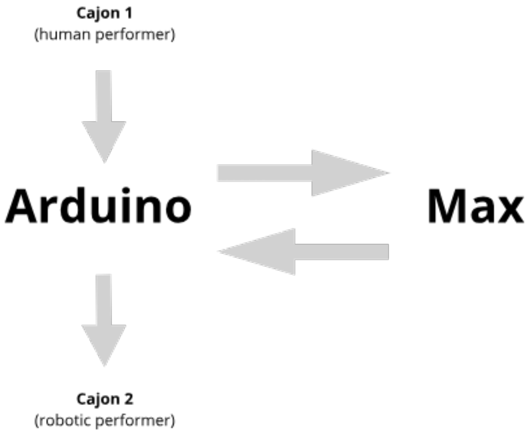

Figure 5. Diagram of *The Robo-Cajon*

4.2 In Max

The piezo values are sent to Max using the 'serial' object and are used to trigger a bang when a threshold has been met. The time (in ms) between bangs is then taken and scaled from an existing tempo first set in the Global Transport to a normalized range between 0-1 (Figure 6). The specific piezo contact microphone is also recorded to indicate whether a slap, tone, or both were played. This list of numbers containing the piezo contact microphone and normalized length of the note are then classified using a Multi-Layer Perceptron Neural Network developed by Ben Smith as part of the ml.star package [12].

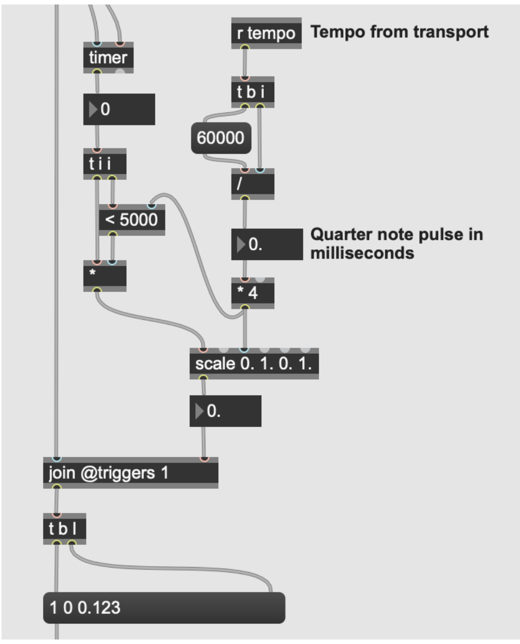

Figure 6. Scaling tempo from the Global Transport

Training data for the Multi-Layer Perceptron only includes note lengths shorter than a whole note and longer than a 1/16th note. The difference between the normalized note lengths shorter than this is so small that it often leads to incorrect classification if not performing perfectly in time (0.041667 of a bar for 1/16th note triplets and 0.03125 for 1/32nd notes) [13]. Two additional training sets are also used to classify and discard note lengths that are larger and smaller than whole notes and 1/16th notes. Once classifications are made, the floating point values are rounded, sorted by slap (1), tone (2), or both (3), and the correct classifications are selected using the 'select' object. All other classifications are discarded and return an error message to the console (Figure 7).

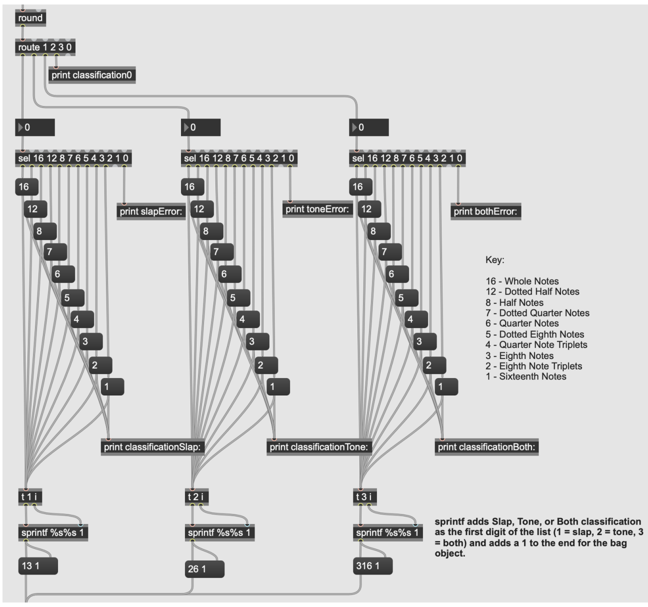

Figure 7. Rounding and Selecting Classifications in Max

All classifications are then stored in the 'bag' object. This object stores a collection of values and dumps them when triggered. Using the Bluetooth foot pedal, the human performer can trigger and dump the contents of the bag into a Hidden Markov Model from the ml.star package [12]. Once dumped, the Hidden Markov Model is immediately trained and begins generating new rhythmic sequences based on the data. These new sequences are then scaled back from normalized values to their corresponding lengths (in ms), originally determined by the Global Transport, before being sent back to 'serial' object to control the solenoids.

5. FUTURE WORK

While the interaction and improvisation between the robotic and human performer is straight-forward, it does require prior knowledge of how the 'bag' and foot pedal collectively stores, dumps, and trains models. Future work could include devising methods for training and interpreting data automatically, such as syncing dumping and training to a downbeat that occurs every N number of measures or beats. Future work could also include BPM estimation algorithms that could set the Global Transport tempo automatically, which would allow musically and non-musically trained users to simply play and allow the robotic performer to self-adjust to a wide variety of tempos and rhythms. The touch or fingers pressure (or amplitude) of the human performer could also be measured and used to control various parameters making the Robo-Cajon more expressive. The addition of instruments (hardware or software) could provide other interesting areas for experimentation, and other parameters could be classified and used to control audio synthesis or non-linear systems. For example, pitched instruments such as a disklavier could be used and MIDI data related to pitch and velocity could be used to train an MLP.

6. CONCLUSIONS

The Robo-Cajon is a robotic musical ‘performer’ capable of real-time improvisation and live performance with a human performer. It uses Arduino to receive piezo contact microphone analog values and control two push/pull solenoids and Max to detect and interpret rhythms and generate new rhythmic sequences using a Multi-Layer Perceptron and Hidden Markov Model. *The Robo-Cajon* demonstrates a novel and lightweight approach to human-computer interaction and real-time applications for various machine learning algorithms.

7. REFERENCES

[1] S. Kemper, “Locating Creativity in Differing Approaches to Musical Robotics,” *Frontiers in Robotics and AI*, vol. 8, p. 52, 2021. [Online]. Available: https://www.frontiersin.org/article/10.3389/frobt.2021.647028

[2] C. Crick, M. Munz, and B. Scassellati, “Synchronization in social tasks: Robotic drumming,” in *ROMAN 2006-The 15th IEEE international symposium on robot and human interactive communication*. IEEE, 2006, pp. 97–102.

[3] J. Bullock and A. Momeni, “Ml.Lib: Robust, Cross-Platform, Open-Source Machine Learning for Max and Pure Data,” in *Proceedings of the International Conference on New Interfaces for Musical Expression*, ser. NIME 2015. Baton Rouge, Louisiana, USA: The School of Music and the Center for Computation and Technology (CCT), Louisiana State University, 2015, p. 265–270.

[4] A. Kapur and M. Darling, “A Pedagogical Paradigm for Musical Robotics.” in *NIME*, 2010, pp. 162–165.

[5] A. Kapur, A. Eigenfeldt, C. Bahn, and W. Schloss, “Collaborative Composition For Musical Robots,” *Journal of Science and Technology of the Arts*, vol. 1, 05 2009.

[6] A. Kapur, “A History of robotic Musical Instruments,” in *ICMC*, 2005.

[7] R. Rowe, *Interactive music systems: machine listening and composing*. MIT press, 1992.

[8] G. E. Lewis, “Too many notes: Computers, complexity and culture in voyager,” *Leonardo Music Journal*, vol. 10, pp. 33–39, 2000.

[9] F. Pachet, “The continuator: Musical interaction with style,” *Journal of New Music Research*, vol. 32, no. 3, pp. 333–341, 2003.

[10] G. Weinberg and S. Driscoll, “The interactive robotic percussionist-new developments in form, mechanics, perception and interaction design,” in *2007 2nd ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 2007, pp. 97–104.

[11] S. Kemper and S. Barton, “Mechatronic Expression: Reconsidering Expressivity in Music for Robotic Instruments,” 06 2018.

[12] B. D. Smith and G. E. Garnett, “Unsupervised Play: Machine Learning Toolkit for Max,” in *NIME*, 2012.

[13] L. Noriega, “Multilayer perceptron tutorial,” *School of Computing. Staffordshire University*, 2005.