MUSIC INFORMATION ROBOTICS: COPING STRATEGIES FOR MUSICALLY CHALLENGED ROBOTS

Steven Ness, Shawn Trail
University of Victoria
sness@sness.net
shawntrail@gmail.com

Peter Driessen
University of Victoria
peter@ece.uvic.ca

Andrew Schloss, George Tzanetakis
University of Victoria
aschloss@uvic.ca
gtzan@cs.uvic.ca

ABSTRACT

In the past few years there has been a growing interest in music robotics. Robotic instruments that generate sound acoustically using actuators have been increasingly developed and used in performances and compositions over the past 10 years. Although such devices can be very sophisticated mechanically, in most cases they are passive devices that directly respond to control messages from a computer. In the few cases where more sophisticated control and feedback is employed it is in the form of simple mappings with little musical understanding. Several techniques for extracting musical information have been proposed in the field of music information retrieval. In most cases the focus has been the batch processing of large audio collections rather than real time performance understanding. In this paper we describe how such techniques can be adapted to deal with some of the practical problems we have experienced in our own work with music robotics. Of particular importance is the idea of self-awareness or proprioception in which the robot(s) adapt their behavior based on understanding the connection between their actions and sound generation through listening. More specifically we describe techniques for solving the following problems: 1) controller mapping 2) velocity calibration, and 3) gesture recognition.

1. INTRODUCTION

There is a long history of mechanical devices that generate acoustic sounds without direct human interaction starting from mechanical birds in antiquity to sophisticated player pianos in the early 19th century that could perform arbitrary scores written in piano roll notation. Using computers to control such devices has opened up new possibilities in terms of flexibility and control while retaining the richness of the acoustic sound associated with actual musical instruments. The terms music robots or music robotic instruments have been used to describe such devices [6].

We believe these new robotic instruments have a legitimate place with potential to become part of an embedded conventional musical practice, not just a research curiosity. While musical-robotics might seem niche and esoteric at this point [2], historic innovations such as monophonic to polyphonic music, electrical amplification of the guitar, or computers in the recording studio all brought skepticism, but eventually became mainstay practices.

Although such music robots have been used in performance of both composed and improvised music as well as with or without human performers sharing the stage, they are essentially passive output devices that receive control messages and in response actuate sound producing mechanisms. Their control is typically handled by software written specifically for each piece by the composer/performer.

Musicians through training acquire a body of musical concepts commonly known as musicianship. Machine musicianship [9] refers to the technology of implementing musical process such as segmentation, pattern processing and interactive improvisation in computer programs. The majority of existing work in this area has focused on symbolic digital representations of music, typically MIDI. The growing research body of music information retrieval, especially audio-based, can provide the necessary audio signal processing and machine learning techniques to develop machine musicianship involving audio signals.

The typical architecture of interactive music robots is that the control software receives symbolic messages based on what the other performers (robotic or human) are playing as well as messages from some kind of score for the piece. It then sends control messages to the robot in order to trigger the actuators generating the acoustic sound. In some cases the audio output of the other performers is automatically analyzed to generate control messages. For example audio beat tracking can be used to adapt to the tempo played.

Self listening is a critical part of musicianship as anyone who has struggled to play music on a stage without a

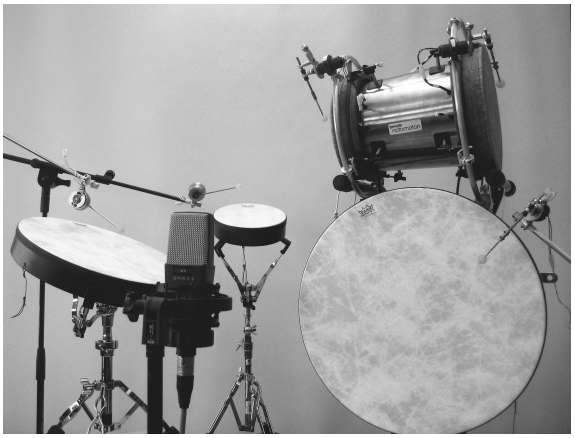


Figure 1. The experimental setup for our robotic based frame drum experiments. In the foreground, three frame drums are shown with solenoids placed to ensure optimal striking of the drum surface. In the background of the picture, the control system is shown.

proper monitor setup has experienced. However this ability is conspicuously absent in existing music robots. One could remove the acoustic drum actuated by a solenoid so that no sound would be produced and the robotic percussionist will continue "blissfully" playing along.

This work has been motivated by practical problems experienced in a variety of performances involving percussive robotic instruments. Figure 1 shows our experimental setup in which solenoid actuators supplied by Karmetik LLC. [1] are used to excite different types of frame drums.

We show how the ability of a robot to "listen" especially to its own acoustic audio output is critical in addressing these problems and describe how we have adapted relevant music information retrieval techniques for this purpose. More specifically, we describe how self-listening can be used to automatically map controls to actuators as well as how it can be used to provide self-adapting velocity response curves. Finally, we show how pitch extraction and dynamic time warping can be used for high-level gesture analysis in both sensor and acoustic domains.

2. RELATED WORK

An early example of an automated, programmable musical instrument ensemble was described by al-Jazari (1136-1206) a Kurdish scholar, inventor, artist, mathematician that lived during the Islamic Golden Age (the Middle Ages in the west). Best known for writing the Book of Knowledge of Ingenious Mechanical Devices in 1206, his automata were described as fountains on a boat featuring four automatic musicians that floated on a lake to entertain guests at royal drinking parties. It had a programmable drum machine with pegs (cams) that bumped into little levers that operated the percussion. The drummer could be made to play different rhythms and different drum patterns if the pegs were moved around, performing more than fifty facial and body actions during each musical selection. This was achieved through the innovative use of hydraulic switching. A modern example of a robotic musical ensemble is guitarist Pat Metheny's Orchestrion which was specifically influenced by the Player Piano [2]. Metheny cites his grandfather's player piano as being the catalyst to his interest in Orchestrions, which is a machine that plays music and is designed to sound like an orchestra or band.

A seminal book in this field is "Machine Musicianship" [9], in which one of the sections describes a comprehensive system for the composition, creation and performance between humans and robots. Rowe describes improvisational and composition systems that combine features of music feature extraction, musical analysis and interactivity to generate engaging experiences for the audience. In our work, the integration of machine musicianship and music robotics has been used to develop a robotic percussionist that can improvise with a human performer playing a sitar enhanced with digital sensors [7].

Another work closely related to ours is the *Shimon* human-robot based Jazz improvisation system [3] that uses a gesture based framework that recognizes that musicianship involves not just the production of notes, but also of the intentional and consequential communication between musicians [4].

Our system also uses these same basic building blocks, but adds the power of machine learning and "proprioception" to the process, enabling the robot itself to perform many of the time consuming mapping and calibration processes that are often performed by hand in performance situations. In this context, a mapping refers to the process of determining which controller output activates which solenoid. In the next section we describe how some practical recurring problems we have experienced with robots in music performance robots have led to the development of signal processing and machine learning techniques informed by music information retrieval ideas.

3. MOTIVATION

Our team has extensive experience designing music robotic instruments, implementing control and mapping strategies, and using them in live and interactive performances with human musicians, frequently in an improvisatory context. In addition two of the co-authors are professional musicians who have regularly performed with robotic instruments. One

[1] http://karmetik.com

[2] http://www.patmetheny.com/orchestrioninfo/

of the most important precursors to any musical performance is the sound check/rehearsal that takes place before a concert in a particular venue. During this time the musicians setup their instruments, adjust the sound levels of each instrument and negotiate information specific to the performance such as positioning, sequencing and cues. A similar activity takes place in performance involving robotic acoustic instruments in which the robots are set up, their acoustic output is calibrated and adjusted to the particular venue and mappings between controls and gestures are established. This process is frequently tedious and typically requires extensive manual intervention. To some extent this paper can be viewed as an attempt to utilize techniques and ideas from MIR to simplify and automate this process. This is in contrast to previous work in robotic musicianship that mostly deals with the actual performance. More specifically we deal with three problems: automatic mapping, velocity calibration, and melodic and kinetic gesture recognition.

The experimental setup that we have used consists of a modular robotic design in which multiple solenoid-based actuators can be attached to a variety of different drums. We use audio signal processing and machine learning techniques to have robotic musical instruments that "listen" to themselves using a single centrally located microphone.

It is a time consuming and challenging process to setup robotic instruments in different venues. One issue is that of mapping, that is, which signal sent from the computer maps to which robotic instrument. As the number of drums grows, it becomes more challenging to manage the cables and connections between the controlling computer and the robotic instruments. The system we propose performs timbre classification of the incoming audio, automatically mapping solenoids correctly in real-time to the note messages sent to the musically desired drum. For example rather than sending an arbitrary control message to actuator 40 the control message is addressed to the bass drum and will be routed to the correct actuator by simply "listening" to what each actuator is playing in a sound-check stage. That way actuators can be moved or replaced easily even during the performance without changes in the control software. The same approach is also used to detect broken or malfunctioning actuators that do not produce sound.

When working with mechanical instruments, there is a great deal of non-linearity and physical complexity that makes the situation fundamentally different from working with electronic sound, which is entirely "virtual" (or at least not physical) until it comes out of the speakers. The moving parts of the actuators have momentum, and changes of direction are not instantaneous. Gravity may also play a part, and there is friction to be overcome. Frequently actuators are on separate power supplies which can result in inconsistencies in the voltage. The compositional process, rehearsal and performance of "The Space Between Us" by by David A. Jaffe, in which Andrew Schloss was soloist on robotic percussion, involved hand-calibrating every note of the robotic chimes, xylophone and glockenspiel. This required 18+23+35 separate hand calibrations and took valuable rehearsal time. In this paper we describe a method for velocity calibration, that is, what voltage should be sent to a solenoid to generate a desired volume and timbre from an instrument. Due to the mechanical properties of solenoids and drums, a small movement in the relative position of these two can lead to a large change in sound output. The most dramatic of these is when during performance a drum moves out of place enough that a voltage that at the start of the performance allowed the drum to be hit now fails to make the drum sound. Depending on the musical context, this can be disastrous in a performance context. Good velocity scaling is essential for a percussion instrument to give a natural graduated response to subtle changes in gesture, e.g. a slight increase in the strength (velocity) of a stroke should not result in a sudden increase in the loudness of sound.

Issues like velocity calibration or control mapping seem quite pedestrian, or even trivial until one has grappled with this problem with real instruments. We believe that the ability of a robotic instrument to perceive at some level its own functioning is important in making robust, adaptive systems that do not require regular human intervention to function properly. We refer to this ability as "proprioception" which in its original definition refers to the ability of an organism to perceive its own status.

Finally we also describe some experiments recognizing melodic and kinetic gestures at different tempi and with variations in how they are performed. This can be viewed as an exchange of cues established before the performance especially in an improvisatory context. This allows higher-level gestures to be used as cues without requiring exact reproduction from the human performer interacting with the robotic instrument and enables a more fluid and flexible structuring of performances.

4. EXPERIMENTS

4.1 Drum Classification for Automatic Mapping

We performed an experiment to investigate the performance of a audio feature extraction and machine learning system to classify drum sounds to perform automatic mapping. The audio features used were the well known Mel-Frequency Cepstral Coefficients (MFCC) calculated with a window size of 22.3ms. These were then used as input to a Support Vector Machine (SVM) machine learning system. We collected a dataset of audio with 4 different frame drums being struck by the robot with a time of 128ms between strikes, then calculated all the MFCC of this audio, and then found the 8 highest MFCC0 (roughly corresponding to perceptual loudness) and marked these as onsets in the audio. The MFCC

Peak offset	Percent correct	Peak offset	Percent correct
0	66.38	4	90.52
1	91.95	5	86.49
2	91.67	6	86.49
3	91.95	7	77.59

Table 1. Classification accuracy of an SVM classifier The Peak offset is the offset from the time the drum is hit.

feature vectors corresponding to these onsets were used to train the classifier. A separate test data set was also collected. Percussive sounds can be challenging to classify as there is not a lot of steady state spectral information. The results of this experiment gave a classification accuracy of 66.38%, as shown in the first line (Peak offset 0) in Table 1. We then performed the same experiment but using instead different offsets from the highest peak in window sizes of 22.3ms. When we classified all frames with the frame immediately after the highest peak, we obtained a classification accuracy of 91.95%. We interpret this result to mean that the resonance after the transient is clearly distinguishable for different drums, whereas the transient at the onset is fairly similar for different drums. This performance quickly degrades as we move away from the onset.

This performance quickly degrades as we move away from the onset. These results are for individual 22.3ms frames so it is easy to get 100% correct identification by voting across the entire recording which can then be used for the automatic mapping. When we setup the robotic instrument we actuate each solenoid in turn, classify the audio and then set the appropriate mappings so that the control software can address the actual frame drums rather than the actuators.

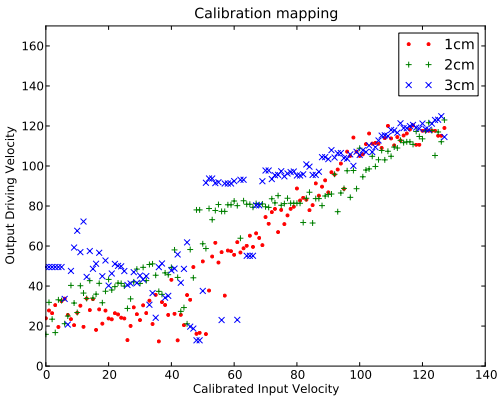


Figure 2. Mapping from calibrated input velocities to output driving velocities for different distances

4.2 Timbre-Adaptive Velocity Calibration

The acoustic response of a drum both in terms of perceived loudness and timbral quality is non-linear with respect to linear increases in voltage as well as to the distance of the solenoid to the vibrating surface. In the past calibration was performed manually by listening to the output and adjusting the mapping of input velocities to voltage until smooth changes in loudness and timbre where heard. In this section we describe how to derive an automatic data-driven mapping that is specific to the particular drum.

Our first objective is to achieve a linear increase in loudness with increasing MIDI velocity for a given fixed distance between beater and drumhead. However, in practice, the beater may be mounted on a stand and placed next to the drumhead mounted on a different stand. Thus the distance between beater and drumhead will vary depending on setup, and may even change during a performance. Thus a second objective is to achieve a similar loudness versus MIDI velocity (corresponding to voltage) curve over a range of distances between beater and drumhead.

To achieve these objectives we collected audio for all velocity values and three distance configuration (near 1cm, medium 2cm, far 3cm). The loudness and timbre variation possible is captured by computing MFCC for each strike. More specifically for each velocity value and a particular distance we obtain a vector of MFCC values. The frequency of beating was kept constant at 8 strikes per second for these measurements. The first MFCC coefficient (MFCC0) at the time of onset is used to approximate loudness. Plots of MFCC0 for the distance configurations are shown in 3(a).

In order to capture some of the timbral variation in addition to the loudness variation we project our MFCC vectors to a single dimension (the first principal component) using Principal Component Analysis (PCA) [5]. As can be seen in 3(c) the PCA0 values follow closely the loudness curve. This is expected as loudness is the primary characteristic that changes with increasing velocity. However, there is also some information about timbre as can be seen by the "near" plot that has higher variance in PCA0 than in MFCC0.

Our goal is to obtain a mapping (from user input calibrated velocity to output driving velocity) such that linear changes in input (MIDI velocity) will yield approximately linear changes in the perceived loudness and timbre as expressed in PCA0. We utilize data from all the three distance configurations for the PCA computation so that the timbrespace is shared. That way even though we get separate calibration mappings for each distance configuration they have the property that the same calibrated input value will generate the same output in terms of loudness and timbre independently of distance.

In order to obtain this mapping we quantize the PCA0 values for each distance configuration into 128 bins that correspond to the calibrated input velocities. The generated

mapping is the wrong way i.e from output driving velocities to calibrated input velocities and is not an injection (one-to-one function) so it can not be directly inverted. To invert the mapping for each calibrated input velocity (or equivalently quantized PCA bin) we take the average of all the output driving velocities that map to it as the output driving value. This calibration mapping is shown in Figure 2. Figures 3(b) and 3(d) show how changing the calibrated input velocity linearly results in a linearized progression through the timbrespace (PCA0) and loudness (MFCC0). In these graphs we show directly the results of this calibration but it is also possible to fit lines to them. In either case (direct calculated mapping or line fit) the calibrated output changes sound more smooth than the original output.

4.3 Gesture recognition using Dynamic Time Warping

Collaborating musicians frequently utilize high-level cues to communicate with each other especially in improvisations. For example a jazz ensemble might agree to switch to a different section/rhythm when the saxophone player plays a particular melodic pattern during soloing. This type communication through high level cues is difficult to achieve when performing with robotic music instruments. In our performances we have utilized a variety of less flexible communication strategies including pre-programmed output (the simplest), direct mapping of sensors on a performer to robotic actions, and indirect mapping through automatic beat tracking. The final experiments described in this paper show how high-level gesture recognition that is robust to changes in tempo and pitch contour can be correctly identified and used as a cue. Our system is flexible and can accept input from a wide variety of input systems. We show experimental results with the radiodrum as well as melodic patterns played on a vibraphone. There has been considerable work done in the area of using Dynamic Time Warping for gesture recognition, including work done by Akl and Valaee [1] and Liu et al. [8].

For the first experiment, we used the most recent iteration of the radiodrum system, a new instrument designed by Bob Boie that dramatically outperforms the original radiodrum in terms of both data rate and accuracy. We instructed a professional musician to generate 8 different instances of 5 types of gestures, which were an open stroke roll, a sweep of the stick through the air, a pinching gesture similar to the pinch to zoom metaphor on touchscreens, a circle in the air and a buzz roll. We collected (X, Y, Z) triplets of data from the sensor at a sample rate of 44100Hz and then downsampled this data to 120Hz to allow us to compare gestures that were on average 1-2 seconds in length while remaining within the memory limits of our computer system. We empirically determined that this rate captured most of the information relevant to gesture recognition.

From this data, the similarity matrix of each gesture to each other gesture is computed. Dynamic Time Warping [10] is used to compute an alignment score for each pair of gestures that correspond to how similar they are. For each query gesture we return a ranked list based on the alignment score and calculate the average precision for each gesture. As can be seen from Table 2 gesture identification is quite reliable in both cases.

radiodrum			Vibraphone		
Gestures	AP	P@1	Gesture	AP	P@1
roll	0.866	1.0	pattern1	0.914	1.0
sweep	0.980	1.0	pattern2	0.812	0.9
pinch	0.837	1.0	pattern3	0.771	0.9
circle	1.000	1.0	pattern4	0.882	1.0
buzz	0.978	1.0	pattern5	0.616	0.9
MAP	0.931	1.0	MAP	0.799	0.94

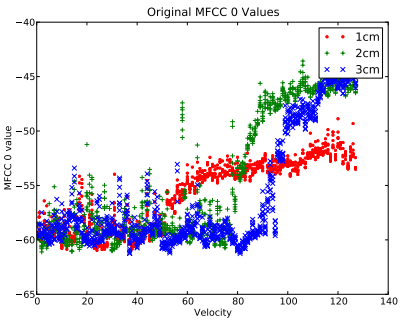
Table 2. Average precision for different gestures on the radiodrum and vibraphone. The Mean Average Precisions (MAP) are 0.931 and 0.799.

5. CONCLUSIONS AND FUTURE WORK

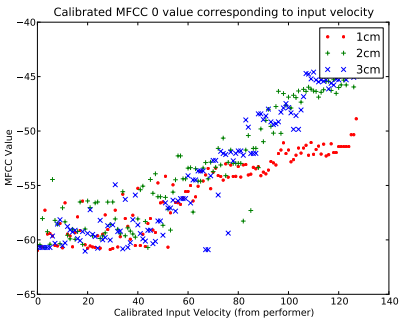
We have shown how techniques from MIR can be adapted and used to solve practical problems in music robotics. More specifically we show how audio classification can be used for automatic mapping, principal component analysis can be used for velocity/timbre calibration and dynamic time warping for gesture recognition. This system has not yet been tried in performance, and we are currently working with musicians to deploy this system in a live setting. In the future we plan to extend this work utilizing more sensors including multiple microphones on both the robot and the performers. To obtain the maximum possible dynamic range we plan to have multiple actuators placed at different distances on the same drum so that the ones that are far are used for loud sounds and the ones that are near are used for soft sounds. The proposed calibration method will be used to drive seamlessly both actuators. We would also like to investigate how MIR techniques can be used to "teach" the robot to play and recognize rhythmic and melodic patterns.

6. ACKNOWLEDGMENTS

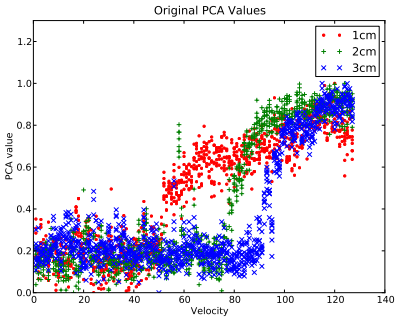
We would like to thank Gabrielle Odowichuk and Anthony Theocharis for help in collecting data. We thank the National Sciences and Engineering Research Council (NSERC) and Social Sciences and Humanities Research Council (SSHRC) of Canada for their financial support.



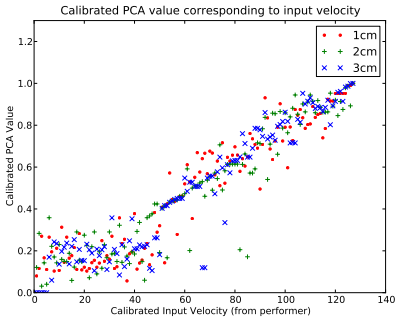
(a) MFCC-values



(b) MFCC-inverse-mapping



(c) PCA-values



(d) Calibrated PCA

Figure 3. Velocity Calibration based on loudness and timbre

7. REFERENCES

[1] A. Akl and S. Valaee. Accelerometer-based gesture recognition via dynamic-time warping, affinity propagation, and compressive sensing. In *ICASSP*, pages 2270 –2273, 2010.

[2] M. Burtner. A theory of modulated objects for new shamanic controller design. In *Proc. Int. Conference on New Interfaces for Musical Expression (NIME)*, 2004.

[3] G. Hoffman and G. Weinberg. Gesture-based human-robot jazz improvisation. In *Int. Conf. on Robotics and Automation (ICRA)*, pages 582–587, 2010.

[4] G. Hoffman and G. Weinberg. Shimon: an interactive improvisational robotic marimba player. In *CHI Extended Abstracts*, pages 3097–3102, 2010.

[5] I.T. Jolliffe. *Principal Component Analysis*. Springer, 2002.

[6] A. Kapur. A history of robotic musical instruments. In *Proc. of the Int. Computer Music Conf. (ICMC)*, 2005.

[7] A. Kapur, E. Singer, M. Benning, G. Tzanetakis, and Trimpin. Integrating hyperinstruments, musical robots and machine musicianship for north indian classical music. In *Proc. Int. Conf. on New Interfaces for Musical Expression (NIME)*, 2005.

[8] J. Liu, Z. Wang, L. Zhong, J. Wickramasuriya, and V. Vasudevan. uwave: Accelerometer-based personalized gesture recognition and its applications. *IEEE Int. Conf. on Pervasive Computing and Communications*, pages 1–9, 2009.

[9] R. Rowe. *Machine Musicianship*. MIT Press, 2001.

[10] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 26(1):43–49, 1978.