

Towards the One-Man Indian Computer Music Performance System

Ajay Kapur^{*}, George Tzanetakis^{*}, Andrew Schloss^{*}, Peter Driessen^{*}, and Eric Singer[†]

^{*}Music Intelligence and Sound Technology Interdisciplinary Centre (MISTIC), University of Victoria
ajay@uvic.ca, gtzan@cs.uvic.ca, aschloss@finarts.uvic.ca, peter@ece.uvic.ca

[†]League of Electronic Music Urban Robots (LEMUR)
e@ericsinger.com

Abstract

This paper presents progress towards a system for human to robot musical performance. Specifically, it focuses on a paradigm based on North Indian classical music, drawing theory from ancient tradition to guide aesthetic and design decisions. Custom built human computer interfaces combined with musical robotic systems using software which take advantage of the state of the art computer music algorithms and theory are presented. The end result of the project is spawning a one-man Indian computer music performance system.

1 Introduction

Research on the process of obtaining sensor data from a human to trigger a machine to deduce a form of physical response is essential in developing advanced human computer interaction systems of the future. Conducting these types of experiments in the realm of music is obviously challenging, but is particularly useful as music is a language with traditional rules which must be obeyed to constrain a machines response. Thus, it is easy for successful algorithms to be observed from the scientists and engineers writing the code. More importantly, it is possible to extend the number crunching into a cultural exhibition, building a system that contains a novel form of artistic expression which can be used on the stage.

The goal of this research is to make progress towards a system for human to musical robotic performance. There has been a number of engineers and artist who have made headway in this area. The art of building musical robots has been explored and developed by musicians and scientists such as (Trimpin 2000), (Takanishi and Maeda 1998), (Jorda 2002), (MacMurtie), (Baginsky) and (Raes). A comprehensive review of the history of musical robots is described in (Kapur 2005).

The area of machine musicianship is another part of the puzzle. Robert Rowe describes a computer system which can analyze, perform and compose music based on traditional music theory (Rowe 2004). Other systems which

have influenced the community in this dominion are Dannenberg's score following (Dannenberg 1984), George Lewis's Voyager (Lewis 2000), and Pachet's Continuator (Pachet 2002).

There are few systems that have closed the loop to create a real live human/robotic performance system. Audiences who experienced Mari Kimura's recital with the LEMUR GuitarBot (Singer, Larke et al. 2003) can testify to its effectiveness. Gil Weinberg demonstrated Haile at ICMC 2005, his team's robotic drum player which can tap along with a human musician (Weinberg, Driscoll et al. 2005).

This paper describes a human to robot performance system based on North Indian classical music, drawing theory from ancient tradition to guide aesthetic and design decisions. Section 2 will present the tools used conduct early experiments including custom built human computer interfaces and musical robotic systems. Software engineering design and algorithms, in computer music languages are discussed in detail in Sections 3 and 4. The end result of this project is spawning a one-man Indian computer music system, which can be performed live on stage.

2 Tools

As shown in Figure 1, in order to carry out this research, one must have access to a custom built musical controller, a functional musical robot, and programmable computer music software. This section describes the tools used in the early stages of experimentation. Keeping with North Indian tradition, in this project the human will be performing a sitar, a 19-stringed gourd-like instrument, and the machine will be providing rhythmic accompaniment. In the early stages of design, algorithms are tested using audio samples played through speakers, as well as experimentation using simple percussion robots.

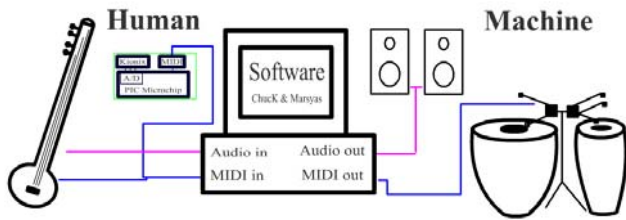


Figure 1 - Human/Machine Music Performance System Diagram.

2.1 Machine Perception

In order for a robot to interact with a human it must be able to sense what the human is doing. In a musical context, the machine can perceive human communication in three general categories. The first is directly through a microphone, amplifying the audio signal of the human's musical instrument. The second is through sensors on the human's musical instrument. The third is through sensors placed on the human's body, deducing gestural movements during performance using camera arrays or other systems for sensing, analogous to the machine's eyes.

Sound. The sound of the sitar is easily obtained via a contact microphone/pickup clipped on to the bridge of the instrument.

Sensors on Musical Instrument. The ESitar (Kapur, Lazier et al. 2004) is a hyperinstrument which obtains musician's gestural data of traditional performance techniques. Fret detection and thumb pressure are the two key pieces of information used to capture musical meaning and method. This year, development on the ESitar 2.0 has begun. A specialized sitar maker in Miraj, India, named Shahid -- was sought out to custom build a traditional sitar which could accommodate digitization.

Sensors on the Human Body. The KiOms (Kapur, Yang et al. 2005) are controllers which can be attached to the human body (shown in Figure 2). These interfaces were influenced by the first ESitar, where a KiOm-like device was attached to the head of the musician. The KiOms convert 3-axis accelerometer readings to MIDI for use with any audio software, or synthesizer. Twelve KiOms were built, so the performer can put them on their head, hands, arms, and feet.



Figure 2 - The KiOm Wearable Sensor.

2.2 LEMUR ModBots

ModBots (Singer, Feddersen et al. 2004) are modular robots that can be attached virtually to any fixture. These devices controlled by Pulse Width Modulation (PWM) on a microcontroller, respond to a varying supply of voltage to control a linear actuator rotary motor. These robotic devices listen for MIDI signals and can be affixed to simple folk Indian drums, a drum set, or any percussion instrument from around the world.

2.3 Chuck

Chuck (Wang and Cook 2003) is a concurrent, strongly-timed audio programming language that can be programmed on-the-fly. It is said to be *strongly-timed* because of its precise control over time. Concurrent code modules that independently process/sonify different parts of the data can be added during runtime, and precisely synchronized. These features make it straightforward to rapidly implement and experiment with composition algorithms, and also to fine tune them on-the-fly. Chuck is a language to get ideas implemented quickly, and test if concepts are a success.

2.4 Marsyas

MARSYAS (Tzanetakis and Cook 2000) is a software framework for rapid prototyping and experimentation with audio analysis and synthesis with specific emphasis to music signal and music information retrieval. It is based on a data-flow architecture that allows networks of processing objects to be created at run-time. A variety of feature extraction algorithms both for audio and general signals are provided. Examples include Short Time Fourier Transform, Discrete Wavelet Transform, Spectral and Temporal Centroid, Mel-Frequency Cepstral Coefficients and many others. In addition Marsyas provides integrated support for machine learning and classification using algorithms such as k-nearest neighbor, Gaussian mixture models and artificial neural networks.

3 Development using ChuckK

Development in ChuckK fall into three categories: sitar synthesis experiments, beat accompaniment, and robotic accompaniment.

3.1 Sitar Synthesis Experiments

This section involves combining the sensor data from the ESitar, KiOms with the audio signal for the pickup to create synthesized modified sounds for artistic expression.

One simple implementation was to use the physical model unit generators embedded from STK Toolkit (Cook and Scavone 1999), such as the flute, clarinet, and even sitar models. Pitch is controlled by fret information, where as excitation is controlled by thumb pressure. Also, experiments with comb-filters, delay lines, ring modulation, and controlling moog synthesizers all proved to be successful and straightforward to implement and control.

Using the class structure inherit in ChuckK, a system to control the Line6 PodXT¹ effects unit using the sensor data obtained. Every midi message which the PodXt accepts was programmed into the class, making it easy to modify parameters and experiment with a large variety of mappings. Successful developments include controlling Wah-Wah effect with the thumb sensor, controlling delay-timing with the KiOm's, and controlling frequency modulation with the fret information.

3.2 Beat Accompaniment

This section involves generating beat accompaniment with the sensor data from the ESitar and KiOms.

The first experiment was to simply loop a series of audio files, which together represent a beat. Sensor data from the ESitar and KiOms are used to control sound effects on of the individual loops, generating an ever-changing texture of rhythm. However, important parameters such as tempo, and beat variation are not possible in this type of implementation.

Thus, the second experiment was to create arrays of sample sounds which together comprise a beat. This way, tempo can be controlled and it is easy to generate beat variation. Audio effects for each sample are still controlled by sensor data. Tempo is automatically adjusted by obtaining tapping information from a piezo mounted underneath the sitar players foot. Figure 3 is a picture of a Taal Tarang digital tabla accompaniment box, which has 24 traditional tabla cycles (*thekas*) with a number of variations on each. The beats from this box are programmed into ChuckK creating a tabla accompaniment toolbox ready to use for performance and more advanced techniques.

3.3 Robotic Control

Initial experiments interfacing the ESitar with Trimpin's eight robotic turntables has already been described (Kapur 2005). In order to start using the ModBots in this system, a special class was created, making a modular system to control the robots from any pre-written program within ChuckK. The two parameters to control, are which beater to strike, and how much voltage to send, which is equivalent to setting the volume. The first goal was to robotify the database obtained from the Taal Tarang box, which is accomplished using the same arrays but triggering beater events instead of samples. Voltage is set by sensors obtained the KiOm or ESitar.



Figure 3 - Taal Tarang Digital Tabla Accompaniment Box.

4 Development using Marsyas

Development in Marsyas fall into two categories: audio-based beat detection experiments and MIR experiments.

4.1 Audio-based Beat Detection Experiments

Marsyas has a large set of tools including feature extraction algorithms and machine learning algorithms which allow for more advanced experimentation in machine audition.

One idea is to use an audio file of a tabla recording or drum beat to trigger of the ModBots. Bass drum ("Boom") detected hits would trigger one beater, where as snare drum ("Chick") detected hits would trigger another. A system was designed to automatically detect "Boom Chick" of any sound files based on subband analysis and a comparison between using a wavelet-based front-end or FFT-based front-end was conducted (Tzanetakis, Kapur et al. 2005). Another method is to use machine learning algorithms to train a computer what different strokes sound like on the tabla, and automatically classify them in real time (Tindale, Kapur et al. 2005) and trigger the appropriate ModBot.

¹ <http://www.line6.com> (Accessed March 2006)

4.2 MIR Experiments

This is the closest scenario to an interactive performance system using music information retrieval (MIR) techniques. In the playing of the sitar rhythm information is conveyed by the direction and frequency of the stroke. The thumb force sensor on the ESitar controller is used to capture this rhythmic information creating a query. The query is then matched into a database in order to provide an automatically-generated tabla accompaniment. The accompaniment is generated by matching the rhythmic information into a database containing variations of different *thekas*. (Kapur, McWalter et al. 2005)

5 Conclusions and Future Work

This paper shows progress towards a one-man computer music performance system. It describes advancement in human computer interface design, musical robot implementation, and software design for musical composition and expression. Future work includes modifications at every level of design, including completion of the ESitar 2.0, designing a robotic tabla player and completion of a MIR-based approach for human/robotic music interaction.

6 Acknowledgments

Thanks to Manjinder Benning for his code to auto detect tempo in ChucK. Thanks to Adam Tindale and Richard McWalter for their related research collaboration. Special thanks to Shahid – the sitarmaker from Miraj, India, for building such a beautiful instrument. Eternal thanks to Ustad Siraj Khan and Rakesh Kumar Parihast for their training in North Indian Classical Music, tabla and sitar. Thanks to Perry Cook and Ge Wang for designing the ChucK language and their constant support. Special thanks to Trimpin for his inspiration.

References

Baginsky, N. A. The Three Sirens: A Self Learning Robotic Rock Band.
Cook, P. R. and G. Scavone (1999). The Synthesis Toolkit (STK). International Computer Music Conference, Beijing, China.
Dannenberg, R. B. (1984). An On-line Algorithm for Real-Time Accompaniment. International Computer Music Conference, Paris, France.

Jorda, S. (2002). Afasia: the Ultimate Homeric One-Man-Multimedia-Band. International Conference on New Interfaces for Musical Expression, Dublin, Ireland.
Kapur, A. (2005). A History of Robotic Musical Instruments. International Computer Music Conference, Barcelona, Spain.
Kapur, A., A. Lazier, et al. (2004). The Electronic Sitar Controller. International Conference on New Interfaces for Musical Expression, Hamamatsu, Japan.
Kapur, A., R. I. McWalter, et al. (2005). New Interfaces for Rhythm Based Information Retrieval. International Conference on Music Information Retrieval, London, UK.
Kapur, A., E. L. Yang, et al. (2005). Wearable Sensors for Real-Time Musical Signal Processing. IEEE Pacific Rim Conference, Victoria, Canada.
Lewis, G. (2000). "Too Many Notes: Computers, Complexity and Culture in Voyager." Leonardo Music Journal 10: 33-39.
MacMurtie, C. Amorphic Robot Works.
Pachet, F. (2002). The Continuator: Musical interaction with Style. International Computer Music Conference, Goteborg, Sweden.
Raes, G. W. Automations by Godfried-Willem Raes.
Rowe, R. (2004). Machine Musicianship. Cambridge, MA, MIT Press.
Singer, E., J. Feddersen, et al. (2004). LEMUR's Musical Robots. International Conference on New Interfaces for Musical Expression, Hamamatsu, Japan.
Singer, E., K. Larke, et al. (2003). LEMUR GuitarBot: MIDI Robotic String Instrument. International Conference on New Interfaces for Musical Expression, Montreal, Canada.
Takanishi, A. and M. Maeda (1998). Development of Anthropomorphic Flutist Robot WF-3RIV. International Computer Music Conference, Michigan, USA.
Tindale, A. R., A. Kapur, et al. (2005). Indirect Acquisition or Percussion Gestures Using Timbre Recognition. Conference on Interdisciplinary Musicology, Montreal, Canada.
Trimpin (2000). SoundSculptures: Five Examples. Munich MGM MediaGruppe Munchen.
Tzanetakis, G. and P. R. Cook (2000). "Marsyas: a Framework for Audio Analysis." Organized Sound 4(3).
Tzanetakis, G., A. Kapur, et al. (2005). Subband-based Drum Transcription for Audio Signals. IEEE International Workshop on Multimedia Signal Processing, Shanghai, China.
Wang, G. and P. R. Cook (2003). ChucK: A Concurrent, On-the-fly Audio Programming Language. International Computer Music Conference, Singapore.
Weinberg, G., S. Driscoll, et al. (2005). Haile - A Preceptual Robotic Percussionist. International Computer Music Conference, Barcelona, Spain.