HAILE – AN INTERACTIVE ROBOTIC PERCUSSIONIST

Gil Weinberg
Georgia Tech
Music Department
840 McMillan St.
Atlanta, GA, 30332

Scott Driscoll
Georgia Tech
School of Mechanical Engineering
771 Ferst Drive
Atlanta, GA, 30332

Mitchell Parry
Georgia Tech
College of Computing
85 5th Street, NW
Atlanta, GA, 30332

ABSTRACT

We present the current state of development of an interactive robotic percussionist, titled Haile, which is designed to demonstrate musicianship. We define robotic musicianship in this context as a combination of low- and high-level music perception skills, symbolic generation and transformation of musical material, and the capacity to produce varied acoustic responses.

1. INTRODUCTION

Musical robots, unlike other devices that electronically amplify and reproduce music through speakers, can convert digital musical instructions directly into acoustic generation of sound. Robots, therefore, can bring together the unique advantages of digital music (such as in the ability to perform stochastic compositional algorithms or to generate musical outcome in virtuosity and ranges not possible by human players) and the versatility and richness of acoustic generation of sound that cannot be imitated by electronic means. Musical robots also provide a visual and physical representation of the sound generation process, which unlike speakers, can help establish performance cues for players and audiences. We believe that such a visual and physical representation can also be helpful as an educational tool. Current research directions in musical robotics focus mostly on sound production and rarely address perceptual aspects of musicianship, such as listening, analysis, improvisation, or group collaboration. Both "robotic instruments" (mechanically automated instruments that can be played by live musicians or triggered by pre recorded sequences such as in [5] [8] and [9]) and "anthropomorphic robots" (hominoid robots that attempt to imitate the action of human musicians such as in [10] [11] and [14]) function mostly as mechanical apparatuses that follow deterministic rules. Only few attempts have been made to develop "perceptual" robots that are controlled by neural networks or other autonomous methods (for example [1]). Our goal for the Haile project is to develop a robotic percussionist that would demonstrate perceptual, physical, and social aspects of musicianship. Ultimately, such a robot would be able to analyze live musical input in real-time and react in an expressive manner by generating responsive acoustic responses that would inspire humans to interact with it in novel manners.

2. GOALS AND MOTIVATIONS

Our main goal for this project is to combine elements of music perception and robotics to form a novel musical experience that is based on responsive and inspiring human-machine interactions. Mechanically, we plan to develop a dexterous robotic apparatus that will be able to translate perceptually based performance algorithms into a virtuosic acoustic performance, extending and enhancing what is possible by human players. Our goal is not to fully capture the versatility of human players but to create a sonic richness that facilitates expressive musical interaction with human players. We believe that by creating a compelling physical apparatus that provides a visual connection to the sound production process, we can encourage humans to collaborate expressively with the machine. Our challenge here is to create a device that is familiar and stimulating both audibly and visually for humans to perform with. Perceptually, our goal is to detect fundamental musical aspects such as note onsets, pitch, amplitude and timbre and to develop analysis algorithms for high-level rhythmic aspects such as stability and similarity. Unlike the robotic instrument approach, Haile is therefore designed to utilize autonomous behaviors that support expressive collaboration with human musicians. Also, in contrast to most commercial humanized robots, Haile is designed to listen and improvise, not merely translate a set of instructions in a mechanical manner. As opposed to the autonomous generative approach, Haile's improvisation algorithms are based on transformation and modification of human input.

3. IMPLEMENTATION

In the current state of our research, we achieved some of these goals, addressing both robotics and perceptual challenges. Using a microphone installed on a Pow Wow drum, Haile can detect pitch, volume, and rhythmic aspects of human drumming and utilize this analysis to generate rhythmic responses that are based on stochastic modification in sequential and synchronous manners. We also developed a software application that listens to note onsets of human players and responds with improvisational algorithms that are based on a perceptual model of rhythmic stability and similarity (based on [4] and [12]). In terms of sound production, Haile can currently hit a drum in a variety of drumhead locations using different hit strength and speeds.

3.1. Robotics

Our approach for robotics stemmed from the attempt to facilitate artistic virtuosic performance as well as from our interest in providing an easy-to-use environment for novices to program and compose for Haile. To address this educational goal, we decided to use the graphical programming environment Max/MSP [6] and the Teleo System [13], which is designed for rapid prototyping of music applications that interact with the physical world, controlling solenoids, servos, motors, etc. To address our performance goals, we developed a robotic arm that can hit the drumhead in different locations, speeds and strengths. At this point, we have not yet implemented a second robotic arm to dampen the skin and change hitting materials for further enrichment of acoustic production. Our initial prototype is designed to explore the capabilities and limitations of the Teleo system and to provide basic drumming functionality for experimentation with human collaborators. Haile can currently adjust the sound of a hit in two manners: it can adjust pitch by striking in different locations and volume by hitting harder or softer. The main hardware components consist of a linear slide driven by a Pittman DC gear motor (see Figure 1) and a hitting mechanism that uses a Ledex solenoid (see Figure 2).

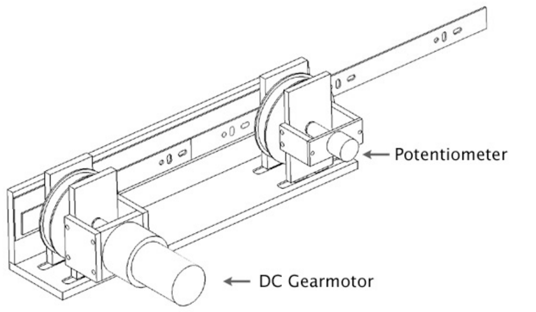

Figure 1. Haile's slider mechanism - A DC Gearmotor and a potentiometer are used to provide continuous position control to any point within a range of 10 inches.

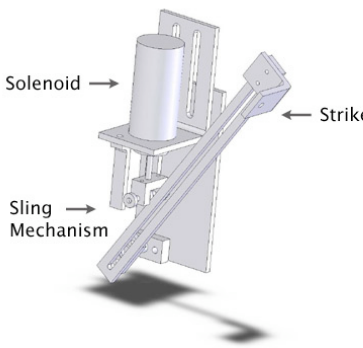

Figure 2. Haile's hitter mechanism is designed to take advantage of a solenoid's exponential force curve. It "throws" the Striker similarly to a piano key.

Pulse width modulated (PMW) and digital output signals from the USB controlled Teleo board drive the motor and solenoid through h-bridge amplifiers. Slider position is fed back through the Teleo's analog inputs using a rotational potentiometer. As opposed to robotic drumming systems that allow hits at only a few discrete locations, Haile's arm can be moved to any location within a range of 10 inches. Position control is calculated within Max/MSP using a variant of simple Proportional Derivative (PD) control, with a 100Hz control signal. Although fairly simple to implement, one drawback to this approach is that control effectiveness is dependent on the USB communication, which is in turn subject to other USB loads, and ultimately, the non-real-time Apple operating system. Future versions will incorporate direct low level position control via microprocessor, using USB only to send strike times and position set points. The hitting mechanism is loosely inspired by a piano action and was designed to enable quick design configuration changes during development. The hitter can strike at 15 Hz with approximately 10 noticeable volume levels, and can be moved from lowest to highest pitch (strike location) at 3-4Hz.

3.2. Perception

Before implementing the analysis algorithms for high-level musical percepts of rhythmic stability and similarity, we focused on detecting note onsets, pitch (which corresponds to hit location on the drumhead), and amplitude (which corresponds to hit strength). The Max/MSP object bonk~ was used to detect hit onset times while the object pitch~ was used to ascertain pitch and timbre information. The analyzed audio was collected from a microphone placed 1.5 inches above the rim of the drum. Pre-filters and other bonk~ parameters were tuned to optimize detection accuracy, which was complicated by loud hits masking softer following hits, noises from the hands contact with the drum skin, and additional ambient sounds. In general, a "hit" was characterized as a sharp change in amplitude and spectral composition. Pitch information was obtained by looking at the amplitudes of spectral peaks received from the pitch~ object every time a hit was recorded. Currently we can detect about 3 or 4 different pitch ranges with reasonable accuracy. Detection of pitch on a drum is not a trivial task since the same approximate frequencies are excited with each hit, and high frequencies are sometimes equally present in both low and high sounds (due to the initial slap). We also attempted to ascertain pitch in real-time, using only a small analysis window, which limits the fidelity of the FFT. In addition to this low-level analysis, we developed a higher-level rhythmic analysis application in a Max/MSP External format. This part of the application has not been implemented in the robot yet and currently exists in software form only. We based our application on a number of computational approaches to rhythmic similarity and stability. Similarity comparisons typically focus on how well two rhythms overlap. For instance, Paulus and Klapuri correlate low-level features of the audio signal [7], Tanguiane counts the number of coincident onsets [12], and Coyle and Schmulevich correlate a sequence of note duration ratios [2]. While similarity represents the identicalness between rhythms, stability represents the expectedness of a single rhythm. For instance, the most stable rhythms are metronomic, while the least stable are chaotic. We use a tradeoff between similarity and stability to generate new rhythms given an example. We implement Desain and Honing's computational model for rhythmic stability based on the relationship between pairs of adjacent note durations [4]. Adjacent duration pairs are rated according to their perceptual expectancy. This depends on three main criteria: perfect integer relationships are favored, ratios have inherent expectancies (*i.e.,* 1:2 is favored to 1:3 and 3:1 is favored to 1:3), and durations of 0.6 seconds are preferred. Desain and Honing define this expectancy as the following [3]:

$$E_b(A,B) = \int_0^r \left(\text{round}(r) - r\right) \times \left|2\left(r - \text{floor}(r) - 0.5\right)\right|^p \times \text{round}(r)^d \, dr, \tag{1}$$

where $r = \max(A/B, B/A)$ represents the (near) integer relationship between note durations. This function is symmetric around r=1 when the total duration is fixed. Generally, the expectancy function favors small near-integer ratios and becomes asymmetric when the total duration varies, exhibiting the bias toward the 0.6 second tempo interval. In addition to basic note durations, Desain and Honing consider implicit durations comprising two or more adjacent notes. They compute the stability of a rhythmic pattern by evaluating every pair of adjacent time intervals according to the expectancy function. Desain suggests using the ratio between the sum of two intervals and a preferred interval, $(A+B)/T_{pref}$, to favor a 0.6 second tempo [4]. We implement this using a likelihood function for tempos defined as a lognormal distribution [15]:

$$\mathrm{L}_{\text{tempo}}(A+B) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{ -\frac{1}{2\sigma^2} \left[\log_{10}\left(\frac{A+B}{T_{pref}} \right) \right]^2 \right\}, \tag{2}$$

where T_{pref} = 1.2 and σ = 0.25. So far, we have treated inverted ratios identically. However, Desain indicates that the ratio 3:1 is stronger than 1:3. To incorporate this, we penalize ratios less than one. This accounts for inversions. However, a more robust solution would estimate the true likelihood function for all ratios, by collecting a histogram on a large representative corpus. We numerically integrate Equation (1) by first sampling the function inside the integral for ratios between 1 and 10 at steps of 0.001. This provides a smooth representation of the function before integration. Then, we simply compute a cumulative sum of this function, approximating E_b for ratios between 1 and 10. Equation (1) is then scaled by Equation (2) to account for the preferred tempo, and attenuated by a 0.5 scale factor for integer ratios less than one. To compute the stability rating for an entire rhythmic pattern, we consider each onset in turn. The stability of an onset is the sum of the stabilities of all duration pairs it divides. The stability score for the pattern is the geometric mean of the onset stabilities. In order to generate new rhythms, we combine the stability rating with a similarity metric. Our similarity rating is derived from the binary representation described by Tanguiane **[12].** We quantize all measure-length rhythms to 48 time steps and shift so that the first onset is on the first beat of the measure. Then, we count the number of overlapping onsets, allowing for small deviations at a cost. For instance, if two onsets are one step away they count half as much. We can retrieve new rhythms from a database based on its similarity to an input rhythm and a desired stability level. Based on the 48-step measure, we construct a database of over 100,000 rhythms. We do this by considering half, quarter, quarter-triplet, eighth, eighth-triplet, and sixteenth notes constructed by recursively dividing each half-measure into all combinations of twos and threes. Then, every combination of rests is considered (except on the first onset). We imagine that a composer might specify an initial rhythm and then modulate that rhythm by a desired stability score. In this case, the algorithm attempts to satisfy two constraints: the relative stability and the similarity to the original rhythm. In a more performance driven application, we record measure-length rhythms from a performer and respond with more or less stable versions of that rhythm according to user settings or a preprogrammed stability composition. In order to support real-time and performance driven applications we implement the stability and similarity algorithms as externals to the Max/MSP programming environment. The stability object accepts a list of onset times in seconds and outputs a floating-point stability score. The similarity object contains the database of rhythms (one 64-bit integer per rhythm) and the similarity algorithm. In our current implementation, it accepts a list of onset times as an input rhythm, the stability of the input, a stability coefficient, and a similarity coefficient. The stability coefficient indicates the desired output stability as a floating point number between zero and one; one indicates the most stable rhythm in the database, zero the least stable, and ½ equates to input stability. We linearly interpolate between the minimum, maximum, and input stabilities to find intermediate values. The similarity coefficient determines the relative contribution of similarity and stability during retrieval. A similarity coefficient of one requires that the output is identical to the input. A similarity coefficient of zero requires that the output has the desired stability regardless of similarity. Intermediate values offer a compromise between the two.

4. PROOF OF CONCEPT – "POW"

As a proof of concept for our robotic and perceptual approaches we wrote a musical piece for Haile titled "Pow" (composed by Gil Weinberg and Scott Driscoll [15]). The composition incorporates our developments in robotics, low-level musical analysis, and employs several models of interconnected musical collaboration. The piece begins with a robotic imitation of human drumming based on a sequential decentralized approach [16]. It develops into a centralized call-and-response improvisatory section where Haile transforms the analyzed rhythmic material played by human players using stochastic algorithms. The analyzed pitch, amplitude and rhythmic data is used for the generation of simple stochastic manipulations such as dividing hits into doubles or triplets or inverting the pitch of the recorded segments. The piece ends with a structured section where Haile and the human players interact in a synchronous manner, taking turns as soloists. We believe that the unique human-machine collaboration established in the performance of "Pow" (see Figure 8) can lead to a novel and exciting musical outcome that cannot be conceived by other means.

Figure 8. The composition "Pow" for a robot and humans. Performed in concert at the Eyedrum, Atlanta.

5. FUTURE WORK

Hand drums' timbre is highly dependent on a variety of factors including hand shape, contact area, contact location, contact duration, and pressure on the skin. Human players use their palm, fingers, fingernails, and different hand shapes on different areas of the drum to vary and enrich their playing**.** For our next design, therefore, we plan to add another arm that would dampen and/or stretch the drumhead skin. We also plan to make the robot more visually communicative by utilizing an anthropomorphic design with larger hit motions and more hand-shaped strikers. Perceptually, our short-term goal is to embed in Haile the similarity and stability algorithms described above and to design a graphical user interface that would allow users to determine the nature of interaction based on the detected rhythmic stability and similarity. In the long term we plan to model additional high-level percepts and embed them in a real-time collaborative context. From an educational point of view, we plan to create an intuitive environment for learners to easily program and interact with Haile. We intend to develop a constructionist-based educational application for hands-on experimentation with mathematical, scientific, and technological aspects of music. The proposed interdisciplinary pedagogy will address aspects such as rhythm, beat, meter, sound production, percussion mechanics, programming, rhythmic variations, collaborative improvisation, probability, and polyrhythm among others.

6. REFERENCES

[1] Baginsky, N.A. "The Three Sirens: A Self Learning Robotic Rock Band," *http://www.the-three-sirens.info*.

[2] Coyle, E.J. and Shmulevich, I. A. "System for Machine Recognition of Music Patterns," *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Seattle, WA, 1998.

[3] Desain, P. and Honing, H. "Quantization of Musical Time: A Connectionist Approach," In P. M. Todd and D. G. Loy (eds.), *Music and Connectionism*, Cambridge: MIT Press, 1991.

[4] Desain, P. and Honing, H.J.. "Rhythmic stability as explanation of category size." *Proceedings of the International Conference on Music Perception and Cognition* (pp. CD-rom). Sydney: UNSW, 2002.

[5] Jordà, S. "Afasia: The Ultimate Homeric One-man multimedia- band." *Proceedings of the 2002 Conference on New Interfaces for Musical Expressio*, Dublin, Ireland, 2002.

[6] Max/MSP *http://www.cycling74.com*.

[7] Paulus, J. and Klapuri, A. "Measuring the Similarity of Rhythmic Patterns," *Proceedings of the International Conference on Music Information Retrieval*, Paris, France, 2002.

[8] Singer, E., Feddersen, J., Redmon C., and Bowen, B. 2004. "LEMUR's Musical Robots," *Proceedings of the 2003 Conference on New Interfaces for Musical Expression*, Hamamatsu, Japan, 2004.

[9] Singer, E., Larke, K. and Bianciardi, D. "LEMUR GuitarBot: MIDI Robotic String Instrument," *Proceedings of the 2003 Conference on New Interfaces for Musical Expression*, Montreal, Canada, 2003.

[10] Sony QRIO conductor Robot. *http://www.sony.net/SonyInfo/QRIO/works/20040325e.html*.

[11] Takanishi, A., Maeda, M. Development of Anthropomorphic Flutist Robot WF-3RIV. *Proceedings of International Computer Music Conference*, 328-331, 1998.

[12] Tanguiane, A. *Artificial Perception and Music Recognition.* New York: Springer-Verlag, 1993.

[13] Teleo system *http://www.makingthings.com,* Accessed *2005.*

[14] Toyota Trumpet Robot. *http://www.toyota.co.jp/en/special/robot*.

[15] Weinberg, G., and Driscoll S., "Pow" *http://undertow.arch.gatech.edu/homepages/gweinberg/pow.mov*.

[16] Weinberg, G. "Interconnected Musical Networks - Toward a Theoretical Framework," To be published in *Computer Music Journal* 29(2) Cambridge: MIT Press (2005)