9th IEEE-RAS International Conference on Humanoid Robots
December 7-10, 2009 Paris, France

Using Miniature Humanoids as Surrogate Research Platforms

Robert Ellenberg, David Grunberg, Paul Y. Oh*, Youngmoo Kim
Drexel University, Philadelphia, PA, 19104
rwe24@drexel.edu, dkg43@drexel.edu, paul@coe.drexel.edu,ykim@drexel.edu

Abstract— **As research in biped gait, human interaction, and social robotics expands, hardware to explore these fields is becoming valuable. The high cost and risk of full-sized humanoid robots prevents many small laboratories for exploring these areas, however. In recent years, many models of miniature humanoid robot have been introduced to the pro-sumer market. These small humanoid robots cost 1000 to 2000 USD. They are easy to operate and maintain, yet lack articulation and processing power of full-size humanoids. The objective of the authors' research is to implement a miniature humanoid robot as a surrogate for larger humanoid robotics. To demonstrate this, a miniature humanoid was used to explore creativity and dance with a humanoid robot. The authors' particular interest in humanoids is dance as an expression of creativity and hence intelligence. To move beyond preprogrammed choreography requires the ability to listen to music, interpret rhythm and express a message through dance. Employing miniature humanoids as surrogate test platforms reduces risk before algorithms are ported to full-size ones. Experimental results are presented that support the viability of this approach.**

I. INTRODUCTION

Recently, small-sized humanoids costing under 2000 USD have become readily and widely available. Such humanoids first arrived on the US market in 2007. Currently there are almost a dozen different models offered by companies like Futaba, Hitec and Graupner. These models differ in motion range, software sophistication and sensors, but are readily available and relatively easy to maintain. By contrast, full-size humanoids like the Honda ASIMO [1] and KAIST Hubo [2] (see Figure 1) cost at least 2 to 4 orders of magnitude more. Furthermore, they often require highly skilled personnel to maintain and repair. As such, researchers working with full-sized humanoids are understandably conservative when implementing new algorithms; any risky motion, like walking on rugged terrain, can result in a catastrophic fall that can halt research. Another point is that full-sized humanoids are not mass- or even batch-produced. This limits the number of researchers who can perform experiments on such platforms.

The authors' particular interest in humanoids stems from our desire to gift such robots with abilities to listen to music, interpret beat and mood, and express itself through dance. Our conviction is that humanoids can serve as a platform to apply machine learning algorithms and that dance can be a metric to gage creativity and hence intelligence. Towards such research, the authors procured a full-sized humanoid (Figure 1) from the KAIST Humanoid Lab in Korea. Called Jaemi Hubo, this procured humanoid is very similar in form and function as the Honda ASIMO [3].

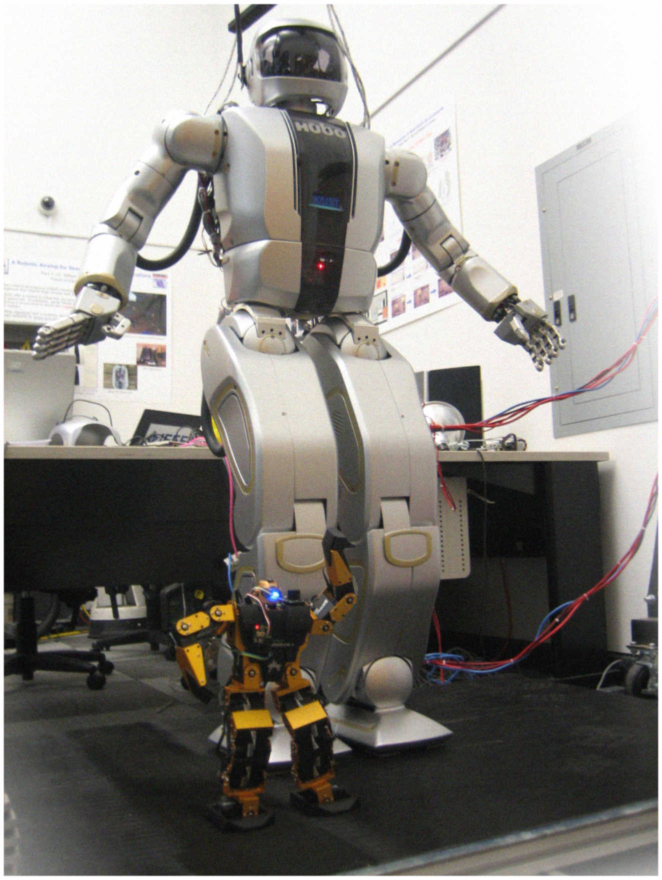

Fig. 1. Jaemi Hubo and Robonova-1 Robot demonstrating dance poses. The Robonova's low cost and simplicity make it very desirable as a research prototype

A big risk with full-scale humanoids like Hubo is the complexity of motion planning. The current system of choreographed gestures must be meticulously hand-tuned to avoid collisions and falls. Repair costs and delays make testing new motions risky. Miniature humanoids like the Robonova, however, are widely available, cheap, and relatively easy to

---

This work was funded through the National Science Foundation's PIRE: Humanoids Grant No. 0730206 and the NSF Career Grant No. IIS-0644151. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

*IEEE Member, direct all correspondence to this author

978-1-4244-4588-2/09/$25.00 ©2009 IEEE

service. As such, they could serve as prototyping tools for larger and more complex humanoids.

The primary issue with using a miniature humanoid as a prototyping tool is scaling. The Hubo has significantly different kinematics from miniature humanoids like the Robonova (Figure 1). Other factors like material strength, actuator power, limb sizes and cross sections, and relative foot area vary as well. However, from a research perspective, it is not clear how much impact such differences make on higher-level algorithms. Previous work by the authors' research team suggest this surrogate approach has merit. In [4] the Robonova was taught to recognize beats and execute dance moves. In [5] human-robot-interaction algorithms were ported to the Robonova. In [6] external processing and machine learning were implemented on the Robonova surrogate to handle walking over rough terrain. The previous work does not depend on specific motion characteristics, but on the general behavior of a humanoid robot. Such first-stage validation provides a measure of confidence and risk reduction.

This paper will present the case for minature humanoids as research platforms, and show progress on a simplified dance implementation. Section II discusses both the Robonova and Jaemi Hubo. Section III will discuss the methods of music analysis and motion synthesis, while Section V details the results to date. Finally Section VI concludes and discusses ongoing work.

II. PRIOR WORK

The state of the art in humanoid robotics has robots with 34-43 degrees of freedom (DOF), which gives approximately one joint for every major human joint, in a roughly analogous configuration. Humanoid robots such as ASIMO, HUBO (Figure 2), and HRP-2 [7] have all been developed to be equivalent to a human. The ASIMO and HUBO each have 6 DOF in each leg, 6 DOF in each arm, 1 DOF for torso twist, and 1-3 DOF to actuate the neck. These robots use rotary joints with intersecting axes to approximate the complex geometry of hips, ankles, and knees. All of these robots replicate human actuation and motion constraints, yet do not try to simulate specific human facial or body features.

Due to its size and joint complexity, however, Hubo is relatively risky to work with in a lab setting. The current iteration of the operating system requires lengthly calibration and setup at every power-on. Gesture programming requires careful hand tuning and testing to avoid body collisions and over-rotation of joints. Improper balance control could lead to falls and costly repairs, all of which would hinder smooth development. Significant infrastructure is required to perform experiments: treadmills for testing walking algorithms, rolling safety harnesses to catch the robot, and lifts to allow transport and maintenance. While technologies such as motion capture can alleviate some of the gesture design burden, this introduces additional infrastructure.

Of the many choices of miniature humanoids, Hitec's Robonova-1 (see Figure 3) was chosen as a surrogate. The Robobasic programming environment allowed gestures to be posed and tested instantly, with minimal coding.

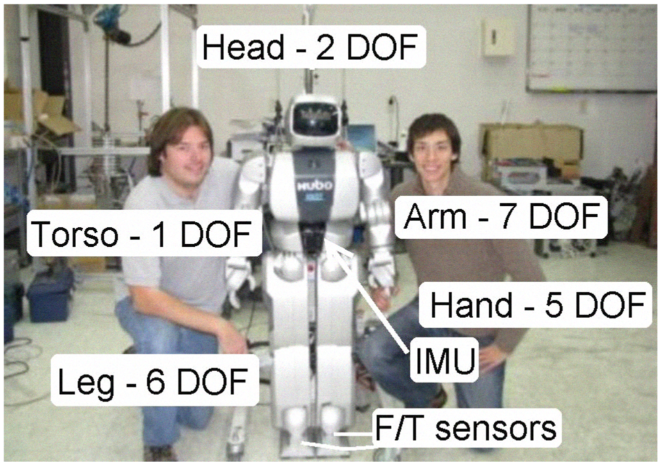

Fig. 2. The Jaemi Hubo, with 43 degrees of freedom, inertial measurement sensors, and force/torque sensors. These features allow the robot a very similar motion range to a human.

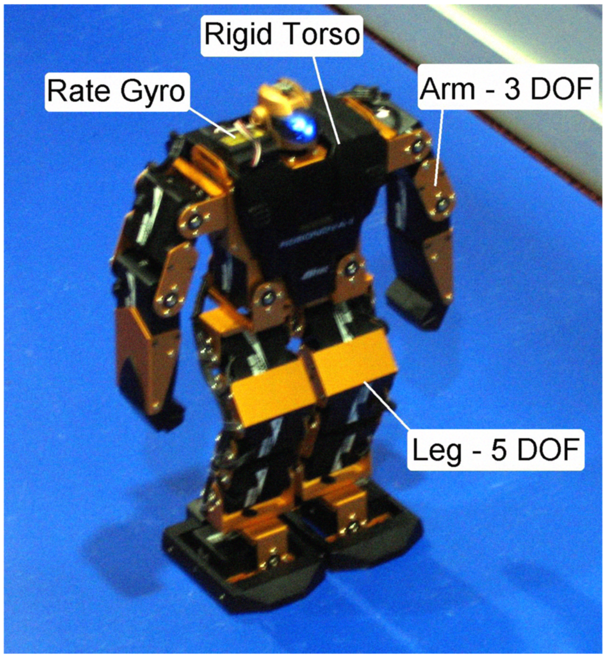

Fig. 3. Hitec's Robonova-1 uses standard robot servos for simple construction. It has 16 DOF total, distributed as shown. While its motion is limited, its small size and intuitive programming environment accelerate development

A more detailed comparison of features shows the many differences between Hubo and Robonova (Table I). Advanced sensors and human-equivalent actuation give the Hubo the ability to smoothly mimic human motion. Unlike the open loop control of robonova, the Hubo can perform continuous ZMP-based balancing.

TABLE I

A COMPARISON OF HUMANOID ROBOT SPECIFICATIONS

Category	Robonova-1	KAIST Hubo
Degrees of Freedom	16	43
Controller	8-bit MCU	2 800MHz x86 PC's
Sensors	Analog gyro Accelerometer	IMU Ankle F/T sensors
Vision	None	USB camera
Sound Input/Output	None	2 Microphones and speakers

III. ROBOT DANCE

To take advantage of the development tools afforded by Robonova, the issue of scaling must be resolved for prototype results to be useful. The significant differences in size and structure of the robots require modification of parameters, and in some cases fundamental limits on performance and speed. As such, the platform is not useful as a direct kinematic prototype. Research that depend on general characteristics of motion, rather than the specific kinematics, do not share as many scaling issues.

Dance is one such area; it is a complex but formal language for which clear rules and even notation exist. Laban Notation [8], a famous and widely used system of dance notation, describes classical and modern dance in terms of specific motion characteristics. Liwen Huang et al[9] have explored software adapting laban notation to describe dance motions independent of robot kinematics. Thus, a higher level description and planning of dance allows such software to be relatively independent. While a miniature humanoid robot lacks the articulation to perform the full spectrum of human dance motions, it could reproduce a simplified version.

The majority of the research in this area focuses on kinematics, often using motion capture data as a source. In [10], the authors developed humanoid dance motions this way. Combining laban notation and motion capture, in [11], the authors' system captures a dancer's motions and replays them accurately from primitives.

One flaw of kinematic simulation in duplicating human motions is the lack of a "realism" constraint. In [12], researchers developed a system to generate arbitrary yet realistic animated motion. Using recorded human gestures as a base, an optimization function could produce natural-looking gestures from a very high-level description. Though a real robot requires motion planning and control, its natural actuator constraints reduce the calculation necessary to produce natural-looking motions. Thus, the use of a humanoid robot could expedite development of dance. The authors prior on humanoid dance showed that primitive dance motions for a miniature humanoid could be developed in a short time [4], support this assumption.

Many humanoids are capable of producing dance motions, such as MSDanceR [13], is a robot created as a dance partner and teacher. An ASIMO robot was modified to perform a simple dance in response to external music, demonstrating real-time beat tracking[14]. These robots demonstrate the importance of the interface; to dance with even a simple robot has a unique engagement. To interact immersively with a simulation requires a more expensive and sophisticated interface as developed in [11].

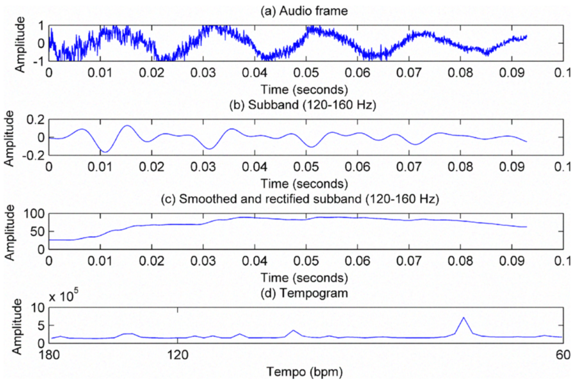

Fig. 4. Audio frame in a) original form, b) filtered to a given frequency subband, c) downsampled, d) filtered with comb filter bank to find maximum energy and resonance. The tempogram shows maximum resonance energy with respect to filter delays that correspond to the given tempos

The quality of human interaction with the dancer depends on the interface. The Keepon [15] explores how, even with minimal actuation, a robot can create life-like motions. It uses dance and music to connect with autistic children. The results are promising as a therapy, and help establish dance as a means of human-robot interaction. The suggestion of dance even with minimal actuation suggests that Robonova's simplified kinematics are capable of acheiving convincing dance motions.

The authors' simple and direct approach to robot dance divides the process into three steps: music analysis, dance creation, and dance synthesis. A music analysis stage extracts fundamental chracteristics such as tempo, beat times, and rhythym. A dance creation stage plans a dance sequence, choosing and varying motion parameters to reflect the music. Finally, motion synthesis translates the high-level sequence into joint motion.

A. Music Analysis

The selected beat-tracking algorithm uses the perceptual model of human hearing described in [16]. The audio is filtered into several sub-bands, then downsampled to reduce calculation overhead. The frame is augmented with approximately 3 seconds of audio history and passed through a comb filter bank (Figure 4). The tempogram depicts the amount of resonance of the audio signal corresponding to each delay. Maximum resonance occurs when the comb filter delay matches the tempo of the audio. The filter that produces the most resonance is identified and the tempo of the audio is found from the delay of that filter. The phase of the next beat is then found by finding the largest delay state in the selected filter.

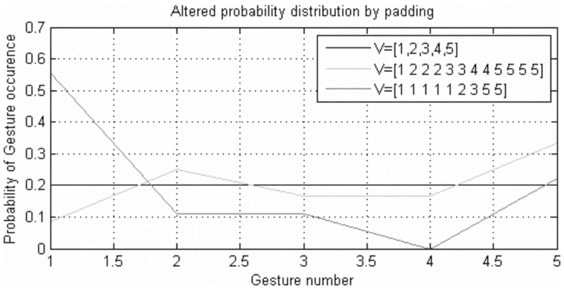

Fig. 5. Altering the probabilities of the next chosen gesture by padding the choice vector with copies of desired gestures (represented by numbers 1-5). A uniform choice from this vector gives non-uniform choices depending on this weighting

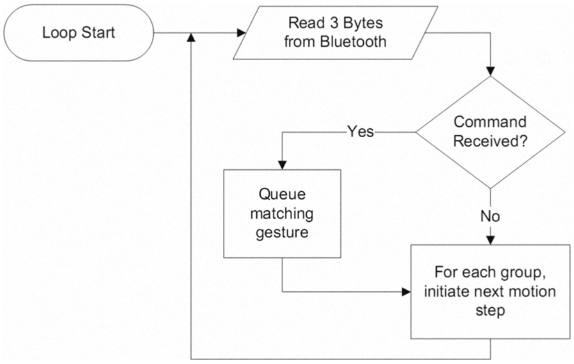

Fig. 6. Flow of Robobasic program to receive commands and output gesture commands

An initialization period of a few seconds allows the system to gather enough information to begin confidently predicting beats. After this time, the beat tracker begins analyzing incoming audio, finding and storing beats in realtime. The realtime analysis allows the gestures to be chosen as a function of incoming audio data. By selecting gestures pseudo-randomly, the chosen probability distribution controls the gesture choices. A simple implementation (Figure 5), shows the effects of varying distributions.

B. Dance Synthesis

While the beat tracker implemented in MATLAB, the motion control was programmed into the Robonova itself. The overall structure of the program is shown in figure 6. For each loop iteration, the Robonova checks the serial buffer for a command packet, then searches a tree of motion commands to execute up to 3 simultaneous gestures. Movement commands are issued independently for each group of servos, allowing gestures to overlap. The approximate time for a gesture to reach its apex was calculated for each of the 30 gestures designed. By sending commands advanced in time, the apex of the gesture and a musical beat occur simultaneously, without an on-board timer.

Preliminary results showed that the Robonova's basic language was very limiting for research. With a measured instruction rate of just 1200-1500 instructions/second, the Robonova could only process gesture commands at 2-5Hz. The loop speed varied by as much as 0.3 seconds per iteration. A timing resolution of +/0.15 seconds occasionally allows commands to arrive after the input is read. This lack of error checking was a deliberate choice because of the limited hardware speed. Because leg gestures assume the initial position is the default position, losing a return gesture makes subsequent gestures start from the wrong place, and could cause a fall.

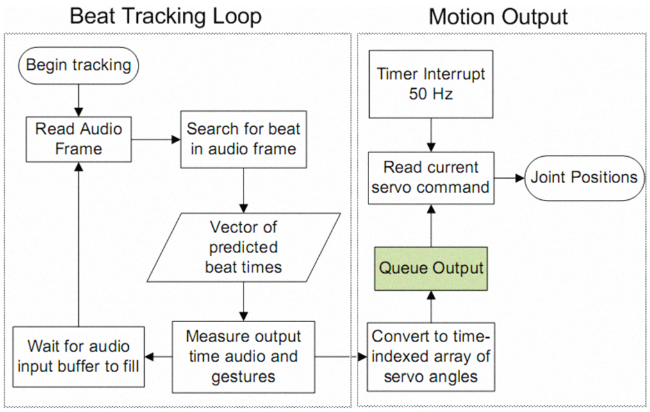

Fig. 7. Flow of modified dance program showing separation of motion planning and beat tracking loops

Clearly, the motions need to be more flexible and controllable. To address the inflexibility of Robobasic for motion planning, a modification of the Robonova's code was developed by [6]. The standard operating system was replaced on the MR-C3024 microcontroller with what is effectively a serial command parser. A command packet can be sent at 115200 baud that sets servo positions of all 16 servos. Operating at a maximum update speed of 100Hz, this offers much finer motion control than the original program. In digital mode, the servo resolution of 2000 steps, represented an eleven-fold resolution increase. To balance the processes of beat tracking and motion generation, an interrupt timer is used to send motion commands. This divides the software into distinct, semi-independent functions (Figure 7).

MATLAB's timer function implements a crude interrupt, allowing beat tracker to run freely and maximize processing. To compare timing and motion smoothness, the original gestures were converted to a vector of times and a corresponding matrix of joint angles. The times were all normalized to 1 second for ease of time scaling. Motion error was calculated as the mean-square error between the command position and actual servo position as a function of time. The command latency was calculated numerically with 1 by finding the command time offset t_lag required to minimize the error sum between the command y and the measured position $\hat{y}$, where n is the number of data points.

$$E(t_{lag}) = \frac{\Sigma(y(t) - \hat{y}(t+t_{lag})_i^2}{n} \tag{1}$$

Storing gestures as joint angles and corresponding times allows modification of the gestures that would not be possible with the old code. For instance, functions have been implemented to:

- scale gesture in time
- scale amplitude of gesture motion
- shift gesture in time
- cross-fade between gestures of overlapping times

Scaling in time affects both the continuity of the gesture and it's abruptness. Choosing a timescale smaller than the gesture spacing will make a gesture complete quickly and leave a gap in between, giving a stiff, abrupt feel to the motion. When the time spacing and scale match, then the motion of gesture sequence is continuous. The cross-fade is achieved by making the scale factor larger than the time spacing, causing the gestures to blend.

IV. APPLICATIONS TO HUBO

To produce a gesture on the Hubo, a curve through joint space must be chosen and sampled at approximately 100Hz. The joint motor controllers in Hubo take this data and interpolate at a rate of 1000Hz to produce position commands, giving approximately 20x the time resolution of a typical Robonova motion.

To prototype the dance system, the software on Hubo was modified from its original form to accept command input from a serial port. The low level software parsed the input, and executed canned motion, with simple velocity scaling to produce the appropriate timing. The MATLAB software was also modified to produce a serial command at a specified time in advance of each detected beat. This code is structurally identical to the original dance software, except that the command packet for the initial Hubo experiment was only two characters long. The first character represented the commanded gesture; the second a time-scale that sped or slowed the gesture in response to tempo. The tap gesture was produced as a metric to measure timing accuracy. To smooth acceleration, all joint displacements were specified as scaled cycloid functions similar to equation 2. The constant C controls the initial and final acceleration, such that a lower value produces a more linear displacement curve.

$$\theta_i = \frac{2\pi t - Csin(2\pi t)}{2\pi} \tag{2}$$

The resulting joint motion as seen in figure 8 is clearly smoother.

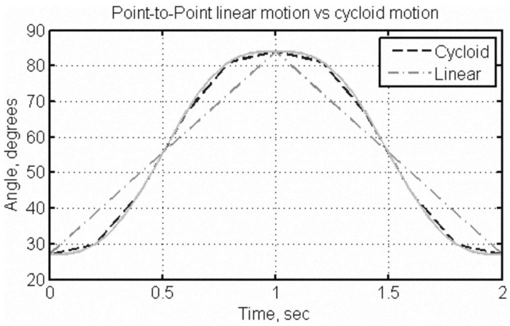

Fig. 8. Comparison of motion smoothness between original arm lift gesture (dashed) and cycloid implementation (dash-dot). Interpolation between points approximates the ideal shape (solid).

V. RESULTS

The first implementation of beat tracker used only one frequency band, which limited the genre of music that could be analyzed. A low frequency band centered at 200Hz was used to detect strong beats such as those from bass guitar and drums. Club-style dance music proved to be the most compatible, as its strong back beat and constant tempo produced the most consistent clicks. Even with very rhythmic songs, however, addition of higher frequency bands in the filter showed an improvement in the ability to track tempo (Figure 9). The song "That's All" by Genesis was scaled up in pitch by a fifth, and octave and two octaves. The frequency bands for the filters were 80Hz, 120Hz , 160Hz, 200Hz, and 240Hz. The lowest 1, 3 or 5 of these bands was used for the three plotted cases.

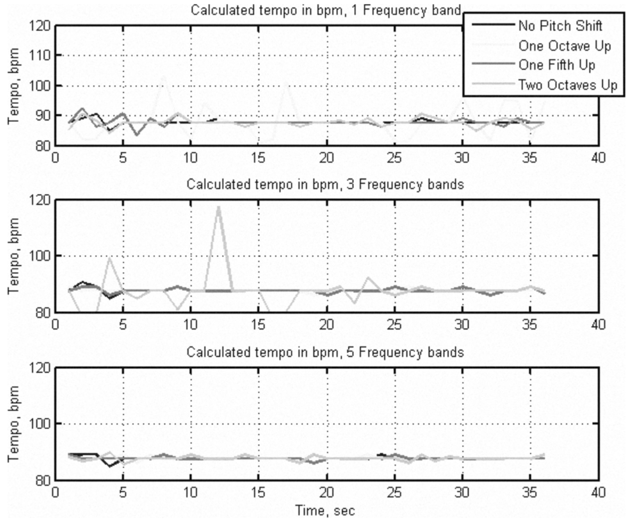

Fig. 9. Comparison of beat tracker performance

Clearly, adding more frequency bands improves beat tracking performance, especially with higher frequency content.

The maximum angle a servo can traverse during a gesture at the chosen maximum tempo of 180 bpm is given in equation 3. The constant C controls the initial and final acceleration, such that a lower value produces a more linear displacement curve.

$$\theta_{max} = \omega_{max}\Delta(t) = 4.8 * .333 \approx 91^\circ \tag{3}$$

For lightly loaded joints like those of the arms, this gives a reasonable limit for the amplitude of gesture motion in any one beat. The accuracy of position and timing was assessed numerically by direct measurement of servo position. As shown in Figure 10, the tracking error increases with motion speed and acceleration.

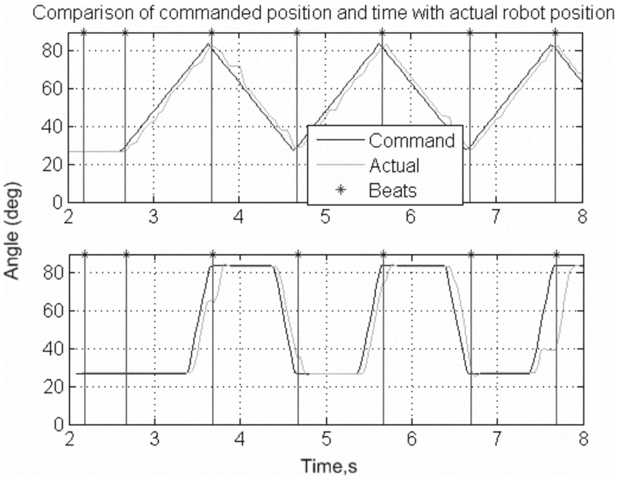

Fig. 10. Timing and position error for a simple arm-swing gesture

Latency was calculated using (1) for the same arm swing gesture. A simple linear point-to-point motion was used to minimize isolate errors due to timing. The latency stays consistent except at very high speeds, averaging 0.032s This data suggests that timing error can be adequately corrected by a fixed advance.

VI. CONCLUSIONS AND FUTURE WORK

The results of the redesign of the beat tracker and gesture command system showed that the Robonova has promise as a platform. Despite its limitations, the Robonova was nonetheless able to perform gestures that could be completely specified and altered in real-time. The simple "language developed", while simpler than Laban notation, demonstrates that robot-independent motion is possible on low-power hardware. These motion sequences are interpreted differently by Robonova and Hubo, while maintaining a similar overall appearance. This fundamental similarity allows audio analysis and dance generation to be demonstrated and debugged safely using the Robonova, then reprogrammed for the Hubo with fewer alterations to the resulting motion. Significant development remains to demonstrate portability between platforms, but these experiments demonstrate the concept is possible. Future experiments will broaden music analysis and gesture selection methods, and more fully explore the capabilities of the Hubo.

REFERENCES

[1] Honda, "Honda develops intelligence technologies enabling multiple asimo robots to work together in coordination," Press Release, December 2007.

[2] Oh, J.-H., Hanson, D., Kim, W.-S., Han, I.-Y., Kim, J., and Park, I.-W., "Design of android type humanoid robot albert hubo," in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, Oct. 2006, pp. 1428–1433.

[3] Park, I.-W., Kim, J.-Y., and Oh, J.-H., "Online biped walking pattern generation for humanoid robot khr-3(kaist humanoid robot - 3: Hubo)," in *Humanoid Robots, 2006 6th IEEE-RAS International Conference on*, Dec. 2006, pp. 398–403.

[4] Ellenberg, R., Grunberg, D., Oh, P. Y., and Kim, Y., "Exploring creativity through humanoids and dance," in *Ubiquitous Robotics and Ambient Intelligence Conference*, 2008.

[5] Maxwell, B. A., Leighton, B., and Ramsay, A., "Development of human-robot interaction systems for humanoid robots," in *5th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, November 2008.

[6] Kushleyev, A., Garber, B., and Lee, D. D., "Learning humanoid locomotion over rough terrain," in *5th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, 2008.

[7] Kankeo, K., Kanehiro, F., Kajita, S., Hirukawa, H., Kawasaki, T., Hirata, M., akachi, K., and Isozumi, T., "Humanoid robot hrp-2," in *Proceedings of the 2004 IEEE International Convention on Robotics & Automation*. New Orleans: IEEE, April 2004.

[8] Camurri, A., Hashimoto, S., Suzuki, K., and Trocca, R., "Kansei analysis of dance performance," *Systems, Man, and Cybernetics, 1999. IEEE SMC '99 Conference Proceedings. 1999 IEEE International Conference on*, vol. 4, pp. 327–332 vol.4, 1999.

[9] Huang, L. and Hudak, P., "Dance: A declarative language for the control of humanoid robots," Yale, Department of Computer Science, Yale University New Haven, CT 06520, Tech. Rep., July 2003.

[10] Nakaoka, S., Nakazawa, A., Yokoi, K., and Ikeuchi, K., "Leg motion primitives for a dancing humanoid robot," in *in Proceedings of IEEE International Conference on Robotics and Automation*, 2004, pp. 610–615.

[11] Nahrstedt, K., Bajcsy, R., Wymore, L., Sheppard, R., and Mezur, K., "Computational model of human creativity in dance choreography," in *AAAI 2008 Spring Symposium*, 2008.

[12] Safonova, A., Hodgins, J. K., and Pollard, N. S., "Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 514–521, 2004.

[13] Kosuge, K., Hayashi, T., Hirata, Y., and Tobiyama, R., "Dance partner robot - ms dancer," *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, vol. 3, pp. 3459– 3464, October 2003.

[14] Murata, K., Nakadai, K., Takeda, R., Okuno, H. G., Torii, T., Hasegawa, Y., and Tsujino, H., "A beat-tracking robot for human-robot interaction and its evaluation," *IEEE-RAS International Conference on Humanoid Robots*, pp. 79–84, 2008.

[15] Michalowski, M. P., Sabanovic, S., and Kozima., H., "A dancing robot for rhythmic social interaction," in *Proceedings of the 2nd Annual Conference on Human-Robot Interaction (HRI 2007)*, vol. 103. ACM, March 2007, pp. 89 – 96.

[16] Klapuri, A. P., Eronen, A. J., and Astola, J. T., "Analysis of the meter of acoustic musical signals," *Audio, Speech and Language Processing, IEEE Transactions on [see also Speech and Audio Processing, IEEE Transactions on]*, vol. 14, no. 1, pp. 342–355, 2006. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1561290