2011 IEEE/RSJ International Conference on
Intelligent Robots and Systems
September 25-30, 2011. San Francisco, CA, USA

Robot Audition and Beat Identification in Noisy Environments

David K. Grunberg, Daniel M. Lofaro, Paul Y. Oh, and Youngmoo E. Kim

Abstract— **In pursuit of our long-term goal of developing an interactive humanoid musician, we are developing robust methods to determine musical beat locations from live acoustic sources. A variety of beat tracking systems have been previously developed, but for the most part they are optimized for direct audio input (no acoustic channel and no noise). The presence of an acoustic channel and noise typically degrades performance substantially. A robot's motors, in particular, create non-stationary noise that can be difficult for a beat detection system to accommodate, Using an algorithm previously developed by the authors, we explore techniques for reducing the effects of the acoustic channel and noise on the system, enabling a humanoid to robustly follow music under realistic conditions.**

I. INTRODUCTION

Our long-term goal is to enable a humanoid to perform alongside humans in a musical ensemble. Achieving such a goal requires substantial gains in audio-visual understanding, fast and precise motion planning, and dexterous manipulation. To be truly interactive in terms of music performance, a robot must be able to understand and respond to live audio as its input, rather than responding to electronic control signals or choreographing motions based on pre-recorded audio. Only a robot able to process live music and determine appropriate motions in real-time would be able to handle the vagaries inherent in live musical shows without constraining the performers.

A particularly important skill for ensemble musical performance is the ability to listen to music and locate beat positions, called *auditory beat tracking*. Numerous algorithms for this skill have been developed, and some are already in common use in the field of robot audition [1], [2]. However, many of these beat trackers are optimized for audio taken directly from an audio source such as a CD. When the audio is sensed over an acoustic channel via microphones, beat tracker performance can degrade [3]. This degradation is caused by the reverberation and noise that occur in the open, real-world environments that we would like the robots to be able to perform in. The result of this degradation is that a robot using a fragile beat tracking algorithm becomes desynchronized from the music, with a corresponding decrease in its performance quality. In order to maximize the quality of the performances that occur in live environments, the beat tracking algorithm must be able to compensate for noise.

Multiple types of noise must be considered when modifying auditory algorithms to perform more accurately in live, open environments. The acoustic channel of the room and local noise sources (such as HVAC systems and computers) will produce sounds that are more or less stationary. The robot itself will also produce constant sounds (such as humming from its computers and fans), and will also generate noises as it moves its motors. These motor noises are not stationary, and in fact can vary based on the feedback of the beat tracker. For example, if a robot moves its arms in synchrony with the beat of a song, the motor noises may be approximately periodic, with the period equivalent to the tempo of the audio [4]. All of these sources of noise can contribute to the degradation of performance exhibited by audio beat trackers.

We seek to implement a robust beat tracking algorithm that can function in live environments onboard Hubo, an adult-sized humanoid robot developed by the Korean Advanced Institute of Science and Technology (KAIST) (Figure 1). Hubo is a highly capable robot with forty-two degrees of freedom and sophisticated sensors to help it to move fluidly. This robot, were it able to determine the beat positions in live audio, could then respond in synchrony with musical performances (for example, by hitting a drum or pressing a piano key). Also, by combining the auditory beat tracker with other music-analysis algorithms, Hubo's ability to react to music could be further enhanced. For example, our team has also developed a visual beat tracker that can follow a conductor's gestures. By combining both modalities for beat detection, the robot's knowledge of beat positions could be made even more robust.

II. LITERATURE REVIEW

Existing beat tracking algorithms can identify beat locations with a high level of accuracy, particularly on pieces of popular music, which generally have strong beats [5]. One common algorithm is based on the work of Scheirer [6]. In this method, audio is split into multiple subbands, and periodicity and beat locations are estimated by filtering the subband envelopes with a bank of comb filters. This method is quite accurate, particularly for music with heavy drum sections, and is the basis for several modern beat trackers [7], [8]

Another popular beat tracking algorithm is based on the work of Goto [9]. This algorithm also splits audio up into several subbands, but then performs additional processing to determine drum patterns and chord changes. This additional information is used to help identify the general metric

This work was supported by a US PIRE NSF Grant 0730206 and by a National Science Foundation Graduate Research Fellowship

David Grunberg[1], Daniel Lofaro[1], Paul Oh[2], and Youngmoo Kim[1] are with Drexel University, 3141 Market Street, Philadelphia, PA, USA. {dgrunberg,dlofaro,pyo22,ykim}@drexel.edu

[1] Affiliated with the Electrical and Computer Engineering Department

[2] Affiliated with the Mechanical Engineering Department

978-1-61284-455-8/11/$26.00 ©2011 IEEE

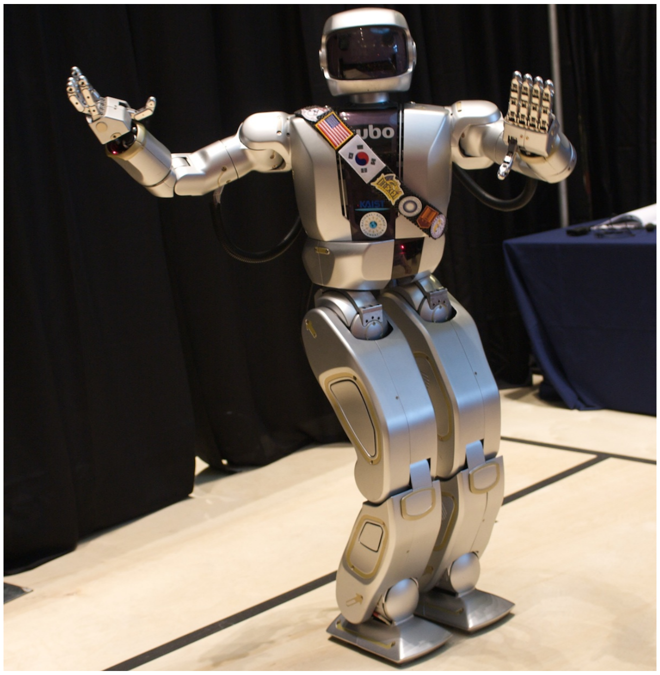

Fig. 1. Hubo, Drexel's humanoid robot

structure of the audio. Additionally, multiple beat hypotheses are considered, so that if one series of beats initially looks probable but is rendered unlikely by new information (such as beats that do not fit the previous pattern), another hypothesis can be used.

Robots that can play instruments have been researched for quite some time. In particular, research on drumming robots extends back to the late nineties [10], [11]. These robots could not detect audio, though, and were instead demonstrations of how to use oscillators and other mechanical devices to produce interesting drum patterns. A more recent development in this area is Nico [12]. This is a drummer robot which has a simple microphone setup that detects audio intensity. Sufficiently intense moments are determined to be beats, and the robot uses its knowledge of prior beat locations to perform. This system, however, does not have any noise filtering. Also, the system is only used with pure drum beats that have no accompaniment.

Kaspar, Keepon, and Robovie are three small robots that perform beat detection on real-world audio signals. Kaspar, produced by the University of Hertfordshire, has been enabled to play a drum in response to audio [13]. Keepon, developed by Carnegie Mellon University, dances in response to audio beats [14]. Neither of these robots use filtering or other noise reduction techniques. Robovie, which counts off beats as it detects them, uses a technique called semi-blind Independent Component Analysis (ICA) to help disregard the noise of its own speech [1]. This type of ICA, though, requires knowledge of the noise signal (in this case,the robot's voice), and cannot be used on unknown noises or the room reverberation.

Some robots are able to use audio beat tracking to perform in live music ensembles. One of the earliest of these robots was Haile, a robotic drummer produced by the Georgia Institute of Technology [15]. This robot listened to a human drummer and then synthesized its own beats in response. In addition to intensity, it could detect pitch and also form an estimate of how stable the beat was. While it used a real acoustic channel to detect the audio, its beat detection algorithm worked exclusively with drum signals (instead of polyphonic music). Shimon, a marimba-playing robot developed by the same group, uses the same type of beat detection algorithm but can also identify beats in both drum and keyboard signals [16].

One of the most advanced humanoid robots enabled to respond to beats is ASIMO [2]. This robot, a humanoid developed by Honda, was programmed to sing, scat, and step in response to musical beats [17]. ASIMO also uses ICA to help ignore the sound of its own voice. Unlike Robovie, this robot steps on the beats and therefore produces a large amount of noise. ASIMO therefore uses chord-change and beat-pattern detection algorithms to increase the system's robustness to noise.

There is also a subset of dancing robots that simply use direct audio, instead of audio sensed over an acoustic channel, to avoid the problem of noise. One example of this is Tai-chi, a small humanoid developed by Nirvana Technology [18]. This robot is able to dance by performing beat detection off of direct audio from CD as well as keyboard audio.

One algorithm that has proven useful for enabling robots to hear audio in the presence of noise is based on Geometric Source-Separation (GSS) [4]. For this system, given a room and the robot's location in the room, impulses are performed at various angles around the robot. The microphones record the impulses and a model of the room acoustics is developed. This model is then used in a source-separation problem to attempt to distinguish the audio from various noise sources in the room. While this system produces a high accuracy in distinguishing multiple speakers while the robot remains stationary, it requires a great deal of setup for any given room. Furthermore, because the robot will very likely move during the performance, the model could become inaccurate if the robot moves far enough away from its initial position.

III. Beat tracking

A. Beat tracker for clean audio

In our prior work, we developed a beat tracker for use on audio taken directly from CD [19]. This algorithm could function in real-time and was extremely accurate for popular music, but was untested on audio delivered via a real acoustic channel. The algorithm is based off of the ones developed by Scheirer and Klapuri [6], [8].

The frequency content of an acoustic signal is first divided into subbands with triangular bandpass filters. The subbands expand in range as they increase in frequency, to more closely simulate how the human ear hears sounds. The system then calculates the energy in each subband and stores those energies in an energy history matrix. The rows of the matrix are autocorrelated, the autocorrelations are

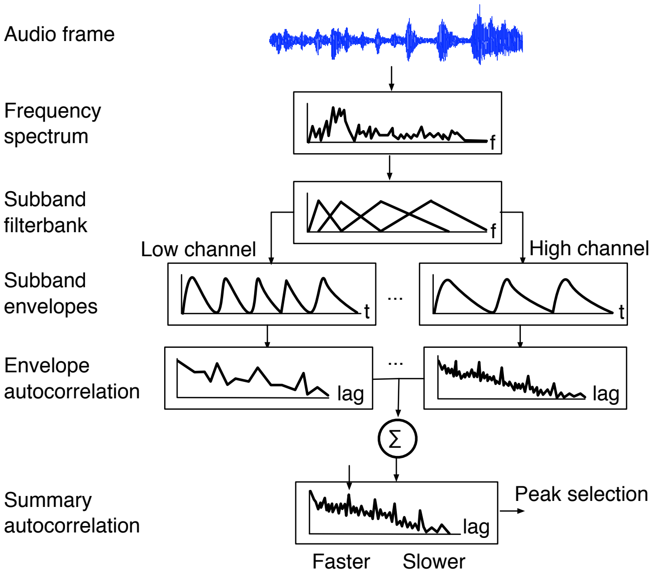

Fig. 2. Flowchart of autocorrelation-based beat detection algorithm. The arrow in the 'Summary autocorrelation' box indicates the point that will be used for estimating the tempo.

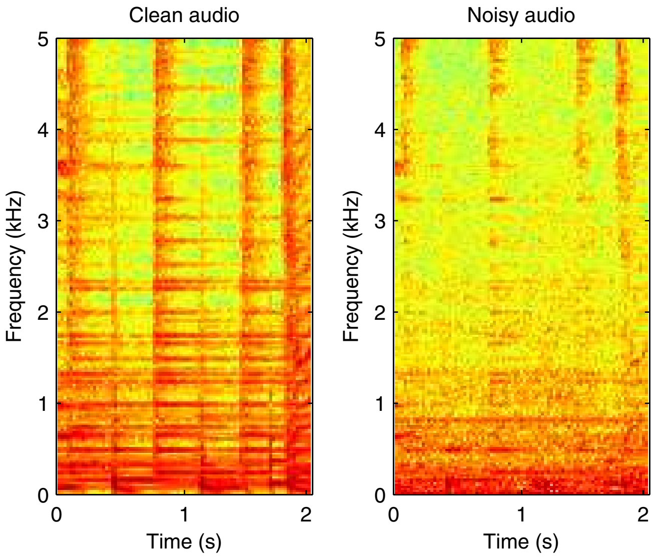

Fig. 3. An excerpt from Genesis's 'That's All.' The clean audio on the left has noticeably clearer structure than the audio on the right which has been contaminated by noise. In particular, the lowest frequencies have periods of strong and weak intensity with regularity in the clean audio, but the entire lower bands appears intense in the noisy audio.

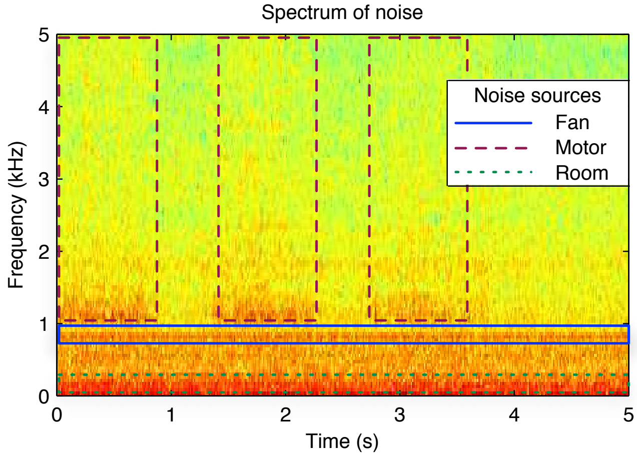

Fig. 4. Spectrogram of noise in Drexel's Autonomous Systems Laboratory. There is a narrow band at about 800 Hz that primarily consists of fan noise, bursts of primarily motor noise above 1 kHz, and strong room noise below 200 Hz.

summed, and the peak of the summary autocorrelation is found. Autocorrelations have strong responses when the lag is close to the periodicity of the signal, and audio can be treated as periodic, with the period being the tempo of the audio. Thus, the lag at the peak location can be used to calculate the tempo, or *audio period*.

The audio period and subband energies are used to determine which frames contain a beat. As the system processes a frame, it sums the subband energies to obtain the frame's total energy. It then sums this total energy with the energies from previous frames, delayed from the current frame by multiples of the audio period, to determine a multi-frame energy value. If a given frame contains a beat, not only will it likely have a high energy value, but the frames that are delayed by multiples of the audio period are also likely to have high energies. The multi-frame energy for the current frame is compared with that of all the previous frames in the audio period. If the current frame's multi-frame energy is at least a specified percentage of the maximum value, the system declares that a beat is in that frame.

B. *Noise estimation*

Different recording environments will have different amounts of stationary noise. The acoustics of a room result in resonant frequencies that interfere with acoustic musical signals. An example of the effects of this interference is shown in Figure 3. Furthermore, additional sources of noise, such as fans, can start and stop within the room. As a result, the room itself must be frequently analyzed in order to better understand, and thus compensate for, both sorts of audio distortion.

A spectrogram can be used to analyze the noise in a room over a specified time period. Figure 4 is an example of such a spectrogram, recorded in Hubo's room at Drexel by a microphone array placed in proximity to Hubo's head. During this recording, the robot was commanded to move three times, resulting in the three dense areas at approximately 1.2 kHz. The spectrogram also indicated the presence of a strong band of noise at about 800 Hz, which was later determined to be due to fans in the Hubo's body, as well as strong room reverberation below 200 Hz. By recording both when the robot is moving and when it is not, the spectrogram contains information for analyzing both stationary noise features and the motor noise of the robot.

C. *Spectral subtraction*

Spectral subtraction is a technique used to reduce or remove additive noise from a signal. It has been used in

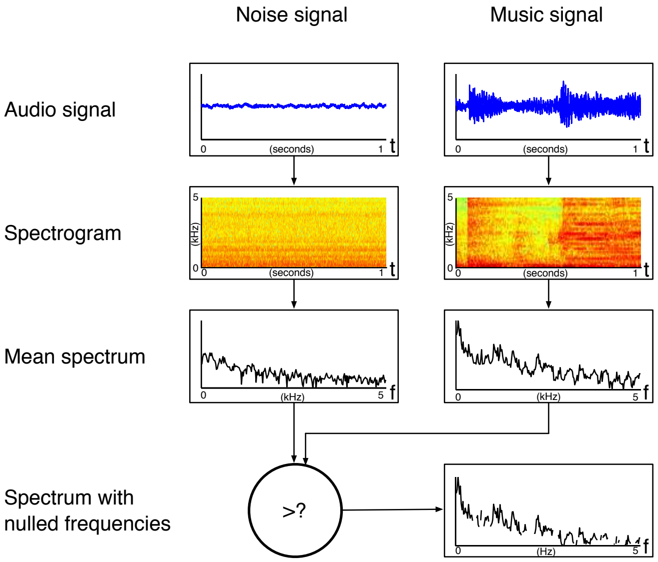

Fig. 5. Flowchart of the spectral subtraction algorithm. The spectrum of an audio signal is compared to a threshold that is based on the noise spectrum, in this case, the mean of the spectrum. This determines which frequencies are subtracted out.

speech recognition in order to improve the accuracy of speech-to-text algorithms [20]. We are exploring its use in the context of beat recognition. Its performance will indicate to us which direction we should take the beat tracker to further reduce the effect of noise on the system.

Spectral subtraction is the term for an algorithm that subtracts the contributions of various frequencies from a signal. By using an algorithm for noise estimation (such as analyzing a spectrogram of noise for strong frequencies), a system can identify which frequencies are likely to be noisy. When these frequencies are attenuated, the effects of noise on the system should be reduced, and the beat tracker should perform better.

This algorithm, however, raises two important concerns. One problem is that any useful information in the attenuated frequencies is also lost. Thus, if certain frequencies are always set to zero (as is the case with a *static filter*), information in those frequencies will be discarded even during less noisy sections of the audio. As a result, it is important to only reduce a frequency when the system is confident that the noise outweighs whatever audio information the frequency possessed. The other problem is that spectral subtraction introduces additional spurious sinusoidal distortion into the signal. [4]. It is possible that this distortion could reduce the effectiveness of the tracker, reducing or eliminating the benefits from using spectral subtraction.

In order to deal with the first problem, we designed an *adaptive filter*. For each frequency bin, the system sets a separate attenuation threshold. This threshold is based on the mean and standard deviation of the noise in that frequency bin during a segment of recorded noise (Figure 5). The system checks the amplitude at each frequency bin to determine if it is larger than the threshold at that frequency bin. If it is, then the audio is judged to be louder than the noise, and that frequency bin is not attenuated. If it is not, then the audio may be overpowered by the noise, and that frequency bin is suppressed. Whether or not the second problem had a significant effect was examined by our experiments.

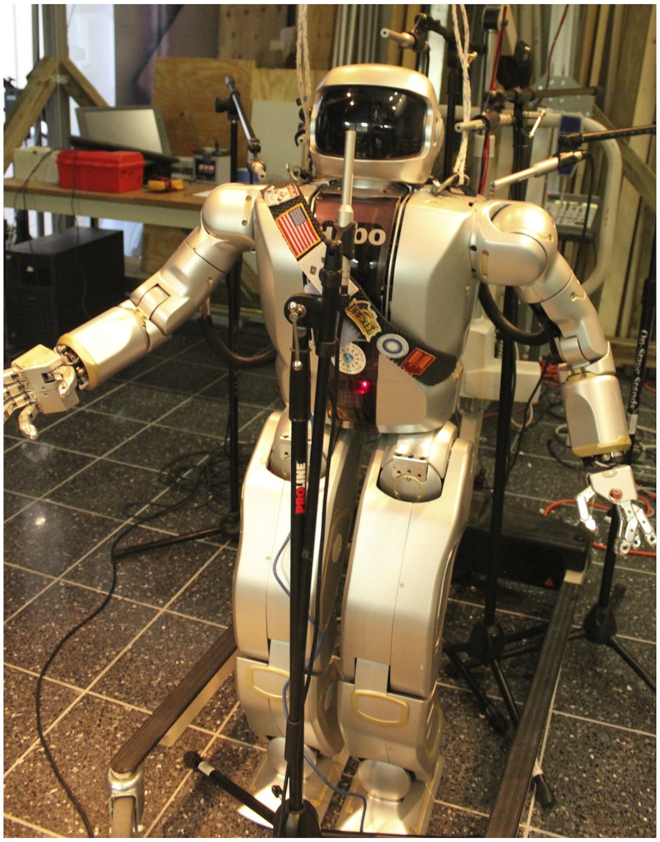

Fig. 6. The Hubo and the array of microphones

IV. Experimental procedure

The primary author listened to twenty songs in the popular music genre and marked their beat locations. The total length of these audio recordings was approximately one hour. The pieces were then played on a Roland speaker positioned ten feet away from the Hubo robot. An array of six Behringer condenser microphones was positioned around the Hubo, in the configuration shown in Figure 6. The array was used to record the audio. We programmed Hubo to move its shoulders in two different ways during the trials:

- Hubo moved its arms randomly during the test. This produced random ego noise that would then interfere with the audio signal. This represented a 'complete desynchronization' case in which the motor noise was completely uncorrelated from the audio. Such a situation occurs in the first few seconds of an audio piece, when the beat tracker has not had time to lock onto the beat positions yet, and during tempo changes in a piece, when the beat tracker has not yet switched to the new beat pattern.
- Hubo moved its arms when prompted by the beat

TABLE I

BEAT TRACKER RESULTS FOR DIFFERENT CASES, USING THE F-MEASURE METRIC.

	No. of	Direct audio	Acoustic channel, random motions			Acoustic channel, synchronized motions		
Tempo	songs	No filter	No filter	Static filter	Adaptive filter	No filter	Static filter	Adaptive filter
90-115	5	.98	.82	.92	.92	.83	.85	.94
116-126	5	.98	.90	.96	.96	.89	.88	.92
127-132	5	.98	.88	.95	.96	.95	.93	.95
133-152	5	.96	.77	.87	.95	.76	.83	.86
Total	20	.98	.84	.93	.94	.86	.87	.92

tracker, which was tracking clean audio at the same time as the microphones recorded the noisy audio. As our beat tracker performs well on clean audio, the motions were completely synchronized with the audio. This case presented additional challenges compared to the prior test, because the motor noise could become periodic with the same period as the audio signal, which could reduce the ability of the tracker to correctly identify beats.

After evaluating several different methods of setting the attenuation threshold, we experimentally determined that the best choice was the threshold shown in Equation 1.

$$T(b) = \mu_{N(b)} + .98 * \sigma_{N(b)} \tag{1}$$

where T is the threshold vector, N is the noise spectrum, b is the bin, μ is the mean of the noise energy at each bin, and σ is the standard deviation. A noise sample was recorded by the same microphone array prior to the music being played. This sample was analyzed over a five-second recording to obtain the values for μ and σ in each frequency bin. Audio was divided into .025 second frames for the main beat tracking algorithm, but for the spectral subtraction section, the previous .25s of audio was averaged together before the frequency values were compared to the threshold. This was to stop repeated 'on-off' switching of the frequencies, which could cause rapidly fluctuating frame energies and reduce the tracker's performance.

The audio was recorded on the microphone array, averaged over the microphones, and then beat-tracked with our algorithm. Tracking was performed with no filtering, with static filtering, and with adaptive filtering. These results were also compared with the case of audio taken directly from CD as a baseline.

V. RESULTS

Our beat tracking results are shown in Table I. All of the tempos are in beats per minute. All of the beat tracker accuracies were calculated according to the F-Measure metric [21]. This metric combines both precision (2) and recall (3) information to determine quantitatively the accuracy of the tracker. A high F-Measure (4) indicates that the tracker can identify most of the beat positions in a piece and does not identify many erroneous beats.

$$P = \frac{\text{correctly estimated beats}}{\text{all estimated beats}} \tag{2}$$

$$R = \frac{\text{correctly estimated beats}}{\text{all beats in song}} \tag{3}$$

$$\text{F-Measure} = \frac{2 * P * R}{P + R} \tag{4}$$

Compared to the baseline for direct audio, the tracker's performance when the robot moves its arms at random, without using any filtering, is significantly decreased. This decrease is overcome somewhat by using static spectral subtraction. However, results are most improved using adaptive spectral subtraction. In fact, the F-Measure increased to about .95, which is relatively close to the baseline score of .98. Faster pieces were particularly improved. This indicates both the utility of spectral subtraction as a noise-reduction tool, and the importance of using adaptive filtering instead of just static filtering.

Accuracy in this case is still not quite as high as in the direct case. This is likely due to a combination of the filters not eliminating 100% of the noise from the audio, the information that is lost when some frequencies are set to 0, and the spurious sinusoidal distortion created by the spectral subtraction technique. Still, the adaptive filter is still able to close the gap between the live and direct audio cases by about 77%, a considerable amount.

Similar results are seen in the case where the motions are synchronized to the music using beat tracking. The unfiltered audio case is actually slightly improved from the situation when the robot's motions are random. This may indicate that the robot is beginning to track the noise as well as the audio. Static filtering improves the accuracy slightly, but not significantly. This is only logical; since the robot's motions are periodic and predictable, it stands to reason that a constant nulling of certain frequencies would either eliminate motor noise frequencies when the motors were not moving, or would not eliminate them when the motors were moving. Much better are the adaptive filter results, which increase accuracy up to over .92. Although this score is still not as good as the direct audio score, it is much better than the unfiltered score, and indicates the potential inherent in spectral subtraction algorithms for this type of problem.

The synchronized case, when filtered, still performs more poorly than the non-synchronized case. This is likely due to the motor noise coinciding with much of the beat information, since the motions happen almost exactly on the beats. When the noise frequencies are nulled, the beat information is also lost. Future techniques could exploit knowledge of when the motors are moving in order to better account for their effects on distortion.

VI. Conclusion

Our results show that spectral subtraction is a viable technique to increase the accuracy of beat trackers in noisy environments. While there is still room for improvement, we have increased the accuracy of the real-world beat tracker to above 92% even in the completely synchronized case.

Future work will involve comparing our system to other noise-reduction algorithms. One such algorithm is HARK, produced for use with ASIMO [4]. HARK uses Geometric Source Separation to localize and then separate out the audio from different sources, and could conceivably be used to enable the robot to single out a music source and discard motor and other noises. A test against HARK could provide a good benchmark for our system.

We also plan to incorporate more information about the robot's motions into the beat tracker. Because the robot knows when it is about to move, it can communicate this information to the beat tracker. If the tracker knows when the robot is moving, and to what position, it can more accurately determine the exact effects of ego noise and can more accurately filter it out.

Lastly, we plan to use the beat tracker to advance on our goal of enabling Hubo to perform in musical ensembles. Along these lines, we plan to integrate additional music-information retrieval algorithms, such as visual beat tracking. As the robot understands music more, it will be able to react to the audio in increasingly intelligent ways. Eventually, we hope to enable the Hubo to understand music enough to fully participate in a variety of live music performances.

References

[1] Takeshi Mizumoto, Ryu Takeda, Kazuyoshi Yoshii, Kazunori Komatani, Tetsuya Ogata, and Hiroshi G Okuno, "A robot listens to music and counts its beats aloud by separating music from counting voice," in *Proceedings of the IEEE-RSJ International Conference on Intelligent Robots and Systems*, 2008.

[2] Kazuyoshi Yoshii, Kazuhiro Nakadai, Toyotaka Torii, Yuji Hasegawa, Hiroshi Tsujino, Kazunori Komatani, Tetsuya Ogata, and Hiroshi G Okuno, "A biped robot that keeps steps in time with musical beats while listening to music with its own ears," in *Proceedings of the IEEE-RSJ International Conference on Intelligent Robots and Systems*, 2007.

[3] Kazumasa Murata, Kazuhiro Nakadai, Kazuyoshi Yoshii, Ryu Takeda, Toyotaka Torii, Hiroshi G Okuno, Yuji Hasegawa, and Hiroshi Tsujino, "A robot uses its own microphone to synchronize its steps to musical beats while scatting and singing," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, September 2008.

[4] Gokhan Ince, Kazuhiro Nakadai, Tobias Rodemann, Hiroshi Tsujino, and Jun-Ichi Imura, "Robust ego noise suppression of a robot," *Lecture Notes in Computer Science*, vol. 6096, pp. 62–71, 2010.

[5] Peter Grosche, Meinard Muller, and Craig Stuart, "What makes beat tracking diffucult? a case study on chopin mazuraks," in *Proceedings of the 2010 International Society for Music Information Retrieval*, 2010.

[6] Eric D. Scheirer, "Tempo and beat analysis of acoustic musical signals," *Journal of the Acoustic Society of America*, vol. 103, no. 1, 1998.

[7] Matthew E. P. Davies and Mark D. Plumbley, "Context-dependent beat tracking of musical audio," *IEEE Transactions on Audio, Speech, and Language Processiong*, vol. 15, no. 3, 2007.

[8] Anssi P. Klapuri, Antti J. Eronen, and Jaakko T. Estola, "Analysis of the meter of acoustic musical signals," *IEEE Transactions on Speech and Audio Processing*, 2004.

[9] Masataka Goto, "An audio-based real-time beat tracking system for music with or withotu drum-sounds," *Journal of New Music Research*, vol. 30, no. 2, pp. 159–171, June 2001.

[10] A Hajian, D Sanchez, and R Howe, "Drum roll: Increasing bandwidth through passive impedance modulation," in *Proceedings of the 1997 IEEE International Conference on Robotics and Automation*, Albuquerque, April 1997.

[11] M Williamson, *Robot Arm Control Exploiting Natural Dynamics*, Ph.D. thesis, Massachusetts Institute of Technology, 1999.

[12] Christopher Crick, Matthew Munz, Tomislav Nad, and Brian Scassellati, "Robotic drumming: synchronization in social tasks," in *Proceedings of the IEEE International Workshop on Robot and Human Interactive Communication*, 2005.

[13] Hatice Kose-Bagci, Kerstin Dautenhahn, Dag Sverre Syrdal, and Chrystopher L. Nehaniv, "Drum-mate: A human-humanoid drumming experience," in *Proceedings of the IEEE-RAS International Conference on Humanoid Robots*, 2007.

[14] Marek P. Michalowski, Reid Simmons, and Hideki Kozima, "Rhythmic attention in child-robot dance play," in *Proceedings of the 18th IEEE International Symposium on Robot and Human Interactive Communication*, 2009.

[15] Gil Weinberg and Scott Driscoll, "Robot-human interaction with an anthropomorphic percussionist," in *Proceedings of the Conference on Human Factors in Computer Systems*, 2006.

[16] Gil Weinberg, Aparna Raman, and Trishul Mallikarjuna, "Interactive jamming with shimon: A social robotic musician," in *Proceedings of the 4th ACM/IEEE International Conference on Human Robot Interaction*, 2009.

[17] Kazumasa Murata, Kazuhiro Nakadai, Ryu Takeda, Hiroshi G Okuno, Toyotaka Torii, Yuji Hasegawa, and Hiroshi Tsujino, "A beat-tracking robot for human-robot interaction and its evaluation," in *Proceedings of the IEEE-RAS International Conference on Humanoid Robots*, 2008.

[18] Naoto Nakahara, Koji Miyazaki, Hajime Sakamoto, Takashi X. Fujisawa, Noriko Nagata, and Ryohei Nakatsu, "Dance motion control of a humanoid robot based on real-time tempo tracking from musical audio signals," in *Proceedings of the 8th International Conference on Entertainment Computing*, 2009.

[19] David Grunberg, Robert Ellenberg, In Hyeuk Kim, Jun Ho Oh, Paul Y. Oh, and Youngmoo E. Kim, "Development of an autonomous dancing robot," *International Journal of Hybrid Information Technology*, vol. 3, no. 2, pp. 33–44, April 2010.

[20] Sunil D. Kamath and Philipos C. Loizou, "A multiband spectral subtraction method for enhancing speech corrupted by colored noise," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2002.

[21] Matthew E. P. Davies, Norburto Degara, and Mark D. Plumbley, "Evaluation methods for musical audio beat tracking algorithms," Tech. Rep. C4DM-09-06, Queen Mary University of London, October 2009.