2008 IEEE/RSJ International Conference on Intelligent Robots and Systems
Acropolis Convention Center
Nice, France, Sept, 22-26, 2008

A Robot Uses Its Own Microphone to Synchronize Its Steps to Musical Beats While Scatting and Singing

Kazumasa Murata, Kazuhiro Nakadai, Kazuyoshi Yoshii, Ryu Takeda, Toyotaka Torii, Hiroshi G. Okuno, Yuji Hasegawa, and Hiroshi Tsujino

Abstract— **Musical beat tracking is one of the effective technologies for human-robot interaction such as musical sessions. Since such interaction should be performed in various environments in a natural way, musical beat tracking for a robot should cope with noise sources such as environmental noise, its own motor noises, and self voices, by using its own microphone. This paper addresses a musical beat tracking robot which can step, scat and sing according to musical beats by using its own microphone. To realize such a robot, we propose a robust beat tracking method by introducing two key techniques, that is, spectro-temporal pattern matching and echo cancellation. The former realizes robust tempo estimation with a shorter window length, thus, it can quickly adapt to tempo changes. The latter is effective to cancel self noises such as stepping, scatting, and singing. We implemented the proposed beat tracking method for Honda ASIMO. Experimental results showed ten times faster adaptation to tempo changes and high robustness in beat tracking for stepping, scatting and singing noises. We also demonstrated the robot times its steps while scatting or singing to musical beats.**

I. INTRODUCTION

Humanoid robots have been studied for many years. Thanks to recent progress in controlling a humanoid, we can see several robots which achieved walking, running, and whole body motions. Human-robot interaction is another challenging research topic for humanoids. To realize richer and more natural human-robot interaction, artificial listening capabilities and speech dialog systems for a robot have been studied. The research area to realize such an artificial listening capability is called "robot audition"[1], and some robot audition systems which achieved highly noise-robust recognition have been reported so far [2], [3]. Several dialog systems for a robot also have been reported [4], [5], [6], [7]. However, they focused on only speech signals for their human-robot interaction.

On the other hand, human-robot interaction using other sounds like music draws attention of robotics researchers. Sony exhibited a singing and dancing robot called QRIO. Kosuge *et al.* showed that a robot dancer, MS DanceR, performed social dances with a human partner [8]. Nakazawa *et al.* reported that HRP-2 imitated the spatial trajectories of complex motions of a Japanese traditional folk dance by using a motion capture system [9]. Although these robots performed dances and/or singing according to music, most of them were programmed in advance, and did not react to musical audio signals. Some robots have music listening functions. Kotosaka and Schaal [10] developed a robot that plays drum sessions with a human drummer. Michalowski *et al.* developed a small robot called Keepon which can move its body quickly according to musical beats [11]. These robots work well when a musical audio signal is given without noises. However, it is difficult for them to cope with noises such as environmental noises, self voices, and so on. Thus, they have difficulties in singing and stepping.

To realize a robot which interacts more naturally with humans through music in a real environment, we developed a prototype of a beat tracking robot using Honda ASIMO [12]. This robot was able to detect a tempo and a beat time of music by using a real-time beat tracking algorithm proposed by Goto *et al.* [13], and the robot that times its steps to the detected musical beats was demonstrated. However, the robot mainly had two issues as follows:

1) The robot adapted slowly to a beat when music started, or a tempo was changed.
2) The robot performed only stepping as a result.

The first issue was caused by the beat tracking algorithm. The robot took around ten seconds to adapt to beat change on average. Actually, the beat tracking algorithm was able to cope with real musical signals even when vocal parts are included. However, instead of improving the robustness, sensitivity was sacrificed. The second issue is not a simple problem, that is, it is not so easy to add a new behavior. When we implement the new behavior and synchronize it to musical beats, it makes noises. In addition, the noises are periodic because they are generated according to "periodic" beat signals. If the noises and the beats are properly synchronized, there will be no problem. However, it takes a while for them to be fully synchronized. Thus, the noises affect the performance of beat tracking badly. When sound generation functions such as humming, scatting or singing are implemented as such a new behavior, this problem is much harder to be solved, because the loudspeaker embedded in a robot is usually closer to a robot-embedded microphone than a music source. Such a noise generated by additional

K. Murata is with Graduate School of Information Science and Engineering, Tokyo Institute of Technology, Tokyo, 152-8552, Japan murata@cyb.mei.titech.ac.jp

K. Nakadai is with Honda Research Insititute Japan Co., Ltd., 8-1 Honcho, Wako, Saitama 351-0114, JAPAN, and also with Graduate School of Information Science and Engineering, Tokyo Institute of Technology nakadai@jp.honda-ri.com

K. Yoshii, R. Takeda, and H. G. Okuno are with Graduate School of Informatics, Kyoto University, Kyoto, 606-8501, Japan {yoshii,rtakeda,okuno}@kuis.kyoto-u.ac.jp

T. Torii, Y. Hasegawa, and H. Tsujino are with the Honda Research Institute Japan Co., Ltd. {tory, yuji.hasegawa, tsujino}@jp.honda-ri.com

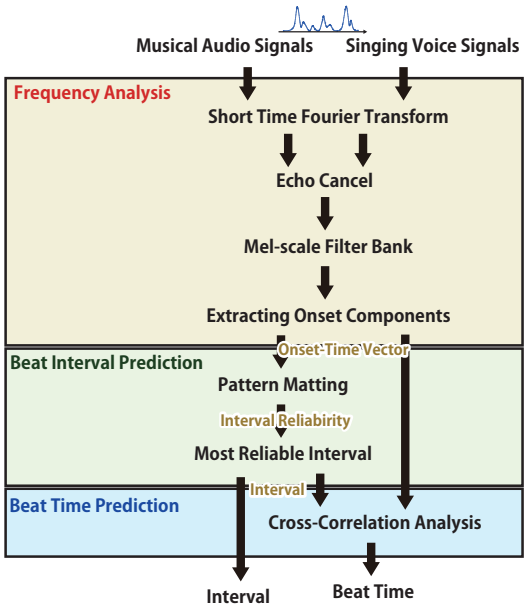

Fig. 1. Overview of a new real-time beat-tracking method

behaviors should be suppressed.

In this paper, we propose two methods to solve the above issues. One is a newly-developed beat tracking algorithm based on spectro-temporal pattern matching to realize faster adaptation to tempo changes. The other is the introduction of noise cancellation based on semi-blind Independent Component Analysis (semi-blind ICA) proposed by Takeda *et al.*[14]. We developed a new beat tracking robot with these methods by using ASIMO so that it can step, scat and sing according to musical beats. We evaluated the performance of the robot in terms of adaptation speed, and noise-robustness.

The rest of this paper is organized as follows: Section II describes a new beat tracking algorithm. Section III explain newly added behaviors such as scatting and singing. Section IV implements our beat tracking robot by using the proposed methods. Section V conducts some experiments to confirm the effectiveness of our beat tracking robot. The last section concludes this paper.

II. New Beat Tracking Algorithm

A lot of beat-tracking methods have been studied in the field of music information processing [15]. They focus on extraction of complicated beat structures with off-line processing, although there are some exceptions like [16], [17]. Nakadai *et al.* reported the importance of auditory processing by using robots' own ears. They proposed "robot audition" as a new research area[1]. Some robot audition systems which achieved highly noise-robust speech recognition have been reported [2], [3]. However, beat tracking for noisy signals such as robot-noise-contaminated music signals has not been studied so far. For example, a real-time beat tracking method which was used in our reported system consists of three kinds of detection modules – onset detection, tempo detection, and beat time detection[16]. The onset detection module extracts a 7-dimensional onset time vector for each time frame in the frequency domain. Each value in the onset time vector is a belief factor on onset existence at the corresponding frequency band. Thus, the onset time is strongly related to beat time. The tempo detection module extracts a tempo defined as the average beat interval by using a self-correlation function. The beat time detection module predicts the next beat time by using the detected tempo and onset time vectors. This procedure is performed by using 12 agents with different detection parameter values, that is, this algorithm is based on multi-agent architecture. Finally, 12 results are integrated, and the most reliable result is outputted as the predicted beat time. This algorithm works in real time, and it shows high robustness for noises. Thus, this algorithm often has been used for real-world applications.

However, the algorithm has the following three problems.

- slow adaptation to tempo changes,
- a large amount of calculation,
- a large number of parameters which are difficult to be tuned manually.

The first problem is caused by long window length for a self correlation function. When the window length of a self correlation function is long enough, it shows high noise-robustness because of the effect of temporal integration. However, it becomes insensitive to tempo changes. In this algorithm, 12 agents are processed in parallel. This means that it requires high computational costs. Thus, to maintain real-time processing, some frames are ignored. This leads to the second problem. The last problem comes from multi-agent architecture. Indeed, multi-agent architecture is effective to improve noise-robust, but it is sometimes difficult to control these agents and their integration. Thus, it uses many parameters which are difficult to be optimized.

We propose a new beat tracking algorithm which is free from these problems by taking the following three strategies:

- *Strategy 1 :* fast adaptation to tempo changes by using spectro-temporal pattern matching,
- *Strategy 2 :* reduction of computational costs by using mel frequency,
- *Strategy 3 :* realization of easier control of stability and adaptation speed by not using any multi-agent architecture.

Figure 1 shows an overview of the new beat tracking algorithm based on the above approaches. The new algorithm also outputs the predicted beat time, and a tempo value. It consists of three parts – frequency analysis, beat interval prediction and beat time prediction.

A. Frequency Analysis

This system has two input signals. One is a musical audio signal recorded with a robot-embedded microphone. The recorded signal is usually contaminated by noise sources such as self-noises. The other input is a self-noise signal such as a scatting or a singing voice. Because the self-noise is known in advance for the system, pure self-noise can be directly obtained from line-in without using a microphone.

In the frequency analysis stage, spectra are consecutively obtained by applying the short time Fourier transform

(STFT) to two input signals sampled at 44.1 kHz. The Hanning window of 4096 points is used as a window function, and its shift length is 512 points.

Echo canceling is, then, applied. It is essential to eliminate self-noises such as singing and scatting voices to improve beat tracking. We introduced semi-blind ICA for echo cancellation[14], because this method shows good performance and is easily extended to a multi-channel input. We used a two-channel version of semi-blind ICA. One channel takes the musical audio spectra contaminated by self-noises as an input, and the other channel takes a pure self-noise as an input. The noise-suppressed musical audio spectra are sent to Mel-scale Filter Bank.

It reduces the number of frequency bins from 2,049 linear-scale frequency bins to 64 mel-scale frequency bins. Mel-scale frequency is decided by taking human ear's perceptual characteristics into account, and it is often used for automatic speech recognition and music recognition.

A frequency bin where a spectral power increases rapidly is detected as an onset candidate at the mel-scale frequency domain. We used the Sobel filter to detect frequency bins only with rapid power increase. Let $p_{mel}(t,f)$ be the spectral power at the t-th time frame and the f-th mel-filter bank bin. An onset belief $d_{inc}(t,f)$ is estimated by

$$d_{inc}(t,f) = \begin{cases} d(t,f) & \text{if } \mathrm{d(t,f)} > 0, \\ 0 & \text{otherwise,} \end{cases} \quad (1)$$

$$\begin{aligned} d(t,f) = & -p_{mel}(t-1,f+1)+p_{mel}(t+1,f+1) \\ & -p_{mel}(t-1,f-1)+p_{mel}(t+1,f-1) \\ & -2p_{mel}(t-1,f)+2p_{mel}(t+1,f), \end{aligned} \quad (2)$$

where $f = 1,2,...,62$. Eq. (2) shows the Sobel filter. Thus, a 62-dimensional onset time vector is extracted for each time frame.

B. Beat Interval Prediction

To estimate a beat interval which is defined as the temporal difference between two neighboring beats, spectro-temporal pattern matching is performed by using the onset time vector. As a pattern matching function, we used Normalized Cross-Correlation (NCC) defined by

$$\begin{aligned} R(t,i) &= \frac{A(t,i)}{\sqrt{B(t)C(t,i)}}, \quad (3) \\ A(t,i) &= \sum_{f=1}^{62}\sum_{k=0}^{P_{width}-1} d_{inc}(t-k,f)d_{inc}(t-i-k,f), \\ B(t) &= \sum_{f=1}^{62}\sum_{k=0}^{P_{width}-1} d_{inc}(t-k,f)^2, \\ C(t,i) &= \sum_{f=1}^{62}\sum_{k=0}^{P_{width}-1} d_{inc}(t-i-k,f)^2, \end{aligned}$$

where P_{width} is window length for pattern matching, and i is the shift parameter (Fig. 2).

In the Goto's beat tracking algorithm, instead of using spectro-temporal pattern matching, a frequency-line-based self-correlation function is used. Because it did not use

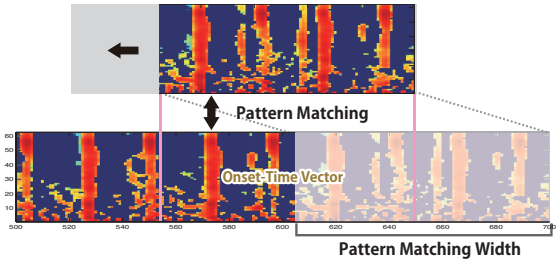

Fig. 2. Spectro-Temporal Pattern Matching

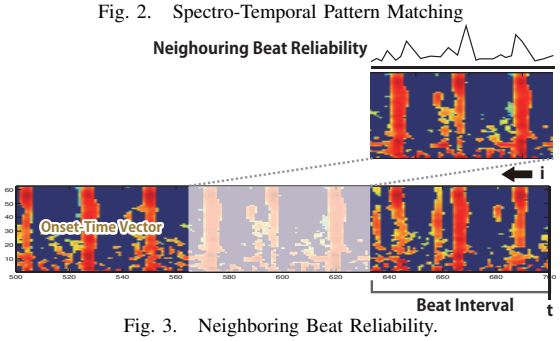

Fig. 3. Neighboring Beat Reliability.

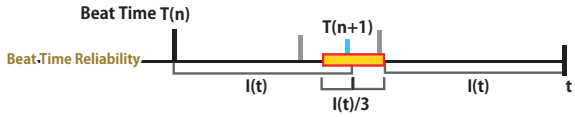

Fig. 4. Beat Time Detection

spectral integration, a longer window length (about 6 – 11 sec) of the self-correlation function was necessary to improve robustness. This leads to insensitivity to tempo changes. The proposed spectro-temporal pattern matching uses both temporal and spectral integration. Thus, only a short window length of 1 sec is necessary while keeping noise-robustness. This makes faster adaptation to tempo changes.

A set of local peaks is, then, extracted from $R(t,i)$

$$R_p(t,i) = \begin{cases} R(t,i) & \text{if } \mathrm{R(t,i-1)} < \mathrm{R(t,i)} < \mathrm{R(t,i+1)}, \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

When two peaks have comparable reliabilities, mis-detection of beat interval occurs. To avoid this mis-detection, a beat interval is limited from 61 to 120 M.M.[1] When beat intervals for the first and the second biggest local peaks in $R_p(t,i)$ are I_1 and I_2, beat interval at time t is estimated by

$$\begin{aligned} I(t) &= \begin{cases} 2|I_1-I_2| & (|I_{n2}-I_1|<\delta \text{ or } |\mathrm{I_{n2}-I_2}|<\delta), \\ 3|I_1-I_2| & (|I_{n3}-I_1|<\delta \text{ or } |\mathrm{I_{n3}-I_2}|<\delta), \\ I_1 & \text{otherwise.} \end{cases} \quad (5) \\ I_{n2} &= 2|I_1-I_2|, \\ I_{n3} &= 3|I_1-I_2|, \end{aligned}$$

where δ means an error margin parameter. This formulation are defined empirically to avoid mis-estimation such as double and triple tempos.

C. Beat Time Prediction Stage

In this stage, beat reliability is estimated from two types of reliabilities – neighboring beat reliability and continuous beat

[1]Mälzel's Metronome: the number of quarter notes per minute. For example, if the tempo is 60 M.M., the quarter-note length is 1,000 [ms].

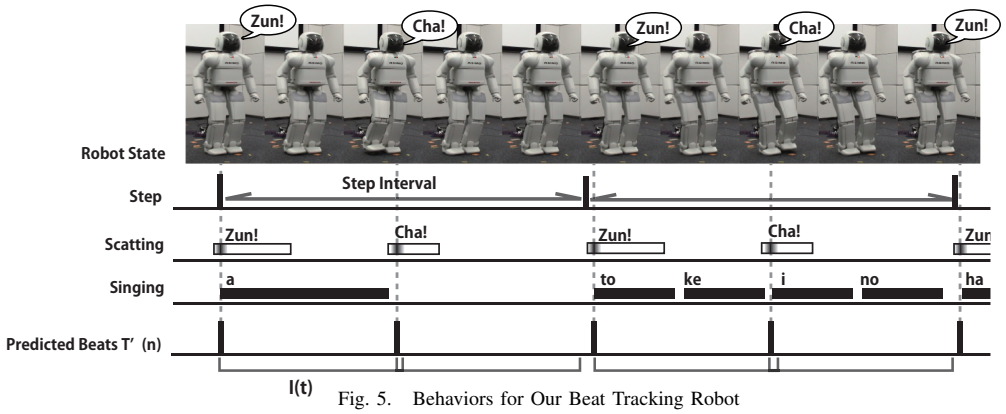

Fig. 5. Behaviors for Our Beat Tracking Robot

reliability. Beat time is predicted according to beat reliability.

1) Neighboring beat reliability: By using the beat interval, a reliability on beat existence, that is, neighboring beat reliability is calculated at the current time and at the previous beat time shown in Fig. 3.

$$S_c(t,i) = \begin{cases} F_s(t-i)+F_s(t-i-I(t)) & (i \le I(t)), \\ 0 & (i > I(t)), \end{cases} \quad (6)$$

$$F_s(t) = \sum_{f=1}^{62} d_{inc}(t,f),$$

where $S_c(t,i)$ denotes a neighboring beat reliability for time $t-i$ at time t.

2) Continuous beat reliability: From neighboring beat reliabilities, a reliability of a temporal beat sequence, that is, continuous beat reliability is calculated.

$$S_r(t,i) = \sum_{m}^{N_{S_r}} S_c(T_p(t,m),i), \quad (7)$$

$$T_p(t,m) = \begin{cases} t-I(t) & (m=0), \\ T_p(t,m-1)-I(T_p(t,m)) & (m \ge 1), \end{cases}$$

where $S_r(t,i)$ denotes continuous beat reliability for time $t-i$ at time t. $T_p(t,m)$ means the m-th previous beat time for time t, and N_{S_r} is the number of beats to calculate continuous beat reliability. This reliability is effective to decide the best beat sequence such as strong beats when multiple beat sequences are detected.

3) Beat reliability: The neighboring beat reliability and the continuous beat reliability are integrated into a beat reliability defined by

$$S(t) = \sum_i (S_c(t-i,i)S_r(t-i,i)). \quad (8)$$

4) Beat time detection: Let the n-th beat time be $T(n)$. When $T(n) \ge t - \frac{3}{4}I(t)$, three-best peaks in $S(t)$ are extracted from $T(n)+\frac{1}{2}I(t)$ to $T(n)+\frac{3}{2}I(t)$. The peak which is closest to $T(n)+I(t)$ is estimated as the next beat time $T(n+1)$ shown in Fig. 4. In case no peak is found from $T(n)+\frac{2}{3}I(t)$ to $T(n)+\frac{4}{3}I(t)$, $T(n)+I(t)$ is regarded as $T(n+1)$.

5) Prediction of future beat time: The beat time $T(n+1)$ detected in beat time detection is a past beat, that is, $t > T(n+1)$. To apply beat tracking to scatting or singing, a future beat time T' should be predicted. By using the following extrapolation, a future beat time is predicted.

$$T' = \begin{cases} T_{\mathrm{tmp}} & \text{if } \mathrm{T_{tmp}} \ge \frac{3}{2}\mathrm{I_m(t)} + \mathrm{t}, \\ T_{\mathrm{tmp}} + I_m(t) & \text{otherwise}, \end{cases} \quad (9)$$

$$T_{\mathrm{tmp}} = T(m) + I_m(t) + (t - T(m)) - \{(t - T(m)) \mod I_m(t)\},$$

where $I_m(t)$ is a median value of a set of $I(t)$, and $T(m)$ is the latest beat time detected in beat time detection.

III. Behaviors for Our Beat Tracking Robot

This section explains robot behaviors synchronized with musical beats. Fig. 5 shows an example of our beat tracking robot with such behaviors.

A. Stepping

We reported the stepping function in [12]. This function is to adjust step timings to musical beats only by using a command of specifying a step interval. Because an accurate target value is unavailable, it is theoretically difficult to control a robot even when a model-based feedback control is used in this case. Thus, we used a simple feedback control to reduce the errors of step timing and intervals.

B. Scatting

Scatting means, here, that a robot outputs sounds according to a beat pattern. In this paper, we assume that strong and weak beats appear in music alternately. For scatting, "zun" was outputted for a strong beat, and "cha" for a weak beat. Since these words have some durations, synchronization between these words and beat time includes some ambiguities. When their correspondence is slightly changed, people easily feel that it is unnatural or the robot is tone deaf. As beat times for these words, we empirically decided to use the onset times of their audio signals. The onset times are detected by using the algorithm for the onset detection module.

C. Singing

Singing means that a robot sings according to musical beats. Thus, when a music tempo decreases, the robot can sing slowly. As prior information, the melody and lyrics of the music are given to the system as MIDI data. VOCALOID developed by YAMAHA is used as a singing engine. It

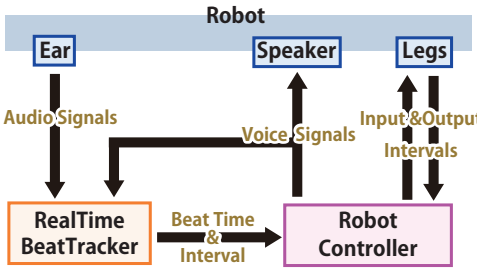

Fig. 6. Architecture of a beat tracking robot

achieves a singing function with around 200 ms latency. The robot outputs singing voices synchronizing them to musical beats by taking such latency into account.

IV. Implementation of Our Beat Tracking Robot

Fig. 6 shows the architecture of our beat tracking robot. The system mainly consists of three parts – Humanoid robot, Realtime Beat tracker and Robot Controller.

We used Honda ASIMO with a microphone embedded in the head as a testbed. It has two legs like humans and can stamp its feet on the floor, i.e., perform steps in a stationary location. The step interval is limited to between 1 and 2 seconds due to our robot's mechanistic limitations. If the tempos of musical pieces are between 61 and 120 M.M., The robot records these signals with its own single microphone embedded in the front of the head. It has a loudspeaker at the position of its chest. In terms of implementation, Real-time Beat Tracker was implemented by C++ on Mac OSX. This part works in real time on a remote PC with Core 2 Duo. In Robot Controller, scatting and stepping are running the same PC as the above component, which only singing function is running on Windows PC.

A Real Time Beat Tracker output predicted beat time and a beat interval from a signal captured by a robot's microphone. Synthesized singing or scatting voices are inputted from line-in to be used for noise suppression. The detailed algorithm is described in Section II.

A Robot Controller controls ASIMO to time its steps to musical beats, and to synchronize them with singing or scatting. The voices are outputted from a loudspeaker inside ASIMO. The control of stepping is done by using a command via a TCP/IP network.

V. Evaluation

We evaluate our beat tracking robot in terms of the following two points:

- adaptation speed to tempo changes, and
- noise-robustness using a beat prediction success rate.

Two kinds of musical signals were used for these experiments.

T1 *musical signal including tempo changes*
We prepared a 4-minute musical signal by selecting three songs (No. 11, No. 18, and No. 62) from the RWC music database (RWC-MDB-P-2001) developed by Goto *et al.* [18]. They include vocals and various instruments as commercial CDs do. Their tempos were 90, 112, and 81 M.M, respectively. We concatenated four 60-s segments that were extracted from the four pieces.

T2 *musical signal with fixed tempo*
We synthesized a musical signal of No. 62 by using MIDI data. MIDI data provides reference data of beat times. MIDI data is not used as a prior information for tempo and beat time detection.

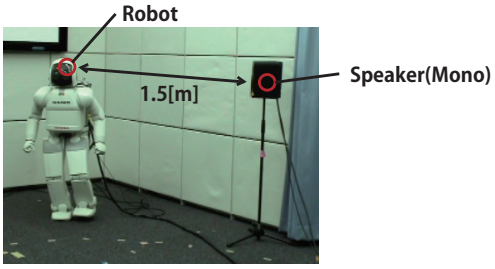

Fig. 7. Overview of experimental condition: The system concerning to the robot is completely separated form that concerning to the music playback.

In every experiment, a loudspeaker was set in a 4 m×7 m room with 0.2 seconds of reverberation time (RT_{20}). The distance between the robot and the speaker was 1.5 m. The musical signals were played from the loudspeaker. This situation is outlined in Fig. 7. For the first experiment, we used **T1**, and beat tracking delay was measured in five conditions. The beat tracking delay was defined as the time difference between when an actual tempo was changed and when the system adapted to the tempo change. Two conditions of the five were the ones with and without scatting when ASIMO was turned off. The other three conditions were the ones without scatting, with scatting and with singing when ASIMO was turned on and performed stepping. For the second experiment, we used **T2**, and the beat prediction success rate was measured in five conditions. The beat prediction success rate r is defined by

$$r = \frac{N_{\text{success}}}{N_{\text{total}}} \times 100. \tag{10}$$

where N_{success} is the number of successfully predicted beats, and N_{total} is the number of total beats. When the error of a predicted beat time is within $\pm 0.35I(t)$ as described in [13], it is regarded as successfully predicted. Three conditions of the five are the ones when ASIMO was turned off. One was the condition without scatting and with echo canceling. Another two were the ones with and without canceling while scatting. The other two conditions of the five are the ones with and without echo canceling when ASIMO was turned on with stepping while scatting.

A. Results

Fig. 8 shows an example of beat tracking in the first experiment. The proposed method adapted to tempo changes more quickly than the previously reported method. Disordered tracking results were found at around 100 seconds because **T1** had missing beats. A longer window length like the reported method is necessary to cope with this situation. Statistical data is shown in Table I. This shows that our proposed method adapted to the tempo changes 20 times

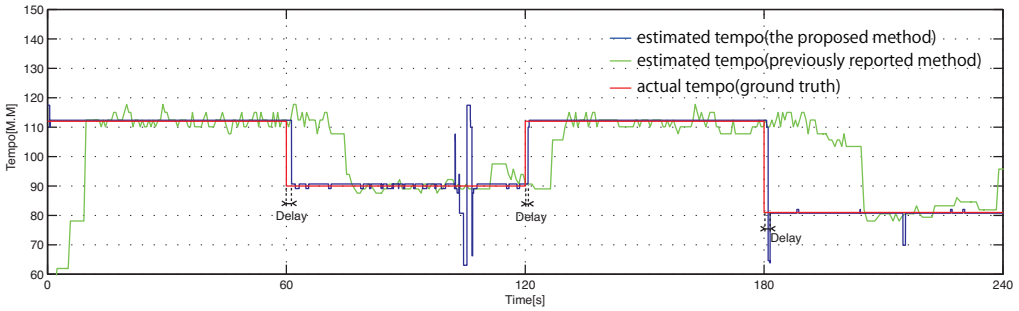

Fig. 8. Result of using music of tempo chage.

TABLE I

TRACKING DELAY FOR TEMPO CHANGES

	ASIMO power off		ASIMO power on(with step)		
	w/o scatting	w/ scatting	w/o scatting	w/ scatting	w/ singing
reported	11.24	29.91	14.66	20.43	N/A
proposed	1.31	1.31	1.29	1.29	1.29

TABLE II

BEAT PREDICTION SUCCESS RATE

	ASIMO power off			ASIMO power on (with step)	
	w/o scatting	w/ scatting		w/ scatting	
		w/ echo cancel	w/o echo cancel	w/ echo cancel	w/o echo cancel
Correct	95%	97%	68%	95%	64%
Half shifted	5%	1%	40%	4%	40%

faster than the reported one when no voice exists, and it is still 10 times faster than when scatting voices exist. The reported system failed in beat tracking when singing voices existed, while the proposed was still robust.

Table II shows the results of the second experiment. "Correct" means the beat tracking system correctly predicted beats, that is, strong beats. "Half-shifted" means that a weak beat is mis-predicted as a strong beat. This shows self-noises affected beat tracking due to its periodicity, and echo cancel drastically reduced the effect of such self-noises.

As another result, we confirmed that the robot times its steps to musical beats while scatting or singing.

VI. CONCLUSIONS

We developed a robust and quickly-adaptable musical beat tracking method for human-robot interaction. The method uses spectro-temporal pattern matching to improve the adaptation speed against tempo changes, and echo canceling based on semi-blind independent component analysis to suppress self periodic noises such as scatting and singing. Performance evaluation using Honda ASIMO with the proposed beat tracking method showed high noise-robustness and quick adaptation to tempo changes. We believe that the proposed method and its extension will help to realize more natural and richer human-robot interaction. More sophisticated robot motions such as dancing, a human-robot musical session, and improvements of robustness of beat tracking by using other modalities such as vision are remaining future work.

ACKNOWLEDGMENTS

This research was partially supported by JSPS, Grant-in-Aid for Young Scientists (B) 19700158.

REFERENCES

[1] K. Nakadai *et al.*, "Active audition for humanoid," in *Proc. of 17th National Conference on Artificial Intelligence (AAAI-2000)*. AAAI, 2000, pp. 832–839.

[2] I. Hara *et al.*, "Robust speech interface based on audio and video information fusion for humanoid HRP-2," *IROS 2004*. IEEE/RSJ, 2004, pp. 2404–2410.

[3] S. Yamamoto *et al.*, "Real-time robot audition system that recognizes simultaneous speech in the real world," *IROS 2006*. IEEE/RSJ, 2006, pp. 5333–5338.

[4] H. Asoh *et al.*, "Socially embedded learning of the office-conversant mobile robot *jijo-2*," in *Proc. of 15th Int'l Joint Conf. on Artificial Intelligence (IJCAI-97)*, vol. 1. AAAI, 1997, pp. 880–885.

[5] Y. Matsusaka *et al.*, "Multi-person conversation via multi-modal interface — a robot who communicates with multi-user," in *Proc. of Eurospeech-1999*, 1999, pp. 1723–1726.

[6] M. Nakano *et al.*, "A two-layer model for behavior and dialogue planning in conv ersational service robots," *IROS-2005*, IEEE/RSJ, 2005, pp. 1542–1547.

[7] N. Mavridis and D. Roy, "Grounded situation models for robots: Where words and percepts meet," *IROS 2006*. IEEE/RSJ, 2006, pp. 4690–4697.

[8] T. Takeda *et al.*, "Hmm-based error detection of dance step selection for dance partner robot –MS DanceR–," *IROS-2006*, IEEE/RSJ, 2006, pp. 5631–5636.

[9] A. Nakazawa *et al.*, "Imitating human dance motions through motion structure analysis," *IROS 2002*, IEEE/RSJ, 2002, pp. 2539–2544.

[10] S. Kotosaka and S. Schaal, "Synchronized robot drumming by neural oscillators," in *Proc. of Int'l Sympo. Adaptive Motion of Animals and Machines*, 2000.

[11] M. Michalowski *et al.*, "A dancing robot for rhythmic social interaction," in *Proc. of ACM/IEEE Int'l Conf. on Human-Robot Interaction (HRI 2007)*. IEEE, 2007, pp. 89–96.

[12] K. Yoshii *et al.*, "A biped robot that keeps steps in time with musical beats while listening to music with its own ears," *IROS-2007*, IEEE/RSJ, 2007, pp. 1743–1750.

[13] M. Goto, "An audio-based real-time beat tracking system for music with or without drum-sounds," *J. of New Music Research*, vol. 30, no. 2, pp. 159–171, 2001.

[14] R. Takeda *et al.*, "Exploiting known sound sources to improve ica-based robot audition in speech separation and recognition," *IROS 2007*, IEEE/RSJ, 2007, pp. 1757–1762.

[15] F. Gouyon *et al.*, "An experimental comparison of audio tempo induction algorithms." *IEEE Trans. Audio, Speech and Language Processing*, vol. 14(5), pp. 1832–1844, 2006.

[16] M. Goto and Y. Muraoka, "A real-time beat tracking system for audio signals." in *Proceedings of the International Computer Music Conference*. ICMA, 1995, pp. 171–174.

[17] K. Jensen and T. Andersen, "Real-time beat estimation using feature extraction," *Proc. Computer Music Modeling and Retrieval Symposium, Lecture Notes in Computer Science. Springer Verlag*, 2003.

[18] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "Rwc music database: Popular, classical, and jazz music databases," in *Int'l Conf. Music Information Retrieval*, 2002, pp. 287–288.