Proceedings of the 2007 IEEE/RSJ International
Conference on Intelligent Robots and Systems
San Diego, CA, USA, Oct 29 - Nov 2, 2007

A Biped Robot that Keeps Steps in Time with Musical Beats while Listening to Music with Its Own Ears

Kazuyoshi Yoshii, Kazuhiro Nakadai, Toyotaka Torii, Yuji Hasegawa,
Hiroshi Tsujino, Kazunori Komatani, Tetsuya Ogata, and Hiroshi G. Okuno

Abstract— **We aim at enabling a biped robot to interact with humans through real-world music in daily-life environments, e.g., to autonomously keep its steps (stamps) in time with musical beats. To achieve this, the robot should be able to robustly predict the beat times in real time while listening to musical performance with its own ears (head-embedded microphones). However, this has not previously been addressed in most studies on music-synchronized robots due to the difficulty in predicting the beat times in real-world music. To solve this problem, we implemented a beat-tracking method developed in the field of music information processing. The predicted beat times are then used by a feedback-control method that adjusts the robot's step intervals to synchronize its steps in time with the beats. The experimental results show that the robot can adjust its steps in time with the beat times as the tempo changes. The resulting robot needed about 25 [s] to recognize the tempo change after it and then synchronize its steps.**

I. INTRODUCTION

Humanoid robots that behave like humans have recently gained a lot of popularity. They can interact with humans through conversations and gestures in many science-fiction stories. If robots are endowed with intelligence and shapes similar to those of humans, we can easily predict their mental and physical dynamics without any special knowledge, i.e., by reflecting our mental and physical models. In other words, the thoughts and movements of humanoid robots should be similar to those of humans in various situations. This is an important factor for seamlessly introducing robots into our daily-life environments. In the future, humanoid robots will play roles not only in task-oriented missions (e.g., assisting the elderly) but also in leisure-time activities (e.g., playing sports with humans). We focus on music appreciation, which is one of the most popular leisure-time activities.[1]

Our goal is to build an *intelligent* humanoid-robot dancer that can interact with humans through various musical pieces. Capturing rhythm structures and synchronizing body motions (e.g., hand-clapping and foot-tapping) are fundamental capabilities in music appreciation.[2] To acquire these capabilities, robots should be able to understand music and synchronize dancing movements in time with it while listening to it with their own head-embedded microphones (ears). As an initial step toward achieving our goal, we developed a biped robot that can stamp its feet in time with musical beats (temporal positions of quarter notes).

Researchers' interests have recently centered on improving physical functions to imitate the complex human movements. For example, humanoid robots developed by Toyota Motor Co., Ltd. that play the trumpet with their mouths and fingers appeared in the 2005 World Exposition [1]. However, they cannot play non-registered musical pieces because motion sequences should be manually programmed for each piece. That is, these programmed robots cannot dynamically interact with humans through various pieces; their bodies are like "buried treasure" without human-like intelligence.

We aim at associating an *intelligent function* that predicts the next beat times in real time with a *physical function* that synchronizes steps with the predicted beats. This is a comprehensive approach that imitates the human intelligence of associating the brain with the body in dancing. Our study differs in this point from many previous ones that have focused on imitating the external body motions of humans. To achieve the intelligent function, we used a beat-tracking method based on a pure signal-processing algorithm [2]. To achieve the physical function, we used a feedback control method that reduces the temporal differences between the actual steps and musical beats.

The rest of this paper is organized as follows. Section II introduces related work. Section III discusses problems and approaches in developing the intelligent and physical functions of a beat-synchronized biped robot. Sections IV and V explain the implementations of these functions. Section VI reports on our experiments that used popular CD recordings. Section VII summarizes the key points of this study.

II. STATE-OF-THE-ART MUSIC ROBOTS

The humanoid robot is an interesting research subject not only in the field of mechanical engineering but also in that of intelligence science. However, there are a few studies that integrate promising methods from both fields.

Many researchers of mechanical engineering have engaged in improving physical functions. In the history of hardware improvements, one of the most active areas concerns biped robots. ASIMO, developed by Honda Motor Co., Ltd. [3], is an advanced biped robot that can walk using its two legs. QRIO, by SONY Co., Ltd. [4], can generate various dancing movements. Nakazawa *et al.* [5] let a biped robot called

K. Yoshii, K. Komatani, T. Ogata, and H. G. Okuno are with the Department of Intelligence Science and Technology, Graduate School of Informatics, Kyoto University, Kyoto, 606-8501, Japan (e-mail: yoshii@kuis.kyoto-u.ac.jp, {komatani, ogata, okuno}@i.kyoto-u.ac.jp).

K. Nakadai, T. Torii, Y. Hasegawa, and H. Tsujino are with the Honda Research Institute Japan Co., Ltd., Wako, Saitama, 351-0114, Japan (e-mail: {nakadai, tory, yuji.hasegawa, tsujino}@jp.honda-ri.com).

[1]In Japan, the three most popular leisure-time activities are watching movies, dining out, and music appreciation.

[2]Three major elements of music are rhythm, melody, and harmony. Among them, the rhythm is considered as the most fundamental element.

HRP-2 imitate the spatial trajectories of complex motions of a Japanese traditional folk dance by using a motion capture system. Despite the flexibility of motion generation, a problem is that these robots cannot autonomously determine the appropriate timing of dancing movements while interacting with auditory environments, i.e., while listening to music.

In the field of intelligence science concerning robotics, one of the hottest topics is *robot audition*, which aims at understanding external auditory environments through robot-embedded microphones (ears). For example, some studies tried to recognize simultaneous speech signals of multiple people [8], [9] or to determine the spatial positions of multiple sound sources [10]. However, understanding musical audio signals (*music scene analysis*) through robot-embedded ears has not previously been addressed.

A few studies have developed music-synchronized robots by using promising results from both fields. The earliest work was done by researchers at Waseda University, Japan in 1984 [11]. They created a robot musician called WABOT-2 that autonomously played the electronic organ with its own fingers while reading ordinary musical scores with its camera (eye) and determining the fingering. Afterwards, a more refined robot named WABIAN was developed that could generate dancing motions according to MIDI signals [12]. The motion speed could be interactively changed by moving a conductor's baton fast or slowly. MS DanceR, developed by Kosuge's group [13]–[15], is a robot dancer that dances with a human partner while predicting the steps of the partner. Note that these robots do not react to musical audio signals. Kotosaka and Schaal [16] developed a robot that plays the drum while synchronizing its drumming strokes to those of a human drummer. Although this robot detects the stroke times from the monophonic audio signals of human drumming with its own ears, it is quite difficult to predict beat times from the polyphonic audio signals of real-world music.

The development of artificial dancers was also addressed in the field of music information processing. M. Goto [2] developed a computer-graphics system named Cindy that displays virtual dancers whose motions change in time to musical beats in real time. However, the physical constraints of real robot dancers were not taken into account, because the virtual dancers can precisely and instantaneously change the positions of their body parts in time to musical beats.

To develop a robot that autonomously synchronizes its steps with beat times of musical audio signals, we should achieve high temporal accuracy of motions by predicting unavailable future information (beat times). This is a key point that differs from conventional studies on robotics that focus on improving the spatial accuracy of motions by using complete information for environment recognition.

III. ARCHITECTURE

This section describes the system architecture for developing the stamping humanoid robot. First, we define our task. Second, we explain the physical specifications of our robot. Finally, we discuss the problems and approaches for achieving the intelligent and physical functions.

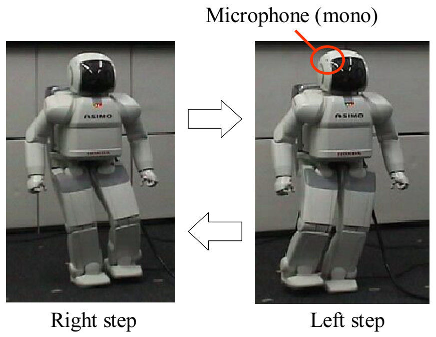

Fig. 1. Foot stamping of biped robot while listening to music with its head-embedded microphone.

A. Beat-synchronized Stamping Task

Our goal is to synchronize robot steps with musical beats that are automatically predicted from musical audio signals. This task can be divided into two parts: *beat prediction* and *step control*, which correspond to the intelligent and physical functions. To achieve human-like foot stamping, the system should satisfy the following two requirements:

- *Executing beat prediction and step control in parallel*: The robot should not alternate between beat prediction and step control to avoid the negative effects of motor sounds during beat prediction.
- *Continuing steps during change in tempo*: The robot should not stop performing steps when a change in tempo is detected. Humans usually deal with a change in tempo by gradually making their steps faster or slower.

The former requirement makes our task more difficult compared with standard human-robot interaction where robots first concentrate on environment recognition (e.g., speech recognition) and then move their bodies.

B. Specifications of Humanoid Robot

Our humanoid robot is ASIMO, which has two legs like humans and can stamp its feet on the floor, i.e., perform steps in a stationary location, as outlined in Fig. 1. The step interval is limited to between 1,000 and 2,000 [ms]. If the tempos of musical pieces are between 61 and 120 M.M.,[3] the robot can perform one step per two beats (quarter notes). We assume that the tempos of input audio signals are between 61 and 120 M.M. and that the time signature is 4/4. The robot records these signals with its own single microphone embedded in the front of the head. Of course, the recorded signals include many noises and motor sounds caused by the real-time motions.

We can only control step intervals externally by sending control commands to the robot via a TCP socket. Here, we should take into account physical constraints (e.g., inertia, friction, and latency) and the effects of autonomous posture control. For example, the robot needs some time to start moving its legs after sending it a command. In addition, the robot cannot accurately perform steps at an interval specified

[3]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].

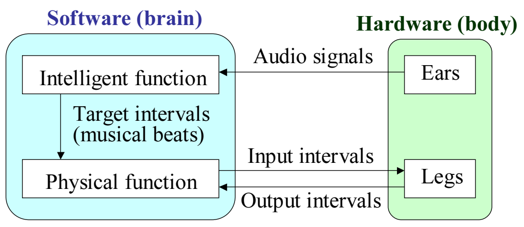

Fig. 2. Overview of system architecture based on integration of intelligent and physical functions.

by the command. Note that we receive an actual step interval from the robot via a TCP socket when each step is completed. To clarify the concept, we define the following three terms, as outlined in Fig. 2.

- A *target* interval is a temporal duration obtained by analyzing the tempo of a musical audio signal. That is, the target interval is equal to two or four times the quarter-note length.
- An *input* interval is an instruction value specified by a control command.
- An *output* interval is an actual interval of robot steps obtained as feedback information.

Therefore, our goal is to make the timing and intervals of robot steps (outputs) equal to those of musical beats (targets) by adjusting input intervals.

C. Intelligent Function: Beat Prediction

This function predicts the next beat times in real time because the robot cannot immediately change behaviors after sending it control commands. Therefore, control commands should be sent in advance of the beat times to generate beat-synchronized steps. This function is called "real-time beat tracking" in the field of music information processing.

1) Problems: There are three major problems associated with tracking the beat times in real time.

- *Problem 1:* The beat-time candidates should be detected from polyphonic audio signals by focusing on appropriate musical contents.
- *Problem 2:* There is ambiguity in selecting appropriate combinations of beat times from among the detected beat-time candidates.
- *Problem 3:* The real-time beat tracking should be robust against motor sounds and other noises to satisfy the first requirement mentioned in Section III-A.

2) Approach: By implementing a real-time beat-tracking method [2], these three problems can be solved as follows:

- *Strategy 1 (detection of chord changes and drum-sound onsets):* The method focuses on drum-sound onsets and chord changes as important musical cues that indicate beat-time candidates. This is based on the general fact that drum-sound onsets and chord changes likely occur at beat times.
- *Strategy 2 (multi-agent architecture):* The method manages multiple agents that have different interpretations of locations for beat times. These agents always predict the next beat times simultaneously, and an agent that has the most confident interpretation outputs the beat times. This enables reliable beat tracking.
- *Strategy 3 (weak musical-content analysis):* The method roughly calculates the reliabilities of chord changes and drum-sound onsets continuously, i.e., it does not try to accurately judge whether chord changes and drum-sound onsets occur. The comprehensive judgment is finally done on the basis of the multi-agent architecture. This results in improved robustness.

D. Physical Function: Step Control

This function synchronizes the timing of robot steps (outputs) with that of musical beats (targets) as promptly as possible. To achieve this, we should dynamically adjust the input intervals on the basis of a feedback control method. This method reduces the amount of the differences (errors) not only in timing but also in the intervals between outputs and targets.

1) Problems: There are two major problems associated with designing the feedback control method.

- *Problem 1:* Physical constraints and the second requirement mentioned in Section III-A prohibit the robot from immediately changing its actual step interval to match the target interval when the tempo change is detected by the intelligent function.
- *Problem 2:* Adjustment of input intervals for reducing the amount of interval errors often conflicts with that for reducing the amount of timing errors.

2) Approach: To solve these two problems, we propose the following two strategies.

- *Strategy 1 (gradual adjustment of input intervals):* The method gradually changes the input intervals so that the outputs gradually converge to the targets. This contributes to fast and stable convergence.
- *Strategy 2 (dynamic weighting):* The method dynamically switches weighting strategies to achieve not only fast convergence but also good stability after convergence. When the amount of errors in the output intervals is large, the method reduces it. When convergence has almost been achieved, the method mainly reduces the amount of timing errors without largely changing the input intervals.

IV. REAL-TIME BEAT TRACKING

This section describes the implementation of the intelligent function. In this paper, we implemented a real-time beat-tracking method proposed by Goto [2]. Figure 3 shows an overview of the method, which is based on the multi-agent architecture. The method outputs the next beat time and the current tempo in real time.

In the frequency analysis stage, the spectrogram is consecutively obtained by applying the short time Fourier transform (STFT) with a Hanning window of 4096 [points] and a shifting interval of 512 [points] to input audio signals sampled at 44.1 [kHz]. Then, the frequency components derived from the onsets are roughly extracted from the spectrogram. Using

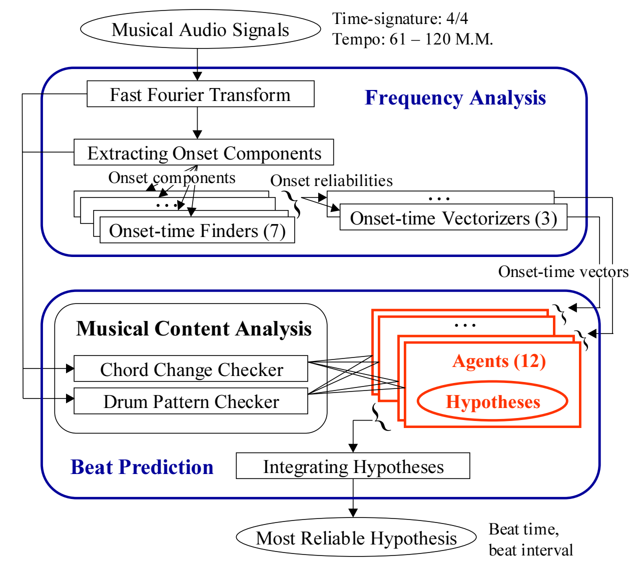

Fig. 3. Overview of real-time beat-tracking method based on multi-agent architecture.

these onset components, seven onset-time finders that focus on different frequency ranges evaluate onset reliabilities at each frame. The seven onset reliabilities of each frame are transformed into vector representations called *onset-time vectors* by three onset-time vectorizers.

In the beat prediction stage, the method manages twelve agents that create parallel hypotheses from the onset-time vectors on the basis of different strategies, i.e., each agent calculates the beat interval and predicts the next beat time. By communicating with a chord change checker and a drum pattern checker, the agent evaluates the reliability of its own hypothesis. Then, the method gathers all the hypotheses and determines the most reliable one as the final output.

A. Strategy Parameters of Multiple Agents

Each agent uses different strategy parameters for predicting the next beat time, i.e., for maintaining a hypothesis of the beat-time sequence. There are three kinds of parameters (The effects of these parameters are described later):

- *Frequency focus type* (S_1): This parameter determines an onset-time vectorizer that provides an agent with onset-time vectors. Its value is chosen from among *type-all*, *type-low*, and *type-mid* corresponding to the three vectorizers focusing on the all, low, and middle frequency ranges, respectively.
- *Auto-correlation period* (S_2): This parameter takes a value of either 500 or 1000 [frames], which determines the window size for calculating the vector auto-correlation of the onset-time vector sequence.
- *Initial peak selection* (S_3): This parameter takes a value of either *primary* or *secondary*, which only makes sense when the hypothesis reliability is low. When the value is primary, the largest peak in a prediction field is initially selected and regarded as the next beat time. Otherwise, the second-largest peak is selected.

All twelve agents are grouped into six pairs. Two agents in each pair share the same values of S_1 and S_2, and have the different values of S_3. These paired agents examine the same beat interval and then cooperatively predict the next beat times, which will always differ by half the beat interval (eighth-note duration). To achieve this, one agent interacts with the other in each pair through a prediction field, which is an expectancy curve that represents the time at which the next beat is expected to occur (Fig. 4).

B. Frequency Analysis Stage

We explain the algorithm of the frequency analysis stage.

1) Extraction of Onset Components: The onset times are detected by frequency analysis that takes into account factors such as the rapidity of an increase in power. Here, let $p(t,f)$ is the power at time frame t and frequency bin f. An onset component (a frequency component that is likely derived from an onset) at t and f, $d(t,f)$, is obtained by

$$d(t,f) = \begin{cases} \max(p(t,f), p(t+1,f)) - PrevPow, \\ \quad \text{if } \min(p(t,f), p(t+1,f)) > PrevPow, \\ 0, \text{ otherwise}, \end{cases} \quad (1)$$

$$\text{where } PrevPow = \max(p(t-1,f), p(t-1,f\pm 1)). \quad (2)$$

2) Onset-time Finders: Seven onset-time finders assign the onset reliabilities to each time frame in seven frequency ranges ($0-125$ [Hz], $125-250$ [Hz], $250-500$ [Hz], $500-1000$ [Hz], $1-2$ [kHz], $2-4$ [kHz], and $4-11$ [kHz]). In each range, onset times are roughly detected by picking peak times in the sum, $D_f(t)$, along the time axis, where $D_f(t) = \sum_f d(t,f)$, and the frequency range of $\sum_f$ is limited. The sum, $D_f(t)$, is linearly smoothed with a convolution kernel before its peak time is calculated.

The onset reliability is then assigned to each time frame. If an onset is found at frame t, the onset reliability is given by $D_f(t)$, otherwise it is set to zero.

3) Onset-time Vectorizers: Three onset-time vectorizers transform the seven reliabilities provided by the seven finders into three seven-dimensional onset-time vectors at each frame on the basis of the different sets of frequency weights (strategy parameter S_1). A sequence of onset-time vectors is sent to an agent specified by the strategy parameter S_1.

C. Beat Prediction Stage

We explain the algorithm of the beat prediction stage.

1) Calculation of Beat Interval: To determine the beat interval (temporal duration between adjacent beat times), each agent receives a sequence of onset-time vectors and calculates its vector auto-correlation. The windowed and normalized auto-correlation, $Ac(\tau)$, is given by

$$Ac(\tau) = \frac{\sum_{t=c-AcLen}^{c} \mathrm{win}(c-t, AcLen)(o(t)\cdot o(t-\tau))}{\sum_{t=c-AcLen}^{c} \mathrm{win}(c-t, AcLen)(o(t)\cdot o(t))}, \quad (3)$$

where $o(t)$ is a 7-dimensional onset-time vector at frame t, and c is the current time. $AcLen$ is strategy parameter S_2, and$\mathrm{win}(t,s)$ is a window function whose size is s:

$$\mathrm{win}(t,s) = \begin{cases} 1.0 - 0.5\ t/s & (0 \le t \le s), \\ 0 & (\text{otherwise}). \end{cases} \quad (4)$$

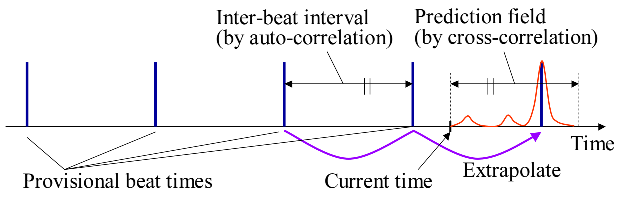

Fig. 4. Predicting next beat time using cross-correlation.

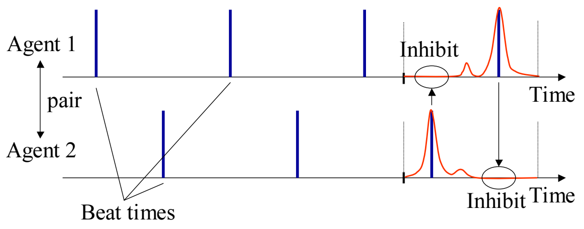

Fig. 5. Inhibiting prediction fields of paired agents.

The beat interval is determined as τ that maximizes $Ac(\tau)$ under the condition that τ is between 43 [frames] (the length of quarter note in 120 M.M) and 85 [frames] (that in 61 M.M).

2) Prediction of Next Beat Time: To predict the next beat time, each agent forms a prediction field, as outlined in Fig. 4. The prediction field is obtained by calculating the windowed cross-correlation, $Cc(\tau)$, using onset-time vectors:

$$\begin{aligned} Cc(\tau) &= \sum_{t=c-CcLen}^{c} \Big(\mathrm{win}(c-t, CcLen) O(t) \\ &\qquad \sum_{m}^{CcNumBeats} \delta(t - T_{\mathrm{p}}(c+\tau, m))\Big), \qquad (5) \\ T_{\mathrm{p}}(t,m) &= \begin{cases} t - I(t) & (m=1), \\ T_{\mathrm{p}}(t,m-1) - I(T_{\mathrm{p}}(t,m)) & (m \geq 1), \end{cases} \qquad (6) \\ \delta(x) &= \begin{cases} 1 & (x=0), \\ 0 & (x \neq 0), \end{cases} \qquad (7) \end{aligned}$$

where $CcLen$ $(= CcNumBeats\ I(c))$ is the window size for calculating the cross-correlation, and $O(t)$ is the sum of seven dimensions of $o(t)$. $I(t)$ is the beat interval at frame t, and $CcNumBeats$ is a constant factor that determines how many previous beats are taken into account in calculating the cross-correlation. $\{T_{\mathrm{p}}(t,1), \cdots, T_{\mathrm{p}}(t, CcNumBeats)\}$ is a sequence of provisional beat times that is determined recursively when frame t is regarded as the next beat time. The prediction field is given by $Cc(\tau)$, where $0 \leq \tau \leq I(c) - 1$.

Each agent then selects the next beat time (time $c+\tau$) that gives the maximum peak in the prediction field after the field is inhibited by its paired agent, as outlined in Fig. 5. When the reliability of a hypothesis is low, the agent initially selects the peak in the prediction field on the basis of strategy parameter S_3. The agent then pursues a peak close to time $(c+\tau)+I(c+\tau)$.

3) Calculation of Hypothesis Reliability: To evaluate the hypothesis reliability, each agent communicates with a chord-change checker and a drum-pattern checker. The former evaluates the coincidence of beat times with chord changes. The latter evaluates the degree of similarity between a set of roughly-detected drum-sound onsets and registered typical drum patterns.

The chord-change checker calculates two kinds of chord-change possibilities, i.e., quarter-note-level possibility and eighth-note-level possibility, by slicing the spectrogram into strips at the provisional beat times. These possibilities represent how likely a chord is to change at each quarter-note time and at each eighth-note time under the current beat-time hypothesis. The use of the two kinds of possibilities enables us to select an appropriate hypothesis from the paired hypotheses that differ by eighth-note duration.

The drum-pattern checker first roughly detects the onset time of a bass drum by using onset components and the onset time of a snare drum by using noise components. These onset times are then formed into the drum patterns by making use of the provisional beat times. The checker then compares the drum patterns with registered drum patterns.

4) Integration of Multiple Hypotheses: The manager classifies all agent-generated hypotheses into groups on the basis of the beat time and beat interval. Each group has an overall reliability given by the sum of the reliabilities of the group's hypotheses. The manager then selects the dominant group that has the highest reliability. The reliable hypothesis in the most dominant group is thus selected as the output.

V. FEEDBACK STEP CONTROL

This section describes the implementation of the physical function, which aims at synchronizing the timing of robot steps (outputs) with that of musical beats (targets) by only adjusting the step intervals of commands (inputs) sent to the robot. In general, the feedback-based robot control is quite difficult when precise target values are not available, e.g., when the beat times include prediction errors and quantization errors. Although typical feedback-control theories allow outputs to include errors, those included in targets are not considered. Note that in related work, the correct beat times were given as MIDI signals [12] or could be easily detected from monophonic audio signals [16].

Putting aside this theoretical difficulty (beyond our scope), we take a simple approach that simultaneously reduces the amount of errors in step timing and those in output intervals.

A. Basic Algorithm

Our method adjusts input intervals by using feedback information, as outlined in Fig. 6. Here, let $T_{out}(n)$ be the time when the previous step-completion signal sent by the robot is received by the control PC, where n is the index of the previous beat time. The objective of the method is given as follows:

$$\begin{aligned} \text{Time:}\quad T_{out}(n+1) &\rightarrow T(n+1), \qquad (8) \\ \text{Interval:}\quad I_{out}(n+1) &\rightarrow T(n+1) - T(n), \qquad (9) \end{aligned}$$

where $T_{out}(n+1)$ is the next receiving time and $I_{out}(n+1)$ is the actual interval of the next step. $T(n)$ and $T(n+1)$ are the previous and next beat times. $I(n+1)$ is the interval between adjacent beats $T(n)$ and $T(n+1)$. Although it is theoretically

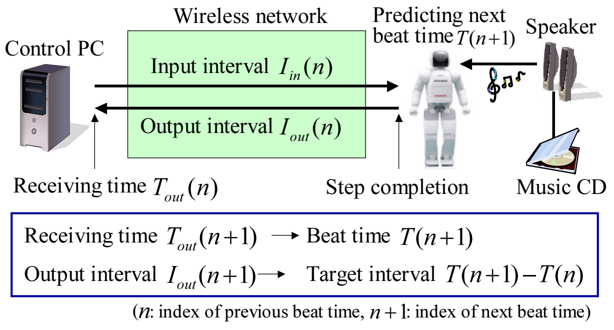
Wireless network
Predicting next beat time $T(n+1)$
Speaker
Control PC
Input interval $I_{in}(n)$
Output interval $I_{out}(n)$
Receiving time $T_{out}(n)$
Step completion
Music CD
Receiving time $T_{out}(n+1)$ → Beat time $T(n+1)$
Output interval $I_{out}(n+1)$ → Target interval $T(n+1)-T(n)$
(n: index of previous beat time, $n+1$: index of next beat time)

Fig. 6. Overview of feedback control method: The objective is to reduce the amount of errors not only in output timing but also in output intervals.

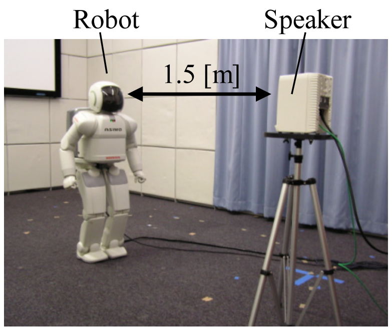

Fig. 8. Overview of experimental condition: The system concerning to the robot is completely separated from that concerning to the music playback.

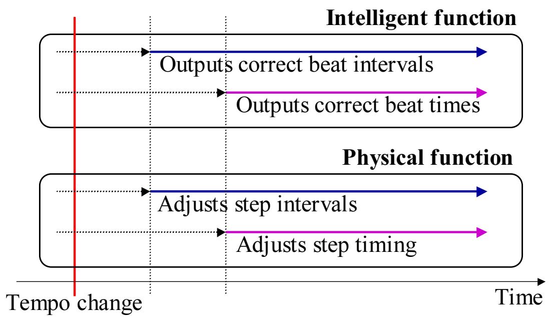

Fig. 7. Overview of pipeline-processing-like step control.

sufficient to take into account the synchronization of beat times with receiving times (Eq. (8)), we also focused on that of their intervals (Eq. (9)). This contributes to good stability. Therefore, the updating formula is given by

$$I_{in}(t+1) = I_{in}(t) + \beta_I (I(t) - I_{out}(t))) + \beta_T (T(t) - T_{out}(t))), \quad (10)$$

where β_I and β_T are weighting factors that determine the adjustment of the actual step interval and that of the step-completion time.

B. Dynamic Adjustment of Weighting Factors

To determine the values of the two weighting factors, β_I and β_T, we should take into account the two strategies described in Section III-D.2. Here, we judge the convergence of output intervals by

$$|I(t) - I_{out}(t)| < \varepsilon I(t), \quad (11)$$

where ε is an error tolerance, which we set to a small value, 0.02. When the amount of errors in output intervals is large, i.e., Eq. (11) is not satisfied, β_I and β_T are empirically set to 0.30 and 0.00. After Eq. (11) is satisfied, β_I and β_T are set to 0.10 and 0.02.

There are two important reasons for switching the two weighting strategies. The first reason is to take a pipeline-processing-like approach, as outlined in Fig. 7. The intelligent function (beat-prediction function) requests less time to estimate the correct beat intervals than to output the correct beat times. Therefore, the physical function can adjust the step intervals ahead of receiving the correct beat times.

The second reason is to improve robustness of the system against inevitable errors caused by the intelligent function. Although the beat times often include prediction errors, the beat intervals tend to be accurately estimated. Therefore, it is important to constantly make the output intervals to be close to the target intervals.

VI. EXPERIMENTS

This section reports on our experiments that evaluated whether our robot could synchronize its steps with musical beats while predicting the beats from popular music.

A. Conditions

To prepare a 4-minute input audio signal, we selected four songs (No. 57, No. 87, No. 18, and No. 62) from the RWC music database (RWC-MDB-P-2001) developed by Goto *et al.* [17]. They include vocals and various instruments as commercial CDs do. Their tempos were 70, 90, 112, and 81 M.M, respectively. We concatenated four 60-s segments that were extracted from the four pieces. The prepared audio signal was played back by using an ordinary speaker in a standard room, as outlined in Fig. 8. The distance between the robot and the speaker was 1.5 [m].

We tested two systems for comparison. The one was our proposed version that used both the intelligent and physical functions. We called the other an "oracle" version in which the intelligent function was disabled, where the correct beat times were given as MIDI signals.

B. Results

We evaluated the experimental results in terms of errors (temporal gaps) included in intervals and timing of robot steps. Small errors between targets and outputs indicate the better performance.

We will first discuss the oracle system to evaluate basic capabilities of the physical function. Figure 9 shows the historical record of target intervals, input intervals, and output intervals. We confirmed that the robot can adjust the output intervals to match the precise target intervals given as MIDI signals. Note that the robot needed about 10 [s] to perform the adjustment after the tempo was changed. Figure 10 shows the historical record of errors included in the output timing (temporal gaps between musical beats and

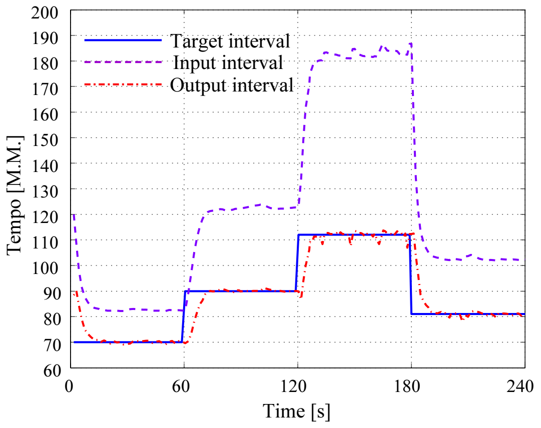

Fig. 9. Results of controlling step intervals by oracle system.

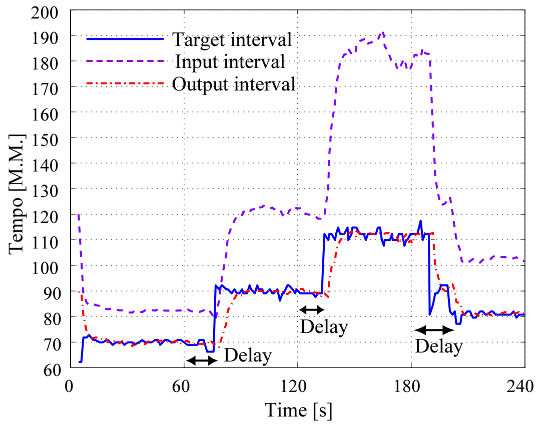

Fig. 11. Results of controlling step intervals by our proposed system.

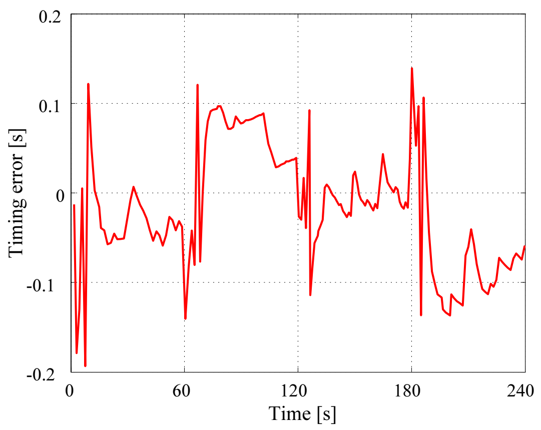

Fig. 10. Results of controlling step timing by oracle system.

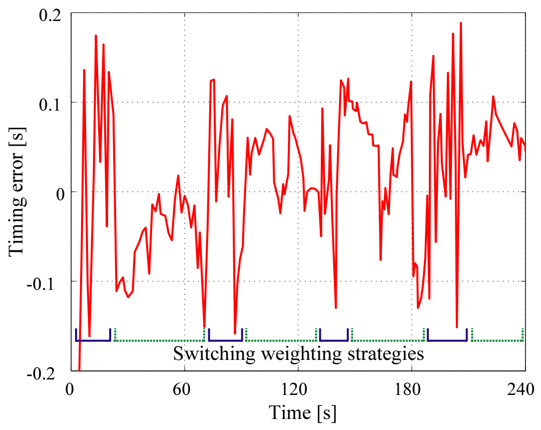

Fig. 12. Results of controlling step timing by our proposed system.

robot steps). The amount of the timing errors was retained under 100 [ms] when the output intervals were close to the target intervals, i.e., after the interval convergence was almost achieved. These results proved that the physical function worked well.

Then, we evaluated the proposed system. Figure 11 shows the historical record of the three types of intervals. We confirmed the following capabilities of the system:

- The intelligent function can estimate the correct tempos (beat intervals) of musical audio signals, although it takes about 15 [s] to detect a change in change.
- The physical function can adjust the output intervals to match the target intervals while automatically estimating the target intervals from musical audio signals.

Figure 12 shows the historical record of timing errors. Although the amount of errors tend to be larger than that obtained by the oracle system, it was also retained under 100 [ms] when the physical function took a weighting strategy that focused on reducing the amount of timing errors. In total, the robot needed 25 [s] to synchronize its stamps with the predicted musical beats. We can conclude that this is a promising result although the robot requires more time to perform the synchronization than humans.

VII. CONCLUSION

We developed a biped humanoid robot that stamps its feet in time with musical beats like humans. This was achieved by building a computational mechanism that duplicates the natural human ability in terms of associating intelligent and physical functions. The former predicts the beat times in real time for polyphonic musical audio signals. The latter then synchronizes step motions with the beat times by gradually reducing the amount of errors in intervals and timing. To implement these functions, we associated a beat-tracking method with a feedback control method. These methods have been originally investigated in different research fields. This is an indispensable study for advanced robotics.

Our robot represents the significant first step in creating an intelligent robot dancer that can generate rich and appropriate dancing movements that correspond to properties (e.g., genres and moods) of musical pieces. To achieve this, we should increase the variety of robot's movements and use promising techniques analyzing musical content, e.g., genre classification and mood detection. In addition, we plan to enhance human-machine interaction through dancing by improving the response time with an adaptive control method such as a neuronal controller [18].

REFERENCES

[1] The 2005 World Exposition (EXPO 2005), Aichi, Japan, http://www.expo2005.or.jp/en/index.html.

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

[3] ASIMO, Honda Motor Co., Ltd., http://world.honda.com/ASIMO/.

[4] Sony Co., Ltd., http://www.sony.net/SonyInfo/QRIO/.

[5] A. Nakazawa, S. Nakaoka, K. Ikeuchi, and K. Yokoi, "Imitating Human Dance Motions through Motion Structure Analysis," *IROS*, pp. 2539–2544, 2002.

[6] S. Kawamura, J. Jun, K. Kanaoka, and H. Ichii, "A Cascaded Feedback Control Scheme for Trajectory Tracking of Robot Manipulator Systems with Actuator Dynamics," *IROS*, pp. 1504–1509, 2006.

[7] P. Yuan, "An Adaptive Feedback Scheduling Algorithm for Robot Assembly and Real-Time Control Systems," *IROS*, pp. 2226–2231, 2006.

[8] R. Takeda, S. Yamamoto, K. Komatani, T. Ogata, and H. G. Okuno, "Missing-Feature based Speech Recognition for Two Simultaneous Speech Signals Separated by ICA with a pair of Humanoid Ears," *IROS*, pp. 878–885, 2006.

[9] S. Yamamoto, K. Nakadai, M. Nakano, H. Tsujino, J.-M. Valin, K. Komatani, T. Ogata, and H. G. Okuno, "Real-Time Robot Audition System that Recognizes Simultaneous Speech in the Real World," *IROS*, pp. 5333–5338, 2006.

[10] K. Nakadai, H. Nakajima, M. Murase, H. G. Okuno, Y.Hasegawa, and H. Tsujino, "Real-Time Tracking of Multiple Sound Sources by Integration of In-Room and Robot-Embedded Microphone Arrays," *IROS*, pp. 852–859, 2006.

[11] Science and Engineering Research Laboratory, "Special Issue on Wabot-2," Bull. No. 112, Waseda University, Tokyo, 1985.

[12] WABIAN, http://www.takanishi.mech.waseda.ac.jp/research/

[13] K. Kosuge, T. Hayashi, Y. Hirata, and R. Tobiyama, "Dance Partner Robot –MS DanceR–," *IROS*, pp. 3459–3464, 2003.

[14] T. Takeda, Y. Hirata and K. Kosuge, "HMM-based Dance Step Estimation for Dance Partner Robot –MS DanceR–," *IROS*, pp. 1602–1607, 2005.

[15] T. Takeda, Y. Hirata, Z. Wang, and K. Kosuge, "HMM-based Error Detection of Dance Step Selection for Dance Partner Robot –MS DanceR–," *IROS*, pp. 5631–5636, 2006.

[16] S. Kotosaka and S. Schaal, "Synchronized Robot Drumming by Neural Oscillators," *Int. Sympo. Adaptive Motion of Animals and Machines*, 2000.

[17] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "RWC Music Database: Popular, Classical, and Jazz Music Databases," *Int. Conf. Music Information Retrieval*, pp. 287–288, 2002.

[18] T. Geng, B. Porr, and F. Wörgötter, "Fast BipedWalking with a Sensor-driven Neuronal Controller and Real-time Online Learning," *The Int. Journal of Robotics Research*, Vol. 25, No. 3, pp. 243–259, 2006.