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

# A Robot Listens to Music and Counts Its Beats Aloud by Separating Music from Counting Voice

Takeshi Mizumoto, Ryu Takeda, Kazuyoshi Yoshii, Kazunori Komatani, Tetsuya Ogata, Hiroshi G. Okuno
*Graduate School of Informatics, Kyoto University, Sakyo, Kyoto 606-8501, Japan*
{*mizumoto, rtakeda, yoshii, komatani, ogata, okuno*}*@kuis.kyoto-u.ac.jp*

***Abstract*— This paper presents a beat-counting robot that can count musical beats aloud, i.e., speak "one, two, three, four, one, two, ..." along music, while listening to music by using its own ears. Music-understanding robots that interact with humans should be able not only to recognize music internally, but also to express their own internal states. To develop our beat-counting robot, we have tackled three issues: (1) recognition of hierarchical beat structures, (2) expression of these structures by counting beats, and (3) suppression of counting voice (self-generated sound) in sound mixtures recorded by ears. The main issue is (3) because the interference of counting voice in music causes the decrease of the beat recognition accuracy. So we designed the architecture for music-understanding robot that is capable of dealing with the issue of self-generated sounds. To solve these issues, we took the following approaches: (1) beat structure prediction based on musical knowledge on chords and drums, (2) speed control of counting voice according to music tempo via a vocoder called STRAIGHT, and (3) semi-blind separation of sound mixtures into music and counting voice via an adaptive filter based on ICA (Independent Component Analysis) that uses the waveform of the counting voice as a prior knowledge. Experimental result showed that suppressing robot's own voice improved music recognition capability.**

## I. INTRODUCTION

Interaction through music is expected to improve the quality of symbiosis between robots and people in daily-life environment. Because human emotions have close relationship to music, music gives another communication channel besides spoken language. Music understanding robot may open new possible interactions with people by, for example, dancing, playing the instruments, or singing together.

We assume that music-understanding of robots consists of two capabilities: *music recognition* and *music expression*. Music expression is significant for the interaction because people cannot know the inner state of robots without observing its expression. In other words, this assumption means that we evaluate the capability of music understanding only by the Turing Test [1]. In addition, unbalanced design of music recognition and music expression should be avoided for symbiosis between people and robots, although it is not difficult to implement sophisticated robot behaviors without recognizing music.

One of critical problems in achieving such music-understanding robots is the fact that sounds generated by a robot itself (self-generated sound) interferes in music, for example, motor noises, musical instrument sounds, or singing voice. These noises cannot be ignored even if they are not loud, because their sound sources are very closed to the robot's ears. Please note that the power of sounds decreases according to the square of the distance. The performance of music expression usually generates sounds, which cannot be ignored by robot audition systems. In other words, music-understanding robot is a challenge toward intelligent robots in robot audition, because it needs to capture the self auditory model of its behaviors.

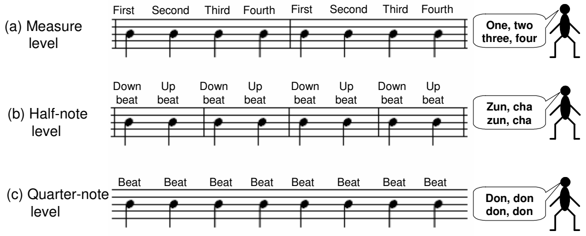

Fig. 1. Hierarchical beat structure

In this paper, we designed the architecture for music-understanding robot that is capable of dealing with the problem of self-generated sounds. The architecture integrates music recognition and expression capabilities, which have been dealt separately in conventional studies. Based on this architecture, we developed a beat counting robot. The robot listens to music with its own ear (one channel microphone) and counts the beats of 4-beat music by saying "one, two, three, four, one, two, three four, ..." aloud, as is shown in Fig. 1. The three main functions are required to build such a music robot:

(1) recognition of hierarchical beat structures of musical audio signal in the measure-level,
(2) expression of the beats with counting voice, and
(3) suppressing the robot's own counting voice

In this paper, we used the real-time beat tracking [6] for (1), selecting appropriate voices and controlling the timing of them for (2), and ICA based adaptive filter [7] for (3). The beat counting robot is considered as the first step toward singer robots, because the robot should recognize the hierarchical beat structures in order to align its singing voice to a music score.

The rest of paper is organized as follows: Section II introduces related works about music robots. Section III describes architecture for music-understanding robot. Section IV, V and VI explains the solutions of three problems for robot's capability of music recognition, expression, and suppressing self-generated sound, respectively. Section VII shows the experimental result about the capability of music

**978-1-4244-2058-2/08/$25.00 ©2008 IEEE.**

TABLE I

CAPABILITIES OF ROBOTS FOR MUSIC UNDERSTANDING IN RELATED WORKS.

| Conventional studies | Music recognition | | Music expression | |
|---|---|---|---|---|
| | Recognition target | Suppressing self-generated sounds | Means for expression | Expressed information |
| Conventional dancing robots | None | × | Previously prepared | - |
| Kozima et al. [2] | Power | × | Random motion | Quarter-note level |
| Kotosaka et al. [3] | Power | × | Playing drum | Quarter-note level |
| Yoshii et al. [4] | Beat structure | × | Keep stepping | Quarter-note level |
| Murata et al. [5] | Beat structure | × | Keep stepping and Humming | Half-note level |
| Our beat-counting robot | Beat structure | ○ | Counting beats | Measure level |

recognition and Section VIII summarizes this paper.

## II. STATE-OF-THE-ART MUSIC ROBOTS

Let us now introduce robots whose performance is related to music. From the viewpoint of our concept about understanding music, conventional humanoid robots that can dance or play instruments, such as QRIO or Partner Robot, seem only to have the capability of expressing music. To achieve the capability of recognizing music, the easiest strategy is to extract and predict the rhythm or melody from music that the robot's ear (microphone) hears. However, this is not sufficient for solving music recognition by robots, because they hear a mixture of music and self-generated sounds.

Some robots have explored the capability of music recognition, although none of them have dealt with this problem. Kozima *et al.* developed Keepon that dances while listening to music [2]. Its recognition failures are not obvious because Keepon has a small body, low DOFs (degrees of freedom) and random motion. Suppressing self-generated sounds is not required but this situation is specific to Keepon. Kotosaka *et al.* developed a robot that plays a drum synchronized to the cycle of periodic input using neural oscillators [3]. Their purpose was to make a robot that could generate rhythmic motion. Their robot could achieve synchronized drumming, although it only heard external sounds for synchronization. Yoshii *et al.* implemented a function on Asimo where it stepped with musical beats by recognizing and predicting the beat of popular music it heard [4]. Asimo was able to keep stepping even if the musical tempo changed. Murata *et al.* improved this function by adding to hum */zun/* and */cha/* synchronously according to the musical beats [5]. They pointed out that interference from the robot's humming voice degraded the performance of recognizing music, because the robot's voice was closer to the robot's microphone. The reason is that real-time beat tracking assumes that the only input is music. Therefore, self-generated sound has to be suppressed to improve the performance of beat tracking.

Table I compares the capabilities for recognizing and expressing music in related work. According to this table, even if a robot has the same capability for recognizing music, a different capability to express it makes an enormous different impression. Therefore, an intelligent music-understanding robot needs to integrate two capabilities for recognizing and expressing music. In addition, only our robot has the function of suppressing self-generated sound.

The aim of this study was for a robot to recognize and express a hierarchical beat structures (Fig. 1). Yoshii *et al.*'s, Murata *et al*'s and our robot shared the same capability for recognizing music, but their music expression capabilities were different. Yoshii *et al.*'s robot expresses its recognition by keeping steps, which means it expresses in quarter-note level. (Fig. 1 (c)) Murata *et al.*'s robot expresses its recognition by keeping steps and humming, which means that its expression is in half-note level. (Fig. 1 (b)) Our robot expresses it by counting voice, it means that our expression is in measure level. (Fig. 1 (a)) Thus, people can judge how the robot understands music by observing its expressions or behaviors, just like the Turing Test [1].

## III. ARCHITECTURE

### A. General Architecture

We encountered three issues in developing on music-understanding robot. These were:

1) its capability of music recognition,
2) its capability of music expression, and
3) suppressing its self-generated sounds.

To solve these problems systematically, we designed an architecture for our music-understanding robot. In designing the architecture, we referred to the model of "A Blueprint for the Speaker" proposed by Levelt [8]. According to this model, a human speaks through three modules: Conceptualizer, Formulator and Articulator. Similarly, a human listens to his own voice through two modules: Audition and Speech-Comprehension System.

Fig. 2 outlines the architecture for the music-understanding robot. It is composed of music-recognition and music-expression modules.

Let us explain the music-expression module. First, the Conceptualizer creates a plan about what to express, using knowledge about expression, e.g., lyrics, musical scores and a primitive choreography. Second, the Formulator generates a motion sequence according to the plan and generates motor instructions (inner expression). Consistency with musical knowledge is required while generating a motion sequence and motor instructions.

Next, we will explain the music-recognition module. First, the robot listens to a mixture of music and self-generated sound. Second, source separation separates the mixture into music and self-generated sound using inner expression. The separated music is sent to the Music recognizer and self-generated sound is sent to the Conceptualizer for feedback.

The music-expression module sends two sets of information to the music-recognition module: self-generated sound

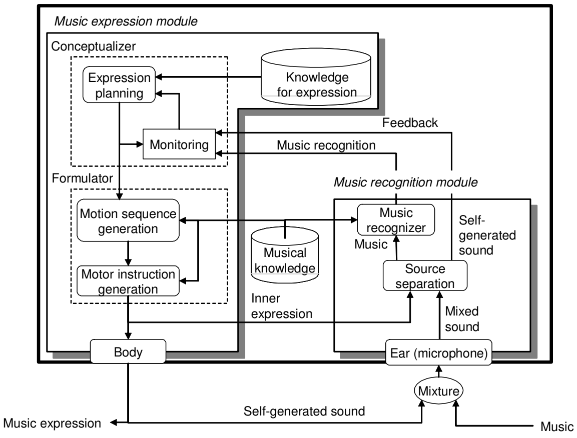

Fig. 2. General Architecture

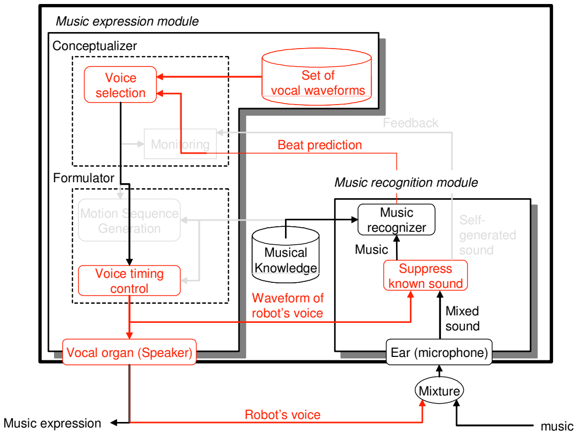

Fig. 3. Architecture of Beat-counting Robot

and inner expression. The music-recognition module sends two sets of information to the music-expression module: the results from the music recognizer and the separated self-generated sound. This interaction achieves cooperation between music-expression and music-recognition modules.

### *B. Specific Architecture for Beat-counting Robot*

We customized the general architecture for our beat-counting robot based on four assumptions:

1) The voice is used for music expression
We can generally express music in three ways, i.e., (a) Voice, (b) Motion, and (c) Voice and Motion. We adopted voice (a) because the main purpose of this study was suppressing the robot's self-generated sound. This assumption simplified the problem and enabled influences to be identified. Therefore, we replaced "Knowledge for Expression" (Fig. 2) with "Set of Vocal Waveforms." (Fig. 3) and "Body" (Fig. 2) with "Vocal Organ (Speaker) " (Fig. 3)

2) The voice of the robot is selected
We were able to find two methods of selecting for the robot. (a) Selecting from a set of voices and (b) Generating sound using templates on-demand.
We selected (a) because it is the simplest method observer can judge that our robot has capability of music recognition. Our strategy: first, generate typical variation of expression in advance. Second, select them according to predicted beat. Therefore, we replace "Expression Planning" (Fig. 2) with "Voice Selection"(Fig. 3)

3) The wave form of self-generated sound is known
Because we decided that the robot would express music using its voice, this assumption is true. In this situation, we can use techniques in echo cancellation problems. This assumption is false when the self-generated sound is not voice, e.g., when the robot is playing an instrument.

4) Only the separated music is used
We do not use separated self-generated sound as a feedback from expression to recognition. This means that we deal with self-generated sound as noise to suppress it. Therefore, the feedback loop from the "Source Separation" to "Conceptualizer" in Fig. 2 was eliminated.

## IV. Music Recognition

Our aim was to recognize the hierarchical beat structure in music. We need a method that can recognizes this from a musical audio signal directly. This is because it is not reasonable to assume that the sounds of musical instruments in a musical piece are well known.

### *A. Real-time Beat Tracking*

*1) Overview:* We used the real-time beat-tracking method proposed by Goto [6]. Fig. 4 provides an overview of real-time beat-tracking system. The method outputs three information about beat structure: (1) predicted next beat time, (2) predicted beat interval and (3) beat type that means the position of the predicted beat in measure level.

Beat tracking system consists of two stages: the frequency analysis stage and the beat prediction stage. In the frequency analysis stage, system obtains onset-time and its reliability using power spectrum of musical audio signal. In the beat prediction stage, multiple agents predict next beat time with different strategy parameters. Reliability of agents are evaluated by checking chord-change and drum-pattern. System selects the most reliable agent, and its prediction is the output of beat tracking system.

*2) Frequency Analysis Stage:* At first, the system obtains the spectrogram of musical audio signal by applying the short time Fourier transform (STFT). STFT is applied with a Hanning window of 4096 [points], a shifting interval of 512 [points] and sampling rate of 44.1 [kHz].

Second, system extracts onset components taking into account factors such as the rapidity of an increase in power. Onset component is defined as below:

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

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

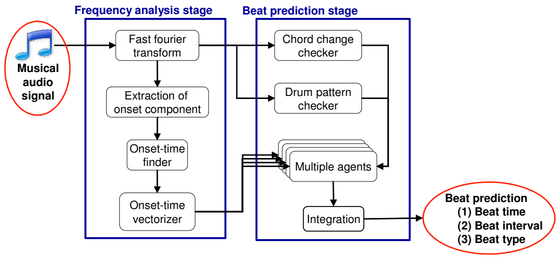

Fig. 4. Overview of Real-time Beat Tracking System.

Here, $d(t,\omega)$ is the onset component, $p(t,\omega)$ is the power of musical audio signal at time frame t and frequency bin $\omega$.

Third, onset-time finder in the system finds onset-time and onset-reliability from onset component $d(t,\omega)$. The onset reliability has seven frequency ranges in each time frame (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, sum of onset component $D_\omega(t) = \sum_\omega d(t,\omega)$ is calculated. Where, $\omega$ is the limited frequency range. The onset times each range are roughly detected by picking the peak of $D_\omega(t)$. If onset time found, its reliability is given by $D_\omega(t)$, otherwise it is set to zero. Finally, onset-time vectorizer in the system vectorizes onset-time reliabilities into onset-time vectors with different sets of frequency weights. The set is one of the parameters of the strategy of agents in multiple agent system.

*3) Beat Prediction Stage:* Multiple agent system predicts beats with different strategies. The strategy consists of three parameters:

1) Frequency focus type:
The parameter defines the set of weights for onset vectorizers. It means the frequency focus of an agent. The value is taken from three types: *all-type*, *law-type* and *mid-type*.
2) Auto-correlation period:
The parameter defines a window size to calculate the vector auto-correlation. The value is taken from two periods: *1000* and *500* [frames].
3) Initial peak selection:
The parameter takes two values: *primary* or *secondary*. If the value is primary, the agent selects the largest peak for beat prediction. Otherwise, the second-largest peak is selected.

Each of multiple agents calculates auto-correlation of onset-time vectors respectively to determine the beat interval. The method assumes that beat interval is between 43 [frames] (120 M.M; Melzel's Metronome) and 85 [frames] (61 M.M). To evaluate reliabilities of agent the system uses two components: (1) the chord-change checker and (2) drum-pattern checker. (1) The chord-change checker slices the spectrogram into stripes at the agent's provisional beat interval. The system assumes that chord-change between stripes is large at the onset-time. (2) The drum-pattern checker has typical drum patterns in advance. First, it finds onset-time of snare and bass drums. Next, it compares drum pattern and onset-time of drums. An agent's reliability increases if its provincial beat interval is consistent with chord-change or drum-pattern.

Beat predictions of the system are obtained by integrating multiple agents. Integration is achieved by selecting the agent that has the highest reliability.

## V. Music Expression

### A. Design of Vocal Content

We used four vocal-content items of "one, two, three, four" to express the musical-beat structure. Each number describes the position of the beat in a measure. By this expression, people can identify that the robot recognizes music in the measure level. The vocal content was recorded in advance with sampling frequency 16 [kHz]. We changed the speed of the vocal content to express the musical tempo. We slowed down the voice speed when musical tempo was slow and speed it up when it was fast. We used STRAIGHT to naturally synthesize different voice speeds [9]. We synthesized two kinds of speeds: half and twice the speed. We achieved musical tempo expression by selecting the speed based on the predicted beat interval.

### B. Control of Vocal Timing

The timing of a robot's voice is basically consistent with the predicted beat time that is fed from real-time beat tracking. However, true timing depends on the characteristics of vocal content, e.g., accent. Therefore, we have to control the timing of the voice based on vocal content. We adopted the onset-detecting algorithm used for real-time beat tracking described Eqs. (1) and (2). To apply the algorithm, there is a problem that multiple onset is detected because whole peaks of onset component is assumed the onset. To solve this problem, we selected the first onset whose reliability was more than threshold $\theta$. Here, we used $\theta = 0.5$. In this way, we can find the onset time more accurately than just by calculating the power spectrum and taking its peak.

## VI. Suppressing Self-generated Sound

### A. ICA based Adaptive Filter

We used the ICA based adaptive filter [7] because we can assume that the waveform of self-generated sound is known. The reason why this assumption is true is that robot expresses music with only counting voice. Therefore, this is similar to the echo canceling problem. A typical solution for echo cancellation is using a Normalized Least Mean Square (NLMS) filter [10]. However, the NLMS filter does not solve our problem. It needs a double-talk detector to sense noise sections and stop updating filter coefficients while there is noise, because NLMS is not robust against noise. As noise was music in this study, it existed in on all sections. In contrast, a ICA based adaptive filter [7] is double-talk free because it has a nonlinear function in its learning rule. Thus, even if noise power is high, estimation error reflected filter coefficients is saturated by the nonlinear function. We will explain the principle underlying the ICA based adaptive filter in the following subsections.

*1) Modeling of Mixing and Unmixing Process:* We used the time-frequency (T-F) model proposed by Takeda *et al.* [7]. The reasons for this was that it would be easy to integrate with other source separation methods in future, such as microphone-array processing.

All signals in the time domain were analyzed by STFT with a window of size $T$, and shift $U$. We assumed that the original source spectrum $S(\omega,f)$ at time frame $f$ and frequency $\omega$ would affect the succeeding $M$ frames of observed sound. Thus, $S(\omega,f-1),S(\omega,f-2),\cdots,S(\omega,f-M)$ were treated as virtual sound sources. The observed spectrum $X(\omega,f)$ at the microphone is expressed as ,

$$X(\omega,f) \quad = \quad N(\omega,f)+\sum_{m=0}^{M} H(\omega,m)S(\omega,f-m), \quad (3)$$

where $N(\omega,f)$ is the noise spectrum and $H(\omega,m)$ is the $m$th delay's transfer function in the T-F domain.

The unmixing process for ICA separation is represented as:

$$\begin{pmatrix} \hat{N}(\omega,f) \\ \mathbf{S}(\omega,f) \end{pmatrix} = \begin{pmatrix} 1 & -\mathbf{w}^T(\omega) \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \begin{pmatrix} X(\omega,f) \\ \mathbf{S}(\omega,f) \end{pmatrix}, \quad (4)$$

$$\mathbf{S}(\omega,f) = [S(\omega,f),S(\omega,f\text{–}1),\ldots,S(\omega,f\text{–}M)]^T, \quad (5)$$

$$\mathbf{w}(\omega) = [w_0(\omega),w_1(\omega),\ldots,w_M(\omega)]^T, \quad (6)$$

where $\mathbf{S}$ is a source spectrum vector and $\hat{N}(\omega,f)$ is an estimated noise spectrum. $\mathbf{w}$ is an unmixing filter vector. Therefore, the unmixing process is described as a linear system with ICA.

*2) Online Learning Algorithm for Unmixing Filter Vector:* An algorithm based on minimizing Kullback-Leibler divergence (KLD) is commonly used to estimate the unmixing filter, $\mathbf{w}(\omega)$, in Eq. (4). Based on KLD, we applied the following iterative equations with non-holonomic constraints [11] to our model because of their fast convergence,

$$\mathbf{w}(\omega,f\text{+}1) = \mathbf{w}(\omega,f)+\mu_1\phi_{\hat{N}(\omega)}\left(\hat{N}(\omega,f)\right)\bar{\mathbf{S}}(\omega,f), \quad (7)$$

$$\phi_x(x) = -\frac{\mathrm{d}\log p_x(x)}{\mathrm{d}x}, \quad (8)$$

where $\mu_1$ is a step-size parameter that controls the speed of convergence, and $\bar{y}$ represents the conjugate of $y$. $p_x(x)$ is defined as the probability distribution of $x$.

The online algorithms for the ICA based adaptive filter are summarized as follows ($\omega$ has been omitted for the sake of readability),

$$\hat{N}(f) = Y(f)-\mathbf{S}(f)^T\mathbf{w}(f), \quad (9)$$

$$\hat{N}_n(f) = \alpha(f)\hat{N}(f), \quad (10)$$

$$\mathbf{w}(f\text{+}1) = \mathbf{w}(f)+\mu_1\phi_{N_n}(\hat{N}_n(f))\bar{\mathbf{S}}_\mathbf{n}(f), \quad (11)$$

$$\alpha(f\text{+}1) = \alpha(f)+\mu_2[1-\phi_{N_n}(\hat{N}_n(f))\bar{\hat{N}}_n(f)]\alpha(f), \quad (12)$$

where $\alpha(f)$ is the positive normalizing factor of $\hat{N}$. $\phi(x)=\tanh(|x|)e^{j\theta(x)}$ is often used for a normalized super-Gaussian distribution such as a speech signal [12].

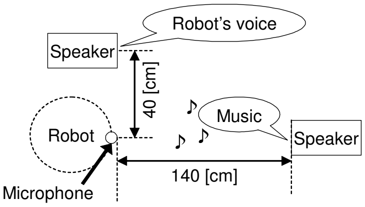

Fig. 5. Set up for sound sources and microphone

## VII. Experiments

We evaluated our system in real environment by comparing predicted beat intervals by suppressing and not suppressing self-generated sound.

### A. Conditions

We used Robovie-R2 which has a one-channel microphone on its nose. To prepare a 3-min input musical audio signal, we selected three songs (No. 52, No. 94 and No. 56) from th RWC music database (RWC-MDB-P-2001) developed by Goto *et al.* [13]. We used 1 minute respectively. These included vocals and instruments. These three pieces had different tempos of, 70, 81 and 75[bpm]. We could evaluate the tracking performance when the musical tempo changed.

Fig. 5 outlines the setup for the experiment. Distance between the microphone and the speaker which plays the robot's voice is 40 [cm] and the microphone and the speaker which plays the music is 140 [cm], respectively.

We experimented under two conditions to evaluate what effect suppressing self-generated sound would have.

1) Periodic counting: Count the beats according to the prediction and
2) Non-periodic counting: Count the beats at random intervals.

### B. Results and Discussion

*1) Periodic Counting:* Fig. 6 plots the results. At the beginning of the first and third songs, beat prediction fails because the robot's voice did not suppressed. Thus, this confirmed that the robot's voice interfered to music recognizing on beat prediction and suppressing robot's voice can improve beat prediction.

At the beginning of the second song, it took about 10 [sec] to adjust the beat interval. The reason for this is the latency, until the appropriate agent in real-time beat tracking changes become reliable. Suppressing self-generated sound will not reduce this latency, so we need to improve real-time beat tracking itself to deal with this problem.

*2) Non-periodic Counting:* According to the results in Fig. 7, beat prediction failed three times without the robot's voice begin suppressed. In contrast, when it was suppressed, the stability of beat prediction was improved. However, the difference the between predicted and correct intervals is larger than that between periodic voice and it. We think that this phenomenon is caused by remnant components of robot's voice which the adaptive filter could not suppress.

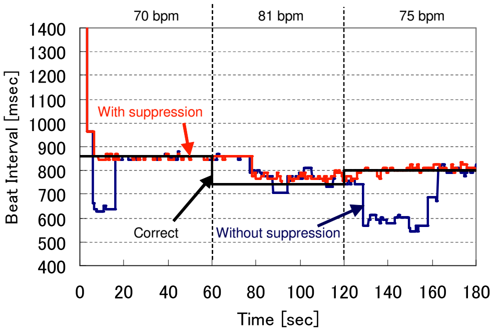

Fig. 6. Predicted beat interval with periodic counting voice

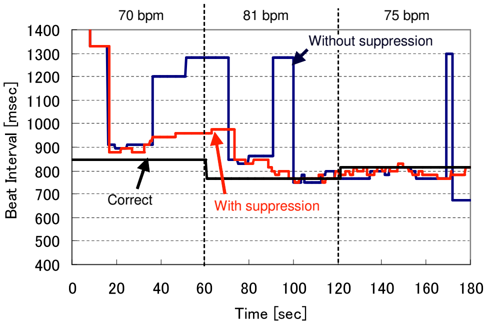

Fig. 7. Predicted beat interval with non-periodic counting voice

According to our architecture (Fig. 3), we know when robot counts the beat accurately. Therefore, it is possible to solve this problem by masking the spectrogram in beat-tracking system when robot is counting.

*3) offline evaluation:* In this experiment, we evaluated only the capability of music recognition, and the capability of music expression was not considered. We evaluated only music recognition in two reasons: (1) our main issue is suppressing robot's own counting voice so the evaluation of the capability of music recognition is the most important, (2) our beat-counting expression is preliminary in two reasons: (a) expression is simple. Although the beat-counting expression have a structure and capable of changing its speed, there is essentially just one pattern. (b) The timing of counting voice is heavily depend on the result of music recognition although it is adjusted using onset in advance. Therefore, to evaluate the capability of music expression, we need to improve expression, for example, singing or dancing.

## VIII. Conclusion

Our aim was to achieve a robot that could understand music. The capability to understood music involves two capabilities: *its recognition* and *expression*. We designed an architecture for a robot that could understand music and developed a robot that could count beats according to our architecture. We pointed out the inevitable problem that self-generated sounds mix into music, and solved it by a ICA based adaptive filter. The experimental results indicated suppressing the robot's voice reduced the beat prediction error regardless of periodic or non-periodic voice. However, our method had less effect on non-periodic counting. To improve this, we need to deal with not only mixed sounds, but also separated music.

In future work, we intend to improve the music expression capability of the robot to extend its appeal. For example, singing a song with listening to music or expressing by motion behavior. To achieve singing, we need to align music score with beats in measure level more strictly. Moreover, predicting basic frequency will needed to sing in appropriate pitch. Expressing motion behaviors is achieved by Yoshii *et al.* in quarter-note level. To extend it to higher level, it is necessary to prepare motion pattern and align it to music. We also intend to suppress self-generating sound in case its waveform is unknown. If it is achieved, robots will be able to play instruments, or dance with active motion.

When the improved expression is achieved, we will be able to evaluate the music expression capability. For example, interaction with a human, rating of a human or Turing Test.

## IX. Acknowledgments

We would like to thank Toru Takahashi for helpful comments about STRAIGHT.

## References

[1] A. Turing. Computing machinery and intelligence. *Mind*, LIX(235):433–460, Oct. 1950.

[2] H. Kozima and M. P. Michalowski. Rhythmic synchrony for attractive human-robot interaction. In *Proc. of Entertainment Computing*, 2007.

[3] S. Kotosaka and S. Shaal. Synchronized robot durumming by neural oscillator. *Journal of Robotics Society of Japan*, 19(1):116–123, 2001.

[4] K. Yoshii, K. Nakadai, T. Torii, Y. Hasegawa, H. Tsujino, K. Komatani, T. Ogata, and H. G. Okuno. A biped robot that keeps steps in time with musical beats while listening to music with its own ears. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2007)*, pages 1743–1750, 2007.

[5] K. Murata, K. Yoshii, H. G. Okuno, T. Torii, K. Nakadai, and Y. Hasegawa. Assessment of a beat-tracking robot for music countaminated by periodic self noises. In *SI2007*, pages 1258–1259, 2007.

[6] M. Goto. An audio-based real-time beat tracking system for music with or without drum-sounds. *Journal of New Music Research*, 30(2):159–171, Jun. 2001.

[7] R. Takeda, K. Nakadai, K. Komatani, T. Ogata, and H. G. Okuno. Robot audition with adaptive filter based on independent component analysis. In *Proc. of the 25th Annual Conference of the Robotics Society of Japan (in Japanese).*, page 1N16, 2007.

[8] W. J. M. Levelt. *Speaking: From Intention to Articulation.* ACL-MIT Press Series in Natural Language Processing. 1989.

[9] H. Kawahara. STRAIGHT, exploration of the other aspect of vocoder: Perceptually ismorphic decompositon of speech sounds. *Acoustic Science and Technology*, 27(6):349–353, 2006.

[10] S. Haykin. *Adaptive filter theory.* Prentice Hall, Englenwood Cliffs, 4th edition, 2001.

[11] S. Choi, S. Amari, A. Cichocki, and R. Liu. Natural gradient learning with a nonholonomic constraint for blind deconvolution of multiple channels. In *Proc. of. International Workshop on ICA and BSS*, pages 371–376, 1999.

[12] H. Sawada, R. Mukai, and S. Araki. Polar coordinate based nonlinear function for frequency-domain blind source separation. *IEICE Trans. Fundamentals*, 86(3):590–596, Mar. 2003.

[13] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka. RWC music database : Popular music database and royalty-free music database. In *IPSJ SIG Notes*, volume 2001, pages 35–42, 2001.