Music-Ensemble Robot That Is Capable of Playing the Theremin While Listening to the Accompanied Music

Takuma Otsuka[1], Takeshi Mizumoto[1], Kazuhiro Nakadai[2], Toru Takahashi[1], Kazunori Komatani[1], Tetsuya Ogata[1], and Hiroshi G. Okuno[1]

[1] Graduate School of Informatics, Kyoto University, Kyoto, Japan
{ohtsuka,mizumoto,tall,komatani,ogata,okuno}@kuis.kyoto-u.ac.jp
[2] Honda Research Institute Japan, Co., Ltd., Saitama, Japan
nakadai@jp.honda-ri.com

Abstract. Our goal is to achieve a musical ensemble among a robot and human musicians where the robot listens to the music with its own microphones. The main issues are (1) robust beat-tracking since the robot hears its own generated sounds in addition to the accompanied music, and (2) robust synchronizing its performance with the accompanied music even if humans' musical performance fluctuates. This paper presents a music-ensemble Thereminist robot implemented on the humanoid HRP-2 with the following three functions: (1) self-generated Theremin sound suppression by semi-blind Independent Component Analysis, (2) beat tracking robust against tempo fluctuation in humans' performance, and (3) feedforward control of Theremin pitch. Experimental results with a human drummer show the capability of this robot for the adaptation to the temporal fluctuation in his performance.

Index Terms: Music robot, Musical human-robot interaction, Beat tracking, Theremin.

1 Introduction

To realize a joyful human-robot interaction and make robots more friendly, music is a promising medium for interactions between humans and robots. This is because music has been an essential and common factor in most human cultures. Even people who do not share a language can share a friendly and joyful time through music although natural communications by other means are difficult. Therefore, *music robots* that can interact with humans through music are expected to play an important role in natural and successful human-robot interactions.

Our goal is to achieve a musical human-robot ensemble, or *music-ensemble robot*, by using the robot's microphones instead of using symbolic musical representations such as MIDI signals. "Hearing" music directly with the robot's "ears", i.e. the microphones, like humans do is important for naturalness in the musical interaction because it enables us to share the acoustic sensation.

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 102–112, 2010.

We envision a robot capable of playing a musical instrument and of synchronizing its performance with the human's accompaniment. However, the difficulty resides in that human's musical performance often includes many kinds of fluctuations and variety of music sounds. For example, we play a musical instrument with a temporal fluctuation, or when we sing a song, the pitch often vibrates.

Several music ensemble robots have been presented in a robotics field. However, the ensemble capability of these robots remains immature in some ways. A. Alford *et al.* developed a robot that plays the theremin [1], however, this robot is intended to play the theremin without any accompaniments. Petersen *et al.* reported a musical ensemble between a flutist robot, WF-4RIV, and a human saxophone player [2]. However, this robot takes turns with the human playing musical phrases. Therefore, the musical ensemble in a sense of performing simultaneously is yet to be achieved. Weinberg *et al.* developed a percussionist robot called "Haile" that improvises its drum with human drum players [3],[4]. The adaptiveness in this robot to the variety of human's performance is limited. For example, this robot allows for little tempo fluctuation in the human performance, or it assumes a specific musical instrument to listen to.

This paper presents a robot capable of music ensemble with a human musician by playing the electric musical instrument called theremin [5]. This robot plays the theremin while it listens to the music and estimates the tempo of the human accompaniment by adaptive beat tracking method. The experiment confirms our robot's adaptiveness to the tempo fluctuation by the live accompaniment.

2 Beat Tracking-Based Theremin Playing Robot

2.1 Essential Functions for Music Robots

Three functions are essential to the envisioned music ensemble robot: (1) listening to the music, (2) synchronizing its performance with the music, and (3) expressing the music in accordance with the synchronization.

The first function works as a preprocessing of the music signal such that the following synchronization with the music is facilitated. Especially, self-generated sound such as robot's own voice that is mixed into the music sound robot hears has been revealed to affect the quality of subsequent synchronization algorithm [6][7].

The second function extracts some information necessary for the robot's accompaniment to the human's performance. The tempo, the speed of the music, and the time of beat onsets, where you step or count the music, are the most important pieces of information to achieve a musical ensemble. Furthermore, the robot has to be able to predict the coming beat onset times for natural synchronization because it takes the robot some time to move its body to play the musical instrument.

The third function determines the behavior of the robot based on the output of the preceding synchronization process. In the case of playing the musical instrument, the motion is generated so that the robot can play a desired phrase at the predicted time.

2.2 System Architecture

Figure 1 outlines our beat tracking-based theremin player robot. This robot has three functions introduced in Section 2.1. The robot acquires a mixture sound of human's music signal and the theremin sound and plays the theremin in synchronization with the input music.

For the first listening function, independent component analysis (ICA)-based self-generated sound suppression [8] is applied to suppress the theremin sound from the sound that the robot hears. The inputs for this suppression method are the mixture sound and the clean signal of the theremin. The clean signal from the theremin is easily acquired because theremin directly generates an electric waveform.

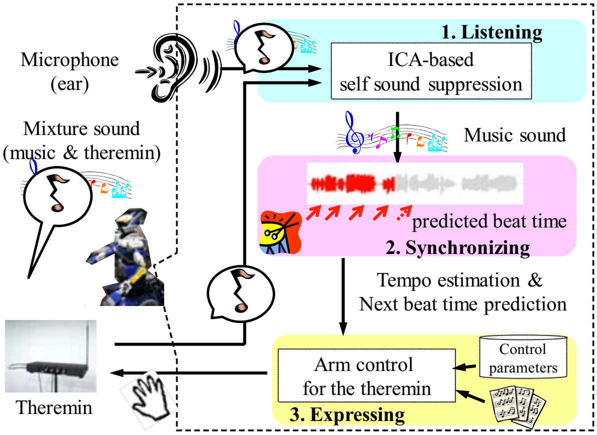

Fig. 1. The architecture of our theremin ensemble robot

The second synchronization function is a tempo-adaptive beat tracking called spectro-temporal pattern matching [7]. The algorithm consists of tempo estimation, beat detection, and beat time prediction. Some formulations are introduced in Section 3.1.

The third expression function is a the pitch and timing control of the theremin. A theremin has two antennae: vertical one for pitch control and horizontal one for volume control. The most important feature of a theremin is a **proximity control**: without touching it, we can control a theremin's pitch and volume [5]. As a robot's arm gets closer to the pitch-control antenna, the theremin's pitch increases monotonically and nonlinearly. The robot plays a given musical score in synchronization with the human's accompaniment.

3 Algorithm

3.1 Beat Tracking Based on Spectro-temporal Pattern Matching

This beat tracking algorithm has three phases: (1) tempo estimation, (2) beat detection, and (3) beat time prediction. The input is a music signal of human performance after the self-generated sound suppression.

Tempo estimation. Let $P(t, f)$ be the mel-scale power spectrogram of the given music signal where t is time index and f is mel-filter bank bin. We use 64 banks, therefore $f = 0, 1, ..., 63$. Then, Sobel filtering is applied to $P(t, f)$ and the onset belief $d_{inc}(t, f)$ is derived.

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

$$\begin{aligned} d(t,f) = &-P(t-1,f+1) + P(t+1,f+1) \\ &-2P(t-1,f) + 2P(t+1,f) \\ &-P(t-1,f-1) + P(t+1,f-1), \end{aligned} \tag{2}$$

where $f = 1, 2, ..., 62$. Equation (2) shows the Sobel filter.

The tempo is defined as the interval of two neighboring beats. This is estimated through Normalized Cross Correlation (NCC) as Eq. (3).

$$R(t,i) = \frac{\sum_{f=1}^{62}\sum_{k=0}^{W-1} d_{inc}(t-k,f)d_{inc}(t-i-k,f)}{\sqrt{\sum_{f=1}^{62}\sum_{k=0}^{W-1} d_{inc}(t-k,f)^2 \cdot \sum_{f=1}^{62}\sum_{k=0}^{W-1} d_{inc}(t-i-k,f)^2}} \tag{3}$$

where W is a window length for tempo estimation and i is a shift offset. W is set to be 3 [sec]. To stabilize the tempo estimation, the local peak of $R(t, i)$ is derived as

$$R_p(t,i) = \begin{cases} R(t,i) \text{ if } \ R(t,i-1) < R(t,i) \text{ and } R(t,i+1) < R(t,i) \\ 0 \qquad\qquad\qquad\qquad \text{otherwise} \end{cases} \tag{4}$$

For each time t, the beat interval $I(t)$ is determined based on $R_p(t, i)$ in Eq. (4). The beat interval is an inverse number of the musical tempo. Basically, $I(t)$ is chosen as $I(t) = \underset{i}{\operatorname{argmax}} R_p(t, i)$. However, when a complicated drum pattern is performed in the music signal, the estimated tempo will fluctuate rapidly.

To avoid the mis-estimation of the beat interval, $I(t)$ is derived as Eq. (5). Let I_1 and I_2 be the first and the second peak in $R_p(t, i)$ when moving i.

$$I(t) = \begin{cases} 2\|I_1 - I_2\| \text{ if } (\|I_{n2} - I_1\| < \delta \ \text{ or } \|I_{n2} - I_2\| < \delta) \\ 3\|I_1 - I_2\| \text{ if } (\|I_{n3} - I_1\| < \delta \ \text{ or } \|I_{n3} - I_2\| < \delta) \\ I_1 \qquad\qquad\qquad\qquad \text{otherwise,} \end{cases} \tag{5}$$

where $I_{n2} = 2\|I_1 - I_2\|$ and $I_{n3} = 3\|I_1 - I_2\|$. δ is an error margin parameter.

The beat interval $I(t)$ is confined to the range between 61 – 120 beats per minute (bpm). This is because this range is suitable for the robot's arm control.

Beat detection. Each beat time is estimated using the onset belief $d_{inc}(t, f)$ and the beat interval $I(t)$. Two kinds of beat reliabilities are defined: neighboring beat reliability and continuous beat reliability. Neighboring beat reliability $S_n(t, i)$ defined as Eq. (6) is a belief that the adjacent beat lies at $I(t)$ interval.

$$S_n(t,i) = \begin{cases} \sum_{f=1}^{62} d_{inc}(t-i,f) + \sum_{f=1}^{62} d_{inc}(t-i-I(t),f) & \text{if } (i \le I(t)), \\ 0 & \text{if } (i > I(t)) \end{cases} \tag{6}$$

Continuous beat reliability $S_c(t, i)$ defined as Eq. (7) is a belief that the sequence of musical beats lies at the estimated beat intervals.

$$S_c(t,i) = \sum_{m=0}^{N_{beats}} S_n(T_p(t,m), i), \tag{7}$$

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

where $T_p(t, m)$ is the m-th previous beat time at time t, and N_{beats} is the number of beats to calculate the Continuous beat reliability.

Then, these two reliabilities are integrated into beat reliability $S(t)$ as

$$S(t) = \sum_i S_n(t-i,i) \cdot S_c(t-i,i). \tag{8}$$

The latest beat time $T(n + 1)$ is one of the peak in $S(t)$ that is closest to $T(n) + I(t)$, where $T(n)$ the n-th beat time.

Beat time prediction. Predicted beat time T' is obtained by extrapolation using the latest beat time $T(n)$ and the current beat interval $I(t)$.

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

$$T_{tmp} = T(n) + I(t) + (t - T(n)) - \{(t - T(m)) \mod I(t)\} \tag{10}$$

3.2 Theremin Pitch Control by Regression Parameter Estimation

We have proposed a theremin's model-based feedforward pitch control method for a thereminist robot in our previous work [9]. We introduce our method in the following order: model formulation, parameter estimation and feedforward arm control.

Arm-position to pitch model. We constructed a model that represents a relationship between a theremin's pitch and a robot's arm position. According to the fact that a theremin's pitch increases monotonically and nonlinearly, we formulated our model as follows:

$$\hat{p} = M_p(x_p; \boldsymbol{\theta}) = \frac{\theta_2}{(\theta_0 - x_p)^{\theta_1}} + \theta_3 \tag{11}$$

where, $M_p(x_p; \boldsymbol{\theta})$ denotes our pitch model, x_p denotes a pitch-control arm, $\boldsymbol{\theta} = (\theta_0, \theta_1, \theta_2, \theta_3)$ denotes model parameters, $\hat{p}$ denotes an estimated pitch using a pitch model ([Hz]).

Parameter estimation for theremin pitch control. To estimate model parameters, $\boldsymbol{\theta}$, we obtain a set of learning data as following procedure: at first, we equally divide a range of robot's arm into N pieces (we set $N = 15$). For each boundary of divided pieces, we extract a theremin's corresponding pitch. Then, we can obtain a set of learning data, i.e., pairs of pitch-control arm positions ($x_{pi}, i = 0 \cdots N$) and corresponding theremin's pitches ($p_i, i = 0 \cdots N$). Using the data, we estimate model parameters with Levenberg-Marquardt (LM) method, which is one of a nonlinear optimization method. As an evaluation function, we use a difference between measured pitch, p_i, and estimated pitch, $M_p(x_{pi}, \boldsymbol{\theta})$.

Feedforward arm control. Feedforward arm control has two aspects: arm-position control and timing control. A musical score is prepared for our robot to play the melody. The musical score consists of two elements: the note name that determines the pitch and the note length that relates to the timing control.

To play the musical notes in a correct pitch, a musical score is converted into a sequence of arm position. We first convert musical notes (e.g., $C4$, $D5$, ..., where the number means the octave of each note) into a sequence of corresponding pitches based on equal-temperament:

$$p = 440 \cdot 2^{o-4} \sqrt[12]{2^{n-9}}, \tag{12}$$

where p is the pitch for the musical note, o is the octave number. The variable n in Eq. (12) represents the pitch class where $n = 0, 1, ..., 11$ correspond to $C, C\sharp, ..., B$ notes, respectively. Then, we give the pitch sequence to our inverse pitch model:

$$\hat{x_p} = M_p^{-1}(p, \boldsymbol{\theta}) = \theta_0 - \left(\frac{\theta_2}{p - \theta_3}\right)^{1/\theta_1} \tag{13}$$

where, $\hat{x}_p$ denotes an estimated robot's arm position. Finally, we obtain a sequence of target arm positions. By connecting these target positions linearly, the trajectory for a thereminist robot is generated.

The timing of each note onset, the beginning of a musical note, is controlled using the predicted beat time T' in Eq. (9) and the current beat interval $I(t)$ in Eq. (5). When T' and $I(t)$ are updated, the arm controller adjusts the timing such that the next beat comes at time T', and the time duration of each note is calculated by multiplying relative note length such as quarter notes by $I(t)$.

4 Experimental Evaluation

This section presents the experimental results of our beat tracking-based thereminist robot. Our experiments consist of two parts. The first experiment proves our robot's capability of quick adaptation to tempo change and robustness against the variety of musical instruments. The second experiment shows that our robot is able to play the theremin with a little error even when fluctuations in human's performance are observed.

4.1 Implementation on Humanoid Robot HRP-2

We implemented our system on a humanoid robot HRP-2 in Fig. 3 [10]. Our system consists of two PCs. The ICA-based self-generated sound suppression and the beat tracking system is implemented by C++ on MacOSX. The arm control for the theremin performance is implemented by Python on Linux Ubuntu 8.04. The predicted beat time T' and the beat interval $I(t)$ are sent to the arm controller through socket communication at time $T' - \Delta t$, where Δt is the delay in the arm control. Δt is set 40 [msec] empirically.

The settings for the beat tracking is as follows: the sampling rate is 44100 [Hz], the window size for fast Fourier transform is 4096 [pt], and the hop size of the window is 512 [pt]. For the acoustic input, a monaural microphone is attached to the HRP-2's head as indicated in Fig. 3.

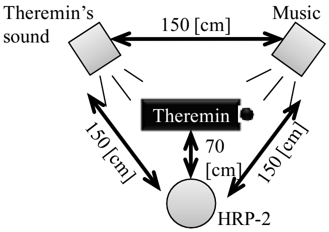

Fig. 2. Experimental setup

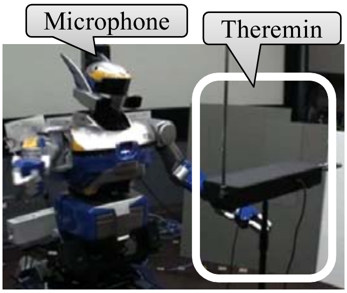

Fig. 3. Humanoid robot HRP-2

4.2 Experiment 1: The Influence of the Theremin on Beat Tracking

Figure 2 shows the experimental setup for the experiment 1. The aim of this experiment is to reveal the influence of theremin's sound on the beat tracking algorithm. Music sound comes out of the right loudspeaker while the robot is playing the theremin and its sound comes out of the left loudspeaker.

The music signal used in the experiment is three minutes long that is the excerpts from three popular music songs in RWC music database (RWC-MDB-P-2001) developed by Goto *et al* [11]. These three songs are No. 11, No. 18,

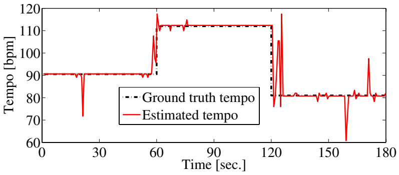

Fig. 4. Tempo estimation result w/ self-generated sound suppression

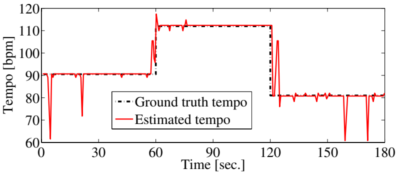

Fig. 5. Tempo estimation result w/o self-generated sound suppression

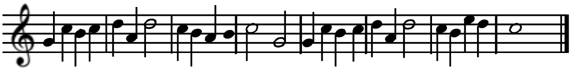

Fig. 6. The musical score of Aura Lee

No. 62. The tempo for each song is 90, 112, 81 [bpm], respectively. One-minute excerpts are concatenated to make the three-minute music signal.

Figure 4 and 5 are the tempo estimation results. The self-generated sound suppression is active in Fig. 4 while it is disabled in Fig. 5. The black line shows the ground truth tempo, and the red line shows the estimated tempo.

These results prove prompt adaptation to the tempo change and robustness against the variety of the musical instruments used in these music tunes. On the other hand, a little influence of the theremin sound on the beat tracking algorithm is observed. This is because theremin's sound does not have impulsive characteristics that mainly affect the beat tracking results. Though the sound of theremin has little influence on the beat tracking, self-generated sound suppression is generally necessary.

4.3 Experiment 2: Theremin Ensemble with a Human Drummer

In this experiment, a human drummer stands in the position of the right loudspeaker in Fig. 2. At first, the drummer beats the drum slowly, then he hastes the drum beating. The robot plays the first part of "Aura Lee," American folk song. The musical score is shown in Fig. 6.

Figure 7 and 8 show the ensemble of the thereminist robot and the human drummer. Top plots indicate the tempo of human's drumming and estimated tempo by the system. Middle plots are the theremin's pitch trajectory in a red line and human's drum-beat timings in black dotted lines. The bottom plots show the onset error between human's drum onsets and the theremin's note onsets. Positive error means the theremin onset is earlier than the drum onset. The pitch trajectories of the theremin are rounded off to the closest musical note on a logarithmic frequency axis.

The top tempo trajectory shows that the robot successfully tracked the tempo fluctuation in the human's performance whether the self-generated sound suppression because the tempo, or the beat interval, is estimated after a beat is

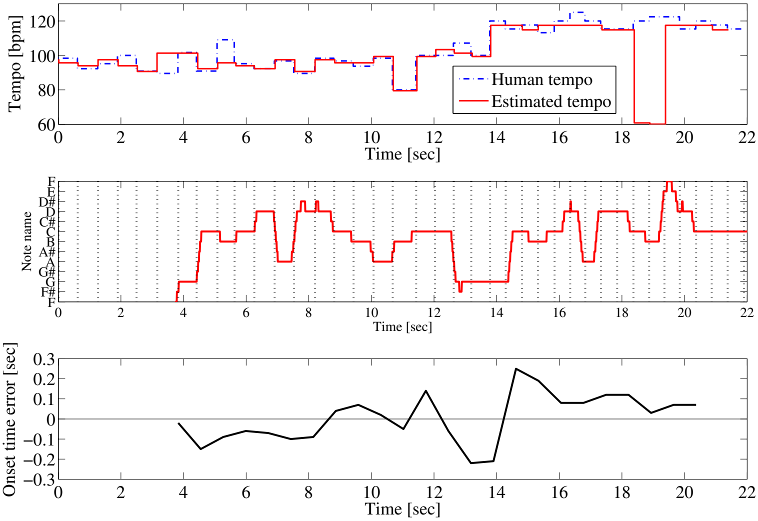

Fig. 7. Theremin ensemble with human drum w/ self-generated sound suppression Top: tempo trajectory, Mid: theremin pitch trajectory, Bottom: Onset time error

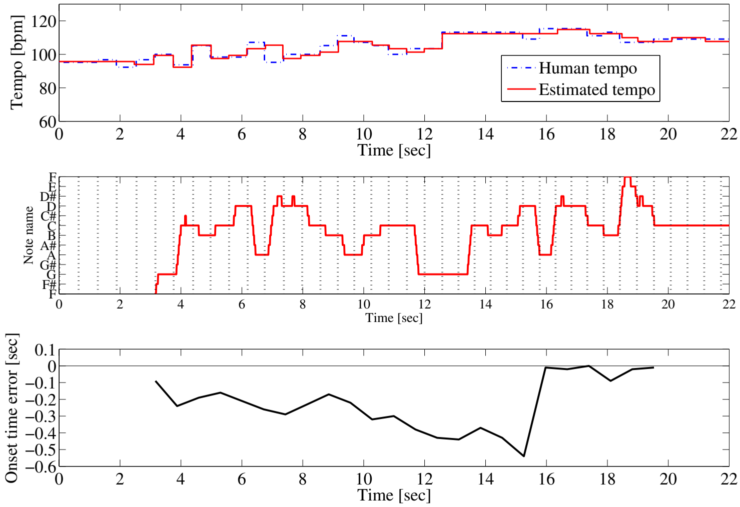

Fig. 8. Theremin ensemble with human drum w/o self-generated sound suppression Top: tempo trajectory, Mid: theremin pitch trajectory, Bottom: Onset time error

observed. However, some error was observed between the human drum onsets and theremin pitch onsets especially around 13 [sec], where the human player hastes the tempo. The error was then relaxed from 16 [sec], about 6 beat onsets after the tempo change.

The error at the bottom plot of Fig. 8 first gradually increased toward a negative value. This is because the human drummer hasted its performance gradually, therefore, the robot did not catch up the speed and produced an increasing negative error value. The error at the bottom plot of Fig. 7 went zigzag because both the human and the robot tried to synchronize their own performance with the other's. The mean and standard deviation of the error for Fig. 7 and 8 were 6.7 ± 120.2 [msec] and -225.8 ± 157.0 [msec], respectively. It took 3–4 [sec] the robot before it starts playing the theremin because this time is necessary to estimate the tempo with stability.

5 Conclusion

This paper presented a robot capable of playing the theremin with human's accompaniment. This robot has three functions for the ensemble: (1) the ICA-based self-generated sound suppression for the listening function, (2) the beat tracking algorithm for the synchronization function, (3) the arm control to play the theremin in a correct pitch for the expression function.

The experimental results revealed our robot's capability of adaptiveness to the tempo fluctuation and of robustness against the variety of musical instruments. The results also suggest that the synchronization error increases when the human player gradually changes his tempo.

The future works are as follows: First, this robot currently considers the beat in the music. For richer musical interaction, the robot should allow for the pitch information in the human's performance. Audio to score alignment [12] is promising technique to achieve a pitch-based musical ensemble. Second, the ensemble with multiple humans is a challenging task because the synchronization becomes even harder when all members try to adapt to another member. Third, this robot requires some time before it joins the ensemble or the ending of the ensemble is still awkward. To start and conclude the ensemble, quicker adaptation is preferred.

Acknowledgments. A part of this study was supported by Grant-in-Aid for Scientific Research (S) and Global COE Program.

References

1. Alford, A., et al.: A music playing robot. In: FSR 1999, pp. 29–31 (1999)
2. Petersen, K., Solis, J.: Development of a Aural Real-Time Rhythmical and Harmonic Tracking to Enable the Musical Interaction with the Waseda Flutist Robot. In: Proc. of IEEE/RSJ Int'l Conference on Intelligent Robots and Systems (IROS), pp. 2303–2308 (2009)
3. Weinberg, G., Driscoll, S.: Toward Robotic Musicianship. Computer Music Journal 30(4), 28–45 (2006)
4. Weinberg, G., Driscoll, S.: The interactive robotic percussionist: new developments in form, mechanics, perception and interaction design. In: Proc. of the ACM/IEEE Int'l Conf. on Human-robot interaction, pp. 97–104 (2007)

5. Glinsky, A.V.: The Theremin in the Emergence of Electronic Music. PhD thesis, New York University (1992)
6. Mizumoto, T., Takeda, R., Yoshii, K., Komatani, K., Ogata, T., Okuno, H.G.: A Robot Listens to Music and Counts Its Beats Aloud by Separating Music from Counting Voice. In: IROS, pp. 1538–1543 (2008)
7. Murata, K., Nakadai, K., Yoshii, K., Takeda, R., Torii, T., Okuno, H.G., Hasegawa, Y., Tsujino, H.: A Robot Uses Its Own Microphone to Synchronize Its Steps to Musical Beats While Scatting and Singing. In: IROS, pp. 2459–2464 (2008)
8. Takeda, R., Nakadai, K., Komatani, K., Ogata, T., Okuno, H.G.: Barge-in-able Robot Audition Based on ICA and Missing Feature Theory under Semi-Blind Situation. In: IROS, pp. 1718–1723 (2008)
9. Mizumoto, T., Tsujino, H., Takahashi, T., Ogata, T., Okuno, H.G.: Thereminist Robot: Development of a Robot Theremin Player with Feedforward and Feedback Arm Control based on a Theremin's Pitch Model. In: IROS, pp. 2297–2302 (2009)
10. Kaneko, K., Kanehiro, F., Kajita, S., Hirukawa, H., Kawasaki, T., Hirata, M., Akachi, K., Isozumi, T.: Humanoid robot HRP-2. In: Proc. of IEEE Int'l Conference on Robotics and Automation (ICRA), vol. 2, pp. 1083–1090 (2004)
11. Goto, M., Hashiguchi, H., Nishimura, T., Oka, R.: RWC Music Database: Popular Music Database and Royalty-Free Music Database. IPSJ Sig. Notes 2001(103), 35–42 (2001)
12. Dannenberg, R., Raphael, C.: Music Score Alignment and Computer Accompaniment. Communications of the ACM 49(8), 38–43 (2006)