Two-level Synchronization using Particle Filter for Co-player Music Robots

Takuma Otsuka, Kazuhiro Nakadai, Toru Takahashi, Tetsuya Ogata, Hiroshi G. Okuno

***Abstract*— Our goal is to develop a *co-player* music robot, i.e., a robot that presents a musical expression together with humans. A music interaction requires two important functions: synchronization with the music and musical expression, such as dancing or playing a musical instrument. Many instrument-performing robots are only capable of the latter function, they may have difficulty in playing live with human performers. The synchronization function is critical for the interaction. We classify synchronization and musical expression into two levels: (1) the rhythm level and (2) the melody level. Two issues in achieving two-level synchronization and musical expression are: (1) simultaneous estimation of the rhythm structure and the current part of the music and (2) derivation of the estimation confidence to switch behavior between the rhythm level and the melody level. This paper presents a score following algorithm, incremental audio to score alignment, that conforms to the two-level synchronization design using a particle filter. Our method estimates the score position for the melody level and the tempo for the rhythm level. The reliability of the score position estimation is extracted from the probability distribution of the score position. Experiments are carried out using 20 polyphonic jazz songs. The results confirm that our method switches levels to alleviate the error in the score position estimation. When some of the musical notes sound vague or when temporal fluctuations are frequently observed, the estimated score position tends to be erroneous. In the experiment, these errors turn out to be reduced by our two-level synchronization strategy.**

I. Introduction

Music robots capable of, for example, dancing, singing, or playing an instrument with humans will play an important role in the symbiosis between robots and humans. Even people who do not share a language can share a friendly and joyful time through music beyond ages, regions, and races. Music robots can be classified into two categories; *entertainment-oriented robots* like trumpeter robots or dancer robots and *co-player robots* for natural interaction. Although the former type has been studied extensively, our research aims at the latter type, i.e., a robot that presents a musical expression together with humans.

Music robots should be co-players rather than entertainers for human-robot symbiosis and richer musical experiences. Their music interaction requires two important functions; synchronization with the music and generation of musical expressions, such as dancing or playing a musical instrument. Many instrument-performing robots such as those presented in [1], [2] are only capable of the latter function, they may have difficulty in playing live with human performers. The former function is essential to the interaction.

T. Otsuka, T. Takahashi, T. Ogata, H. G. Okuno are with Graduate School of Informatics, Kyoto University, Kyoto, 606-8501, Japan. {ohtsuka, tall, ogata, okuno}@kuis.kyoto-u.ac.jp

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

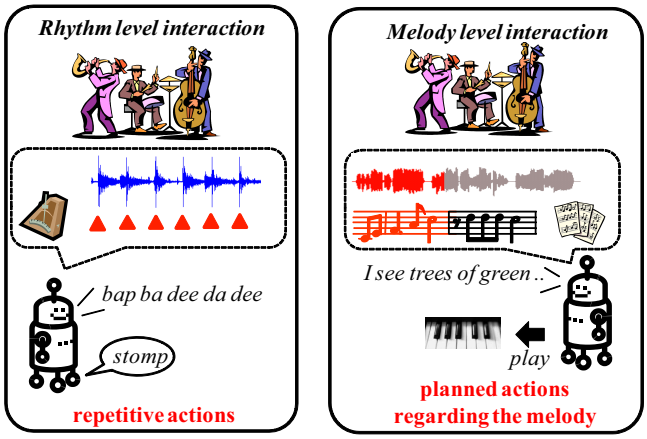

Fig. 1. Two levels in musical interactions

We classify synchronization and musical expression into two levels: (1) *the rhythm level* and (2) *the melody level*. The rhythm level is used when the robot misses what part in a song is being performed, and the melody level is used when the robot is aware of what part is being played. Figure 1 illustrates the two-level synchronization with the music. When we try to synchronize with the song being unaware of the exact part, we can follow the beats imagining a corresponding metronome and stomp our feet, clap our hands or scat to the rhythm. Or, even if we do not know the song or the lyrics to sing, we can still hum the tune. On the other hand, when we know the song and understand which part is being played, we can sing along or dance to a certain choreography. Two issues arise in achieving the two-layer synchronization and musical expression. First, the robot must be able to estimate the rhythm structure and the current part of the music. Second, the robot needs a confidence in how accurately the score position is estimated, hereafter referred to as an estimation confidence, to switch its behavior between the rhythm level and melody level.

Since most conventional music robots have focused on the rhythm level, their musical expressions are limited to repetitive or random expressions such as drumming, shaking their body, stepping, or scatting. A percussionist robot, called *Haile*, developed by Weinberg et al. [3] uses MIDI signals to account for the melody level. However, this approach limits the naturalness of the interaction because live performances with acoustic instruments and singing voices do not have

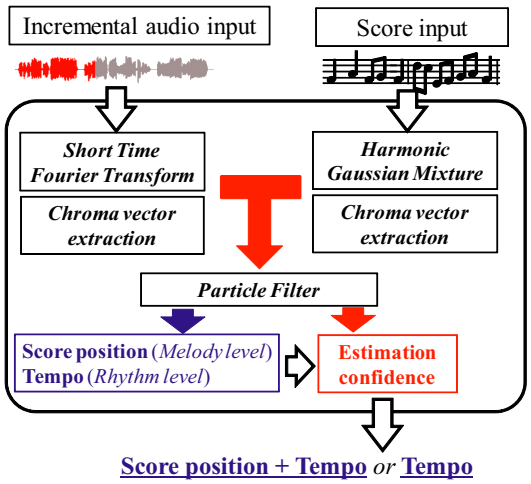

Fig. 2. Two-level synchronization architecture

corresponding MIDI signals. If we stick to MIDI signals, we would have to develop a conversion system that can take any musical audio signal including singing voices and change them into MIDI representations.

An incremental audio to score alignment [4] is introduced for the melody level for the purpose of a robot singer [5], but this method won't work if the robot fails to track the musical score. The important principle in designing a co-player robot is to accept the score follower's errors and to try to recover from them to make ensemble performances more stable.

This paper presents a score following algorithm that conforms to the two-level model using a particle filter [6]. Our method estimates the score position for the melody level and tempo (speed of the music) for the rhythm level. The reliability of the score position estimation is determined from the probability distribution of the score position. Thus, when the estimation of the score position is unreliable, only tempo is reported in order to prevent the robot from performing incorrectly; when the estimation is reliable, it reports the score position.

II. Requirements in Score Following for Music Robots

Music robots have to not only *follow* the music but also *predict* coming musical notes. This is because a music robot cannot present a musical expression without any delay when it detects the current position in the score. For example, Murata *et al.* [7] reports that it takes around 200 (ms) to generate a singing voice using singing voice synthesizer VOCALOID [8]. This is also the case with humans; it takes around 200 (ms) to respond to something one hears. Therefore, a robot for our purpose needs the capability to predict future musical events at least 200 (ms) in advance.

A. State-of-the-art Score Following Systems

Most conventional score following methods are based on either dynamic time warping (DTW) [9] or hidden Markov model (HMM) [10]. The target of these systems are a MIDI-based automatic accompaniments. Since MIDI systems can synthesize audio signals without delay, they only report the current score position without any prediction.

Another score following method [11] uses a hybrid HMM and semi-Markov chain model to predict the duration of each musical note. However, this method reports the most likely score position whether it is reliable or not. Our idea is that using an estimation confidence of the score position to switch between behaviors would make the robot more intelligent in the music interaction.

Our method is an extension of the particle filter-based score following [12] that is apt to fail when the tempo is misestimated. We use a prior tempo information specified by the score to stabilize the tempo estimation.

B. Problem Statement

The problem is specified as follows:

Input: incremental audio signal and the corresponding musical score,
Output: predicted score position, or the tempo
Assumption: the tempo is provided by the musical score with a margin of error.

The issues are (1) simultaneous estimation of the score position and tempo and (2) the design of the estimation confidence. Generally, the tempo given by the score and the actual tempo in the human performance is different partly due to the preference or interpretation of the song, or partly due to the temporal fluctuation in the performance. Therefore, some margin of error should be assumed in the tempo information. In Section IV-B several values of the margin are tested.

We model this simultaneous estimation as a state-space model and obtain the solution with a particle filter. The particle filter approximates the simultaneous distribution of score position and tempo by the density of particles with a state transition model and an observation model. With incremental audio input, the particle filter updates the distribution and estimates the score position and tempo. The reliability is determined from the probability distribution. Figure 2 outlines our method. The particle filter outputs three types of information: the predicted score position, tempo, and estimation confidence. According to the estimation confidence, the system reports either the score position or the tempo.

III. Score Following using Particle Filter

A. Overview of Particle Filter

Let $X_{f,t}$ be the amplitude of the input audio signal in the time frequency domain with frequency bin f and time t, and let k be the score frame. The score is divided into frames such that the length of a quarter note equals to 12 frames to account for the resolution of sixteenth-note and triplets. Musical notes $\mathbf{n}_k = [n_k^1 \ldots n_k^{r_k}]^T$ are placed at frame k, and r_k is the number of musical notes. Each particle p_i has score position, beat interval, and weight: $p_i = (\hat{k}_i, \hat{b}_i, w_i)$, and N is the number of particles, i.e., $1 \leq i \leq N$. The unit for $\hat{k}_i$ is a beat (the quarter note position), and the unit for $\hat{b}_i$ is

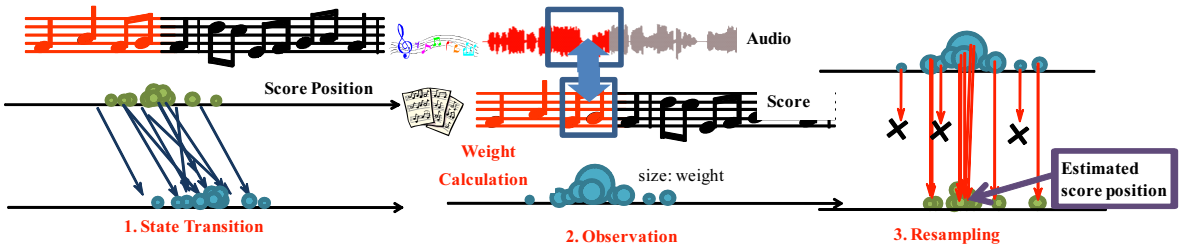

Fig. 3. Overview of the Score Following using Particle Filter

seconds per a beat. Although the actual score position k is a discrete in steps of $1/12$, the value held by the particles $\hat{k}_i$ is continuous.

At every ΔT time, the following procedure is carried out: (1) state transition, (2) observation, (3) resampling, and then estimation of the tempo and the score position. Figure 3 illustrates these steps. The size of each particle represents its weight. After the resampling step, the weights of all particles are set to be equal. Each procedure is described in the following subsections.

B. State Transition Model

The beat interval is sampled from the proposal distribution $q(b|\mathbf{X}_t,\tilde{b}^s)$ that consists of normalized cross correlation of an audio spectrogram $\mathbf{X}_t$ and the window function derived from the tempo $\tilde{b}^s$ provided by the musical score.

$$\hat{b}_i \sim q(b|\mathbf{X}_t,\tilde{b}^s), \quad (1)$$

$$q(b|\mathbf{X}_t,\tilde{b}^s) \propto R(b,\mathbf{X}_t)\times\psi(b|\tilde{b}^s). \quad (2)$$

The audio spectrogram is denoted by $\mathbf{X}_t=[X_{f,\tau}]$, where $t-L<\tau\le t$ and L denotes the window length of the spectrogram. The normalized cross correlation is defined as

$$R(b,\mathbf{X}_t) = \frac{\sum_{\tau=t-L}^{t}\sum_f X_{f,\tau}X_{f,\tau-b}}{\sqrt{\sum_{\tau=t-L}^{t}\sum_f X_{f,\tau}^2\sum_{\tau=t-L}^{t}\sum_f X_{f,\tau-b}^2}}. \quad (3)$$

The window function is centered at $\tilde{b}^s$ that is the tempo specified by the musical score.

$$\psi(b|\tilde{b}^s) = \begin{cases} 1 & |60/b-60/\tilde{b}^s|<W \\ 0 & \text{otherwise} \end{cases}, \quad (4)$$

where W is the width of the window in beats per minute (bpm). A beat interval b (sec) is converted into a tempo value m (bpm) by the equation $m=60/b$. Eq. (4) limits the beat interval value of particles so as not to miss the score position by a falut tempo estimation. The score position is sampled from the normal distribution whose mean value is obtained by adding an offset corresponding to the beat interval $\hat{b}_i$ to the previous score position.

$$\hat{k}_i \sim \mathcal{N}(k|\hat{k}_i^{old}+\Delta T/\hat{b}_i,\sigma_k^2), \quad (5)$$

where $\hat{k}_i^{old}$ is the previous score position, and the variance σ_k^2 is empirically set to 1.

State transition probabilities are defined as follows:

$$p(\hat{b}_i,\hat{k}_i|\hat{b}_i^{old},\hat{k}_i^{old}) = \mathcal{N}(\hat{b}_i|\hat{b}_i^{old},\sigma_b^2)\times\mathcal{N}(\hat{k}_i|\hat{k}_i^{old}+\Delta T/\hat{b}_i,\sigma_k^2), \quad (6)$$

where the variance for the beat interval transition σ_b^2 is empirically set to 0.2. These probabilities are used for the weight calculation in Eq. (7).

C. Observation Model and Weight Calculation

At time t, a spectrogram $\mathbf{X}_t=[X_{f,\tau}](t-L<\tau\le t)$ is used for the weight calculation. The weight of each particle $w_i, 1\le i\le N$ is calculated as:

$$w_i = \frac{p(\mathbf{X}_t|\hat{b}_i,\hat{k}_i)p(\hat{b}_i,\hat{k}_i|\hat{b}_i^{old}\hat{k}_i^{old})}{q(b|\mathbf{X}_t,\tilde{b}^s)}, \quad (7)$$

where $p(\hat{b}_i,\hat{k}_i|\hat{b}_i^{old}\hat{k}_i^{old})$ is defined in Eq. (6) and $q(b|\mathbf{X}_t,\tilde{b}^s)$ is defined in Eq.(2). The observation probability $p(\mathbf{X}_t|\hat{b}_i,\hat{k}_i)$ consists of three parts as

$$p(\mathbf{X}_t|\hat{b}_i,\hat{k}_i) \propto w_i^{ch}\times w_i^{sp}\times w_i^t. \quad (8)$$

The two weights, the chroma vector weight w_i^{ch} and spectrogram weight w_i^{sp}, are measures of pitch information. The weight w_i^t is a measure of temporal information. We use both the chroma vector similarity and the spectrogram similarity to estimate the score position because they have a complementary relationship. A chroma vector has 12 elements corresponding to the pitch name, $C, C\sharp, ..., B$. This is a good feature for audio-to-score matching because the chroma vector is easily derived from both the audio signal and the musical score. However, the elements of a chroma vector become ambiguous when the pitch is low due to the frequency resolution limit. The harmonic structure observed in the spectrogram alleviates this problem because it makes the pitch distinct in the higher frequency region.

To match the spectrogram $X_{f,\tau}$, where $t-L<\tau\le t$, the audio sequence is aligned with the corresponding score for each particle, as shown in Figure 4. Each frame of the spectrogram at time τ is assigned to the score frame k_τ^i that is discrete at 1/12 interval using the estimated score position $\hat{k}_i$ and the beat interval (tempo) $\hat{b}_i$ as:

$$k_\tau^i = \frac{1}{12}\lfloor 12\times(\hat{k}_i-(t-\tau)/\hat{b}_i)+0.5\rfloor, \quad (9)$$

where $\lfloor x\rfloor$ is the floor function.

The sequence of chroma vectors $\mathbf{c}_\tau^a$ is calculated from the spectrum $X_{f,\tau}$ using 12 types of band-pass filters for each element [13]. The value of each element in the score chroma vector $\mathbf{c}_{k_\tau^i}^s$ is 1 when the score has a corresponding note, and 0 otherwise. The chroma weight w_i^{ch} is calculated as:

$$w_i^{ch} = \frac{1}{L_{frm}}\sum_{\tau=t-L}^{t}\mathbf{c}_\tau^a\cdot\mathbf{c}_{k_\tau^i}^s, \quad (10)$$

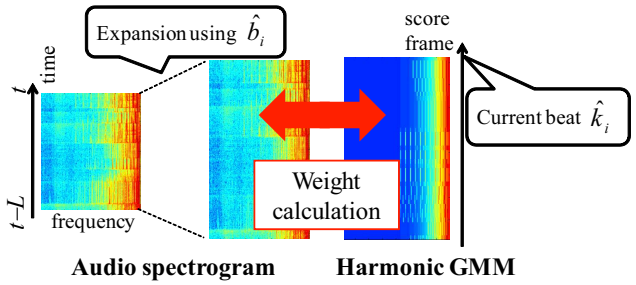

Fig. 4. Weight calculation for pitch information

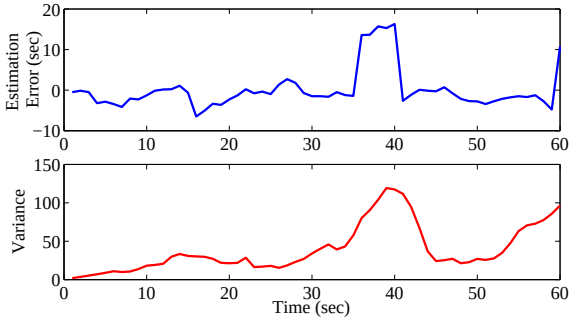

Fig. 5. Relationship between estimation error (top) and variance (bottom).

where L_{frm} is the number of audio frames equivalent to L (sec). Both vectors bfc^a_τ and $\mathbf{c}^s_{k^i_\tau}$ are normalized before applying them to Eq. (10).

The spectrogram weight w^{sp}_i is derived from the Kullback-Leibler divergence with regard to the shape of spectrum between the audio and the score.

$$w^{sp}_i = \left(1+D^{KL}_i\right)\exp\left(-D^{KL}_i\right), \quad (11)$$

$$D^{KL}_i = \frac{1}{L_{frm}}\sum_{\tau=t-L}^{t}\sum_f X_{f,\tau}\log\frac{X_{f,\tau}}{\hat{X}_{f,k_{\tau^i}}}, \quad (12)$$

where D^{KL}_i in Eq. (12) is the dissimilarity between the audio and score spectrograms. Before calculating Eq. (12), the spectrum is normalized such that $\sum_f X_{f,\tau} = \sum_f \hat{X}_{f,k^i_\tau} = 1$. The positive value D^{KL}_i is mapped to the weight w^{sp}_i by Eq. (12) where the range of w^{sp}_i is between 0 and 1. For the calculation of w^{sp}_i, the spectrum $\hat{X}_{f,k^i_\tau}$ is generated from the musical score by using the harmonic gaussian mixture model (GMM), the first term in Eq. (13).

$$\hat{X}_{f,k^i_\tau} = \sum_{r=1}^{r_{k\tau^i}}\sum_{g=1}^{G} h(g)N(f;gF_{n^r_{k\tau^i}},\sigma^2)+C(f), \quad (13)$$

$$C(f) = A\exp(-\alpha f). \quad (14)$$

In Eq. (13), g is the harmonic index, G is the number of harmonics, and $h(g)$ is the height of each harmonics. $F_{n^r_{k\tau^i}}$ is the fundamental frequency of note $n^r_{k\tau^i}$ and the variance σ^2. The parameters are empirically set as: $G=10$, $h(g)=0.2^g$, $\sigma^2=0.8$. To avoid zero divides in Eq. (12), pink noise is added to the score spectrogram (Eq. (14)). A is a constant that makes the power of the pink noise 5% of that of the harmonic GMM. α is determined such that $\log_{10}(C(f+\Delta f)/C(f)) = -0.6$, where Δf is the number of frequency bins corresponding to 1000 (Hz).

The weight w^t_i is the measure of the beat interval and obtained from the normalized cross correlation of the spectrogram through a shift by $\hat{b}_i$:

$$w^t_i = R(\hat{b}_i, \mathbf{X}_t), \quad (15)$$

where $R(\hat{b}_i, \mathbf{X}_t)$ is defined in Eq. (3).

D. Resampling Based on the Weights

After calculating the weight of all particles, the particles are resampled. In this procedure, particles with a large weight are selected many times, whereas those with a small weight are discarded because their score position is unreliable. A particle p is drawn independently N times from the distribution:

$$P(p=p_i) = \frac{w_i}{\sum_{i=1}^N w_i}. \quad (16)$$

A set of resampled particles that have the equal weight approximate the distribution of the current score position. After N particles are resampled, the beat interval, equivalent to the tempo, $\hat{b}$ and the score position $\hat{k}$ are estimated by averaging the values that densely distributed particles hold. Then, the score position ΔT ahead in time $\hat{k}^{pred}$ is predicted by the following equation:

$$\hat{k}^{pred} = \hat{k}+\Delta T/\hat{b}. \quad (17)$$

E. Initial Probability Distribution

The initial particles are set as follows: (1) draw N samples of the beat interval $\hat{b}_i$ value from a uniform distribution ranging from $\tilde{b}^s - 60/W$ to $\tilde{b}^s + 60/W$ where W is the window width in Eq. (4). (2) Set the score position of each particle $\hat{k}_i$ to 0. The bpm x (beat/min) is converted into the corresponding beat interval b (sec) with the equation $b = 60/x$.

F. Estimation Confidence of Score Following

The variance $s^2(t)$ of the predicted score position is used as the estimation confidence:

$$s^2(t) = \sum_{i=1}^{N}\left(\hat{k}_i-\mu\right)^2/N, \quad (18)$$

where $\hat{k}_i$ comes from Eq. (5) and μ is the mean of $\hat{k}_i, 1 \le i \le N$. In general, the high variance means that particles are widely distributed over the score. The relationship between the variance and the estimation error is shown in Figure 5. The estimation error is defined as Eq. (21). The variance tends to increase faster when the cumulative error grows larger around 35–40 (sec) in Figure 5. A rapid drop in variance means the majority of particles converge to a certain score position. If the particles converges to a correct score position, the variance remains stable. On the other hand, if the particles move to the wrong score position, the variance starts soaring again.

Switching between the melody level and rhythm level is carried out as follows:

1) First, the system reports the score position and the tempo.
2) If Eq. (19) is satisfied, the system switches to the rhythm level and stops reporting the score position.
3) After a drop in the variance described in Eq. (20), and if Eq. (19) remains unsatisfied for the subsequent $I\Delta T$, the system switches back to the melody level and resumes reporting the estimated score position.

$$s^2(t) - s^2(t - I\Delta T) > \gamma^{inc} I \quad (19)$$
$$s^2(t) - s^2(t - I\Delta T) < -\gamma^{dec} I \quad (20)$$

These parameters are set as: $I = 5$, $\gamma^{inc} = \gamma^{dec} = 4$.

IV. Experimental Evaluation

This section presents the prediction error of the score following in various conditions: (1) comparisons with Antescofo [14], (2) the effect of two-level synchronization, (3) the effect of the number of particles N, and (4) the effect of the width of window function W in Eq. (4).

A. Experimental Setup

Our system was implemented in C++ on a MacOSX with an Intel Core2 Duo processor. We used 20 jazz songs from the RWC Music Database [15] listed in Table II. The sampling rate was 44100 (Hz) and Fourier transform was executed with a 2048 (pt) window length and 441 (pt) window shift. The parameter settings are listed in Table I.

TABLE I
Parameter settings

Denotation		Value	
Look-ahead time	ΔT	1	(sec)
Window length	L	2.5	(sec)
Score position variance	σ_k^2	1	(beat^2)
Beat duration variance	σ_b^2	0.2	(sec^2/beat^2)

TABLE II
Songs used for the experiments

Song ID	File name	Tempo (bpm)	Instruments[1]
1	RM-J001	150	Pf
2	RM-J003	98	Pf
3	RM-J004	145	Pf
4	RM-J005	113	Pf
5	RM-J006	163	Gt
6	RM-J007	78	Gt
7	RM-J010	110	Gt
8	RM-J011	185	Vib & Pf
9	RM-J013	88	Vib & Pf
10	RM-J015	118	Pf & Bs
11	RM-J016	198	Pf, Bs & Dr
12	RM-J021	200	Pf, Bs, Tp & Dr
13	RM-J023	84	Pf, Bs, Sax & Dr
14	RM-J033	70	Pf, Bs, Fl & Dr
15	RM-J037	214	Pf, Bs, Vo & Dr
16	RM-J038	125	Pf, Bs, Gt, Tp & Dr etc.
17	RM-J046	152	Pf, Bs, Gt, Kb & Dr etc.
18	RM-J047	122	Kb, Bs, Gt & Dr
19	RM-J048	113	Pf, Bs, Gt, Kb & Dr etc.
20	RM-J050	157	Kb, Bs, Sax & Dr

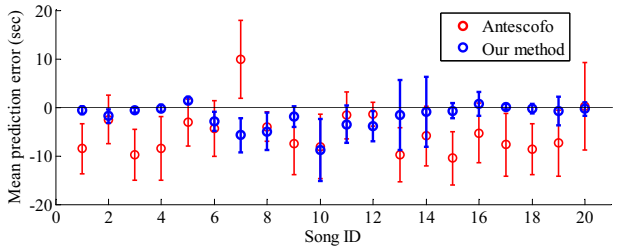

Fig. 6. Mean prediction errors in our method and Antescofo
the number of particles N is 500
the width of the tempo window W is 15 (bpm)

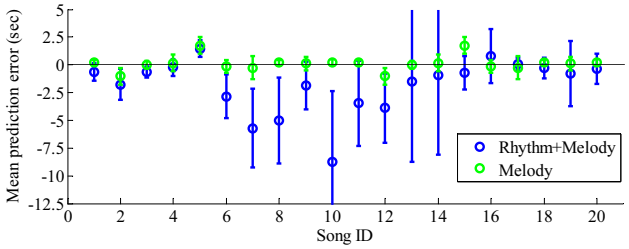

Fig. 7. Mean prediction errors in the melody level and both levels
the number of particles N is 500
the width of the tempo window W is 15 (bpm)

B. Score Following Error

At ΔT intervals, our system predicts the score position $\hat{k}(t+\Delta T)$ when the current time is t. Let $s(k)$ be the ground truth time at beat k in the music. $s(k)$ is defined for positive continuous k by linear interpolation of musical event times. The prediction error $e^{pred}(t)$ is defined as:

$$e^{pred}(t) = t + \Delta T - s(\hat{k}(t+\Delta T)). \quad (21)$$

Positive $e^{pred}(t)$ means the estimated score position is ahead of the true position.

a) Our method vs HMM-based score following method: Figure 6 shows the errors in the predicted score positions for 20 songs when the number of particles N is 500 and the width of the tempo window W corresponds to 15 (bpm). The comparison between our method in blue plots and Antescofo [14] in red plots. The mean values of our method is calculated by averaging all prediction errors both on the rhythm level and on the melody level. This is because Figure 6 is intended to compare the particle filter-based score following algorithm with HMM-based one. Our method reports less mean error values for 16 our of 20 songs than existing score following algorithm Antescofo.

There can be observed striking errors in songs ID 6–14. Main reasons are two-fold: (1) In songs ID 6–10, a guitar or multiple instruments are used. Among their polyphonic sounds, some musical notes sound so vague that the audio spectrogram is different from the GMM-based spectrogram generated by Eq. (13). (2) On top of the first reason, temporal fluctuation is observed in songs ID 11–14. These two factors

[1]abbreviations: Pf=Piano, Gt=Guitar, Vib=Vibraphone, Bs=Bass, Dr=Drums, Tp=Trumpet, Sax=Saxophone, Fl=Flute, Vo=Vocal, Kb=Keyboard

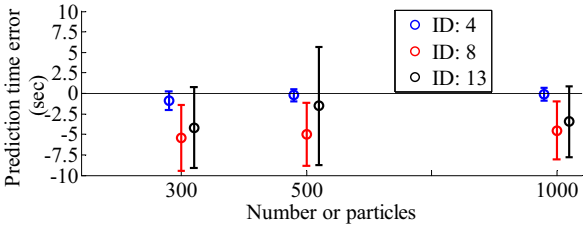

Fig. 8. Number of particles N vs prediction errors

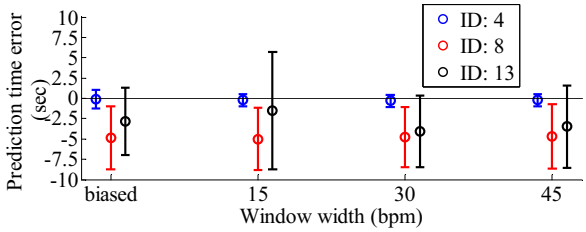

Fig. 9. Window width W vs prediction errors

lead score following algorithms to fail to track a musical audio signal.

b) The effect of two-level switching: Figure 7 shows the errors in only the melody level and both the melody and the rhythm level. The effect of 2-level synchronization is outstanding in songs ID 6–14. The errors in these songs are more than the other songs as in Figure 6. Therefore, the particles are tend to spread over the score position dimension and lead the variance to increase. This is why our two-level synchronization strategy is very effective for these songs.

c) Prediction error vs the number of particles: Figure 8 shows the mean prediction errors for various numbers of particles N. Three songs were chosen for the comparison songs ID 4, 8, 13. The window width W is fixed to 15 (bpm). When the number of particles increases in tracking a low-error song ID 4, the error is slightly alleviated. On the other hand, when the number increases in tracking an errorful songs like ID 8 or 13, an increase in the number does not contribute to the reduction of mean errors. This result indicates that these errors are caused by other reasons such as mismatch between the audio and the score in the observation step instead of the number of particles.

d) Prediction error vs the width of the tempo window: Figure 9 shows the mean prediction errors for various widths of tempo window W. In this experiment, W is set to $15, 30, and 45$ (bpm). To simulate the situation that the given tempo is different from the performance, a *biased* case is also tested. In this case, W is set to 15 and the tempo given by the score is biased by 15 (bpm). Therefore, the true tempo is located at the edge of the window function is Eq. (4). Intuitively, the narrower the width is, the closer to zero the error value should be because the chance of choosing a wrong tempo will be reduced. However, almost the same results are obtained for various W and even in the biased case. This is because peaks in the normalized cross correlation in Eq. (3) are sufficiently striking to choose an appropriate beat interval value from the proposal distribution in Eq. (2).

V. Discussion and Future Work

Experimental results show that the score following performance varies with the music played. Needless to say, a music robot hears a mixture of musical audio signals and its own singing voice or instrumental performance. Some musical robots [5], [7], [16] use self-generating sound cancellation [17] from a mixture of sounds. Our score following should be tested with such a cancellation because the performance of score following may deteriorate if such a cancellation is used.

The design of the two-level synchronization is intended to improve existing methods reported in the literature. Some of the existing beat tracking [7] and score following [5] methods are not robust against temporal fluctuations in the performance. This is similar to the case of spoken dialogue systems. Since no one projects that a 100%-accurate ASR is forthcoming, a quick and easy way to correct recognition errors is mandatory [18]. We have developed the two-level synchronization to make score following usable for co-player robots. The next step to enrich the score following is a recovery mechanism that occurs when the score position is lost. When human musicians miss the score position, they try to recover the error by looking for landmarks ahead such as the beginning of a chorus part. Once landmarks are automatically extracted from the musical score and are detected in the audio signal, music robots can recover to the landmarks by distributing enough particles at the detected landmarks. For this recovery mechanism, automatic extraction of these landmarks from the score and the landmark detection from the audio should be realized.

We are currently developing ensemble robots with a human flutist. The human flutist leads the ensemble, and a singer and thereminist robot follows [19]. The two-level synchronization approach benefits this ensemble as follows: when the score position is uncertain, the robot starts scatting the beats, or faces downward and sings in a low voice; when the robot is aware of the part of the song, it faces up and presents a loud and confident voice. This posture-based voice control is attained through the voice manipulation system [20].

Our score following using the particle filter should also be able to improve an instrument-playing robot. In fact, the theremin player robot moves its arms to determine the pitch and the volume of theremin. Therefore, the prediction mechanism enables the robot to play the instrument in synchronization with the human performance. In addition, a multimodal ensemble system using a camera [21] can be naturally aggregated with our particle-filter-based score following system. This is because the flexible framework of the particle filter facilitates aggregation of multimodal information sources [22].

VI. Conclusion

This paper presented a score following system based on a particle filter to attain the two-stage synchronization for

interactive music robots that presents musical expressions. A two-level synchronization is performed at the rhythm level and the melody level. The reliability of score following is calculated from the density of particles and is used to switch between levels. The experimental results demonstrated the feasibility of the system. The future work includes development of interactive ensemble robots, and it will be reported in the near future.

ACKNOWLEDGMENT

This research was supported in part by Kyoto University Global COE, in part by JSPS Grant-in-Aid for Scientific Research (S) 19100003, and in part by a Grant-in-Aid for Scientific Research on Innovative Areas (No. 22118502) from the MEXT, Japan.

REFERENCES

[1] A. Alford, S. Northrup, K. Kawamura, K-W. Chan, and J. Barile. A music playing robot. In *FSR 99*, pages 29–31, 1999.

[2] K. Shibuya, S. Matsuda, and A. Takahara. Toward Developing a Violin Playing Robot - Bowing by Anthropomorphic Robot Arm and Sound Analysis -. In *Proc. of IEEE International Conference on Robot and Human Interactive Communication*, pages 763–768, 2007.

[3] G. Weinberg and S. Driscoll. Toward Robotic Musicianship. *Computer Music Journal*, 30(4):28–45, 2006.

[4] R. Dannenberg and C. Raphael. Music Score Alignment and Computer Accompaniment. *Comm. ACM*, 49(8):38–43, 2006.

[5] T. Otsuka, K. Nakadai, Toru Takahashi, K. Komatani, T. Ogata, and H. G. Okuno. Incremental Polyphonic Audio to Score Alignment using Beat Tracking for Singer Robots. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2289–2296, 2009.

[6] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking. *IEEE Transactions on Signal Proc.*, 50(2):174–189, 2002.

[7] K. Murata, K. Nakadai, K. Yoshii, R. Takeda, T. Torii, H. G. Okuno, Y. Hasegawa, and H. Tsujino. A Robot Uses Its Own Microphone to Synchronize Its Steps to Musical Beats While Scatting and Singing. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2459–2464, 2008.

[8] H. Kenmochi and H. Ohshita. Vocaloid – commercial singing synthesizer based on sample concatenation. In *Proc. of INTERSPEECH*, pages 4010–4011, 2007.

[9] S. Dixon. An On-line Time Warping Algorithm for Tracking Musical Performances. In *Proc. of the International Joint Conference on Artificial Intelligence*, pages 1727–1728, 2005.

[10] N. Orio, S. Lemouton, and D. Schwarz. Score Following: State of the Art and New Developments. In *Proc. of International Conference on New Interfaces for Musical Expression*, pages 36–41, 2003.

[11] A. Cont. ANTESCOFO: Anticipatory Synchronization and Control of Interactive Parameters in Computer Music. In *Proc. of International Computer Music Conference*, 2008.

[12] T. Otsuka, K. Nakadai, Toru Takahashi, K. Komatani, T. Ogata, and H. G. Okuno. Design and Implementation of Two-level Synchronization for Interactive Music Robot. In *Proc. of Twenty-Fourth AAAI Conference on Artificial Intelligence*, pages 1238–1244, 2010.

[13] M. Goto. A Chorus Section Detection Method for Musical Audio Signals and Its Application to a Music Listening Station. *IEEE Transactions on Audio, Speech and Language Proc.*, 14(5):1783–1794, 2006.

[14] A. Cont. A Coupled Duration-Focused Architecture for Realtime Music to Score Alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32, 2010. to appear.

[15] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka. RWC Music Database: Music Genre Database and Musical Instrument Sound Database. In *Proc. of International Conference on Music Information Retrieval*, pages 229–230, 2003.

[16] T. Otsuka, K. Nakadai, T. Takahashi, K. Komatani, T. Ogata, and H. G. Okuno. Music-ensemble robot that is capable of playing the theremin while listening to the accompanied music. *Trends in Applied Intelligent Systems*, LNAI 6096:102–112, 2010.

[17] R. Takeda, K. Nakadai, K. Komatani, T. Ogata, and H. G. Okuno. Barge-in-able Robot Audition Based on ICA and Missing Feature Theory under Semi-Blind Situation. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1718–1723, 2008.

[18] K. Larson and D. Mowatt. Speech Error Correction: The Story of the Alternatives List. *International Journal of Speech Technology*, 6(2):183–194, 2003.

[19] A. Lim, T. Mizumoto, L. Cahier, T. Otsuka, T. Takahashi, K. Komatani, T. Ogata, and H. G. Okuno. Robot Musical Accompaniment: Integrating Audio and Visual Cues for Real-time Synchronization with a Human Flutist. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*. *to appear*.

[20] T. Otsuka, K. Nakadai, T. Takahashi, K. Komatani, T. Ogata, and H. G. Okuno. Voice-awareness control for a humanoid robot consistent with its body posture and movements. *PALADYN Journal of Behavioral Robotics*, 1(1):80–88, 2010.

[21] D. Overholt, J. Thompson, L. Putnam, B. Bell, J. Kleban, B. Sturm, and J. Kuchera-Morin. A Multimodal System for Gesture Recognition in Interactive Music Performance. *Computer Music Journal*, 33(4):69–82, 2009.

[22] K. Nickel, T. Gehrig, R. Stiefelhagen, and J. McDonough. A Joint Particle Filter for Audio-visual Speaker Tracking. In *Proc. of International Conference on Multimodal Interfaces*, pages 61–68, 2005.