Design and Implementation of Two-Level Synchronization for an Interactive Music Robot

Takuma Otsuka†, Kazuhiro Nakadai*, Toru Takahashi†,
Kazunori Komatani†, Tetsuya Ogata†, and Hiroshi G. Okuno†
†Graduate School of Informatics, Kyoto University, Kyoto, 606-8501, Japan.
{ohtsuka, tall, komatani, ogata, okuno}@kuis.kyoto-u.ac.jp
∗Honda Research Institute Japan, Co., Ltd., Wako, Saitama, 351-0114, Japan.
nakadai@jp.honda-ri.com

Abstract

Our goal is to develop an interactive 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 singing and dancing. 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-layer 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 polyphonic jazz songs. The results confirm that our method switches levels in accordance with the difficulty of the score estimation. When the tempo of the music is less than 120 (beats per minute; bpm), the estimated score positions are accurate and reported; when the tempo is over 120 (bpm), the system tends to report only the tempo to suppress the error in the reported score position predictions.

1 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. Their music interaction requires two important functions; synchronization with the music and generation of musical expressions, such as singing and dancing. Many instrument-performing robots such as those presented in (Alford et al. 1999; Shibuya, Matsuda, and Takahara 2007) are only capable of the latter function, they may have difficulty in playing live with human performers. In fact, synchronization with the music is critical for interaction.

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. 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. (Weinberg and Driscoll 2006) 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 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 (Dannenberg

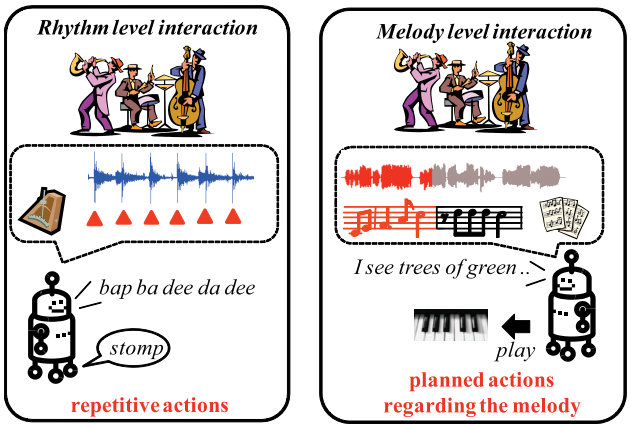

Figure 1: Two levels in musical interactions

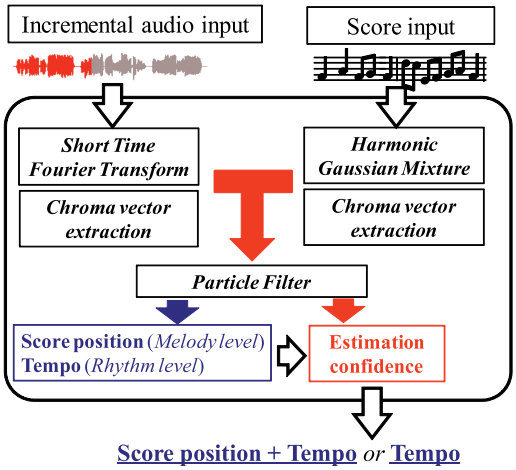

Figure 2: Two-level synchronization architecture

and Raphael 2006), is introduced for the melody level for the purpose of a robot singer (Otsuka et al. 2009), but this method is no good if the robot fails to track the musical score. The important principle in designing a co-player robot is to allow 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 (Arulampalam et al. 2002). 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.

2 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.* (2008) reports that it takes around 200 (ms) to generate a singing voice using singing voice synthesizer VOCALOID (Kenmochi and Ohshita 2007). 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.

2.1 State-of-the-art Score Following Systems

Most conventional score following methods are based on either dynamic time warping (DTW) (Dixon 2005) or hidden Markov model (HMM) (Orio, Lemouton, and Schwarz 2003). 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 (Cont 2008) 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.

2.2 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 unknown; only pairs of pitch and length (e.g., quarter note) are given as a score.

The issues are (1) simultaneous estimation of the score position and tempo and (2) the design of the estimation confidence. The assumption conflicts the idea that the tempo information in the score can improve the performance of the algorithm by limiting a range of possible tempo. The problem is the numerical interpretation of a qualitative tempo like "moderato" in beat-per-minute (bpm). The tempo is not always given in a quantitative way but in a qualitative way. Our current approach copes with the both situations by estimating the tempo directly from the audio signal.

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.

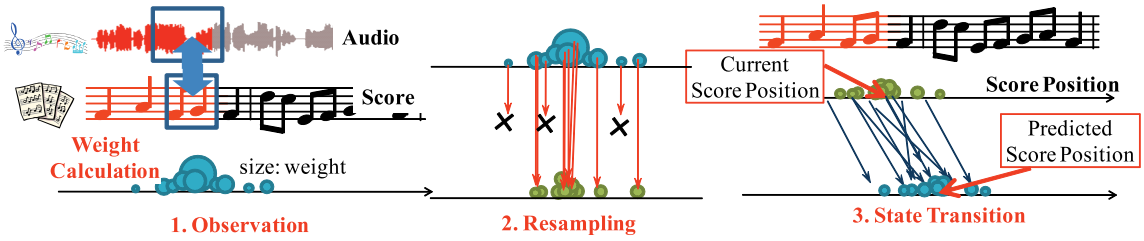

Figure 3: Overview of the Score Following using Particle Filter

3 Score Following using Particle Filter

3.1 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 12 frames to account for the resolution of sixteenth-note and triplets. Musical notes $\mathbf{n}_k = [n_k^1 ... n_k^{r_k}]^T$ 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 beat (the quarter note position), and the unit for $\hat{b}_i$ is 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) observation, (2) resampling, and (3) state transition (prediction). 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. The state transition corresponds to the prediction of the future score position ΔT ahead in time. Each procedure is described in the following subsections.

3.2 Observation Model and Weight Calculation

At time t, a spectrogram $X_{f,\tau}, t - L < \tau \leq t$ is used for the weight calculation. L denotes the window length of the spectrogram. The weight of each particle $w_i, 1 \leq i \leq N$ is a product of three weights:

$$w_i = w_i^{ch} \times w_i^{sp} \times w_i^t. \quad (1)$$

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 $t - L < \tau \leq t$, 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:

$$k_\tau^i = \frac{1}{12} \lfloor 12 \times (\hat{k}_i - (t - \tau)/\hat{b}_i) + 0.5 \rfloor, \quad (2)$$

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 (Goto 2006). 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:

$$w_i^{ch} = \frac{1}{L_{frm}} \sum_{\tau=t-L}^{t} \mathbf{c}_\tau^a \cdot \mathbf{c}_{k_\tau^i}^s, \quad (3)$$

where L_{frm} is the number of audio frames equivalent to L (sec). Both vectors bfc_τ^a and $\mathbf{c}_{k_\tau^i}^s$ are normalized before applying them to Eq. (3).

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

$$w_i^{sp} = \left(1 + D_i^{KL}\right) \exp\left(-D_i^{KL}\right), \quad (4)$$

$$D_i^{KL} = \frac{1}{L_{frm}} \sum_{\tau=t-L}^{t} \sum_f X_{f,\tau} \log \frac{X_{f,\tau}}{\hat{X}_{f,k_{\tau^i}}}, \quad (5)$$

where D_i^{KL} in Eq. (5) is the dissimilarity between the audio and score spectrograms. Before calculating Eq. (5), the spectrum is normalized such that $\sum_f X_{f,\tau} = \sum_f \hat{X}_{f,k_\tau^i} = 1$. The positive value D_i^{KL} is mapped to the weight w_i^{sp} by Eq. (5) where the range of w_i^{sp} is between 0 and 1. For the calculation of w_i^{sp}, the spectrum $\hat{X}_{f,k_\tau^i}$ is generated from the musical score by using the harmonic gaussian mixture model (GMM), the first term in Eq. (6).

$$\hat{X}_{f,k_\tau^i} = \sum_{r=1}^{r_{k\tau^i}} \sum_{g=1}^{G} h(g) N(f; g F_{n_{k\tau^i}^r}, \sigma^2) + C(f), \quad (6)$$

$$C(f) = A \exp\left(-\alpha f\right). \quad (7)$$

In Eq. (6), g is the harmonic index, G is the number of harmonics, and $h(g)$ is the height of each harmonics. $F_{n_{k\tau^i}^r}$ is the fundamental frequency of note $n_{k\tau^i}^r$ and the variance σ^2. The parameters are empirically set as: $G = 10$,

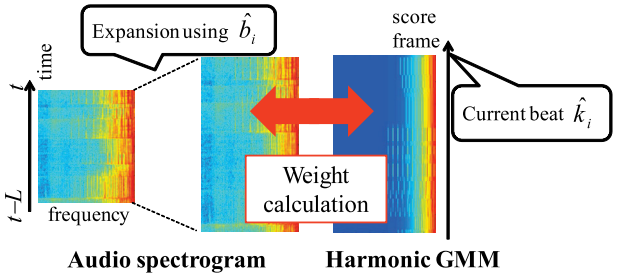

Figure 4: Weight calculation for pitch information

$h(g) = 0.2^g$, $\sigma^2 = 0.8$. To avoid zero divides in Eq. (5), pink noise is added to the score spectrogram (Eq. (7)). 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_i^t is the measure of the beat interval and obtained from the normalized cross correlation of the spectrogram through a shift by $\hat{b}_i$:

$$w_i^t = \frac{\sum_{\tau=t-L}^{t}\sum_{f} X_{f,\tau}X_{f,\tau-\lfloor\hat{b}_i+0.5\rfloor}}{\sqrt{\sum_{\tau=t-L}^{t}\sum_{f} X_{f,\tau}^2 \sum_{\tau=t-L}^{t}\sum_{f} X_{f,\tau-\lfloor\hat{b}_i+0.5\rfloor}^2}}, \quad (8)$$

Eq. (8) is defined in case $\hat{b}_i < \Delta T$; otherwise, $w_i^t = 0$.

3.3 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:

$$P(p = p_i) = \frac{w_i}{\sum_{i=1}^{N} w_i}. \quad (9)$$

A set of resampled particles that have the equal weight approximate the distribution of the current score position. If the frame number of the current score position has to be estimated, it is derived by taking the mean value of the score positions that densely distributed particles hold.

3.4 State Transition Model

The future score position is predicted by updating particles with the following state transition model:

$$\hat{k}_i \leftarrow \hat{k}_i + \Delta T/\hat{b}_i + u, \quad (10)$$

$$\hat{b}_i \leftarrow \hat{b}_i + v, \quad (11)$$

where u and v are gaussian random variables with means 0 and variances σ_u^2 and σ_v^2. The score position ΔT ahead in time is predicted by taking the mean value of densely distributed particles. A further prediction can be obtained by applying Eq. (10, 11) repeatedly, or simply extrapolating the score position with current tempo.

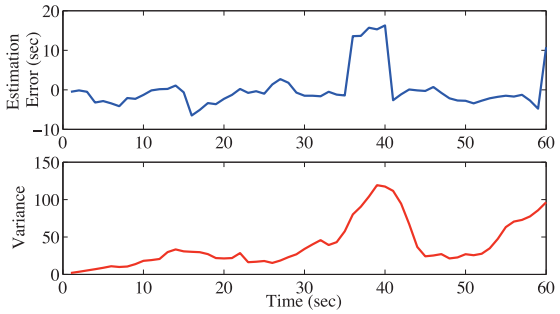

Figure 5: Relationship between estimation error (top) and variance (bottom).

3.5 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 60 (bpm; beats per minute) to 180 (bpm). (2) Set the score position of each particle to $\Delta T/\hat{b}_i$. The bpm x (beat/min) is converted into the corresponding beat interval b (sec) with the equation $b = 60/x$.

3.6 Estimation Confidence of Score Following

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

$$s^2(t) = \sum_{i=1}^{N}\left(\hat{k}_i - \mu\right)^2/N, \quad (12)$$

where $\hat{k}_i$ comes from Eq. (10) 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. (15). 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. (13) 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. (14), and if Eq. (13) remains unsatisfied for the subsequent $I\Delta T$, the system switches back to the melody level and resumes reporting the estimated score position.

$$s^2(t) - s^2(t - I\Delta T) > \gamma^{inc} I \quad (13)$$

$$s^2(t) - s^2(t - I\Delta T) < -\gamma^{dec} I \quad (14)$$

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

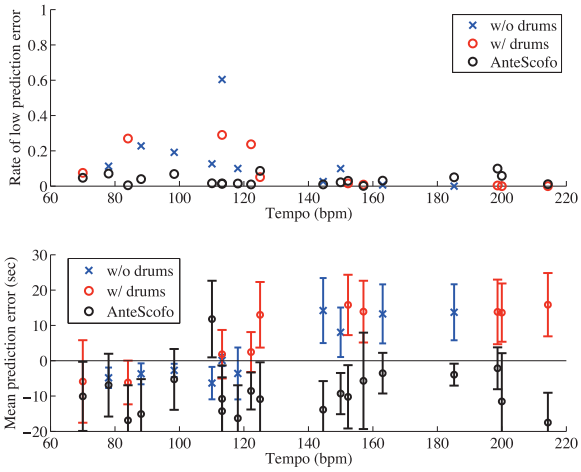

Figure 6: Score following performance (overall result)
Top: rate of low prediction error
Bottom: mean and standard deviation of prediction errors

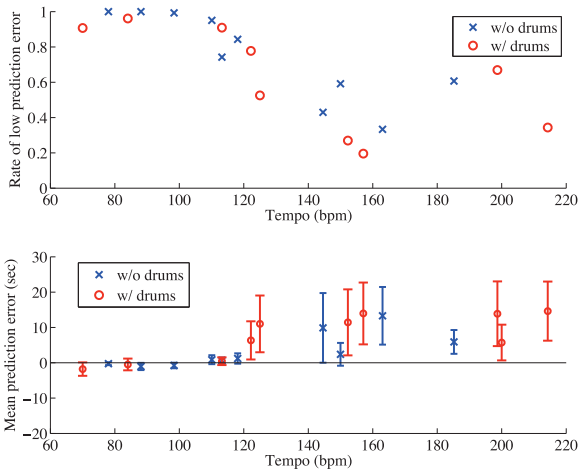

Figure 7: Score following performance (melody level)
Top: rate of low prediction error
Bottom: mean and standard deviation of prediction errors

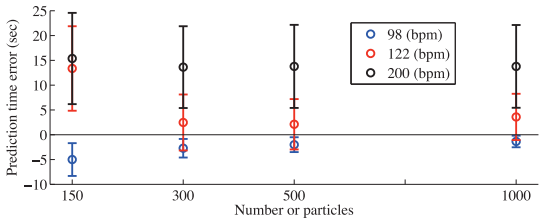

Figure 8: Number of particles vs prediction errors

4 Experimental Evaluation

This section presents the results of experiments on our score following system: (1) the prediction error the score following and (2) the rate of successful switching between music synchronization levels.

4.1 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 (Goto et al. 2003). Ten songs included drum sounds; while the others did not. 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 1.

Table 1: Parameter settings

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

4.2 Score Following Error

At ΔT intervals, our system predicts the score position $\hat{k}(t)$ at $t + \Delta T$ 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 $e(t)$ is defined as:

$$e(t) \quad = \quad t + \Delta T - s(\hat{k}(t)). \tag{15}$$

Positive $e(t)$ means the estimated score position is before the true position. For each song, we calculated the rate of low prediction error, where $|e(t)| < 1$ (sec), in a song, mean of $e(t)$, and standard derivation $e(t)$.

Figure 6 shows the relationship between the tempo of the music and the errors in the predicted score positions when the number of particles N is 300. The comparison between our method in red and blue plots and Antescofo (Cont 2010) in black plots. The tempo is representative of that song. It is also confirmed that we have similar rates of low prediction error even when the 1-second error threshold is replaced with 0.5 or 2 (sec). Blue plots are for drum-less songs, and the red plots are for songs including drum sounds. The top figure shows that the prediction is less erroneous when the tempo is under 120 (bpm), where as the accuracy rate is extremely low when the tempo is over 120 (bpm). This is because the normalized cross correlation in Eq. (8) has multiple peaks when the tempo is over 120 (bpm) i.e., when the beat interval is under $\Delta T/2 = 0.5$ (sec). For example, when the tempo is 150 (bpm), meaning the beat interval is 0.4 (sec/beat), the normalized cross correlation has peaks at $\hat{b}_i = 0.4$ and 0.8. When the tempo of the song is under 120 (bpm), our method tends to exceed existing score following method, Antescofo. However, when the tempo is over 120 (bpm) and multiple candidates of the tempo exist, the performance of our method is sometimes worse than Antescofo. The reason why the rate of low prediction error for each song is overall low is because some of the songs used in the experiment have multiple sounds of various instruments such as an ensemble using a guitar, a piano, a saxophone a bass, and drums. This polyphonic characteristics makes the score following problem even more difficult.

Figure 8 shows the mean prediction time errors for various

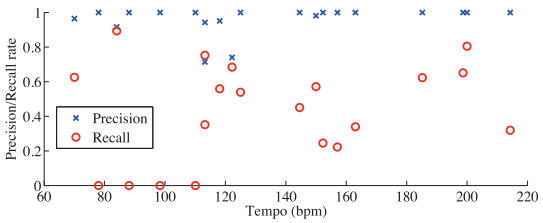

Figure 9: Precision and recall of rhythm level outputs

numbers of particles. Three songs were chosen for the comparison: low tempo of 98 (bpm), mid tempo of 112 (bpm), and high tempo of 200 (bpm). The errors in the low-tempo and mid-tempo songs decrease until the number of particles reaches 500. However, such an effect of the particle number does not occur for high tempo songs. The effect saturates when the number of particles exceeds 500.

4.3 Evaluation of Switching Synchronization

Figure 7 shows the rate of prediction error less than 1 (sec) and the mean and standard deviation among the score position outputs when the system is on the melody level. The figure confirms that erroneous score position outputs are reduced by our switching strategy. This switching mechanism also contributes to stabilizing the score position outputs when the tempo is under 120 (bpm).

This section presents how accurate the switching mechanism is and how the melody level improves the reported score position. The number of particles is fixed to be 300. For each song, the precision ξ_p and recall ξ_r rates of the rhythm level synchronization is defined as: $\xi_p = x^t/y$, $\xi_r = x^t/z$, where x^t is the number of rhythm level outputs when the error of the score position prediction $e(t)$ is over 1 (sec), y is the total number of rhythm level outputs, and z is the total times at which $e(t)$ is over 1 (sec).

Figure 9 shows the precision and recall for 20 songs. The precision of songs over tempo 120 (bpm) is 1 in many cases because there are huge erroneous estimations of the score position (see Figure 6). The mean recall for the 20 songs is 0.43.

The tempo reported on the rhythm level was evaluated as follows. The tempo was regarded as the correct bpm value by the system matched the ground truth tempo within a 10% margin. The rate of correctly reported rhythm-level outputs was calculated as the number of correct tempo reports divided by the total number of rhythm level outputs. The average rate of the correct tempo reports for the 20 songs was 0.46 with a standard deviation of 0.22.

These experiments show that the key to successful score following with our method is correct estimation of the tempo. The reason why our system fails to find correct tempos over 120 (bpm) is that the normalized cross correlation in Eq. (8) has multiple peaks. Refinement of the observation, Eq. (8), or state transition, Eq. (11), will improve both the score following prediction and tempo estimation.

5 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 (Murata et al.; Mizumoto et al.; Otsuka et al.) use self-generating sound cancellation (Takeda et al. 2008) 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 (Murata et al. 2008) and score following (Otsuka et al. 2009) 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 (Larson and Mowatt 2003). 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 two robots, a singer and thereminist, follow. 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 (Otsuka et al. 2010).

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 (Overholt et al. 2009) 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 (Nickel et al. 2005). Furthermore, alternative particle filter algorithm can improve the performance of the score position estimation. Our method generates particles based on the previous set of particles as shown in Eq. (10, 11) without using the observed audio signal. However, the observation can be a useful cue to estimate the score position and musical tempo. For example, the NCC in the right-hand side of Eq. (8) provides

the likelihood of the tempo derived from the observed audio signal. For the efficient use of the audio signal, the design of the proposal distribution from which particles are drawn is important. The algorithm of a particle filter using a proposal distribution is explained in (Thrun, Burgard, and Fox 2005; Arulampalam et al. 2002)

6 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 JSPS Grant-in-Aid for Scientific Research (S) 19100003 and in part by Kyoto University Global COE. The authors would like to thank the members of Okuno and Ogata Laboratory for their discussion and valuable suggestions.

References

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

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

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

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

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

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

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

Goto, M. 2006. 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.

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

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

Mizumoto, T.; Tsujino, H.; Takahashi, T.; Ogata, T.; and Okuno, H. G. 2009. Thereminist Robot: Development of a Robot Theremin Player with Feedforward and Feedback Arm Control based on a Theremin's Pitch Model. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2297–2302.

Murata, K.; Nakadai, K.; Yoshii, K.; Takeda, R.; Torii, T.; Okuno, H. G.; Hasegawa, Y.; and Tsujino, H. 2008. 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*, 2459–2464.

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

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

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

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

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

Shibuya, K.; Matsuda, S.; and Takahara, A. 2007. 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*, 763–768.

Takeda, R.; Nakadai, K.; Komatani, K.; Ogata, T.; and Okuno, H. G. 2008. 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*, 1718–1723.

Thrun, S.; Burgard, W.; and Fox, D. 2005. *Probabilistic Robotics*. Cambridge, MA: MIT Press.

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