2012 IEEE International Conference on Robotics and Automation
RiverCentre, Saint Paul, Minnesota, USA
May 14-18, 2012

Online Audio Beat Tracking for a Dancing Robot in the Presence of Ego-Motion Noise in a Real Environment

João Lobato Oliveira, Gökhan Ince, Keisuke Nakamura and Kazuhiro Nakadai

Abstract— **This paper presents the design and implementation of a real-time real-world beat tracking system which runs on a dancing robot. The main problem of such a robot is that, while it is moving, ego noise is generated due to its motors, and this directly degrades the quality of the audio signal features used for beat tracking. Therefore, we propose to incorporate ego noise reduction as a pre-processing stage prior to our tempo induction and beat tracking system. The beat tracking algorithm is based on an online strategy of competing agents sequentially processing a continuous musical input, while considering parallel hypotheses regarding tempo and beats. This system is applied to a humanoid robot processing the audio from its embedded microphones on-the-fly, while performing simplistic dancing motions. A detailed and multi-criteria based evaluation of the system across different music genres and varying stationary/non-stationary noise conditions is presented. It shows improved performance and noise robustness, outperforming our conventional beat tracker (*i.e.*, without ego noise suppression) by 15.2 points in tempo estimation and 15.0 points in beat-times prediction.**

I. Introduction

In order to introduce the ability of music listening and cognition into musical expressive robotic agents, which simultaneously generate corporeal motor responses, robot audition algorithms must consider causal and low-cost computations. Besides, they must cope with the signal distortions caused by environmental and robot's ego noises (*i.e.*, motor noises generated during the robot's motion), since these degrade the performance of *Music Information Retrieval (MIR)* algorithms at the audio signal level.

In this paper we address the problem of online audio beat tracking for a dancing robot with embedded microphones in the presence of ego noise, by integrating an online template-based ego noise suppression scheme [1] with a real-time beat tracking system [2]. To the knowledge of the authors, this is the first study in musical robotics, which tackles non-stationary ego-motion noise directly, instead of ignoring it or circumventing the problem by increasing the volume of the music drastically. In fact, the developed system was applied to a dancing robot performing simplistic periodic movements in natural (*i.e.*, real-world) conditions under varying music loudness. We evaluated our system in terms of *Signal-to-Noise Ratio (SNR)* improvement and accuracies of tempo and beat tracking in response to eleven different musical genres and noise conditions with increasing levels of difficulty.

João Lobato Oliveira is with the Artificial Intelligence and Computer Science Laboratory (LIACC), FEUP, and with the Institute for Systems and Computer Engineering of Porto (INESC Porto), Rua Dr. Roberto Frias, 4200-465 Porto, Portugal joao.lobato.oliveira@fe.up.pt

Gökhan Ince, Keisuke Nakamura and Kazuhiro Nakadai are with Honda Research Institute Japan Co., Ltd. 8-1 Honcho, Wako-shi, Saitama 351-0188, Japan {gokhan.ince,keisuke,nakadai}@jp.honda-ri.com

II. Related Work

Worldwide, artificial intelligence and robotics researchers are trying to make robots dance to the sound of music [3], and make them participate in ensemble musical performances with humans [4]. These musical robots rely on various music-based interaction schemes, which depend on perceptual algorithms attending to musical qualities such as melody, loudness, pitch, harmony, timbre, and rhythm. Given the focus of this paper, this section describes how different musical robotic agents interact with low-level rhythmic aspects such as note onsets, tempo and beats, giving special attention to the reported tempo and beat tracking algorithms, and, if applied, their noise suppression strategies.

Focused on real-world scenarios, Michalowski *et al.* [5] investigated the role of rhythm and synchronism in human-robot interactions, and their application in pedagogical and therapeutic scenes. Their robot, Keepon, acquired both auditory and visual live rhythmic data from embedded microphones and cameras, in the form of amplitude peaks of the auditory signal or onsets of the optical flow of the visual signal. These multi-modal onsets are used to generate, on-the-fly, a stream of commands that cyclically moves Keepon's bobbing and rocking degrees-of-freedom.

In another study using a robot with embedded sensors, Crick *et. al* [6] integrated live audio-visual sensory inputs into synchronous ensemble drumming performances with humans. Their rhythmic model fuses audio-visual beat events, acquired in real-time from zero-crossings of the human arm motion trajectory over the *ictus* line, and drum onsets (drum-beats).

Despite the real-time concepts of [5]–[8], none of the approaches regarded the effects of noise, of different natures, in their music processing modules. By taking noise into account, Yoshii, Mizumoto, Murata *et al.* [9]–[11] proposed a set of beat-synchronous experiments with a human-size humanoid, by means of different real-time beat tracking systems, processing live auditory signals. These included stamping the robot's feet in time with the estimated musical beats [9]; a beat-counting robot that can count musical beats aloud from live music [10]; and a robot that can simultaneously step, scat and sing also according to the musical beats [11]. The [9] and [10] studies made use of Goto's [12] real-time beat tracker, which manages a competing multi-agent system with different autocorrelation and cross-correlation strategies for beat prediction, based

978-1-4673-1404-6/12/$31.00 ©2012 IEEE

on note-onsets and drum-sounds components. In order to improve the beat tracker's robustness against self-voice noises, Mizumoto [10] additionally proposed to integrate an *Independent Component Analysis (ICA)*-based adaptive filter [13] for suppressing the robot's own counting voice, by using the waveform of the counting voice as a prior knowledge. Later, Murata proposed a more efficient beat tracking scheme based on *Spectro-Temporal Pattern Matching (STPM)* [11]. Similarly to the former, the authors also applied a two-channel version of the same semi-blind ICA to suppress the captured auditory signal from the robot's self-voice (*i.e.*, from the robot's scatting and singing). Stepping noise, however, was ignored. They considered that the beats are not affected by the ego-motion noise because they made the highly-restrictive assumption that the beats and the motion cycles are synchronized (*i.e.*, in phase).

Ultimately, in order to improve the interactivity of humanoid musicians in live environments, Grunberg *et al.* [14] integrated a simplified version of Klapuri's real-time beat tracker [15] with a noise adaptive filter based on separate attenuation thresholds for each spectral frequency bin. This noise-robust beat tracker was tested on humanoid Hubo while performing random motions synchronized and unsynchronized to live music stimuli. The results suggested overall improvements in the beat tracking performance when applying the adaptive filter under different conditions.

In the line of the former research and envisioning the application of our system in a real-time robotic system freely dancing in a real environment, we propose:

1) The use of a *real-time* beat tracking system on a *robot with embedded microphones* to cope with the online requirements of our system, at low-cost computations;
2) The use of a *sequentially-driven* causal beat tracking algorithm to *adapt faster* to expressive musical changes of different natures;
3) The basis on a *continuous onset detection function* for intrinsically *improving the noise robustness* of our beat tracking system against stationary noise;
4) The integration of an online ego-noise suppression scheme to *eliminate the non-stationary motor noise*;
5) A *thorough, multi-criteria based evaluation* across a wide variety of music genres and different SNR levels.

III. Ego Noise Robust Real-time Beat Tracking System

As illustrated in Fig. 1, the proposed system architecture is composed of two functional blocks: (1) a pre-processing block of online ego noise suppression [1], and (2) the actual real-time beat tracking algorithm – *IBT (INESC Beat Tracker)* [2].

A. Ego Noise Suppression

Parameterized template estimation [1] is a noise estimation method, which associates joint (motor) status data with ego noise data. In this approach, the robot predicts an arbitrary sequence of audio data from a large dataset of audio templates recorded in advance, based on the observations

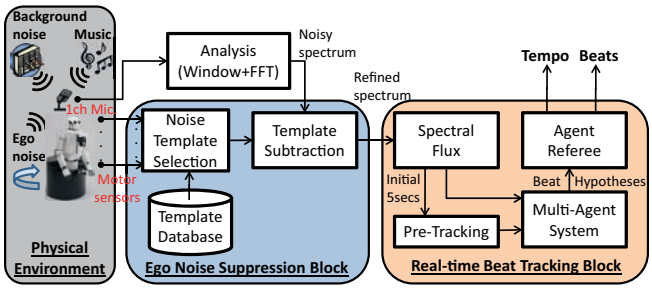


Fig. 1. Block diagram of the proposed ego noise robust real-time beat tracking system.

on the current motion. The underlying justification, why a template-based approach is suitable for ego noise, is that both the spectrum of the noise and the joint states do not change significantly when the same motion in the training session is performed again in the estimation session. This method has been already effective to suppress ego noise for various robot audition applications, such as automatic speech recognition [1] and sound source localization [16].

Technically, this method utilizes encoders attached to the motors of the robot, which measure the angular position of each joint. During the motion of the robot, actual position of each motor, $\theta(n)$, is acquired regularly at each audio frame n. Using the difference between consecutive sensor outputs, velocities, $\dot{\theta}(n)$, are calculated. Considering that J joints are active, $2J$ attributes are generated. Each feature is normalized to [0 1] so that all features have the same contribution on the prediction. The resulting feature vector has the form of $\overrightarrow{F}(n) = [\theta_1(n), \dot{\theta}_1(n), \ldots, \theta_J(n), \dot{\theta}_J(n)]$. In the template generation (database creation) phase, one feature vector is assigned to the current noise spectrum $\overrightarrow{N}(n)$ and used to label the instantaneous noise fragment; this data block $\overrightarrow{T}_n = [\overrightarrow{F}(n) : \overrightarrow{N}(n)]$ is called a *parameterized template*.

During the estimation phase, a nearest neighbor search in the database is conducted for the best matching template of motor noise for the current time instance (frame at that moment) using its feature vector label. The estimated noise is used to compute the gains of spectral subtraction and, finally, to obtain the refined audio spectrum to which the beat tracking is applied.

B. IBT Real-time Beat Tracking System

As depicted in Fig. 1, the actual beat tracking system [2] follows a classic modular architecture which assent on: (1) extraction of a midlevel rhythmic representation (*i.e.*, feature) from the refined audio signal; (2) a pre-tracking stage to induce the main tempo and beat hypotheses; and (3) the actual sequentially-driven causal beat tracking algorithm to estimate the musical beats and tempo on-the-fly. The system works in an online fashion by making beat predictions without prior knowledge (*i.e.*, without look-ahead) on the incoming signal.

1) Audio Feature Extraction: We selected the continuous spectral flux onset detection function as the midlevel representation over which all further processing is done. This feature was calculated as proposed in [17] over the

consecutive frames of the refined audio spectrum. In order to smooth the onset detection function and reduce false detections, a low-pass Butterworth filter is sequentially applied on the extracted spectral flux values.

2) Pre-Tracking: The system is initialized on an induction window, with a predefined fixed length. At the end of that pre-processing step, hypotheses regarding periods, phases and scores (P_i, ϕ_i, S_i) of a set of M initial beat agents (indexed by i) are passed along to the beat tracking algorithm. The first step in the pre-tracking stage is to compute a continuous periodicity function, based on the spectral flux autocorrelation, along time-lags. This periodicity function is parsed by an adaptive peak-picking algorithm to retrieve M global maxima, whose time-lags constitute the initial set of period hypotheses P_i. For each one of the period hypothesis P_i, a number of phase hypotheses ϕ_i^j (where j is the index of the alternative hypotheses for the i-th period hypothesis) are considered among detected onsets (also computed as proposed in [17], over the induction window). Finally, a raw score S_i^{raw} is given to each (P_i, ϕ_i) hypothesis, corresponding to the sum of time deviations between elements of the chosen beat train template and local maxima in the *spectral flux*.

3) Beat Tracking: Following the pre-tracking stage, the process of online beat tracking consists of the supervision of the incoming spectral flux values by a set of beat agents representing alternative hypotheses regarding beat positions and tempo. These agents propagate predictions with respect to their goodness-of-fit on incoming data, this way handling tempo and timing variations while keeping a good balance between reactiveness (speed of response to rhythmic changes) and inertia (stability of the system).

All agents are mediated by a central referee which incrementally evaluates their predictions with respect to their deviation (*i.e., error*) to the local maximum in the observed data, within a two-level tolerance window: an *inner* tolerance window TW_{in}, for handling short period and phase deviations; and an *outer* window TW_{out} to cope with eventual sudden expressive rhythmic changes. Consequently, if the considered local maximum m is found inside the *inner* tolerance window, the agent's period and phase are compensated by a fraction of that error.

If, on the other hand, the local maximum m is found in the *outer* tolerance window, the agent under analysis keeps its period and phase, but in order to cope for potential sudden variations of tempo and/or timing, it generates three children $\{C_1, C_2, C_3\}$. These children follow three alternative hypotheses, considering alternative possible deviations of their father's current hypothesis: timing (phase), tempo (period), or timing and tempo.

In order for the *Agent Referee* to determine the best agent at each data frame, the following evaluation function Δs evaluates the distance between each beat prediction and the respective local maximum m, inside either TW_{in} or TW_{out}:

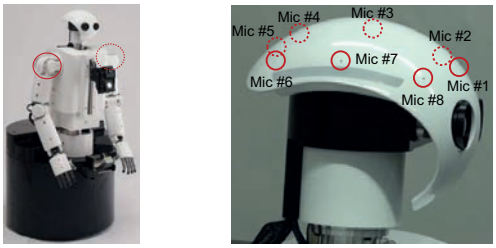
$$\begin{cases} \Delta s = \left(1 - \frac{|error|}{TW_{out}^r}\right) \cdot \left(\frac{P_i}{P_m}\right) \cdot SF(m), \exists\, m \in TW_{in} \\ \Delta s = -\left(\frac{|error|}{TW_{out}^r}\right) \cdot \left(\frac{P_i}{P_m}\right) \cdot SF(m), \exists\, m \in TW_{out}, \end{cases} \quad (1)$$

where P_m is the maximum admitted period, in frames, and SF is the spectral flux function. Finally, the actual beats outputted from the system are retrieved from the best agent at each time-frame.

IV. Experiments and Results

A. Experimental Settings

1) Hardware Specifications: The used robotic platform is a humanoid robot from *Honda Research Institute Japan (HRI-JP)*, called HEARBO, with an 8 channel omni-directional microphone array on top of its head, as illustrated on Fig. 2. The audio signals were synchronously captured by a RASP-24 bits unit, developed by System in Frontier Inc. (8 ch and 16 ch A/D converter with wired/wireless LAN connection) at a 44.1 kHz sampling rate, and transmitted at $\approx$ 86.0 Hz. The joint sensor data was acquired at $\approx$ 50.0 Hz. All processes were handled by an Intel Core i5 quadcore laptop PC at 2.53 GHz, with 8 GB of RAM.



(a) Moving joints (b) Close-up of the head

Fig. 2. HRI-JP humanoid robot HEARBO

2) Software Specifications: The audio spectrum was calculated from a single microphone input using a Complex window with 1024 samples (23.2 msec at a 44.1 kHz sampling rate), and with a 50% overlap (*i.e.*, hop size of 512 samples). We used an initial IBT's induction window of 5 sec in length, and limited the beat tracking workflow to a maximum of 30 agents. Additionally, in order to avoid metrical interchanges during the online processing of IBT we constrained all beat estimates to an octave, with tempi ranging from 80 to 160 Beats-Per-Minute (BPM). This falls within the "preferred tempo-octave" fitting the majority of tempi distributions [18]. For template subtraction, we used a minor spectral floor of 0.1 [1].

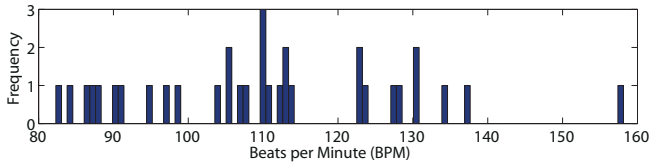
IBT was implemented on *MARSYAS (Music Analysis, Retrieval and Synthesis for Audio Signals)*[1], an open source software framework for MIR, and wrapped into *HARK (HRI-JP Audition for Robots with Kyoto University)*[2], an open-source software with functional modules for robot audition. The ego noise suppression modules and the integration of the final blocks were also implemented on HARK. Finally, the motion generation and recording processes, and the bi-directional dataflow between HARK and the robotic platform were handled by *ROS (Robot Operating System)*[3].

[1] http://marsyas.info/
[2] http://winnie.kuis.kyoto-u.ac.jp/HARK/
[3] http://www.ros.org

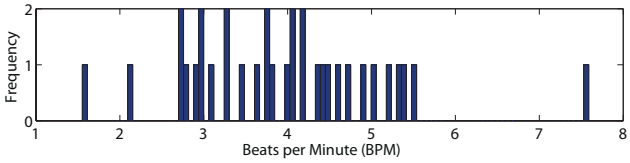
3) Recording Setup: For the recordings we used only the frontal microphone #1 of the robot (see Fig. 2(b)). The musical stimuli were played by a loudspeaker standing 2 m away in front (*i.e.*, 0°) of the robot. Audio data were recorded in a noisy room with the dimensions of 4.0 m×7.0 m×3.0 m with a reverberation time (RT_{20}) of 0.2 sec. In order to guide the recording process and strictly trim the music from the recorded audio stream, for offline evaluation purposes, we appended a pilot signal to the beginning of each music file, with a silence chunk in between (see Fig. 4).

Additionally, to better control the experimental conditions among recordings, we generated a simplistic periodic dancing motion, repeated until the end of each recorded musical stimuli. For this purpose, and in order to maximize the disturbing effects of the ego noise, we used the closest (*i.e.*, the loudest) joint to the robot's microphone – the shoulder *pitch* joint (see Fig. 2(a)) – and rotated it between −30° and 30°, back and forward. We assured the motion started in advance to music by triggering the motion performance at an arbitrary time-point, right after hearing the pilot signal. To train our template database in advance, we recorded data of 10 cycles of this ego-motion only (*i.e.*, without music).

B. System Evaluation



(a) Tempo range.



(b) Absolute standard deviation of the tempo.

Fig. 3. Histograms of tempo variation in the dataset, per musical piece.

1) Evaluation Musical Stimuli: In order to evaluate our system's robustness against different audio signal conditions, our experiments involved recordings on a musical dataset consisting of 33 pieces, with 20 sec each, uniformly chosen from 11 different genres: *classical*, *dance*, *folk*, *greek*, *hiphop*, *jazz*, *latin*, *poprock*, *rockmetal*, *samba*, and *soul*. As illustrated in Fig. 3, the dataset's musical tempi are spread from 83 to 158 BPM, with a mean of 110 BPM, and a mean standard deviation (*i.e.*, variation) of 4.0±1.2 BPM per piece. These data were picked from a beat-labeled dataset, comprising 1360 musical files used for the evaluation of beat tracking [2]. In order to evaluate the noise robustness of the system fairly, we selected our evaluation dataset from the data where IBT performed well upon the clean signal, *i.e.*, over the music file itself. (For a thorough evaluation on IBT's performance refer to [2].) These musical stimuli were played, and recorded, at three incremental SNRs of {-3.8, 0.2, 4.2} dB labeled as *low, moderate, high*. Under these conditions, the audio signals were recorded separately: 1) in the presence of only *BackGround Noise (BGN)*, and 2) in the presence of both BGN and *Ego Noise (EN)*.

2) Evaluation Measures: To evaluate the accuracy of tempo estimation and beat tracking of our system with respect to the different genres and SNR conditions, we used ground-truth beat data, manually annotated by experts. For this purpose, we propose the use of three quantitative metrics:

- *SNR Improvement*: This is a conventional metric that measures the signal's SNR before and after applying ego noise suppression.
- *Global Tempo Estimation*: It measures the difference between the estimated tempi t_e and ground-truth tempi t_{gt}, within a 4% tolerance window w_t. As such, a tempo estimation t_e is considered correct if:

$$t_e \in [t_{gt} \cdot (1 - \frac{w_t}{100}) \quad t_{gt} \cdot (1 + \frac{w_t}{100})]. \tag{2}$$

  Since IBT is restrained to the same tempo octave as the dataset, we did not need to also consider tempo estimation at metrical levels related to the ground-truth.
- *Beat Tracking Accuracy*: We used the *AMLt* (*Allowed Metrical Levels, continuity not required*) as proposed by [19], calculated as follows:

$$AMLt = max_m \left(\frac{\sum_{s=1}^{S} B_s}{G'_m} \right), \tag{3}$$

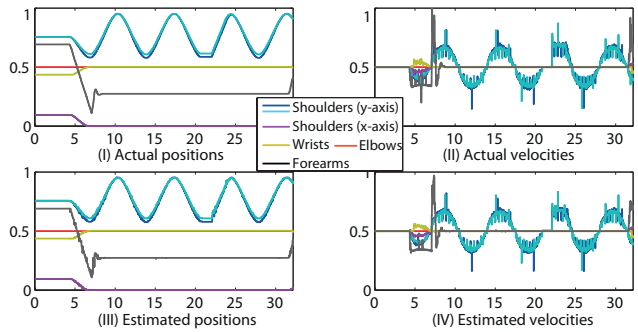
  where $G'_m = 2G \pm \pi, G, 2G, \frac{1}{2}G$ is the number of ground-truth beat-times, at π-phase or the considered metrical level (by resampling the ground-truth beat-times by different m factors), and B_s is the number of estimated beat-times b_i in each continuously correct segment s. Every estimated beat-time b_i is considered correct if:

$$\begin{cases} g_i - w_b * \Delta_{g_i} < b_i < g_i + w_b * \Delta_{g_i} \\ g_{i-1} - w_b * \Delta_{g_{i-1}} < b_{i-1} < g_{i-1} + w_b * \Delta_{g_{i-1}} \\ (1 - w_b) * \Delta_{g_i} < \Delta_{b_i} < g_i + (1 + w_b) * \Delta_{g_i} \end{cases}, \tag{4}$$

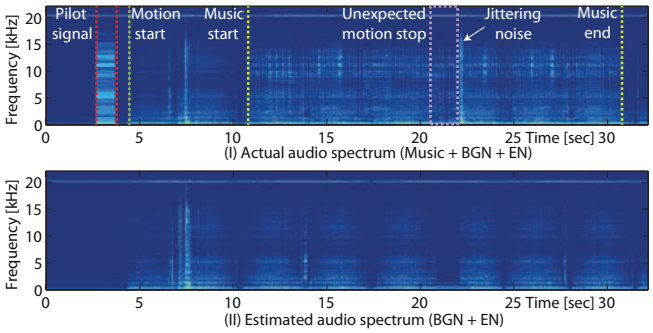
  where $\Delta_{b_i} = b_i - b_{i-1}$ and $\Delta_{gt_i} = gt_i - gt_{i-1}$ are, respectively, the estimated and ground-truth current *Inter-Beat-Interval (IBI)*, and w_b is a tolerance window set to 17.5% of the considered IBI. The choice of this particular measure relies on its invariance with respect to beat estimations at double/half the annotated tempo or at constant off-beat (the so called π-phase error). This way we could better focus our evaluation on the system's noise robustness by discarding rhythmic ambiguities in terms of phase and period that typically affect beat trackers.

C. Results

Fig. 4 illustrates the process of ego noise estimation. The upper panels show the actual robot joint states, *i.e.*, joint positions (Fig. 4(a-I)) and velocities (Fig. 4(a-II)) based on the acquired sensory data, and the actual audio spectrum (Fig. 4(b-I)) recorded in our experiment. As depicted in



(a) Actual and estimated joint states



(b) Actual recorded signal and estimated noise signal

Fig. 4. Ego noise suppression results

Fig. 4(a-III) and Fig. 4(a-IV), the estimation is performed correctly from the template database, and as a consequence we can subtract the estimated noise spectrum template (Fig. 4(b-II)) from the captured audio spectrum (Fig. 4(b-I)), respectively, to refine it. Using "parameterized templates", the system can still estimate the unexpected robot behaviors (*e.g.*, 20.5 sec-22 sec) correctly. On the other hand, the robot's mechanical system create a bursting shudder noise similar to a high-pitched click sound, which is tackled by a naive power thresholding to prevent incorrectly dominant peaks. Fig. 5 presents our system's online beat predictions (in red vertical bars, after the 5 sec induction) against the ground-truth (in green bars) for the same musical piece from Fig. 4, in which we could completely recover the correct beat-times even if there was a mechanical jittering noise (12 sec).

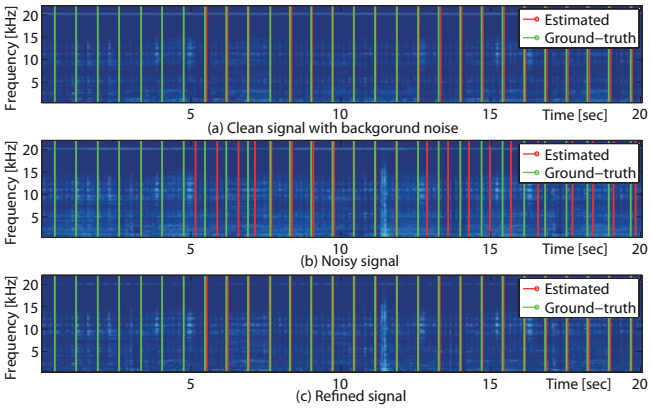


Fig. 5. Online predicted beat-times (red), against the ground-truth (green).

In terms of computational time, the system efficiently fulfills the required real-time processing. It took ≈10% of the length of music data to process the whole dataset without ego noise suppression, and ≈11% when applying the ego noise suppression as a pre-processing.

1) SNR Improvement: Table I shows the average SNR rates before and after the template subtraction.

2) Global Tempo Estimation: Fig. 6 presents accuracies obtained for our system's global tempo estimation, given by the median IBI of the final beat predictions, *i.e.* after tracking the beats of the whole musical piece.

3) Beat Tracking Accuracy: Fig. 7 presents the *AMLt* accuracy obtained for the beat-times estimated on-the-fly.

TABLE I

AVERAGE SNR RESULTS.

Label	Signal-BGN-Ratio [dB]	Signal-EN-Ratio [dB]	Processed SNR [dB]
Low	14.7	-3.8	**3.3**
Moderate	20.1	0.2	**4.6**
High	24.6	4.2	**6.8**

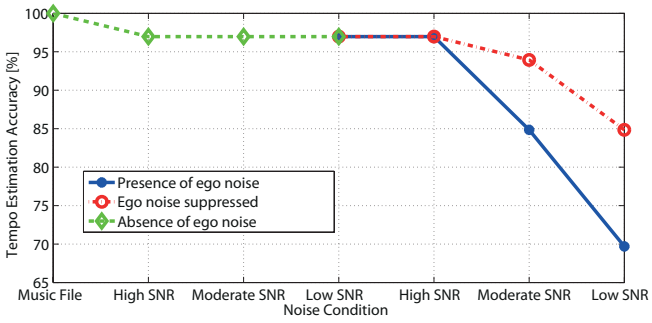


Fig. 6. Average tempo estimation results under different noise conditions.

D. Discussion

1) SNR Improvement: As described in Table I, using template subtraction, we achieved a 4.7±2.3 dB improvement on the SNR of the refined signals across all SNR levels. As in all single channel noise suppression methods, the improvements tend to get higher as the initial SNR drops.

2) Global Tempo Estimation: As depicted in Fig. 6, our system's tempo estimation scored 100% on the original music files. This is expected, since, as referred, we purposely selected our evaluation dataset from the data on which IBT performed well for the clean signals. Yet, although being manually selected, these results evince the capability of IBT

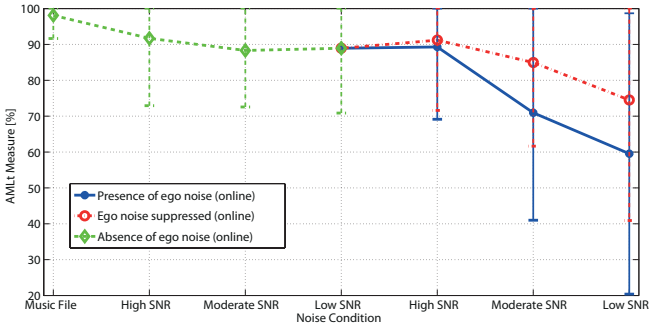


Fig. 7. AMLt beat tracking accuracies under different noise conditions.

to handle different genres and tempo variations (see Sec. IV-B.1). This is even supported by the still high lowest accuracy of the IBT (69.7%), despite the difficult SNR conditions of -3.8 dB (*Low SNR*) without ego noise suppression. In the presence of background noise only (*i.e.,* absence of ego noise), we observe a decrease of 3.0 points in tempo estimation, which is kept constant among all SNR conditions. This clearly indicates that our system is not significantly influenced by stationary noise (BGN), due to its intrinsic noise robustness supported by the use of a *continuous onset detection function*. On the contrary, the robustness of IBT alone against ego noise proportionally decreases when the SNR gets lower. Yet, it seems that, in some extent (at least, for SNR=4.2 dB), the system can still cope with non-stationary noises, keeping the same tempo estimation performance as in the absence of ego noise. As a final claim, we clearly see the improving effect of our ego noise suppression scheme, increasing the tempo estimation results in 8.2 ± 7.7 points, across SNRs. Besides, and as desirable, its effect is greater when the SNR gets lower, contributing with a maximum 15.2 points improvement at *Low SNR*.

3) Beat Tracking Accuracy: Similarly to the tempo estimation results, and as illustrated in Fig. 7, on the music files IBT scored well (98.1%) in terms of online beat-times prediction. This additionally confirms that, despite the considerable extent of timing/tempo changes, at a maximum of 7.6 BPM difference between two consecutive IBIs from our data (see Sec. IV-B.1), IBT could still correctly adapt supported by its *sequentially-driven* causal beat tracking algorithm. Also identical to tempo estimation, the beat tracking results in the BGN seem to be uniform across all SNRs with a maximum 3.4 points variation, and only diverging from the music file results by an average 8.4 ± 1.8 points. Again, at 4.2 dB (*High SNR*) the system could cope with ego noise also for beat prediction, diverging from the beat tracking results in the absence of ego noise for the same SNR conditions, by only 2.4 points, without ego noise suppression. As expected, the ego noise suppression scheme also improved the online beat prediction, and the improvement is greater when lowering the signal's SNR. As observed in Fig. 7, when subtracting the estimated ego noise template our system's beat tracking results improved by 10.3 ± 7.3 points, across all SNRs, with a maximum of 15.0 points at -3.9 dB (*Low SNR*).

As final remarks, we must point out the limitations of our approach. First of all, the training and test motions should be the same. If this condition is met, the method easily scales to a humanoid robot with more than 20 joints (see [1]). Secondly, our system is still susceptible to instantaneous bursting noises. The presence of unexpected transient noises (*e.g.*, mechanical jittering and shuddering sounds) is extremely problematic to beat tracking systems in general, contributing as a powerful false-estimation which bias the system from that point on. Transient noises during the training session also cause incorrect template estimates such as around $t = 7.5$ and $t = 14$ in Fig. 4(b-II). In order to better tackle such bursting noises, a more advanced noise suppression scheme, or a more intelligent beat tracking algorithm (*e.g.*, making beat estimations based on a running SNR confidence) is required.

V. Conslusion

In this paper we presented an online audio beat tracking system enhanced with high robustness against the ego-motion noise of a dancing robot. The proposed system combined a real-time beat tracking algorithm with a parameterized template subtraction in a single framework. We evaluated our proof-of-concept system in a real environment and showed that our integration method achieves: 1) high performance on ego noise suppression, 2) improved estimation of tempo, and 3) more accurate beat tracking performance. In future work, we plan to improve the robustness of our system and apply it for the generation of beat-synchronous robot dancing motions in real-time.

References

[1] G. Ince *et al.*, "Whole Body Motion Noise Cancellation of a Robot for Improved Automatic Speech Recognition," *Advanced Robotics*, vol. 25, no. 11, pp. 1405–1426, 2011.

[2] J. L. Oliveira *et al.*, "IBT: A Real-time Tempo and Beat Tracking System," in *Int. Society for Music Information Retrieval Conference*, 2010, pp. 291–296.

[3] J.-J. Aucouturier *et al.*, "Cheek to Chip: Dancing Robots and AI's Future," *IEEE Intelligent Systems*, vol. 23, no. 2, pp. 74–84, 2008.

[4] G. Weinberg, *Robotic Musicianship – Musical Interactions Between Humans and Machines*. I-Tech Education and Publishing, 2007, no. September, ch. Robotic Musicianship, p. 22.

[5] M. P. Michalowski, S. Sabanovic, and H. Kozima, "A Dancing Robot for Rhythmic Social Interaction," in *ACM/IEEE HRI*, 2007, pp. 89–96.

[6] C. Crick, M. Munz, and T. Nad, "Robotic Drumming: Synchronization in Social Tasks," in *IEEE RO-MAN*, 2006, pp. 97–102.

[7] J.-J. Aucouturier, Y. Ogai, and T. Ikegami, "Making a Robot Dance to Music Using Chaotic Itinerancy in a Network of FitzHugh-Nagumo Neurons," in *Neural Information Processing*, 2007, pp. 647–656.

[8] R. Ellenberg, D. Grunberg, Y. Kim, and P. Y. Oh, "Exploring Creativity through Humanoids and Dance," in *Int. Conf. on Ubiquitous Robotics and Ambient Intelligence*, 2008, p. 6.

[9] K. Yoshii *et al.*, "A Biped Robot that Keeps Steps in Time with Musical Beats while Listening to Music with Its Own Ears," in *IEEE/RSJ IROS*, 2007, pp. 1743–1750.

[10] T. Mizumoto *et al.*, "A Robot Listens to Music and Counts its Beats aloud by Separating Music from Counting Voice," in *IEEE/RSJ IROS*, 2008, pp. 1538–1543.

[11] K. Murata *et al.*, "A Robot Uses Its Own Microphone to Synchronize Its Steps to Musical Beats While Scatting and Singing," in *IEEE/RSJ IROS*, 2008, pp. 2459–2464.

[12] M. Goto and Y. Muraoka, "A Real-time Beat Tracking System for Audio Signals," in *Int. Computer Music Conf.*, 1995, pp. 171–174.

[13] R. Takeda *et al.*, "Exploiting Known Sound Source Signals to Improve ICA-based Robot Audition in Speech Separation and Recognition," in *IEEE/RSJ IROS*, 2007, pp. 1757–1762.

[14] D. K. Grunberg, D. M. Lofaro, P. Y. Oh, and Y. E. Kim, "Robot Audition and Beat Identification in Noisy Environments," in *IEEE/RSJ IROS*, 2011, pp. 2916–2921.

[15] A. Klapuri *et al.*, "Analysis of the Meter of Acoustic Musical Signals," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 14, no. 1, pp. 342–355, Jan. 2006.

[16] G. Ince *et al.*, "Assessment of General Applicability of Ego Noise Estimation," in *ICRA*, 2011, pp. 3517–3522.

[17] S. Dixon, "Onset Detection Revisited," in *Int. Conf. on Digital Audio Effects*, 2006, pp. 133–137.

[18] D. Moelants, "Dance Music, Movement and Tempo Preferences," in *5th Triennial ESCOM Conference*, Hannover, Germany, 2003, pp. 649–652.

[19] M. E. P. Davies, N. Degara, and M. D. Plumbley, "Evaluation Methods for Musical Audio Beat Tracking Algorithms," *Technical Report C4DM-TR-09-06*, p. 17, 2009.