2012 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 7-12, 2012. Vilamoura, Algarve, Portugal

Live Assessment of Beat Tracking for Robot Audition

João Lobato Oliveira[1,2,4], Gökhan Ince[3], Keisuke Nakamura[3], Kazuhiro Nakadai[3],
Hiroshi G. Okuno[4], Luis Paulo Reis[1,5], and Fabien Gouyon[2]

***Abstract*— In this paper we propose the integration of an online audio beat tracking system into the general framework of robot audition, to enable its application in musically-interactive robotic scenarios. To this purpose, we introduced a state-recovery mechanism into our beat tracking algorithm, for handling continuous musical stimuli, and applied different multi-channel preprocessing algorithms (e.g., beamforming, ego noise suppression) to enhance noisy auditory signals lively captured in a real environment. We assessed and compared the robustness of our audio beat tracker through a set of experimental setups, under different live acoustic conditions of incremental complexity. These included the presence of continuous musical stimuli, built of a set of concatenated musical pieces; the presence of noises of different natures (e.g., robot motion, speech); and the simultaneous processing of different audio sources on-the-fly, for music and speech. We successfully tackled all these challenging acoustic conditions and improved the beat tracking accuracy and reaction time to music transitions while simultaneously achieving robust automatic speech recognition.**

I. Introduction

When listening to various auditory scenes one must simultaneously process and understand different sound sources mixed together into a single audio *cocktail* while dealing with noises of different natures [1]. To reproduce this kind of complex reasoning in artificial machines, such as robots, Computational Auditory Scene Analysis (CASA) algorithms must be able to localize, separate and enhance various kinds of continuous acoustic signals (*e.g.*, speech, music) in real unconstrained (*i.e.*, noisy) environments while applying signal processing algorithms on-the-fly according to specific perceptual tasks. Thus, musically-aware robots interacting with humans in real-world scenarios must address the same concerns of CASA while applying real-time Music Information Retrieval (MIR) algorithms.

In this paper we introduce a state-recovery mechanism into our online beat tracker in order to rapidly recover from signal losses and abrupt music transitions in continuous musical stimuli. Furthermore, we propose to integrate an audio beat tracking algorithm [2] with different multi-channel preprocessing strategies (*e.g.*, Sound Source Localization (SSL), Sound Source Separation (SSS), ego noise suppression) to enhance the quality of the captured audio signal. We assess the robustness and performance of the proposed audio beat tracking system through a set of live experimental setups with different acoustic conditions of incremental complexity to verify its applicability and compatibility into the general framework of robot audition.

This work was partially supported by SFRH/BD/43704/2008 PhD scholarship endorsed by the Portuguese Government through FCT.

[1] Artificial Intelligence and Computer Science Laboratory (LIACC) – FEUP, Porto, Portugal. (joao.lobato.oliveira@fe.up.pt)

[2] Institute for Systems and Computer Engineering of Science and Technology (INESC TEC), Porto, Portugal.

[3] Honda Research Institute Japan Co., Ltd., Saitama, Japan.

[4] Department of Intelligence Science and Technology, Graduate School of Informatics, Kyoto University, Kyoto, Japan.

[5] University of Minho, School of Engineering - DSI, Guimarães, Portugal.

II. Related research

Robotic musical instruments have been designed for decades by creative scientists from art and entertainment industry, which make use of sensorimotor algorithms and proper mechanical designs recurring to motors, solenoids and gears to create multiple forms of music [3]. Musically expressive robots are however a more recent story, that sets back to the 80's with the first instrument robotic players [4]. Since then, worldwide researchers are determined to apply all kinds of "off-the-shelf" human control interfaces (*e.g.*, acceleration sensors, sonars, infra-reds, and wireless gesture controls) towards building fully autonomous robots and entire robotic bands [5] that can act together and interact with human musicians and dance performers. Yet, this so-called "robotics musicianship" [6] is still taking its first steps and more effort is still needed to be put on fundamental qualities of musical interaction (*e.g.*, improvisation/imitation, expression/emotion, anticipation/synchronization) and most especially on robust real-time reasoning of high-level musical qualities for robot audition (*e.g.*, beat, tempo, meter, pitch, genre, tonality, texture, melody) in real-world noisy scenarios. Only a few attempts have been made recently to implement and assess these perceptual musical modules in live conditions and most of them do not go beyond note onset detection, tempo and beat tracking in simplified/restrictive conditions. Weinberg *et al.* [7] and Mizumuto *et al.* [8] followed different approaches for online beat tracking on human drum performances. Both methods were applied for human-robot musical ensembles in order to detect the human's drum-beat and lead their robots into synchronized and/or improvised interactions through drum [7] or theremin [8] performances. Murata, Mizumuto, Otsuka *et al.* [9]–[11] took a step further and used two different beat trackers for processing live musical signals while stepping [9], scatting [9], beat-counting [10], and singing [9], [11] in synchrony (*i.e.*, through feedback-control) to the musical beat [9], [10], tempo [9] or score position [11]. In order to suppress the robot's self-voice from the captured auditory signals, all authors used a one- [10], [11] or two- [9] channel versions of a semi-blind Independent Component Analysis (ICA)-

based adaptive filter that performs spectral subtraction on the captured (mixed) audio based on the clean signals of the generated voice. Similarly, Otsuka *et al.* [12] applied the same beat tracking procedure with ICA-based filter they previously used in [11] to synchronize a theremin playing robot while suppressing the generated theremin sounds.

Ultimately, four different studies so far used audio beat tracking in live experiments in the presence of robot motor noise. The first two, presented by Yoshii, Murata *et al.* [9], [13], applied a real-time beat tracker to synchronize the stepping of a humanoid robot to the estimated beat-times of captured musical stimuli. Yet, both assumed that the stepping noise did not affect the beat predictions, since the motion was in phase with the beat. The latter two studies, presented by Grunberg *et al.* [14] and Oliveira *et al.* [15], applied different strategies to suppress motor noise generated from random [14] and/or periodic [14], [15] motions of humanoid robots, while estimating the beat-times of a set of musical pieces on-the-fly. For suppressing the motor noise from a single-channel audio input, Grunberg *et al.* applied (and compared) a static and an adaptive filter for spectral subtraction using separate attenuation thresholds for each spectral frequency bin. On the other hand, Oliveira *et al.* utilized a template-based ego noise suppression scheme which associates joint (motor) status data with ego noise data, recorded in advance, to estimate the gains of spectral subtraction and obtain a refined audio spectrum of the single-channel signal. Both strategies were able to improve the noise-robustness of the assessed beat trackers for application on musical performing and dancing robots in live, real-world conditions.

In this paper, we propose to extend our latter approach [15] for its application on musically-interactive robotic systems in real-world acoustic scenarios. To this purpose, we assessed the performance and robustness of our beat tracker under different live acoustic conditions, and through different CASA strategies for robot audition:

- **Multiple audio sources of different kinds:** use of *SSL* and *SSS* methods to retrieve and separate the active sound sources (*i.e.*, music and speech) on-the-fly;
- **Multiple noises of different natures:** use of *multi-channel beamforming* and *multi-channel ego noise suppression* methods to improve the quality of the acquired audio signal against stationary and non-stationary noises of multiple natures (*e.g.*, robot fans, robot motion, speech).
- **Continuous musical stimuli of different musical pieces:** use of a *state-recovery* mechanism to recover the beat tracker state whenever there is indications that the tracking system lost track of reliable beat predictions (*e.g.*, at transitions between musical pieces, or when the SSL mechanism fails to detect the musical source).
- **Multiple evaluation criteria of different tasks:** assess multiple perceptual tasks running simultaneously (*i.e.*, beat tracking and ASR).

III. System Overview

As illustrated in Fig. 1, the proposed system architecture is composed of three main functional blocks: *i)* a multi-channel preprocessing block consisting of SSL, SSS, and ego noise suppression algorithms; *ii)* a speech processing block performing ASR; and *iii)* a music processing block consisting of the integrated audio beat tracking system.

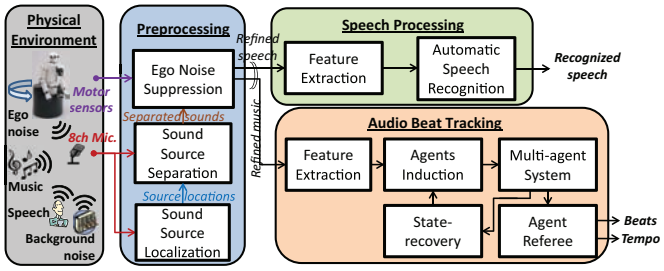


Fig. 1. Overview of the system architecture.

A. Preprocessing and speech processing

In the *preprocessing block*, the recorded audio signals are first subject to SSL, which passes the location of each sound source to the SSS module. Because separated signals still contain diffuse ego noise, we apply sound enhancement relying on template-based multi-channel ego noise estimation that utilizes the angular state of the robot joints. The difference between the current ego noise suppression and previous single-channel noise suppression system we used in [15] is that it is able to separate the overall ego noise among all separated sound sources. By doing so, spectral subtraction can be applied on the audio spectrum of each individual sound source (*e.g.*, music, speech) using its corresponding ego noise spectrum. The details of this block can be found in our complementary paper [16]. In addition, a power threshold filter was applied atop of this ego noise suppression scheme for handling unpredictable robot noises (*e.g.*, jittering).

The outputs of the preprocessing, namely the refined speech and music spectra are sent to speech and music processing blocks. In the *speech processing block*, we extract 13 static Mel-Scale Log Spectrum (MSLS) features, 13 delta MSLS features and 1 delta power feature and send them to the real-time ASR engine, which is based on Julius.

B. Audio beat tracking

The used online *audio beat tracking* system, IBT, was first proposed in [2] and used in [15]. The algorithm is based on a multi-agent architecture composed of (see Fig. 1): *i)* an *audio feature extraction* module that parses the preprocessed audio data into a mid-level rhythmic feature; followed by *ii)* an *agents induction* module, which (re-)generates the initial and new sets of hypotheses regarding possible beat periods and phases; and followed by *iii)* a multi-agent-based *beat tracking* module, which propagates hypotheses, proceeds to their online creation, killing and ranking, and outputs beats on-the-fly without prior knowledge (*i.e.*, without look-ahead) on the incoming signal. In addition, the current implementation of IBT extends the one used in [15] by integrating *iv)* a *state-recovery* mechanism responsible for supervising the beat tracking analysis of the signal and, if needed, recover the state of the beat tracker by resetting the multi-agent system with re-inductions of beat and tempo.

This mechanism, created to contend with situations that might require the state recovery of our beat tracking system (*e.g.*, music transitions in a continuous data stream), looks for abrupt changes in the score evolution of the current best agent (which leads the system's current beat predictions) as an indication that the algorithm had lost track of reliable beat hypotheses. This monitoring runs at time increments of $t_{hop} = 1\,s$ and it looks for the variation $\delta\overline{sb}_n$ of the current mean chunk of measurements of the best score $\overline{sb}_n$ in comparison to the previous $\overline{sb}_{n-t_{hop}}$, as follows:

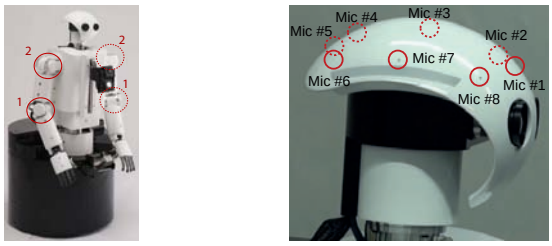
$$\delta\overline{sb}_n = (\overline{sb}_n - \overline{sb}_{n-t_{hop}}) * \overline{sb}_n : \overline{sb}_n = \frac{1}{W}\sum_{w=n-W}^{W} sb(n-w), \quad (1)$$

where n is the current time-frame, $W = 3\,s$ is the size of the considered chunk of best score measurements, and $sb(n)$ is the best score measurement at frame n.

IV. Experimental settings

A. Hardware specifications

Our experiments were run on HEARBO, a humanoid robot from *Honda Research Institute Japan (HRI-JP)* (see Fig. 2(a)). HEARBO integrates an 8-channel omni-directional microphone array on top of its head (see Fig. 2(b)). All audio signals were synchronously captured from the 8 channels, at a 16 kHz sampling rate. All recordings and evaluation procedures were processed on an Intel Core i7 quadcore PC at 2.3 GHz, with 16 GB of RAM.



(a) Positions and number of moving joints. (b) Close-up of the head.

Fig. 2. HRI-JP humanoid robot HEARBO.

B. Software specifications

All system's modules were implemented and integrated into *HARK (HRI-JP Audition for Robots with Kyoto University)*. The robot control and communication were handled by *ROS (Robot Operating System)*. The dataflow of the whole system was run at time increments of 10 ms, using a Complex window of 512 samples and 32% overlap (*i.e.*, hop size of 160 samples) for computing the audio spectrum.

The SSL was based on MUltiple Signal Classification (MUSIC) [17], and for SSS we applied Geometric High-order Decorrelation-based Source Separation (GHDSS) [18].

For template subtraction we used a spectral floor of 0.1.

IBT was set with an induction window of 5 sec in length, and constrained to a tempo octave ranging from 80 to 160 beats-per-minute (bpm), which falls within the "preferred tempo-octave" and fits the majority of tempi distributions [19]. This restriction was to avoid metrical-level interchanges that would compromise the beat tracking evaluation. Finally, according to eq. (1) a new induction of the system is requested if $\delta\overline{sb}_{n-1} \geq 0 \wedge \delta\overline{sb}_n < 0$.

C. Auditory signals

1) Musical stimuli: To reproduce the realistic scenario of continuous musical stimuli, we concatenated a set of individual musical excerpts into a music stream without any gaps. We selected 31 beat-annotated music excerpts from the dataset used in [2]. (Note that the selected data was different from the one used in [15].) The data comprised 7 different genres: *pop*, *rock*, *jazz*, *hiphop*, *dance*, *folk*, and *soul*; with tempi ranging from 81 to 140 bpm, with a mean 109±17.6 bpm, and all with a $\frac{4}{4}$ meter. So that the evaluation focuses on the specific ability of the system to cope with abrupt signal changes, caused by transitions between musical pieces, the 31 pieces were selected from a sub-set of data restricted by the following two conditions:

- *Stable data*: musical pieces with low varying tempi among all Inter-Beat-Intervals (IBI), on which the maximum IBI variation did not exceed the mean IBI by more than 40%.
- *Reliable data*: music files on which IBT scored 100% in beat tracking accuracy, with AMLt (see Section IV-E).

To maximize the disturbing effect of the music transitions, the selected pieces were trimmed and concatenated considering two conditions:

- *Abrupt shifts of beat-timing at transitions*: each individual musical piece was trimmed between the time-point t_i of an arbitrary annotated beat-time and the time-point given by $t_f = t_i + b_f + 0.25IBI_f$, where b_f is the first annotated beat time 20 s after t_i, $IBI_f = b_{f+1} - b_f$, and b_{f+1} is the first annotated beat time after b_f.
- *Significant tempo differences at transitions*: the concatenated excerpts were randomly organized while ensuring a ratio of tempo between consecutive excerpts in the range of [10-54.4]%.

This process resulted in a continuous music data stream with a total length of ≈10 min consisting of 31 excerpts (*i.e.*, 30 transitions) of ≈20 sec each. We generated a beat annotation sequence for the created data stream by mapping and concatenating the annotated beats of each excerpt accordingly.

2) Speech data: The speech data was recorded by us and consisted of 8 audio files with the utterances of 4 male and 4 female Japanese speakers used in a typical human-robot interaction dialog. Each audio file was constituted by a set of 236 different Japanese words concatenated into continuous streams, with a silence gap of ≈1 sec in between them.

D. Periodic dance motions

For measuring the effect of ego-motion noise in its most challenging condition we considered robot dancing motion, as the most complex kind of musically expressive movement. To this purpose, we created 3 different periodic dance motions. Each of them was defined by 2 key-poses to be successively interpolated (*i.e.*, transited) during motion generation. In order to increase the disturbing effects of the robot's ego noise, the dance motions were designed to simultaneously move 6 joints: the shoulders *pitch* and *yaw*,

and the elbows *pitch* (see Fig. 2(a)); each with a rotational variation in the range of [10-20]° to maximize the number of transitions. During recordings the dance motions were continuously generated into a full dance sequence by using a uniform number of periodic repetitions of the 3 dances. The periodic dances were generated at random tempi (*i.e.*, random velocities) in the octave of 40 to 80 bpm, which represent the maxima motor-rate frequencies achievable by our robot.

E. Evaluation criteria

1) Beat tracking accuracy: The beat tracking accuracy was measured against the beat-annotation (*i.e.*, groundtruth) of the generated music data stream. We relied on the AMLt (Allowed Metrical Levels, continuity not required), as described in [20], for being the most permissive continuity-based beat tracking evaluation measure that considers beats estimated at double and half the tempo, or in the off-beat (π-phase error) as also correct. This metric considers the total number of correct pairs of estimated beats with a tolerance of $\pm 17.5\%$ around each pair of annotated beats. To better identify the effect of the music transitions in the beat tracking accuracy, we propose two variants of AMLt: $AMLt_{stream}$, which measures the accuracy over the whole stream, discarding the initial 5 secs of data needed for the first induction of the system; and $AMLt_{excerpts}$ that simulates the evaluation over all individual excerpts by measuring the accuracy of the whole stream but discarding the first 5 secs after each music transition.

2) Reaction time (r_t): This metric measures the time of reaction taken to recover from music transitions. It is defined as the time difference, in seconds, between the timing of the transition and the beat-time of the first four continuously correct estimated beats in the considered musical excerpt. In addition, a transition is considered successful if r_t is less than the duration of the considered musical excerpt, *i.e.*, if the system is able to recover the track of the beat at some point after transiting to the current musical excerpt.

3) ASR accuracy: Speech recognition results are given as average Word Correct Rate (WCR), which is defined as the number of correctly recognized words from the test set divided by the number of all instances in the test set.

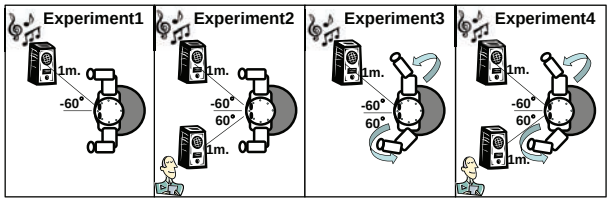


Fig. 3. Experiments for the four proposed real-world acoustic conditions.

V. Experiments and Results

As illustrated in Fig. 3, we created four real-world experimental conditions to lively assess our audio beat tracking system in incremental levels of acoustic complexity:

- **Experiment1:** live audio beat tracking.
- **Experiment2:** simultaneous live audio beat tracking and automatic speech recognition.
- **Experiment3:** live audio beat tracking during robot dancing motion.
- **Experiment4:** simultaneous live audio beat tracking and automatic speech recognition during robot dancing motion.

In all experiments the musical stimulus was played from a single loudspeaker standing at -60° and 1 m away from the robot position. The music signals were recorded with decreasing Music-Signal-to-Noise Ratio (M-SNR) among the four experiments, using the recording of *experiment1* as a baseline: *M-SNR*$= 1\,dB$ for *experiment2*, *M-SNR*$= 0\,dB$ for *experiment3*, and *M-SNR*$= -2\,dB$ for *experiment4*. For the experiments using speech stimuli (*i.e.*, *experiment2*, and *experiment4*) we played it from a second loudspeaker standing at 60° and also 1 m away from the robot. The speech signals were recorded with a segmental-Speech-SNR (S-SNR) of $0\,dB$ on *experiment2* and $-3\,dB$ on *experiment4*.

All recordings were processed in a noisy room environment with the dimensions of 4.0 m x 7.0 m x 3.0 m and a Reverberation Time (RT_{20}) of 0.2 sec. For training our ASR module we used matched acoustic models trained with a Japanese Newspaper Article Sentences (JNAS) corpus with 60-hours of speech spoken by 306 male and female speakers.

The template database for ego noise suppression was created by generating 5 min of the 3 periodic dance motions at random tempi, as described in Section IV-D.

A. Compared variants of the system

In order to demonstrate the capability of the proposed system under the presented experimental conditions we evaluated and compared the beat tracking and ASR accuracies using different input signals, resultant from different preprocessing strategies:

- `AF`: audio stream file.
- `1C`: audio captured from a single (frontal #1 – see Fig. 2(b)) microphone.
- `CE`: `1C` refined by ego noise suppression.
- `FB`: audio signal after applying fixed beamforming on the audio captured by an 8-channel microphone array.
- `FE`: `FB` refined by ego noise suppression.
- `SS`: separated audio signal, captured from an 8-channel microphone array.
- `SE`: `SS` refined by ego noise suppression.

In addition, to clearly observe the effect of the state-recovery mechanism to contend with continuous musical stimuli, we simultaneously assessed three variants of IBT:

- `IBT-default`: IBT with a single induction on the beginning (*i.e.*, first 5 sec) of the signal's analysis.
- `IBT-transitions`: IBT applying the state-recovery of the system exactly, and only, at the time-points of each annotated music transition.
- `IBT-recovery`: the implementation of IBT using the state-recovery mechanism as proposed in Section III-B.

B. Results

1) Audio beat tracking: Fig. 5 presents a 20 sec excerpt of the `1C` music only signal for *experiment1* (Fig. 5(a)) and of the `1C` (Fig. 5(b)) and `SE` (Fig. 5(c)) signals of

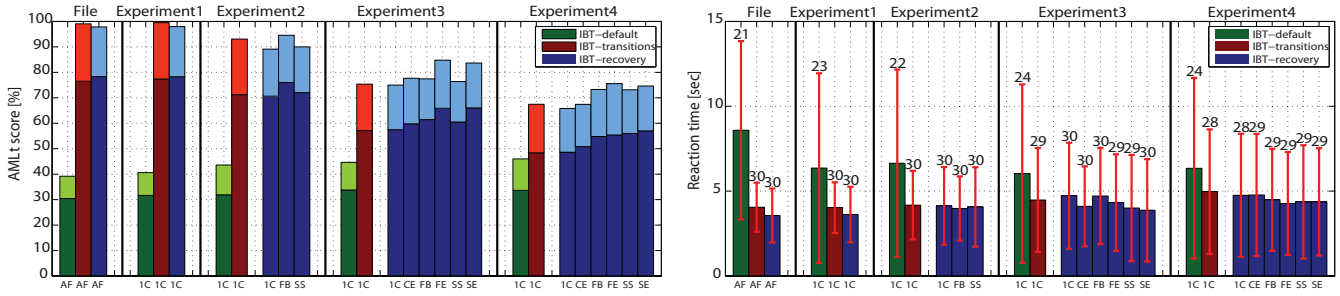


Fig. 4. Beat tracking results: (a) AMLt score: $AMLt_{stream}$ (dark) and $AMLt_{excerpts}$ (light); (b) Reaction time (r_t) and number of successful transitions atop.

the same 20 sec excerpt for *experiment4*. Fig. 5(b) and Fig. 5(c) additionally respresent the beats estimated by `IBT-recovery` (in red), respectively under `1C` and `SE` conditions, against the groundtruth (in yellow). Moreover, Fig. 5(c) depicts two important situations: *i)* a reaction time of ≈2 sec for recovering from a music transition (see 159-161 sec), and *ii)* a set of beats getting affected (see 164-167 sec) after an unpredictable jittering noise (occurred at 163 sec), when no power threshold is applied atop of ego noise suppression. Fig. 4 presents the beat tracking AMLt scores and reaction time results achieved among all variants of the system, for all experiments. The results of *experiment2* and *experiment4* represent the mean among the 8 speakers.

2) ASR: Fig. 6 presents the mean word correct rate for the ASR among the 8 speakers achieved on *experiment2* (Fig. 6(a)) and *experiment4* (Fig. 6(b)), by applying different preprocessing strategies.

VI. Discussion

A. On handling continuous musical stimuli

The overall results suggest that a continuous musical stimuli scenario is a highly challenging situation for real-time beat tracking systems to contend. As observed in Fig. 4, `IBT-default` performed poorly in all experiments, and even on the audio stream file (AF) itself. Across all experiments and preprocessing variants of the system, `IBT-default` managed to handle only a mean of 76% of the music transitions, at a mean r_t of 6.8±5.4 sec. This resulted in a mean score of 32.6% in $AMLt_{stream}$ and 42.8% in $AMLt_{excerpts}$, which is a significant drop when compared to the 100% score obtained over the audio files of each selected excerpt in the stream. Yet, when introducing the state-recovery mechanism, in the audio stream file and in *experiment1* `IBT-recovery` was able to recover almost to the original 100% $AMLt_{excerpts}$ score, and to the level of `IBT-transitions` among all experiments and preprocessing variants. Moreover, `IBT-recovery` in `1C` obtained a mean gain of 34.4 points (pts) in $AMLt_{stream}$ and 42.3 pts in $AMLt_{excerpts}$ when compared to `IBT-default`, and achieved a mean reaction time of 4.2±2.5 sec, and 100% successful transitions. This reaction time is even lower than the one achieved with `IBT-transitions` under most conditions and than the 5 secs that IBT requires for induction.

B. On handling multiple noise sources

As observed in the results of *experiment2* (see Fig. 4), and as expected, the disturbing effect of speech alone as a noise source for audio beat tracking was rather small. For `1C` it caused a mean drop of 7.6 pts in $AMLt_{stream}$ and 8.9 pts in $AMLt_{excerpts}$ when compared to *experiment1*. In addition, `IBT-recovery`'s accuracy was also slightly improved by 5.4 pts and 1.5 pts in $AMLt_{stream}$ and 5.5 pts and 0.9 pts in $AMLt_{excerpts}$ respectively with `FB` and `SS`. On the other hand, the effect of music as a noise source for ASR greatly affected its performance leading it to a poor word correct rate of 16.7%. Yet, we could significantly improve the ASR results when applying fixed beamforming (`FB`), and an additional improvement when applying sound-source localization and separation (*i.e.*, `SS`), to a total gain of 48 pts with the latter.

Regarding *experiment3*, and also as expected, ego-motion noise played greater disturbance as a noise source for beat tracking. In comparison to *experiment1* `IBT-recovery` in `1C` presented a drop of 23.0 pts in $AMLt_{stream}$ and 20.7 pts in $AMLt_{excerpts}$. When only applying beamforming (*i.e.*, `FB`) we enhanced these results up to 4.0 pts in $AMLt_{stream}$ and 2.4 pts in $AMLt_{excerpts}$. Moreover, by additionally applying ego noise suppression (*i.e.*, `FE`) we outperformed `1C` by 9.9 pts in $AMLt_{stream}$ and 8.4 pts in $AMLt_{excerpts}$.

Ultimately, in *experiment4* we observed a similar trend as in *experiment3* across the different system's variants. Yet, due to the additional disturbance of speech the results dropped on average 8.9 pts in $AMLt_{stream}$ and 9.2 pts in $AMLt_{excerpts}$ in `1C`, which is akin to the drop of *experiment2* in comparison to *experiment1*. Again, by applying beamforming we were able to sum the enhancing effect achieved with the same preprocessing on *experiment2* and *experiment3*, to a maximum of 7.5 pts in $AMLt_{stream}$ and 6.2 pts in $AMLt_{excerpts}$. Furthermore, we overcame some of the disturbance caused by ego-motion noise, by a maximum of more 1.4 pts in $AMLt_{stream}$ and 2.3 pts in $AMLt_{excerpts}$, achieved in `FE`. Although ego noise suppression improved the beat tracking accuracy its effect was quite less significant than the obtained in [15]. This is justified by the use of more complex (*i.e.*, noisier) robot motions, at varying and unpredictable tempi, that caused inaccuracies in the template predictions of our ego noise suppression algorithm. In addition, the abrupt motion transitions lead to enormous unpredictable noise bursts caused by mechanical jittering and shuddering sounds (Fig. 5(b) – 163 sec) that created spurious magnitude peaks in the spectrum. Some of these peaks were successfully filtered out by the power thresholding mechanism proposed in [15].

On the other hand, since ASR uses spectral features (*e.g.*, MSLS), on which ego noise suppression is more effective, it significantly improved the ASR accuracy by a mean 14.8 pts.

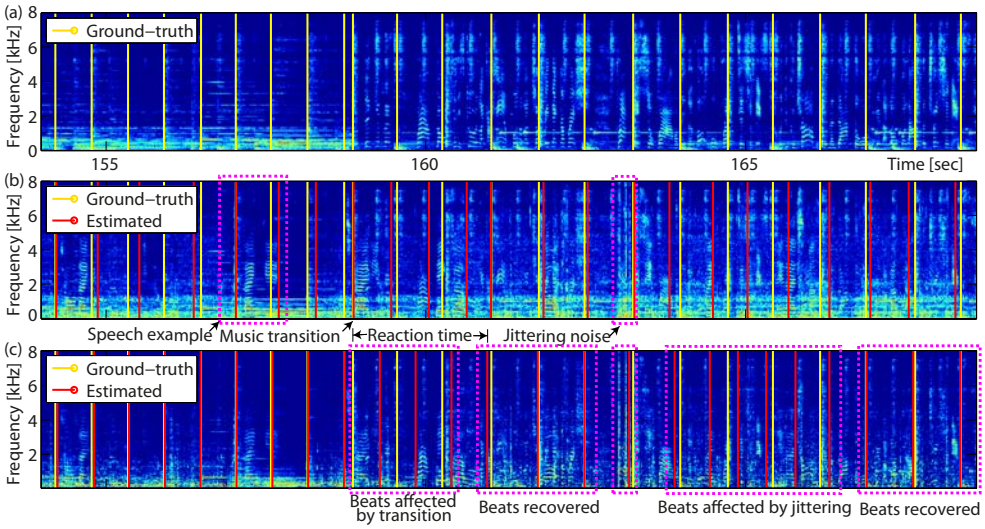


Fig. 5. Excerpt of 20 sec of the recorded/preprocessed signals for: (a) `1C` on *experiment1*; (b) `1C` on *experiment4*; (c) `SE` on *experiment4*. The beats in red were estimated by `IBT-recovery` under the respective conditions.

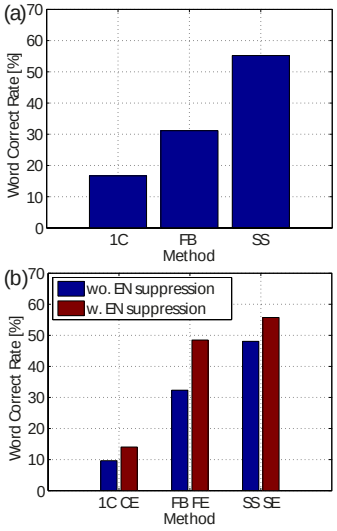


Fig. 6. ASR results for: (a) experiment2; (b) experiment4.

C. On processing multiple audio sources simultaneously

In order to automatically and efficiently process multiple audio sources of different natures, in a real-world scenario, sound source separation and localization is needed. Although `SS` greatly improved the ASR results on both *experiment2* and *experiment4*, by on average 27.6 pts in comparison to `FB`, the same trend did not occurred for the beat tracking accuracy. This is justified by the occurrence of instantaneous flaws in the SSL when detecting the musical source, which generates source breaks that lead to time inconsistencies causing gaps in the beat estimations and off-sets in the beat tracking predictions, both penalizing IBT's accuracy.

VII. Conclusions and Future Work

In this paper we introduced a state-recovery mechanism into our beat tracking algorithm to deal with continuous musical stimuli, and applied different multi-channel preprocessing algorithms (*e.g.*, beamforming, ego noise suppression) to enhance the noisy auditory signals lively captured in a real environment. By assessing and comparing the robustness of the whole system through a set of experimental live acoustic conditions, we confirm its applicability into the general framework of robot audition. On the most challenging conditions the proposed solutions *i)* improved the default beat tracking accuracy to a total of 29.6 pts; *ii)* decreased the reaction time to music transition up to 4.3 sec; *iii)* enhanced the noise robustness of the beat tracker against speech and ego-motion noises by 9.8 pts; *iv)* improved the ASR accuracy by 47.5 pts and *v)* efficiently processed simultaneous audio sources of music and speech. In the future, we plan to apply the integrated beat tracking system into an interactive robot dancing system reacting to continuous musical stimuli with synchronized dance motions while responding to human speech commands.

References

[1] H. G. Okuno and K. Nakadai, "Computational Auditory Scene Analysis and its Application to Robot Audition," in *Hands-Free Speech Communication and Microphone Arrays*, 2008, pp. 124–127.

[2] J. L. Oliveira *et al.*, "IBT: A Real-time Tempo and Beat Tracking System," in *ISMIR*, 2010, pp. 291–296.

[3] A. Kapur, "A History of Robotic Musical Instruments," in *International Computer Music Conference (ICMC)*, 2005.

[4] S. Sugano and I. Kato, "WABOT-2: Autonomous Robot with Dexterous Finger-Arm Coordination Control in Keyboard Performance," in *IEEE ICRA*, 1987, pp. 90–97.

[5] E. Singer *et al.*, "LEMUR' s Musical Robots," in *NIME*, 2004, pp. 181–184.

[6] G. Weinberg, *Robotic Musicianship - Musical Interactions Between Humans and Machines*. InTech, 2007.

[7] G. Weinberg *et al.*, "The Creation of a Multi-Human, Multi-Robot Interactive Jam Session," in *NIME*, 2009, pp. 70–73.

[8] T. Mizumoto *et al.*, "Human-Robot Ensemble between Robot Thereminist and Human Percussionist using Coupled Oscillator Model," in *IEEE/RSJ IROS*, 2010, pp. 1957–1963.

[9] 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.

[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] T. Otsuka *et al.*, "Incremental Polyphonic Audio to Score Alignment using Beat Tracking for Singer Robots," in *IEEE/RSJ IROS*, 2009, pp. 2289–2296.

[12] ——, "Music-Ensemble Robot that is Capable of Playing the Theremin while Listening to the Accompanied Music," in *IEA/AIE - Volume Part I*, 2010, pp. 102–112.

[13] 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.

[14] D. K. Grunberg *et al.*, "Robot Audition and Beat Identification in Noisy Environments," in *IEEE/RSJ IROS*, 2011, pp. 2916–2921.

[15] J. L. Oliveira *et al.*, "Online Audio Beat Tracking for a Dancing Robot in the Presence of Ego-Motion Noise in a Real Environment," in *IEEE ICRA*, 2012, to appear.

[16] G. Ince *et al.*, "Online Learning for Template-based Multi-Channel Ego Noise Estimation," accepted at IEEE/RSJ IROS, 2012.

[17] R. Schmidt, "Multiple Emitter Location and Signal Parameter Estimation," *IEEE Trans. on Antennas and Propagation*, vol. 34, no. 3, pp. 276–280, 1986.

[18] H. Nakajima *et al.*, "Blind Source Separation with Parameter-Free Adaptive Step-Size Method for Robot Audition," *IEEE Trans. Audio, Speech, and Language Proc.*, vol. 18, no. 6, pp. 1476–1484, 2010.

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

[20] M. E. P. Davies *et al.*, "Evaluation Methods for Musical Audio Beat Tracking Algorithms," *Technical Report C4DM-TR-09-06*, p. 17, 2009.