ROBOT AUDITION FOR DYNAMIC ENVIRONMENTS

Kazuhiro Nakadai, Gökhan Ince, Keisuke Nakamura, Hirofumi Nakajima

Honda Research Institute Japan Co., Ltd., 8-1 Honcho, Wako, Saitama, 351-0188, JAPAN
nakadai@jp.honda-ri.com

ABSTRACT

This paper addresses robot audition for dynamic environments, where speakers and/or a robot is moving within a dynamically-changing acoustic environment. *Robot Audition* studied so far assumed only stationary human-robot interaction scenes, and thus they have difficulties in coping with such dynamic environments. We recently developed new techniques for a robot to listen to several things simultaneously using its own ears even in dynamic environments; MUltiple SIgnal Classification based on Generalized Eigen-Value Decomposition *(GEVD-MUSIC)*, Geometrically constrained High-order Decorrelation based Source Separation with Adaptive Step-size control *(GHDSS-AS)*, Histogram-based Recursive Level Estimation *(HRLE)*, and Template-based Ego Noise Suppression *(TENS)*. GEVD-MUSIC provides noise-robust sound source localization. GHDSS-AS is a new sound source separation method which quickly adapts its sound source separation parameters to dynamic changes. HRLE is a practical post-filtering method with a small number of parameters. ENS estimates the motor noise of the robot by using templates recorded in advance and eliminates it. These methods are implemented as modules for our open-source robot audition software HARK to be easily integrated. We show that each of these methods and their combinations are effective to cope with dynamic environments through off-line experiments and on-line real-time demonstrations.

Index Terms— Robot audition, Sound source separation, Sound source localization, Ego noise suppression, Dynamic environment, Microphone array

1. INTRODUCTION

Robot audition is a research area which aims at realizing auditory functions by using a robot's own ears [1]. One of the most essential topics in robot audition is to maintain "robustness" in a daily environment. In such an environment, the robot has to deal with dynamically-changing acoustic environment, e.g., moving sound sources, environmental changes caused by robot's motions, dynamical changes of the number of sound sources, and the variety of sound sources. A lot of intelligent systems including robots have been developed for many years. However, these systems mainly focused on making a new function in stationary environments, and dynamic environments have not been considered that much. Thus, these systems have difficulties in maintaining robustness in real-world environments. For instance, an Automatic Speech Recognition (ASR) system uses quite sophisticated methods. Many noise adaptation techniques has been developed. Some ASR technologies which are used in Google voice search and Apple's Siri improved the performance of ASR, and thus a lot of people use these systems on PCs and smartphones[2, 3].

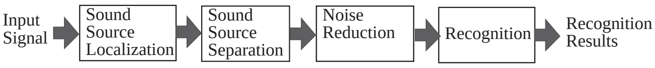


Fig. 1. A Basic Flow of Robot Audition Systems

However, these system still has problems to be applied to robot audition. Since they rely on ASR, only speech can be a target source and other sound sources such as music and environmental sounds are not regarded as target sound sources. Even when they deal with a speech input, it is assumed that the input is clean or slightly-contaminated with noise, while various sound sources can be mixed through robot-embedded microphones. Sound sources and/or robot-embedded microphones can be moving while in recognition, but only stationary cases are considered. These unnatural assumptions degrade the robustness of ASR. Therefore, we are focusing on developing robot audition techniques to deal with dynamic environments such as recognition of moving speakers by making the best use of active motions. This is one of the most essential research topics to realize highly-robust real-world scene analysis.

There are a lot of issues in coping with dynamic environments in a robot audition system which has to deal with distant and multiple target sound sources simultaneously such as speech, music and environmental sounds. The basic processing flow of a typical robot audition system is shown in Fig. 1. There are mainly three processing blocks such as *Sound Source Localization (SSL)*, *Sound Source Separation (SSS)*, and *Noise Reduction (NR)* before a Recognition block which includes ASR. These preprocessing blocks should consider dynamic environments like moving sound sources and robot motions. For SSL, some studies mentioned moving robots in robot audition, that is, low-level active audition [4, 5]. However, their robots almost ignored robot-generated noises which degrade SSL. For SSS, a lot of studies can be found [6], but most of them assumed stationary scenes. We also reported *Geometric Source Separation (GSS)* utilizing measured transfer functions [7]. Although our robot audition system with GSS attained three simultaneous speech recognition like a rock-paper-scissors sound game, its scene was still stationary. For SE, we used a multi-channel post-filter based on Minimum Mean Square Error Estimation (MMSE) [8] which enhances the output of SSS by eliminating stationary and non-stationary interference noises. However, it has a lot of parameters that needed to be tuned, and it is impractical for these parameters to be tuned whenever an environment changes.

In this paper, we present four methods: noise-robust SSL, quickly adaptable SSS, NR through stationary and non-stationary noise noise estimators to improve the performance of robot audition in dynamic environments. These methods were separately reported in our recent publications, and this paper integrates all of them into a robot audition system which can deal with speech and music sources at the same time in a robot-moving dynamic environment.

H.Nakajima is currently working with Kogakuin Univ.

These methods are implemented as modules for *HRI-JP Audition for Robots with Kyoto University (HARK)*[7] which is open-sourced real-time robot audition software, and we show the effectiveness of these methods through off-line experiments and online demonstrations.

2. SIGNAL PROCESSING METHODS FOR DYNAMIC ENVIRONMENTS

Three approaches mentioned in the previous section are described. Noise-robust sound source localization uses *MUltiple SIgnal Classification (MUSIC)* based on *Generalized Eigen-Value Decomposition (GEVD-MUSIC)*[9]. This method is able to take robot's stationary noise into account and localize only specific sound sources even while a robot's head is rotating. *Geometrically constrained High-order Decorrelation based Source Separation (GHDSS)* is newly developed as a better SSS method, and also its performance for dynamic environments is improved by introducing *Adaptive Step-size (AS)* control [10]. AS makes the adaptation speed of a separation matrix faster. A new NR method called *Histogram-based Recursive Level Estimation (HRLE)* enhances speech with a small number of parameters. Using instantaneous joint status data of the robot, *Template-based Ego Noise Suppression (TENS)* estimates the ego noise data from a large dataset of audio templates recorded in advance, and applies spectral subtraction on the noisy audio spectrum to suppress the predicted noise.

2.1. Formulation of Observation Model

Suppose that there are N sources and M ($\geq N$) microphones. A spectrum vector of N sources at frequency ω, $\mathbf{s}(\omega)$, is denoted as $[s_1(\omega)\ s_2(\omega)\ \cdots\ s_N(\omega)]^T$, and a spectrum vector of signals captured by the M microphones at frequency ω, $\mathbf{x}(\omega)$, is denoted as $[x_1(\omega)\ x_2(\omega)\ \cdots\ x_M(\omega)]^T$, where T represents a transpose operator. $\mathbf{x}(\omega)$ is, then, calculated as

$$\mathbf{x}(\omega) = \mathbf{D}(\omega)\mathbf{s}(\omega) + \mathbf{n}(\omega), \tag{1}$$

where $\mathbf{D}(\omega)$ is a transfer function (TF) matrix between a microphone array and a sound source, $\mathbf{n}(\omega)$ is a noise vector which shows diffuse and dynamically changing colored noise. It is assumed to be statistically independent of $\mathbf{s}(\omega)$.

We omit ω for simplification.

2.2. Sound Source Localization for Dynamic Environments

We first introduced MUSIC based on *Standard Eigen-Value Decomposition (SEVD)* (*SEVD-MUSIC*). The MUSIC spatial spectrum is defined by

$$\hat{P}(\psi) = \frac{|\boldsymbol{G}^*(\psi)\boldsymbol{G}(\psi)|}{\sum_{m=N+1}^{M} |\boldsymbol{G}^*(\psi)\boldsymbol{e}_m|}, \tag{2}$$

where $\boldsymbol{G}^*(\psi)$ is a steering vector at sound source direction ψ, and $\boldsymbol{e}_m$ is an eigen vector of a input correlation matrix, $\boldsymbol{R} = \mathbf{x}\mathbf{x}^T$.

By searching N peaks in $\hat{P}(\psi)$, N sound sources are localized with a MUSIC algorithm. Note that $\mathbf{D}$ is represented as $[\boldsymbol{G}^*(\psi_1), \boldsymbol{G}^*(\psi_2), \cdots, \boldsymbol{G}^*(\psi_N)]$.

SEVD-MUSIC works when $\mathbf{n}$ has sufficiently low power compared to target sound sources, that is, it assumes that N large eigen values correspond to target signals. This means that it mis-localizes high-powered noise sources as target signals. Therefore we proposed *GEVD-MUSIC* [9].

GEVD-MUSIC redefines an eigen value and vector, $\hat{\lambda}_m$ and $\hat{\boldsymbol{e}}_m$ by using a correlation matrix of pre-measured noise, $\boldsymbol{K}$.

$$\boldsymbol{K}^{-1}\boldsymbol{R}\hat{\boldsymbol{e}}_m = \hat{\lambda}_m\hat{\boldsymbol{e}}_m\ . \tag{3}$$

Since $\boldsymbol{K}^{-1}$ has an effect to whiten the pre-measured noise, no mis-localization occurs with the noise. $\boldsymbol{K}$ can be incrementally estimated, and thus it is able to cope with dynamically-changing noise. GEVD-MUSIC is effective for a robot because it basically generates high-powered noise from fans and dynamically-changing ego-motion noise. Although GEVD-MUSIC has such an advantage when $\boldsymbol{n}$ is white noise, $\boldsymbol{K}$ in Eq. (3) is cross-correlated when $\mathbf{n}$ is colored noise, and the Hermitian symmetry in $\boldsymbol{K}^{-1}\boldsymbol{R}$ is not satisfied. Eventually, the signal subspace spanned by $\hat{\boldsymbol{e}}_m$ $(1 \leq m \leq N)$, and the noise subspace spanned by $\hat{\boldsymbol{e}}_m$ $(N+1 \leq m \leq M)$ are not always orthogonal to each other, i.e., the inner product of $\hat{\boldsymbol{e}}_m$ in the noise subspace and a steering vector at the direction of a target sound does not equal to be 0. This makes the peaks in $\hat{P}(\psi)$ smaller and drops the performance of SSL.

To solve this problem, this paper extends Eq. (3) to utilize the following GEVD by a square root of $\boldsymbol{K}$:

$$\boldsymbol{K}^{-\frac{1}{2}}\boldsymbol{R}\boldsymbol{K}^{-\frac{1}{2}}\hat{\boldsymbol{e}}_m = \hat{\lambda}_m\hat{\boldsymbol{e}}_m\ . \tag{4}$$

In this formulation, even when $\boldsymbol{K}$ is colored noise, $\boldsymbol{K}^{-\frac{1}{2}}\boldsymbol{R}\boldsymbol{K}^{-\frac{1}{2}}$ holds its Hermitian symmetry [11]. As a result, the GEVD can whiten the colored noise properly.

2.3. Sound Source Separation for Dynamic Environments

We developed a hybrid algorithm of beamforming and blind separation, GHDSS. It is based on GSS proposed by Parra *et al.* [12]. We extended an online version of GSS developed by Valin *et al.*[6]. We first replace the cost function based on cross power correlation with the one based on higher order correlation to improve its separation performance. We then introduced AS to provide faster adaptation using stochastic gradient and shorter time frame estimation.

Using the separated signal $\mathbf{y}$, SSS is usually defined by $\mathbf{y} = \mathbf{W}\mathbf{x}$ in frequency domain. $\mathbf{W}$ is called a *separation matrix*. We intentionally ignored $\boldsymbol{n}$ in Eq. (1), because such dynamical and additive noise can be suppressed by post-processing of SSS called NR.

In order to estimate $\mathbf{W}$, GHDSS introduces two cost functions, that is, separation sharpness (J_{SS}) and geometric constraints (J_{GC}):

$$\begin{aligned} J_{SS}(\mathbf{W}) &= \|\phi(\mathbf{y})\mathbf{y}^H - \mathrm{diag}[\phi(\mathbf{y})\mathbf{y}^H]\|^2 && (5)\\ J_{GC}(\mathbf{W}) &= \|\mathrm{diag}[\mathbf{W}\mathbf{D} - \mathbf{I}]\|^{\mathbf{2}} && (6)\end{aligned}$$

where $\|\cdot\|^2$ indicates the Frobenius norm, $\mathrm{diag}[\cdot]$ is the diagonal operator, and H represents the conjugate transpose operator. For a nonlinear function, $\phi(\mathbf{y})$, we selected a hyperbolic-tangent-based function [13] in this paper. Since the best $\mathbf{W}$ is always changing in the real world, $\mathbf{W}$ is adaptively updated by using

$$\mathbf{W}_{t+1} = \mathbf{W}_t - \mu_{SS}\mathbf{J}'_{SS}(\mathbf{W}_t) + \mu_{GC}\mathbf{J}'_{GC}(\mathbf{W}_t). \tag{7}$$

where $\mathbf{W}_t$ denotes $\mathbf{W}$ at the current time frame t, $\mathbf{J}'_{SS}(\mathbf{W})$ and $\mathbf{J}'_{GC}(\mathbf{W})$ are complex gradients [14] of $J_{SS}(\mathbf{W})$ and $J_{GC}(\mathbf{W})$, which decide an update direction of $\mathbf{W}$. μ_{SS} and μ_{GC} are called step-size parameters.

These two step-size parameters usually have fixed values decided heuristically or empirically, although they should be frequency-dependent and time-variant values according to environmental changes. To decide the optimal design of these step-size parameters for faster adaptation of $\mathbf{W}$, we propose AS for multi-channel signal processing[15]. AS is well-studied in the field of echo cancellation [16]. We extended AS to support multi-channel input and complex number signals by using the multi-dimensional version of Newton's method and linear approximation formula for a complex gradient matrix[15]. In GHDSS with AS (GHDSS-AS), μ_{SS} and μ_{GC} are defined by,

$$\mu_{SS} = \frac{\|\phi(\mathbf{y})\mathbf{y}^H - \mathrm{diag}[\phi(\mathbf{y})\mathbf{y}^H]\|^2}{8\|\phi(\mathbf{y})\mathbf{y}^H - \mathrm{diag}[\phi(\mathbf{y})\mathbf{y}^H]\tilde{\phi}(\mathbf{y})\mathbf{x}^H\|^2} \quad (8)$$

$$\mu_{GC} = \frac{\|\mathrm{diag}[\mathbf{WD} - \mathbf{I}]\|^{\mathbf{2}}}{8\|\mathrm{diag}[\mathbf{WD} - \mathbf{I}]\mathbf{D^H}\|^{\mathbf{2}}}.$$

These two parameters, μ_{SS} and μ_{GC}, become large values when a separation error is high, for example, due to source position changes. It will be low when the error is small due to the convergence of the separation matrix. Thus, step-size and weight parameters are controlled optimally at the same time.

2.4. Noise Reduction for Dynamic Environments

Since we ignored $\boldsymbol{n}$ in Eq. (1), we need to have a process to suppress it. The first cost function in SSS shown in Eq. (5) is less affected by $\boldsymbol{n}$, because $\boldsymbol{n}$ and $\boldsymbol{s}$ are assumed to be statistically independent each other. Only the second cost function shown in Eq. (6) can be affected by $\boldsymbol{n}$. Since this cost function corresponds to beamforming, we can say that a part of diffuse noise power, which corresponds to target directions, can remain after SSS. This means that $\boldsymbol{n}$ is somewhat suppressed after SSS, and we assume that the power of $\boldsymbol{n}$ is smaller than that of any target source included in $\boldsymbol{s}$. For NR, we proposed two methods, HRLE and TENS.

2.4.1. Histogram-based Recursive Level Estimation (HRLE)

HRLE requires 5 parameters, of which only 2 parameters are to be optimized. HRLE estimates input noise levels by taking L_χ from an input power level histogram, which is commonly used in environmental noise measurement using sound-level meters (see Fig. 2). Since HRLE uses recursive averages, HRLE calculates a time-varying cumulative histogram $h_c(t,i)$ in real-time from a histogram $h_n(t,i)$ shown in Eq.(10), where t shows a time frame index, and i shows a bin index in a histogram. Therefore, noise level estimation smoothly and quickly adapts to the environmental changes. HRLE is represented by

$$L_\chi(t) = L_{min} + L_{st} \cdot \underset{I}{\mathrm{argmin}}\,(\chi h_c(t, I_{max}) - h_c(t,I))\ , (9)$$

$$h_c(t,i) = \sum_{k=0}^{i} \alpha h_n(t-1,k) + (1-\alpha)\delta(k - I_u(t)), \quad (10)$$

$$I_u(t) = \lfloor (20\log_{10}|u(t)| - L_{min})/L_{st} \rfloor, \quad (11)$$

where $u(t)$ represents an input signal in time-frequency domain. L_{min}, L_{st} and I_{max} are the minimum level, the level width of a bin and the maximum index of the histogram, respectively, χ indicates the position (0–1) of the cumulative frequency, α is the time decay parameter calculated from T_r and sampling frequency F_s as $\alpha = 1 - 1/(T_r F_s)$, $L_\chi(t)$ is the estimated level, $\delta(t)$ is the Dirac delta function and $\lfloor \cdot \rfloor$ is the flooring function. This method uses 5 parameters. Three of them, L_{min}, L_{step} and I_{max}, determine the range and sharpness of the histogram, they are insensitive to the estimated results. The other 2 parameters, χ and α, should be empirically optimized.

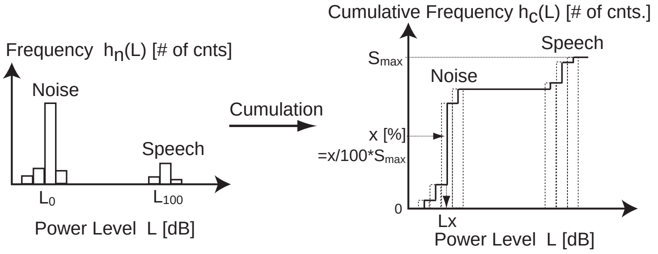


Fig. 2. Histogram (left) and Cumulative histogram (right) of input power level: χ in L_χ indicates the position of the cumulative histogram, e.g., L_0, L_{100}, and L_{50} show the minimum, the maximum, and the median.

2.4.2. Template-based Ego Noise Suppression (TENS)

Template Estimation (TE) [17] 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 on the current motion assuming 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.

Technically, this method utilizes motor encoders of the robot, which measure the angular position of each joint. During the motion of the robot, actual position of each motor, θ, is acquired regularly at each time frame. Using the difference between consecutive sensor outputs, velocities, $\dot{\theta}$, and accelerations, $\ddot{\theta}$ are calculated. Considering that ξ joints are active, 3ξ attributes are generated. Each feature is normalized to [01] so that all features have the same contribution on the prediction. The resulting feature vector has the form of $\boldsymbol{\theta} = [\theta_1, \dot{\theta}_1, \ddot{\theta}_1, \ldots, \theta_\xi, \dot{\theta}_\xi, \ddot{\theta}_\xi]$. In the template generation (database creation) phase, one feature vector is assigned to the current audio spectrum $\boldsymbol{x}$ and used to label the instantaneous noise fragment; this data block $\boldsymbol{T} = [\boldsymbol{\theta}, \boldsymbol{x}]$ is called a *parameterized template*.

Incremental Learning (IL) of the templates makes use of previously learned knowledge about the templates to speed up learning. It makes the noise estimation module more robust because errors in the training set can be corrected *during operation* and it enables the system to adapt to partially-known or dynamic environments. The learning system checks whether the acquired audio signal is mixed with a directional audio signal so that the template is discarded. During the learning interval, the system also decides if each observed template is a known template or a new template to be learned. For this, the smallest distance of observed template and most similar template in the database is compared to a given fixed distance threshold, d_{th}. When the similarity is low, the template is treated as a missing template and inserted into a database; otherwise the adaptive update mechanism is active, which computes the weighted sum of the old and current template by introducing a forgetting factor η with $0 \leq \eta \leq 1$. This factor helps to provide a moderate balance between adaptivity (learning quality) and stability (robustness against errors).

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.

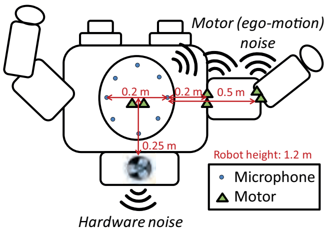


Fig. 3. Sketch of the humanoid robot of Honda.

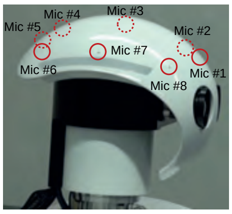


Fig. 4. Close-up of HEARBO head.

3. IMPLEMENTATION

The methods to deal with dynamic environment are implemented as modules of HARK (Honda Research Institute Japan Audition for Robots with Kyoto University) which is our open source software for robot audition[7]. HARK consists of a complete set of modules for robot audition as component blocks on FlowDesigner[18][1], which works on Linux in real time. Many multi-channel sound cards are supported to build a real-time robot audition system easily. For preprocessing, sound source localization, tracking and separation are available. These preprocessing modules are able to be integrated with automatic speech recognition (ASR) based on missing feature theory (MFT) [19, 20]. For MFT, modules such as acoustic feature extraction for ASR, automatic missing feature mask generation, and ASR interface are prepared. Missing-feature-theory based ASR (MFT-ASR) is provided as a patch for Julius [21] which is a Japanese open source speech recognition system. Only MFT-ASR is implemented as a non-FlowDesigner module in HARK, but it connects with FlowDesigner by using modules from the ASR interface. Users are able to flexibly build robot audition systems by using the GUI interface. The robot audition system was tested on two different robots: one robot developed by Honda as shown in Fig. 3 and one robot developed by HRI-JP called HEARBO. 8 ch microphone arrays were embedded in the head of both robots as in Fig. 4.

4. EVALUATION

We evaluate the proposed methods individually and also show an online demonstration using these methods to show their effectiveness in dynamic environments.

4.1. GEVD-MUSIC

Numerical comparisons between SEVD-, GEVD-MUSIC with Eq. (3), and GEVD-MUSIC with Eq. (4) are shown. In the experiment, there were a target sound (white noise) and a noise source (robot's fan noise) 1m away from the microphone array in the directions of 60° and 180°, respectively. $\boldsymbol{G}(\psi)$ of $\psi = \{-175°, -170°, ..., 180°\}$ were used. Hence, the resolution of SSL was 5°. We evaluated the relationship between *Signal-to-Noise Ratio* (*SNR*) and localization accuracy. The signal ratio between the target sound(s) and the robot noise changes from -25 dB to 15 dB. The localization accuracy was defined as the number of frames whose highest peaks in Eq. (2) were in the direction of the target sound source (60°) in 100 frames.

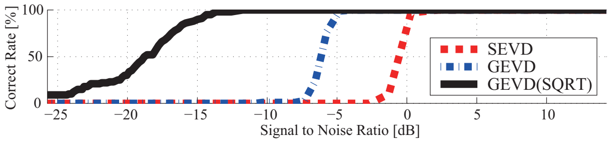


Fig. 5. Localization correct rates for SEVD-, and GEVD-MUSICs

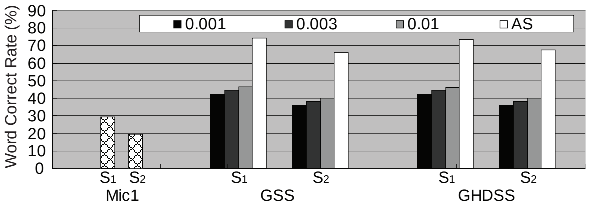


Fig. 6. WCRs of separated speech: S_1 is stationary, S_2 is moving.

Fig. 5 shows the result. The horizontal axis shows SNR, and the vertical axis shows the localization accuracy. The dotted-, chained-, and solid-line show the result of SEVD-MUSIC, GEVD-MUSIC with Eq. (3) and GEVD-MUSIC with Eq. (4), respectively. The solid-line shows better performance than others. This means that GEVD in Eq. (4) whitened the noise signals and obtained mutually-orthogonal signal and noise subspaces. The computational cost of each method was also evaluated. We conducted SSL over 1000 frames for each method and measured the averaged processing time. As a result, the average processing time for SEVD-MUSIC, GEVD-MUSIC with Eq. (3), and GEVD-MUSIC with Eq. (4) was 11.9 ms, 13.6 ms, and 16.9 ms, respectively. GEVD-MUSIC requires slightly higher computational cost than SEVD-MUSIC. Although the frame period was 10 ms, it is practically sufficient that the system executes GEVD operation once in 25 frames (250 ms) in SSL, and thus it works in real time. We have already developed another MUSIC algorithm which has less computational cost with maintaining noise-robustness based on generalized singular-value decomposition. Since it reduces the computational cost to be half, it works in real time in a frame-by-frame manner (10 ms), and it will be published in another paper.

4.2. GHDSS-AS

We performed separation experiments with GHDSS-AS in a dynamic environment. In this experiment, two speakers were used. One speaker (S_1) was fixed in front of the robot $\theta_1 = 0$. S_1 played ATR 216 phonetically-balanced Japanese words[2] with a 1-2 sec pause between words. The location of another speaker (S_2) was changed after every utterance. The direction of S_2 was randomly selected from $\theta_2 = -90°, -60°, -30°, 30°, 60°, 90°$. The output gain was descretely selected as $\{-6, -3, 0, 3, 6\}$ dB. S_2 uttered different words than S_1, with the utterance period of S_2 almost overlapping S_1. For S_1 and S_2, we used 12 combinations of mixed speech wordsets synthesized from the six ATR wordsets of three female (f1-f3) and three male (m1-m3). For comparison, captured signal with one microphone was used in addition to separation with GSS and GHDSS. In GSS and GHDSS, three fixed stepsize values and the proposed AS method were compared. Word Correct Rates (WCRs) were used as a metric to show the performance.

Fig. 6 shows the WCRs. AS shows the best performance. For the stationary source, GSS-AS and GHDSS-AS have higher perfor-

[1] http://flowdesigner.sourceforge.net/

[2] http://research.nii.ac.jp/src/eng/list/detail.html

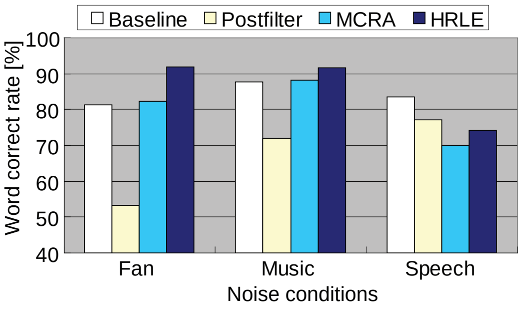


Fig. 7. Word correct rate for each condition

Table 1. Ego noise suppression performance for all methods

SNR	*No Processing*	*HRLE*	*TE wo. IL*	*TE w. IL*
-3dB	28.3	47.2	58.7	**69.9**
3dB	78.0	84.1	87.4	**89.3**

mances. By introducing AS to GSS and GHDSS, their performances increased by around 25%. The WCRs of the moving source were worse by roughly 10% compared to those the stationary source; however, the plotted trends were almost the same. From these results, we can conclude that GHDSS has higher performance and AS is effective for real-world applications using ASR.

4.3. HRLE

We evaluated WCRs of isolated word recognition. An acoustic model was trained with JNAS and the test dataset had a 236 wordset. The main speaker was located $1m$ in front of the robot. Three different noise types were used, that is, *Fan* (diffuse robot's fan noise, SNR = 0 dB), *Music* (music from 30° and BGN, SNR 2 dB), *Speech* (speech from 30° and BGN, SNR = 2 dB). Four methods were examined followed by GSS-AS.

- **Baseline**: any non-linear NR is performed.
- **Postfilter**: an MMSE-based non-linear NR sub-process is performed (previously-used)[20].
- **MCRA**: a typical non-linear NR based on SS[22] and MCRA[23] is used.
- **HRLE**: the proposed NR, HRLE is used.

Fig. 7 shows the WCR for each condition. We found out that our previous method **Postfilter** was the worst in almost all conditions because of the lack of robustness against environmental changes. For *Fan* and *Music* noises, **HRLE** was highest in all methods. For *Speech* noise, all methods were lower than the Baseline. We suppose that because statistical characteristics of noise and speech are the same, all methods failed to estimate the noise level precisely.

4.4. TENS

To assess the learning capability of our system with respect to threshold d_{th}, we evaluated the Normalized Noise Estimation Error (NNEE) in incremental steps, i.e. after repeating the same motion Z times ($1 \leq Z \leq 20$). NNEE computes the error of the noise estimate normalized by the energy of the actual noise averaged by the number of frames and frequency bins. The optimal value of $\eta = 0.9$ for incremental learning, which pursues stability rather than adaptivity is found empirically. Because we used bounded ([0 1]) features, the values of d_{th} are also bounded to [0 $\sqrt{3\xi}$].

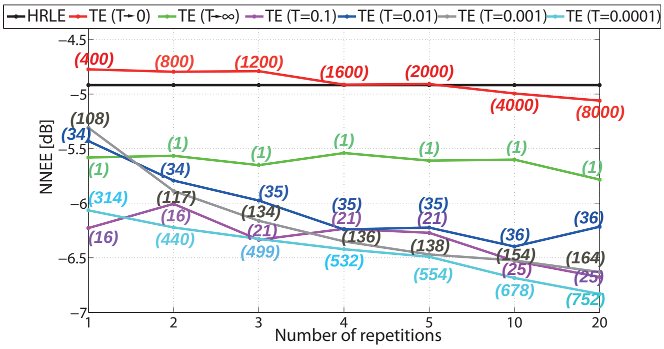


Fig. 8. Estimation error in relation with number of iterations

Fig. 8 demonstrates the tendency of reduced errors with respect to increased repetitions. The numbers in brackets indicate the size of the database. The settings denoted as "$T \rightarrow 0$" indicates that there is a continuous insertion of every incoming template into the database like in conventional template estimation method [17] and "$T \rightarrow \infty$" indicates that there is only one single (mean) template updated during all repetitions. They both yield the baseline performance. Because HRLE is a stationary noise estimator, it cannot deal with non-stationary ego noise and shows also a poor performance. The error decreases when d_{th} is sufficiently low. We also observe that there is a negative correlation between the number of templates stored and the value of d_{th}.

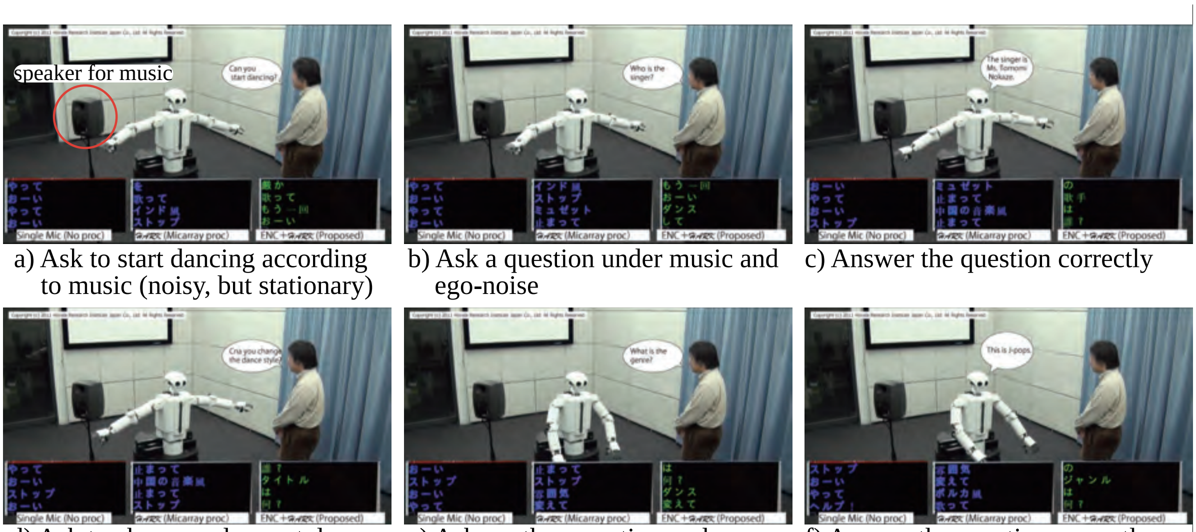
Finally, we evaluated the noise reduction performance in an ASR task using an acoustic model trained with JNAS and a test dataset with a 236 wordset. As Tab. 1 demonstrates, TENS with incremental learning achieves the largest WCRs among all methods for both the SNRs. However, under changing stationary noise conditions, HRLE contributes better for eliminating background noise. Hence we propose to combine TE with HRLE so that TE deals only with the non-stationary part of the noise regarding the ego-motion noise. This kind of configuration increases the robustness of the noise suppression system and makes it independent of any change in the environmental noise condition.

4.5. Online Demonstration

The total system was evaluated through a speech dialog scenario for a dancing robot, where a user was asking questions about the songs (*e.g.*, name, year, composer, etc.) while the robot was dancing to the beats of music pieces selected/changed by the user. Fig. 9 shows the snapshots of this scenario, in which all the above-mentioned modules were integrated into one single system.

5. CONCLUSION

We proposed four methods to improve robot audition in dynamic environments, that is, MUltiple SIgnal Classification based on Generalized Eigen-Value Decomposition *(GEVD-MUSIC)*, Geometrically constrained High-order Decorrelation based Source Separation with Adaptive Step-size control *(GHDSS-AS)*, Histogram-based Recursive Level Estimation *(HRLE)*, Template-based Ego Noise Suppression *(TENS)*. We showed that each method improves the performance of automatic speech recognition. In addition, we developed a real-time robot audition system based on HARK with the proposed techniques, and showed the effectiveness through two dialog sce-



a) Ask to start dancing according to music (noisy, but stationary)

b) Ask a question under music and ego-noise

c) Answer the question correctly

d) Ask to change a dance style (ego-noise changes drastically)

e) Ask another question under music and ego-noise

f) Answer the question correctly

Fig. 9. Speech interaction under music (directional) and ego-motion noise.

narios. Our future work includes more detailed evaluation to obtain more concrete results in various dynamically-changing environments and computational scene analysis mentioned in the previous section.

6. REFERENCES

[1] K. Nakadai *et al.*, "Active audition for humanoid," in *Proc. of 17th National Conference on Artificial Intelligence (AAAI-2000)*. 2000, pp. 832–839, AAAI.

[2] H. Lin *et al.*, "Recognition of multilingual speech in mobile applications," in *Proceedings of 2012 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2012)*, 2012.

[3] M. Mohri *et al.*, *Speech Recognition with Weighted Finite-State Transducers*, chapter Handbook on Speech Processing and Speech Communication, Part E: Speech recognition, pp. 559–582, Springer-Verlag, 2008.

[4] K. Nakadai *et al.*, "Improvement of recognition of simultaneous speech signals using av integration and scattering theory for humanoid robots," *Speech Communication*, vol. 44, no. 1-4, pp. 97–112, 2004.

[5] Y. Sasaki *et al.*, "Daily sound recognition using pitch-cluster-maps for mobile robot audition," in *Proc. IROS 2009*, 2009, pp. 2724–2729.

[6] J.-M. Valin *et al.*, "Robust recognition of simultaneous speech by a mobile robot," *IEEE Transactions on Robotics*, vol. 23, no. 4, pp. 742–752, 2007.

[7] K. Nakadai *et al.*, "Design and implementation of robot audition system HARK," *Advanced Robotics*, vol. 24, no. 5-6, pp. 739–761, 2009.

[8] Y. Ephraim and D. Malah, "Speech enhancement using minimum mean-square error short-time spectral amplitude estimator," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. ASSP-32, no. 6, pp. 1109–1121, 1984.

[9] K. Nakamura *et al.*, "Intelligent sound source localization for dynamic environments," in *IROS*, 2009, pp. 664–669.

[10] K. Nakadai *et al.*, "Sound source separation of moving speakers for robot audition," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2009)*, 2009, pp. 3685–3688.

[11] G. Strang, *Linear Algebra and its Applications Third Edition*, Harcourt Brace Jovanovich, 1988.

[12] L. C. Parra and C. V. Alvino, "Geometric source separation: Mergin convolutive source separation with geometric beamforming," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 6, pp. 352–362, 2002.

[13] H. Sawada *et al.*, "Polar coordinate based nonlinear function for frequency-domain blind source separation," in *2002 IEEE Int 'l. Conf. on Acoustics, Speech and Signal Processing (ICASSP 2002)*, 2002, pp. 1001–1004.

[14] D. H. Brandwood, "A complex gradient operator and its application in adaptive array theory," *IEE Proc.*, vol. 130, no. 1, pp. 251–276, 1983.

[15] H. Nakajima *et al.*, "Adaptive step-size parameter control for real-world blind source separation," in *Proceedings of the 2008 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2008)*. Apr. 2008, pp. 149–152, IEEE.

[16] S. Yamamoto and S. Kitayama, "An adaptive echo canceller with variable step gain method," *Trans. of the IECE of Japan*, vol. E65, no. 1, pp. 1–8, 1982.

[17] G. Ince *et al.*, "Whole body motion noise cancellation of a robot for improved automatic speech recognition," *Advanced Robotics*, vol. 25, pp. 1405–1426, 2011.

[18] C. Côté *et al.*, ," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2004)*. 2004, pp. 1820–1825, IEEE.

[19] J. Barker *et al.*, "Robust ASR based on clean speech models: An evaluation of missing data techniques for connected digit recognition in noise," in *Proc. of Eurospeech-2001*. 2001, pp. 213–216, ESCA.

[20] S. Yamamoto *et al.*, "Design and implementation of a robot audition system for automatic speech recognition of simultaneous speech," in *Proceedings of the 2007 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU-2007)*. Dec. 2007, pp. 111–116, IEEE.

[21] T. Kawahara and A. Lee, "Free software toolkit for Japanese large vocabulary continuous speech recognition," in *International Conference on Spoken Language Processing (ICSLP)*, 2000, vol. 4, pp. 476–479.

[22] S. F. Boll, "A spectral subtraction algorithm for suppression of acoustic noise in speech," in *Proceedings of 1979 International Conference on Acoustics, Speech, and Signal Processing (ICASSP-79)*. 1979, pp. 200–203, IEEE.

[23] I. Cohen and B. Berdugo, "Speech enhancement for non-stationary noise environments," *Signal Processing*, vol. 81, no. 2, pp. 2403–2418, 2001.