A Real-Time Super-Resolution Robot Audition System that Improves the Robustness of Simultaneous Speech Recognition

Keisuke Nakamura¹,², Kazuhiro Nakadai¹, Hiroshi G. Okuno²

1. Honda Research Institute Japan Co., Ltd., 8-1 Honcho, Wako, Saitama 351-0114, JAPAN., {keisuke, nakadai}@jp.honda-ri.com
2. Graduate School of Informatics, Kyoto University, 36-1 Yoshida-Honmachi, Sakyo-ku, Kyoto 606-8501, JAPAN., okuno@kuis.kyoto-u.ac.jp

Abstract

This study addresses a framework for a robot audition system, including sound source localization (SSL) and sound source separation (SSS), that can robustly recognize simultaneous speeches in a real environment. Because SSL estimates not only the location of speakers but also the number of speakers, such a robust framework is essential for simultaneous speech recognition. Moreover, improvement in the performance of SSS is crucial for simultaneous speech recognition because the robot has to recognize the individual source of speeches. For simultaneous speech recognition, current robot audition systems mainly require noise-robustness, high resolution, and real-time implementation. Multiple signal classification (MUSIC) based on standard Eigenvalue decomposition (SEVD) and Geometric-constrained high-order decorrelation-based source separation (GHDSS) are techniques utilizing microphone array processing, which are used for SSL and SSS, respectively. To enhance SSL robustness against noise while detecting simultaneous speeches, we improved SEVD-MUSIC by incorporating generalized Eigenvalue decomposition (GEVD). However, GEVD-based MUSIC (GEVD-MUSIC) and GHDSS mainly have two issues: (1) the resolution of pre-measured Transfer Functions (TFs) determines the resolution of SSL and SSS and (2) their computational cost is expensive for real-time processing. For the first issue, we propose a TF-interpolation method integrating time-domain-based and frequency-domain-based interpolation. The interpolation achieves super-resolution robot audition, which has a higher resolution than that of the pre-measured TFs. For the second issue, we propose two methods for SSL: MUSIC based on generalized singular value decomposition (GSVD-MUSIC) and hierarchical SSL (H-SSL). GSVD-MUSIC drastically reduces the computational cost while maintaining noise-robustness for localization. In addition, H-SSL reduces the computational cost by introducing a hierarchical search algorithm instead of using a greedy search for localization. These techniques are integrated into a robot audition system using a robot-embedded microphone array. The preliminary experiments for each technique showed the following: (1) The proposed interpolation achieved approximately 1-degree resolution although the TFs are only at 30-degree intervals in both SSL and SSS; (2) GSVD-MUSIC attained 46.4% and
40.6% of the computational cost compared to that of SEVD-MUSIC and GEVD-MUSIC, respectively; (3) H-SSL reduced 71.7% of the computational cost to localize a single speaker. Finally, the robot audition system, including super-resolution SSL and SSS, is applied to robustly recognize four sources of speech occurring simultaneously in a real environment. The proposed system showed considerable performance improvements of up to 7% for the average word correct rate during simultaneous speech recognition, especially when the TFs were at more than 30-degree intervals.

Keywords: robot audition, sound-source localization and separation, automatic speech recognition

1 Introduction

Automatic Speech Recognition (ASR) by a robot is one of the most common human-robot interactions and is essential for a robot working in a human environment. Different from ASR, conducted via a close-talking microphone as observed in mobile phone applications, ASR by a robot has to use the robot’s own microphones, and the location, timing, and number of speech sources are unknown. Moreover, because a robot has its own ego-noise (self-generated noise), the signal-to-noise ratio (SNR) of the speech is low in most of the cases. To achieve ASR in a highly dynamic and noisy environment, robot audition [1] has been studied for a decade. Fig. 1 shows the overview of the robot audition system. In addition to the signal processing for ASR by a close-talking microphone, the robot audition system employs sound source localization (SSL) and sound source separation (SSS) to recognize an unknown number of individual sources of speech conducted simultaneously. Because SSL estimates the location, timing, and number of speech sources, and SSS separates the speech from each source, an improvement in the performances of SSL and SSS is essential.

Techniques using microphone array processing such as multiple signal classification (MUSIC) based on standard Eigenvalue decomposition (SEVD-MUSIC) [2] and geometric-constrained high-order decorrelation-based source separation (GHDSS) [3] are used for SSL and SSS in our robot audition system, respectively. However, while considering the practicality of using current robot audition systems in a highly dynamic and noisy environment, we recognized that they have three problems: they are not adequately noise-robust, their resolution is not sufficiently high, and they exhibit difficulties during real-time implementation.

To solve the noise-robustness problem for SSL, we previously extended SEVD-MUSIC by incorporating generalized Eigenvalue decomposition (GEVD), called GEVD-MUSIC [4]. The method successfully realized the localization of speech sources when the noise levels were high and the SNR was less than zero.

Although GEVD-MUSIC has such an advantage in noise-robustness, the robot audition system mainly has two issues:

1) The resolution of pre-measured transfer functions (TFs) determine the resolution for both SSL and SSS.
2) MUSIC algorithms such as GEVD-MUSIC are computationally expensive for subspace decomposition and high resolution SSL.

The first issue, concerning resolution, is critical for simultaneous speech recognition because roughly pre-measured TFs might not be able to localize all the speakers and would have difficulty in separating each source of speech accurately. The issue is that the resolution of pre-measured TFs determines the resolution of SSL and SSS; measuring fine resolutions is time consuming. To obtain fine resolution without fine measurements, one brute-force solution would be a numerical TF calculation using the geometric information of a microphone array, known as the numerical method. However, numerical methods are not completely accurate because a robot-embedded microphone array is attached to a complex robot surface. Several numerical methods adaptable to a robot-embedded microphone array have high computational costs [5, 6], which are not suitable for a real-time robot audition system. A better solution would be interpolating the pre-measured TFs to obtain TFs with any desired resolution, called as an interpolation method. Because interpolation methods include a complex robot surface in pre-measured TFs, they are suitable for robots. There are several interpolation methods, which can be divided into two categories: all-points methods and adjacent-points methods. The all-points methods [8, 9, 10, 11, 12, 13, 14] utilize all the known TFs to estimate a TF, and they cause a low number of estimation errors. However, their algorithms still have difficulties in real-time processing. On the other hand, the adjacent-points methods [15, 16, 17] utilize only the two closest pre-measured TFs from the estimated point. Therefore, they have advantages in terms of real-time processing and are suitable for robot audition. The estimation error is relatively large because the calculation relies on two adjacent TFs.

The second issue, concerning the computation cost, is also a crucial for robots to interact with speakers in real time. This issue comprises two main points. First, the subspace decomposition in MUSIC algorithms, namely SEVD in SEVD-MUSIC and GEVD in GEVD-MUSIC, increases the computational cost dramatically and still has difficulties especially during real-time frame-by-frame operations. Second, to obtain a higher resolution, SSL needs more time to calculate a spatial spectrum and search the peaks of the spectrum. For effective robot audition, we need to achieve both high resolution and real-time processing simultaneously.

This study intends to propose a robot audition system that solves these issues and implements simultaneous speech recognition robustly in a real environment.

For the issue concerning the resolution, we propose TF interpolation based on the integration of frequency- and time-domain linear interpolation (FTDLI), which is based on an adjacent-points method in considering the real-time interpolation. This is the extension of a correlation matrix interpolation
for TFs [18]. Two interpolation methods for estimating the amplitude and phase are integrated, which improves the interpolation accuracy for both the amplitude and phase of TFs. Using FTDLI, we can generate TFs with the desired resolution and achieve super-resolution SSL and SSS, in which super resolution represents the resolution exceeding that of the pre-measured TFs.

For the issue concerning the computation cost, we initially extended GEVD-MUSIC [4] to utilize generalized singular-value decomposition (GSVD), called as GSVD-MUSIC. The extension not only maintains noise-robustness in GEVD-MUSIC but also reduces the computational cost enormously. Next, we introduced hierarchical SSL (H-SSL) based on a coarse-to-fine approach [19]. It roughly localizes sound sources using the pre-measured TFs, and consequently, it precisely localizes the sound source again around the estimated location using the interpolated TFs. By using this extension, H-SSL first confirms the resolution of SSL with pre-measured TFs, and later, it improves the resolution using the interpolated TFs. H-SSL facilitates the execution of super-resolution SSL in real time.

The rest of the study is organized as follows: Section 2 explains the details of TF interpolation by FTDLI for an improved resolution. Section 3 gives a brief introduction of SEVD-MUSIC and GEVD-MUSIC, and it describes GSVD-MUSIC and H-SSL to solve the concerns related to computational costs. Section 4 shows the system structure of SSL and SSS in our proposed robot audition system. Section 5 evaluates the techniques used in the system, and Section 6 concludes this study.

2 Transfer Function Interpolation Using FTDLI

To solve the resolution issue discussed in Section 1, FTDLI is specified to obtain TFs with the desired resolution for super-resolution SSL and SSS.

2.1 Related Works

In this section, frequency-domain linear interpolation (FDLI) [10, 15, 16] and time-domain linear interpolation (TDLI) [17] are explained.

Let \( \mathbf{A}(\omega, \psi_1) = [A_1(\omega, \psi_1), \ldots, A_M(\omega, \psi_1)]^T \in \mathbb{C}^M \) and \( \mathbf{A}(\omega, \psi_2) = [A_1(\omega, \psi_2), \ldots, A_M(\omega, \psi_2)]^T \in \mathbb{C}^M \) denote pre-measured TFs between a microphone array and sound sources, in other words, steering vectors. \( M \) denotes the number of microphones, \( \psi_1 \) and \( \psi_2 \) are the directions of the pre-measured TFs, and \( \omega \) represents the frequency. Our objective is to estimate \( \hat{\mathbf{A}}(\omega, \hat{\psi}) \) by interpolation, where \( \hat{\psi} \) is the direction of an estimated point, which is in the order \( \psi_1 < \hat{\psi} < \psi_2 \). FDLI [10] in the frequency domain interpolates a TF by:

\[
\hat{A}_{m[\psi_1, \psi_2]}(\omega, \hat{\psi}) = D_A A_m(\omega, \psi_1) + (1 - D_A) A_m(\omega, \psi_2),
\]

where \( \hat{A}_{m[\psi_1, \psi_2]}(\omega, \hat{\psi}) \) is an interpolated TF of the \( m \)-th microphone at \( \hat{\psi} \) using \( A_m(\omega, \psi_1) \) and \( A_m(\omega, \psi_2) \). \( D_A \in \mathbb{R} \) represents an interpolation factor, such that \( 0 \leq D_A \leq 1 \). TDLI [17] is described as follows:
With respect to the idea of correlation-matrix interpolation [18], we extended the method for TF interpolation. We integrated the phase interpolation results of FDLI and the amplitude interpolation results of TDLI to achieve the best interpolation performance. FDLI and TDLI are integrated by the following steps:

1) Take two interpolation results from Eqs. (1) and (4). Here, \( \hat{A}_{m|\psi_1,\psi_2}(t, \dot{\psi}) \) in Eqs. (1) and (4) are redefined as \( \hat{A}_{m|F|\psi_1,\psi_2}(\omega, \dot{\psi}) \) and \( \hat{A}_{m|T|\psi_1,\psi_2}(\omega, \dot{\psi}) \), respectively.

2) Decompose \( \hat{A}_{m|F|\psi_1,\psi_2}(\omega, \dot{\psi}) \) and \( \hat{A}_{m|T|\psi_1,\psi_2}(\omega, \dot{\psi}) \) into phase and gain.

\[
\hat{A}_{m|F|\psi_1,\psi_2}(\omega, \dot{\psi}) = \lambda_m[F] \exp(-j\omega t_{m[F]}) \\
\hat{A}_{m|T|\psi_1,\psi_2}(\omega, \dot{\psi}) = \lambda_m[T] \exp(-j\omega t_{m[T]})
\]

Fig. 2 shows the intuitive image of the TF interpolation by FDLI and TDLI, respectively. Both FDLI and TDLI can rapidly interpolate transfer functions between \( A(\omega, \psi_1) \) and \( A(\omega, \psi_2) \) continuously by a factor \( D_A \). Previously, we have evaluated their interpolation errors on the basis of amplitude and phase [18] and have found that FDLI and TDLI have problems in estimating the amplitude and phase, respectively.

### 2.2 Frequency- and Time-Domain Linear Interpolation

With respect to the idea of correlation-matrix interpolation [18], we extended the method for TF interpolation. We integrated the phase interpolation results of FDLI and the amplitude interpolation results of TDLI to achieve the best interpolation performance. FDLI and TDLI are integrated by the following steps:

1) Take two interpolation results from Eqs. (1) and (4). Here, \( \hat{A}_{m|\psi_1,\psi_2}(t, \dot{\psi}) \) in Eqs. (1) and (4) are redefined as \( \hat{A}_{m|F|\psi_1,\psi_2}(\omega, \dot{\psi}) \) and \( \hat{A}_{m|T|\psi_1,\psi_2}(\omega, \dot{\psi}) \), respectively.

2) Decompose \( \hat{A}_{m|F|\psi_1,\psi_2}(\omega, \dot{\psi}) \) and \( \hat{A}_{m|T|\psi_1,\psi_2}(\omega, \dot{\psi}) \) into phase and gain.

\[
\hat{A}_{m|F|\psi_1,\psi_2}(\omega, \dot{\psi}) = \lambda_m[F] \exp(-j\omega t_{m[F]}) \\
\hat{A}_{m|T|\psi_1,\psi_2}(\omega, \dot{\psi}) = \lambda_m[T] \exp(-j\omega t_{m[T]})
\]
3) Calculate \( \hat{A}_{m[\psi_1,\psi_2]}(\omega, \hat{\psi}) \) as follows:

\[
\hat{A}_{m[\psi_1,\psi_2]}(\omega, \hat{\psi}) = \lambda_m[T] \exp(-j\omega t_m[\hat{\psi}]) .
\] (7)

For SSL, \( \hat{A}_{m[\psi_1,\psi_2]}(\omega, \hat{\psi}) \) in Eq. (7) will be later utilized as \( A(\omega, \psi) \) in Eq. (10). For SSS, \( \hat{A}_{m[\psi_1,\psi_2]}(\omega, \hat{\psi}) \) will be utilized to set geometric constraints for separation in GHDSS. Then, we can select a desired resolution in SSL and SSS depending on the resolution of \( D_A \) in Eqs. (1) and (4).

3 SSL with GSVD-MUSIC and H-SSL

To solve the issue concerning computation costs, this section investigates computational-cost reduction by using GSVD-MUSIC and H-SSL. This section first gives a brief introduction of SEVD-MUSIC [2] and GEVD-MUSIC [4]. Next, it describes the details of GSVD-MUSIC and H-SSL.

3.1 SEVD-MUSIC and its Extension to GEVD-MUSIC

Before performing SSL, we need TFs (steering vectors), i.e., \( A(\omega, \psi) \). In Section 2, we obtained \( A(\omega, \psi) \) for a desired interval of \( \psi \) by measurements and interpolation.

In SSL, we initially perform a short-time Fourier transform of multi-channel input acoustic signals, denoted by \( X(\omega, f) \). The correlation matrix of \( X(\omega, f) \), denoted by \( R(\omega, f) \), is obtained as follows:

\[
R(\omega, f) = \frac{1}{T_R} \sum_{\tau_R=0}^{T_R-1} X(\omega, f + \tau_R) X^*(\omega, f + \tau_R) , \] (8)

where \((\cdot)^*\) is a complex conjugate transpose operator. We normalize \( R(\omega, f) \) over \( T_R \) frames to obtain a robust SSL against noise.

SEVD-MUSIC [2] performs SEVD of \( R(\omega, f) \) to decompose the signal space into noise- and signal-subspaces as follows:

\[
R(\omega, f) = E(\omega, f)A(\omega, f)E^{-1}(\omega, f) \] (9)

where \( A(\omega, f) = \text{diag}(\lambda_1(\omega, f), \ldots, \lambda_M(\omega, f)) \) and \( E(\omega, f) = [e_1(\omega, f), \ldots, e_M(\omega, f)] \) are Eigenvalues and vectors, respectively. Now, \( e_m(\omega, f) \) is sorted in the order of \( \lambda_m(\omega, f) \) \((1 \leq m \leq M)\). The spatial spectrum for SSL is determined by

\[
P(\omega, \psi, f) = \frac{|A^*(\omega, \psi)A(\omega, \psi)|}{\sum_{m=L_s+1}^{M} |A^*(\omega, \psi)e_m(\omega, f)|} ,
\] (10)

where \( L_s \) is the number of sound sources considered in SSL. To estimate the direction of arrival (DoA), we accumulate \( P(\omega, \psi, f) \) in Eq. (10) over \( \omega \) as follows:

\[
P(\psi, f) = \frac{1}{k_h - k_l + 1} \sum_{k=k_l}^{k_h} P(\omega[k], \psi, f) ,
\] (11)

where \( k_h \) and \( k_l \) are the frequency bin indices, which represent the maximum and minimum frequencies for SSL, respectively.
As an extension of SEVD-MUSIC, GEVD-MUSIC whitens the arbitrary noise in SSL even when the power of the noise is higher than that of target sound sources [4]. GEVD-MUSIC extends Eq. (9) to perform GEVD as follows:

\[
K^{-1}(\omega, f)R(\omega, f) = E(\omega, f)\Lambda(\omega, f)E^*(\omega, f),
\]

where \(K(\omega, f)\) is a freely-designable correlation matrix and can be utilized for various purposes. For instance, when the high power noise \(N(\omega)\) exists, \(K(\omega, f)\) is designed as follows:

\[
K(\omega, f) = \frac{1}{T_K} \sum_{\tau_K=0}^{T_K-1} N(\omega, f + \tau_K)N^*(\omega, f + \tau_K).
\]

Eq. (13) makes the Eigenvalues in noise subspaces \((\lambda_m(\omega, f)|_{L_s+1\leq m \leq M})\) to be one, which means that the noise becomes spatially white in the subspaces. Since the Eigenvalues in signal subspaces \((\lambda_m(\omega, f)|_{1\leq m \leq L_s})\) remain to be greater than one, we can correctly separate the signal space into signal- and noise-subspaces even if the noise is directional/diffuse. Despite this theoretical basis, there are two main practical limitations to maintain the high performance in the experiments. First, the Eigenvalues in noise subspace become exactly one only if \(K(\omega, f)\) perfectly matches the existing noise in the input acoustic signal. Second, the Eigenvalues in signal subspaces become close to one when having high power diffuse noise or directional noise in the same direction as target sound sources. With these noises, Eq. (13) have a difficulty in separating the signal space into signal- and noise-subspaces since all the Eigenvalues are close to one. In Section 5.2, the SSL performance with a robot’s fan noise is shown. This demonstrates a practical robot application addressing a robot’s ego-noise using our method.

### 3.2 GSVD-MUSIC

Theoretically, GEVDs of \(K(\omega, f)\) and \(R(\omega, f)\) are equivalent and can be described as the following SEVD:

\[
K^{-\frac{1}{2}}(\omega, f)R(\omega, f)K^{-\frac{1}{2}}(\omega, f) = E(\omega, f)\Lambda(\omega, f)E^*(\omega, f).
\]

Nakamura et al. [4] considered Eq. (12) as GEVD instead of Eq. (14) because it whitened the noise without calculating \(K^{-\frac{1}{2}}(\omega, f)\), which has a large calculation cost. However, Eq. (12) has the following problems:

- The SEVD still has a large calculation cost for frame-by-frame localization while working in real time.
- The Eigenvectors are not mutually orthogonal because \(K^{-1}(\omega, f)R(\omega, f)\) is not hermitian, which eventually degrades the SSL performance.

To solve these problems, we extended Eq. (12) to utilize the following GSVD:

\[
K^{-1}(\omega, f)R(\omega, f) = E_l(\omega, f)\Lambda(\omega, f)E_r^*(\omega, f),
\]

where \(K(\omega, f)\) is a freely-designable correlation matrix and can be utilized for various purposes. For instance, when the high power noise \(N(\omega)\) exists, \(K(\omega, f)\) is designed as follows:

\[
K(\omega, f) = \frac{1}{T_K} \sum_{\tau_K=0}^{T_K-1} N(\omega, f + \tau_K)N^*(\omega, f + \tau_K).
\]
where $E_l(\omega, f)$ and $E_r(\omega, f)$ are left- and right-singular vectors, respectively, which are unitary and mutually orthogonal. $E_l(\omega, f)$ is used instead of $E(\omega, f)$ in Eq. (9).

The calculation cost of Eq. (15) is less than that of Eq. (12), which is evaluated in Section 5.

We note the following aspects of GSVD-MUSIC. First, GSVD-MUSIC is equivalent to the following GEVD-MUSIC:

$$R^2 e_m = \lambda_m K^2 e_m$$  \hspace{1cm} (16)

Therefore, the Eigenvectors are mutually orthogonal, which improves the SSL performance compared to the results of Nakamura et al. [4].

In addition, GSVD-MUSIC is equivalent to SEVD-MUSIC when $K = I$, where $I \in \mathbb{C}^{M \times M}$ is an identity matrix. Therefore, we can reduce the calculation cost of SEVD-MUSIC with GSVD-MUSIC because GSVD is less computationally demanding than SEVD.

### 3.3 H-SSL for Fast Super-resolution SSL

Although we can obtain fine TFs by using FTDLI, super-resolution SSL is computationally expensive. Therefore, it is unsuitable for real-time processing for robotic applications. To solve the issue, we introduced H-SSL.

In GSVD-MUSIC, $P(\omega, \psi, f)$ as in Eq. (10) and $\bar{P}(\psi, f)$ as in Eq. (11) are $\psi$-dependent processes. In these processes, the finer values of $\hat{A}_{m[\psi_1, \psi_2]}(\omega, \hat{\psi})$ increase the calculation cost linearly.

In H-SSL, we reduced the number of processed TFs while maintaining the resolution by the following steps:

1) Conduct SSL with pre-measured TFs with a rough resolution, and search for the peaks of spatial spectrum in Eq. (11). Let $\psi^{[l]}$ be the direction that has the $l$-th largest $\bar{P}(\psi, f)$, where $l$ is the index of sound sources ($1 \leq l \leq L_s$).

2) Take the two closest $\psi$ from $\psi^{[l]}$, which are denoted as $\psi^{[l-]}$ and $\psi^{[l+]}$. Suppose $\psi^{[l-]} < \psi^{[l]} < \psi^{[l+]}$.

3) Generate $\hat{A}_{m[\psi^{[l-]}, \psi^{[l]}]}(\omega, \hat{\psi})$ and $\hat{A}_{m[\psi^{[l]}, \psi^{[l+]}]}(\omega, \hat{\psi})$ using Eq. (7), depending on a given finer resolution.

4) Conduct SSL again only with $\hat{A}_{m[\psi^{[l-]}, \psi^{[l]}]}(\omega, \hat{\psi})$ and $\hat{A}_{m[\psi^{[l]}, \psi^{[l+]}]}(\omega, \hat{\psi})$, and search for the peaks of Eq. (11).

The upper layer of the hierarchical process is used for broad localization, whereas the lower layer provides a finer localization. This means that SSL with H-SSL follows coarse-to-fine [19] localization, thereby giving rough SSL results initially, followed by fine SSL results.

### 4 System Structure

Figs. 3 and 4 show the system structures of super-resolution SSL and SSS in our proposed robot audition system, respectively to achieve robust, simultaneous speech recognition while working in real time. The
algorithm in SSL part is discussed in Sections 2 and 3. This section provides an overview of the SSS part in Fig. 4 and its implementation.

4.1 Super-resolution SSS

In our robot audition system, we used GHDSS [3] for the SSS module especially to separate speech sources and suppress directional noise sources. GHDSS integrates high-order decorrelation-based source separation (HDSS) and geometric constraints (GC). GC introduces TFs in order to overcome permutation and scaling problems. Furthermore, GHDSS features an adaptive step-size control for fast separation in a dynamic environment.

We previously confirmed that introduction of GC (namely TFs) to HDSS improved the SNR of separated sound source[3]. Therefore, the purpose of Super-resolution SSS is to utilize interpolated TFs to make GC with a desired resolution. This paper discusses only the application to GHDSS while the method can be generally applicable to other TF-based methods such as geometric-constrained independent component analysis[20], geometric-constrained source separation[21], linearly-constrained minimum variance[22], and Griffiths-and-Jim adaptive beamformer[23], etc.

Mathematically, the cost function for SSS in GHDSS, denoted as $J_{GHDSS}(W)$, is described as follows:

$$
J_{GHDSS}(W) = \alpha J_{HDSS}(W) + \beta J_{GC}(W),
$$

(17)

where $W$, $J_{HDSS}(W)$, and $J_{GC}(W)$ denote a separation matrix, cost function for HDSS, and cost function for GSS, respectively; $\alpha$ and $\beta$ are weights for the two algorithms, where $\alpha + \beta = 1$. Now, $J_{GC}(W)$ is specifically explained because interpolated TFs affect the term. $J_{GC}(W)$ is a quadratic cost.
function with a geometric constraint, described as follows:

\[
J_{GC}(W) = \| E_{GC} \|^2, \\
E_{GC} = \text{diag}(WD - I), \\
\text{(18)}
\]

where \(D\) is a mixing matrix based on the direct sound paths between localized sound sources and a microphone array, namely \(A(\omega, \psi)\), for localized sound sources. We improved the SSS performance by introducing FTDLI to the geometric constraint in Eq. (18).

4.2 Implementation

All blocks explained above were implemented with a humanoid robot, called Hearbo [see Fig. 5(a)], located in a typical room in which the reverberation time was 0.2 s \((RT_{20})\). Fig. 5(b) shows the coordinate system for \(\psi\). We utilized an 8-ch circular microphone array embedded in the robot’s head and measured the TFs \(A(\omega, \psi)\) at every 1°, which were obtained by time-stretched pulse recordings. The acoustic signal was sampled with 16 kHz and 16 bits. The window and shift length for frequency analysis were set to 512 and 160 samples, respectively. All the proposed functions were implemented as modules for the robot audition software, HARK \cite{24}. The system operated in real time with a laptop having a 2.0 GHz Intel Core i7 CPU and 8GB SDRAM running Linux.

5 Experimental Validation

This section shows five types of evaluations as follows:

1) Error in TF interpolation by FTDLI
2) Noise-robustness and computational cost comparison among SEVD-, GEVD-, GSVD-MUSIC
3) The computational cost of H-SSL
4) Applicability of the proposed methods to SSL for a robot
5) Applicability of the proposed methods to simultaneous speech recognition including super-resolution SSS

5.1 Error of TF interpolation Using FTDLI

We evaluated the estimation errors of three interpolation methods, namely FDLI, TDLI, and FTDLI. For general evaluation, we used the following two types of reference TFs:

- TFs measured by our robot microphone array,
- KEMAR TFs [25] ¹.

We considered the difference between the estimated $\hat{A}_{[\psi_1, \psi_2]}(\omega, \hat{\psi})$ and pre-measured $A(\omega, \psi)$. The parameter $\psi_1$ was fixed at 0°, and $\psi_2 = \{30^\circ, 60^\circ, 90^\circ, 120^\circ\}$ was used. $\hat{A}_{[\psi_1, \psi_2]}(\omega, \hat{\psi})$ was estimated by every 1° for our robot TFs and 5° for the KEMAR TFs. We averaged the estimated errors for $\omega$ and $\hat{\psi}$. The average error, $\bar{e}_{[\psi_1, \psi_2]}$, was calculated as follows:

$$\bar{e}_{[\psi_1, \psi_2]} = \frac{1}{k_\psi} \sum_{i=1}^{k_\psi} \frac{1}{k_h - k_l + 1} \sum_{k=k_l}^{k_h} f_{[\psi_1, \psi_2]}(\omega[k], \hat{\psi}[i]) ,$$

where $f_{[\psi_1, \psi_2]}(\omega[k], \hat{\psi}[i])$ is the estimation error for $\hat{\psi}$ and $\omega$. The values $k_l$ and $k_h$ are same as those in Eq. (11) with the frequency band $500[\text{Hz}] \leq \omega \leq 2800[\text{Hz}]$. The number of $\hat{\psi}$ on which we conducted interpolation is represented by $i_\psi$. For $\hat{\psi}$, we utilized all $\psi$ of the pre-measured $A(\omega, \psi) = [A_1(\omega, \psi), \ldots, A_M(\omega, \psi)]^T$ in the range $\psi_1 < \psi < \psi_2$.

We evaluated three types of $f_{[\psi_1, \psi_2]}(\omega[k], \hat{\psi}[i])$ in Eq. (19).

The first criterion is the summation of normalized inner products of all channels:

$$f_{1[\psi_1, \psi_2]}(\omega, \hat{\psi}) = \sum_{m=1}^{M} \left| A_m(\omega, \hat{\psi}) \cdot \hat{A}_{m[\psi_1, \psi_2]}(\omega, \hat{\psi}) \right| - 1 ,$$

which represents the phase estimation error (PEE).

The second criterion is the summation of spectral distortion (SD) of all channels:

$$f_{2[\psi_1, \psi_2]}(\omega, \hat{\psi}) = \sum_{m=1}^{M} 20 \log \left| \frac{A_{m[\psi_1, \psi_2]}(\omega, \hat{\psi})}{A_m(\omega, \psi)} \right| ,$$

which shows the amplitude-estimation performance.

The third criterion is the summation of signal-to-distortion ratio (SDR) of all channels ²:

$$f_{3[\psi_1, \psi_2]}(\omega, \hat{\psi}) = \sum_{m=1}^{M} \left( \frac{|A_m(\omega, \hat{\psi}) - \hat{A}_{m[\psi_1, \psi_2]}(\omega, \hat{\psi})|}{|A_m(\omega, \psi)|} \right)^2 ,$$

which represents the total estimation performance. Let $\bar{e}_{1[\psi_1, \psi_2]}$, $\bar{e}_{2[\psi_1, \psi_2]}$, and $\bar{e}_{3[\psi_1, \psi_2]}$ denote $\bar{e}_{[\psi_1, \psi_2]}$ of PEE, SD, and SDR, respectively.

¹See the detail of TF in [25].
²We have inverted the original SDR discussed in [10] since it shows the best estimation performance with $f_{1[\psi_1, \psi_2]}(\omega, \hat{\psi}) = 0$. 

11
using KEMAR TFs. The horizontal and vertical axes indicate the measurement units, same as those in a widely known as a standard binaural HRTF dataset. Fig. 7 shows the comparison of average errors achieved by TDLI because of integration. Hence, it had the smallest $\bar{D}$ where intuitive parameter determination.

The horizontal and vertical axes in all the figures show the difference between $\hat{\psi}_i$ and $\psi_i$. Figure 6 shows the comparison of average errors using TFs measured by a robot-embedded microphone array, namely $\bar{\bar{D}}$, $\bar{D}_e$, and $\bar{D}_d$, and their linearity of $D_A$, namely $\bar{D}_1$, $\bar{D}_2$, and $\bar{D}_3$, respectively. Figure 7: Interpolation error and the linearity of $D_A$ for PEE, SD, SDR using KEMAR TFs

In addition, we evaluated the linearity of $D_A$ calculated by

$$\bar{d}_{\psi_1,\psi_2} = \frac{1}{i_\psi} \sqrt{\sum_{i=1}^{i_\psi} \left( D_A[i] - \frac{\hat{\psi}_i - \psi_i}{\psi_1 - \psi_2} \right)^2},$$

where $D_A[i]$ denotes $D_A$ having the smallest $f_{\psi_1,\psi_2}(\omega[i], \hat{\psi}_i)$ for each $\hat{\psi}_i$. A small $\bar{d}_{\psi_1,\psi_2}$ means that $D_A$ is approximately equal to $\frac{\hat{\psi}_i - \psi_i}{\psi_1 - \psi_2}$, which is utilized for practical interpolation. Let $\bar{d}_1[\psi_1,\psi_2]$, $\bar{d}_2[\psi_1,\psi_2]$, and $\bar{d}_3[\psi_1,\psi_2]$ denote $\bar{d}_{\psi_1,\psi_2}$ of PEE, SD, and SDR, respectively.

Fig. 6 shows the comparison of average errors using TFs measured by a robot-embedded microphone array, namely $\bar{e}_1$, $\bar{e}_2$, and $\bar{e}_3$, and their linearity of $D_A$, namely $\bar{d}_1$, $\bar{d}_2$, and $\bar{d}_3$, respectively.

The horizontal axis in all the figures shows the difference between $\hat{\psi}_1$ and $\psi_2$ for the interpolation.

The $\bar{e}_1$ achieved by FTDLI was as small as that achieved by FDLI, and the $\bar{e}_2$ achieved was as small as that achieved by TDLI because of integration. Hence, it had the smallest $\bar{e}_3$. Also, FTDLI showed the smallest $\bar{d}_1$, $\bar{d}_2$, and $\bar{d}_3$. Therefore, FTDLI has advantages with respect to both high accuracy and intuitive parameter determination.

To generalize our approach, we also applied the proposed interpolation for KEMAR TFs, which is widely known as a standard binaural HRTF dataset. Fig. 7 shows the comparison of average errors using KEMAR TFs. The horizontal and vertical axes indicate the measurement units, same as those in
Clearly, similar to the results in Fig. 6, FTDLI shows the best interpolation performance. Thus, we confirmed the general applicability of the proposed interpolation method.

5.2 Noise-robustness and computational-cost comparison among SEVD-, GEVD-, GSVD-MUSIC

In the experiment, there was a target sound 1m away from the microphone array in the direction of 60°. Since human speech has short pauses and is difficult to define ground truth of voice activity, we used white noise as a target sound, which has uniform and steady power for the whole frequency range. We recorded a robot’s fan noise as a noise source. Since the fan was located on the robot’s back and close to the microphone array, the noise showed a high peak at 180° with diffuseness.

In this experiment, we used only pre-measured TFs to exclude the effect of interpolation performance and evaluate only SSL performance. A(\(\omega,\psi\)) was measured with \(\psi = \{-175°, -170°, ..., 180°\}\). Therefore, the resolution of SSL was 5°.

We pre-measured the robot’s fan noise to create \(K(\omega,f)\) in Eq. (13). We set \(T_R = T_K = 25\) in Eq. (8) and Eq. (13), respectively.

We evaluated the relationship between the signal-to-noise ratio (SNR) and localization accuracy. SNR is defined as follows:

1) The calculation of the average spectrum of \(M\)-ch input acoustic signals, defined as \(X_{sa}(\omega)\). The power spectrum density (PSD) of \(X_{sa}(\omega)\) is derived as follows:

\[
P_{sa}(\omega) = \frac{1}{k_{wl}}X_{sa}(\omega)X_{sa}^*(\omega),
\]  

where \(k_{wl}\) is the window length.

2) The determination of PSD of the noise source \(P_{na}(\omega)\) by using the same process as that shown above.

3) The calculation of SNR for each frequency bin, and the normalization of SNR of the bins between  

\[^3\text{In Fig. 11(b) in Section 5.5, we conducted SSL for four simultaneous speeches and confirmed that our proposed SSL could localize them during voice activity periods.}\]

\[^4\text{For SSL with more diffuse noise, we evaluated our method with a robot’s motor noise[26] and confirmed the validity.}\]

\[^5\text{This definition is the same as that in [27]. This paper includes the result of GSVD-MUSIC.}\]
We observed that (a) GSVD- and GEVD-MUSIC showed better performance than that of SEVD-MUSIC (b) GSVD-MUSIC showed the best performance. (a) confirms that subspace decomposition with $K(\omega, f)$ works for whitening a robot’s practical fan noise even under the limitations discussed in Section 3.1. (b) means that GSVD can obtain mutually-orthogonal signal- and noise-subspaces. Thus, we confirmed the improvement of noise-robustness by the proposed method.

In addition, the computational cost of each method was evaluated. We conducted SSL with 1000 frames for each method and measured the averaged processing time only for Eqs. (9)-(12), and (15). Therefore, the average processing time for SEVD-, GEVD-, and GSVD-MUSIC methods was 11.9 ms, 13.6 ms, and 5.52 ms, respectively. Clearly, GSVD-MUSIC showed a considerable improvement in the computational cost when compared to that of SEVD-MUSIC (approximately two times faster). Because the frame period was 10 ms, only GSVD-MUSIC could function frame-by-frame in real time.

Finally, we confirmed that GSVD-MUSIC was the best method in terms of both noise-robustness and computational efficiency.

### 5.3 Computational cost of H-SSL

We compared the computational cost of SSL with and without hierarchy, as explained in Section 3.3. As in Section 5.2, we conducted SSL with 1000 frames for each method and measured the averaged processing time only for Eqs. (9)-(12), and (15). For SSL without hierarchy, $A(\omega, \psi)$ of 1-degree intervals were used to obtain a higher resolution. For H-SSL, $A(\omega, \psi)$ of 10-degree intervals were used as the pre-measured TFs, and we conducted FTDLI to generate $\hat{A}_{m[\psi_1, \psi_2]}(\omega, \hat{\psi})$ of 1-degree intervals. Therefore, these two SSLs have the same resolution.

Tables 1 and 2 show the average processing time for SEVD-, GEVD-, and GSVD-MUSIC methods with and without H-SSL, respectively. The tables include the change in $L_s$ in Eq. (10), which affects the hierarchical process. Compared to Table 2, Table 1 shows less than half of the processing time regardless of $L_s$. As observed in case of GSVD-MUSIC with SSL, 71.7% of the computational cost was reduced, which shows the validity of H-SSL. Moreover, only the integration of H-SSL and GSVD-MUSIC achieved
Table 1: Frame processing time with H-SSL

<table>
<thead>
<tr>
<th>Source</th>
<th>1 src</th>
<th>2 srcs</th>
<th>3 srcs</th>
<th>4 srcs</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEVD-MUSIC</td>
<td>11.3ms</td>
<td>12.3ms</td>
<td>13.0ms</td>
<td>13.3ms</td>
</tr>
<tr>
<td>GEVD-MUSIC</td>
<td>14.7ms</td>
<td>15.9ms</td>
<td>16.3ms</td>
<td>16.5ms</td>
</tr>
<tr>
<td>GSVD-MUSIC</td>
<td>6.8ms</td>
<td>7.7ms</td>
<td>8.3ms</td>
<td>8.7ms</td>
</tr>
</tbody>
</table>

Table 2: Frame processing time without H-SSL

<table>
<thead>
<tr>
<th>Source</th>
<th>1 src</th>
<th>2 srcs</th>
<th>3 srcs</th>
<th>4 srcs</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEVD-MUSIC</td>
<td>30.1ms</td>
<td>28.6ms</td>
<td>24.0ms</td>
<td>21.7ms</td>
</tr>
<tr>
<td>GEVD-MUSIC</td>
<td>32.1ms</td>
<td>29.8ms</td>
<td>27.3ms</td>
<td>25.5ms</td>
</tr>
<tr>
<td>GSVD-MUSIC</td>
<td>24.1ms</td>
<td>22.0ms</td>
<td>19.6ms</td>
<td>17.5ms</td>
</tr>
</tbody>
</table>

faster processing than the frame period (10 ms), which successfully confirmed real-time, frame-by-frame super-resolution SSL.

5.4 Applicability of the proposed methods to SSL for a robot

We applied GSVD-MUSIC, FTDLI, and H-SSL to our robot-embedded microphone array. We evaluated our methods by using two measures: (1) Error in DoA estimation toward a fixed sound source and (2) SSL performance toward a dynamic sound source.

5.4.1 SSL for a Fixed Sound Source

This section compares the error in estimating DoA toward a fixed sound source using different interpolation methods. A target sound source (white noise) was recorded for every 1° of ψ (−90° ≤ ψ ≤ 90°). In this experiment, we excluded the robot’s fan noise to see the effectiveness of interpolation. We localized the white noise and obtained the average error between the estimated and recorded DoAs of the white noise. We assumed having \( A(\omega, \psi) \) at the intervals of \( \psi = \{1°, 5°, 10°, 30°, 60°, 90°, 120°\} \). By interpolation, \( \hat{A}_{[\psi_1, \psi_2]}(\omega, \hat{\psi}) \) of 1-degree intervals was estimated so that we could accurately localize the sound source.

Fig. 9 shows the comparison of DoA estimation errors among the interpolation methods discussed in Section 2. The horizontal axis shows the intervals of \( \psi \) for \( A(\omega, \psi) \). The vertical axis shows the average error of DoA estimation. “NONE” means that no interpolation method was used.

We observed that (a) FTDLI had the smallest DoA estimation errors and (b) up to the intervals of 30°, FTDLI maintained approximately the same performance as the case at 1-degree intervals, which showed the validity of FTDLI in SSL.

5.4.2 SSL for a Dynamic Sound Source

We used a moving white noise source. All other conditions are the same as those in Section 5.4.1.

Similar to Fig. 9, Fig. 10(a) shows the comparison between DoA estimation errors with and without FTDLI. FTDLI performed better than NONE, and it maintained the SSL performance until the intervals...
Fig. 10(b) shows the deletion error rate in SSL. The deletion error rate was defined as the rate of frames that missed the DoA estimation results. FTDLI perfectly localized the sound source until the intervals of 90°.

Fig. 10(c) shows an SSL example using $A(\omega, \psi)$ at 30-degree intervals. Intervals of 30° were selected considering the evaluation results in Figs. 9 and 10(a). The horizontal axis shows the frame index, and the vertical axis shows the DoA estimation results of $\psi$. In addition to the SSL results, we plotted the reference trajectory, recorded by an ultrasonic positioning system [28]. The SSL with FTDLI showed a better and smoother trajectory than that with NONE.

Therefore, we could confirm the validity of the SSL system in both static and dynamic environments.

5.5 Applicability of the proposed methods to simultaneous speech recognition including super-resolution SSS

Finally, we applied the proposed methods to the system, shown in Fig. 1, to improve the performance of simultaneous speech recognition.

The proposed super-resolution SSL system in Fig. 3 was first applied to the SSL module in Fig. 1. We performed a coarse voice-activity detection of the simultaneous speech in the SSL module, based on tracking the individual speech source.

By utilizing the tracking result for the individual speech source, the super-resolution SSS system in Fig. 4 separated the speech sources.

To improve the simultaneous speech-recognition performance, we conducted speech enhancement between GHDSS and acoustic feature extraction. We used histogram-based recursive level estimation (HRLE) [29]. Because HRLE uses recursive averages, it calculates a time-varying histogram in real
time. Therefore, the noise-level estimation adapts to the environmental changes smoothly and rapidly.

In the experiment, there were four people 1-m away from the robot in the directions of \(-60^\circ, -20^\circ, 20^\circ, \text{ and } 60^\circ\), respectively. Table 3 shows the conditions for automatic speech recognition. We selected the dataset for evaluation in order to conduct word- and speaker-open tests. We simulated the simultaneous speech using an ATR Japanese word dataset, which included 2160 words (216 words for five female and five male speakers). We used an acoustic model trained with the ATR dataset and the Japanese Newspaper Article Sentences (ASJ-JNAS) corpus, which comprises 60 hours of spoken data by 306 male and female speakers. The acoustic model used a tied-state triphone HMM whose numbers of states and mixtures were 3 and 8, respectively. For feature extraction, we used 13 static Mel-Scale Log Spectrum (MSLS) \[30\] features, 13 delta MSLS features, and one delta power feature. Speech recognition results are given as average word correct rates (WCR) of instances from the noisy test set. WCR is normally defined for a single speaker. Because this evaluation has four simultaneous speakers, we used all the test set for each speaker, and WCR of all the speakers are averaged.

We conducted the following two experiments: (1) a comparison of WCRs when the locations, timing, and number of speech sources are known and (2) a comparison of WCRs of the entire robot audition system, as shown in Fig. 1.

The first experiment evaluated the SSS performance depending on conventional/proposed interpolation methods. The second experiment simulated a more realistic application, which includes SSL. In the second experiment, TF interpolation methods, namely NONE, FDLI, TDLI, and FTDLI were applied for both SSL and SSS.

To evaluate the TF-interpolation methods, we assumed having \(A(\omega, \psi)\) at the intervals of \(\psi = \{1^\circ, 5^\circ, 10^\circ, 30^\circ, 60^\circ, 90^\circ\}\), similar to Figs. 9 and 10. We omitted the case with \(\psi = 120^\circ\) because it uses \(A(\omega, \pm 60^\circ)\), which are TFs of the speaker directions. Similar to the evaluations in Section 5.4, \(\hat{A}_{[\psi_1, \psi_2]}(\omega, \hat{\psi})\) of 1-degree intervals were estimated in order to accurately localize the sound source.

Fig. 11 shows the WCR comparison among the interpolation methods, discussed in Section 2. The horizontal axis shows the intervals of \(\psi\) for \(A(\omega, \psi)\). The vertical axis shows WCR. As specified in Figs. 9 and 10, NONE means that no interpolation method was used. In the figure, WCRs for \(\psi = 1^\circ\) using FDLI, TDLI, and FTDLI are not plotted because the case with \(\psi = 1^\circ\) does not need interpolation.

Fig. 11(a) shows the WCR comparison for the first experiment. Because this case omitted SSL, the difference among the four interpolation methods is the TFs used for geometric constraints in GHDSS.

Table 3: Conditions for Automatic Speech Recognition

<table>
<thead>
<tr>
<th>Acoustic Model</th>
<th>Model: Tied-state triphone HMM (3 states, 8 mixtures)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training dataset</td>
<td>ASJ-JNAS + ATR dataset (60 hours, 306 male and female speakers)</td>
</tr>
<tr>
<td>Test dataset</td>
<td>ASJ-JNAS (216 words \times 10 speakers, 5 male and 5 female speakers)</td>
</tr>
<tr>
<td>Acoustic Feature</td>
<td>27 dimensions (MSLS (12 dimensions) + \Delta MSLS + \Delta power)</td>
</tr>
<tr>
<td>Language Model</td>
<td>Statistical Language Model (JNAS, 20000 words)</td>
</tr>
</tbody>
</table>
The figure shows: (a) SSS performances of NONE and TDLI were degraded when the interval was more than 30° and (b) FTDLI and FDLI performed approximately the same but showed 0%-1% improvement in FTDLI when the interval was greater than or equal to 30°. Because GHDSS includes not only geometric constraints in GSS but also HDSS, the adaptation in GHDSS makes the difference with respect to smaller TFs. However, FTDLI maintained the best ASR performance compared to all other methods.

Finally, Fig. 11(b) shows the WCR comparison for the second experiment, which evaluated the entire robot audition system. Producing different results than those shown in Fig. 11(a), the evaluation included the SSL system shown in Fig. 3, which utilizes TFs interpolated by four methods. The figure shows that (a) NONE, FDLI, and FTDLI maintained their ASR performance up to 10-degree intervals, (b) the performance of NONE drastically degraded from 30-degree intervals, (c) FTDLI maintained at least the same performance as that with FDLI when having finer TFs than 60-degree intervals, and (d) FTDLI improved the performance approximately 7% compared to that with FDLI from 60-degree intervals. Because SSL contained deletion errors, shown in Fig. 10(b), the system could not separate all the speech sources, which degraded the ASR performance drastically compared to the results shown in Fig. 11(a). In this experiment, we used the test set of 10 speakers for each direction, indicating 40 subjects’ speeches in total. Since we confirmed (c) and (d) for all 40 subjects, the performance of FTDLI is statistically significant compared to that of FDLI.

Finally, we could improve not only the SSL performance but also the performance of the entire robot audition system by using our proposed methods.

6 Conclusion

This study investigated a super-resolution robot audition system to improve the performance of simultaneous speech recognition. Because a robot audition system should work in real-time in a real environment with a sufficiently high accuracy, we focused on two issues: a pre-defined resolution based on the resolution of measurements and a high computational cost for conducting GEVD and high-resolution SSL. To solve them, we proposed an interpolation of TFs by FTDLI and a computational cost reduction by GSVD-MUSIC and H-SSL.
All the proposed functions were integrated into super-resolution SSL and SSS systems. Finally, each proposed technique was evaluated, and the systems were applied to simultaneous speech recognition. The evaluation showed that (1) FTDLI showed a better interpolation performance compared to the existing methods and provided super-resolution SSL, (2) GSVD-MUSIC reduced the computational cost approximately by half and achieved a better noise-robustness compared to that by GEVD-MUSIC, (3) H-SSL reduced the computational cost drastically, and only the integration of GSVD-MUSIC and H-SSL achieved super-resolution SSL in real time, and (4) the super-resolution SSL and SSS with FTDLI improved the performance of ASR of simultaneous speech. All these results successfully confirmed the validity of the proposed techniques in the entire robot audition system.

Our future study will be an extension of FTDLI to a three-dimensional case and the construction of a real-time three-dimensional super-resolution robot audition system.

REFERENCES


