

Determined BSS Based on Time-Frequency Masking and Its Application to Harmonic Vector Analysis

Kohei Yatabe , *Member, IEEE*, and Daichi Kitamura , *Member, IEEE*

Abstract—This paper proposes harmonic vector analysis (HVA) based on a general algorithmic framework of audio blind source separation (BSS) that is also presented in this paper. BSS for a convolutive audio mixture is usually performed by multichannel linear filtering when the numbers of microphones and sources are equal (determined situation). This paper addresses such determined BSS based on batch processing. To estimate the demixing filters, effective modeling of the source signals is important. One successful example is independent vector analysis (IVA) that models the signals via co-occurrence among the frequency components in each source. To give more freedom to the source modeling, a general framework of determined BSS is presented in this paper. It is based on the plug-and-play scheme using a primal-dual splitting algorithm and enables us to model the source signals implicitly through a time-frequency mask. By using the proposed framework, determined BSS algorithms can be developed by designing masks that enhance the source signals. As an example of its application, we propose HVA by defining a time-frequency mask that enhances the harmonic structure of audio signals via sparsity of cepstrum. The experiments showed that HVA outperforms IVA and independent low-rank matrix analysis (ILRMA) for both speech and music signals. A MATLAB code is provided along with the paper for a reference.

Index Terms—Blind source separation (BSS), independent component analysis (ICA), cepstrum analysis, Wiener-like mask, plug-and-play scheme, proximal splitting algorithm.

I. INTRODUCTION

BLIND source separation (BSS) is a methodology to recover the source signals from multiple mixtures (audio recordings in the case of this paper) without any knowledge about the mixing system. Let a convolutive mixing process of the signals be approximated in the time-frequency domain as

$$\mathbf{x}[t, f] \approx \mathbf{A}[f]\mathbf{s}[t, f], \quad (1)$$

where $\mathbf{x} = [x_1, x_2, \dots, x_M]^T \in \mathbb{C}^M$ is the observed mixtures obtained by M microphones, $\mathbf{s} = [s_1, s_2, \dots, s_N]^T \in \mathbb{C}^N$ is the vector of source signals to be recovered, N is the number of source signals that is assumed to be known, $\mathbf{A}[f] \in \mathbb{C}^{M \times N}$ is the

mixing matrix, and $t = 1, \dots, T$ and $f = 1, \dots, F$ are indices of time and frequency, respectively. Throughout this paper, each element of multichannel signals in $\mathbb{C}^{M \times T \times F}$ is denoted by $x_m[t, f] \in \mathbb{C}$, and the m th spectrogram is represented as $x_m[\cdot, \cdot] \in \mathbb{C}^{T \times F}$. The aim of BSS is to recover the unknown source signals, \mathbf{s} , only from the mixtures, \mathbf{x} . This paper considers batch processing, i.e., mixtures for all t are given beforehand. In the determined ($M = N$) or overdetermined ($M > N$) situation, the usual strategy for solving the BSS problem is to formulate an estimation problem of finding (or approximating) a demixing matrix, $\mathbf{W}[f] \in \mathbb{C}^{N \times M}$, that is a left inverse of $\mathbf{A}[f]$ (i.e., $\mathbf{W}[f]\mathbf{A}[f] = \mathbf{I}$, where \mathbf{I} is the identity matrix). Then, the source signals are recovered by multiplication of the estimated demixing matrix as follows:

$$\mathbf{W}[f]\mathbf{x}[t, f] \approx \mathbf{W}[f]\mathbf{A}[f]\mathbf{s}[t, f] = \mathbf{s}[t, f]. \quad (2)$$

By reducing the BSS problem into the demixing matrix estimation problem, the difficulty of directly tackling the unknown mixing process in Eq. (1) is circumvented. This paper focuses on the above formulation of the (over)determined BSS, where the demixing matrix $\mathbf{W}[f]$ is estimated for all f only from the observed data $\mathbf{x}[t, f]$.

Statistical independence between the source signals is the well-accepted assumption for handling this ill-posed problem [1], [2]. While there exists several formulations depending on a method to measure the independence, many of them fall into a minimization problem of the following form [1]:

$$\underset{(\mathbf{W}[f])_{f=1}^F}{\text{Minimize}} \sum_{n=1}^N \mathcal{P}_n(\mathbf{W}[f]\mathbf{x}[t, f]) - \sum_{f=1}^F \log |\det(\mathbf{W}[f])|, \quad (3)$$

where the log-determinant term is obtained from either maximum likelihood (ML) estimation or minimizing mutual information [1], and \mathcal{P}_n is a real-valued function corresponding to the model of the n th source (in the case of ML estimation, $C \exp(-\mathcal{P}_n(\cdot))$ corresponds to the density function of the n th source). For example, with some constant C , the ℓ_1 norm,

$$\mathcal{P}_n(\mathbf{y}[t, f]) = C \|\mathbf{y}_n[\cdot, \cdot]\|_1 = C \sum_{t=1}^T \sum_{f=1}^F |y_n[t, f]|, \quad (4)$$

Manuscript received August 31, 2020; revised February 3, 2021 and March 19, 2021; accepted April 14, 2021. Date of publication April 16, 2021; date of current version May 7, 2021. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Huseyin Hacihabiboglu. (Corresponding author: Kohei Yatabe.)

Kohei Yatabe is with the Department of Intermedia Art and Science, Waseda University, Tokyo 169-8555, Japan (e-mail: k.yatabe@asagi.waseda.jp).

Daichi Kitamura is with the Department of Electrical and Computer Engineering, National Institute of Technology, Kagawa College, Kagawa 7618058, Japan (e-mail: kitamura-d@t.kagawa-nct.ac.jp).

Digital Object Identifier 10.1109/TASLP.2021.3073863

recovers the frequency-domain independent component analysis (FDICA) [3]–[9] based on the Laplace distribution, and the $\ell_{2,1}$ -mixed norm that treats each time segment as the group,

$$\mathcal{P}_n(\mathbf{y}[t, f]) = C \|\mathbf{y}_n[\cdot, \cdot]\|_{2,1} = C \sum_{t=1}^T \left(\sum_{f=1}^F |\mathbf{y}_n[t, f]|^2 \right)^{\frac{1}{2}}, \quad (5)$$

obtains independent vector analysis (IVA) [10]–[13] with the spherical Laplace distribution. The state-of-the-art method called the independent low-rank matrix analysis (ILRMA) [14]–[16] can also be interpreted as Eq. (3) by considering a function that depends on the rank of each spectrogram,

$$\mathcal{P}_n(\mathbf{y}[t, f]) = C \mathcal{D}_R(\mathbf{y}_n[\cdot, \cdot]), \quad (6)$$

where $\mathcal{D}_R(\mathbf{y}_n[\cdot, \cdot])$ is the scalar-valued function that measures low-rankness of a power spectrogram based on the Itakura–Saito non-negative matrix factorization (IS-NMF) [17]:

$$\mathcal{D}_R(\mathbf{y}_n[\cdot, \cdot]) = \min_{\varphi_{f,r}^{[n]} \geq 0, \psi_{r,t}^{[n]} \geq 0} \sum_{t=1}^T \sum_{f=1}^F \left(\frac{|\mathbf{y}_n[t, f]|^2}{\sum_{r=1}^R \varphi_{f,r}^{[n]} \psi_{r,t}^{[n]}} + \log \sum_{r=1}^R \varphi_{f,r}^{[n]} \psi_{r,t}^{[n]} \right), \quad (7)$$

where $\varphi_{f,r}^{[n]}$ and $\psi_{r,t}^{[n]}$ capture the spectral and activation patterns of $|\mathbf{y}_n[t, f]|^2$, respectively, and R represents rank.

The key to success in these BSS methods is to incorporate prior knowledge on the source signals through the penalty function, $\mathcal{P}_n(\cdot)$. IVA imposes co-occurrence among the frequency components in each source signal by a frequency-grouped measure as in Eq. (5). Such grouping of frequency components assists in resolving the permutation problem [18]–[21] associated with the frequency-wise treatment of the demixing matrices. ILRMA takes advantage of a more sophisticated source model to achieve the superior performance. It assumes low-rankness on the power spectrogram of each source so that inter-frequency and inter-frame dependences of a source signal are captured via NMF. Recent advancement of (over)determined BSS relies on even more sophisticated source models to improve the separation performance, e.g., super-/sub-Gaussian ILRMA [22], [23], tensor-factorization-based ILRMA [24], [25], and determined BSS methods based on supervised learning of the source models [26], [27]. Therefore, seeking a better source model is important for developing a novel and effective BSS method.

However, the mainstream algorithms as auxIVA [13] and ILRMA [14] cannot easily handle a new source model because they are specialized to each model. These state-of-the-art algorithms are based on the majorization-minimization (MM) principle [28] that requires upper-bounds approximating the objective function for easier minimization. Since an upper-bound must be specially designed for each objective function, this requirement forces one to derive a new algorithm each time when a source model is newly defined. Derivation of the upper-bound is usually heuristic, and it might take a lot of time before examining the performance with a new source model. One reason for such complication is that those BSS methods are based on the explicit source models (i.e., the source models are explicitly defined as

an objective function). Although a framework based on implicit models has potential of resolving this issue, such framework for determined BSS has rarely been investigated in the literature.

In this paper, to realize effortless investigation of a new source model, we present a general algorithmic framework based on an implicit source model defined via a time-frequency mask. Since the basic principle of the BSS methods is super-Gaussianity, or sparsity, of the source signals in the time-frequency domain, the difference among the determined BSS methods is the way how to impose the sparsity within their separation processes. In this respect, the techniques developed with sparsity-based signal processing [29]–[34], such as the proximal splitting technique [35]–[38], should be beneficial to BSS. By applying one of the proximal algorithms called primal-dual splitting (PDS) algorithm [39], the determined BSS problem in Eq. (3) is handled in a unified manner (Section III). Then, the algorithm is heuristically extended by incorporating a general time-frequency masking method (Section IV). This kind of heuristic extension is called the plug-and-play scheme whose effectiveness has been confirmed in several applications [40]–[44]. The resulted algorithm offers tremendous flexibility into determined BSS because any masking method can be utilized to estimate the demixing matrix, even if the corresponding source model cannot be explicitly written as a formula.

As an application of the general algorithm, we propose a novel BSS method termed *harmonic vector analysis (HVA)*. To model the source signals, HVA focuses on the harmonic structure of audio signals as a cue for separation. By considering sparsity of the cepstrum coefficients, the co-occurrence of the harmonic components is captured. Then, HVA constructs a Wiener-like mask so that the separated signals in each iteration become more exclusive and unmixed. HVA has the properties of both IVA and ILRMA because HVA can consider the spectral pattern of audio signals as ILRMA while it independently treats each time segment as IVA. The experimental results showed that the proposed HVA can achieve the state-of-the-art performance for both speech and music signals.

A. Contribution and Outline

This paper is an extension of the preliminary versions published in the conference proceedings [45], [46]. The contribution of this paper can be summarized as follows:

- unified and detailed presentation of the separately introduced algorithms [45], [46];
- new extensive experiments for investigating parameters and performance of the algorithms;
- proposal of a new BSS method, HVA, with some new ideas for realizing it, including cepstrum thresholding, non-separable masking, and cosine shrinkage operator;
- provision of computational procedures and MATLAB code [47].

The rest of the paper is organized as follows. The technical contents begin with some brief explanation of the proximal algorithm and the proximity operator in Section II. Then, their application to the determined BSS problem in Eq. (3) is presented in Section III, and its heuristic extension based on

time-frequency masking is explained in Section IV. After HVA is proposed in Section V, they are experimentally evaluated in Section VI. Finally, the paper is concluded in Section VII.

II. PRELIMINARIES

A. Primal-Dual Splitting (PDS) Algorithm

In this paper, one of the PDS algorithms is adopted for splitting the first and second terms in Eq. (3). At first, let us briefly summarize it in the usual setting. The PDS algorithm can handle the following general minimization problem:

$$\underset{\mathbf{w}}{\text{Minimize}} \quad g(\mathbf{w}) + h(\mathbf{L}\mathbf{w}), \quad (8)$$

where \mathbf{w} is the vector to be optimized, g and h are proper lower-semicontinuous convex functions, and \mathbf{L} is a bounded linear operator. Here, both g and h can be non-differentiable, and hence a gradient-based optimization method may not be applicable for solving it.

When h is non-differentiable in particular, its composition with \mathbf{L} makes the problem difficult. To handle such difficulty, the associated dual problem is also considered [37]:

$$\underset{\mathbf{y}}{\text{Minimize}} \quad g^*(-\mathbf{L}^*\mathbf{y}) + h^*(\mathbf{y}), \quad (9)$$

where \mathbf{y} is the dual variable, \mathbf{L}^* is the adjoint of \mathbf{L} , and g^* is the Fenchel conjugate of g . Note that this paper avoids explicit consideration of the conjugate by Moreau's identity [37],

$$\text{prox}_{h^*}[\mathbf{z}] = \mathbf{z} - \text{prox}_h[\mathbf{z}], \quad (10)$$

and therefore we leave its details in the reference [37]¹ (the definition of the proximity operator, prox_h , will be given in the next subsection). In the dual problem, h^* is free from the linear operator, \mathbf{L} , whereas g is free from it in the primal problem in Eq. (8). Thus, by simultaneously solving the primal and dual problems in Eqs. (8) and (9), the iterative procedure of the PDS algorithm can circumvent the composition of \mathbf{L} with the objective functions g and h as follows [39]:

$$\begin{cases} \tilde{\mathbf{w}} = \text{prox}_{\mu_1 g}[\mathbf{w}^{[k]} - \mu_1 \mu_2 \mathbf{L}^* \mathbf{y}^{[k]}], \\ \mathbf{z} = \mathbf{y}^{[k]} + \mathbf{L}(2\tilde{\mathbf{w}} - \mathbf{w}^{[k]}), \\ \tilde{\mathbf{y}} = \mathbf{z} - \text{prox}_{\frac{\mu_2}{2} h}[\mathbf{z}], \\ [(\mathbf{w}^{[k+1]}, \mathbf{y}^{[k+1]})^\top] = \alpha(\tilde{\mathbf{w}}, \tilde{\mathbf{y}}) + (1 - \alpha)(\mathbf{w}^{[k]}, \mathbf{y}^{[k]}), \end{cases} \quad (11)$$

where \mathbf{z} is a temporary variable introduced for simpler notation, $\mu_1 > 0$ and $\mu_2 > 0$ are step sizes, and $2 > \alpha > 0$ is a parameter that adjusts the speed of convergence [note that the last line of Eq. (11) can be omitted when $\alpha = 1$ is chosen]. The above iterative procedure enables full splitting of the optimization problem in Eq. (8). That is, the objective functions, g and h , as well as the linear operator, \mathbf{L} , can be calculated independently of each other. Therefore, changing one of them (g , h or \mathbf{L}) only requires modification of the corresponding operator while the others are intact.

¹Although Moreau's identity holds only when the function h is convex, this identity is essential for our heuristic extension to a non-convex function. This is because, by definition, the proximity operator of the Fenchel conjugate of a sparsity-inducing non-convex function is useless for processing.

For guaranteed convergence, the step-size parameters must be chosen to satisfy the following inequality [39]:²

$$\mu_1 \mu_2 \|\mathbf{L}\|_s^2 \leq 1, \quad (12)$$

where $\|\cdot\|_s$ denotes the spectral norm ($\|\mathbf{L}\|_s = \sigma_1(\mathbf{L})$), and $\sigma_1(\mathbf{L})$ is the largest singular value of \mathbf{L} . The parameter α can be arbitrarily chosen from $(0, 2)$, where $\alpha = 1$ is the standard speed, $\alpha > 1$ accelerates, and $\alpha < 1$ slows down the algorithm. Note that the above condition is valid only for a convex problem. The heuristic extension in Section IV will remove the theoretical guarantee of the algorithm, and hence empirical convergence must be experimentally investigated for a general problem. Even so, we will use Eq. (12) to set the parameters because we empirically found its usefulness.

B. Proximity Operator and Thresholding Operator

In the above PDS algorithm, the objective functions, g and h , are minimized via the *proximity operator* [36]:

$$\text{prox}_{\mu g}[\mathbf{z}] = \arg \min_{\boldsymbol{\xi}} \left[g(\boldsymbol{\xi}) + \frac{1}{2\mu} \|\mathbf{z} - \boldsymbol{\xi}\|_2^2 \right], \quad (13)$$

where the left-hand side is regarded as an element of the right-hand side that is singleton for convex g . This subproblem is much easier than the original problem in Eq. (8). Hence, the PDS algorithm splits the original problem into easier subproblems so that the difficulty is alleviated. The proximity operator is particularly useful for handling a non-differentiable function (e.g., a sparsity-inducing function) or a differentiable function whose gradient is not Lipschitz continuous (e.g., $-\log(\cdot)$). Since the determined BSS problem in Eq. (3) consists of such two functions, it seems natural to handle the BSS problem by the proximity operator.

As is well-known, the proximity operators of some sparsity-inducing penalty functions are closely related to the thresholding (or shrinkage) operators. For example, the proximity operator associated with the ℓ_1 norm in Eq. (4) is given by the bin-wise soft-thresholding operator [36],

$$(\text{prox}_{\lambda \|\cdot\|_1}[\mathbf{z}])_n[t, f] = \left(1 - \frac{\lambda}{|z_n[t, f]|} \right)_+ z_n[t, f], \quad (14)$$

where $\lambda \geq 0$ is the thresholding parameter, $(\cdot)_+ = \max\{0, \cdot\}$ is the half-wave rectifier that replaces negative values by zero, and $(\cdot)_n[t, f]$ denotes the (n, t, f) th element of the $N \times T \times F$ array. The proximity operator of the $\ell_{2,1}$ -mixed norm in Eq. (5) is also given by the group-thresholding operator [36],

$$(\text{prox}_{\lambda \|\cdot\|_{2,1}}[\mathbf{z}])_n[t, f] = \left(1 - \frac{\lambda}{(\sum_{f=1}^F |z_n[t, f]|^2)^{\frac{1}{2}}} \right)_+ z_n[t, f]. \quad (15)$$

Proximity operators associated with many other sparsity-inducing functions can also be computed as thresholding operators [29]. While the penalty functions in the above examples are all convex, the proximity operator is also well-defined for some non-convex functions [34], which may be able to induce sparsity more strongly than the convex ones.

²Strictly speaking, some conditions on the problem (such as non-emptiness of the solution set) are necessary for discussing the convergence [37].

III. PDS ALGORITHM FOR DETERMINED BSS

In this section, the PDS algorithm given in Section II-A is applied to the general determined BSS problem in Eq. (3) for obtaining a base algorithm [45].

A. Reformulation and Vectorization of the BSS Problem

To apply the PDS algorithm, the BSS problem is reformulated into the form of Eq. (8). First, to consider the proximity operator, the second term is modified. Since the determinant of a matrix can be expressed in terms of its singular values as $|\det(\mathbf{W}[f])| = \prod_{n=1}^N \sigma_n(\mathbf{W}[f])$, Eq. (3) can be rewritten as

$$\text{Minimize}_{\mathbf{W}[f]} \mathcal{P}(\mathbf{W}[f]\mathbf{x}[t, f]) - \sum_{f=1}^F \sum_{n=1}^N \log \sigma_n(\mathbf{W}[f]), \quad (16)$$

where $\sigma_n(\mathbf{W}[f]) \geq 0$ is the n th singular value of $\mathbf{W}[f]$ in descending order. Note that the penalty function, \mathcal{P} , is also slightly generalized by omitting the summation so that it can be a non-separable function.

Next, the optimization variables are vectorized. All demixing matrices, $(\mathbf{W}[f])_{f=1}^F$, are vectorized and vertically concatenated to construct an NMF -dimensional vector \mathbf{w} :

$$\mathbf{w} = [\mathbf{w}[1]^T, \mathbf{w}[2]^T, \dots, \mathbf{w}[F]^T]^T \in \mathbb{C}^{NMF}, \quad (17)$$

$$\mathbf{w}[f] = \text{vec}(\mathbf{W}[f]) \in \mathbb{C}^{NM}, \quad (18)$$

where vec is the vectorizing operator converting a matrix into the corresponding vector in the row-major numbering scheme,

$$\text{vec}(\mathbf{W}[f]) = [W_{1,1}[f], \dots, W_{1,M}[f], W_{2,1}[f], \dots, W_{N,M}[f]]^T \quad (19)$$

The linear operator that converts the f th part of the vector $\mathbf{w}[f]$ back into the matrix $\mathbf{W}[f]$ is also defined as

$$\text{mat}(\mathbf{w}[f]) = \mathbf{W}[f] \in \mathbb{C}^{N \times M}, \quad (20)$$

which indicates that $\mathbf{w}[f] = \text{vec}(\text{mat}(\mathbf{w}[f]))$. With these notations, Eq. (16) can be expressed as follows:

$$\text{Minimize}_{\mathbf{w}} \mathcal{P}(\mathbf{X}\mathbf{w}) - \sum_{f=1}^F \sum_{n=1}^N \log \sigma_n(\text{mat}(\mathbf{w}[f])), \quad (21)$$

where \mathbf{X} is an $NTF \times NMF$ sparse matrix constructed by copying the observed data, $\mathbf{x}[t, f]$, as

$$\mathbf{X} = \text{blkdiag}(\boldsymbol{\chi}[1], \boldsymbol{\chi}[2], \dots, \boldsymbol{\chi}[F]) \in \mathbb{C}^{NTF \times NMF}, \quad (22)$$

$$\boldsymbol{\chi}[f] = \text{blkdiag}(\chi[f], \chi[f], \dots, \chi[f]) \in \mathbb{C}^{NT \times NM}, \quad (23)$$

$$\chi[f] = [\tau_1[f], \tau_2[f], \dots, \tau_M[f]] \in \mathbb{C}^{T \times M}, \quad (24)$$

$$\tau_m[f] = [x_m[1, f], x_m[2, f], \dots, x_m[T, f]]^T \in \mathbb{C}^T, \quad (25)$$

and blkdiag is the operator constructing a block-diagonal matrix by concatenating inputted matrices diagonally.

Let the second term in Eq. (21) be shortly denoted by \mathcal{I} :

$$\mathcal{I}(\mathbf{w}) = - \sum_{f=1}^F \sum_{n=1}^N \log \sigma_n(\text{mat}(\mathbf{w}[f])). \quad (26)$$

Algorithm 1: PDS-BSS [45].

- 1: **Input:** $\mathbf{X}, \mathbf{w}^{[1]}, \mathbf{y}^{[1]}, \mu_1, \mu_2, \alpha$
 - 2: **Output:** $\mathbf{w}^{[K+1]}$
 - 3: **for** $k = 1, \dots, K$ **do**
 - 4: $\tilde{\mathbf{w}} = \text{prox}_{\mu_1 \mathcal{I}}[\mathbf{w}^{[k]} - \mu_1 \mu_2 \mathbf{X}^H \mathbf{y}^{[k]}]$
 - 5: $\mathbf{z} = \mathbf{y}^{[k]} + \mathbf{X}(2\tilde{\mathbf{w}} - \mathbf{w}^{[k]})$
 - 6: $\tilde{\mathbf{y}} = \mathbf{z} - \text{prox}_{\frac{1}{\mu_2} \mathcal{P}}[\mathbf{z}]$
 - 7: $\mathbf{y}^{[k+1]} = \alpha \tilde{\mathbf{y}} + (1 - \alpha) \mathbf{y}^{[k]}$
 - 8: $\mathbf{w}^{[k+1]} = \alpha \tilde{\mathbf{w}} + (1 - \alpha) \mathbf{w}^{[k]}$
 - 9: **end for**
-

Then, Eq. (21) can be rewritten with a compact notation:

$$\text{Minimize}_{\mathbf{w}} \mathcal{I}(\mathbf{w}) + \mathcal{P}(\mathbf{X}\mathbf{w}). \quad (27)$$

Since this form is the same as Eq. (8), the PDS algorithm in Eq. (11) can be applied at least as a procedure.

B. PDS Algorithm for Determined BSS

Direct application of the PDS algorithm to Eq. (27) obtains Algorithm 1. To realize BSS with this algorithm, two proximity operators, $\text{prox}_{\mathcal{I}}$ and $\text{prox}_{\mathcal{P}}$, must be evaluated.

It is known that the proximity operator of an orthogonally invariant function can be evaluated by applying the corresponding proximity operator to the singular values of the inputted matrix [36]. By regarding $-\log \sigma_n$ in Eq. (26) as $-\log |\sigma_n|$, the proximity operator of $\mathcal{I}(\mathbf{w})$ is obtained [37]:

$$\text{prox}_{\mu \mathcal{I}}[\mathbf{w}] = [(\text{prox}_{\mu \tilde{\mathcal{I}}}[\mathbf{w}[1]])^T, \dots, (\text{prox}_{\mu \tilde{\mathcal{I}}}[\mathbf{w}[F]])^T]^T, \quad (28)$$

where $\text{prox}_{\mu \tilde{\mathcal{I}}}$ is the following proximity operator that moderately increases the singular values of $\text{mat}(\mathbf{w}[f])$,

$$\text{prox}_{\mu \tilde{\mathcal{I}}}[\mathbf{w}[f]] = \text{vec}(\mathbf{U}[f] \tilde{\boldsymbol{\Sigma}}(\text{mat}(\mathbf{w}[f])) \mathbf{V}[f]^H), \quad (29)$$

$\mathbf{W}[f] = \mathbf{U}[f] \boldsymbol{\Sigma}[f] \mathbf{V}[f]^H$ is the singular value decomposition of $\mathbf{W}[f]$ ($= \text{mat}(\mathbf{w}[f])$), $\tilde{\boldsymbol{\Sigma}}(\cdot)$ is the diagonal matrix,

$$\tilde{\boldsymbol{\Sigma}}(\mathbf{W}) = \text{diag}(\text{prox}_{-\mu \log}[\sigma_1(\mathbf{W})], \dots, \text{prox}_{-\mu \log}[\sigma_N(\mathbf{W})]), \quad (30)$$

whose diagonal elements comprise the modified singular values given by applying the proximity operator of $-\mu \log$ [37],

$$\text{prox}_{-\mu \log}[\sigma] = (\sigma + \sqrt{\sigma^2 + 4\mu})/2, \quad (31)$$

and diag is the operator constructing a diagonal matrix from inputted scalars. In other words, applying the proximity operator of $-\mu \log$ to each singular value of $\mathbf{W}[f]$ gives $\text{prox}_{\mu \mathcal{I}}[\cdot]$ as shown in Algorithm 2, where $\text{svd}(\cdot)$ denotes the singular value decomposition. This operation is numerically stable because it does not excessively magnify $\|\mathbf{w}\|_2$ in contrast to the MM algorithms [13], [14] that involve inversion of the matrices, which may lead to instability.

Note that the matrix, \mathbf{X} , is defined only for the formulation and is unnecessary for the implementation [47]. This is because the matrix-vector multiplications, $\mathbf{X}\mathbf{w}$ and $\mathbf{X}^H \mathbf{y}$, can be algorithmically computed as shown in Algorithm 3 and 4, respectively,

Algorithm 2: Computation of $\text{prox}_{\mu_1 \mathcal{I}}[\mathbf{w}]$.

```

1: Input:  $\mathbf{w}, \mu_1$ 
2: Output:  $\tilde{\mathbf{w}}$ 
3: for  $f = 1, \dots, F$  do
4:    $[\mathbf{U}, \Sigma, \mathbf{V}] = \text{svd}(\text{mat}(\mathbf{w}[f]))$ 
5:    $\Sigma_{n,n} = (\Sigma_{n,n} + (\Sigma_{n,n}^2 + 4\mu_1)^{\frac{1}{2}})/2 \quad \forall n$ 
6:    $\tilde{\mathbf{w}}[f] = \text{vec}(\mathbf{U}\Sigma\mathbf{V}^H)$ 
7: end for

```

Algorithm 3: Computation of $\mathbf{X}\mathbf{w}$.

```

1: Input:  $\mathbf{X}, \mathbf{w}$ 
2: Output:  $\hat{\mathbf{s}}$ 
3:  $\hat{\mathbf{s}}[t, f] = \text{mat}(\mathbf{w}[f]) \mathbf{x}[t, f] \quad \forall t, f$ 

```

Algorithm 4: Computation of $\mathbf{X}^H \mathbf{y}$.

```

1: Input:  $\mathbf{X}, \mathbf{y}$ 
2: Output:  $\mathbf{w}'$  [whose  $(n, m, f)$ th element is  $W'_{n,m}[f]$ ]
3:  $W'_{n,m}[f] = \sum_{t=1}^T y_n[t, f] \overline{x_m[t, f]} \quad \forall n, m, f$ 

```

where the overline in Algorithm 4 denotes complex conjugation. Since the matrix, \mathbf{X} , is given by copying the same components as in Eq. (23), avoiding its construction can reduce the required amount of memory.

Algorithm 1 can be applied to many BSS models by only changing $\text{prox}_{\mathcal{P}/\mu_2}[\cdot]$ in the 6th line. For example, an algorithm for FDICA is obtained by inserting the soft-thresholding operator in Eq. (14), while that for IVA is obtained by the group-thresholding operator in Eq. (15). Thus, Algorithm 1 can be used to test performance of BSS models without effort on modifying the code whenever the proximity operator of \mathcal{P} is computable. A source model consisting of two or more penalty functions can also be easily handled by this PDS algorithm (see Section 3.4 and Algorithm 2 of [45] for details).

C. Some Notes on Practical Issues

In this paper, as explained in Section II-A, the inequality condition in Eq. (12) is applied for setting μ_1 and μ_2 . To do so, $\|\mathbf{X}\|_s^2$ must be calculated. This can be easily done by using an iterative algorithm that computes the largest singular value of \mathbf{X} . Since $\|\mathbf{X}\|_s^2 = \|\mathbf{X}^H \mathbf{X}\|_s$, applying an iterative method (e.g., the power method) to $\mathbf{X}^H \mathbf{X}$ instead of \mathbf{X} can reduce the computational cost.

To make the choice of the parameters simpler, the following normalization is considered in this paper:

$$\tilde{\mathbf{X}} = \mathbf{X} / \|\mathbf{X}\|_s. \quad (32)$$

Then, the rule for choosing the step sizes can be simplified as

$$\mu_1 \mu_2 = 1. \quad (33)$$

Therefore, with the above normalization, the number of parameters can be reduced by setting $\mu_2 = 1/\mu_1$. Note that a computationally cheaper norm can be used in place of $\|\mathbf{X}\|_s$

to upper bound the data matrix, $\|\tilde{\mathbf{X}}\|_s^2 \leq 1$. Typical choices are 1-norm, ∞ -norm, and max norm because they can be quickly computed by comparison of the elements.

The whitening of the observed data [2] is strongly recommended for Algorithm 1. This is because $\mathbf{w}^{[k]} - \mu_1 \mu_2 \mathbf{X}^H \mathbf{y}^{[k]}$ in the 4th line updates the demixing filter, \mathbf{w} , in a step-by-step manner like a gradient descent method (see Section IV-C for some intuition). The whitening can act as preconditioning that accelerates the optimization algorithms. In addition, it can normalize the level of the observed signals, and therefore the whitening makes it easier to set a parameter that depends on the scale of the signals, such as λ in Eq. (15). This paper will consider the whitening as default.

As usual, the scales of the separated signals cannot be uniquely determined. To align the frequency-wise scales, post-processing based on the minimal distortion principle, called back projection [48], is used in this paper.

IV. GENERAL TIME-FREQUENCY MASKING AS A HEURISTIC SUBSTITUTE OF THE PROXIMITY OPERATOR

In this section, the proximity operator in Algorithm 1 is substituted by a time-frequency-masking function [46]. This modification enables us to design a variety of BSS methods without derivation of the corresponding algorithms.

A. Generalized Thresholding/Shrinkage Operators

The proximity operators of several sparsity-inducing penalty functions can be computed analytically as in Eqs. (14) and (15). However, this is not the case for many other functions. Although there are some formulas that allow computation of a proximity operator from already known ones, e.g. [49],

$$\text{prox}_{\lambda_1 \|\cdot\|_{2,1} + \lambda_2 \|\cdot\|_1}[\mathbf{z}] = \text{prox}_{\lambda_1 \|\cdot\|_{2,1}}[\text{prox}_{\lambda_2 \|\cdot\|_1}[\mathbf{z}]], \quad (34)$$

applicability of such easy-to-use formula is limited to a specific class of penalty functions [50]–[52]. Thus, one has to run an additional iterative algorithm to compute a proximity operator that cannot be written in a closed form. This can be regarded as a trade-off between flexibility and computational efficiency because types of proximity operators that can be written in closed forms do not have much variety.

To circumvent such trade-off, the generalized thresholding focuses on closed-form (or cheaply computable) operators by directly defining a thresholding/shrinkage operator without defining the corresponding penalty function [53]–[57]. For example, the p -shrinkage operator [54], defined as

$$(\mathcal{T}_p^\lambda[\mathbf{z}])_n[t, f] = \left(1 - \frac{\lambda^{2-p}}{|z_n[t, f]|^{2-p}}\right)_+ z_n[t, f], \quad (35)$$

corresponds to some penalty function that does not have an explicit formula for general p . That is, this shrinkage operator defines an implicit model that induces sparsity.³

³ An element-wise function $(\mathcal{T}[\mathbf{z}])_n = a(|z_n|) \text{sign}(z_n)$ is the proximity operator of some function that may not have closed-form expression. The condition for \mathcal{T} to be a proximity operator is that a is non-decreasing, $a(|z_n|) \rightarrow \infty$ as $|z_n| \rightarrow \infty$, and $0 \leq a(|z_n|) \leq |z_n|$ [55]. This fact motivated us to consider the time-frequency masking in Section IV-B.

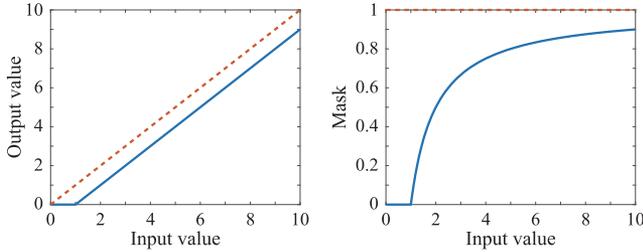


Fig. 1. Soft-thresholding and its mask. The soft-thresholding operator shown on the left (solid line) corresponds to the mask in Eq. (38) shown on the right (solid line). The dashed lines represent the identity operator for a reference.

Another example is one of the social sparsity operators [53],

$$(\mathcal{T}_h^\lambda[\mathbf{z}])_n[t, f] = \left(1 - \frac{\lambda}{(h * |z_n[t, f]|^2)^{\frac{1}{2}}}\right)_+ z_n[t, f], \quad (36)$$

where $h*$ represents the convolution with a two-dimensional filter kernel h , whose elements are non-negative, in the time-frequency domain. Although its effectiveness has been empirically shown [53], this operator is not a proximity operator of some function in general [58]. That is, the social sparsity operator goes beyond the proximity operator and realizes an efficient algorithm with a flexible implicit signal model.

B. Determined BSS Based on Time-Frequency Masking

The soft- and group-thresholding operators [in Eqs. (14) and (15)] and the generalized thresholding/shrinkage operators [in Eqs. (35) and (36)] can be summarized as follows,

$$(\mathcal{T}^\lambda[\mathbf{z}])_n[t, f] = (\mathcal{M}(\mathbf{z}))_n[t, f] z_n[t, f], \quad (37)$$

where $0 \leq (\mathcal{M}(\mathbf{z}))_n[t, f] \leq 1$ is a non-negative scalar that depends on the input, \mathbf{z} . This process can be interpreted as time-frequency masking using a data-dependent mask $\mathcal{M}(\mathbf{z})$. For instance, the soft-thresholding operator in Eq. (14) is the time-frequency masking using the following mask:

$$(\mathcal{M}_{\ell_1}^\lambda(\mathbf{z}))_n[t, f] = \left(1 - \frac{\lambda}{|z_n[t, f]|}\right)_+, \quad (38)$$

which is illustrated on the right side of Fig. 1. Similarly, the group-thresholding operator in Eq. (15) uses

$$(\mathcal{M}_{\ell_{2,1}}^\lambda(\mathbf{z}))_n[t, f] = \left(1 - \frac{\lambda}{(\sum_{f=1}^F |z_n[t, f]|^2)^{\frac{1}{2}}}\right)_+ \quad (39)$$

as the mask. This observation leads us to an idea of substituting a general time-frequency mask into the PDS algorithm so that the mask defines an implicit source model.

By heuristically substituting a time-frequency mask for the proximity operator, Algorithm 1 is extended as Algorithm 5, where \odot denotes the element-wise product, and θ represents a set of parameters for generating the mask. Although stability and convergence of the algorithm with a general mask can be investigated only by experiments, testing several algorithms is easy because the only effort for rewriting the code is in the 6th line. One can just insert a masking method into the algorithm and run it for checking the performance. Any thresholding function and/or sound enhancement method based on time-frequency

Algorithm 5: PDS-BSS-Masking [46].

- 1: **Input:** $\mathbf{X}, \mathbf{w}^{[1]}, \mathbf{y}^{[1]}, \mu_1, \mu_2, \alpha$
 - 2: **Output:** $\mathbf{w}^{[K+1]}$
 - 3: **for** $k = 1, \dots, K$ **do**
 - 4: $\tilde{\mathbf{w}} = \text{prox}_{\mu_1 \mathcal{I}}[\mathbf{w}^{[k]} - \mu_1 \mu_2 \mathbf{X}^H \mathbf{y}^{[k]}]$
 - 5: $\mathbf{z} = \mathbf{y}^{[k]} + \mathbf{X}(2\tilde{\mathbf{w}} - \mathbf{w}^{[k]})$
 - 6: $\tilde{\mathbf{y}} = \mathbf{z} - \mathcal{M}^\theta(\mathbf{z}) \odot \mathbf{z}$
 - 7: $\mathbf{y}^{[k+1]} = \alpha \tilde{\mathbf{y}} + (1 - \alpha) \mathbf{y}^{[k]}$
 - 8: $\mathbf{w}^{[k+1]} = \alpha \tilde{\mathbf{w}} + (1 - \alpha) \mathbf{w}^{[k]}$
 - 9: **end for**
-

masking can collaborate with determined BSS through this algorithm, and hence tremendous flexibility is brought by Algorithm 5.

This heuristic generalization is closely related to the plug-and-play scheme [40]–[42]. By regarding the definition of the proximity operator in Eq. (13) as the negative log-likelihood, it can be viewed as a maximum *a posteriori* (MAP) estimator with a prior distribution $C \exp(-\mathcal{P}(\cdot))$ as follows [40]:

$$\text{prox}_{\lambda \mathcal{P}}[\mathbf{z}] = \arg \max_{\boldsymbol{\xi}} \left[e^{-\frac{1}{2\lambda} \|\mathbf{z} - \boldsymbol{\xi}\|_2^2} e^{-\mathcal{P}(\boldsymbol{\xi})} \right]. \quad (40)$$

This interpretation suggests that substituting a general Gaussian denoiser, that approximately solves Eq. (40), in place of the proximity operator results in an algorithm that works as if the (implicit) function \mathcal{P} is minimized [40]–[42]. When the underlying penalty function is a sum of the penalty functions corresponding to each source ($\mathcal{P} = \sum_{n=1}^N \mathcal{P}_n$), the algorithm can be interpreted as an independence-based BSS method (ML estimation) with $C \exp(-\mathcal{P}_n(\cdot))$ being the density function of the n th source signal [1]. In this sense, Algorithm 5 recasts the BSS problem into the denoising problem in Eq. (40) with the same prior distribution of the sources.

Note that a BSS algorithm beyond the independence-based framework can also be realized with Algorithm 5, at least as a procedure, by inserting a mask that is not separable for each source. The independence assumption leads to the BSS problem in Eq. (3) whose source model is given by a sum of the penalty functions. This is because each source is assumed to be independent from the others, i.e., the source model cannot use information from the other sources to separate a source signal. Then, the corresponding masking functions are also separately applied to each signal without consideration of the others. This is a disadvantage of the independence-based framework because full information on all signals cannot be used. In contrast, Algorithm 5 allows us to use a mask that simultaneously considers all signals. We will propose such a non-separable mask in the next section to obtain HVA.

C. Intuitive Interpretation of the Proposed Algorithm

To obtain a better intuition of the working principle, each line of Algorithm 5 is roughly explained. For convenience of explanation, we start it from the 6th line.

The 6th line calculates difference between the auxiliary variable, \mathbf{z} , and its masked version, $\mathcal{M}^\theta(\mathbf{z}) \odot \mathbf{z}$. Note that the time-frequency masking is applied to spectrograms, and hence

\mathbf{z} is some sort of (vectorized) spectrograms. The difference, $\tilde{\mathbf{y}}$, contains information about how the masking changed the variable, \mathbf{z} . This information is carried to the 4th line via the 7th and 8th lines. Since the 7th and 8th lines are weighted averages with variables in the previous iterate, they are merely controllers of speed of the update. Therefore, the essential information on the difference, $\tilde{\mathbf{y}}$, is not changed by the 7th line and is brought to the 4th line.

The 4th line is the composition of three operations: multiplication of \mathbf{X}^H , subtraction from $\mathbf{w}^{[k]}$, and application of $\text{prox}_{\mu_1 \mathcal{I}}$. The multiplication of \mathbf{X}^H converts the information on masking contained in $\mathbf{y}^{[k]}$ to the domain of the demixing filters, $\mathbf{w}^{[k]}$. Then, the subtraction updates the demixing filters based on that information. Since $\mathbf{y}^{[k]}$ is obtained by the difference between the variables before and after masking, it can be regarded as something similar to gradient that informs the effect of masking. In this sense, the subtraction, $\mathbf{w}^{[k]} - \mu_1 \mu_2 \mathbf{X}^H \mathbf{y}^{[k]}$, updates the demixing filter, $\mathbf{w}^{[k]}$, like the gradient descent method. It is also updated by $\text{prox}_{\mu_1 \mathcal{I}}$ to avoid the undesired result, e.g., $\mathbf{w}^{[k]} = \mathbf{0}$.

The information on updated demixing filter, $\tilde{\mathbf{w}}$, is reflected in the auxiliary variable, \mathbf{z} , by the 5th line. For convenience of explanation, this line is rewritten as $\mathbf{z} = \mathbf{X}\tilde{\mathbf{w}} + [\mathbf{X}\tilde{\mathbf{w}} - (\mathbf{X}\mathbf{w}^{[k]} - \mathbf{y}^{[k]})]$. The first term, $\mathbf{X}\tilde{\mathbf{w}}$, is the result of filtering applied to the observed data, \mathbf{X} , i.e., separated sources. This filtered signal is modified by the addition of the later terms. The subtraction of the last two terms, $\mathbf{X}\mathbf{w}^{[k]} - \mathbf{y}^{[k]}$, is similar to that in the 4th line, $\mathbf{w}^{[k]} - \mu_1 \mu_2 \mathbf{X}^H \mathbf{y}^{[k]}$, but is performed in the domain of the auxiliary variable, \mathbf{y} . By subtracting it from $\mathbf{X}\tilde{\mathbf{w}}$, the effect of the domain difference and $\text{prox}_{\mu_1 \mathcal{I}}$ is obtained as $[\mathbf{X}\tilde{\mathbf{w}} - (\mathbf{X}\mathbf{w}^{[k]} - \mathbf{y}^{[k]})]$. This effect is added to $\mathbf{X}\tilde{\mathbf{w}}$ and handled by the auxiliary variable, \mathbf{z} . Therefore, the input of the masking, \mathbf{z} , consists of not only the separated result for that iteration, $\mathbf{X}\tilde{\mathbf{w}}$, but also the mismatch between the domains. This involved structure allows us to consider the masking separately from the demixing filter update.

D. Relation to the Model-Based IVA

Here, relation between Algorithm 5 and the model-based IVA [59] is discussed. The model-based IVA is an extension of IVA that utilizes a single-channel enhancement method to define an implicit source model. By considering the time-frequency-variant Gaussian distribution as the source model, with variance $v_n[t, f]$, the penalty function corresponding to the model-based IVA can be written as a weighted ℓ_2 norm,

$$\mathcal{P}_n(\mathbf{y}[t, f]) = C \|\mathbf{y}_n[\cdot, \cdot]\|_{2, \mathbf{v}}^2 = C \sum_{t=1}^T \sum_{f=1}^F \frac{|y_n[t, f]|^2}{v_n[t, f]}, \quad (41)$$

which penalizes a time-frequency bin with small $v_n[t, f]$ more than that with large $v_n[t, f]$. This variance is chosen as $v_n[t, f] = |\hat{x}_n[t, f]|^2$, where $\hat{x}_n[t, f]$ is a (roughly) separated signal estimated by some single-channel source enhancement method (e.g., spectral subtraction applied to a mixture signal observed by one of the channels [59]).

As its proximity operator is the shrinkage operator [36],

$$\left(\text{prox}_{\frac{\lambda}{2} \|\cdot\|_{2, \mathbf{v}}}[\mathbf{z}]\right)_n[t, f] = \left(\frac{v_n[t, f]}{v_n[t, f] + \lambda}\right) z_n[t, f], \quad (42)$$

the model-based IVA can also be handled by Algorithm 5 via the mask, $(\mathcal{M}(\mathbf{z}))_n[t, f] = v_n[t, f]/(v_n[t, f] + \lambda)$, which is independent of the inputted variable \mathbf{z} (i.e., constant for every iteration). Although the two methods are related in terms of using a general time-frequency masking method for estimating the demixing matrix, the model-based IVA utilizes the mask only once, before starting iteration, to calculate the weight, $v_n[t, f] = |(\mathcal{M}(\mathbf{x}))_n[t, f] x_n[t, f]|^2$. In contrast, the proposed algorithm uses the mask within the iteration by updating it based on the inputted variable at that time. Therefore, the model-based IVA can be regarded as a special case of the proposed masking-based BSS framework.

V. HARMONIC VECTOR ANALYSIS (HVA): A NOVEL BSS METHOD BASED ON THE HARMONIC STRUCTURE

As an application of Algorithm 5, a BSS method named HVA is proposed in this section. It is based on some ideas novel for determined BSS, including cepstrum thresholding, non-separable masking, and cosine shrinkage operator.

A. Harmonic Structure of Audio Signals

In HVA, the harmonic structure is considered as the basis of the mask generation. As an illustrative example of the harmonic structure, a log-amplitude spectrum of a voiced segment of a speech signal is shown on the left side of Fig. 2. The periodic repetition of the peaks and dips is called harmonic structure and is typical of real-world audio signals. That is, in a short-time segment of a typical source signal, multiple peaks (or harmonic components) simultaneously occur. This co-occurrence of the harmonic components should be useful for resolving the permutation problem and separating the source signals because a prominent peak can inform the positions of the other peaks.

To incorporate this prior knowledge into determined BSS based on Algorithm 5, a time-frequency masking method should be designed so that the harmonic components are enhanced compared to the other components. In HVA, this is realized by two ideas that are new to determined BSS: cepstrum thresholding and a Wiener-like mask.

B. Cepstrum Thresholding Enhancing the Harmonic Structure

One of the well-accepted concepts related to the harmonic structure is *cepstrum*. When a log-amplitude spectrum exhibits the harmonic structure, it can be well-approximated by a few Fourier-series coefficients because of the periodic repetition. To capture such property, cepstrum is defined as the Fourier transform of a log-amplitude spectrum.⁴ By denoting the element-wise absolute value as $(\text{abs}(\mathbf{z}))_n[t, f] = |z_n[t, f]|$, cepstrum of a multi-channel signal, $\text{cep}(\mathbf{z})$, for all time segments and channels can be written as

$$(\text{cep}(\mathbf{z}))_n[t, c] = (\mathcal{F}_f(\log(\text{abs}(\mathbf{z}))))_n[t, c], \quad (43)$$

⁴In the literature, cepstrum may be defined by the *inverse* Fourier transform of a log-amplitude spectrum. Such difference is not important for HVA since the cepstrum thresholding does not depend on the phase difference.

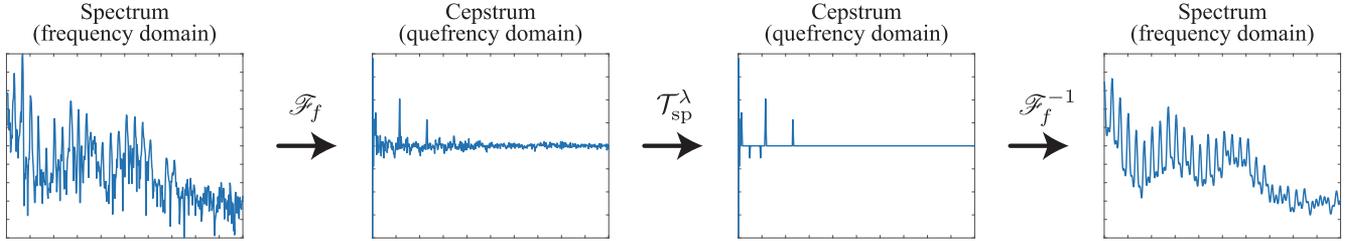


Fig. 2. Typical example of a voiced speech signal and its enhancement by the cepstrum (hard-)thresholding. The log-amplitude spectrum of a segment of voiced speech (leftmost) is converted to cepstrum (second from the left) by the Fourier transform. By thresholding the cepstrum coefficients (second from the right) and taking the inverse Fourier transform, the enhanced version of the log-amplitude spectrum is obtained (rightmost) as described in Section V-B.

where $\log(\cdot)$ is the element-wise logarithmic function, \mathcal{F}_f is the (normalized) frequency-directional Fourier transform,

$$(\mathcal{F}_f(\mathbf{z}))_n[t, c] = \frac{1}{F} \sum_{f=1}^F z_n[t, f] e^{-2\pi i(c-1)(f-1)/C}, \quad (44)$$

and $c = 1, \dots, C$ is the index of quefrequency. Note that zero-padding can be used to make $C > F$, which might improve the performance because of higher redundancy.

By introducing a Fourier thresholding operator $\mathcal{T}_{\mathcal{F}}^\lambda$ as

$$\mathcal{T}_{\mathcal{F}}^\lambda(\mathbf{z}) = \mathcal{F}_f^{-1}(\mathcal{T}_{sp}^\lambda(\mathcal{F}_f(\mathbf{z}))), \quad (45)$$

we define the *cepstrum thresholding* that applies a sparsity-inducing operator in the cepstrum domain:

$$\mathcal{T}_{cep}^\lambda(\mathbf{z}) = \exp(\mathcal{T}_{\mathcal{F}}^\lambda(\log(\text{abs}(\mathbf{z}))), \quad (46)$$

where $\exp(\cdot)$ is the element-wise exponential function, \mathcal{F}_f^{-1} is the frequency-directional inverse Fourier transform,

$$(\mathcal{F}_f^{-1}(\mathbf{z}))_n[t, f] = \frac{F}{C} \sum_{c=1}^C z_n[t, c] e^{2\pi i(c-1)(f-1)/C}, \quad (47)$$

and \mathcal{T}_{sp}^λ is a sparsity-promoting thresholding operator, e.g., the soft-thresholding operator in Eq. (14) or the p -shrinkage operator in Eq. (35).

The effect of the cepstrum thresholding is shown on the right side of Fig. 2, where the hard-thresholding was chosen as the cepstrum-domain thresholding operator, \mathcal{T}_{sp}^λ , for clear demonstration. The cepstrum of the log-amplitude spectrum contains some larger peaks that correspond to the harmonic components. The other small cepstrum coefficients correspond to the non-structured details. By removing small cepstrum coefficients and retaining larger coefficients using hard-thresholding, the harmonic structure is enhanced as in the rightmost figure. For mixture signals, the cepstrum thresholding enhances the dominant signal having harmonic structure and attenuates the other components. We expect that, for each channel, such attenuation principally occurs to the interference signals that have less energy than the main signal.

C. Non-Separable Masking for Source Separation

Based on a set of enhanced signals, a mask is constructed and applied in Algorithm 5. Ordinarily, the independence criterion has forced a BSS algorithm to be a procedure separable for each

source signal. In contrast, it is also possible to define a non-separable BSS method that simultaneously considers all source signals to extract separation cues. Algorithm 5 can realize such method by using a non-separable mask-generating function, for example, the Wiener-like mask [60]:

$$(\mathcal{M}_{WL}(\hat{\mathbf{x}}))_n[t, f] = \left(\frac{|\hat{x}_n[t, f]|^2}{\sum_{n=1}^N |\hat{x}_n[t, f]|^2} \right)^\gamma, \quad (48)$$

which takes values between 0 and 1, where $\hat{x}_n[\cdot, \cdot]$ is the enhanced spectrogram corresponding to the n th source signal, and $\gamma > 0$ is a parameter adjusting the level of attenuation. Note that, when $\gamma = 1$, this mask can be viewed as a non-separable version of the mask in Eq. (42), which is related to the time-frequency-variant Gaussian model, by replacing the constant λ in the denominator of Eq. (42) with the sum of the other source signals, $\sum_{l \neq n} |\hat{x}_l[t, f]|^2$.

This non-separable mask is more effective for promoting source separation than ordinary separable masks because it simultaneously uses information on all signals and encourages each bin to be more exclusive and unmixed.

D. Harmonic Vector Analysis (HVA)

We propose HVA by defining a specific mask that uses the harmonic structure for enhancing the source signals. It is a combination of the cepstrum thresholding and Wiener-like masking introduced in the previous subsections:

$$(\mathcal{M}_{HVA}^{\lambda, \kappa}(\mathbf{z}))_n[t, f] = \left(\frac{v_n^{\mathbf{z}, \lambda, \kappa, \varepsilon}[t, f]}{\sum_{n=1}^N v_n^{\mathbf{z}, \lambda, \kappa, \varepsilon}[t, f]} \right)^\gamma, \quad (49)$$

where $v_n^{\mathbf{z}, \lambda, \kappa, \varepsilon}[\cdot, \cdot]$ is a squared amplitude spectrogram whose harmonic structure is enhanced by the cepstrum thresholding [corresponding to $|\hat{x}_n[\cdot, \cdot]|^2$ in Eq. (48)].

The cepstrum thresholding in Eq. (46) is composition of \exp , $\mathcal{T}_{\mathcal{F}}^\lambda$, \log and abs . Therefore, $v_n^{\mathbf{z}, \lambda, \kappa, \varepsilon}[\cdot, \cdot]$ in Eq. (49) is explained in this order. By applying the exponential function to a thresholded log-amplitude spectrogram, $\varrho_n^{\mathbf{z}, \lambda, \kappa, \varepsilon}[\cdot, \cdot]$, and squaring it, $v_n^{\mathbf{z}, \lambda, \kappa, \varepsilon}[\cdot, \cdot]$ is obtained as follows:

$$v_n^{\mathbf{z}, \lambda, \kappa, \varepsilon}[t, f] = \exp(2\varrho_n^{\mathbf{z}, \lambda, \kappa, \varepsilon}[t, f]), \quad (50)$$

where 2 comes from the squaring in Eq. (48). The thresholded log-amplitude spectrogram, $\varrho_n^{\mathbf{z}, \lambda, \kappa, \varepsilon}[\cdot, \cdot]$, is given by Fourier

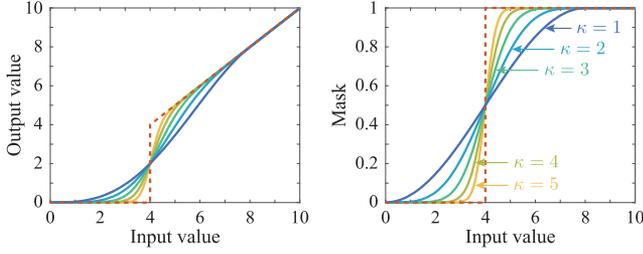


Fig. 3. Cosine shrinkage and its mask for $\lambda = 4$ and $\kappa \in \{1, 2, 3, 4, 5\}$. The cosine shrinkage operator in Eq. (54) (solid line) is shown on the left with the hard-thresholding operator (dashed line). The corresponding masks (the composition of raised cosine functions drawn by solid lines, and the step function drawn by dashed line) are shown on the right.

thresholding, $\mathcal{T}_{\mathcal{F}}^{\lambda, \kappa}(\cdot) = \mathcal{F}_f^{-1}(\mathcal{T}_{\text{sp}}^{\lambda, \kappa}(\mathcal{F}_f(\cdot)))$, applied to mean-subtracted log-amplitude spectrograms, $\rho^{\mathbf{z}, \varepsilon}$, as follows:

$$\varrho_n^{\mathbf{z}, \lambda, \kappa, \varepsilon}[t, f] = (\mathcal{T}_{\mathcal{F}}^{\lambda, \kappa}(\rho^{\mathbf{z}, \varepsilon}))_n[t, f] + \mu_n^{\mathbf{z}, \varepsilon}[t], \quad (51)$$

where $\rho^{\mathbf{z}, \varepsilon}$ is obtained by subtracting time-dependent mean,

$$\rho_n^{\mathbf{z}, \varepsilon}[t, f] = \log(|z_n[t, f]| + \varepsilon) - \mu_n^{\mathbf{z}, \varepsilon}[t], \quad (52)$$

$\mu_n^{\mathbf{z}, \varepsilon}[t]$ is the mean value of the log-amplitude spectrum,

$$\mu_n^{\mathbf{z}, \varepsilon}[t] = \frac{1}{F} \sum_{f=1}^F \log(|z_n[t, f]| + \varepsilon), \quad (53)$$

and $\varepsilon > 0$ is a small constant for preventing $\rho_n^{\mathbf{z}, \varepsilon}[t, f]$ to be $-\infty$. Mean subtraction is performed in Eq. (52) to make the cepstrum coefficients sparser. It also has another benefit that adding the mean value after the thresholding as in Eq. (51) can restore the level of log-amplitude. Therefore, the Wiener-like mask can enhance the level difference of the signals even when λ is exceedingly large and the thresholder eliminates all cepstrum coefficients.

For the cepstrum-domain operator, $\mathcal{T}_{\text{sp}}^{\lambda, \kappa}$, included in $\mathcal{T}_{\mathcal{F}}^{\lambda, \kappa}$, any sparsity-promoting operator can be adopted. In this paper, we newly propose the following operator, which is named *cosine shrinkage operator*, for the enhancement:

$$(\mathcal{T}_{\cos}^{\lambda, \kappa}(\mathbf{z}))_n[t, c] = \Xi^{\kappa}[(|z_n[t, c]|/(2\lambda))_{\overline{1}}] z_n[t, c], \quad (54)$$

where Ξ^{κ} is κ -times composition of Ξ (i.e., $\Xi^{\kappa} = \Xi \circ \dots \circ \Xi$),

$$\Xi[z_n[t, c]] = (1 - \cos(\pi z_n[t, c]))/2, \quad (55)$$

and $(\cdot)_{\overline{1}} = \min\{1, \cdot\}$. For intuitive explanation, it is illustrated in Fig. 3 with the corresponding mask, $\Xi^{\kappa}[(|z_n[t, c]|/(2\lambda))_{\overline{1}}]$. The mask (on the right side of Fig. 3) consists of the half period of (raised) cosine function, where λ is its inflection point. This can be viewed as a smooth approximation of the mask corresponding to the hard-thresholding operator (red dashed line), and the degree of approximation is controlled by κ . Hence, $\mathcal{T}_{\cos}^{\lambda, \kappa}$ is a smooth approximation of the hard-thresholding operator as on the left side of Fig. 3. The reasons why this shrinkage operator is adopted in HVA are as follows: (1) it has no bias for large coefficients similar to hard-thresholding; (2) we found that smoothness is important for stable separation; and (3) HVA does not require to force small coefficients to be exactly zero owing

Algorithm 6: Harmonic Vector Analysis (HVA).

- 1: **Input:** $\mathbf{X}, \mathbf{w}^{[1]}, \mathbf{y}^{[1]}, \mu_1, \mu_2, \alpha$
 - 2: **Output:** $\mathbf{w}^{[K+1]}$
 - 3: **for** $k = 1, \dots, K$ **do**
 - 4: $\tilde{\mathbf{w}} = \text{prox}_{\mu_1 \mathcal{I}}[\mathbf{w}^{[k]} - \mu_1 \mu_2 \mathbf{X}^H \mathbf{y}^{[k]}]$
 - 5: $\mathbf{z} = \mathbf{y}^{[k]} + \mathbf{X}(2\tilde{\mathbf{w}} - \mathbf{w}^{[k]})$
 - 6: $\tilde{\mathbf{y}} = \mathbf{z} - \mathcal{M}_{\text{HVA}}^{\lambda, \kappa}(\mathbf{z}) \odot \mathbf{z}$
 - 7: $\mathbf{y}^{[k+1]} = \alpha \tilde{\mathbf{y}} + (1 - \alpha) \mathbf{y}^{[k]}$
 - 8: $\mathbf{w}^{[k+1]} = \alpha \tilde{\mathbf{w}} + (1 - \alpha) \mathbf{w}^{[k]}$
 - 9: **end for**
-

Algorithm 7: Computation of $\mathcal{M}_{\text{HVA}}^{\lambda, \kappa}(\mathbf{z})$.

- 1: **Input:** $\mathbf{z}, \lambda, \kappa$
 - 2: **Output:** $\mathcal{M}_{\text{HVA}}^{\lambda, \kappa}(\mathbf{z})$
 - 3: $\zeta = \log(\text{abs}(\mathbf{z}) + \varepsilon)$
 - 4: $\mu_n[t] = (1/F) \sum_{f=1}^F \zeta_n[t, f] \quad \forall n, t$
 - 5: $\rho_n[t, f] = \zeta_n[t, f] - \mu_n[t] \quad \forall n, t, f$
 - 6: $\nu = \mathcal{F}_f(\rho)$
 - 7: $\varsigma_n[t, c] = \min\{1, |\nu_n[t, c]|/(2\lambda)\} \quad \forall n, t, c$
 - 8: **for** $k = 1, \dots, \kappa$ **do**
 - 9: $\varsigma_n[t, c] = (1 - \cos(\pi \varsigma_n[t, c]))/2 \quad \forall n, t, c$
 - 10: **end for**
 - 11: $\xi = \mathcal{F}_f^{-1}(\varsigma \odot \nu)$
 - 12: $\varrho_n[t, f] = \xi_n[t, f] + \mu_n[t] \quad \forall n, t, f$
 - 13: $v_n[t, f] = \exp(2\varrho_n[t, f]) \quad \forall n, t, f$
 - 14: $(\mathcal{M}_{\text{HVA}}^{\lambda, \kappa}(\mathbf{z}))_n[t, f] = (v_n[t, f] / \sum_{n=1}^N v_n[t, f])^{\gamma} \quad \forall n, t, f$
-

to the Wiener-like mask. Note again that any thresholding-like function can be used in place of this shrinkage operator. The performance of HVA depends on its choice, and some other thresholding/shrinkage operator that performs better for HVA than the cosine shrinkage operator should exist.

The proposed algorithm for HVA is shown in Algorithm 6.⁵ For assisting implementation, the mask-generating function, $\mathcal{M}_{\text{HVA}}^{\lambda, \kappa}(\cdot)$ in the 6th line, is also summarized in Algorithm 7. The subtraction and addition of the mean value, $\mu_n^{\mathbf{z}, \varepsilon}[t]$, in the 5th and 12th lines, respectively, maintain the energy of the squared amplitude spectrograms, $v_n^{\mathbf{z}, \lambda, \kappa, \varepsilon}[\cdot, \cdot]$, similar to that of the input spectrogram, $|z_n[\cdot, \cdot]|^2$. The (frequency-directional) Fourier transform, \mathcal{F}_f in the 6th line, converts the log-amplitude spectrum into cepstrum, and its inverse, \mathcal{F}_f^{-1} in the 11th line, does the opposite. The mask of the cosine shrinkage operator (the right figure of Fig. 3) is denoted by ς and is applied to the cepstrum coefficients, ν , as $\varsigma \odot \nu$ in the 11th line. The exponential in the 13th line cancels the logarithm in the 4th line, and the 14th line computes the Wiener-like mask. Note that $\varsigma_n[t, c] = 1$ when $\lambda = 0$. That is, $v_n[t, f] = |z_n[t, f]|^2$ when $\lambda = 0$ (and $\varepsilon = 0$).

To reduce the number of parameters, ε is fixed and omitted from $\mathcal{M}_{\text{HVA}}^{\lambda, \kappa}$ because, according to our preliminary investigation, its effect to the performance is not notable (ε is set to 10^{-3} in the

⁵Our MATLAB implementation is available at [47].

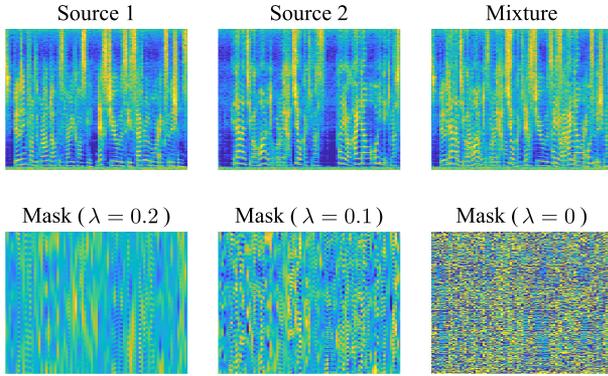


Fig. 4. Whitened spectrograms and masks of HVA. The upper row shows a log-amplitude spectrogram of a speech mixture (right) together with its source signals (left and center). The color range is 60 dB. Due to the frequency-wise whitening, the signal-to-noise ratio seems different for each frequency. The bottom row shows the masks of HVA (with different λ s) calculated from the (top right) mixture. The color range is set to $[0.55, 0.85]$ for enhancing the visibility (note that $\gamma = 1/N$ makes the average around 0.7, and therefore 0.7 ± 0.15 was chosen for the range). These masks correspond to the 1st iteration of HVA (see Fig. 5 for similar masks with color range $[0, 1]$).

rest of the paper). We also heuristically fix γ to $1/N$ in this paper based on the following reason. When $v_n^{z, \lambda, \kappa, \epsilon}[t, f]$ is the same for all n , the value inside the parentheses of Eq. (49) is $1/N$, which depends on N . By setting $\gamma = 1/N$, the value of the mask for that case becomes $(1/N)^{(1/N)} \approx 0.7$ that is approximately independent of N . Therefore, to avoid decrease in the average value of the mask, $\gamma = 1/N$ is chosen tentatively.⁶ The other two parameters, λ and κ , should be chosen based on the distribution of the cepstrum coefficients of the observed signals, which will be investigated in the experimental section.

E. Role of the Cepstrum Thresholding in HVA

For demonstration of the mask of HVA, some examples are shown in the bottom row of Fig. 4. Each figure illustrates one of the 2-channel signals/masks, and the corresponding masks for the other channel are not shown here.

The bottom-right figure shows the mask without cepstrum thresholding (i.e., $\lambda = 0$). Note that the non-separable mask itself can promote separation if, for each time-frequency bin, level difference between the channels exists, because the mask retains louder components and attenuates smaller components. Therefore, the mask in the bottom-right figure is not totally random but exhibits some structure. However, this mask cannot solve the permutation problem because each frequency is treated independently.

⁶We empirically found that the algorithm becomes unstable when the average value of the mask is small. This should be because the PDS algorithm is built upon the proximity operator that has restriction on the amount of change of the signal. To reduce the amount of change caused by the masking, the values of the mask should be close to 1. Therefore, we chose $(1/N)^{(1/N)} \approx 0.7$ that is relatively close to 1. Although we chose $(\cdot)^\gamma$ for the definition of the Wiener-like mask in Eq. (48) because it seems popular in the literature of time-frequency masking, any function that controls the distribution of the value of the mask can be used in place of $(\cdot)^\gamma$. Investigation of such function that performs better than $(\cdot)^\gamma$ can be a part of future works.

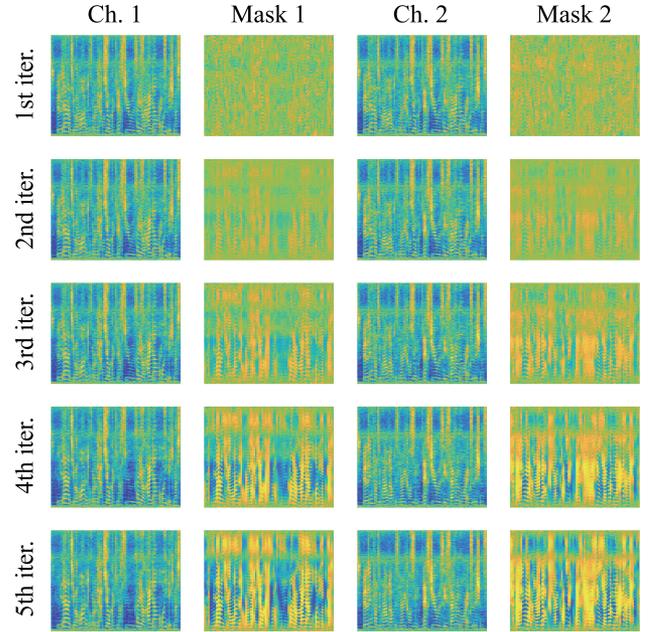


Fig. 5. Whitened spectrograms and masks for the first 5 iterations of HVA. The parameters of HVA were set to $\alpha = \mu_1 = \mu_2 = 1$, $\lambda = 0.08$, and $\kappa = 3$. The signal is Mixture A in Fig. 6. Therefore, these spectrograms and masks correspond to the top central figures in Figs. 9 and 10. The color range for the whitened spectrograms is 60 dB, and that for the masks is $[0, 1]$.

The cepstrum thresholding assists the non-separable mask by enhancing the dominant periodic pattern corresponding to the harmonic structure. Since the cosine shrinkage operator attenuates cepstrum coefficients that are small relative to the parameter λ , a larger λ gives a simpler mask that can be well-described by fewer sinusoidal patterns. These examples show that the cepstrum thresholding is not intended to separate some components but just enhancing the harmonic structure. Note that subtraction of the mean value, $\mu_n^{z, \epsilon}[t]$, in the logarithmic domain normalizes the time-segment-wise scale of the input spectrogram because $\log(\text{abs}(c\mathbf{z})) = \log(\text{abs}(\mathbf{z})) + \log(c)$ for any positive constant $c > 0$. Therefore, the shrinkage parameter, λ , can be chosen without a care of the scale of the observed signals.

For further demonstration, the mask of HVA for each iteration is shown in Fig. 5. This figure is visualization of the first 5 iterations of the experimental result ($\alpha = \mu_1 = \mu_2 = 1$, Mixture A) in Section VI-B. At the 1st iteration, the masks seem somewhat random because the cepstrum thresholding was performed for each time segment independently. Then, the update of the demixing filter collects the information of the mask for all time segments via $\mathbf{X}^H \mathbf{y}$ (see Algorithm 4). By using the updated demixing filter, each signal was enhanced, which made the cepstrum thresholding able to adapt to the spectral patterns better in the later iterations. As can be seen in the figure, the mask rapidly captured the spectral patterns in the first few iterations.

Since the cepstrum thresholding simultaneously processes all frequency components, the mask exhibited vertical patterns. That is, the effect of the demixing filter at some frequency propagates vertically to all the other frequencies. Such masks that

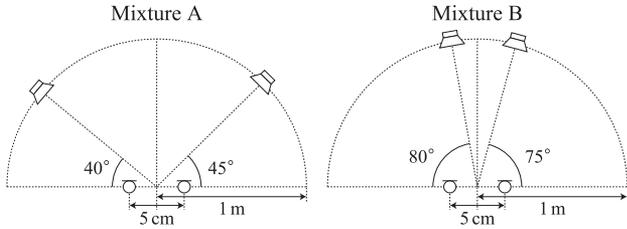


Fig. 6. Recording conditions of the 2-channel speech mixtures utilized in the experiments in Sections VI-A and VI-B using the Laplace IVA.

simultaneously handle all frequencies can solve the permutation problem to some extent. Note that, since this paper considers the time-invariant model in Eq. (1), permutation across the time does not occur. If the cepstrum thresholding is used in a time-varying situation, the permutation across the time should be treated by some additional technique.

VI. EXPERIMENTS

In this paper, we presented the general BSS algorithm and its specific application termed HVA. To show the properties of both the algorithm and HVA, some experiments are conducted in this section. At first, the properties of the algorithm and HVA are qualitatively shown using two 2-channel speech mixtures as examples. Then, the performances of HVA over speech and music mixtures in 2- and 3-channel conditions are compared with IVA and ILRMA quantitatively.⁷

The performance was measured by the standard metrics: the source-to-distortion ratio (SDR), source-to-interferences ratio (SIR), and sources-to-artifacts ratio (SAR) [62]. For all trials, the initial value of the demixing matrices $\mathbf{w}^{[1]}$ was set to the identity matrices ($\mathbf{W}[f] = \mathbf{I}$ for all f), and that of \mathbf{y} was the zero vector. The sampling rate of the signals was 16 kHz. The window length was set according to the previous studies for easier comparison.⁸ The whitening [2] and back projection [48] were applied as pre- and post-processes, respectively.

A. Illustration of Basic Properties of the PDS Algorithm

The PDS algorithms contain three parameters μ_1 , μ_2 , and α . At first, their effects to the performance over iteration are presented using the well-understood IVA. Two pairs of female speech signals recorded as in Fig. 6 were downloaded from SiSEC database [64] (liverec of dev1 in the underdetermined audio source separation task), where the reverberation time was 130 ms. The half-overlapping 2048-point-long Hann window (128 ms) was used for the short-time Fourier transform (STFT). The BSS method tested here was IVA based on the

⁷An audio example for Section VI-C is available at [61].

⁸The window length determines the degree of freedom of the demixing filter. Therefore, the ideal separation performance is higher when the window length is longer. However, a longer window results in more optimization variables, which makes the optimization more difficult. Moreover, the window length determines the appearance of the spectrogram, which makes the characteristics of the source model different. Because of these factors, the relation between the window length and separation performance is very complicated as indicated in [63]. In this paper, we decided to follow the previous studies for the window length so that such complication is avoided.

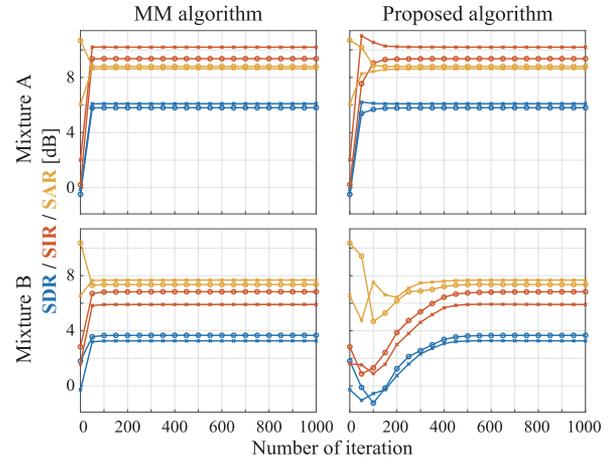


Fig. 7. SDR/SIR/SAR of the signals separated from the mixtures (recorded as in Fig. 6) using Laplace IVA (Section VI-A1). The two sources in each mixture are distinguished by the markers (o and x). The left and right figures were obtained by auxIVA [13] and the proposed algorithm (Algorithm 1), respectively. The parameters were set to $\mu_1 = \mu_2 = 1$ and $\alpha = 1.75$.

spherical Laplace distribution, $\mathcal{P}_n = \|\cdot\|_{2,1}$ in Eq. (5), whose proximity operator is given in Eq. (15). Note that the mixture signals contain ambient noise of the room.

1) *Appropriateness of the PDS Algorithm:* Before showing the effects of parameters, the PDS algorithm was compared with the MM algorithm (auxIVA [13]), based on the iterative projection technique [8], to confirm that the Laplace IVA was appropriately realized by the PDS algorithm. Their performances over iteration are shown in Fig. 7, where the parameters were set to $\mu_1 = \mu_2 = 1$, and $\alpha = 1.75$. As in the figure, both algorithms resulted in the same scores, which indicates that the PDS algorithm was properly working, but the PDS algorithm required more iterations than the MM algorithm. In particular, Mixture B needed significantly more iterations, which should be because it was obtained with a condition more difficult than Mixture A, as in Fig. 6. Note that the computation per iteration of IVA by the PDS algorithm was 1.3 times faster than IVA by the MM algorithm (PDS: 26.7 ms, MM: 35.0 ms) which should be because the MM algorithm calculates a lot of matrix inversions within an iteration.

To see the appropriateness in terms of minimization, the values of the objective function for the PDS algorithm are shown in Fig. 8 with 3 different axes. They confirm that the proposed algorithm properly reduced the objective function in Eq. (3) with $\mathcal{P}_n = \|\cdot\|_{2,1}$ in Eq. (5). Note that, in general, this kind of plot cannot be drawn for the proposed masking-based algorithm because an explicit form of the objective function may not exist.

2) *Effect of the Relaxation Parameter α :* As mentioned in Section II-A, the parameter α can speed up ($2 > \alpha > 1$) or slow down ($1 > \alpha > 0$) the convergence of the algorithm. To illustrate such effect, the performances for different $\alpha \in \{0.5, 1, 1.5\}$ are shown in Fig. 9, where the other parameters were set to $\mu_1 = \mu_2 = 1$. As expected, higher α achieved the scores at the final iteration with less number of iterations (note that the case for $\alpha = 1.75$ is shown in Fig. 7). For the Laplace IVA, the parameter α acted as a stretching factor of the horizontal axis.

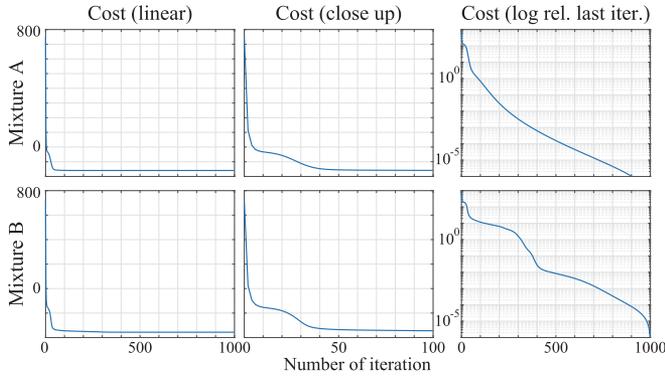


Fig. 8. Values of the objective function. With the penalty function in Eq. (5), the objective function is given in Eq. (3). The setting was the same as that of Fig. 7. The first 100 iterations of the left figure are closed up in the central figure. The right figure shows the same curve in the logarithmic scale with some vertical shift that made the value at the 1000th iteration 0.

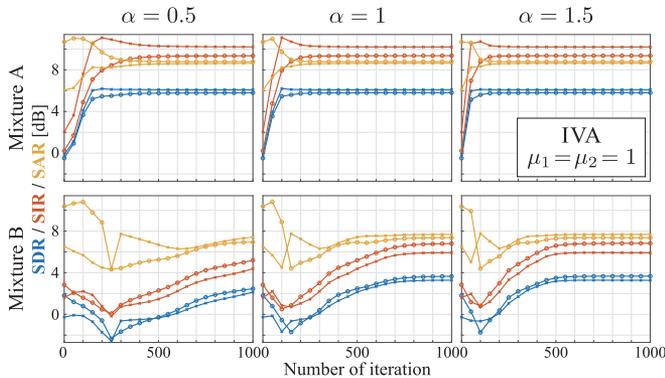


Fig. 9. SDR/SIR/SAR of the signals separated from the mixtures in Fig. 6, where the parameter α was varied (Section VI-A2).

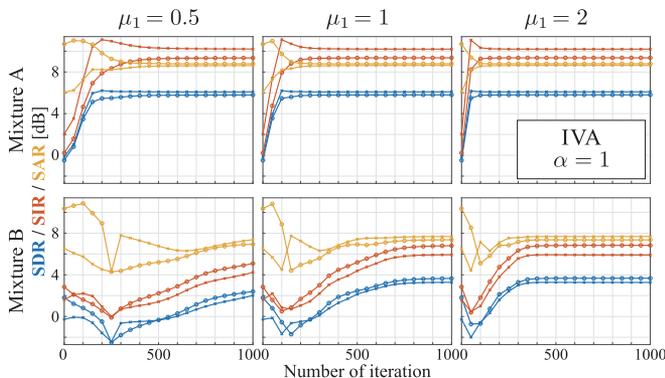


Fig. 10. SDR/SIR/SAR of the signals separated from the mixtures in Fig. 6, where the parameter μ_1 was varied, and $\mu_2 = 1/\mu_1$ (Section VI-A3).

3) *Effect of the Step-Size Parameters μ_1 and μ_2* : As discussed in Section III-C, the data normalization in Eq. (32) allows the choice $\mu_2 = 1/\mu_1$ for the step size, which comes from Eq. (12). Since μ_1 and μ_2 balance the effects of proximity operators ($\text{prox}_{\mu_1 \mathcal{I}}$ and $\text{prox}_{\frac{1}{\mu_2} \mathcal{P}}$ in Algorithm 1), their choice can also affect the convergence. By setting $\mu_2 = 1/\mu_1$ and $\alpha = 1$, the performances of the PDS algorithm were investigated for $\mu_1 \in \{0.5, 1, 2\}$ as illustrated in Fig. 10. From Figs. 9 and 10, a specific choice of the parameters (μ_1, μ_2 and α) seems to

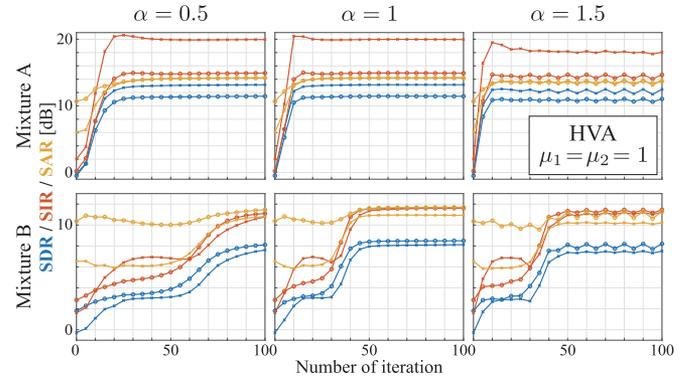


Fig. 11. SDR/SIR/SAR of the signals separated from the mixtures in Fig. 6, where the parameter α was varied (Section VI-B1).

have little impact on the separation performance for the Laplace IVA if the number of iterations is sufficiently large. In contrast, when the number of iterations is limited, these parameters may have some impact on the performance.

B. Illustration of Basic Properties of HVA

As opposed to the Laplace IVA investigated in the previous subsection, HVA does not rely on a theoretical foundation but is heuristically defined by the time-frequency-masking function in Eq. (49). In such cases, theoretically developed criteria like Eq. (12) do not guarantee the convergence because their assumption is not satisfied anymore. Therefore, the performance of HVA as well as its dependency on the algorithmic parameters must be investigated by experiments. Here, the effects of step-size and relaxation parameters (μ_1, μ_2 and α) are qualitatively presented. The experimental conditions are the same as those in the previous subsection. The parameters of the cepstrum thresholding were set to $\lambda = 0.08$ and $\kappa = 3$ based on the experimental results that will be presented in the next subsection.

1) *Effect of the Relaxation Parameter α* : At first, the effect of the relaxation parameter, α , on the performance of HVA was investigated by fixing $\mu_1 = \mu_2 = 1$. The performances for $\alpha \in \{0.5, 1, 1.5\}$ are shown in Fig. 11 (note that the range of vertical and horizontal axes are greatly different from those in Figs. 9 and 10). Since the mask of HVA might not be stable compared to the proximity operator of a convex function, choice of α affected the performance at the final iteration. For HVA, $\alpha = 1$ seems a reasonable choice because it resulted in stable and fast improvement. If a masking function is more unstable, a smaller α should be preferable for stabilizing the performance.

2) *Effect of the Step-Size Parameters μ_1 and μ_2* : The effect of the step-size parameters μ_1 and μ_2 is also presented by fixing $\alpha = 1$ as in Fig. 12. From the results, it can be seen that the effect of the choice of the step sizes on HVA is small. This should be because the masking function of HVA in Eq. (49) is independent of μ_2 . In general, such independence is not a favorable property of a masking function because the balance of the algorithm can collapse. Even so, HVA was able to stably perform separation.

3) *Choice of the Parameters of HVA*: Based on the above experimental results, we suggest to choose the parameters of HVA as $\mu_1 = \mu_2 = \alpha = 1$. Since HVA is not so sensitive to μ_1

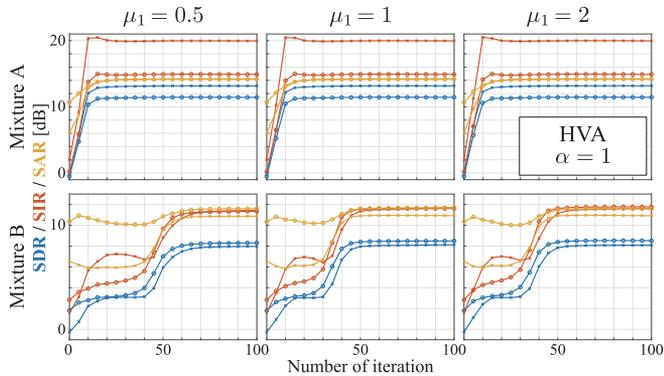


Fig. 12. SDR/SIR/SAR of the signals separated from the mixtures in Fig. 6, where the parameter μ_1 was varied, and $\mu_2 = 1/\mu_1$ (Section VI-B2).

and μ_2 , we suggest setting them to 1 so that their multiplication can be omitted in the actual computation. Also, the 7th and 8th lines of Algorithm 6 can be omitted in the actual computation when $\alpha = 1$.

The other parameters of HVA, λ and κ , will be investigated in the next subsection. Overall, $\lambda = 0.08$ seems a good choice, and $\kappa = 3$ seems slightly better than $\kappa = 1$ or 2.

C. Performance Evaluation of HVA Using Speech Signals

For evaluating the performance of the proposed HVA, it is compared with the standard method, IVA, and the state-of-the-art method, ILRMA. Here, we performed experiments using speech mixtures by improving the experiments in [14].

1) *Experimental Conditions*: The database utilized in this experiment was a part of SiSEC (the underdetermined audio source separation task) [64]. The BSS methods were evaluated using 2- and 3-channel speech mixtures. For the 2-channel mixtures, 20 files (male3, male4, female3, and male4) contained in dev1 and dev2 were utilized. They include live recordings (liverec containing ambient noise) with the reverberation time 130 ms/250 ms and the microphone spacing 1 m/5 cm. The distances between the sources and the center of the microphone array is 1 m. For each situation, all possible pairs of the signals (out of 3 or 4) were selected to make the task determined ($N = M = 2$). As the result, 96 pairs for 20 situations were generated.⁹

Similarly, for the 3-channel mixtures, 8 files in dev3 were utilized. They include female/male speech with the reverberation time 130 ms/380 ms and the microphone spacing 50 cm/5 cm. The distances between the sources and the center of the microphone array is 1 m. By selecting all possible 3-tuples from 4 signals to make the task determined ($N = M = 3$), 32 mixtures for 8 situations were generated.

The half-overlapping 4096-point-long (256 ms) Hann window was used for STFT as in [14]. All algorithms were iterated 200 times. The number of bases of ILRMA for each source was

⁹Note that, although [14] utilized the same dataset, its experiment only contained 12 mixtures for 12 situations because the first two speech sources (out of 3) were selected for each situation. In contrast, this paper utilized all signals included in each situation by selecting all possible pairs of the signals.

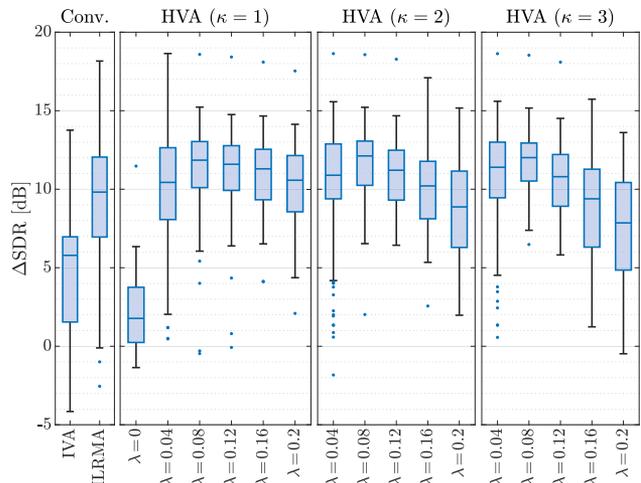


Fig. 13. SDR improvements of the speech signals separated from 2-channel mixtures. The experimental settings are explained in Section VI-C1, where $\alpha = \mu_1 = \mu_2 = 1$ and the number of iterations was 200. The central lines indicate the median, and the bottom and top edges of the box indicate the 25th and 75th percentiles, respectively.

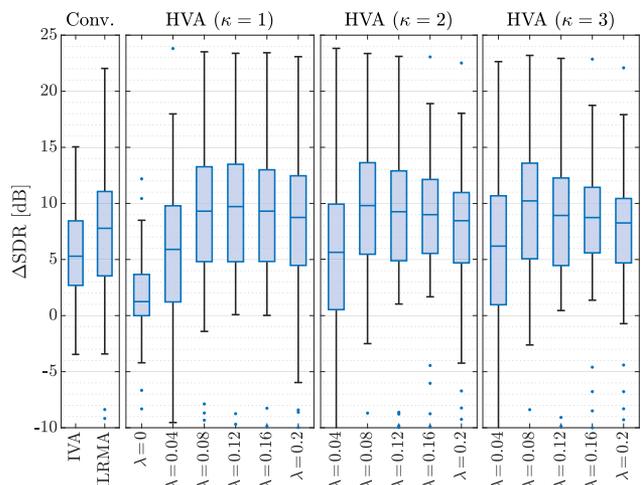


Fig. 14. SDR improvements of the speech signals separated from 3-channel mixtures. The experimental settings are explained in Section VI-C1, where $\alpha = \mu_1 = \mu_2 = 1$ and the number of iterations was 200.

set to 2, which is suitable for speech signals as shown in [14]. The parameters of HVA were set to $\alpha = \mu_1 = \mu_2 = 1$, $\kappa \in \{1, 2, 3\}$, and $\lambda \in \{0.04, 0.08, 0.12, 0.16, 0.2\}$. HVA without cepstrum thresholding ($\lambda = 0$) was also tested.

2) *Results*: The experimental results for the 2- and 3-channel cases are summarized in Figs. 13 and 14, respectively. First of all, HVA without cepstrum thresholding ($\lambda = 0$) could not perform separation. As discussed in Section V-E, this is because the permutation problem cannot be solved only by the non-separable mask. This result indicates the importance of the cepstrum thresholding that simultaneously handles all frequency components. From the figures, it can be seen that κ has less impact than λ . On average, $(\lambda, \kappa) = (0.08, 3)$ seems to perform well, and therefore we will focus on HVA with $\lambda = 0.08$ and $\kappa = 3$.

TABLE I
MUSIC SIGNALS FOR THE 2-CHANNEL MIXTURES [64]

Song name	Source (1/2)
bearlin-roads	acoustic_guit_main/vocals
another_dreamer-the_ones_we_love	guitar/vocals
fort_minor-remember_the_name	violins_synth/vocals
ultimate_nz_tour	guitar/synth
tamy-que_pena_tanto_faz	guitar/vocals

TABLE II
MUSIC SIGNALS FOR THE 3-CHANNEL MIXTURES [64]

Song name	Source (1/2/3)
bearlin-roads	acoustic_guit_main/bass/vocals
another_dreamer-the_ones_we_love	drums/guitar/vocals
fort_minor-remember_the_name	drums/violins_synth/vocals
ultimate_nz_tour	guitar/synth/vocals

The figures show that the proposed HVA outperformed IVA. This result indicates that the harmonic structure can be a useful cue for the separation in determined BSS. Compared to ILRMA, HVA achieved performance similar to ILRMA for the 3-channel case but outperformed it for the 2-channel case. While ILRMA utilizes repetition of the spectral pattern with time as a cue for separation, HVA only focuses on the spectral pattern at each time segment independently, as illustrated in Fig. 4. Since spectral patterns of speech signals widely vary with time, the low-rank structure (or repetitive pattern) of the magnitude spectrogram, assumed in ILRMA, may not effectively serve as a separation cue in this case. In contrast, HVA is not hindered by such variation of signals because HVA considers time-independent information like IVA. An audio example for each method can be found in [61].

D. Performance Evaluation of HVA Using Music Signals

Here, the proposed HVA was tested using music mixtures by following the experiments in [14].

1) *Experimental Conditions*: This experiment also used a part of SiSEC (the professionally produced music recordings) [64]. The combinations of the source signals utilized in the 2- and 3-channel experiments are listed in Tables I and II, respectively. Since *tamy-que_pena_tanto_faz* comprises only two sources (guitar and vocal), it was not included in the 3-channel case. The 2- and 3-channel mixtures were produced by convolving the impulse response E2A or JR2, included in the RWCP database [65], with each source. The recording conditions of these impulse responses are shown in Fig. 15.

As in [14], the 3/4-overlapping 8192-point-long Hann window (512 ms) was used for STFT, and the number of bases of ILRMA for each source was set to 30. The other settings of the algorithmic parameters were the same as those in the previous experiment using speech signals.

2) *Results*: The experimental results for the 2- and 3-channel cases are summarized in Figs. 16 and 17, respectively. From the figures, it can be seen that the proposed HVA was comparable to the other methods. Since the 2-channel mixtures comprise harmonic signals as the sources (see Table I), HVA should have been able to effectively model the harmonic structure through cepstrum. Therefore, even though ILRMA's ability of modeling

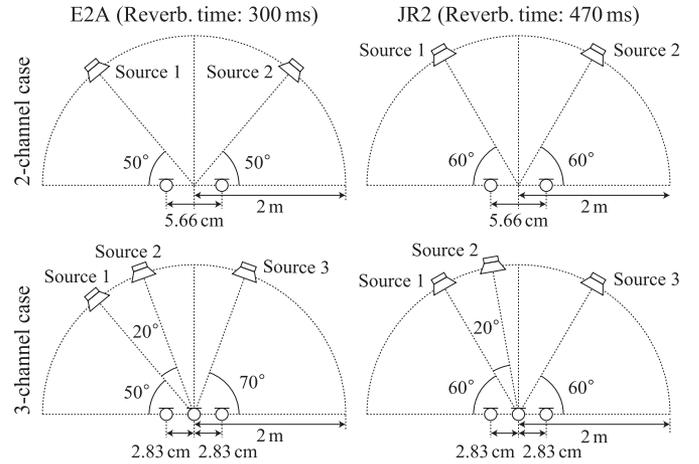


Fig. 15. Recording conditions of the impulse responses (E2A and JR2 [65]) utilized in the experiment in Section VI-D1.

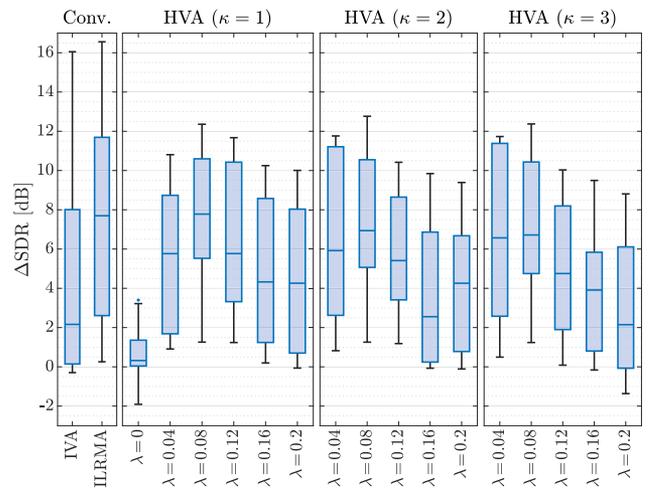


Fig. 16. SDR improvements of the music signals separated from 2-channel mixtures. The experimental settings are explained in Section VI-D1, where $\alpha = \mu_1 = \mu_2 = 1$ and the number of iterations was 200.

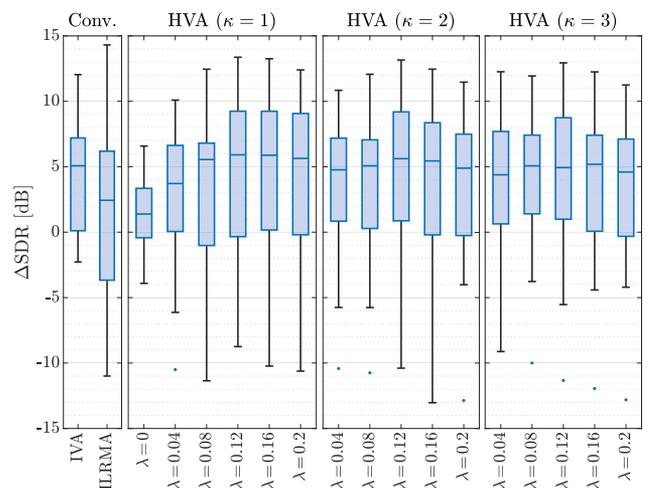


Fig. 17. SDR improvements of the music signals separated from 3-channel mixtures. The experimental settings are explained in Section VI-D1, where $\alpha = \mu_1 = \mu_2 = 1$ and the number of iterations was 200.

TABLE III
COMPUTATIONAL TIME PER ITERATION (W/ CORE I7-8700)

	Speech (Section VI-C)	Music (Section VI-D)
auxIVA	47.6 ms	106.0 ms
ILRMA	116.1 ms	271.2 ms
HVA (Proposed)	60.8 ms	164.3 ms

repetitive spectral patterns is advantageous for music signals, HVA with $\lambda = 0.08$ was able to perform similarly.

For the 3-channel case, IVA performed better than ILRMA. One reason for this result should be the presence of drums in the sources (see Table II). ILRMA can handle such percussive sources by devoting some of the bases, but HVA does not have a mechanism for explicitly handle percussive sources. Even so, HVA was able to perform comparably in this case, too. Note that, as the masking-based BSS framework can simultaneously utilize multiple criteria by combining multiple masks, HVA has a potential of improvement by incorporating other time-frequency masks targeting at the specific structure of the source signals as examined in [66].

E. Computational Efficiency

The computational time of the algorithms depend on the settings such as the window length and the number of channels. Table III shows the computational time per iteration for 2-channel speech and music separation (Section VI-C and VI-D, respectively) as an example. Here, the computational time for the speech and music cases were different because the numbers of time segments and frequencies were different. The computational time of HVA was more than that of IVA because HVA involves additional computation of logarithm, the (frequency-directional) Fourier transform pair, cosine, and exponential. ILRMA required more time for computation because it treats the demixing filters as well as all time-frequency bins as the optimization variables. As in Table III, the proposed HVA tends to be more efficient than ILRMA but less efficient than IVA.

Note that Table III shows computational time *per iteration*, and the total time depends on the number of iterations. The required number of iterations depends on the situations. For example, as shown in Figs. 11 and 12, HVA seems to require 20 and 50 iterations for Mixture A and B, respectively, with the setting explained in Section VI-A. Since HVA is efficient than ILRMA for each iteration and requires relatively small number of iterations (see Fig. 5 for an example of the rapid evolution of the mask of HVA), the computational time of HVA is usually less than ILRMA.

F. Limitations of HVA

Here, some limitations of HVA are discussed. First, since the mask-generating function of HVA targets only harmonic signals, HVA cannot directly handle signals that do not exhibit the harmonic structure (e.g., white noise). The mask must be modified to enhance the non-harmonic signals in that case. Second, the assumed model in HVA (also in IVA and ILRMA) [Eq. (1)] does not consider additional noise and/or time variation

of the mixing system. To handle such cases, the algorithm and mask must be modified accordingly. Third, the mask of HVA cannot be used as a post filter as in [59] because the mask is not intended to separate signals.

Although HVA has these limitations, some of them can be easily resolved because defining another mask-generating function is painless. Note that the ideas behind the mask of HVA can be solely used in other signal processing methods that target harmonic signals. Such extensions and applications of the proposed method and ideas are left as the future works.

VII. CONCLUSIONS

In this paper, the novel BSS method termed HVA was proposed. By modeling the harmonic structure via cepstrum analysis, HVA achieved the performance comparable to the state-of-the-art method, ILRMA, with less computational effort. To realize HVA, the general BSS algorithm based on time-frequency masking was presented. Since it allows any mask for enhancing the source signals, improving HVA as well as investigating a completely new BSS method should be easy. The future works include an extension of HVA by data-adaptation (instead of using the fixed masking function), online extension of the algorithm using adaptive techniques [67], and investigation of combination of the masking-based algorithm and the existing source enhancement techniques.

REFERENCES

- [1] A. Hyvärinen and E. Oja, "Independent component analysis: Algorithms and applications," *Neural Netw.*, vol. 13, no. 4, pp. 411–430, 2000.
- [2] A. Hyvärinen, J. Karhunen, and E. Oja, "Independent component analysis, ser. adaptive and cognitive dynamic systems: Signal processing, learning," *Commun. Control*. Wiley, 2004.
- [3] P. Smaragdis, "Blind separation of convolved mixtures in the frequency domain," *Neurocomputing*, vol. 22, no. 1, pp. 21–34, 1998.
- [4] S. Araki, R. Mukai, S. Makino, T. Nishikawa, and H. Saruwatari, "The fundamental limitation of frequency domain blind source separation for convolutive mixtures of speech," *IEEE Trans. Speech Audio Process.*, vol. 11, no. 2, pp. 109–116, Mar. 2003.
- [5] H. Sawada, R. Mukai, S. Araki, and S. Makino, "Convolutional blind source separation for more than two sources in the frequency domain," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, May 2004, vol. 3, pp. 885–888.
- [6] H. Buchner, R. Aichner, and W. Kellermann, "A generalization of blind source separation algorithms for convolutive mixtures based on second-order statistics," *IEEE Trans. Speech Audio Process.*, vol. 13, no. 1, pp. 120–134, Jan. 2005.
- [7] H. Saruwatari, T. Kawamura, T. Nishikawa, A. Lee, and K. Shikano, "Blind source separation based on a fast-convergence algorithm combining ICA and beamforming," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 14, no. 2, pp. 666–678, Mar. 2006.
- [8] N. Ono and S. Miyabe, "Auxiliary-function-based independent component analysis for super-gaussian sources," in *Proc. Int. Conf. Latent Variable Anal. Signal Separation*, 2010, pp. 165–172.
- [9] K. Yatabe, "Consistent ICA: Determined BSS meets spectrogram consistency," *IEEE Signal Process. Lett.*, vol. 27, pp. 870–874, 2020, doi: [10.1109/LSP.2020.2996904](https://doi.org/10.1109/LSP.2020.2996904).
- [10] A. Hiroe, "Solution of permutation problem in frequency domain ICA, using multivariate probability density functions," in *Proc. Int. Conf. Independent Compon. Anal. Blind Source Separation*, 2006, pp. 601–608.
- [11] T. Kim, T. Eltoft, and T.-W. Lee, "Independent vector analysis: An extension of ica to multivariate components," in *Proc. Int. Conf. Independent Compon. Anal. Blind Source Separation*, 2006, pp. 165–172.
- [12] T. Kim, H. T. Attias, S. Y. Lee, and T. W. Lee, "Blind source separation exploiting higher-order frequency dependencies," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 1, pp. 70–79, Jan. 2007.

- [13] N. Ono, "Stable and fast update rules for independent vector analysis based on auxiliary function technique," in *Proc. IEEE Workshop Appl. Signal Process. Audio Acoust.*, Oct. 2011, pp. 189–192.
- [14] D. Kitamura, N. Ono, H. Sawada, H. Kameoka, and H. Saruwatari, "Determined blind source separation unifying independent vector analysis and nonnegative matrix factorization," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 24, no. 9, pp. 1626–1641, Sep. 2016.
- [15] D. Kitamura, N. Ono, H. Sawada, H. Kameoka, and H. Saruwatari, "Efficient multichannel nonnegative matrix factorization exploiting rank-1 spatial model," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Apr. 2015, pp. 276–280.
- [16] D. Kitamura, N. Ono, H. Sawada, H. Kameoka, and H. Saruwatari, "Relaxation of rank-1 spatial constraint in overdetermined blind source separation," in *Proc. Eur. Signal Process. Conf.*, Aug. 2015, pp. 1261–1265.
- [17] C. Févotte, N. Bertin, and J.-L. Durrieu, "Nonnegative matrix factorization with the itakura-saito divergence. with application to music analysis," *Neural Comput.*, vol. 21, no. 3, pp. 793–830, 2009.
- [18] S. Kurita, H. Saruwatari, S. Kajita, K. Takeda, and F. Itakura, "Evaluation of blind signal separation method using directivity pattern under reverberant conditions," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2000, vol. 5, pp. 3140–3143.
- [19] N. Murata, S. Ikeda, and A. Ziehe, "An approach to blind source separation based on temporal structure of speech signals," *Neurocomputing*, vol. 41, no. 1, pp. 1–24, 2001.
- [20] H. Sawada, R. Mukai, S. Araki, and S. Makino, "A robust and precise method for solving the permutation problem of frequency-domain blind source separation," *IEEE Trans. Speech Audio Process.*, vol. 12, no. 5, pp. 530–538, Sep. 2004.
- [21] H. Sawada, S. Araki, and S. Makino, "Measuring dependence of bin-wise separated signals for permutation alignment in frequency-domain BSS," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2007, pp. 3247–3250.
- [22] D. Kitamura *et al.*, "Generalized independent low-rank matrix analysis using heavy-tailed distributions for blind source separation," *EURASIP J. Adv. Signal Process.*, vol. 2018, no. 1, pp. 1–28, May 2018.
- [23] S. Mogami *et al.*, "Independent low-rank matrix analysis based on time-variant sub-Gaussian source model for determined blind source separation," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 28, pp. 503–518, 2020, doi: [10.1109/TASLP.2019.2959257](https://doi.org/10.1109/TASLP.2019.2959257).
- [24] K. Yoshii, K. Kitamura, Y. Bando, E. Nakamura, and T. Kawahara, "Independent low-rank tensor analysis for audio source separation," in *Proc. Eur. Signal Process. Conf.*, 2018, pp. 1657–1661.
- [25] R. Ikeshita, "Independent positive semidefinite tensor analysis in blind source separation," in *Proc. Eur. Signal Process. Conf.*, 2018, pp. 1652–1656.
- [26] N. Makishima *et al.*, "Independent deeply learned matrix analysis for determined audio source separation," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 27, no. 10, pp. 1601–1615, Oct. 2019.
- [27] H. Kameoka, L. Li, S. Inoue, and S. Makino, "Supervised determined source separation with multichannel variational autoencoder," *Neural Comput.*, vol. 31, no. 9, pp. 1891–1914, 2019.
- [28] K. Lange, *MM Optimization Algorithms*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2016.
- [29] M. Kowalski, "Sparse regression using mixed norms," *Appl. Comput. Harm. Anal.*, vol. 27, no. 3, pp. 303–324, 2009.
- [30] F. Bach, R. Jenatton, J. Mairal, and G. Obozinski, "Optimization with sparsity-inducing penalties," *Found. Trends Mach. Learn.*, vol. 4, no. 1, pp. 1–106, 2012.
- [31] D. Malioutov and A. Aravkin, "Iterative log thresholding," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, May 2014, pp. 7198–7202.
- [32] R. Chartrand and W. Yin, *Nonconvex Sparse Regularization and Splitting Algorithms*. Cham, Switzerland: Springer International Publishing, 2016, pp. 237–249.
- [33] J. Woodworth and R. Chartrand, "Compressed sensing recovery via nonconvex shrinkage penalties," *Inverse Problem*, vol. 32, no. 7, 2016, Art. no. 075004.
- [34] I. Bayram and S. Bulek, "A penalty function promoting sparsity within and across groups," *IEEE Trans. Signal Process.*, vol. 65, no. 16, pp. 4238–4251, Aug. 2017.
- [35] P. L. Combettes and J.-C. Pesquet, *Proximal Splitting Methods in Signal Processing*. New York, NY, USA: Springer, 2011, pp. 185–212.
- [36] N. Parikh and S. Boyd, "Proximal algorithms," *Found. Trends Optim.*, vol. 1, no. 3, pp. 127–239, 2014.
- [37] H. H. Bauschke and P. L. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Cham, Switzerland: Springer, 2017.
- [38] M. Burger, A. Sawatzky, and G. Steidl, *First Order Algorithms in Variational Image Processing*. Cham, Switzerland: Springer, 2016, pp. 345–407.
- [39] N. Komodakis and J. C. Pesquet, "Playing with duality: An overview of recent primal-dual approaches for solving large-scale optimization problems," *IEEE Signal Process. Mag.*, vol. 32, no. 6, pp. 31–54, Nov. 2015.
- [40] S. V. Venkatakrishnan, C. A. Bouman, and B. Wohlberg, "Plug-and-play priors for model based reconstruction," in *Proc. IEEE Global Conf. Signal Inf. Process.*, Dec 2013, pp. 945–948.
- [41] S. H. Chan, X. Wang, and O. A. Elgendy, "Plug-and-play ADMM for image restoration: Fixed-point convergence and applications," *IEEE Trans. Comput. Imag.*, vol. 3, no. 1, pp. 84–98, Mar. 2017.
- [42] S. Ono, "Primal-dual plug-and-play image restoration," *IEEE Signal Process. Lett.*, vol. 24, no. 8, pp. 1108–1112, Aug. 2017.
- [43] Y. Masuyama, K. Yatabe, Y. Koizumi, Y. Oikawa, and N. Harada, "Deep griffin-lim iteration," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2019, pp. 61–65.
- [44] Y. Masuyama, K. Yatabe, Y. Koizumi, Y. Oikawa and N. Harada, "Deep griffin-lim iteration: Trainable iterative phase reconstruction using neural network," *IEEE J. Sel. Topics Signal Process.*, vol. 15, no. 1, pp. 37–50, Jan. 2021.
- [45] K. Yatabe and D. Kitamura, "Determined blind source separation via proximal splitting algorithm," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Apr. 2018, pp. 776–780.
- [46] K. Yatabe and D. Kitamura, "Time-frequency-masking-based determined BSS with application to sparse IVA," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, May 2019, pp. 715–719.
- [47] K. Yatabe, "MATLAB Code for Harmonic Vector Analysis (HVA)," Jan. 2021. Online. [Available]: <https://doi.org/10.24433/CO.9507820.v1>.
- [48] K. Matsuoka, "Minimal distortion principle for blind source separation," in *Proc. SICE Annu. Conf.*, Aug. 2002, vol. 4, pp. 2138–2143.
- [49] A. Gramfort, D. Strohmeier, J. Hauelsen, M. Hamalainen, and M. Kowalski, "Time-frequency mixed-norm estimates: Sparse m/eeg imaging with non-stationary source activations," *NeuroImage*, vol. 70, pp. 410–422, 2013.
- [50] P. Combettes and J. Pesquet, "Proximal thresholding algorithm for minimization over orthonormal bases," *SIAM J. Optim.*, vol. 18, no. 4, pp. 1351–1376, 2008.
- [51] Y.-L. Yu, "On decomposing the proximal map," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 91–99.
- [52] N. Pustelnik and L. Condat, "Proximity operator of a sum of functions; application to depth map estimation," *IEEE Signal Process. Lett.*, vol. 24, no. 12, pp. 1827–1831, Dec. 2017.
- [53] M. Kowalski, K. Siedenburg, and M. Dorfler, "Social sparsity! neighborhood systems enrich structured shrinkage operators," *IEEE Trans. Signal Process.*, vol. 61, no. 10, pp. 2498–2511, May 2013.
- [54] R. Chartrand, "Shrinkage mappings and their induced penalty functions," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, May 2014, pp. 1026–1029.
- [55] M. Kowalski, "Thresholding RULES and iterative shrinkage/thresholding algorithm: A convergence study," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 2014, pp. 4151–4155.
- [56] I. W. Selesnick and I. Bayram, "Sparse signal estimation by maximally sparse convex optimization," *IEEE Trans. Signal Process.*, vol. 62, no. 5, pp. 1078–1092, Mar. 2014.
- [57] I. Bayram, "Penalty functions derived from monotone mappings," *IEEE Signal Process. Lett.*, vol. 22, no. 3, pp. 265–269, Mar. 2015.
- [58] R. Gribonval and M. Nikolova, "A characterization of proximity operators," *J. Math. Imag. Vis.*, vol. 62, no. 6, pp. 773–789, Jul 2020.
- [59] A. R. López, N. Ono, U. Remes, K. Palomäki, and M. Kurimo, "Designing multichannel source separation based on single-channel source separation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Apr. 2015, pp. 469–473.
- [60] H. Erdogan, J. R. Hershey, S. Watanabe, and J. Le Roux, "Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Apr. 2015, pp. 708–712.
- [61] D. Kitamura, "Audio Example of 2-Channel Speech Separation Using IVA, ILRMA, and HVA," 2021. Online. [Available]: <http://d-kitamura.net/demo-HVA%5Fen.html>.
- [62] E. Vincent, R. Gribonval, and C. Févotte, "Performance measurement in blind audio source separation," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 14, no. 4, pp. 1462–1469, Jul. 2006.

- [63] D. Kitamura and K. Yatabe, "Consistent independent low-rank matrix analysis for determined blind source separation," *EURASIP J. Adv. Signal Process.*, vol. 2020, no. 1, pp. 1–46, Nov. 2020.
- [64] S. Araki *et al.*, "The 2011 signal separation evaluation campaign (SiSEC2011)—audio source separation," in *Proc. Int. Conf. Latent Variable Anal. Signal Separation*, 2012, pp. 414–422.
- [65] S. Nakamura, K. Hiyane, F. Asano, T. Nishiura, and T. Yamada, "Acoustical sound database in real environments for sound scene understanding and hands-free speech recognition," in *Proc. Int. Conf. Lang. Resour. Eval.*, 2000, pp. 965–968.
- [66] S. Oyabu, D. Kitamura, and K. Yatabe, "Linear multichannel blind source separation based on time-frequency mask obtained by harmonic/percussive sound separation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2021.
- [67] S. Ono, M. Yamagishi, and I. Yamada, "A sparse system identification by using adaptively-weighted total variation via a primal-dual splitting approach," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2013, pp. 6029–6033.



Daichi Kitamura (Member, IEEE) received the Ph.D. degree from SOKENDAI, Hayama, Japan. He joined The University of Tokyo in 2017 as a Research Associate, and he moved to National Institute of Technology, Kagawa College as an Assistant Professor in 2018. His research interests include audio source separation, statistical signal processing, and machine learning. He was the recipient of the Awaya Prize Young Researcher Award from The Acoustical Society of Japan (ASJ) in 2015, Ikushi Prize from Japan Society for the Promotion of Science in 2017, Itakura Prize Innovative Young Researcher Award from ASJ in 2018, and Young Author Best Paper Award from IEEE Signal Processing Society in 2019.



Kohei Yatabe (Member, IEEE) received the B.E., M.E., and Ph.D. degrees from Waseda University in 2012, 2014, and 2017, respectively. He is currently an Assistant Professor of the Department of Intermedia Art and Science, Waseda University. His research interests include optical measurement of airborne sound.