Signal Processing 93 (2013) 3515–3527

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

A Khatri–Rao subspace approach to blind identification of mixtures of quasi-stationary sources ☆

Ka-Kit Lee [a], Wing-Kin Ma [a,*], Xiao Fu [a], Tsung-Han Chan [b], Chong-Yung Chi [b]

[a] The Chinese University of Hong Kong, Shatin, N.T., Hong Kong Special Administrative Region
[b] National Tsing Hua University, Hsingchu, Taiwan

ARTICLE INFO

Article history:
Received 4 August 2012
Received in revised form
20 March 2013
Accepted 25 March 2013
Available online 5 April 2013

Keywords:
Khatri–Rao subspace
Quasi-stationary signals
Blind identification

ABSTRACT

Blind identification (BID) of mixtures of quasi-stationary sources (QSS) is a vital approach for blind speech or audio source separation, and has attracted much interest for more than a decade. In general, BID-QSS is formulated, and then treated, under either the parallel factor analysis or joint diagonalization framework. This paper describes a Khatri–Rao (KR) subspace formulation of BID-QSS. Like subspace techniques founded in sensor array processing, the KR subspace formulation enables us to decompose the BID problem into a per-source decoupled BID problem. By exploring this new opportunity, we derive an overdetermined BID algorithm that solves BID-QSS in a successive and algebraically simple manner. Analysis shows that under an ideal data setting, the decoupled solutions of the proposed overdetermined BID algorithm yield very fast convergence. We also tackle the underdetermined case by proposing a two-stage strategy where the decoupled solutions are used to warm-start another BID algorithm. Simulation results show that the proposed BID algorithms yield competitive mean-square error and runtime performance in comparison to the state-of-the-arts in BID-QSS.

1. Introduction

In this paper, our interest lies in blind identification (BID), or blind source separation (BSS), of a linear instantaneous mixture of quasi-stationary sources (QSSs). This problem is important, both fundamentally and practically. In terms of applications, the major driving force for the investigation of BID-QSS is blind speech or audio source separation in microphone arrays [2–5]. The idea of BID-QSS is to utilize the statistically time-varying characteristics of QSSs to identify the unknown system mixing matrix. Roughly speaking, BID-QSS amounts to a problem where, given a set of data matrices $\mathbf{R}_1, \ldots, \mathbf{R}_M \in \mathbb{C}^{N\times N}$, we are required to find a matrix $\mathbf{A} \in \mathbb{C}^{N\times K}$ and a set of diagonal matrices $\mathbf{D}_1, \ldots, \mathbf{D}_M \in \mathbb{C}^{K\times K}$ such that

$$\mathbf{R}_m = \mathbf{A}\mathbf{D}_m\mathbf{A}^H, \quad m = 1, \ldots, M. \tag{1}$$

Or, alternatively, we seek to find an appropriate approximation of (1) through certain formulations.

The problem arising in BID-QSS, or (1), has attracted much interest in the signal processing community. There are two major frameworks for the problem. One is to pose (1) as a three-way tensor decomposition problem, which is commonly known as parallel factor analysis (PARAFAC) [6].[1] In PARAFAC, there are elegant algebraic results regarding the unique decomposition conditions of (1) [6–8] (also the references therein), which translates into the key aspect of unique blind identifiability in BID-QSS. Simply stated,

☆ Part of this work was presented at the 2011 Asilomar Conference on Signals, Systems, and Computers [1].

* Corresponding author. Tel.: +852 39434350.
E-mail addresses: kklee@ee.cuhk.edu.hk (K.-K. Lee), wkma@ieee.org, wkma@ee.cuhk.edu.hk (W.-K. Ma), xfu@ee.cuhk.edu.hk (X. Fu), tsunghan@mx.nthu.edu.tw (T.-H. Chan), cychi@ee.nthu.edu.tw (C.-Y. Chi).

[1] Note that PARAFAC is also called canonical decomposition (CANDECOMP) in the literature.

0165-1684/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.sigpro.2013.03.037

these analysis results suggest that fundamentally, BID-QSS is capable of handling rather underdetermined mixing cases, i.e., a lot more sources than sensors. In terms of implementation, PARAFAC often formulates (1) as a least-squares data fitting problem. From this we have the now popularized trilinear alternating least squares (TALS) [6,9] and alternating-columns diagonal-centers (ACDC) [10] algorithms; see also [3,11–13] for other important endeavors.

The second major framework is based on the class of joint diagonalization (JD) techniques, where the aim is often to find a matrix $\mathbf{V}$ such that $\mathbf{V}\mathbf{R}_m\mathbf{V}^H$ are diagonal (or approximately diagonal) for all m. In this context, the development is more on the algorithm side, where now there exists a plethora of JD algorithms, e.g., [14–17]. Some algorithms worth mentioning are Pham's JD [14] and fast Frobenius diagonalization (FFDIAG) [16]. In [14], a connection between JD and maximum-likelihood estimation (under some mild assumptions) is also shown.

It is also worthwhile to notice that while many existing algorithms can be identified as either PARAFAC or JD based, sometimes the line between the two can be blurred. In uniformly weighted exhaustive diagonalization with Gauss iterations (UWEDGE) [18], the formulation can be seen as a combination of JD and PARAFAC criteria. While the principle of JD constrains itself to the overdetermined mixing case only, in second-order blind identification of underdetermined mixtures (SOBIUM) [8], the authors consider PARAFAC and devise a special kind of bilinear mapping to convert an underdetermined problem to a virtually overdetermined problem, which in turn enables application of JD in the underdetermined case. We also refer the readers to the literature [12,18] for a recent coverage of the various PARAFAC and JD formulations.

PARAFAC and JD are considered dominant frameworks, where most existing algorithms may be regarded as being originated from them. In this work, we take inspiration from direction-of-arrival (DOA) estimation and sensor array processing to develop an alternative formulation for BID-QSS. Specifically, we adopt a Khatri–Rao (KR) subspace formulation [19]. As will be shown, the advantage of KR subspace is that we can decouple the BID problem into a per-source BID problem, the latter of which exhibits a much simpler problem structure (relative to a complete BID formulation) and may be solved more efficiently. While this decoupled approach also has its own challenge, namely, on how one may stitch the decoupled results to yield a complete BID, we will study methods for overcoming this issue. We will propose two BID-QSS algorithms based on KR subspace, and their performance and complexity will be compared to those of the state-of-the-arts using simulations. The contribution of this paper lies in deriving highly efficient, algebraically simple, algorithms for per-source decoupled BID, and in using the former to construct BID-QSS algorithms that will be numerically shown to be competitive.

This paper is organized as follows. Section 2 describes the problem formulation. Section 3 establishes criteria of the KR subspace approach. Section 4 develops algorithms for per-source decoupled BID, while Section 5 considers all-sources BID-QSS based on the results in Section 4. Section 6 compares the performance and complexity of the proposed algorithms and several benchmarked algorithms. The paper is concluded in Section 7.

Notation: We largely follow the conventional notation in signal processing. In addition, $\mathrm{Diag}(\mathbf{x})$ denotes a diagonal matrix whose diagonal elements are $x_1, \ldots, x_n$; $\mathrm{vec}(\cdot)$ is a vectorization operator, where, for $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_m] \in \mathbb{C}^{n\times m}$, we have $\mathrm{vec}(\mathbf{X}) = [\mathbf{x}_1^T, \ldots, \mathbf{x}_m^T]^T \in \mathbb{C}^{nm}$; $\mathrm{vec}^{-1}(\cdot)$ represents the inverse operation of $\mathrm{vec}(\cdot)$; $\otimes$ is the Kronecker product; $\odot$ is the Khatri–Rao product, where, given $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_k]$ and $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_k]$, we have $\mathbf{A}\odot\mathbf{B} = [\mathbf{a}_1 \otimes \mathbf{b}_1, \ldots, \mathbf{a}_k \otimes \mathbf{b}_k]$; $\mathcal{R}(\mathbf{X})$ denotes the range space of $\mathbf{X}$; $\lambda_{\min}(\mathbf{X})$ and $\lambda_{\max}(\mathbf{X})$ denotes the magnitude-wise smallest and largest eigenvalues of $\mathbf{X}$, respectively; $\|\mathbf{x}\|_0$ is the zero norm, which counts the number of nonzero elements in $\mathbf{x}$; $\|\mathbf{x}\|_2$ and $\|\mathbf{X}\|_F$ are the vector 2-norm and matrix Frobenius norm, respectively; $\mathbf{X}^\dagger$ denotes the Moore–Penrose pseudo-inverse of $\mathbf{X}$; $\mathbf{X}_{1:k}$ denotes a submatrix of $\mathbf{X}$ that consists of the first k columns of $\mathbf{X}$.

2. Background

In this section we give the basic problem formulation of BID-QSS and KR subspace.

2.1. Physical signal model

We follow a standard BID-QSS formulation wherein the physical signal model is that of linear instantaneous mixtures

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) + \mathbf{v}(t), \quad t = 1, 2, \ldots \tag{2}$$

where we denote $\mathbf{x}(t) = [x_1(t), \ldots, x_N(t)]^T \in \mathbb{C}^N$ to be an N-sensor received signal vector, $\mathbf{s}(t) = [s_1(t), \ldots, s_K(t)]^T \in \mathbb{C}^K$ to be a source signal vector, with K being the number of sources, $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_K] \in \mathbb{C}^{N\times K}$ to be a mixing matrix, and $\mathbf{v}(t) \in \mathbb{C}^N$ to be noise. It is assumed that

(A1) The source signals $s_k(t), k = 1, \ldots, K$, are statistically independent of each other.

(A2) Each $s_k(t)$ is zero-mean wide-sense quasi-stationary; to be specific, $\mathrm{E}\{|s_k(t)|^2\}$ generally changes with t, but is fixed within every local time interval $[(m-1)L+1, mL]$, for some window length L and for any $m = 1, 2, \ldots$.

(A3) The noise vector $\mathbf{v}(t)$ is wide-sense stationary with mean zero and covariance $\sigma^2\mathbf{I}$, and is statistically independent of $\mathbf{s}(t)$.

Under the setup above, we consider the local covariances of $\mathbf{x}(t)$, which is defined as

$$\mathbf{R}_m = \mathrm{E}\{\mathbf{x}(t)\mathbf{x}(t)^H\}, \quad \text{for any } t \in [(m-1)L+1, mL]. \tag{3}$$

Note that, in practice, $\mathbf{R}_m$ can be estimated by local covariance sampling, e.g., $\mathbf{R}_m \simeq (1/L)\sum_{t=(m-1)L+1}^{mL} \mathbf{x}(t)\mathbf{x}(t)^H$. From (2) and its associated assumptions, it is readily shown that $\mathbf{R}_m$ adheres to the model

$$\mathbf{R}_m = \mathbf{A}\mathbf{D}_m\mathbf{A}^H + \sigma^2\mathbf{I}, \tag{4}$$

where $\mathbf{D}_m$ are the local covariances of $\mathbf{s}(t)$ and are given by $\mathbf{D}_m = \mathrm{Diag}(\mathbf{d}_m)$, in which $\mathbf{d}_m = [d_{m,1}, d_{m,2}, \ldots, d_{m,K}]^T$, $d_{m,k} = \mathrm{E}\{|s_k(t)|^2\}$ for any $t \in [(m-1)L+1, mL]$.

2.2. Local covariances model and Khatri–Rao subspace

Suppose that we have measured a number of M local covariances of $\mathbf{x}(t)$, or $\mathbf{R}_1,\dots,\mathbf{R}_M$. Our interest lies in exploiting the subspace characteristics of $\mathbf{R}_1,\dots,\mathbf{R}_M$ for blind identification of $\mathbf{A}$. To put into context, let us assume a noise covariance-free scenario

$$\mathbf{R}_m=\mathbf{A}\mathbf{D}_m\mathbf{A}^H,\quad m=1,\dots,M. \tag{5}$$

It will be reviewed in the next subsection that the noise covariance $\sigma^2\mathbf{I}$ can be removed from (4) using a simple preprocessing procedure. Consider the vectorization of $\mathbf{R}_m$ in (5)

$$\mathbf{y}_m \triangleq \mathrm{vec}(\mathbf{R}_m)=(\mathbf{A}^*\odot\mathbf{A})\mathbf{d}_m\in\mathbb{C}^{N^2}, \tag{6}$$

where $\mathbf{A}^*\odot\mathbf{A}$ is a self-Khatri–Rao product of $\mathbf{A}$ and takes the form

$$\mathbf{A}^*\odot\mathbf{A}=[\mathbf{a}_1^*\otimes\mathbf{a}_1,\dots,\mathbf{a}_K^*\otimes\mathbf{a}_K]\in\mathbb{C}^{N^2\times K}.$$

Note that to arrive at the right hand side of (6), we have used the matrix result $\mathrm{vec}(\mathbf{A}\mathbf{D}\mathbf{B}^H)=(\mathbf{B}^*\odot\mathbf{A})\mathbf{d}$, where $\mathbf{D}=\mathrm{Diag}(\mathbf{d})$ [6,9]. There is an interesting observation with (6), which has sparked interest in some recent DOA estimation studies [19,20]—Eq. (6) is virtually identical to a linear instantaneous mixture signal model, with a mixing matrix $\mathbf{A}^*\odot\mathbf{A}$ and a source vector $\mathbf{d}_m$. Hence, the insight is that one may exploit the self-Khatri-Rao product structure of the virtual mixing matrix $\mathbf{A}^*\odot\mathbf{A}$ to identify its physical counterpart, $\mathbf{A}$, blindly.

There are more than one ways to utilize the structure of $\mathbf{A}^*\odot\mathbf{A}$ for blind identification. For example, in the popularized TALS and ACDC algorithms [9,10], a least-squares fitting formulation for (6) is used. This work considers a subspace formulation. The following assumptions are made.

(A4) The mixing matrix $\mathbf{A}$ has full Kruskal rank; or, equivalently, any $\min\{K,N\}$ columns of $\mathbf{A}$ are linearly independent.

(A5) Let $\boldsymbol{\Psi}=[\mathbf{d}_1,\dots,\mathbf{d}_M]^T\in\mathbb{C}^{M\times K}$. The matrix $\boldsymbol{\Psi}$ has full column rank.

From (A4), it is readily deduced that

Fact 1. Assume (A4). The matrix $\mathbf{A}^*\odot\mathbf{A}$ has full column rank if $K\le 2N-1$ [6].[2]

Moreover, (A5) means that the source local variances, captured by $\mathbf{d}_m$, are assumed to be sufficiently time-varying and different in their variations, thereby satisfying the full column rank assumption on $\boldsymbol{\Psi}$. Also, note that (A5) implies $M\ge K$. Now, let us denote

$$\mathbf{Y}\triangleq[\mathbf{y}_1,\dots,\mathbf{y}_M]=(\mathbf{A}^*\odot\mathbf{A})\boldsymbol{\Psi}^T\in\mathbb{C}^{N^2\times M}. \tag{7}$$

Since $\mathbf{A}^*\odot\mathbf{A}$ and $\boldsymbol{\Psi}$ are of full column rank (assuming $K\le 2N-1$ for the former), basic matrix analysis leads us to the following results. First, $\mathbf{Y}$ has rank K, and admits a compact singular value decomposition (SVD)

$$\mathbf{Y}=\mathbf{U}_s\boldsymbol{\Sigma}_s\mathbf{V}_s^H, \tag{8}$$

where $\boldsymbol{\Sigma}_s\in\mathbb{R}^{K\times K}$ is the nonzero singular value matrix, and $\mathbf{U}_s\in\mathbb{C}^{N^2\times K}$ and $\mathbf{V}_s\in\mathbb{C}^{M\times K}$ are the associated left and right singular matrices, respectively. Second, we have

$$\mathcal{R}(\mathbf{U}_s)=\mathcal{R}(\mathbf{A}^*\odot\mathbf{A}). \tag{9}$$

The subspace $\mathcal{R}(\mathbf{U}_s)$ or $\mathcal{R}(\mathbf{A}^*\odot\mathbf{A})$ will be called the *Khatri–Rao (KR) subspace* in the sequel. Our endeavor will be focused on using the KR subspace identity in (9) for blind identification of $\mathbf{A}$.

2.3. Preprocessing

Before proceeding to the main development in the next sections, we should briefly describe two preprocessing procedures for the sake of self-containedness. The first procedure is noise covariance removal, where we return to the noise covariance-present model in (4). It is well known that $\lambda_{\min}(\mathbf{R}_m)=\lambda_{\min}(\mathbf{A}\mathbf{D}_m\mathbf{A}^H)+\sigma^2$, and that $\lambda_{\min}(\mathbf{A}\mathbf{D}_m\mathbf{A}^H)\ge 0$ (see, e.g., [21]). Moreover, for $N>K$, i.e., more sensors than sources, we have $\lambda_{\min}(\mathbf{A}\mathbf{D}_m\mathbf{A}^H)=0$. Hence, for this strictly overdetermined case, we can estimate σ^2 via

$$\hat{\sigma}^2=\min_{m=1,\dots,M}\lambda_{\min}(\mathbf{R}_m), \tag{10}$$

and then subtract $\sigma^2\mathbf{I}$ from $\mathbf{R}_m$. It should be noted that this noise covariance removal procedure has been previously suggested, e.g., in [3]. Interestingly, under a mild assumption, the noise removal procedure above also works for the case of $N\le K$.

Fact 2. If there exists an index m such that $\|\mathbf{d}_m\|_0<N$, then (10) correctly estimates σ^2.

Fact 2 is a direct consequence of $\lambda_{\min}(\mathbf{A}\mathbf{D}_m\mathbf{A}^H)=0$ for any m that satisfies $\|\mathbf{d}_m\|_0<N$, i.e., the active number of sources at time window m is less than N. Physically, this means that if the sources exhibit a mild amount of local sparsity, then (10) may also estimate σ^2 reliably in the underdetermined case.

The second procedure is prewhitening [22,23], which will be used in one of our algorithms to be developed. The goal is to transform the problem such that $\mathbf{A}$ becomes unitary. The procedure works only for the overdetermined case $N\ge K$, and is described as follows. Assume that the noise covariance has been removed, and consider the time-averaged global covariance

$$\overline{\mathbf{R}}=\frac{1}{M}\sum_{m=1}^{M}\mathbf{R}_m=\mathbf{A}\overline{\mathbf{D}}\mathbf{A}^H, \tag{11}$$

where, by (5), we have $\overline{\mathbf{D}}=(1/M)\sum_{m=1}^{M}\mathbf{D}_m$. Since $\overline{\mathbf{D}}$ is, without loss of generality, a positive definite matrix, we can apply a square-root factorization $\overline{\mathbf{R}}=\mathbf{B}\mathbf{B}^H$, where $\mathbf{B}\in\mathbb{C}^{N\times K}$ has full column rank (for $N\ge K$). The prewhitening operation is given by

$$\tilde{\mathbf{R}}_m=\mathbf{B}^\dagger\mathbf{R}_m(\mathbf{B}^\dagger)^H,\quad m=1,\dots,M. \tag{12}$$

From (5) and (12), the prewhitened local covariances $\tilde{\mathbf{R}}_m$

[2] Note that the condition $K\le 2N-1$ is a safe, but arguably conservative, condition for $\mathbf{A}^*\odot\mathbf{A}$ to have full column rank. There exist probabilistic claims for the full column rank condition of $\mathbf{A}^*\odot\mathbf{A}$, where the requirement can be much more relaxed than $K\le 2N-1$ [7].

can be written as

$$\tilde{\mathbf{R}}_m = \tilde{\mathbf{A}}\tilde{\mathbf{D}}_m\tilde{\mathbf{A}}^H, \quad m = 1, \ldots, M, \tag{13}$$

where $\tilde{\mathbf{A}} = \mathbf{B}^\dagger \mathbf{A}\overline{\mathbf{D}}^{1/2} \in \mathbb{C}^{K \times K}$ is the transformed mixing matrix, and $\tilde{\mathbf{D}}_m = \overline{\mathbf{D}}^{-1}\mathbf{D}_m$ the transformed source local covariances. It can be verified that $\tilde{\mathbf{A}}$ is unitary. Moreover, Eq. (13) follows the same problem structure as in the basic local covariance model (5).

3. KR subspace criteria

BID-QSS aims at estimating the mixing matrix $\mathbf{A}$ from the observed local covariances $\mathbf{R}_1, \ldots, \mathbf{R}_M$, given knowledge of the number of sources K. Following the subspace formulation in Section 2.2, we consider BID based on the KR subspace matrix $\mathbf{U}_s$. From the KR subspace identity (9), we see that any column $\mathbf{a}_k$ of the true mixing matrix $\mathbf{A}$ satisfies $\mathbf{a}_k^* \otimes \mathbf{a}_k \in \mathcal{R}(\mathbf{U}_s)$. This observation leads us to the following criterion for blind identification of $\mathbf{A}$:

Criterion 1 :

$$\begin{aligned} &\text{find} && \mathbf{a} \in \mathbb{C}^N \\ &\text{such that} && \mathbf{a}^* \otimes \mathbf{a} \in \mathcal{R}(\mathbf{U}_s). \end{aligned}$$

Criterion 1 suggests a column-decoupled BID approach—solving Criterion 1 amounts to finding one of the $\mathbf{a}_k$'s, assuming unique identifiability which will be discussed shortly. We should note that Criterion 1 is reminiscent of the criterion leading to MUSIC and some other subspace algorithms in the context of DOA estimation; see [21,24] and the references therein. Like the development in subspace-based DOA estimation, it is essential to prove the theoretical identifiability conditions of Criterion 1. To be specific, while any $\mathbf{a}_k$ is naturally a solution of Criterion 1, is it also true that a solution of Criterion 1 must be an $\mathbf{a}_k$, and no others? Consider the following theorem:

Theorem 1. *Assume* (5), (A4), *and* (A5). *A sufficient and necessary condition for*

$\mathbf{a}$ *satisfies Criterion* 1 $\Longleftrightarrow \mathbf{a} = c\mathbf{a}_k$ *for some k and constant* $c \in \mathbb{C}$

is when $K \le 2N-2$.

The proof of Theorem 1 is shown in Appendix A. Theorem 1 confirms that Criterion 1 is a sound criterion, when the number of sources does not exceed approximately twice of the number of sensors. It also means that Criterion 1 can operate in the underdetermined case.

As we will show in Section 4, a significant advantage of the column-decoupled BID criterion in Criterion 1 is that we can develop efficient algorithms for it. However, the decoupled nature of Criterion 1 does not tell how *all* the columns of $\mathbf{A}$ can simultaneously be identified. Section 5 will study how the column-decoupled BID solutions can be used to estimate the whole $\mathbf{A}$. In doing so, we will consider the following alternative criterion:

Criterion 2 :

$$\begin{aligned} &\text{find} && \mathbf{A} \in \mathbb{C}^{N \times K}, \ \boldsymbol{\Theta} \in \mathbb{C}^{K \times K} \\ &\text{such that} && \mathbf{U}_s = (\mathbf{A}^* \odot \mathbf{A})\boldsymbol{\Theta}. \end{aligned}$$

The rationale behind is that from (9), it holds true that any column of $\mathbf{U}_s$ lies in $\mathcal{R}(\mathbf{A}^* \odot \mathbf{A})$. Note that Criterion 2 is reminiscent of a subspace fitting criterion in DOA estimation [25]. Moreover, if we replace $\mathbf{U}_s$ in Criterion 2 by $\mathbf{Y}$, and expand the dimension of $\boldsymbol{\Theta}$ from $\mathbb{C}^{K \times K}$ to $\mathbb{C}^{K \times M}$ $(M \ge K)$, then the criterion is essentially no different from that of PARAFAC. Hence, Criterion 2 may alternatively be regarded as a dimension reduced PARAFAC. This relation in addition means that the PARAFAC unique identifiability results, which are well established [6,7,26], apply to Criterion 2. For example, using the standard result [26, Theorem 1], we can easily deduce that Criterion 2 uniquely identifies the true $\mathbf{A}$ (subjected to scalings and permutations) if $K \le 2N-2$.

4. Column-decoupled blind identification

In the last section, we have proposed a column-decoupled BID criterion, namely, Criterion 1. This criterion possesses a relatively simple structure when compared to other BID-QSS criteria, e.g., Criterion 2. In this section, we will exploit such structure to derive efficient column-wise BID algorithms.

4.1. Alternating projections

To implement Criterion 1, it is natural for one to formulate it as an optimization problem

$$\begin{aligned} &\min_{\mathbf{a} \in \mathbb{C}^N} && (\mathbf{a}^* \otimes \mathbf{a})^H \mathbf{P}_s^\perp (\mathbf{a}^* \otimes \mathbf{a}) \\ &\text{s.t.} && \|\mathbf{a}\|_2^2 = 1, \end{aligned} \tag{14}$$

where $\mathbf{P}_s^\perp = \mathbf{I} - \mathbf{U}_s\mathbf{U}_s^H$ denotes the orthogonal complement projector of the KR subspace $\mathcal{R}(\mathbf{U}_s)$. In words, we aim at minimizing the projection residual of $\mathbf{a}^* \otimes \mathbf{a}$ on $\mathcal{R}(\mathbf{U}_s)$. Problem (14) is a quartic polynomial optimization problem. Rather than dealing with its fourth-order multivariate polynomial objective directly, which may be difficult, our approach is based on an alternative formulation of (14) that will lead to a simple iterative algorithm. We claim that problem (14) is equivalent to

$$\begin{aligned} &\min_{\alpha \in \mathbb{R}, \mathbf{a} \in \mathbb{C}^N, \mathbf{h} \in \mathbb{C}^{N^2}} && \|\alpha \mathbf{a}^* \otimes \mathbf{a} - \mathbf{h}\|_2^2 \\ &\text{s.t.} && \alpha \in \{\pm 1\}, \quad \|\mathbf{a}\|_2^2 = 1, \quad \mathbf{h} \in \mathcal{R}(\mathbf{U}_s). \end{aligned} \tag{15}$$

The equivalence of problems (14) and (15) is shown as follows. Fixing $(\alpha, \mathbf{a})$, the optimization of (15) over $\mathbf{h}$ is a linear projection problem, whose solution is easily shown to be

$$\mathbf{h} = \mathbf{U}_s\mathbf{U}_s^H(\alpha \mathbf{a}^* \otimes \mathbf{a}). \tag{16}$$

By substituting (16) into (15), problem (15) can be reduced to

$$\begin{aligned} &\min_{\alpha \in \{\pm 1\}, \|\mathbf{a}\|_2^2 = 1} \|(\mathbf{I} - \mathbf{U}_s\mathbf{U}_s^H)(\alpha \mathbf{a}^* \otimes \mathbf{a})\|_2^2 \\ &= \min_{\|\mathbf{a}\|_2^2 = 1} (\mathbf{a}^* \otimes \mathbf{a})^H \mathbf{P}_s^\perp (\mathbf{a}^* \otimes \mathbf{a}), \end{aligned} \tag{17}$$

which is exactly the same as problem (14).

Problem (15) has an interpretation of finding a pair of closest points in two sets, namely, $\mathbf{h} \in \mathcal{R}(\mathbf{U}_s)$ and $(\alpha, \mathbf{a}) \in \{\pm 1\} \times \mathcal{U}^N$, where $\mathcal{U}^n = \{\mathbf{x} \in \mathbb{C}^n \,|\, \|\mathbf{x}\|_2^2 = 1\}$. Moreover, the formulation in (15) enables us to apply alternating projections (APs) [27], or alternating optimization, conveniently.

Essentially, the idea of AP is to fix $(\alpha, \mathbf{a})$ and solve (15) with respect to (w.r.t.) $\mathbf{h}$ at one time, and then fix $\mathbf{h}$ and solve (15) w.r.t. $(\alpha, \mathbf{a})$ at another time. For the partial optimization of (15) over $\mathbf{h}$, we have seen that the solution is (16). Let us examine the partial optimization of (15) over $(\alpha, \mathbf{a})$. By denoting $\mathbf{H} = \text{vec}^{-1}(\mathbf{h}) \in \mathbb{C}^{N \times N}$, problem (15) can be re-expressed as

$$\begin{aligned} \min_{\alpha, \mathbf{a}, \mathbf{H}} \quad & \|\alpha \mathbf{a}\mathbf{a}^H - \mathbf{H}\|_F^2 \\ \text{s.t.} \quad & \alpha \in \{\pm 1\}, \quad \|\mathbf{a}\|_2^2 = 1, \quad \text{vec}(\mathbf{H}) \in \mathcal{R}(\mathbf{U}_s), \end{aligned} \tag{18}$$

where we use the property $\text{vec}(\mathbf{a}\mathbf{a}^H) = \mathbf{a}^* \otimes \mathbf{a}$ to turn (15)–(18). For any $\alpha \in \{\pm 1\}$ and $\|\mathbf{a}\|_2^2 = 1$, the objective function of (18) yields

$$\begin{aligned} \|\alpha \mathbf{a}\mathbf{a}^H - \mathbf{H}\|_F^2 &= 1 - 2\alpha \text{Re}\{\mathbf{a}^H \mathbf{H} \mathbf{a}\} + \|\mathbf{H}\|_F^2 \\ &\geq 1 - 2|\text{Re}\{\mathbf{a}^H \mathbf{H} \mathbf{a}\}| + \|\mathbf{H}\|_F^2, \end{aligned} \tag{19}$$

where equality in (19) holds when $\alpha = \text{Re}\{\mathbf{a}^H \mathbf{H} \mathbf{a}\} / |\text{Re}\{\mathbf{a}^H \mathbf{H} \mathbf{a}\}|$. Moreover, the second term in (19) is minimized when $|\text{Re}\{\mathbf{a}^H \mathbf{H} \mathbf{a}\}|$ is maximized, and the latter is achieved when $\mathbf{a}$ aligns to a magnitude-wise most significant eigenvector of $(\mathbf{H} + \mathbf{H}^H)/2$ (note that $\text{Re}\{\mathbf{a}^H \mathbf{H} \mathbf{a}\} = \frac{1}{2}\mathbf{a}^H(\mathbf{H} + \mathbf{H}^H)\mathbf{a}$). Hence, the partial optimization of (18) w.r.t. $(\alpha, \mathbf{a})$ has a closed-form solution given by

$$\mathbf{a} = \boldsymbol{q}_{\max}\left(\frac{1}{2}(\mathbf{H} + \mathbf{H}^H)\right), \quad \alpha = \frac{\lambda_{\max}\left(\frac{1}{2}(\mathbf{H} + \mathbf{H}^H)\right)}{\left|\lambda_{\max}\left(\frac{1}{2}(\mathbf{H} + \mathbf{H}^H)\right)\right|}, \tag{20}$$

where $\lambda_{\max}(\mathbf{X})$ denotes the largest eigenvalue of $\mathbf{X}$ (magnitude-wise), and $\boldsymbol{q}_{\max}(\mathbf{X})$ denotes a unit-2-norm eigenvector of $\mathbf{X}$ associated with $\lambda_{\max}(\mathbf{X})$.

Our implementation of the AP method is shown in Algorithm 1. We can see that the algorithm is simple to implement.

Algorithm 1. AP algorithm for problem (15).

```
Input: the KR subspace matrix $\mathbf{U}_s$;
$\mathbf{H} := \text{vec}^{-1}(\mathbf{U}_s \boldsymbol{\xi})$, $\boldsymbol{\xi} \sim \mathcal{CN}(\mathbf{0}, \mathbf{I})$ (random initialization);
repeat
    $\mathbf{a} := \boldsymbol{q}_{\max}\left(\frac{1}{2}(\mathbf{H} + \mathbf{H}^H)\right)$; $\alpha := \frac{\lambda_{\max}\left(\frac{1}{2}(\mathbf{H}+\mathbf{H}^H)\right)}{\left|\lambda_{\max}\left(\frac{1}{2}(\mathbf{H}+\mathbf{H}^H)\right)\right|}$;
    $\mathbf{H} := \text{vec}^{-1}(\mathbf{U}_s \mathbf{U}_s^H (\alpha \mathbf{a}^* \otimes \mathbf{a}))$;
until a stopping criterion is satisfied.
Output: $\mathbf{a}$ as an estimate of a column of the mixing matrix.
```

4.2. *Nuclear norm regularization*

While the AP algorithm derived in the last subsection is simple, we found that empirically, AP generally exhibits slow objective value convergence. In this subsection, we consider a modified AP algorithm for improving convergence. Our approach is to apply regularization to the BID problem (14). To be specific, we consider the equivalent AP formulation (18) and add a regularization term on it to obtain

$$\begin{aligned} \min_{\alpha, \mathbf{a}, \mathbf{H}} \quad & \|\alpha \mathbf{a}\mathbf{a}^H - \mathbf{H}\|_F^2 + \gamma \text{rank}(\mathbf{H}) \\ \text{s.t.} \quad & \alpha \in \{\pm 1\}, \quad \|\mathbf{a}\|_2^2 = 1, \quad \text{vec}(\mathbf{H}) \in \mathcal{R}(\mathbf{U}_s), \end{aligned} \tag{21}$$

where $\gamma > 0$ is a regularization parameter. It can be shown that

Fact 3. Suppose that the unique identifiability premises in Theorem 1 hold. Then, for any $\gamma < 1$, problem (21) is equivalent to problem (18) in the sense that their optimal solutions are the same.

Proof. Let $(\alpha^\star, \mathbf{a}^\star, \mathbf{H}^\star)$ denote an optimal solution to problem (18). Under the unique identifiability premises in Theorem 1, $(\alpha^\star, \mathbf{a}^\star, \mathbf{H}^\star)$ must take the form $(\alpha^\star, \mathbf{a}^\star, \mathbf{H}^\star) = (\alpha, \mathbf{a}_i, \alpha \mathbf{a}_i \mathbf{a}_i^H)$ for any $\alpha \in \{\pm 1\}$ and any true mixing matrix column $\mathbf{a}_i$, $i = 1, \ldots, K$. We seek to prove that any optimal solution of problem (21) must also be $(\alpha^\star, \mathbf{a}^\star, \mathbf{H}^\star)$. Let

$$f(\alpha, \mathbf{a}, \mathbf{H}) = \|\alpha \mathbf{a}\mathbf{a}^H - \mathbf{H}\|_F^2 + \gamma \text{rank}(\mathbf{H})$$

be the objective function of problem (21). For any feasible $(\alpha, \mathbf{a}, \mathbf{H})$ with $\text{rank}(\mathbf{H}) \geq 1$, we have

$$f(\alpha, \mathbf{a}, \mathbf{H}) \geq \gamma \text{rank}(\mathbf{H}) \geq \gamma.$$

Moreover, the equalities above are achieved if and only if $\mathbf{H} = \alpha \mathbf{a}\mathbf{a}^H$; this is possible only when $(\alpha, \mathbf{a}, \mathbf{H}) = (\alpha^\star, \mathbf{a}^\star, \mathbf{H}^\star)$. In other words, $(\alpha^\star, \mathbf{a}^\star, \mathbf{H}^\star)$ is optimal to problem (21) if the case of $\text{rank}(\mathbf{H}) = 0$ does not lead to an objective value lower than γ. Let us consider $\text{rank}(\mathbf{H}) = 0$, which is equivalent to $\mathbf{H} = \mathbf{0}$. For any feasible $(\alpha, \mathbf{a}, \mathbf{0})$, we have

$$f(\alpha, \mathbf{a}, \mathbf{H}) = \|\alpha \mathbf{a}\mathbf{a}^H\|^2 = \|\mathbf{a}\|_2^4 = 1.$$

Consequently, for $\gamma < 1$, any feasible $(\alpha, \mathbf{a}, \mathbf{0})$ is not optimal to problem (21). □

The reason for studying problem (21) is with AP. Suppose that AP is applied to problem (21) in the same way as before. Then, for the partial optimization of (21) w.r.t. $\mathbf{H}$, i.e.

$$\begin{aligned} \min_{\mathbf{H}} \quad & \|\alpha \mathbf{a}\mathbf{a}^H - \mathbf{H}\|_F^2 + \gamma \text{rank}(\mathbf{H}) \\ \text{s.t.} \quad & \text{vec}(\mathbf{H}) \in \mathcal{R}(\mathbf{U}_s), \end{aligned} \tag{22}$$

the term $\gamma \text{rank}(\mathbf{H})$ would provide an incentive for (22) to yield a lower rank optimal solution $\mathbf{H}$. Subsequently, we may push $\mathbf{H}$ closer to a rank-one solution, thereby helping AP to converge faster.

To implement AP for problem (21), the key question lies in solving the partial optimization problem (22). Unfortunately, problem (22) is unlikely to be tractable, since $\text{rank}(\mathbf{H})$ is nonconvex. Hence, as a compromise, we replace $\text{rank}(\mathbf{H})$ in (22) by its convex envelope, namely, the nuclear norm

$$\begin{aligned} \min_{\mathbf{H}} \quad & \|\alpha \mathbf{a}\mathbf{a}^H - \mathbf{H}\|_F^2 + \gamma \|\mathbf{H}\|_* \\ \text{s.t.} \quad & \text{vec}(\mathbf{H}) \in \mathcal{R}(\mathbf{U}_s), \end{aligned} \tag{23}$$

where $\|\mathbf{H}\|_* = \sum_{i=1}^{r} \sigma_i(\mathbf{H})$ is the nuclear norm, with $\sigma_i(\mathbf{H})$ denoting the ith nonzero singular value of $\mathbf{H}$ and $r = \text{rank}(\mathbf{H})$. We should note that the notion of using the nuclear norm to approximate the rank function was proposed in low-rank matrix recovery [28,29], a topic that has recently drawn much interest. A significant advantage of problem (23) is that it is a convex tractable problem—it can be reformulated as a semidefinite program [30] and then conveniently processed by a general-purpose interior-point algorithm [31]. Alternatively, one can custom-build simple first-order optimization methods for problem (23), a representative one of which is the augmented Lagrangian method of multipliers (ADMM) [32]. In this paper, we will choose ADMM. Our derivation of the ADMM for problem (23) is described in Appendix B.

To summarize, the nuclear-norm AP (NAP) algorithm proposed in this subsection has its routines identical to Algorithm 1, except for line 4 where the solution $\mathbf{H}$ is obtained by solving (23) via ADMM. The pseudo-code of NAP, with the ADMM routines included, is shown in Algorithm 2.

Algorithm 2. Nuclear-norm AP (NAP) algorithm for problem (21).

```
Input: the KR subspace matrix $\mathbf{U}_s$, a regularization parameter $\gamma$, and an ADMM parameter $\rho$;
1:  $\mathbf{H}:=\mathrm{vec}^{-1}(\mathbf{U}_s\boldsymbol{\xi})$, $\boldsymbol{\xi}\sim\mathcal{CN}(\mathbf{0},\mathbf{I})$, $\mathbf{G}=\mathbf{Z}=\mathbf{0}$ (initializations);
2:  repeat
3:      $\mathbf{a}:=\boldsymbol{q}_{\max}\left(\frac{1}{2}(\mathbf{H}+\mathbf{H}^H\right))$; $\alpha:=\frac{\lambda_{\max}\left(\frac{1}{2}(\mathbf{H}+\mathbf{H}^H)\right)}{\left|\lambda_{\max}\left(\frac{1}{2}(\mathbf{H}+\mathbf{H}^H)\right)\right|}$;
4:      repeat (ADMM loop)
5:          $\mathbf{H}:=\frac{1}{\rho+2}\mathrm{vec}^{-1}\{\mathbf{U}_s\mathbf{U}_s^H(2\alpha\mathbf{a}^*\otimes\mathbf{a}+\rho\mathrm{vec}(\mathbf{G}-\mathbf{Z}))\}$;
6:          compute the SVD $(\mathbf{U},\boldsymbol{\Sigma},\mathbf{V})$ of $\mathbf{H}+\mathbf{Z}$;
7:          $d_i:=\max\{0,\Sigma_{ii}-\gamma/\rho\}$, $i=1,\ldots,K$;
8:          $\mathbf{G}:=\mathbf{U}\mathrm{Diag}(\mathbf{d})\mathbf{V}^H$;
9:          $\mathbf{Z}:=\mathbf{Z}+(\mathbf{H}-\mathbf{G})$;
10:     until a stopping criterion is satisfied.
11: until a stopping criterion is satisfied.
Output: $\mathbf{a}$ as an estimate of a column of the mixing matrix.
```

To demonstrate whether NAP can improve convergence, we herein show a numerical example. A realization of the local covariances $\mathbf{R}_1,\ldots,\mathbf{R}_M$ is synthetically generated according to the basic model (5), where we set $K=5$, $N=6$, $M=200$. The parameter settings of NAP are $\gamma=0.5$, $\rho=1$. Fig. 1 plots the projection residuals $(\mathbf{a}^*\otimes\mathbf{a})^H\mathbf{P}_s^{\perp}(\mathbf{a}^*\otimes\mathbf{a})$ of AP and NAP against the iteration numbers. Notice that reaching a projection residual of –300 dB indicates almost errorless BID of a column of $\mathbf{A}$. We can see that AP is indeed slow in convergence, while NAP exhibits a significantly improved convergence speed.

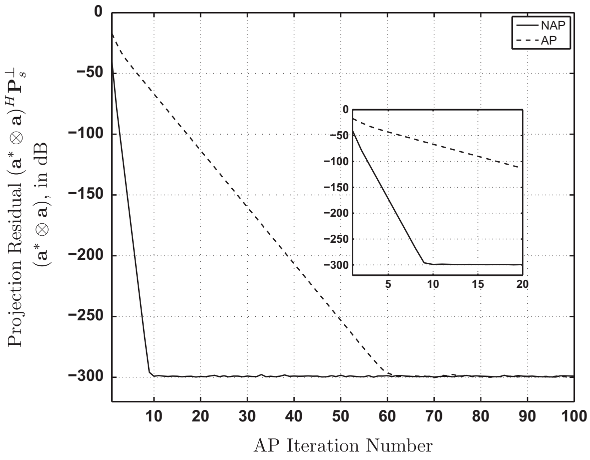

Fig. 1. Convergence improvement of NAP.

4.3. Convergence for unitary mixing matrices

The NAP algorithm in the last subsection uses a rank regularized methodology as the way to improve the convergence of AP, which subsequently requires us to solve a nuclear-norm optimization problem for each AP iteration. The nuclear-norm problem is convex and can be efficiently handled by ADMM, but solving them inevitably incurs a higher per-iteration complexity. While we will employ NAP, especially in the underdetermined case, an alternative perspective is to understand the convergence of the original simple AP by analysis, and see how or when convergence may be improved. In general, analyzing AP convergence for general $\mathbf{A}$ can be a formidable task—the main challenge lies in the nonconvex constraint $\|\mathbf{a}\|_2^2=1$. However, for the case of unitary $\mathbf{A}$, an AP convergence result can be established:

Theorem 2. *Assume (5),* (A4), *and* (A5), *and consider unitary $\mathbf{A}$. With probability one and within one iteration, the iterate $\mathbf{a}$ of Algorithm 1 equals $\mathbf{a}=c\mathbf{a}_k$, for some $c\neq0$ and $k\in\{1,\ldots,K\}$.*

The proof of Theorem 2 will be given by the end of this subsection. Theorem 2 reveals that the AP convergence for unitary $\mathbf{A}$ is drastically different from that of general $\mathbf{A}$. While we have numerically illustrated in Fig. 1 that AP exhibits slow convergence for a general $\mathbf{A}$, Theorem 2 indicates that the AP convergence for unitary $\mathbf{A}$ is within one iteration. The idea behind the proof of Theorem 2 is that for unitary $\mathbf{A}$, there is a strong connection between $\mathbf{A}$ and the eigendecomposition of $\mathbf{H}$. Consequently, we can exploit that connection to obtain the within-one-iteration claim.

It is important to discuss the practical utility of Theorem 2. In general, $\mathbf{A}$ is supposed to be non-unitary. However, for the overdetermined case ($N\geq K$), we can employ prewhitening (see Section 2.3) to transform $\mathbf{A}$ to a unitary matrix. Hence, we can take advantage of the fast AP convergence for unitary $\mathbf{A}$ by performing BID on prewhitened local covariances. Once the prewhitened mixing matrix is estimated, we can "post-dewhiten" the

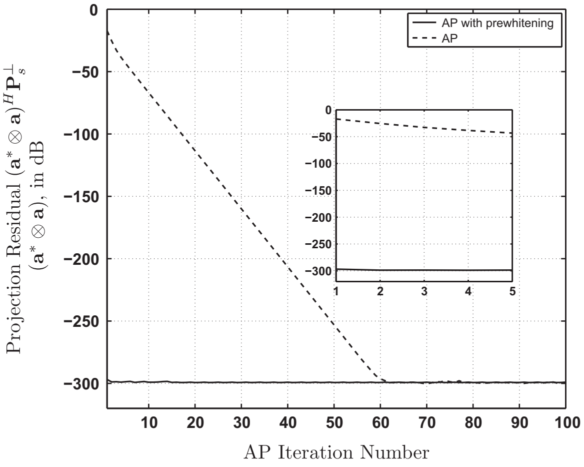

Fig. 2. Convergence improvement with prewhitening.

estimate to recover the original mixing matrix. In Fig. 2, we show an numerical result demonstrating the AP convergence before and after prewhitening; the numerical settings are the same as in Fig. 1. Fig. 2 confirms the fast convergence claim in Theorem 2.

Proof of Theorem 2. By the KR subspace identity (9), we have the relation

$$\mathbf{U}_s = (\mathbf{A}^* \odot \mathbf{A})\boldsymbol{\Theta} \tag{24}$$

for some $\boldsymbol{\Theta} \in \mathbb{C}^{K\times K}$. When $\mathbf{A}$ is unitary, one can verify that $(\mathbf{A}^*\odot\mathbf{A})^H(\mathbf{A}^*\odot\mathbf{A}) = \mathbf{I}$. Using this result and (24), we show from $\mathbf{U}_s^H\mathbf{U}_s = \mathbf{I}$ that $\boldsymbol{\Theta}^H\boldsymbol{\Theta} = \mathbf{I}$, i.e., $\boldsymbol{\Theta}$ is unitary. Consider the random initialization in line 1 of Algorithm 1, which can be expressed as

$$\text{vec}(\mathbf{H}) = \mathbf{U}_s\boldsymbol{\xi} = (\mathbf{A}^*\odot\mathbf{A})\boldsymbol{\eta}, \tag{25}$$

where $\boldsymbol{\eta} = \boldsymbol{\Theta}\boldsymbol{\xi}$. Since $\boldsymbol{\Theta}$ is unitary and $\boldsymbol{\xi} \sim \mathcal{CN}(\mathbf{0}, \mathbf{I})$, we have $\boldsymbol{\eta} \sim \mathcal{CN}(\mathbf{0}, \mathbf{I})$. Next, consider line 3 of Algorithm 1 over the first AP iteration. Devectorizing (25) yields

$$\tfrac{1}{2}(\mathbf{H} + \mathbf{H}^H) = \mathbf{A}\,\text{Diag}(\text{Re}\{\boldsymbol{\eta}\})\mathbf{A}^H. \tag{26}$$

Since $\mathbf{A}$ is unitary, the right-hand side of (26) is already an eigenvalue decomposition (EVD) of $\frac{1}{2}(\mathbf{H}+\mathbf{H}^H)$. The remaining question is whether (26) is the unique EVD. It is known that if the eigenvalues $\text{Re}\{\eta_1\}, \ldots, \text{Re}\{\eta_K\}$ are distinct, then the corresponding EVD is unique. As $\boldsymbol{\eta}$ is a continuous random vector, $\text{Re}\{\eta_i\} = \text{Re}\{\eta_j\}$ holds with probability zero for any $i \neq j$. Therefore, with probability one, the update $\mathbf{a} := \boldsymbol{q}_{\max}(\frac{1}{2}(\mathbf{H}+\mathbf{H}^H))$ picks up $\mathbf{a}_\ell$ (up to a scaling factor), where $\ell = \arg\max_i |\text{Re}\{\eta_i\}|$. It also follows from Algorithm 1 that for the second iteration and onward, the update $\mathbf{a}$ still stays at $\mathbf{a}_\ell$. □

5. Complete blind identification using the column-decoupled solutions

In the previous section, we have developed two efficient column-decoupled BID algorithms (specifically, NAP and prewhitened AP). However, as we have previously noted, they alone do not complete the task of blind identification of the whole mixing matrix $\mathbf{A}$. In this section, we turn our attention to this aspect. We will further devise algorithms that use the column-decoupled BID solutions to perform complete BID.

5.1. *KR subspace fitting*

Let us consider Criterion 2 in Section 3, which is a complete BID criterion. From Criterion 2, an optimization formulation one may have in mind is the least-squares fitting

$$\min_{\mathbf{A}\in\mathbb{C}^{N\times K}, \boldsymbol{\Theta}\in\mathbb{C}^{K\times K}} \|\mathbf{U}_s - (\mathbf{A}^*\odot\mathbf{A})\boldsymbol{\Theta}\|_F^2. \tag{27}$$

Problem (27) is fundamentally a hard optimization problem—its objective is a sixth-order multivariate polynomial w.r.t. $(\mathbf{A}, \boldsymbol{\Theta})$. Moreover, problem (27) is structurally no different from the least-squares data fitting formulation used in PARAFAC, although the former aims at fitting the subspace, rather than the data $\mathbf{Y}$. In this regard, we should note that there exists pragmatic algorithms, like ACDC and TALS, that have been empirically found to produce reasonable estimates for problems in the form of (27). ACDC and TALS are alternating optimization algorithms that require initialization of $\mathbf{A}$. In particular, poor initializations are likely to slow down convergence, or lead to unsatisfactory estimates. Our empirical experience with applying ACDC to problem (27) is that for randomly generated initializations (which is a common initialization scheme), the number of iterations required may be very large. On the other hand, NAP can effectively identify columns of $\mathbf{A}$. Hence, we can consider a two-stage approach where we run NAP multiple times to find some columns of $\mathbf{A}$ (or, if lucky, all), and then use them to initialize ACDC or TALS. It will be demonstrated by simulations in the next section that this two-stage approach can reduce both the number of iterations and the estimation errors.

The detailed implementation of the two-stage approach is given in Algorithm 3. In the algorithm, ACDC is employed to process problem (27). NAP is run multiple times, and we keep only distinct estimates outputted by NAP. They are used to form part of the initialization of $\mathbf{A}$ (or all, if all the columns of $\mathbf{A}$ are successfully identified by NAP), while the rest are randomly generated. The algorithm will be called *NAP-initialized subspace ACDC*.

Algorithm 3. NAP-initialized subspace ACDC

```
Input: local covariance matrices $\mathbf{R}_1, \ldots, \mathbf{R}_M$; a maximum number of
NAP $J$, and a validation parameter $\epsilon$;
1:  compute the SVD $(\mathbf{U}, \boldsymbol{\Sigma}, \mathbf{V})$ of $\mathbf{Y} = [\text{vec}(\mathbf{R}_1, \ldots, \text{vec}(\mathbf{R}_M))]$, $\mathbf{U}_s := \mathbf{U}_{1:K}$;
2:  run Algorithm 2 to obtain $\hat{\mathbf{a}}_1$, and set $k=2$;
3:  for $j = 1 : J$ do
4:     run Algorithm 2 to obtain $\hat{\mathbf{a}}$;
5:     if $|\hat{\mathbf{a}}^H\hat{\mathbf{a}}_\ell| < \epsilon, \forall \ell < k$ then
6:        $\hat{\mathbf{a}}_k := \hat{\mathbf{a}}$ and set $k := k+1$;
7:     end if
8:     if $k > K$ then goto step 10;
9:  end for
10: run ACDC to (27) with $\mathbf{A}_0 = [\hat{\mathbf{a}}_1, \ldots, \hat{\mathbf{a}}_{k-1}, \mathbf{a}_k \ldots, \mathbf{a}_K]$ as an
    initialization to obtain $\hat{\mathbf{A}}$, where $\mathbf{a}_\ell \sim \mathcal{CN}(\mathbf{0}, \mathbf{I})$, for $\ell = k, \ldots, K$
Output: $\hat{\mathbf{A}}$ as an estimate of the mixing matrix.
```

5.2. *Successive optimization for unitary* $\mathbf{A}$

The NAP-initialized subspace ACDC algorithm in Algorithm 3 is derived for blind identification of general $\mathbf{A}$, both overdetermined and underdetermined. However, for the overdetermined case, we can develop a much more efficient algorithm in place of Algorithm 3. A crucial component lies in prewhitening again, which, as described earlier, enables us to transform $\mathbf{A}$ to a unitary matrix.

Let us focus on the case of unitary $\mathbf{A}$. Recall from Criterion 2 that $\mathbf{U}_s = (\mathbf{A}^*\odot\mathbf{A})\boldsymbol{\Theta}$ is desired. When $\mathbf{A}$ is unitary, we have shown in the proof of Theorem 2 that $\boldsymbol{\Theta}$ is unitary. For this reason, we consider a modified form of the KR subspace fitting formulation (27):

$$\begin{aligned} &\min_{\mathbf{A}\in\mathbb{C}^{K\times K}, \boldsymbol{\Theta}\in\mathbb{C}^{K\times K}} \quad \|\mathbf{U}_s - (\mathbf{A}^*\odot\mathbf{A})\boldsymbol{\Theta}\|_F^2 \\ &\quad \text{s.t.} \quad \boldsymbol{\Theta}\boldsymbol{\Theta}^H = \mathbf{I}, \end{aligned} \tag{28}$$

where we incorporate the unitarity of $\boldsymbol{\Theta}$ as a constraint. Our interest lies in reformulating problem (28) to a form that will enable us to derive a divide-and-conquer strategy for handling problem (28). To do so, let $\mathbf{Q}=\mathbf{\Theta}^H$. By the rotational invariance of $\|\cdot\|_F$, we can rewrite (28) as

$$\begin{aligned}\min_{\mathbf{A},\mathbf{Q}} \quad & \|\mathbf{U}_s\mathbf{Q}-(\mathbf{A}^*\odot\mathbf{A})\|_F^2 \\ \text{s.t.} \quad & \mathbf{Q}^H\mathbf{Q}=\mathbf{I}.\end{aligned} \tag{29}$$

In addition, by substituting $\mathbf{h}_k=\mathbf{U}_s\mathbf{q}_k\in\mathcal{R}(\mathbf{U}_s)$, where $\mathbf{q}_k$ is the kth column of $\mathbf{Q}$, problem (29) can be equivalently expressed as

$$\begin{aligned}\min_{\mathbf{A},\mathbf{h}_1,\ldots,\mathbf{h}_K} \quad & \sum_{k=1}^{K}\|\mathbf{h}_k-\mathbf{a}_k^*\otimes\mathbf{a}_k\|_2^2 \\ \text{s.t.} \quad & \mathbf{h}_k\in\mathcal{R}(\mathbf{U}_s), \quad k=1,\ldots,K, \\ & \mathbf{h}_k^H\mathbf{h}_\ell=0, \quad \forall k\neq\ell, \\ & \|\mathbf{h}_k\|_2^2=1, \quad k=1,\ldots,K.\end{aligned} \tag{30}$$

Let us slightly modify problem (30) by replacing the constraints $\|\mathbf{h}_k\|_2^2=1$ with $\|\mathbf{a}_k\|_2^2=1$:

$$\begin{aligned}\min_{\mathbf{A},\mathbf{h}_1,\ldots,\mathbf{h}_K} \quad & \sum_{k=1}^{K}\|\mathbf{h}_k-\mathbf{a}_k^*\otimes\mathbf{a}_k\|_2^2 \\ \text{s.t.} \quad & \mathbf{h}_k\in\mathcal{R}(\mathbf{U}_s), \quad k=1,\ldots,K, \\ & \mathbf{h}_k^H\mathbf{h}_\ell=0, \quad \forall k\neq\ell, \\ & \|\mathbf{a}_k\|_2^2=1, \quad k=1,\ldots,K.\end{aligned} \tag{31}$$

With the formulation in (31), we are ready to describe the proposed optimization strategy. A key observation is that problem (31) can be expressed as (32), shown at the bottom of the page where

$$\mathcal{H}_k(\mathbf{h}_1,\ldots,\mathbf{h}_{k-1})=\{\mathbf{h}\in\mathcal{R}(\mathbf{U}_s)|\mathbf{h}^H\mathbf{h}_\ell=0,\ell=1,\ldots,k-1\}. \tag{33}$$

The expression in (32) suggests that we can apply a successive optimization strategy. To be specific, we decouple (32) into K sequentially processed stages. At stage k, we aim at solving

$$\begin{aligned}(\hat{\mathbf{a}}_k,\hat{\mathbf{h}}_k)=\arg\min_{\mathbf{a}_k,\mathbf{h}_k}\|\mathbf{h}_k-\mathbf{a}_k^*\otimes\mathbf{a}_k\|_2^2 \\ \text{s.t. } \|\mathbf{a}_k\|_2^2=1, \mathbf{h}_k\in\mathcal{H}_k(\hat{\mathbf{h}}_1,\ldots,\hat{\mathbf{h}}_{k-1})\end{aligned} \tag{34}$$

where $\hat{\mathbf{h}}_1,\ldots,\hat{\mathbf{h}}_{k-1}$ are the decisions in the previous stages $1,\ldots,k-1$. Moreover, it can be shown that since $\hat{\mathbf{h}}_1,\ldots,\hat{\mathbf{h}}_{k-1}\in\mathcal{R}(\mathbf{U}_s)$, the subspace $\mathcal{H}_k(\hat{\mathbf{h}}_1,\ldots,\hat{\mathbf{h}}_{k-1})$ takes an explicit form

$$\mathcal{H}_k(\hat{\mathbf{h}}_1,\ldots,\hat{\mathbf{h}}_{k-1})=\mathcal{R}(\mathbf{P}^{\perp}_{\hat{\mathbf{H}}_{1:k-1}}\mathbf{U}_s) \tag{35}$$

where $\hat{\mathbf{H}}_{1:k-1}=[\hat{\mathbf{h}}_1,\ldots,\hat{\mathbf{h}}_{k-1}]$, $\mathbf{P}^{\perp}_{\hat{\mathbf{H}}_{1:k-1}}=\mathbf{I}-\hat{\mathbf{H}}_{1:k-1}(\hat{\mathbf{H}}^H_{1:k-1}\hat{\mathbf{H}}_{1:k-1})^{-1}\hat{\mathbf{H}}^H_{1:k-1}$. Now, we can see the following connection: *problem (34) is equivalently the AP problem (15), with the original subspace matrix* $\mathbf{U}_s$ being replaced by $\mathbf{P}^{\perp}_{\hat{\mathbf{H}}_{1:k-1}}\mathbf{U}_s$. As a result, problem (34) can be readily handled by applying AP (Algorithm 1). It is also interesting to note that if the previous stages $1,\ldots,k-1$ have perfectly identified some of the mixing matrix columns, say, $\hat{\mathbf{h}}_1=\mathbf{a}_1^*\otimes\mathbf{a}_1,\ldots,\hat{\mathbf{h}}_{k-1}=\mathbf{a}_{k-1}^*\otimes\mathbf{a}_{k-1}$, then, by the orthogonality of $\mathbf{A}^*\odot\mathbf{A}$ (implied by the unitarity of $\mathbf{A}$), we can show that

$$\mathcal{H}_k(\hat{\mathbf{h}}_1,\ldots,\hat{\mathbf{h}}_{k-1})=\mathcal{R}([\mathbf{a}_k^*\otimes\mathbf{a}_k,\ldots,\mathbf{a}_K^*\otimes\mathbf{a}_K]),$$

in which the previously found columns are removed from the subspace. Consequently, at stage k, problem (34) will identify a new mixing matrix column.

The AP-based successive optimization method proposed above is summarized in a pseudo-code form in Algorithm 4. The algorithm is named as the *prewhitened alternating projection algorithm* (PAPA) for convenience. There are two major advantages of PAPA compared to the previous subspace ACDC algorithm. First, PAPA deals with K AP problems only. Second, since $\mathbf{A}$ is unitary, the AP convergence is expected to be fast according to Theorem 2. These two merits make PAPA a computationally very competitive algorithm, as our simulation results will demonstrate. However, we should reiterate that PAPA is for the overdetermined case only. It is also interesting to note that the successive operation in PAPA shows a flavor reminiscent of that in deflation-based FastICA [33] (and the references therein), where they both estimate source components in a one-by-one manner.

Algorithm 4. Prewhitened alternating projection algorithm.

```
Input: local covariance matrices $\mathbf{R}_1,\ldots,\mathbf{R}_M$;
1: $\overline{\mathbf{R}}=\frac{1}{M}\sum_{m=1}^{M}\mathbf{R}_m$;
2: compute a square-root factorization $\overline{\mathbf{R}}=\mathbf{B}\mathbf{B}^H$;
3: $\mathbf{R}_m:=\mathbf{B}^{\dagger}\mathbf{R}_m(\mathbf{B}^{\dagger})^H$, $m=1,\ldots,M$; (prewhitening)
4: compute the SVD $(\mathbf{U},\mathbf{\Sigma},\mathbf{V})$ of $\mathbf{Y}=[\mathrm{vec}(\mathbf{R}_1,\ldots,\mathrm{vec}(\mathbf{R}_M))]$, $\mathbf{U}_s:=\mathbf{U}_{1:K}$,
   and set $k=1$;
5: run Algorithm 1 with $\mathbf{P}^{\perp}_{\hat{\mathbf{H}}_{1:k-1}}\mathbf{U}_s$ as the input to obtain $(\hat{\mathbf{a}}_k,\hat{\mathbf{h}}_k)$,
   where $\hat{\mathbf{H}}_{1:k-1}=[\hat{\mathbf{h}}_1,\ldots,\hat{\mathbf{h}}_{k-1}]$;
6: set $k:=k+1$ and goto step 5 until $k>K$;
7: $\hat{\mathbf{A}}:=\mathbf{B}[\hat{\mathbf{a}}_1,\ldots,\hat{\mathbf{a}}_K]$; (post-dewhitening)
Output: $\hat{\mathbf{A}}$ as an estimate of the mixing matrix
```

6. Simulations

We provide simulations to demonstrate the advantages of the proposed algorithms compared to some other benchmarked QSS-based blind identification algorithms. The simulation settings are described as follows. We consider real-valued mixtures and sources. The mixing matrix $\mathbf{A}\in\mathbb{R}^{N\times K}$ is randomly generated at each trial with columns being normalized to unit 2-norm. We use speech recordings as our source signals. We have a database of 23 speech signals, and at each trial we randomly pick K of them as the source signals. In order to obtain more local covariances under limited signal length, we employ 50% overlapping frames in acquiring $\mathbf{R}_m$'s, i.e., $\mathbf{R}_m=(1/L)\sum_{t=0.5(m-1)L+1}^{0.5(m-1)L+L}\mathbf{x}(t)\mathbf{x}(t)^H$. Noisy received signals are assumed. The noise covariance removal procedure described in Section 2.3 is applied to the estimated $\mathbf{R}_m$'s. For our proposed algorithms, we adopt a standard stopping criterion, specifically,

$$\min_{\substack{\|\mathbf{a}_1\|_2^2=1,\\ \mathbf{h}_1\in\mathcal{R}(\mathbf{U}_s)}}\left\{\|\mathbf{h}_1-\mathbf{a}_1^*\otimes\mathbf{a}_1\|_2^2+\left[\min_{\substack{\|\mathbf{a}_2\|_2^2=1,\\ \mathbf{h}_2\in\mathcal{H}_2(\mathbf{h}_1)}}\|\mathbf{h}_2-\mathbf{a}_2^*\otimes\mathbf{a}_2\|_2^2+\cdots+\left(\min_{\substack{\|\mathbf{a}_K\|_2^2=1,\\ \mathbf{h}_K\in\mathcal{H}_K(\mathbf{h}_1,\ldots,\mathbf{h}_{K-1})}}\|\mathbf{h}_K-\mathbf{a}_K^*\otimes\mathbf{a}_K\|_2^2\right)\right]\right\} \tag{32}$$

$|f^{(n)}-f^{(n-1)}| < \epsilon = 10^{-6}$, where $f^{(n)}$ is the objective value of the algorithm at the nth iteration.

The performance measure employed here is the average mean square error (MSE), defined as

$$\mathrm{MSE} = \min_{\substack{\pi\in\boldsymbol{\Pi},\\ c_1,\ldots,c_K\in\{\pm 1\}}} \frac{1}{K}\sum_{k=1}^{K}\left\| \frac{\mathbf{a}_k}{\|\mathbf{a}_k\|_2} - c_k\frac{\hat{\mathbf{a}}_{\pi(k)}}{\|\hat{\mathbf{a}}_{\pi(k)}\|_2}\right\|_2^2,$$

where $\boldsymbol{\Pi}$ is the set of all bijections $\boldsymbol{\pi}:\{1,\ldots,K\}\rightarrow\{1,\ldots,K\}$; $\mathbf{A}$ and $\hat{\mathbf{A}}$ are the true and estimated mixing matrices, respectively. The MSE performance results to be shown are averages of one thousand independent trials. All algorithms are run on a computer with i7 2.8 GHz CPU and 16 GB RAM, with all the codes written in MATLAB. The signal-to-noise ratio (SNR) is defined as $\mathrm{SNR} = ((1/T)\sum_{t=0}^{T-1}\mathrm{E}\{\|\mathbf{As}(t)\|_2^2\})/\mathrm{E}\{\|\mathbf{v}(t)\|_2^2\}$.

6.1. The overdetermined case

We first consider an overdetermined case where we set $(N,K)=(6,5)$ and $(M,L)=(399,200)$. PAPA (Algorithm 4) is used. The algorithms we benchmark against are FFDIAG [16], UWEDGE [18], BGWEDGE [18] and Pham's JD [14]. All the algorithms are run on the same set of noise covariance-removed and prewhitened $\mathbf{R}_m$'s. Fig. 3 shows the average MSEs of the various algorithms w.r.t. the SNRs. It is seen that PAPA provides the best MSE performance for SNR $\leq$30 dB. For SNR > 35 dB, PAPA is outperformed by FFDIAG and UWEDGE. Having said so, the MSE performance of PAPA is still quite on a par.

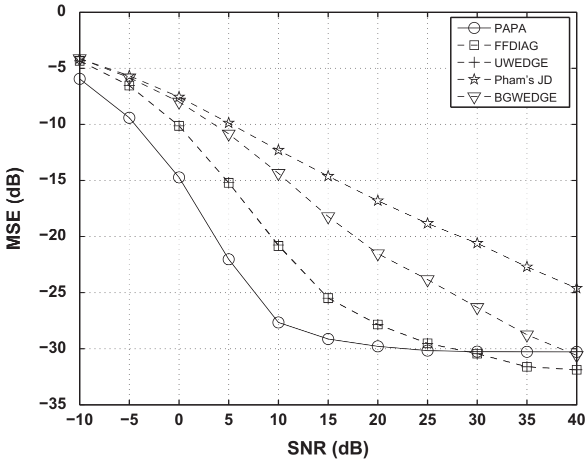

Fig. 3. The average MSEs of the various algorithms w.r.t. the SNRs.

Table 1
The average runtimes (in second) under various SNRs. $N=6$; $K=5$.

Method	SNR (dB)				
	−10	0	10	20	30
PAPA	**0.0055**	**0.0035**	**0.0026**	**0.0024**	**0.0023**
FFDIAG	0.4368	0.1561	0.0819	0.0649	0.0628
UWEDGE	0.5006	0.1318	0.0706	0.0600	0.0781
Pham's JD	8.7037	3.1116	2.3582	1.7417	1.4583
BGWEDGE	0.1315	0.1121	0.0826	0.0656	0.0615

Table 1 lists the average runtimes of the various algorithms corresponding to the above simulation. PAPA clearly demonstrates better computational efficiency compared to the other algorithms. In particular, PAPA is at least 23 times faster than FFDIAG, UWEDGE and BGWEDGE. The reason behind its high efficiency lies in the fact that the number of iterations required by PAPA is small. To get a better idea, in Table 2 we show the average numbers of AP iterations in PAPA. We can see that for SNR≥10 dB, the average numbers of AP iterations are around 3. This is consistent with the fast AP convergence claim in Theorem 2, which says convergence within one iteration, although one may wonder why the former and latter do not exactly collide. We should note at this point that Theorem 2 is established based on the satisfiability of the basic model laid in Section 2.2. In practice, the local covariances $\mathbf{R}_m$'s are subjected to measurement errors and the subsequent subspace perturbation effects may have an impact on the practical AP convergence. Notwithstanding, we see in Table 2 that the impact is insignificant for moderate to high SNRs.

Although the main interest in this paper lies in mixing matrix estimation, it is also interesting to look at source separation performance. Fig. 4 shows the source separation performance obtained by the various algorithms. The performance metric used here is the signal-to-interference-plus-noise ratio (SINR) of the separated signals, where the sources are separated via MMSE demixing; see [34] for details. It can be seen that all the algorithms exhibit similar SINR performance, except for SNR≥30 dB where BGWEDGE and Pham's JD have slightly better SINRs than the others.

More performance comparisons are shown in Figs. 5 and 6. The simulation settings are essentially the same as the previous, and we fix SNR=10 dB. Fig. 5 plots the MSEs w.r.t. the number of local covariances M. We can see that PAPA performs better than the other algorithms for $M\geq 200$, and the otherwise for $M<200$. This suggests that PAPA works better for larger numbers of local covariances. Fig. 6 plots the MSEs w.r.t. the number of sources K, with $N=K+1$. As seen, PAPA performs better for small to moderate numbers of sources, specifically, $K\leq 10$. Table 3 shows the corresponding runtime performance. PAPA remains computationally competitive, except for $K=23$ where BGWEDGE yields the fastest runtime.

6.2. The underdetermined case

Next, we consider an underdetermined case where $(N,K)=(5,7)$ and $(M,L)=(399,400)$. In order to avoid

Table 2
The average number of AP iterations of PAPA under various SNRs. $N=6$; $K=5$.

SNR (dB)	−10	0	10	20	30
Average number of AP iterations	8.5280	5.0336	3.3724	2.9250	2.8654

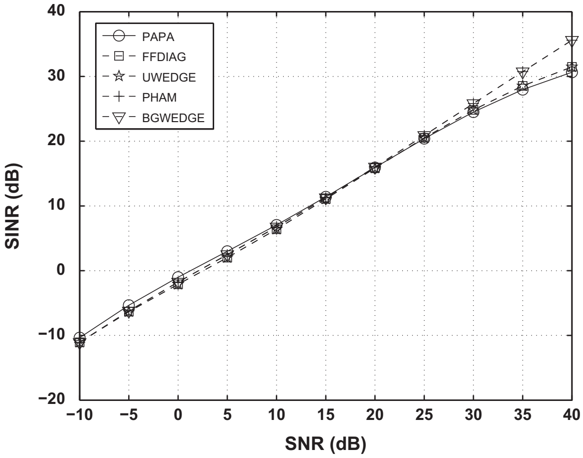

Fig. 4. The average SINRs of the various algorithms w.r.t. the SNRs.

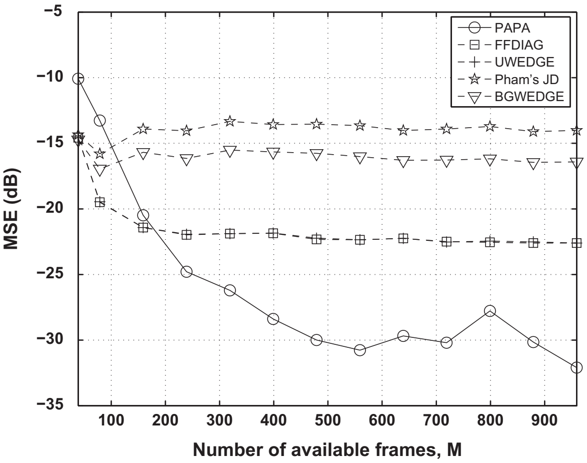

Fig. 5. The average MSEs of the various algorithms w.r.t. *M*.

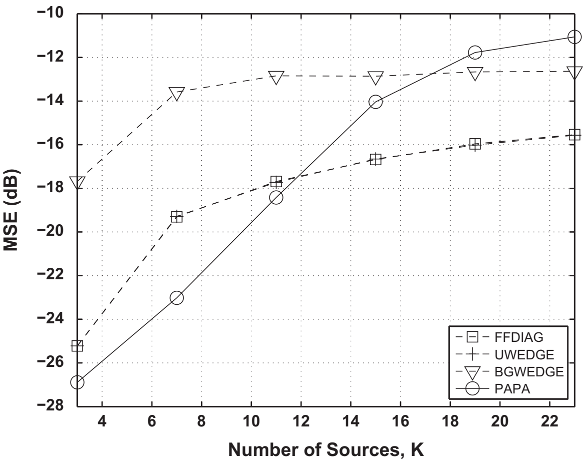

Fig. 6. The average MSEs of the various algorithms w.r.t. *K*.

Table 3
The average runtimes (in second) under various source numbers. $N=K+1$; SNR $=10$ dB.

Method	K					
	3	7	11	15	19	23
PAPA	**0.0072**	**0.0053**	**0.0195**	**0.0738**	**0.2321**	0.6134
FFDIAG	0.0230	0.0634	0.1243	0.2206	0.3889	0.5904
UWEDGE	0.0291	0.0896	0.1726	0.2828	0.5154	0.7799
BGWEDGE	0.0332	0.0819	0.1272	0.1863	0.2697	**0.3769**

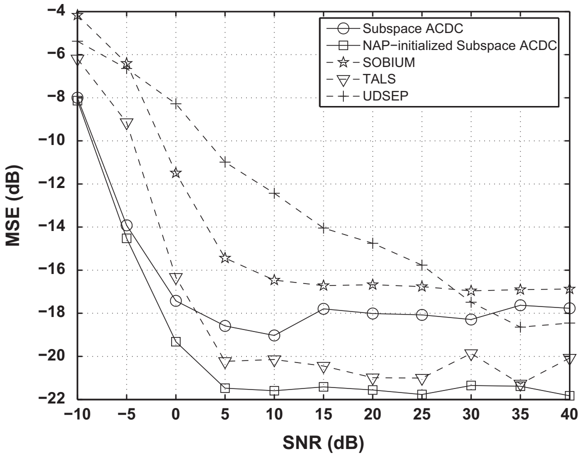

Fig. 7. The average MSEs of the various algorithms w.r.t. the SNRs.

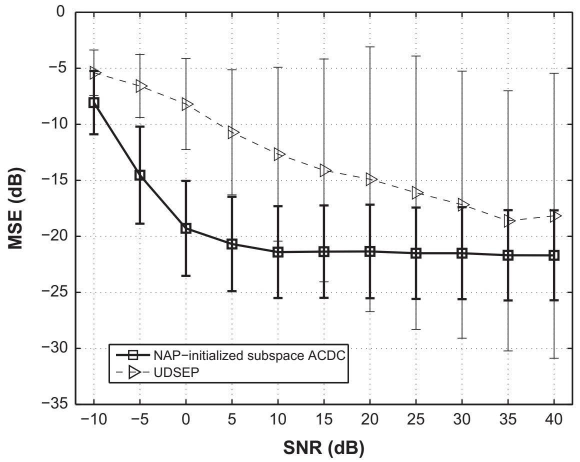

Fig. 8. The average MSEs and corresponding standard deviations of UDSEP and NAP-initialized ACDC.

overly ill-conditioned mixing matrices [12,17,35], we constrain the columns of $\mathbf{A}$ to satisfy $|\mathbf{a}_i^H\mathbf{a}_j|<0.8$, $\forall i\neq j$. The algorithms under comparison are NAP-initialized subspace ACDC (Algorithm 3), SOBIUM [8], TALS [6] and UDSEP [12]. We apply noise covariance-removed $\mathbf{R}_m$'s to all the algorithms, except for UDSEP. UDSEP was found to be numerically sensitive to noise covariance removal, and we run it using the original $\mathbf{R}_m$'s. For Algorithm 3, we set $\epsilon=0.8$ and $J=5\,K$, i.e., NAP is run at most 35 times. For NAP (Algorithm 2), we set $\gamma=0.5, \rho=1$ and we stop the ADMM when the number of ADMM iterations reaches 20. We also try subspace ACDC with completely random

Table 4
The average number of iterations and runtimes under various SNRs. $N=5$; $K=7$.

Method	SNR (dB)				
	−10	0	10	20	30
Subspace ACDC					
Iterations	741.177	292.356	264.536	292.216	282.969
Time (s)	1.5683	0.61939	0.56625	0.62739	0.60435
NAP-initialized subspace ACDC					
Iterations	689.237	134.197	91.737	88.403	89.201
Time (s)	1.4634	0.28498	0.19685	0.18833	0.18793
SOBIUM					
Iterations	**38.92**	**8.681**	**5.37**	**5.241**	**5.266**
Time (s)	**0.063175**	**0.027482**	**0.023618**	**0.023728**	**0.023509**
TALS					
Iterations	1889.775	633.107	479.733	473.49	485.238
Time (s)	39.4489	13.1247	9.97616	9.81866	10.0604
UDSEP					
Iterations	1000	1000	1000	1000	1000
Time (s)	159.7963	159.4327	159.2985	159.0518	158.6283

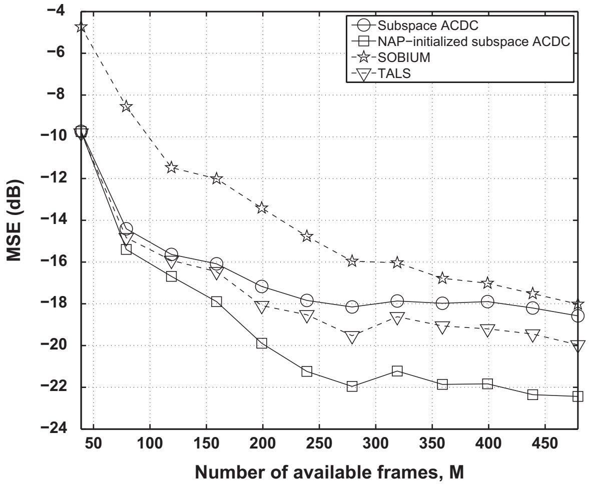

Fig. 9. The average MSEs of the various algorithms w.r.t. M.

initializations (by setting $J=0$), where the purpose is to examine the differences of using "good" and "bad" initializations. Fig. 7 shows the average MSEs of the various algorithms w.r.t. the SNRs. Remarkably, it can be seen that NAP-initialized subspace ACDC gives the best MSE performance. Another observation is that without the NAP initialization, subspace ACDC still works reasonably, but suffer from around 4 dB performance degradation compared to NAP-initialized subspace ACDC. Fig. 8 takes a closer look at the performance of UDSEP and NAP-ACDC by showing both the averages and standard deviations of the MSEs. As seen, UDSEP exhibits large variations with MSEs. This means that there are realizations where UDSEP performs much better than NAP-ACDC; however, there are realizations where UDSEP does not perform well.

Table 5
Performance of NAP under various SNR. $N=5$; $K=7$.

SNR (dB)	−10	0	10	20	30
Average number of AP iterations	16.5163	23.6907	27.3381	27.8929	28.0969
Time (s)	1.7136	1.2812	1.2364	1.2456	1.2437
Success rate (%)	4.3	63.5	73.8	73.5	74.1

Table 4 shows the average numbers of iterations and runtimes of the various algorithms corresponding to the above simulation. We are interested in examining the differences of using and not using NAP initialization, and hence the overheads of NAP are not counted at this point. It can be seen that with NAP initialization, the numbers of iterations required by subspace ACDC are reduced. For SNR≥10 dB, an iterations saving of about 2/3 can be observed for subspace ACDC. Moreover, SOBIUM is seen to yield the best runtime and iteration performance. Hence, we conclude that NAP-initialized subspace ACDC has its edge on MSE performance, but is more expensive to apply. To further confirm the performance advantage claim, we show the MSE performance of the various algorithms under different M in Fig. 9. We fix SNR = 10 dB. NAP-initialized subspace ACDC is seen to yield better performance once again.

We also look at a relatively detailed aspect: how does NAP perform? Table 5 shows the average numbers of AP iterations, the total runtimes spent by the multiple NAPs in NAP-initialized subspace ACDC, and the success rate for NAPs to identify a complete $\mathbf{A}$. The settings are identical to those in Table 4. We can see that NAP converges within 30 iterations, and that we have actually a success rate of 73% for SNR≥10 dB, which is quite promising. However, we should recognize that the computational times are relatively high—it is low compared to TALS, but high compared to SOBIUM; cf. Tables 4 and 5. There are two reasons. First, unlike PAPA, we need to run NAP many times. Second, NAP needs to solve a convex optimization problem at each AP iteration. While the ADMM solver we employ to tackle the problem is well known to be efficient in the context of low-rank matrix recovery, a curious question still lies in whether we can further reduce the complexity by devising more specialized algorithms for NAP. We leave this as a future direction.

7. Conclusion

In this paper, we have established a Khatri–Rao subspace framework for blind identification of mixtures of quasi-stationary sources. A particularly notable result lies in the overdetermined case, where we have developed a blind identification algorithm (PAPA, Algorithm 4) that can provide significant computational performance edge over the other algorithms especially for small to moderate numbers of sources. The algorithm also shows competitive estimation performance. Its computational advantage has also been supported by theoretical analysis. For the underdetermined case, we have developed another algorithm (NAP-initialized subspace ACDC, Algorithm 3) that yields good estimation performance by simulations. For both cases, the key insight lies in using the Khatri–Rao subspace to decouple the problem to column-wise BID problems, which is easier to manage.

Acknowledgements

This work was supported by a General Research Fund of Hong Kong Research Grant Council (CUHK415509).

Appendix A. Proof of Theorem 1

The proof of sufficiency is by contradiction. Suppose that there exists $\mathbf{a}\in\mathbb{C}^N$, $\mathbf{a}\neq c\mathbf{a}_k$ for any $c\neq 0$ and $k\in\{1,\ldots,K\}$, such that $\mathbf{a}^*\otimes\mathbf{a}\in\mathcal{R}(\mathbf{U}_s)$. By the KR subspace identity (9), the condition $\mathbf{a}^*\otimes\mathbf{a}\in\mathcal{R}(\mathbf{U}_s)$ is equivalent to

$$\mathbf{a}^*\otimes\mathbf{a}=\sum_{k=1}^{K}\alpha_k\mathbf{a}_k^*\otimes\mathbf{a}_k, \tag{36}$$

for some $\boldsymbol{\alpha}=[\alpha_1,\ldots,\alpha_K]\in\mathbb{C}^K$, $\boldsymbol{\alpha}\neq\mathbf{0}$. We will show that (36) does not hold whenever $K\leq 2N-2$. Let $P\leq K$ be the number of nonzero elements in $\boldsymbol{\alpha}$, and assume without loss of generality that $\alpha_k\neq 0$ for $k=1,\ldots,P$, and $\alpha_k=0$ for $k=P+1,\ldots,K$. Moreover, by denoting

$$\mathbf{A}_1=[\mathbf{a}_1,\ldots,\mathbf{a}_{\min\{P,N\}}]\in\mathbb{C}^{N\times\min\{P,N\}},$$
$$\mathbf{A}_2=[\mathbf{a}_{\min\{P,N\}+1},\ldots,\mathbf{a}_P,\mathbf{a}]\in\mathbb{C}^{N\times(P-\min\{P,N\}+1)},$$
$$\mathbf{D}_1=\mathrm{Diag}(\alpha_1,\ldots,\alpha_{\min\{P,N\}}),$$
$$\mathbf{D}_2=\mathrm{Diag}(-\alpha_{\min\{P,N\}+1},\ldots,-\alpha_P,1),$$

and by devectorization, we can rewrite (36) as

$$\mathbf{A}_2\mathbf{D}_2\mathbf{A}_2^H=\mathbf{A}_1\mathbf{D}_1\mathbf{A}_1^H. \tag{37}$$

Eq. (37) implies that

$$\mathrm{rank}(\mathbf{A}_2\mathbf{D}_2\mathbf{A}_2^H)=\mathrm{rank}(\mathbf{A}_1\mathbf{D}_1\mathbf{A}_1^H). \tag{38}$$

For convenience, let $r_1=\mathrm{rank}(\mathbf{A}_1\mathbf{D}_1\mathbf{A}_1^H)$, $r_2=\mathrm{rank}(\mathbf{A}_2\mathbf{D}_2\mathbf{A}_2^H)$. The matrix $\mathbf{A}_1$ has full column rank, as a direct consequence of (A4). In addition, using the fact that all the diagonal elements of $\mathbf{D}_1$ are nonzero, one can easily deduce that $r_1=\mathrm{rank}(\mathbf{A}_1)=\min\{P,N\}$. Now, let us consider two cases for r_2, namely, $P\leq N$ and $N<P\leq 2N-2$. For $P\leq N$, where $\mathbf{A}_2=\mathbf{a}$, we have $r_2=1$. The equality $r_1=r_2$ does not hold except for $P=1$, which reduces to the trivial case of $\mathbf{a}=c\mathbf{a}_k$. For $N<P\leq 2N-2$, it can be verified that $\mathbf{A}_2$ is a strictly tall matrix. Hence, we have $r_2\leq N-1$, and $r_1=r_2$ cannot be satisfied. The violation of $r_1=r_2$ in the above two cases contradicts (36).

The proof of necessity is done by finding an $\mathbf{A}$ such that $K>2N-2$ and (36) holds. Consider a Vandemonde $\mathbf{A}$ where each column $\mathbf{a}_k$ takes the form

$$\mathbf{a}_k=[1,e^{j\theta_k},\ldots,e^{j\theta_k(N-1)}]^T\triangleq\mathbf{b}(\theta_k), \tag{39}$$

for which the angles $\theta_k\in[0,2\pi)$ satisfy $\theta_k\neq\theta_\ell$ for all $k\neq\ell$. Such an $\mathbf{A}$ is always of full Kruskal rank [36], thereby satisfying the premise (A4). Also, suppose that $\mathbf{a}$ takes the form $\mathbf{a}=\mathbf{b}(\psi)$ for some ψ, and that $P=2N-1$ (thus $K>2N-2$). Now, if we choose

$$\theta_k=\begin{cases}\dfrac{2\pi(k-1)}{N}, & k=1,\ldots,N,\\[2mm] \dfrac{2\pi(k-N-1)}{N}+\dfrac{\pi}{N}, & k=N+1,\ldots,2N-1\end{cases}$$
$$\psi=\frac{2\pi(N-1)}{N}+\frac{\pi}{N},$$

then it can be verified that $\mathbf{A}_1$ and $\mathbf{A}_2$ are both unitary. Subsequently, by setting $\alpha_1=\cdots=\alpha_N=1$, $\alpha_{N+1}=\cdots=\alpha_{2N-1}=-1$, we get both sides of (37) being equal to $\mathbf{I}$. This in turn means that (36) can be satisfied.

Appendix B. ADMM for problem (23)

In order to apply ADMM, we rewrite problem (23) as

$$\min_{\mathrm{vec}(\mathbf{H})\in\mathcal{R}(\mathbf{U}_s),\mathbf{G}}\ \|\mathbf{B}-\mathbf{H}\|_F^2+\gamma\|\mathbf{G}\|_*$$
$$\text{s.t.}\quad \mathbf{H}=\mathbf{G}, \tag{40}$$

where we denote $\mathbf{B}\triangleq\alpha\mathbf{a}\mathbf{a}^H$ for convenience, and $\mathbf{G}$ is a splitting variable. According to the ADMM literature (e.g., [32]), the augmented Lagrangian of (40) is

$$\tilde{\mathcal{L}}(\mathbf{H},\mathbf{G},\boldsymbol{\Lambda})=\|\mathbf{B}-\mathbf{H}\|_F^2+\gamma\|\mathbf{G}\|_*$$
$$+\mathrm{Re}\{\boldsymbol{\Lambda}^H(\mathbf{H}-\mathbf{G})\}+\frac{\rho}{2}\|\mathbf{H}-\mathbf{G}\|_F^2, \tag{41}$$

where $\boldsymbol{\Lambda}$ is the dual variable for the equality constraint $\mathbf{H}=\mathbf{G}$, and ρ is a penalty parameter for the augmented term. Eq. (41) can be reexpressed as

$$\tilde{\mathcal{L}}(\mathbf{H},\mathbf{G},\mathbf{Z})=\|\mathbf{B}-\mathbf{H}\|_F^2+\gamma\|\mathbf{G}\|_*+\frac{\rho}{2}\|\mathbf{H}-\mathbf{G}$$
$$+\mathbf{Z}\|_F^2-\frac{\rho}{2}\|\mathbf{Z}\|_F^2, \tag{42}$$

where $\mathbf{Z}\triangleq(1/\rho)\boldsymbol{\Lambda}$. The idea of ADMM is to optimize (42) using alternating optimization and gradient ascent; in essence, we alternatingly solve the following two minimization problems:

$$\mathbf{H}^{(k+1)}:=\arg\min_{\mathrm{vec}(\mathbf{H})\in\mathcal{R}(\mathbf{U}_s)}\tilde{\mathcal{L}}(\mathbf{H},\mathbf{G}^{(k)},\mathbf{Z}^{(k)}), \tag{43}$$

$$\mathbf{G}^{(k+1)}:=\arg\min_{\mathbf{G}}\tilde{\mathcal{L}}(\mathbf{H}^{(k+1)},\mathbf{G},\mathbf{Z}^{(k)}), \tag{44}$$

together with the update of the (scaled) dual variable $\mathbf{Z}$ by

$$\mathbf{Z}^{(k+1)}:=\mathbf{Z}^{(k)}+(\mathbf{H}^{(k+1)}-\mathbf{G}^{(k+1)}). \tag{45}$$

There are many convergence results established for ADMM. For example, in [32], it has been shown that $\mathbf{H}^{(k)}-\mathbf{G}^{(k)}\to\mathbf{0}$, $\|\mathbf{B}-\mathbf{H}^{(k)}\|_F^2+\gamma\|\mathbf{G}^{(k)}\|_*\to p^\star,\mathbf{Z}^{(k)}\to\mathbf{Z}^\star$ as $k\to\infty$, where $p^\star$ is the optimal objective value of (40). The solutions of (43) and (44) are as follows. We note that (43) is an unconstrained least-squares problem (the subspace constraint can be eliminated by substituting $\mathbf{H}=\text{vec}^{-1}(\mathbf{U}_s\mathbf{x}), \mathbf{x}\in\mathbb{C}^K$), whose solution is given by

$$\mathbf{H}^{(k+1)}:=\frac{1}{\rho+2}\text{vec}^{-1}\{\mathbf{U}_s\mathbf{U}_s^H\ \text{vec}(2\mathbf{B}+\rho(\mathbf{G}^{(k)}-\mathbf{Z}^{(k)}))\}. \tag{46}$$

Moreover, problem (44), which is given by

$$\min_{\mathbf{G}}\ \gamma\|\mathbf{G}^{(k)}\|_*+\frac{\rho}{2}\|\mathbf{H}^{(k+1)}-\mathbf{G}+\mathbf{Z}^{(k)}\|_F^2, \tag{47}$$

is a proximal minimization problem. This problem arises frequently in low-rank matrix recovery [28,29] and its solution is well known to be that of singular value thresholding (SVT) [37], i.e.

$$\mathbf{G}^{(k+1)}=\mathbf{U}\text{Diag}(\mathbf{d})\mathbf{V}^H, \tag{48}$$

where we have the SVD $\mathbf{H}^{(k+1)}+\mathbf{Z}^{(k)}=\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^H$, and $d_i=\max\{0,\Sigma_{ii}-\gamma/\rho\}, i=1,\ldots,K$.

References

[1] K.-K. Lee, W.-K. Ma, Y.-L. Chiou, T.-H. Chan, C.-Y. Chi, Blind identification of mixtures of quasi-stationary sources using a Khatri–Rao subspace approach, in: Proceedings of the Asilomar Conference on Signals, Systems, and Computers, November 2011, pp. 2169–2173.

[2] L. Parra, C. Spence, Convolutive blind separation of non-stationary sources, IEEE Transactions on Speech and Audio Processing 8 (May (3)) (2000) 320–327.

[3] K. Rahbar, J. Reilly, A frequency domain method for blind source separation of convolutive audio mixtures, IEEE Transactions on Speech and Audio Processing 13 (September (5)) (2005) 832–844.

[4] D. Nion, K.N. Mokios, N.D. Sidiropoulos, A. Potamianos, Batch and adaptive PARAFAC-based blind separation of convolutive speech mixtures, IEEE Transactions on Audio, Speech, and Language Processing 18 (August (6)) (2010) 1193–1207.

[5] Z. Koldovský, P. Tichavský, Time-domain blind separation of audio sources on the basis of a complete ICA decomposition of an observation space, IEEE Transactions on Audio, Speech and Language Processing 19 (February (2)) (2011) 406–416.

[6] N.D. Sidiropoulos, R. Bro, G.B. Giannakis, Parallel factor analysis in sensor array processing, IEEE Transactions on Signal Processing 48 (August (8)) (2000) 2377–2388.

[7] A. Stegeman, J. Berge, L. Lathauwer, Sufficient conditions for uniqueness in Candecomp/Parafac and Indscal with random component matrices, Psychometrika 71 (2006) 219–229.

[8] L.D. Lathauwer, J. Castaing, Blind identification of underdetermined mixtures by simultaneous matrix diagonalization, IEEE Transactions on Signal Processing 56 (March (3)) (2008) 1096–1105.

[9] Y. Rong, S.A. Vorobyov, A.B. Gershman, N.D. Sidiropoulos, Blind spatial signature estimation via time-varying user power loading and parallel factor analysis, IEEE Transactions on Signal Processing 53 (May (5)) (2005) 1697–1710.

[10] A. Yeredor, Non-orthogonal joint diagonalization in the least-squares sense with application in blind source separation, IEEE Transactions on Signal Processing 50 (July (7)) (2002) 1545–1553.

[11] A.-J. van der Veen, Joint diagonalization via subspace fitting techniques, in: Proceedings of the ICASSP, vol. 5, May 2001, pp. 2773–2776.

[12] P. Tichavský, Z. Koldovský, Weight adjusted tensor method for blind separation of underdetermined mixtures of nonstationary sources, IEEE Transactions on Signal Processing 59 (March (3)) (2011) 1037–1047.

[13] G. Chabriel, J. Barrere, A direct algorithm for nonorthogonal approximate joint diagonalization, IEEE Transactions on Signal Processing 60 (1) (2012) 39–47.

[14] D.-T. Pham, J.-F. Cardoso, Blind separation of instantaneous mixtures of nonstationary sources, IEEE Transactions on Signal Processing 49 (September (9)) (2001) 1837–1848.

[15] R. Vollgraf, K. Obermayer, Quadratic optimization for simultaneous matrix diagonalization, IEEE Transactions on Signal Processing 54 (September (9)) (2006) 3270–3278.

[16] A. Ziehe, P. Laskov, G. Nolte, K.-R. Miller, A fast algorithm for joint diagonalization with non-orthogonal transformations and its application to blind source separation, Journal of Machine Learning Research 555 (2004) 777–800.

[17] X.-L. Li, X.-D. Zhang, Nonorthogonal joint diagonalization free of degenerate solution, IEEE Transactions on Signal Processing 55 (May (5)) (2007) 1803–1814.

[18] P. Tichavský, A. Yeredor, Fast approximate joint diagonalization incorporating weight matrices, IEEE Transactions on Signal Processing 57 (March (3)) (2009) 878–891.

[19] W.-K. Ma, T.-H. Hsieh, C.-Y. Chi, DOA estimation of quasi-stationary signals with less sensors than sources and unknown spatial noise covariance: a Khatri–Rao subspace approach, IEEE Transactions on Signal Processing 58 (April (4)) (2010) 2168–2180.

[20] P. Pal, P.P. Vaidyanathan, Nested arrays: a novel approach to array processing with enhanced degrees of freedom, IEEE Transactions on Signal Processing 58 (August (8)) (2010) 4167–4181.

[21] P. Stoica, R. Moses, Spectral Analysis of Signals, Pearson Prentice Hall, 2005.

[22] A. Belouchrani, K. Abed-Meraim, J.-F. Cardoso, E. Moulines, A blind source separation technique using second-order statistics, IEEE Transactions on Signal Processing 45 (February (2)) (1997) 434–444.

[23] P. Common, C. Jutten, Handbook of Blind Source Separation, Elsevier, 2010.

[24] H.L.V. Trees, Optimum Array Processing (Detection, Estimation, and Modulation Theory, Part IV), Wiley, 2002.

[25] M. Viberg, B. Ottersten, Sensor array processing based on subspace fitting, IEEE Transactions on Signal Processing 39 (May (5)) (1991) 1110–1121.

[26] N.D. Sidiropoulos, R. Bro, On the uniqueness of multilinear decomposition of N-way arrays, Journal of Chemometrics 14 (3) (2000) 229–239.

[27] S. Boyd, J. Dattoro, Alternating Projections. EE392o: Optimization Projects, Stanford University, 2003 [Online]. Available from < http://www.stanford.edu/class/ee392o/alt_proj.pdf > .

[28] E. Candés, X. Li, Y. Ma, J. Wright, Robust principal component analysis? Journal of the ACM 58 (May (3)) (2011).

[29] E. Candés, B. Recht, Exact matrix completion via convex optimization, Foundations of Computational Mathematics 9 (6) (2009) 717–772.

[30] B. Recht, M. Fazel, P.A. Parrilo, Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization, SIAM Review 52 (2010) 471–501.

[31] M. Grant, S. Boyd, CVX: MATLAB Software for Disciplined Convex Programming, Version 1.21, 〈http://cvxr.com/cvx/〉, April 2011.

[32] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers, Foundations and Trends in Machine Learning (2011) 1–122.

[33] K. Nordhausen, P. Ilmonen, A. Mandal, H. Oja, E. Ollila, Deflation-based FastICA reloaded, in: Proceedings of 19th European Signal Processing Conference, 2011, pp. 1854–1858.

[34] Z. Koldovský, P. Tichavský, Methods of fair comparison of performance of linear ICA techniques in presence of additive noise, in: Proceedings of the ICASSP, vol. 5, May 2006.

[35] M.R. DeYoung, B.L. Evans, Blind source separation with a time-varying mixing matrix, in: Proceedings of the Asilomar Conference on Signals, Systems, and Computers, November 2007.

[36] N.D. Sidiropoulos, X.-Q. Liu, Identifiability results for blind beamforming in incoherent multipath with small delay spread, IEEE Transactions on Signal Processing 49 (January (1)) (2001) 228–236.

[37] J.-F. Cai, E. Candés, Z. Shen, A singular value thresholding algorithm for matrix completion, SIAM Journal on Optimization 20 (January (4)) (2010) 1956–1982.