IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 13, 2018, accepted June 16, 2018, date of publication June 26, 2018, date of current version July 30, 2018.

*Digital Object Identifier 10.1109/ACCESS.2018.2850784*

# A Jacobi Generalized Orthogonal Joint Diagonalization Algorithm for Joint Blind Source Separation

**XIAO-FENG GONG[1], (Member, IEEE), LEI MAO[1], YING-LIANG LIU[2], AND QIU-HUA LIN[1], (Member, IEEE)**

[1]School of Information and Communication Engineering, Dalian University of Technology, Dalian 116023, China
[2]Qingdao Aerospace Semiconductor Research Institute Company Ltd., Qingdao 266000, China

Corresponding author: Xiao-Feng Gong (xfgong@dlut.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61671106, Grant 61331019, and Grant 61379012, in part by the Scientific Research Fund of Liaoning Education Department under Grant L2014016, and in part by the Fundamental Research Funds for the Central Universities under Grant DUT16QY07.

**ABSTRACT** Joint blind source separation (J-BSS) has emerged as a data-driven technique for multi-set data fusion applications. In this paper, we propose a Jacobi generalized orthogonal joint diagonalization (GOJD) algorithm for J-BSS of multiset signals. By the use of second-order statistics, we can obtain multiple sets of auto-covariance and cross-covariance matrices from the multi-set signals, which together admit a GOJD formulation. For computing the GOJD, we propose a computationally efficient Jacobi algorithm, which uses a sequence of Givens rotations to simultaneously diagonalize the covariance matrices. In comparison with other GOJD algorithms, the proposed algorithm is shown to have fast convergence. Moreover, as the optimal Givens rotation matrix in each update is calculated in closed-form, this algorithm is computationally very efficient. In the application aspect, we have considered the scenario where different data sets in J-BSS may have different number of sources, among which there exist both similar components that are consistently present in multiple data sets, and diverse components that are uniquely present in each data set. We have shown how J-BSS based on the proposed GOJD algorithm can effectively extract both similar and diverse source components. Simulation results are given to show the nice performance of the proposed algorithm, with regards to both speed and accuracy, in comparison with other algorithms of similar type.

**INDEX TERMS** Joint blind source separation, joint diagonalization, Jacobi, Givens rotation.

## I. INTRODUCTION

Recently, joint blind source separation (J-BSS) for multi-set data fusion has attracted much attention in a large variety of applications such as joint EEG-fMRI processing [1], [2], multi-subject fMRI data analysis [3]–[5], speech and array processing [6], [7], etc. In all these applications, the acquired multiple datasets have some inherent similarity, i.e. there exist similar source components that are consistently present in distinct datasets. There may also be a diversity for distinct datasets in J-BSS, i.e. each dataset may have its own specific property or component that other datasets do not have. As such, the goal of J-BSS is to jointly identify the multi-set source components and loading matrices by exploiting the similarity between distinct datasets and the diversity of each dataset.

Much effort has been made on extending classical BSS approaches to J-BSS. To give a non-exhaustive list of examples, we here mention independent vector analysis (IVA) [4], [6], [12], multi-set canonical correlation analysis (MCCA) [3], [13]–[15], generalized joint diagonalization (GJD) [16]–[20], coupled and double coupled canonical polyadic decomposition [21]–[26], which represent the multi-set extensions of independent component analysis (ICA), canonical correlation analysis (CCA), joint diagonalization (JD) and canonical polyadic decomposition (CPD), respectively. For an overview of J-BSS we refer to [28].

Among the above J-BSS techniques, GJD is formulated as the joint diagonalization of multiple sets of data matrices $\{\boldsymbol{C}_k^{(m,n)}, k = 1, \ldots, K\}, m, n = 1, \ldots, M$, which can be written as:

$$\boldsymbol{C}_k^{(m,n)} = \boldsymbol{B}^{(m)} \boldsymbol{\Lambda}_k^{(m,n)} \boldsymbol{B}^{(n)H}, \tag{1}$$

where $\boldsymbol{B}^{(m)}, \boldsymbol{B}^{(n)} \in \mathbb{C}^{N \times N}$ and $\boldsymbol{\Lambda}_k^{(m,n)} \in \mathbb{C}^{N \times N}$ are the $m$th and $n$th loading matrices and a diagonal matrix,

respectively. A GJD is said to be a generalized orthogonal JD (GOJD) if the loading matrices $\boldsymbol{B}^{(1)}, \ldots, \boldsymbol{B}^{(M)}$ are unitary matrices [16], [17]. The joint singular value decomposition (JSVD) [19] can be considered as a simple GOJD method, which jointly diagonalizes the cross-covariance matrices between two datasets. On the other hand, the generalized non-orthogonal JD (GNJD) is defined as GJD with non-orthogonal loading matrices. The data matrices (1) that admit a GJD can be obtained via fourth-order or second-order statistics, as addressed in [16]. Several GOJD and GNJD algorithms have been proposed and applied to various practical problems [16], [17], [20]. In the above-mentioned GJD works, it is generally assumed that all the datasets have identical number of sources, and that the sources with the same index between distinct datasets are similar.

In this paper, we mainly focus on J-BSS based on GOJD. The main contributions are summarized as follows:

(i) In the formulation of the GOJD model, we consider the scenario where different datasets may have different number of sources, and where not all the sources in each dataset have a corresponding ''similar'' component in other datasets, i.e., there may exist specific source components that are uniquely present in each dataset. Correspondingly, we introduce approaches to extract source signals of each dataset, and similar source signals between each pair of datasets.

(ii) We propose a Jacobi GOJD algorithm via successive Givens rotations. This algorithm has nice linear convergence and low complexity in comparison with other existing GOJD algorithms [16], [17]. In comparison with the Jacobi JSVD algorithm [19], which diagonalizes the cross-covariance matrices between two datasets, the proposed algorithm works on more than just two datasets and incorporates both auto-covariance and cross-covariance terms. As will be shown later, the incorporation of both auto-terms and cross-terms gives the proposed GOJD algorithm advantage over JSVD in extracting both similar components that are consistently present in distinct datasets and specific components that are uniquely present in each dataset.

The rest of the paper is organized as follows. In Section II we present the multi-set data model for J-BSS, and explain how to formulate a GOJD model via second-order statistics. In Section III we present the Jacobi GOJD algorithm based on Givens rotations. In Section IV, we introduce strategies for extraction of source signals for each dataset, similar source signals between each pair of datasets. Section V presents remarks on the proposed method and Section VI provides simulation results with comparison to several existing GOJD algorithms and J-BSS algorithms of similar type. Section VII concludes the paper.

*Notations*: vectors, matrices, tensors are denoted by lowercase boldface, uppercase boldface, and uppercase calligraphic letters, respectively. The $r$th column vector and the $(i, j)$th entry of $\boldsymbol{A}$ are denoted by $\boldsymbol{a}_r$ and $a_{i,j}$, respectively.

We denote the identity matrix and all-zero matrix as $\boldsymbol{I}_M \in \mathbb{C}^{M\times M}$ and $\boldsymbol{0}_{M\times N} \in \mathbb{C}^{M\times N}$, respectively. The subscripts are omitted if no ambiguity is caused. Transpose, conjugate, conjugated transpose, and Frobenius norm are denoted as $(\cdot)^T$, $(\cdot)^*$, $(\cdot)^H$, $\|\cdot\|_F^2$, respectively. Operator off$(\cdot)$ sets all the diagonal elements of its input matrix to zero, and diag$(\cdot)$ performs the inverse. Operators Re$(\cdot)$ and Im$(\cdot)$ extracts the real and imaginary parts of its input, respectively. Mathematical expectation is denoted as E$(\cdot)$.

## II. PROBLEM FORMULATION

### A. DATA MODEL AND ASSUMPTIONS

The multi-set data model in J-BSS can be written as follows:

$$\boldsymbol{x}^{(m)}(t) = \boldsymbol{A}^{(m)}\boldsymbol{s}^{(m)}(t), \quad m = 1, \ldots, M, \tag{2}$$

where $\boldsymbol{x}^{(m)}(t) \in \mathbb{C}^{N^{(m)}}$, $\boldsymbol{s}^{(m)}(t) \in \mathbb{C}^{R^{(m)}}$, $\boldsymbol{A}^{(m)} \in \mathbb{C}^{N^{(m)}\times R^{(m)}}$ denote the observed signal, source signal, and loading matrix of the $m$th dataset, respectively. The parameters $N^{(m)}$ and $R^{(m)}$ denote the number of observation channels and the number of source signals of the $m$th dataset, respectively.

We make the following assumptions:

(A1) Source signals $s_u^{(m)}(t)$ and $s_v^{(n)}(t)$ are independent for $u \neq v$, $u \in [1, R^{(m)}]$, $v \in [1, R^{(n)}]$, regardless of the value of $m, n$;

(A2) Sources signals $s_u^{(m)}(t)$ and $s_v^{(n)}(t)$ are dependent if $1 \leq u = v \leq R^{(m,n)} \leq R^{(m)}$, for all $m \neq n$. We note that $R^{(m,n)}$ may vary with different index pair $(m, n)$. For convenience, we here fix $R^{(m,n)} = R$. The derivation remains similar in general case where $R^{(m,n)}$ is different with different $(m, n)$.

(A3) The considered J-BSS problem is overdetermined. We further assume that $N^{(1)} = \cdots = N^{(M)} = N \geq R^{(m)}$, for $m = 1, \ldots, M$, as a preprocessing step (e.g. prewhitening) based on principle component analysis (PCA) can be performed to reduce the number of observation channels of each dataset to a common number $N$ in the overdetermined case.

(A4) The loading matrices $\boldsymbol{A}^{(1)}, \ldots, \boldsymbol{A}^{(M)}$ are unitary. This is a natural result of assumption (A1) and (A3) as the loading matrix of a prewhitened dataset is unitary.

We note that the assumption (A1) corresponds to the intra-set independence, which has been widely adopted in BSS and J-BSS. Assumption (A2) states that the sources with identical channel index in different datasets are dependent, and this inter-set dependence holds for the first $R$ sources of each dataset. Note that in our setting different datasets may have different number of source signals, and there may exist not only similar source signals across different datasets, but also specific source signals that are uniquely present in each dataset. We note that assumption (A2) takes into account both the similarity among multiple datasets and the diversity for each dataset. It generalizes the widely adopted inter-set dependence in GJD based J-BSS [16]–[20], which considers only the similarity.

### B. FORMULATION OF J-BSS INTO A GOJD

Let us now explain how to formulate a GOJD via second-order statistics. In addition to the assumptions in *Subsection II.A*, we further assume that the source signals

are temporally non-stationary. Then we can construct cross-covariance matrices $\boldsymbol{C}_k^{(m,n)} \in \mathbb{C}^{N\times N}$ as follows:

$$\boldsymbol{C}_k^{(m,n)}\mathrm{E}\{\boldsymbol{x}^{(m)}(k)\boldsymbol{x}^{(n)}(k)^H\} = \boldsymbol{A}^{(m)}\boldsymbol{\Sigma}_k^{(m,n)}\boldsymbol{A}^{(n)H}, \tag{3}$$

where $\boldsymbol{\Sigma}_k^{(m,n)}\mathrm{E}\{\boldsymbol{s}^{(m)}(k)\boldsymbol{s}^{(n)}(k)^H\} \in \mathbb{C}^{R^{(m)}\times R^{(n)}}$, $k = 1, \ldots, K$, and $K$ is the number of time frames for which the cross-covariance matrices are computed.

By definition, we know that $\boldsymbol{\Sigma}_k^{(m,n)}$ holds the cross-covariance coefficient $\mathrm{E}\{s_u^{(m)}(k)s_v^{(n)}(k)^H\}$ as its $(u, v)$th entry. According to assumption (A1) and (A2) we have the following results:

$(i)$ $\boldsymbol{\Sigma}_k^{(m,m)}$ is a $R^{(m)} \times R^{(m)}$ diagonal matrix where all the diagonal entries are non-zero;

$(ii)$ The upper-left $R \times R$ submatrix of $\boldsymbol{\Sigma}_k^{(m,n)}$ is diagonal with non-zero diagonal entries, $m \neq n$. The rest entries of $\boldsymbol{\Sigma}_k^{(m,n)}$ are all equal to zero.

We further denote the expanded loading matrix $\boldsymbol{B}^{(m)}$ and diagonal matrix $\boldsymbol{\Lambda}_k^{(m,n)}$ as follows:

$$\begin{cases} \boldsymbol{B}^{(m)} \triangleq [\boldsymbol{A}^{(m)}, \boldsymbol{A}^{\perp(m)}] \in \mathbb{C}^{N\times N}, \\ \boldsymbol{\Lambda}_k^{(m,n)} \triangleq \begin{bmatrix} \boldsymbol{\Sigma}_k^{(m,n)} & \boldsymbol{0}_{R^{(m)}\times(N-R^{(n)})} \\ \boldsymbol{0}_{(N-R^{(m)})\times R^{(n)}} & \boldsymbol{0}_{(N-R^{(m)})\times(N-R^{(n)})} \end{bmatrix}, \end{cases} \tag{4}$$

where $\boldsymbol{A}^{\perp(m)}$ is a unitary matrix of size $N \times (N - R^{(m)})$ that is orthogonal to $\boldsymbol{A}^{(m)}$. Then we can rewrite (3) as follows:

$$\boldsymbol{C}_k^{(m,n)} = \boldsymbol{B}^{(m)}\boldsymbol{\Lambda}_k^{(m,n)}\boldsymbol{B}^{(n)H}, \tag{5}$$

where $\boldsymbol{B}^{(m)}$ is an $N \times N$ unitary matrix and $\boldsymbol{\Lambda}_k^{(m,n)}$ is an $N \times N$ diagonal matrix (with zero diagonal entries). The set of matrices $\{\boldsymbol{C}_k^{(m,n)}\}$ together admits a GOJD model.

Note that by (3) we explicitly have $\boldsymbol{C}_k^{(m,n)} = \boldsymbol{C}_k^{(n,m)H}$. In case that this symmetry is not readily present, we are able to create it. More precisely, let us denote the matrices in an unsymmetric GOJD as $\boldsymbol{C'}_l^{(m,n)} = \boldsymbol{B}^{(m)}\boldsymbol{\Lambda'}_l^{(m,n)}\boldsymbol{B}^{(n)H}$, $l = 1, \ldots, L$, and construct matrices $\boldsymbol{C}_k^{(m,n)}$, $k = 1, ..., K$, with $K = 2L$, as:

$$\begin{cases} \boldsymbol{C}_l^{(m,n)} \triangleq \boldsymbol{C'}_{l(m,n)}, \boldsymbol{C}_{L+l}^{(m,n)} \triangleq \boldsymbol{C'}_l^{(n,m)H}, & m < n, \\ \boldsymbol{C}_l^{(m,n)} \triangleq \boldsymbol{C'}_l^{(n,m)H}, \boldsymbol{C}_{L+l}^{(m,n)} \triangleq \boldsymbol{C'}_l^{(m,n)}, & m > n, \\ \boldsymbol{C}_l^{(m,n)} \triangleq \mathrm{Re}(\boldsymbol{C'}_l^{(m,n)}), \boldsymbol{C}_{L+l}^{(m,n)} \triangleq \mathrm{Im}(\boldsymbol{C'}_l^{(n,m)}), & m = n. \end{cases} \tag{6}$$

Then the constructed matrices $\boldsymbol{C}_k^{(m,n)}$ have the property that $\boldsymbol{C}_k^{(m,n)} = \boldsymbol{C}_k^{(n,m)H}$.

We note that, by performing GOJD of all the matrices $\{\boldsymbol{C}_k^{(m,n)}\}$ with varying $m$, $n$, $k$, we do not directly identify $\boldsymbol{A}^{(m)}$ but an expanded loading matrices $\boldsymbol{B}^{(m)}$, which holds the columns of $\boldsymbol{A}^{(m)}$ as a subset of its columns. Therefore, to extract the source signals we need to separate the columns of $\boldsymbol{A}^{(m)}$ from $\boldsymbol{B}^{(m)}$.

In Section III, we propose a Jacobi GOJD algorithm based on successive Givens rotations. In Section IV, we introduce a source extraction scheme to extract the source signals of each dataset, with the estimates of $\boldsymbol{B}^{(m)}$.

## III. JACOBI GOJD ALGORITHM

In this section we present a Jacobi GOJD algorithm. In *Subsection III.A*, we introduce the criterion for the proposed algorithm. Then, we present the overall framework of the proposed algorithm in *Subsection III.B*, which aims at solving the optimization problem (10) via a series of optimized Givens rotations. In *Subsection C* we derive the closed-form formula for the calculation of the optimal Givens rotation matrices in each update. Finally, we present remarks on the proposed algorithm in *Subsection D*.

### A. CRITERION

To compute a GOJD, we minimize the sum of squared norm of the off-diagonal entries of $\boldsymbol{B}^{(m)}\boldsymbol{C}_k^{(m,n)}\boldsymbol{B}^{(n)H}$ as follows:

$$\left\{\tilde{\boldsymbol{B}}^{(1)}, \ldots, \tilde{\boldsymbol{B}}^{(M)}\right\} = \underset{\boldsymbol{B}^{(1)},\ldots,\boldsymbol{B}^{(M)}}{\arg\min}\ f, \tag{7}$$

where

$$f \triangleq \sum_{m,n,k} \left\|\mathrm{off}\left(\boldsymbol{B}^{(m)}\boldsymbol{C}_k^{(m,n)}\boldsymbol{B}^{(n)H}\right)\right\|_F^2. \tag{8}$$

Since matrices $\boldsymbol{B}^{(m)}$ are unitary, we can rewrite (8) as:

$$\begin{aligned} f &= \sum_{m,n,k} \left\|\boldsymbol{B}^{(m)}\boldsymbol{C}_k^{(m,n)}\boldsymbol{B}^{(n)H}\right\|_F^2 \\ &\quad - \sum_{m,n,k} \left\|\mathrm{diag}\left(\boldsymbol{B}^{(m)}\boldsymbol{C}_k^{(m,n)}\boldsymbol{B}^{(n)H}\right)\right\|_F^2 \\ &= \sum_{m,n,k} \left\|\boldsymbol{C}_k^{(m,n)}\right\|_F^2 - \sum_{m,n,k} \left\|\mathrm{diag}\left(\boldsymbol{B}^{(m)}\boldsymbol{C}_k^{(m,n)}\boldsymbol{B}^{(n)H}\right)\right\|_F^2, \end{aligned} \tag{9}$$

where $\sum \left\|\boldsymbol{C}_k^{(m,n)}\right\|_F^2$ is a constant. Therefore, the off-norm minimization criterion (7) is equivalent to the following criterion that maximizes the sum of squared norm of the diagonal entries of $\boldsymbol{B}^{(m)}\boldsymbol{C}_k^{(m,n)}\boldsymbol{B}^{(n)H}$:

$$\left\{\tilde{\boldsymbol{B}}^{(1)}, \ldots, \tilde{\boldsymbol{B}}^{(M)}\right\} = \underset{\boldsymbol{B}^{(1)},\ldots,\boldsymbol{B}^{(M)}}{\arg\max} \sum_{m,n,k} \left\|\mathrm{diag}\left(\boldsymbol{B}^{(m)}\boldsymbol{C}_k^{(m,n)}\boldsymbol{B}^{(n)H}\right)\right\|_F^2. \tag{10}$$

### B. OVERALL APPROACH

We write $\boldsymbol{B}^{(m)}$ as the successive product of a set of Givens rotation matrices $\boldsymbol{B}^{(m)} = \prod_{u<v} \boldsymbol{G}_{u,v}^{(m)}$, $m = 1, \ldots M$, where the Givens rotation matrix $\boldsymbol{G}_{u,v}^{(m)}$ with indices $(u, v)$ is equal to an identity matrix except for the following submatrix:

$$\begin{bmatrix} (\boldsymbol{G}_{u,v}^{(m)})_{u,u} & (\boldsymbol{G}_{u,v}^{(m)})_{u,v} \\ (\boldsymbol{G}_{u,v}^{(m)})_{v,u} & (\boldsymbol{G}_{u,v}^{(m)})_{v,v} \end{bmatrix} \triangleq \begin{bmatrix} c_{u,v}^{(m)} & s_{u,v}^{(m)} \\ -s_{u,v}^{(m)*} & c_{u,v}^{(m)} \end{bmatrix}, \tag{11}$$

with $c_{u,v}^{(m)} = \cos\theta_{u,v}^{(m)}$, $s_{u,v}^{(m)} = \exp(i\alpha_{u,v}^{(m)})\sin\theta_{u,v}^{(m)}$, $1 \leq u < v \leq N$.

We let the indices $(u, v)$ vary, and for each choice of $(u, v)$, we calculate the optimal estimates $\tilde{\boldsymbol{G}}_{u,v}^{(m)}$ of $\boldsymbol{G}_{u,v}^{(m)}$. Here we use an alternating updating scheme for the calculation of $\tilde{\boldsymbol{G}}_{u,v}^{(m)}$. This scheme contains $M$ steps. In the $m$th step, one Givens

rotation matrix $\tilde{\boldsymbol{G}}_{u,v}^{(m)}$ is calculated with the other Givens rotation matrices fixed, and $\boldsymbol{B}^{(m)}$ and $\boldsymbol{C}_k^{(m,n)}$ are updated by:

$$\boldsymbol{B}^{(m)} \leftarrow \boldsymbol{B}^{(m)}\tilde{\boldsymbol{G}}_{u,v}^{(m)}, \boldsymbol{C}_k^{(m,n)} \leftarrow \tilde{\boldsymbol{G}}_{u,v}^{(m)H}\boldsymbol{C}_k^{(m,n)}. \tag{12}$$

We note that, due to the symmetry of $\boldsymbol{C}_k^{(m,n)}$, there is no need to do the update $\boldsymbol{C}_k^{(n,m)} \leftarrow \boldsymbol{C}_k^{(n,m)}\tilde{\boldsymbol{G}}_{u,v}^{(m)}$. Instead, we directly let $\boldsymbol{C}_k^{(\mathrm{n},m)} \leftarrow \boldsymbol{C}_k^{(m,n)H}$ once $\boldsymbol{C}_k^{(m,n)}$ is updated.

Once all the Givens rotations matrices $\tilde{\boldsymbol{G}}_{u,v}^{(m)}$ are calculated, and all the matrices $\boldsymbol{B}^{(m)}$ and $\boldsymbol{C}_k^{(m,n)}$ are updated via (12), for a fixed index pair $(u, v)$. We move on to the next index pair. The updates for all the combinations of indices $(u, v)$ together constitute a sweep. We repeat the updates (12) for multiple sweeps until convergence.

### C. CALCULATION OF THE OPTIMAL GIVENS ROTATION MATRICES IN EACH UPDATE

Let us now explain how to calculate the optimal $\tilde{\boldsymbol{G}}_{u,v}^{(m)}$ in each update. Recall that each $\tilde{\boldsymbol{G}}_{u,v}^{(m)}$ is calculated with other Givens rotation matrices fixed. Then according to (10) $\tilde{\boldsymbol{G}}_{u,v}^{(m)}$ can be calculated by maximizing the overall squared norm of the diagonal entries of $\tilde{\boldsymbol{G}}_{u,v}^{(m)H}\boldsymbol{C}_k^{(m,n)}$. Here we do not consider the diagonal entries of $\boldsymbol{C}_k^{(n,m)}\tilde{\boldsymbol{G}}_{u,v}^{(m)}$ because they are the same as those of $\tilde{\boldsymbol{G}}_{u,v}^{(m)H}\boldsymbol{C}_k^{(m,n)}$, due to the symmetry of $\boldsymbol{C}_k^{(m,n)}$.

Note that the Givens rotation $\tilde{\boldsymbol{G}}_{u,v}^{(m)H}\boldsymbol{C}_k^{(m,n)}$ only changes the $u$th and $v$th rows of $\boldsymbol{C}_k^{(m,n)}$. That means, among the diagonal entries of $\boldsymbol{C}_k^{(m,n)}$, only the $u$th and $v$th diagonal entries have been changed. Therefore, $\tilde{\boldsymbol{G}}_{u,v}^{(m)}$ can actually be calculated by maximizing the sum of squared norms of these two entries for all the matrices $\tilde{\boldsymbol{G}}_{u,v}^{(m)H}\boldsymbol{C}_k^{(m,n)}$, with fixed $m$, $u$, $v$, and varying $n$ and $k$:

$$\tilde{\boldsymbol{G}}_{u,v}^{(m)} = \underset{\boldsymbol{G}_{u,v}^{(m)}}{\arg\max}\, \lambda_{u,v}^{(m)}, \tag{13}$$

where

$$\lambda_{u,v}^{(m)} \triangleq \sum_{n,k}\left(\left|(\boldsymbol{G}_{u,v}^{(m)H}\boldsymbol{C}_k^{(m,n)})_{u,u}\right|^2 + \left|(\boldsymbol{G}_{u,v}^{(m)H}\boldsymbol{C}_k^{(m,n)})_{v,v}\right|^2\right). \tag{14}$$

By definition, we have the following expressions:

$$\begin{aligned}(\boldsymbol{G}_{u,v}^{(m)H}\boldsymbol{C}_k^{(m,n)})_{u,u} &= c_{u,v}^{(m)}(\boldsymbol{C}_k^{(m,n)})_{u,u} - s_{u,v}^{(m)}(\boldsymbol{C}_k^{(m,n)})_{v,u},\\ (\boldsymbol{G}_{u,v}^{(m)H}\boldsymbol{C}_k^{(m,n)})_{v,v} &= c_{u,v}^{(m)}(\boldsymbol{C}_k^{(m,n)})_{v,v} + s_{u,v}^{(m)*}(\boldsymbol{C}_k^{(m,n)})_{u,v}.\end{aligned} \tag{15}$$

Substituting (15) into (14) yields the following formula after a few derivations:

$$\lambda_{u,v}^{(m)} = \boldsymbol{u}_{u,v}^{(m)T}\sum_{n,k}\left(\boldsymbol{m}_{u,v,k}^{(m,n)}\boldsymbol{m}_{u,v,k}^{(m,n)H} + \boldsymbol{m'}_{u,v,k}^{(m,n)}\boldsymbol{m'}_{u,v,k}^{(m,n)H}\right)\boldsymbol{u}_{u,v}^{(m)}, \tag{16}$$

where $\boldsymbol{u}_{u,v}^{(m)} \triangleq [c_{u,v}^{(m)}, \mathrm{Re}(s_{u,v}^{(m)}), \mathrm{Im}(s_{u,v}^{(m)})]^T$, and $\boldsymbol{m}_{u,v,k}^{(m,n)}$, $\boldsymbol{m'}_{u,v,k}^{(m,n)}$ are given by:

$$\boldsymbol{m}_{u,v,k}^{(m,n)} \triangleq \begin{bmatrix}(\boldsymbol{C}_k^{(m,n)})_{u,u}\\ -(\boldsymbol{C}_k^{(m,n)})_{v,u}\\ -i(\boldsymbol{C}_k^{(m,n)})_{v,u}\end{bmatrix}, \quad \boldsymbol{m'}_{u,v,k}^{(m,n)} \triangleq \begin{bmatrix}(\boldsymbol{C}_k^{(m,n)})_{v,v}\\ (\boldsymbol{C}_k^{(m,n)})_{u,v}\\ -i(\boldsymbol{C}_k^{(m,n)})_{u,v}\end{bmatrix}. \tag{17}$$

As such, the optimal $\boldsymbol{u}_{u,v}^{(m)}$ is chosen as the dominant eigenvector of $\sum_{n,k}(\boldsymbol{m}_{u,v,k}^{(m,n)}\boldsymbol{m}_{u,v,k}^{(m,n)H} + \boldsymbol{m'}_{u,v,k}^{(m,n)}\boldsymbol{m'}_{u,v,k}^{(m,n)H})$. The optimal parameters $c_{u,v}^{(m)}$ and $s_{u,v}^{(m)}$ can be computed from $\boldsymbol{u}_{u,v}^{(m)}$. Then the optimal $\tilde{\boldsymbol{G}}_{u,v}^{(m)}$ can be constructed with the obtained $c_{u,v}^{(m)}$ and $s_{u,v}^{(m)}$.

We summarize the proposed GOJD algorithm in TABLE I.

**TABLE 1. The Jacobi GOJD algorithm.**

**Input:** $\{\boldsymbol{C}_k^{(m,n)}, |1 \le m,n \le M, 1 \le k \le K\}$ admitting a GOJD (1).

1: Initialize the algorithm as addressed in *Remark 3*, *Subsection III.D*.
2: Do the following until convergence:
  **for** $1 \le u < v \le N$ **do**
    **for** $1 \le m \le M$ **do**
    - calculate the matrix $\sum_{n,k}(\boldsymbol{m}_{u,v,k}^{(m,n)}\boldsymbol{m}_{u,v,k}^{(m,n)H} + \boldsymbol{m'}_{u,v,k}^{(m,n)}\boldsymbol{m'}_{u,v,k}^{(m,n)H})$, where $\boldsymbol{m}_{u,v,k}^{(m,n)}, \boldsymbol{m'}_{u,v,k}^{(m,n)}$ are given by (17), and obtain the optimal Givens rotation matrix $\tilde{\boldsymbol{G}}_{u,v}^{(m)}$ from its dominant eigenvector $\boldsymbol{u}_{u,v}^{(m)}$.
    - update $\boldsymbol{B}^{(m)}, \boldsymbol{C}_k^{(m,n)}$ by (12).
    **end for**
  **end for**

**Output:** Estimates of $\boldsymbol{B}^{(m)}$, $1 \le m \le M$.

## IV. SOURCE EXTRACTION

By applying the proposed Jacobi GOJD algorithm on the covariance matrices (5), we can jointly identify the expanded loading matrices $\boldsymbol{B}^{(1)}, \ldots, \boldsymbol{B}^{(M)}$(4). To further extract the source signals of each dataset, we need to separate the columns that are associated with the true loading matrices $\boldsymbol{A}^{(1)}, \ldots, \boldsymbol{A}^{(M)}$ from the estimates of $\boldsymbol{B}^{(1)}, \ldots, \boldsymbol{B}^{(M)}$. In the noiseless case, we know by (4) that the columns of $\boldsymbol{A}^{(m)}$ are associated with non-zero diagonal entries of $\boldsymbol{\Lambda}_k^{(m,m)}, m = 1, \ldots, M$, and this knowledge can be used to identify the columns of $\tilde{\boldsymbol{A}}^{(m)}$ from the estimate of $\tilde{\boldsymbol{B}}^{(m)}$. E.g., we can select columns of $\tilde{\boldsymbol{B}}^{(m)}$, associated with diagonal entries of $\boldsymbol{\Lambda}_k^{(m,m)}$ that are significantly larger, as columns of $\tilde{\boldsymbol{A}}^{(m)}$. Here $\tilde{\boldsymbol{A}}^{(m)}$ and $\tilde{\boldsymbol{B}}^{(m)}$ denote the estimate of $\boldsymbol{A}^{(m)}$ and $\boldsymbol{B}^{(m)}$, respectively.

In practice, however, due to noise and model errors (e.g. the finite sample effects), it may be difficult to identify the diagonal entries of $\boldsymbol{\Lambda}_k^{(m,m)}$ that are associated with the columns of $\tilde{\boldsymbol{A}}^{(m)}$. Therefore, in this section we introduce subspace projection based methods for the extraction of (i) the source signals of each dataset, and (ii) similar source signals between each pair of datasets.

### A. SOURCE EXTRACTION FOR EACH DATASET

To extract source signals of each dataset, we calculate the overall covariance matrix of the observed signal in the $m$th dataset as $\boldsymbol{C}^{(m,m)} \triangleq \sum_k \boldsymbol{C}_k^{(m,m)}$. In the noiseless case, we note that the column space of $\boldsymbol{C}^{(m,m)}$ is identical to that of $\boldsymbol{A}^{(m)}$, $\Upsilon^{(m)}$, which is denoted as the signal subspace. In the noisy case, we estimate a set of basis vectors $\boldsymbol{u}_1^{(m)}, \ldots, \boldsymbol{u}_{R^{(m)}}^{(m)}$ of $\Upsilon^{(m)}$ as the $R^{(m)}$ dominant eigenvectors of $\boldsymbol{C}^{(m,m)}$, and construct a projection matrix $\boldsymbol{P}^{(m)} \triangleq \sum_r \boldsymbol{u}_r^{(m)}\boldsymbol{u}_r^{(m)H} \in \mathbb{C}^{N\times N}$.

Then we perform vector projection of each column of $\tilde{\boldsymbol{B}}^{(m)}$ into $\Upsilon^{(m)}$: $\tilde{\boldsymbol{b}}_r^{(m)} \rightarrow \boldsymbol{P}^{(m)}\tilde{\boldsymbol{b}}_r^{(m)} \in \Upsilon^{(m)}$. The $R^{(m)}$ column vectors of $\tilde{\boldsymbol{A}}^{(m)}$ can be chosen as those for which the norm of the projected vectors $||\boldsymbol{P}^{(m)}\tilde{\boldsymbol{b}}_r^{(m)}||_F$ is significantly larger.

By applying the above subspace projection approach to each dataset, we are able to identify $\tilde{\boldsymbol{A}}^{(m)}$ from the estimates of the expanded loading matrix $\tilde{\boldsymbol{B}}^{(m)}$. Finally, we obtain the source signals of each dataset as $\tilde{\boldsymbol{s}}^{(m)}(t) = \tilde{\boldsymbol{A}}^{(m)\dagger}\boldsymbol{x}^{(m)}(t)$.

### B. EXTRACTION OF SIMILAR SOURCES BETWEEN EACH PAIR OF DATASETS

In some applications, it is of interest to extract similar source signals between each pair of datasets. To achieve this goal, we construct subspace projection matrices from the overall cross-covariance matrix and then perform subspace projection.

For example, to extract the similar source signals between the $m$th and the $n$th datasets, $m \neq n$, we calculate the overall cross-covariance matrix $\boldsymbol{C}^{(m,n)} \triangleq \sum_k \boldsymbol{C}_k^{(m,n)}$. By (4) and (5) we know that in the noiseless case, the column space of $\boldsymbol{C}^{(m,n)}$ is identical to the vector space spanned by the $R$ columns of $\tilde{\boldsymbol{A}}^{(m)}$, which are associated with the source signals in the $m$th dataset that have a corresponding similar source signal in the $n$th dataset. Analogously, the row space of $\boldsymbol{C}^{(m,n)}$ is identical to the vector space spanned by the $R$ columns of $\tilde{\boldsymbol{A}}^{(m)}$ (up to a conjugation), which are associated with the source signals in the $n$th dataset that have a corresponding similar source signal in the $m$th dataset. For clarity, we denote these two subspaces as $\Upsilon^{(m|n)}$ and $\Upsilon^{(n|m)}$, respectively. In the noisy case, we can estimate the basis vectors of $\Upsilon^{(m|n)}$ and $\Upsilon^{(n|m)}$ as the $R$ dominant left and right singular vectors of $\boldsymbol{C}^{(m,n)}$, respectively.

Then for each of $\tilde{\boldsymbol{B}}^{(m)}$ and $\tilde{\boldsymbol{B}}^{(n)}$, we construct a projection matrix with the obtained basis vectors, perform subspace projection for each column into the corresponding subspace, and choose the $R$ columns for which the norm of the projected vectors is significantly larger. Note that the chosen columns constitute submatrices of $\tilde{\boldsymbol{A}}^{(m)}$ and $\tilde{\boldsymbol{A}}^{(n)}$, denoted as $\tilde{\boldsymbol{A}}^{(m|n)}$ and $\tilde{\boldsymbol{A}}^{(n|m)}$, respectively, that correspond to the similar source signals between these two datasets. Therefore, the similar source signals between the $m$th and $n$th datasets can finally be obtained as: $\tilde{\boldsymbol{s}}^{(m|n)}(t) = \tilde{\boldsymbol{A}}^{(m|n)H}\boldsymbol{x}^{(m)}(t)$, $\tilde{\boldsymbol{s}}^{(n|m)}(t) = \tilde{\boldsymbol{A}}^{(n|m)H}\boldsymbol{x}^{(n)}(t)$.

## V. DISCUSSION

In this section we provide several remarks on the proposed GOJD algorithm and the corresponding J-BSS method.

### A. STOPPING CRITERION FOR GOJD

In the proposed GOJD algorithm, the Jacobi iteration is terminated when the overall squared norm of the diagonal entries of $\boldsymbol{C}_k^{(m,n)}$ is not further improved. To put in more details, we denote the overall squared norm of the diagonal entries of $\boldsymbol{C}_k^{(m,n)}$ in the $s$th sweep as $\lambda_s$. The iteration is terminated if

$$\frac{|\lambda_s - \lambda_{s-1}|}{\lambda_{s-1}} \leq \xi, \tag{18}$$

where $\xi$ is the tolerance for the stopping criterion. We may also terminate the update if $c_{u,v}^{(m)}$ and $s_{u,v}^{(m)}$ defined in (11) are smaller than a given tolerance $\xi$, which indicates that the Givens rotation matrix is close to an identity matrix.

### B. COMPUTATIONAL COMPLEXITY OF GOJD

We present the computational cost of each sweep for the proposed GOJD algorithm. We note that each sweep consists of Givens rotations for $N(N-1)/2$ index pairs $(u, v)$, and for each index pair, we calculate $M$ optimal Givens rotation matrices. In the calculation of one Givens rotation matrix, the complexity of the construction of $\sum_{n,k}(\boldsymbol{m}_{u,v,k}^{(m,n)}\boldsymbol{m}_{u,v,k}^{(m,n)H} + \boldsymbol{m}'^{(m,n)}_{u,v,k}\boldsymbol{m}'^{(m,n)H}_{u,v,k})$ is $\mathcal{O}(18MK)$ flops. The EVD of this matrix has complexity of $6*3^3 = 162$ flops. Note that $\boldsymbol{B}^{(m)}\tilde{\boldsymbol{G}}_{u,v}^{(m)}$ only changes the $u$th and $v$th column of $\boldsymbol{B}^{(m)}$, and thus this step has complexity of $\mathcal{O}(4N)$ flops. Analogously, as $\tilde{\boldsymbol{G}}_{u,v}^{(m)H}\boldsymbol{D}_k^{(m,n)}$ only changes the $u$th and $v$th row of $\boldsymbol{D}_k^{(m,n)}$, therefore this step has complexity of $\mathcal{O}(4N)$ flops. As a result, the computational complexity of each sweep is $\mathcal{O}(MN(N-1)(9KM+4KN+81))$ flops. The expression can be simplified to $\mathcal{O}(9KM^2N^2 + 4KMN^3 + 81MN^2)$ flops.

With an analogous analysis, we have that the GOJD algorithm in [16] has complexity of $\mathcal{O}(KM^2N^2 + 12MN^3 + 4KM^2N^3)$ flops per sweep, and the complexity of the GOJD algorithm in [17] is $\mathcal{O}(1.5KM^2N^3 + 28MN^3)$ flops per sweep. We note that the per-sweep complexity of the proposed algorithm has a lower-order dependence on parameters $K$, $M$, and $N$ than the ones in [16] and [17]. In addition, we have observed through experiments that the proposed GOJD algorithm takes less sweeps until convergence than the ones in [16] and [17], if they are initialized with identity matrices. In general, we have observed that the proposed algorithm is computationally more efficient.

### C. INITIALIZATION OF GOJD

Prior knowledge of $\boldsymbol{B}^{(1)}, \ldots, \boldsymbol{B}^{(M)}$, if available, can be used to initialize the proposed GOJD algorithm. More precisely, denoting the prior estimates of $\boldsymbol{B}^{(1)}, \ldots, \boldsymbol{B}^{(M)}$ as $\boldsymbol{B}_0^{(1)}, \ldots, \boldsymbol{B}_0^{(M)}$, we can initially update the covariance data matrices as: $\boldsymbol{C}'^{(m,n)}_k \leftarrow \boldsymbol{B}_0^{(m)H}\boldsymbol{C}_k^{(m,n)}\boldsymbol{B}_0^{(n)}$. GOJD can then be performed on the initialized data matrices $\boldsymbol{C}'^{(m,n)}_k$ to obtain a set of matrices $\boldsymbol{B}'^{(1)}, \ldots, \boldsymbol{B}'^{(M)}$. Estimates of $\boldsymbol{B}^{(1)}, \ldots, \boldsymbol{B}^{(M)}$ can finally be obtained as: $\boldsymbol{B}^{(m)} = \boldsymbol{B}_0^{(m)}\boldsymbol{B}'^{(m)}$.

We note that the initial knowledge may be obtained via a low-cost J-BSS algorithm, e.g. MCCA, or via BSS of each separate dataset, e.g. joint approximate diagonalization of eigenmatrices (JADE). If no such information is available, we initialize the algorithm with identity matrices.

### D. CROSS-TERM VS. AUTO-TERM

In the proposed algorithm, GOJD jointly diagonalizes all the covariance matrices $\{\boldsymbol{C}_k^{(m,n)}, m,n = 1,\ldots,M, k = 1,\ldots,K\}$ to accomplish J-BSS of multi-set signals. We note that, with varying indices $m$ and $n$, there are in fact two types of terms in GOJD, which play different roles in J-BSS: (i) the cross-term $\{\boldsymbol{C}_k^{(m,n)}, m \neq n\}$ and (ii) the auto-term $\{\boldsymbol{C}_k^{(m,m)}\}$.

More specifically, in the expression of a cross-term $\boldsymbol{C}_k^{(m,n)}$, as shown in (3), only the first $R$ columns of $\boldsymbol{A}^{(m)}$ and $\boldsymbol{A}^{(n)}$ are associated with non-zero diagonal entries in $\boldsymbol{\Sigma}_k^{(m,n)}$, and these columns are in fact mixing vectors corresponding to source signals that are similar between the $m$th and $n$th dataset. Hence, in a GJD based method, diagonalization of cross-terms mainly exploits the pairwise similarity among the multiset signals. In addition, if only the cross-terms are used, a GJD based J-BSS method can only find the similar source components that are consistently present in multiple datasets, i.e., it does not find specific source signals that are uniquely present in each dataset.

The auto-term, on the other hand, is calculated from each dataset itself. Therefore, the auto-term contains specific information of each dataset, but does not indicate the similarity between different datasets. As a result, in a GJD based method, diagonalization of auto-terms can extract specific source signals that are uniquely present in each dataset. However, if only the auto-terms are used, the inter-set similarity is not exploited, and the GJD based J-BSS method will have worsened accuracy and misaligned permutation for the estimation of similar sources signals in different datasets.

Based on the above analysis, we know that the cross-term and the auto-term characterize the similarity and diversity of a J-BSS problem, respectively. And only when both of them are jointly exploited, can a GJD method extract both similar sources between different datasets and specific sources that are uniquely present in each dataset.

## VI. SIMULATION

In this section, we present simulation results to demonstrate the performance of the proposed algorithm (GOJD-JCB) in comparison with two existing GOJD algorithms, namely GOJD based on sequential orthogonal procrustes (GOJD-SOP) [16], and orthogonal joint blind separation (GOJD-OJB) [17]. In the second and third simulations, we also include JSVD [19] and MCCA [3] in the comparison. All the simulations are performed under the following configurations: CPU–Intel Xeon E5-2640 2.4 GHz; Memory–64 GB; System–64 bit Windows 10; Matlab R2016b.

### A. COMPARISON OF CONVERGENCE PATTERNS

In this simulation, we compare the convergence patterns of the compared GOJD algorithms in the exact case. We construct $N \times N$ data matrices $\boldsymbol{C}_k^{(m,n)} \in \mathbb{C}^{N\times N}$ according to (1) where diagonal entries of $N \times N$ matrices $\boldsymbol{\Lambda}_k^{(m,n)}$ are drawn from a standard normal distribution. The loading matrices $\boldsymbol{B}^{(m)} \in \mathbb{C}^{N\times N}$ are randomly generated unitary matrices, with $1 \leq m,n \leq M, 1 \leq k \leq K$.We use the overall off-norm to evaluate the algorithms:

$$ORON(it) = \frac{\sum_{m,n,k} \left\| \text{off}(\boldsymbol{B}_{it}^{(m)H} \boldsymbol{C}_k^{(m,n)} \boldsymbol{B}_{it}^{(n)}) \right\|_F^2}{\sum_{m,n,k} \left\| \text{diag}(\boldsymbol{B}_{it}^{(m)H} \boldsymbol{C}_k^{(m,n)} \boldsymbol{B}_{it}^{(n)}) \right\|_F^2}, \quad (19)$$

where $\boldsymbol{B}_{it}^{(m)}$ denotes the $m$th updated loading matrix in the $it$-th sweep.

We use the stopping criterion (18) for the compared algorithms, where the tolerance is set to $\xi = 10^{-6}$. The curves of 10 independent runs are drawn in Fig. 1 in following cases: (a) $M = 5, N = 10, K = 20$; (b) $M = 5, N = 20, K = 20$; (c) $M = 20, N = 5, K = 20$; (d) $M = 20, N = 20, K = 20$.

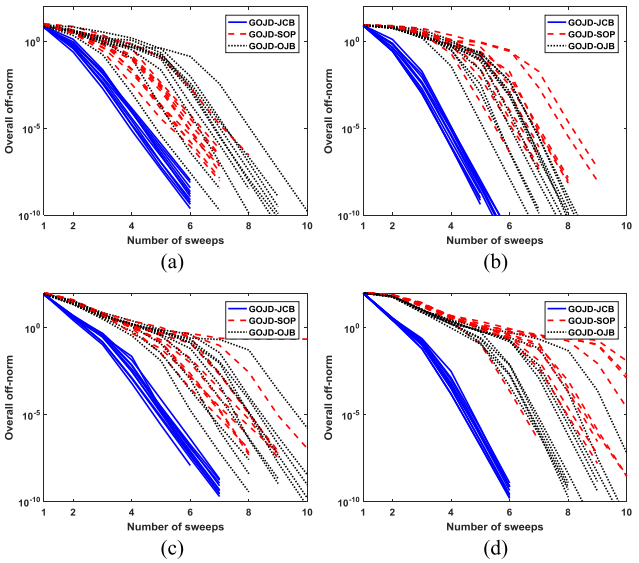


**FIGURE 1. Overall off-norm versus the number of sweeps in the exact case, under four settings with different $M$ and $N$. It is generally shown that the proposed GOJD-JCB algorithm has better and more consistent convergence pattern than GOJD-SOP and GOJD-OJB. (a) $M = 5, N = 5, K = 20$. (b) $M = 5, N = 20, K = 20$. (c) $M = 20, N = 5, K = 20$. (d) $M = 20, N = 20, K = 20$.**

From FIGURE 1, we see that GOJD-JCB takes less sweeps than GOJD-SOP and GOJD-OJB, and that the convergence pattern of GOJD-JCB is less affected by $M$ and $N$. This observation clearly suggests that the proposed algorithm has a better and more consistent convergence pattern than GOJD-SOP and GOJD-OJB.

### B. J-BSS OF HYBRID MULTISET SIGNALS

In this simulation, we apply GOJD-JCB, GOJD-SOP, GOJD-OJB, JSVD and MCCA in J-BSS of hybrid multi-set signals. The term "hybrid" means that the generated multi-set signals contain both similar source components that are consistently present in different datasets, and diverse source components that are specifically present in each dataset. The multi-set signals are generated as follows:

$$\boldsymbol{X}^{(m)} = \sigma_s \boldsymbol{A}^{(m)} \frac{\boldsymbol{S}^{(m)T}}{\left\|\boldsymbol{S}^{(m)T}\right\|_F} + \sigma_n \frac{\boldsymbol{N}^{(m)}}{\left\|\boldsymbol{N}^{(m)}\right\|_F}, \quad (20)$$

where $\boldsymbol{A}^{(m)}, m = 1, \ldots, M$ of size $N \times R^{(m)}$ are randomly generated unitary matrices. For each sampled time instant $t$, we define a new source matrix:

$$\boldsymbol{S}_r \triangleq [\boldsymbol{s}_r^{(1)}, \ldots, \boldsymbol{s}_r^{(M)}] \in \mathbb{C}^{Q \times M}, \tag{21}$$

where $\boldsymbol{s}_r^{(m)}$ is the $r$th column of $\boldsymbol{S}^{(m)} \in \mathbb{C}^{Q \times R^{(m)}}$, and $Q$ denotes the number of time samples. We generate $\boldsymbol{S}_r$ for all the values of $r = 1, \ldots, R$, with $R < R^{(m)}$, as follows:

$$\boldsymbol{S}_r = \boldsymbol{S'}_r \boldsymbol{Q}_r, \tag{22}$$

$\boldsymbol{Q}_r \in \mathbb{C}^{M \times M}$ is a full rank matrix used to introduce inter-set dependence between the corresponding source signals in different datasets. Both the real and imaginary part of each entry of $\boldsymbol{Q}_r$ are drawn from a standard normal distribution. The underlying generating source matrix $\boldsymbol{S}'_r(t) \triangleq [\boldsymbol{s}_r'^{(1)}, \ldots, \boldsymbol{s'}_r^{(M)}]$ consists of complex binary phase shift keying (BPSK) signals that are amplitude modulated across $P$ time slots of length $L'$:

$$\boldsymbol{s'}_r^{(m)} = [\eta_1 \cdot \boldsymbol{s'}_{r,1}^{(m)T}, \ldots, \eta_P \cdot \boldsymbol{s'}_{r,P}^{(m)T}]^T \in \mathbb{C}^Q, \tag{23}$$

where $\eta_1, \ldots, \eta_P$ are amplitude modulation coefficients that are randomly drawn from a uniform distribution over [0, 1], and where $Q = PL'$. The sub-vector $\boldsymbol{s'}_{r,p}^{(m)}$ of length $L'$ is a complex BPSK sequence with entries chosen from symbols $\{-1, 1\}$ with equal probability. By definition, $\boldsymbol{s'}_r^{(m)}$ is the concatenation of $P$ BPSK sequences, the amplitudes of which have been modulated by coefficients $\eta_1, \ldots, \eta_P$.

Note that the above source construction procedure is conducted for $r = 1, \ldots, R$, indicating that the first $R$ source signals of identical channel index of different datasets are statistically dependent. In fact, the above procedure only generates the 'similar' components in the multi-set signals.

For $r = R + 1, \ldots, R^{(m)}$, we generate the source signals $\boldsymbol{s}_r^{(m)}$ independently for each dataset, as amplitude modulated BPSK signals.

By definition, we know that for each dataset, the first $R$ source signals correspond to the similar part, while the rest correspond to the diverse part, of the source components, respectively. Correspondingly, we denote the first $R$ columns and the last $(R^{(m)} - R)$ columns of $\boldsymbol{A}^{(m)}$ as the similar part and diverse part of the mixing matrix, respectively, $m = 1, \ldots, M$.

The noise term $\boldsymbol{N}^{(m)} \in \mathbb{C}^{N \times Q}$ is generated as white Gaussian noise. We define the signal-to-noise ratio ($SNR$) with the signal level $\sigma_s$ and noise level $\sigma_n$, as follows:

$$SNR = 20 \log \left( \sigma_s / \sigma_n \right). \tag{24}$$

We calculate the data matrices $\boldsymbol{C}_k^{(m,n)}$ by a finite sampling version of (3) as follows:

$$\boldsymbol{C}_k^{(m,n)} = \frac{1}{L} \cdot \left[ \sum\nolimits_{l=1}^{L} \left( \boldsymbol{X}^{(m,k)} \right)_{(:,l)} \left( \boldsymbol{X}^{(n,k)} \right)_{(:,l)}^H \right], \tag{25}$$

where $\boldsymbol{X}^{(m,k)} \in \mathbb{C}^{N \times L}$ denotes the $k$th temporal frame of $\boldsymbol{X}^{(m)}$ with frame length $L$, overlapped with adjacent frames with an overlapping rate $\alpha \in [0, 1]$.

We adopt the inter-symbol-interference (ISI) [7] to evaluate the compared algorithms. More exactly, for the estimate $\tilde{\boldsymbol{J}}$ of a unitary matrix $\boldsymbol{J}$, ISI is defined as follows:

$$ISI(\boldsymbol{F}) \frac{1}{2N(N-1)} \left[ \sum_{i=1}^{N} \left( \sum_{j=1}^{N} \frac{f_{ij}}{\max_k f_{ik}} - 1 \right) + \sum_{j=1}^{N} \left( \sum_{i=1}^{N} \frac{f_{ij}}{\max_k f_{kj}} - 1 \right) \right], \tag{26}$$

where $\boldsymbol{F} \triangleq |\tilde{\boldsymbol{J}}^H \boldsymbol{J}|$. In the simulation, we calculate the ISI for the estimates of the similar part of the mixing matrix of each dataset and average over all the datasets. The obtained metric is denoted as the mean $\text{ISI}_{(\text{sim})}$ ($\text{M-ISI}_{(\text{sim})}$), with subscript '(sim)' suggesting that it evaluates the estimation of the similar part of the mixing matrix. Similarly, we calculate the mean ISI for the diverse part of each dataset, denoted as $\text{M-ISI}_{(\text{div})}$.

In the simulation, we let $R^{(1)} = R^{(2)} = \cdots = R^{(M)} = N$, and $R = 0.5N$. First we fix $M = 8$, $N = 4$, $Q = 5000$, and set the frame length to $L = Q/20$, and the overlapping factor to $\alpha = 0.5$. With varying SNR, we draw the curves of $\text{M-ISI}_{(\text{sim})}$, $\text{M-ISI}_{(\text{div})}$, number of sweeps, and CPU time in FIGURE 2. Note that each point in the curves is calculated as the average of 200 Monte-Carlo runs. In the simulation, we use the stopping criterion (18) for the compared algorithms, where the tolerance is set to $\xi = 10^{-6}$.

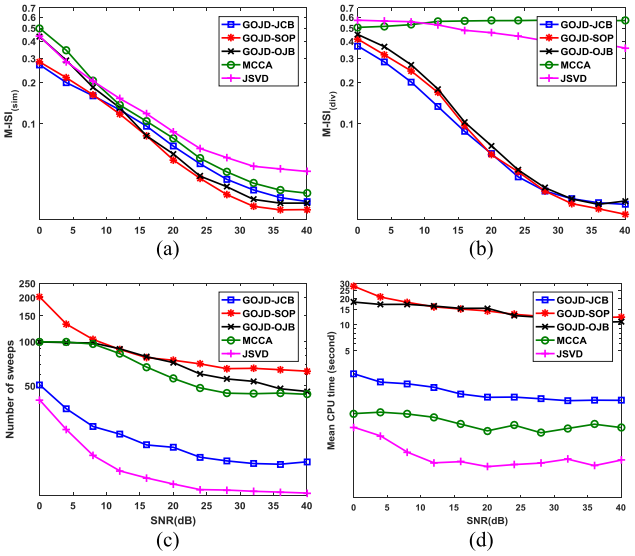


**FIGURE 2.** **$\text{M-ISI}_{(\text{sim})}$, $\text{M-ISI}_{(\text{div})}$, number of sweeps, and mean CPU time versus SNR, with $M = 8$, $N = 4$, $T = 5000$, $R^{(m)} = 2$, $R = 2$. It is in general shown that GOJD-JCB yields accurate estimates of both the similar and diverse components of the multi-set source signals, and that GOJD-JCB is also computationally very efficient. (a) $\text{M-ISI}_{(\text{sim})}$ versus SNR. (b) $\text{M-ISI}_{(\text{div})}$ versus SNR. (c) Number of sweeps versus SNR. (d) Mean CPU time versus SNR.**

It is shown in FIGURE 2 (a) that all the compared algorithms generate reasonably accurate estimates of the similar part of the mixing matrices. We note that GOJD-JCB

and GOJD-SOP provide more accurate results in low SNR. In high SNR, all the GOJD algorithms have almost identical accuracy, which is slightly better than that of JSVD and MCCA. This observation generally suggests that the compared algorithms can be used to extract the similar source components that are consistently present in different datasets.

In FIGURE 2 (b), we see that the three GOJD algorithms provide correct results for the estimation of the diverse part of each mixing matrix, while JSVD and MCCA do not generate correct results. This is precisely due to the fact that the auto-term, which characterizes the diversity of each dataset, is not included in JSVD and MCCA, as explained in *Subsection V.D*. In addition, we see that the proposed GOJD-JCB algorithm is more accurate than GOJD-SOP and GOJD-OJB.

In FIGURE 2 (c), we see that GOJD-JCB takes much fewer sweeps than GOJD-SOP, GOJD-OJB, and MCCA. This observation shows that the proposed algorithm has nice convergence pattern. Note that the number of sweeps for GOJD-JCB is slightly more than that for JSVD, while the former algorithm is significantly more accurate than the latter, as shown in FIGURE 2 (a) and FIGURE 2 (b).

In FIGURE 2 (d), we see that GOJD-JCB is much faster than GOJD-SOP and GOJD-OJB. This is not only because the proposed method has better convergence pattern than the latter two methods, but also due to the fact that the per-sweep complexity of GOJD-JCB is lower, as explained in *Sub-section V.B*. In comparison with JSVD and MCCA, we note that GOJD-JCB is slightly slower, but significantly more accurate. In general, we observe in the four sub-figures of FIGURE 2, that the proposed GOJD-JCB algorithm yields accurate estimates of both similar and diverse components of the multi-set source signals, and it is also computationally very efficient with varying SNR.

Then, we fix SNR $= 20$dB, $T = 5000$, $N = 4$, and draw the curves of M-ISI$_{(\text{sim})}$, M-ISI$_{(\text{div})}$, number of sweeps and the mean CPU time of the compared algorithms with $M$ varying from 2 to 10, in FIGURE 3. We observe from FIGURE 3 (a) and FIGURE 3 (b) that the GOJD algorithms yield reasonably accurate estimates for both similar and diverse components. JSVD obtains correct results in the estimation of the similar source components, but fails to extract the diverse components. MCCA only yields correct estimates of the similar components when $M$ is sufficiently large.

From FIGURE 3 (c) and FIGURE 3 (d) we note that GOJD-JCB takes much fewer sweeps and less CPU time than GOJD-SOP and GOJD-OJB. In comparison with JSVD, GOJD-JCB is only slightly slower while significantly more accurate. In comparison with MCCA, GOJD-JCB is faster for small $M$ while slightly slower when $M$ exceeds the number of six. The results in FIGURE 3 generally demonstrate the nice performance of GOJD-JCB in handling large number of datasets.

Last, we fix SNR $= 20$dB, $T = 5000$, $M = 8$, and draw the curves of M-ISI(sim), M-ISI(div), number of sweeps and the mean CPU time of the compared algorithms with $N$ varying from 4 to 10, in FIGURE 4. We note that in the simulation

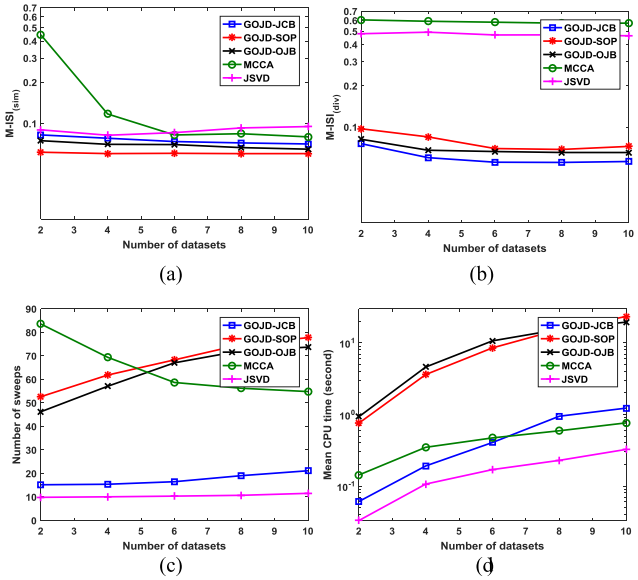


**FIGURE 3.** M-ISI$_{(\text{sim})}$, M-ISI$_{(\text{div})}$, number of sweeps, and mean CPU time versus $M$, with $SNR = 20$dB, $N = 4$, $T = 5000$, $R^{(m)} = 2$, $R = 2$. It is shown that GOJD-JCB is much faster than GOJD-OJB and GOJD-SOP, with either small or large number of datasets. GOJD-JCB is only slightly slower while significantly more accurate than MCCA and JSVD. (a) M-ISI$_{(\text{sim})}$ versus $M$. (b) M-ISI$_{(\text{div})}$ versus $M$. (c) Number of sweeps versus $M$. (d) Mean CPU time versus $M$.

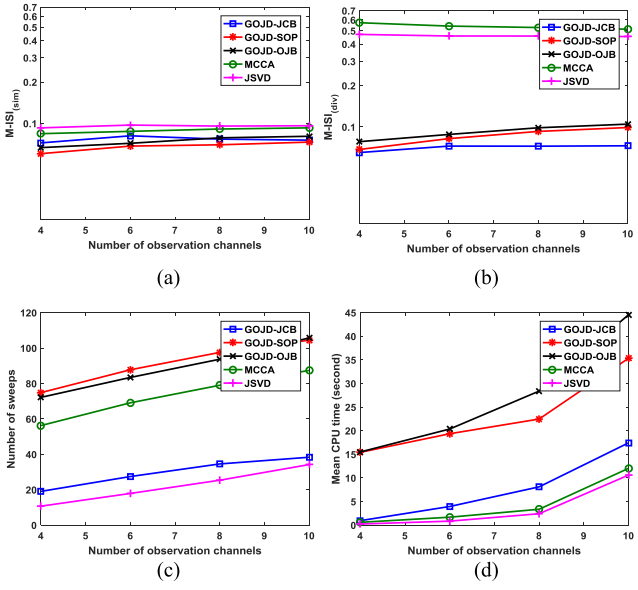


**FIGURE 4.** M-ISI$_{(\text{sim})}$, M-ISI$_{(\text{div})}$, number of sweeps, and mean CPU time versus $N$, with $SNR = 20$dB, $M = 8$, $T = 5000$, $R^{(m)} = N$, $R = 0.5N$. It is shown that GOJD-JCB is much faster than GOJD-OJB and GOJD-SOP, with either small or large $N$. GOJD-JCB is only slightly slower while significantly more accurate than MCCA and JSVD. (a) M-ISI$_{(\text{sim})}$ versus $N$. (b) M-ISI$_{(\text{div})}$ versus $N$. (c) Number of sweeps versus $N$. (d) Mean CPU time versus $N$.

we keep $R^{(1)} = R^{(2)} = \cdots = R^{(M)} = N$, and $R = 0.5N$. Therefore, with $N$ increasing, the number of both similar and diverse source components also increases, and the problem actually becomes more challenging.

We see from FIGURE 4 (a) and FIGURE 4 (b) that the GOJD algorithms can accurately estimate both similar

and diverse source signals, while JSVD and MCCA only extract the similar source signals. From FIGURE 4 (c) we see that with $N$ increasing the number of sweeps increases for all the compared algorithms, implying that the problem indeed becomes more challenging. Note that the number of sweeps that GOJD-JCB takes is much smaller than that of GOJD-SOP, GOJD-OJB and MCCA. It is slightly larger than that of JSVD. From FIGURE 4 (d) we note that GOJD-JCB consumes much less CPU time than GOJD-SOP and GOJD-OJB. In comparison with JSVD and MCCA, GOJD-JCB is slightly slower while significantly more accurate. The results in FIGURE 4 generally show the nice performance of GOJD-JCB in challenging problems where the number of observation channels and the number of source signals are large.

### C. J-BSS OF MULTI-SUBJECT FMRI SIGNALS

In this simulation, we apply the proposed GOJD-JCB algorithm in J-BSS of multi-subject fMRI signals. We adopt the benchmark simulated complex fMRI sources, obtained from [30], to generate the multi-subject fMRI signals. The magnitude images of the simulated fMRI spatial maps (SM) and their corresponding time courses (TC) are shown in FIGURE 5.

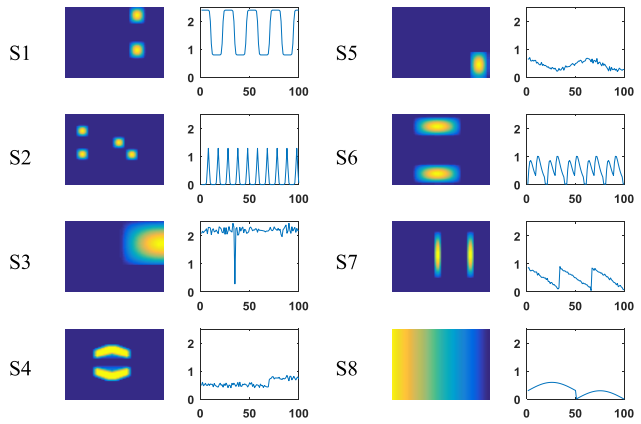


**FIGURE 5.** The amplitude of the simulated fMRI spatial maps (SM) and their corresponding time courses (TC). The eight sources are placed in two columns. The left column holds the SMs and TCs corresponding to sources 1–4 and the right column holds the SMs and TCs corresponding to sources 5–8. In each column, the left sub-column holds the SMs and the right sub-column holds the TCs.

The multi-subject fMRI signal is generated as follows:

$$\boldsymbol{X}^{(m)} = \sigma_s \sum_{r\in\Omega^{(m)}} \boldsymbol{a}_r \boldsymbol{s}_r^T + \sigma_n \boldsymbol{N}^{(m)}, \tag{27}$$

where $\boldsymbol{X}^{(m)}, \boldsymbol{N}^{(m)} \in \mathbb{C}^{N\times T}$ denotes the observed signal and noise term of the $m$th subject, respectively, $m = 1, \ldots, M$. The vector $\boldsymbol{a}_r \in \mathbb{C}^N$ denotes the $r$th TC, and $\boldsymbol{s}_r \in \mathbb{C}^T$ denotes the vectorized version of the $r$th SM, $r = 1, \ldots, 8$. For the benchmark fMRI source signals used here, we note that $N = 100$, and $T = 3600$. The source index set $\Omega^{(m)}$ of the $m$th subject is a subset of $\{1, \ldots, 8\}$, indicating which sources (TCs and SMs) exist in the mixture of the $m$th subject. By definition, the number of sources of the $m$th subject is equal to the cardinality of $\Omega^{(m)}$ : $R^{(m)} = \text{card}(\Omega^{(m)})$.

In the simulation, we consider that there are $M = 6$ subjects. We set the source index set $\Omega^{(m)}$ for each subject as: $\Omega^{(1)} = \{1, 2, 5, 6, 7\}$; $\Omega^{(2)} = \{1, 2, 4\}$; $\Omega^{(3)} = \{1, 2, 4, 5\}$; $\Omega^{(4)} = \{1, 2, 8\}$; $\Omega^{(5)} = \{1, 2, 3, 5\}$; $\Omega^{(6)} = \{1, 2, 3, 4\}$. Note that the first and second source signals are consistently present in all the datasets and thus are the similar source components.

We fix SNR = 50dB. The amplitude of the SM estimates for all the datasets via GOJD-JCB is shown in TABLE I. In addition, we calculate the ISI (ISI) for both the SM and TC estimates, averaged over 100 Monte-Carlo runs, for GOJD-JCB, MCCA, and JSVD. The results are given in FIGURE 6. We see from TABLE II that J-BSS based on the proposed GOJD-JCB algorithm extracts both similar components and diverse components for all the datasets. We note that, due to the exploitation of the similarity among distinct subjects, the similar source components across different datasets are naturally aligned. The diverse components for each dataset, on the other hand, are identified due to the use of auto-terms in the proposed algorithm. We also note that there exist a certain level of cross-talk artifact in some extracted components. This is mainly due to the fact that the source SMs are not mutually independent in the strict sense.

**TABLE 2.** Magnitude of SM estimates via GOJD-JCB.

| Subject | Similar components | | Diverse components | | |
|---|---|---|---|---|---|
| #1 | | | | | |
| #2 | | | | | |
| #3 | | | | | |
| #4 | | | | | |
| #5 | | | | | |
| #6 | | | | | |

FIGURE 6 generally shows that GOJD-JCB has very good performance with regards to both TC and SM estimation. MCCA fails to generate reasonable results, mainly because it does not make use of the non-stationarity structure of the SM components, and it does not make use of the auto-covariance terms. The performance of JSVD is also worse than that

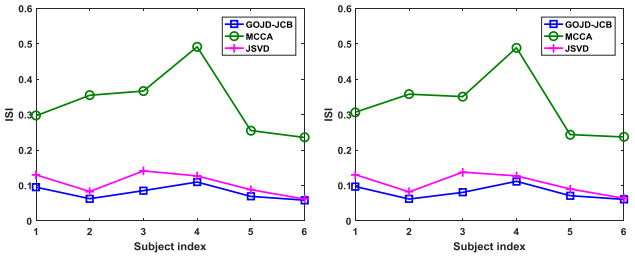


**FIGURE 6.** ISI for both SM and TC estimates for each subject, with $SNR = 50$dB, $M = 6$, $T = 3600$, $N = 100$. It is shown that GOJD-JCB is more accurate than MCCA and JSVD. (c) ISI for SM estimates (d) ISI for TC estimates.

of GOJD. In addition, we note that the similar components extracted by JSVD are not aligned, due to the fact that the method works on two datasets each time.

## VII. CONCLUTION

J-BSS is an emerging topic with applications in multi-set data fusion problems, and GOJD is an important tool for performing J-BSS. In this study, we have proposed a Jacobi GOJD algorithm by using Givens rotations. The proposed algorithm alternates between updates of different loading matrices, each solved by a sequence of elementary Givens rotations obtained in closed-form. We have considered the scenario where different datasets in J-BSS may have different number of sources, among which there exist both similar components that are consistently present in multiple datasets, and diverse components that are uniquely present in each dataset. We have shown how J-BSS based on the proposed GOJD algorithm can effectively extract both similar and diverse source components. Analysis and simulation results have shown that, in comparison with two existing GOJD algorithms, the proposed GOJD algorithm has lower complexity, faster convergence, and slightly better accuracy in the extraction of the diverse source components. In comparison with MCCA and JSVD, the proposed GOJD algorithm is slightly slower, yet significantly more accurate.

## REFERENCES

[1] N. M. Correa, T. Eichele, T. Adali, Y.-O. Li, and V. D. Calhoun, "Multi-set canonical correlation analysis for the fusion of concurrent single trial ERP and functional MRI," *NeuroImage*, vol. 50, no. 4, pp. 1438–1445, May 2010.

[2] M. Moosmann, T. Eichele, H. Nordby, and V. D. Calhoun, "Joint independent component analysis for simultaneous EEG–fMRI: Principle and simulation," *Int. J. Psychophysiol.*, vol. 67, no. 3, pp. 212–221, Mar. 2008.

[3] Y.-O. Li, T. Eichele, V. D. Calhoun, and T. Adali, "Group study of simulated driving fMRI data by multiset canonical correlation analysis," *J. Signal Process. Syst.*, vol. 68, no. 1, pp. 31–48, Jul. 2012.

[4] J.-H. Lee, T.-W. Lee, F. A. Jolesz, and S.-S. Yoo, "Independent vector analysis (IVA): Multivariate approach for fMRI group study," *NeuroImage*, vol. 40, no. 1, pp. 86–109, 2008.

[5] L.-D. Kuang, Q.-H. Lin, X.-F. Gong, F. Cong, J. Sui, and V. D. Calhoun, "Multi-subject fMRI analysis via combined independent component analysis and shift-invariant canonical polyadic decomposition," *J. Neurosci. Methods*, vol. 256, pp. 127–140, Dec. 2015.

[6] Y. Liang, J. Harris, S. M. Naqvi, G. Chen, and J. A. Chambers, "Independent vector analysis with a generalized multivariate Gaussian source prior for frequency domain blind source separation," *Signal Process.*, vol. 105, no. 12, pp. 175–184, Dec. 2014.

[7] J. Hao, I. Lee, T.-W. Lee, and T. J. Sejnowski, "Independent vector analysis for source separation using a mixture of Gaussians prior," *Neural Comput.*, vol. 22, no. 6, pp. 1646–1673, May 2010.

[8] T. Kim, T. Eltoft, and T.-W. Lee, "Independent vector analysis: An extension of ICA to multivariate components," in *Proc. ICA*, Charleston, NC, USA, Mar. 2006, pp. 165–172.

[9] T. Adalı, M. Anderson, and G.-S. Fu, "Diversity in independent component and vector analyses: Identifiability, algorithms, and applications in medical imaging," *IEEE Signal Process. Mag.*, vol. 31, no. 3, pp. 18–33, May 2014.

[10] S. Ma, V. D. Calhoun, R. Phlypo, and T. Adali, "Dynamic changes of spatial functional network connectivity in healthy individuals and schizophrenia patients using independent vector analysis," *NeuroImage*, vol. 90, pp. 196–206, Apr. 2014.

[11] I. Lee, T. Kim, and T.-W. Lee, "Fast fixed-point independent vector analysis algorithms for convolutive blind source separation," *Signal Process.*, vol. 87, no. 8, pp. 1859–1871, Aug. 2007.

[12] M. Anderson, T. Adali, and X.-L. Li, "Joint blind source separation with multivariate Gaussian model: Algorithms and performance analysis," *IEEE Trans. Signal Process.*, vol. 60, no. 4, pp. 1672–1683, Apr. 2012.

[13] N. M. Correa, T. Adali, Y.-O. Li, and V. D. Calhoun, "Canonical correlation analysis for data fusion and group inferences," *IEEE Signal Process. Mag.*, vol. 27, no. 4, pp. 39–50, Jun. 2010.

[14] Y.-O. Li, T. Adali, W. Wang, and V. D. Calhoun, "Joint blind source separation by multiset canonical correlation analysis," *IEEE Trans. Signal Process.*, vol. 57, no. 10, pp. 3918–3929, Oct. 2009.

[15] T. Adalı, Y. Levin-Schwartz, and V. D. Calhoun, "Multimodal data fusion using source separation: Application to medical imaging," *Proc. IEEE*, vol. 103, no. 9, pp. 1494–1506, Sep. 2015.

[16] X.-L. Li, T. Adali, and M. Anderson, "Joint blind source separation by generalized joint diagonalization of cumulant matrices," *Signal Process.*, vol. 91, no. 10, pp. 2314–2322, Oct. 2011.

[17] M. Congedo, R. Phlypo, and J. Chatel-Goldman, "Orthogonal and non-orthogonal joint blind source separation in the least-squares sense," in *Proc. EUSIPCO*, Bucharest, Romania, Aug. 2012, pp. 1885–1889.

[18] X.-F. Gong, Q.-H. Lin, and K. Wang, "Joint non-orthogonal joint diagonalization based on LU decomposition and Jacobi scheme," in *Proc. ChinaSIP*, Beijing, China, Jul. 2013, pp. 25–29.

[19] M. Congedo, R. Phlypo, and D.-T. Pham, "Approximate joint singular value decomposition of an asymmetric rectangular matrix set," *IEEE Trans. Signal Process.*, vol. 59, no. 1, pp. 415–424, Jan. 2011.

[20] X. F. Gong, X. L. Wang, and Q. H. Lin, "Generalized non-orthogonal joint diagonalization with LU decomposition and successive rotations," *IEEE Trans. Signal Process.*, vol. 63, no. 5, pp. 1322–1334, Mar. 2015.

[21] X.-F. Gong, Y.-N. Hao, and Q.-H. Lin, "Joint canonical polyadic decomposition of two tensors with one shared loading matrix," in *Proc. MLSP*, Southampton, U.K., Sep. 2013, pp. 1–6.

[22] M. Sorensen and L. De Lathauwer, "Coupled canonical polyadic decompositions and (coupled) decompositions in multilinear rank-$(L_{r,n}, L_{r,n}, 1)$ terms—Part I: Uniqueness," *SIAM J. Matrix Anal. Appl.*, vol. 36, no. 2, pp. 496–522, Apr. 2015.

[23] M. Sorensen, I. Domanov, and L. De Lathauwer, "Coupled canonical polyadic decompositions and (coupled) decompositions in multilinear rank-$(L_{r,n}, L_{r,n}, 1)$ terms—Part II: Algorithms," *SIAM J. Matrix Anal. Appl.*, vol. 36, no. 3, pp. 1015–1045, Jul. 2015.

[24] M. Sørensen, I. Domanov, and L. De Lathauwer, "Coupled canonical polyadic decompositions and multiple shift invariance in array processing," *IEEE Trans. Signal Process.*, vol. 66, no. 13, pp. 3475–3490, Jul. 2018.

[25] X.-F. Gong, Q.-H. Lin, F.-Y. Cong, and L. De Lathauwer, "Double coupled canonical polyadic decomposition for joint blind source separation," *IEEE Trans. Signal Process.*, vol. 66, no. 13, pp. 3475–3490, Jul. 2018, doi: 10.1109/TSP.2018.2830317.

[26] X.-F. Gong, Q.-H. Lin, O. Debals, N. Vervliet, and L. De Lathauwer, "Coupled rank-$(L_m, L_n, \cdot)$ terms block term decomposition by coupled block simultaneous generalized Schur decomposition," in *Proc. ICASSP*, Shanghai, China, Mar. 2016, pp. 2554–2558.

[27] E. Sanchez, L. S. Ramos, and B. R. Kowalski, "Generalized rank annihilation method: I. Application to liquid chromatography—Diode array ultraviolet detection data," *J. Chromatogr. A*, vol. 385, no. 1, pp. 151–164, Jan. 1987.

[28] D. Lahat, T. Adali, and C. Jutten, "Multimodal data fusion: An overview of methods, challenges, and prospects," *Proc. IEEE*, vol. 103, no. 9, pp. 1449–1477, Sep. 2015.

[29] X.-L. Wang, X.-F. Gong, and Q.-H. Lin, "A study on parallelization of successive rotation based joint diagonalization," in *Proc. DSP*, Hong Kong, Aug. 2014, pp. 1–6.
[30] MLSP-LAB. *Simulating Complex fMRI-Like Sources*. [Online]. Available: http://mlsp.umbc.edu/simulated_complex_fmri_data.html

**XIAO-FENG GONG** (M'10) received the bachelor's degree in information engineering from the Beijing Institute of Technology, China, in 2003. From 2003 to 2009, he took the combined master and Ph.D. program of the Beijing Institute of Technology, and received the Ph.D. degree in communication and information system in 2009.

From 2014 to 2015, he was a Visiting Research Associate with KU Leuven Kortrijk Campus, Kortrijk, Belgium. He is currently an Associate Professor with the Dalian University of Technology, Dalian, China. His research concerns tensor-based signal processing.

**LEI MAO** received the B.S. degree in electronic and information engineering from the Dalian University of Technology, Dalian, China, in 2015, where she is currently pursuing the M.S. degree.

Her research interests are joint blind source separation methods and applications.

**YING-LIANG LIU** received the B.S. degree in communication engineering from Shandong University, Weihai Campus, China, in 2013, and the M.S. degree in information and communication engineering from the Dalian University of Technology, Dalian, China, in 2016. She is currently an Engineer at the Qingdao Aerospace Semiconductor Research Institute Company Ltd.

**QIU-HUA LIN** (M'10) received the bachelor's degree in wireless communication, the master's degree in communications and electronic systems, and the Ph.D. degree in signal and information processing from the Dalian University of Technology, Dalian, China, in 1991, 1994, and 2006, respectively.

She has been with the Dalian University of Technology since 1994. She was a Teaching Assistant from 1994 to 1995, a Lecturer from 1996 to 2001, and an Associate Professor from 2002 to 2006. She was a Visiting Scholar with the University of New Mexico, Albuquerque, NM, USA, in 2006. Since 2007, she has been a Professor with the School of Information and Communication Engineering. Her research interests include array signal processing, biomedical signal processing, image processing, and machine learning.