A Tensor Framework for Nonunitary Joint Block Diagonalization

Dimitri Nion

***Abstract*—This paper introduces a tensor framework to solve the problem of nonunitary joint block diagonalization (JBD) of a set of real or complex valued matrices. We show that JBD can be seen as a particular case of the block-component-decomposition (BCD) of a third-order tensor. The resulting tensor model fitting problem does not require the block-diagonalizer to be a square matrix: the over- and underdetermined cases can be handled. To compute the tensor decomposition, we build an efficient nonlinear conjugate gradient (NCG) algorithm. In the over- and exactly determined cases, we show that exact JBD can be computed by a closed-form solution based on eigenvalue analysis. In approximate JBD problems, this solution can be used to efficiently initialize any iterative JBD algorithm such as NCG. Finally, we illustrate the performance of our technique in the context of independent subspace analysis (ISA) based on second-order statistics (SOS).**

***Index Terms*—Blind source separation, conjugate gradient, independent subspace analysis, joint block diagonalization (JBD), second-order statistics, tensor decomposition.**

I. Introduction

LET $\{\mathbf{X}_k\}_{k=1}^K \in \mathbb{K}^{I\times I}$ be a set of K matrices, that can approximately be jointly block diagonalized

$$\mathbf{X}_k = [\mathbf{A}_1,\ldots,\mathbf{A}_R]\begin{bmatrix}\mathbf{D}_{k1} & & \mathbf{0}\\ & \ddots & \\ \mathbf{0} & & \mathbf{D}_{kR}\end{bmatrix}\begin{bmatrix}\mathbf{A}_1^\ddagger\\ \vdots\\ \mathbf{A}_R^\ddagger\end{bmatrix} + \mathbf{N}_k = \mathbf{A}\mathbf{D}_k\mathbf{A}^\ddagger + \mathbf{N}_k \quad (1)$$

where $\mathbb{K} = \mathbb{C}$ or $\mathbb{K} = \mathbb{R}$, $\cdot^\ddagger$ denotes either the transpose ($\mathbf{A}^\ddagger = \mathbf{A}^T$) or the conjugate transpose ($\mathbf{A}^\ddagger = \mathbf{A}^H$). The matrix $\mathbf{A} \in \mathbb{K}^{I\times N}$ is partitioned in R blocks $\{\mathbf{A}_r\}_{r=1}^R \in \mathbb{K}^{I\times L_r}$, $I \geq L_r$, $N = \sum_{r=1}^R L_r$, the block-diagonal matrix $\mathbf{D}_k$, $k = 1,\ldots,K$, is built from the R blocks $\{\mathbf{D}_{kr}\}_{r=1}^R \in \mathbb{K}^{L_r\times L_r}$ and the matrix $\mathbf{N}_k \in \mathbb{K}^{I\times I}$ denotes residual noise. The joint block diagonalization (JBD) problem consists of the estimation of $\mathbf{A}$ and $\{\mathbf{D}_k\}_{k=1}^K$, given the matrices $\{\mathbf{X}_k\}_{k=1}^K$. It can be noticed that the JBD model remains unchanged if one substitutes $\mathbf{A}$ by $\mathbf{A}\tilde{\mathbf{D}}\tilde{\mathbf{P}}$ and $\mathbf{D}_k$ by $\tilde{\mathbf{P}}^{-1}\tilde{\mathbf{D}}^{-1}\mathbf{D}_k\tilde{\mathbf{D}}^{-\ddagger}\tilde{\mathbf{P}}^{-\ddagger}$, where $\tilde{\mathbf{D}}$ is a nonsingular block-diagonal matrix, with arbitrary blocks of the same dimensions as in $\mathbf{D}_k$, and $\tilde{\mathbf{P}}$ is an arbitrary block-wise permutation matrix. The JBD model is *essentially unique* when it is only subject to these indeterminacies. Practically speaking, this means that one can only estimate the R subspaces $\{\mathrm{Span}(\mathbf{A}_r)\}_{r=1}^R$ in an arbitrary order, i.e., the matrices $\{\mathbf{A}_r\}_{r=1}^R$ remain unknown, unless additional constraints are imposed, e.g., a particular algebraic structure such as Vandermonde or Toeplitz.

The particular case $L_r = 1, \forall r$, of JBD is known as joint diagonalization (JD). The JD problem was first investigated under the unitary constraint ($\mathbf{A}^H\mathbf{A} = \mathbf{I}_N$, where $\mathbf{I}_N$ is the $N \times N$ identity matrix), after which several nonunitary algorithms have emerged, see [1], [2], and references therein. Most of the nonunitary JD algorithms assume that $\mathbf{A}$ is square or tall with full column rank. Recently, it has been shown [3] that, in a tensor framework, the JD problem can be seen as a particular case of the PARAllel FACtor (PARAFAC) decomposition [4], [5], also known as CANonical DECOMPosition (CANDECOMP) [6], of the third-order tensor $\mathcal{X} \in \mathbb{K}^{I\times I\times K}$ built by stacking the K matrices $\mathbf{X}_k$ along the third mode. This link was exploited to build the PARAFAC-based Second-Order Blind Identification of Underdetermined Mixtures (SOBIUM) algorithm [3], which covers the overdetermined case ($\mathbf{A}$ is tall and full column rank) but also several underdetermined cases ($\mathbf{A}$ is fat and full row rank), thanks to powerful uniqueness properties of the PARAFAC decomposition [7]. This equivalence between JD and PARAFAC can also be exploited for blind separation of convolutive mixtures in the time-frequency domain [8].

As a challenging generalization of JD, JBD is becoming a popular signal processing tool in applications such as BSS of convolutive mixtures in time-domain [9]–[12], independent subspace analysis (ISA) [13], [14] or blind localization of multiple targets in a multistatic MIMO radar system [15]. Existing JBD techniques can be categorized in two groups:

1) *Unitary JBD* [9]–[11], [16], [17], where $\mathbf{A}$ is assumed square and unitary and is sought as the matrix that makes the matrices $\{\mathbf{A}^\ddagger\mathbf{X}_k\mathbf{A}\}_{k=1}^K$ jointly as block diagonal as possible. This is achieved via the criterion

$$\max_{\mathbf{A}} \sum_{k=1}^K \left\|\mathrm{bdiag}(\mathbf{A}^\ddagger\mathbf{X}_k\mathbf{A})\right\|_F^2 \quad (2)$$

or

$$\min_{\mathbf{A}} \sum_{k=1}^K \left\|\mathrm{boff}(\mathbf{A}^\ddagger\mathbf{X}_k\mathbf{A})\right\|_F^2 \quad (3)$$

Manuscript received October 29, 2010; revised March 23, 2011 and June 26, 2011; accepted June 27, 2011. Date of publication July 12, 2011; date of current version September 14, 2011. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Peter J. Schreier. This work was supported by the Research Council K.U.Leuven: GOA-Ambiorics, GOA-MaNet, CoE EF/05/006 Optimization in Engineering (OPTEC), CIF1, STRT1/08/023, by the F.W.O.: (a) projects G.0321.06 and G.0427.10N, (b) Research Communities ICCoS, ANMMM, and MLDM, by the Belgian Federal Science Policy Office: IUAP P6/04 (DYSCO, "Dynamical systems, control and optimization", 2007–2011), and by the EU: ERNSI.

The author is with the Group Science, Engineering and Technology, K.U. Leuven, Campus Kortrijk, 8500 Kortrijk, Belgium (e-mail: dimitri.nion@gmail.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2011.2161473

where $\|\cdot\|_F$ denotes the Frobenius norm, $\mathrm{bdiag}(\mathbf{M})$ is the block diagonal matrix defined by

$$\mathrm{bdiag}(\mathbf{M}) = \begin{bmatrix} \mathbf{M}_{11} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathbf{M}_{RR} \end{bmatrix}$$

given the $N \times N$ matrix $\mathbf{M} \stackrel{\text{def}}{=} [\mathbf{M}_{ij}]_{1 \le i,j \le R}$ ($\mathbf{M}_{ij}$ being $L_i \times L_j$ matrices) and $\mathrm{boff}(\mathbf{M}) \stackrel{\text{def}}{=} \mathbf{M} - \mathrm{bdiag}(\mathbf{M})$.

2) *Nonunitary JBD*, with $I \ge N$ and $\mathbf{A}$ full column rank, such that it admits a left pseudoinverse denoted by $\mathbf{B} \stackrel{\text{def}}{=} \mathbf{A}^\dagger \in \mathbb{K}^{N \times I}$. The matrices $\{\mathbf{X}_k\}_{k=1}^K$ may additionally be assumed positive definite [18] or not [12]. In the latter case, it was proposed in [12] to seek for the joint block diagonalizer $\mathbf{B}$ with the criterion

$$\min_{\mathbf{B}} \phi_{\text{off}} \stackrel{\text{def}}{=} \sum_{k=1}^{K} \left\| \mathrm{boff}(\mathbf{B}\mathbf{X}_k\mathbf{B}^\ddagger) \right\|_F^2 \tag{4}$$

for which two algorithms have been proposed: $\mathrm{JBD}_{\mathrm{OG}}$, based on gradient descent with optimal step and $\mathrm{JBD}_{\mathrm{ORG}}$ based on the relative gradient with optimal step.

However, the minimization of ϕ_{off} in the nonunitary JBD case may be problematic for the following reasons:

- additional constraints have to be embedded in the optimization strategy to avoid the trivial solution $\mathbf{B} = \mathbf{0}$ and degenerate solutions [19];
- it cannot handle the underdetermined case $I < N$; to the best of our knowledge, none of the existing JBD algorithms can handle this case;
- it does not guarantee essential uniqueness of the solution in the overdetermined case $I > N$, as explained in the following. Let $\mathbf{A} \in \mathbb{K}^{I \times N}$ be a full column rank matrix, $I > N$, and $\mathbf{B} = \mathbf{A}^\dagger$ such that $\|\mathrm{boff}(\mathbf{B}\mathbf{X}_k\mathbf{B}^T)\|_F^2 = 0, \forall k$. Substitute $\mathbf{B} = [\mathbf{B}_1^T, \mathbf{B}_2^T]^T$ by $\tilde{\mathbf{B}} = [\mathbf{A}^\perp, \mathbf{B}_2^T]^T$, where the rows of $\mathbf{A}^\perp$ live in the orthogonal complement of $\mathrm{Span}(\mathbf{A})$. It follows that $\tilde{\mathbf{B}}$ is also solution of (4) but it is not solution of the JBD problem since $\tilde{\mathbf{B}}\mathbf{A}$ has a zero-block on its diagonal. Also, if one substitutes $\mathbf{B}$ by $\tilde{\mathbf{B}} = [\mathbf{B}_1^T + \mathbf{A}^\perp, \mathbf{B}_2^T]^T$, the matrix $\tilde{\mathbf{B}}\mathbf{A}$ remains unchanged but essential uniqueness of the solution is lost.

For these well-motivated reasons, we will investigate the nonunitary JBD problem through the JBD subspace-fitting least squares criterion

$$\min_{\{\mathbf{A}_r, \mathbf{D}_{kr}\}_{r,k=1}^{R,K}} \phi_{\text{LS}} \stackrel{\text{def}}{=} \sum_{k=1}^{K} \left\| \mathbf{X}_k - \sum_{r=1}^{R} \mathbf{A}_r \mathbf{D}_{kr} \mathbf{A}_r^\ddagger \right\|_F^2. \tag{5}$$

The main contributions of this paper are the following:

i) we show that JBD can be compactly written as a particular case of the block-component-decomposition in rank-$(L_r, M_r, \cdot)$ terms of a third-order tensor [20], [21], denoted by BCD-$(L_r, M_r, \cdot)$. This tensor-based reformulation of JBD allows immediate use of powerful results concerning essential uniqueness of the BCD-$(L_r, M_r, \cdot)$ to establish a set of sufficient conditions for which JBD is guaranteed to be essentially unique;
ii) we elaborate a nonlinear conjugate gradient (NCG) algorithm with exact line search for efficiently solving (5), that works in the over-, under-, and exactly determined cases;
iii) in the exactly and overdetermined cases ($I \ge N$) we propose a closed-form solution to the exact JBD problem, based on the generalized eigenvalue decomposition. If the JBD problem is not exact (e.g., when $\mathbf{X}_k$ is perturbed by additive noise), this technique can be used to find a good starting point for any JBD algorithm;
iv) extensive numerical experiments, including a comparison with $\mathrm{JBD}_{\mathrm{OG}}$ and $\mathrm{JBD}_{\mathrm{ORG}}$ and an ISA-based application are conducted to illustrate our findings.

Throughout this paper, we will distinguish between the three following cases.

Case C1: The data are real-valued, i.e., $\mathbb{K} = \mathbb{R}$;
Case C2: The data are complex-valued, i.e., $\mathbb{K} = \mathbb{C}$, and hermitian symmetry is assumed, i.e., $\mathbf{X}_k = \mathbf{A}\mathbf{D}_k\mathbf{A}^H + \mathbf{N}_k$;
Case C3: The data are complex-valued, and symmetry is assumed, i.e., $\mathbf{X}_k = \mathbf{A}\mathbf{D}_k\mathbf{A}^T + \mathbf{N}_k$.

This paper is organized as follows. In Section II, the JBD model (1) is rewritten in tensor format. In Section III, the algebraic expression of the gradient of ϕ_{LS} is derived. In Section IV, we build a NCG algorithm to solve (5). In Section V, we propose a closed form solution to over- and exactly determined JBD problems. In Section VI, we introduce a new performance index for evaluation of JBD algorithms. Section VII consists of numerical experiments and Section VIII summarizes our conclusions.

II. Tensor Formulation of JBD

In this section, we show that the JBD problem can be seen as a particular case of the BCD-$(L_r, M_r, \cdot)$. This link is established for the case C2 but the derivation is similar for C1 and C3 (it suffices to substitute $\mathbf{A}^*$ by $\mathbf{A}$ in the second mode).

A. JBD as a Tensor Decomposition

We first need the following definition.

Definition 1. (Mode-n Tensor-Matrix Product): The mode-1 product of a third-order tensor $\mathcal{X} \in \mathbb{C}^{L \times M \times N}$ by a matrix $\mathbf{A} \in \mathbb{C}^{I \times L}$, denoted by $\mathcal{X} \bullet_1 \mathbf{A}$, is an $(I \times M \times N)$-tensor with elements defined, for all index values, by $(\mathcal{X} \bullet_1 \mathbf{A})_{imn} = \sum_{l=1}^{L} x_{lmn} a_{il}$. Similarly, the mode-2 product by a matrix $\mathbf{B} \in \mathbb{C}^{J \times M}$ and the mode-3 product by $\mathbf{C} \in \mathbb{C}^{K \times N}$ are the $(L \times J \times N)$ and $(L \times M \times K)$ tensors, respectively, with elements defined by $(\mathcal{X} \bullet_2 \mathbf{B})_{ljn} = \sum_{m=1}^{M} x_{lmn} b_{jm}$ and $(\mathcal{X} \bullet_3 \mathbf{C})_{lmk} = \sum_{n=1}^{N} x_{lmn} c_{kn}$.

Denote by $\mathcal{X} \in \mathbb{K}^{I \times I \times K}$, $\mathcal{N} \in \mathbb{K}^{I \times I \times K}$, $\mathcal{D} \in \mathbb{K}^{N \times N \times K}$ the third-order tensors built by stacking the matrices $\{\mathbf{X}_k\}_{k=1}^K$, $\{\mathbf{N}_k\}_{k=1}^K$, $\{\mathbf{D}_k\}_{k=1}^K$, respectively, along the third dimension. The JBD model (1) can be written in tensor format (see Fig. 1) as follows:

$$\mathcal{X} = \mathcal{D} \bullet_1 \mathbf{A} \bullet_2 \mathbf{A}^* + \mathcal{N}. \tag{6}$$

Since the K slices of $\mathcal{D}$ are exactly block diagonal, (6) is equivalent to

$$\mathcal{X} = \sum_{r=1}^{R} \mathcal{D}_r \bullet_1 \mathbf{A}_r \bullet_2 \mathbf{A}_r^* + \mathcal{N} \tag{7}$$

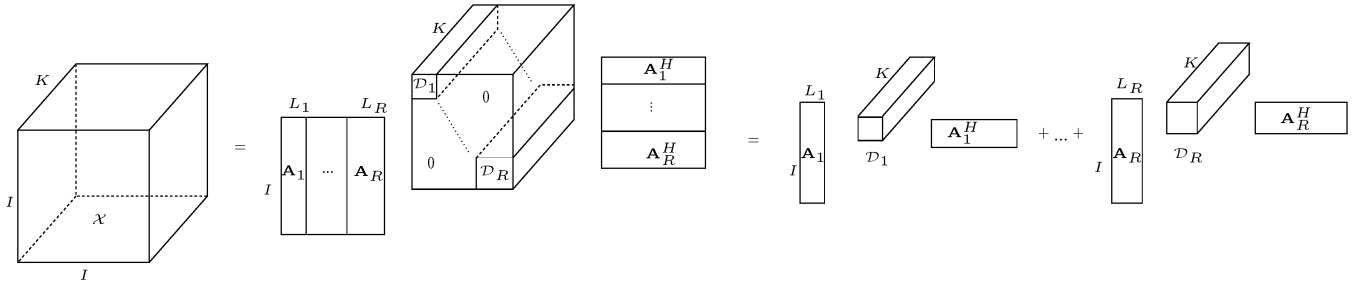

Fig. 1. The JBD problem in tensor format. The submatrices $\{\mathbf{A}_r\}_{r=1}^R \in \mathbb{K}^{I\times L_r}$ are assumed full column rank. The matrix $\mathbf{A} = [\mathbf{A}_1, \ldots, \mathbf{A}_R] \in \mathbb{C}^{I\times N}$, $N = \sum_{r=1}^R L_r$, may be fat or tall and is assumed full rank.

where $\mathcal{D}_r \in \mathbb{K}^{L_r\times L_r\times K}$ is built by stacking the matrices $\{\mathbf{D}_{kr}\}_{k=1}^K$ along the third dimension. Equation (7) is equivalent to

$$\mathbf{X} = (\mathbf{A}^* \odot \mathbf{A})\mathbf{D} + \mathbf{N} \tag{8}$$

where $\odot$ denotes the block-wise Kronecker product

$$(\mathbf{A}^* \odot \mathbf{A}) = [\mathbf{A}_1^* \otimes \mathbf{A}_1, \ldots, \mathbf{A}_R^* \otimes \mathbf{A}_R]$$

$\otimes$ is the Kronecker product, $\mathbf{X} \in \mathbb{K}^{I^2\times K}$ is built from $\mathcal{X}$ as follows: $[\mathbf{X}]_{(i_2-1)I+i_1,k} \stackrel{\text{def}}{=} [\mathcal{X}]_{i_1,i_2,k}$, $\mathbf{N}$ is built from $\mathcal{N}$ in the same way, and $\mathbf{D}$ is the $\tilde{L}\times K$ matrix, $\tilde{L} \stackrel{\text{def}}{=} \sum_{r=1}^R L_r^2$, built as follows:

$$\mathbf{D} \stackrel{\text{def}}{=} \begin{bmatrix} \text{vec}(\mathbf{D}_{11}) & \cdots & \text{vec}(\mathbf{D}_{K1}) \\ \vdots & & \vdots \\ \text{vec}(\mathbf{D}_{1R}) & \cdots & \text{vec}(\mathbf{D}_{KR}) \end{bmatrix} \tag{9}$$

where $\text{vec}(\cdot)$ is the operator that stacks the columns of a matrix one after each other in a single vector. Hence, the JBD criterion (5) is equivalent to

$$\begin{aligned} &\min_{\mathbf{A},\mathbf{D}} \phi_{\text{LS}} = \|\mathbf{N}\|_F^2 \\ &\text{where} \quad \mathbf{N} \stackrel{\text{def}}{=} \mathbf{X} - (\mathbf{A}^* \odot \mathbf{A})\mathbf{D}. \end{aligned} \tag{10}$$

B. Uniqueness of JBD

The Block Component Decompositions (BCD) of a third-order tensor have been introduced in [21] and can be seen as a generalization of the PARAFAC decomposition. The BCD of a third-order tensor $\mathcal{X} \in \mathbb{K}^{I\times J\times K}$ in a sum of R rank-$(L_r, M_r, \cdot)$ terms, denoted by BCD-$(L_r, M_r, \cdot)$, is written as

$$\mathcal{X} = \sum_{r=1}^R \mathcal{D}_r \bullet_1 \mathbf{A}_r \bullet_2 \mathbf{B}_r \tag{11}$$

where $\mathcal{D}_r \in \mathbb{K}^{L_r\times M_r\times K}$, $\mathbf{A}_r \in \mathbb{K}^{I\times L_r}$ is rank-L_r and $\mathbf{B}_r \in \mathbb{K}^{J\times M_r}$ is rank-M_r. Let $\mathbf{A} = [\mathbf{A}_1, \ldots, \mathbf{A}_R] \in \mathbb{K}^{I\times N}$ and $\mathbf{B} = [\mathbf{B}_1, \ldots, \mathbf{B}_R] \in \mathbb{K}^{J\times Q}$, $N = \sum_{r=1}^R L_r$, $Q = \sum_{r=1}^R M_r$. The following theorem has been derived in [21].

Theorem 1: Suppose that $I \geq N$, $J \geq Q$, $\text{rank}(\mathbf{A}) = N$, $\text{rank}(\mathbf{B}) = Q$, $K \geq 3$, and that $\{\mathcal{D}_r\}_{r=1}^R$ are generic, then the BCD-$(L_r, M_r, \cdot)$ of $\mathcal{X}$ (11) is essentially unique.

We call a tensor generic when its entries can be considered drawn from continuous probability density functions.

From the previous section, it follows that JBD is a particular case of the BCD-$(L_r, M_r, \cdot)$ where $\mathbf{B}_r$ is substituted by $\mathbf{A}_r^*$ or $\mathbf{A}_r$. It can easily be checked in [21] that Theorem 1 remains valid in this case. Note that the uniqueness condition given by Theorem 1 is only sufficient. In some cases where it is not satisfied, e.g., when $\mathbf{A}$ is full row rank rather than full column rank, uniqueness can possibly be still guaranteed but is more difficult to prove. The uniqueness issue when Theorem 1 is not satisfied would deserve further work that is beyond the scope of this paper.

III. Computation of the Gradient

In this section, we derive the algebraic expression of the gradient of ϕ_{LS} w. r. t. $\mathbf{A}$ and $\mathbf{D}$ in the cases C1, C2 and C3.

A. Case C1: Real Data

From (1) and (5), $\phi_{\text{LS}} = \sum_{k=1}^K \text{Tr}(\mathbf{N}_k^T\mathbf{N}_k)$ with $\mathbf{N}_k = \mathbf{X}_k - \mathbf{A}\mathbf{D}_k\mathbf{A}^T$, where $\text{Tr}(\cdot)$ denotes the trace of a matrix. Derivatives of traces [22] yield

$$\nabla_{\mathbf{A}}(\phi_{\text{LS}}) = -2\sum_{k=1}^K \left(\mathbf{N}_k^T\mathbf{A}\mathbf{D}_k + \mathbf{N}_k\mathbf{A}\mathbf{D}_k^T\right) \tag{12}$$

where $\nabla_{\mathbf{A}}(\phi_{\text{LS}})$ denotes the gradient of ϕ_{LS} w. r. t. $\mathbf{A}$, $[\nabla_{\mathbf{A}}(\phi_{\text{LS}})]_{i,n} \stackrel{\text{def}}{=} \partial\phi_{\text{LS}}/\partial\mathbf{A}_{i,n}$, $i = 1, \ldots, I$, $n = 1, \ldots, N$. From (10), ϕ_{LS} can be written as $\phi_{\text{LS}} = \text{Tr}(\mathbf{N}^T\mathbf{N})$ with $\mathbf{N} = \mathbf{X} - (\mathbf{A} \odot \mathbf{A})\mathbf{D}$ and one gets

$$\nabla_{\mathbf{D}}(\phi_{\text{LS}}) = -2(\mathbf{A} \odot \mathbf{A})^T\mathbf{N} \tag{13}$$

where $[\nabla_{\mathbf{D}}(\phi_{\text{LS}})]_{\tilde{l},k} \stackrel{\text{def}}{=} \partial\phi_{\text{LS}}/\partial\mathbf{D}_{\tilde{l},k}$, $\tilde{l} = 1, \ldots, \tilde{L}$, $k = 1, \ldots, K$.

B. Case C2: Complex Data, Hermitian Symmetry

Let us write $\mathbf{A} = \mathbf{A}^{(\mathcal{R})} + j\mathbf{A}^{(\mathcal{I})}$ where $\mathbf{A}^{(\mathcal{R})} = \text{Re}(\mathbf{A})$ denotes the real part of $\mathbf{A}$ and $\mathbf{A}^{(\mathcal{I})} = \text{Im}(\mathbf{A})$ its imaginary part. Similarly, we write $\mathbf{D} = \mathbf{D}^{(\mathcal{R})} + j\mathbf{D}^{(\mathcal{I})}$. Derivatives of traces [22] yield

$$\nabla_{\mathbf{A}^{(\mathcal{R})}}(\phi_{\text{LS}}) = -2\text{Re}\left(\sum_{k=1}^K \left(\mathbf{N}_k^H\mathbf{A}\mathbf{D}_k + \mathbf{N}_k\mathbf{A}\mathbf{D}_k^H\right)\right) \tag{14}$$

$$\nabla_{\mathbf{A}^{(\mathcal{I})}}(\phi_{\text{LS}}) = -2\text{Im}\left(\sum_{k=1}^K \left(\mathbf{N}_k^H\mathbf{A}\mathbf{D}_k + \mathbf{N}_k\mathbf{A}\mathbf{D}_k^H\right)\right) \tag{15}$$

$$\nabla_{\mathbf{D}^{(\mathcal{R})}}(\phi_{\text{LS}}) = -2\text{Re}\left((\mathbf{A} \odot \mathbf{A}^*)^T\mathbf{N}\right) \tag{16}$$

$$\nabla_{\mathbf{D}^{(\mathcal{I})}}(\phi_{\text{LS}}) = -2\text{Im}\left((\mathbf{A} \odot \mathbf{A}^*)^T\mathbf{N}\right). \tag{17}$$

The directions of the maximum rate of change in the real-valued cost function ϕ_{LS} w. r. t. the complex variables $\mathbf{A}$ and $\mathbf{D}$ are given by the gradients w. r. t. to the conjugate of $\mathbf{A}$ and $\mathbf{D}$ [23], [24]:

$$\begin{aligned}\nabla_{\mathbf{A}^*}(\phi_{\mathrm{LS}}) &= \frac{1}{2}\left(\nabla_{\mathbf{A}^{(\mathcal{R})}}(\phi_{\mathrm{LS}}) + j\nabla_{\mathbf{A}^{(I)}}(\phi_{\mathrm{LS}})\right) \\ &= -\sum_{k=1}^{K}\left(\mathbf{N}_k^H\mathbf{A}\mathbf{D}_k + \mathbf{N}_k\mathbf{A}\mathbf{D}_k^H\right) \end{aligned} \tag{18}$$

and

$$\begin{aligned}\nabla_{\mathbf{D}^*}(\phi_{\mathrm{LS}}) &= \frac{1}{2}\left(\nabla_{\mathbf{D}^{(\mathcal{R})}}(\phi_{\mathrm{LS}}) + j\nabla_{\mathbf{D}^{(I)}}(\phi_{\mathrm{LS}})\right) \\ &= -(\mathbf{A}\odot\mathbf{A}^*)^T\mathbf{N}. \end{aligned} \tag{19}$$

C. Case C3: Complex Data, Symmetry

Similarly, one can show that, in the symmetric case

$$\nabla_{\mathbf{A}^*}(\phi_{\mathrm{LS}}) = -\sum_{k=1}^{K}\left(\mathbf{N}_k^T\mathbf{A}^*\mathbf{D}_k^* + \mathbf{N}_k\mathbf{A}^*\mathbf{D}_k^H\right) \tag{20}$$

and

$$\nabla_{\mathbf{D}^*}(\phi_{\mathrm{LS}}) = -(\mathbf{A}\odot\mathbf{A})^H\mathbf{N}. \tag{21}$$

IV. A NCG Algorithm

A. Algorithm Overview

In this section, we propose a NCG algorithm to solve (5). NCG has become a popular technique in nonlinear optimization; it converges faster than the steepest descent method and has lower complexity than Newton or Quasi-Newton methods [25], [26]. In the context of tensor decompositions, such an optimization technique has been proposed for fitting the PARAFAC model in [27]. Denote by $\mathbf{u}$ the $(NI+K\tilde{L})\times 1$ vector in which all unknowns have been stacked as follows

$$\mathbf{u} \stackrel{\mathrm{def}}{=} \left[\mathrm{vec}(\mathbf{A})^T, \mathrm{vec}(\mathbf{D})^T\right]^T. \tag{22}$$

It follows that:

$$\nabla_{\mathbf{u}^*} = \left[\mathrm{vec}(\nabla_{\mathbf{A}^*})^T, \mathrm{vec}(\nabla_{\mathbf{D}^*})^T\right] \tag{23}$$

where $\boldsymbol{\nabla}_{\mathbf{u}^*}$ denotes the gradient of ϕ_{LS} w. r. t. $\mathbf{u}^*$. The pth NCG iteration consists of the following steps:

$$\begin{cases} \text{Step 1.} & \text{Compute the steepest descent direction:} -\nabla_{\mathbf{u}^*}^{(p)}; \\ \text{Step 2.} & \text{Update the search direction: compute } \beta^{(p)} \text{ and } \mathbf{d}_{\mathbf{u}}^{(p)} \leftarrow -\nabla_{\mathbf{u}^*}^{(p)} + \beta^{(p)}\mathbf{d}_{\mathbf{u}}^{(p-1)}; \\ \text{Step 3.} & \text{Compute the step size } \alpha^{(p)}; \\ \text{Step 4.} & \text{Update } \mathbf{u}: \mathbf{u}^{(p+1)} \leftarrow \mathbf{u}^{(p)} + \alpha^{(p)}\mathbf{d}_{\mathbf{u}}^{(p)}. \end{cases}$$

The first iteration is made in the steepest descent direction, i.e., $\mathbf{d}_{\mathbf{u}}^{(0)} = -\boldsymbol{\nabla}_{\mathbf{u}^*}^{(0)}$. The gradient in Step 1 is given by the algebraic expressions derived in Section III. In the following, we explain how Steps 2 and 3 can be adapted to the JBD optimization problem (5), (10).

B. Update of the Search Direction

A crucial point in the update strategy of the search direction $\mathbf{d}_{\mathbf{u}}$ is the choice of the real-valued scalar β. Numerous methods have been proposed in the specialized literature, see [25] and [26] for a survey. A popular choice is the stabilized Polak-RibiÃĺre formula

$$\beta^{(p)} = \max\left\{\frac{\mathrm{Re}\left(\left(\nabla_{\mathbf{u}^*}^{(p)}\right)^H\left(\nabla_{\mathbf{u}^*}^{(p)} - \nabla_{\mathbf{u}^*}^{(p-1)}\right)\right)}{\left\|\nabla_{\mathbf{u}^*}^{(p-1)}\right\|^2}, 0\right\}. \tag{24}$$

In practice, NCG algorithms are often coupled with a restart strategy, i.e., the value $\beta^{(p)} = 0$ is enforced if a restart criterion is satisfied, such that the algorithm is refreshed: the search direction is reset to a standard steepest descent direction. Consider a cost function that is strongly convex quadratic in a neighborhood of the solution, but nonquadratic everywhere else. The most popular restart strategy makes use of the observation that the gradients are mutually orthogonal when the cost function is quadratic. Thus, a restart can be performed whenever two consecutive gradients are far from orthogonal, as measured by the test

$$\frac{\left|\mathrm{Re}\left(\left(\nabla_{\mathbf{u}^*}^{(p)}\right)^H\nabla_{\mathbf{u}^*}^{(p-1)}\right)\right|}{\left\|\nabla_{\mathbf{u}^*}^{(p)}\right\|^2} \geq \nu \tag{25}$$

where a typical value for the parameter ν is 0.1 [26].

C. Exact Line Search

Given the search direction $\mathbf{d}_{\mathbf{u}}^{(p)}$ at iteration p, it is crucial to find a good step-size α in this direction. Exact Line Search (ELS) consists of the computation of the *optimal* step:

$$\alpha^{(p)} \leftarrow \arg\min_{\alpha\in\mathbb{R}} \phi_{\mathrm{LS}}\left(\mathbf{u}^{(p)} + \alpha\mathbf{d}_{\mathbf{u}}^{(p)}\right). \tag{26}$$

If the line search is exact, $\left(\nabla_{\mathbf{u}^*}^{(p)}\right)^T\mathbf{d}_{\mathbf{u}}^{(p)} < 0$, so that $\mathbf{d}_{\mathbf{u}}^{(p)}$ is a descent direction.[1]

From the partitioning defined in (22) and (23), Step 2 is equivalent to

$$\begin{cases} \mathbf{d}_{\mathbf{A}}^{(p)} \leftarrow & -\nabla_{\mathbf{A}^*}^{(p)} + \beta^{(p)}\mathbf{d}_{\mathbf{A}}^{(p-1)} \\ \mathbf{d}_{\mathbf{D}}^{(p)} \leftarrow & -\nabla_{\mathbf{D}^*}^{(p)} + \beta^{(p)}\mathbf{d}_{\mathbf{D}}^{(p-1)} \end{cases}$$

where $\mathbf{d}_{\mathbf{A}}$ and $\mathbf{d}_{\mathbf{D}}$ are the search directions for $\mathbf{A}$ and $\mathbf{D}$, respectively. In the case C2

$$\phi_{\mathrm{LS}}(\alpha) = \left\|\left(\left(\mathbf{A}^* + \alpha\mathbf{d}_{\mathbf{A}}^*\right)\odot\left(\mathbf{A} + \alpha\mathbf{d}_{\mathbf{A}}\right)\right)\left(\mathbf{D} + \alpha\mathbf{d}_{\mathbf{D}}\right) - \mathbf{X}\right\|_F^2 \tag{27}$$

where $\phi_{\mathrm{LS}}(\alpha)$ stands for $\phi_{\mathrm{LS}}(\mathbf{u}^{(p)} + \alpha\mathbf{d}_{\mathbf{u}}^{(p)})$ and the superscript p has been omitted for simplicity. In cases C1 and C3, it suffices to substitute $\mathbf{A}^*$ and $\mathbf{d}_{\mathbf{A}}^*$ by $\mathbf{A}$ and $\mathbf{d}_{\mathbf{A}}$, respectively. It follows that $\phi_{\mathrm{LS}}(\alpha)$ is a polynomial of degree six in α and can easily be

[1] We have $\left(\nabla_{\mathbf{u}^*}^{(p)}\right)^T\mathbf{d}_{\mathbf{u}}^{(p)} = -\|\nabla_{\mathbf{u}^*}^{(p)}\|_F^2 + \beta^{(p)}\left(\nabla_{\mathbf{u}^*}^{(p)}\right)^T\mathbf{d}_{\mathbf{u}}^{(p-1)}$ with $\left(\nabla_{\mathbf{u}^*}^{(p)}\right)^T\mathbf{d}_{\mathbf{u}}^{(p-1)} = \phi'_{\mathrm{LS}}(\alpha) = 0$.

Algorithm 1 Nonlinear Conjugate Gradient algorithm for JBD, denoted $\mathrm{JBD_{NCG}}$.

Initialization: $\mathbf{A}^{(0)}$, $\mathbf{D}^{(0)}$, $p=0$, $p_{\max}$(e.g. $p_{\max}=2000$) , ϵ (e.g. $\epsilon=10^{-6}$), $N=\sum_{r=1}^{R} L_r$, $\tilde{L}=\sum_{r=1}^{R} L_r^2$.

First iteration in the steepest descent direction.

- Compute $\mathbf{d}_{\mathbf{A}}^{(0)} \leftarrow -\nabla_{\mathbf{A}^*}^{(0)}$ and $\mathbf{d}_{\mathbf{D}}^{(0)} \leftarrow -\nabla_{\mathbf{D}^*}^{(0)}$ from $\mathbf{A}^{(0)}$ and $\mathbf{D}^{(0)}$.
- Exact Line Search: $(\alpha_{\mathbf{A}}^{(0)}, \alpha_{\mathbf{D}}^{(0)}) \leftarrow \underset{\alpha_{\mathbf{A}},\alpha_{\mathbf{D}}}{\operatorname{argmin}} \phi_{\mathrm{LS}}(\mathbf{A}^{(0)}+\alpha_{\mathbf{A}}\mathbf{d}_{\mathbf{A}}^{(0)},\ \mathbf{D}^{(0)}+\alpha_{\mathbf{D}}\mathbf{d}_{\mathbf{D}}^{(0)})$.
- Updates:
  $\mathbf{D}^{(1)} \leftarrow \mathbf{D}^{(0)}+\alpha_{\mathbf{D}}^{(0)}\mathbf{d}_{\mathbf{D}}^{(0)}$
  $\mathbf{A}^{(1)} \leftarrow \mathbf{A}^{(0)}+\alpha_{\mathbf{A}}^{(0)}\mathbf{d}_{\mathbf{A}}^{(0)}$

Conjugate gradient updates.

While $\left(|\phi_{\mathrm{LS}}^{(p+1)}-\phi_{\mathrm{LS}}^{(p)}|/\phi_{\mathrm{LS}}^{(p)} > \epsilon\right)$ *and* $(p<p_{\max})$

$p \leftarrow p+1$

Step 1. Compute $\nabla_{\mathbf{A}^*}^{(p)}$ and $\nabla_{\mathbf{D}^*}^{(p)}$ from $\mathbf{A}^{(p)}$ and $\mathbf{D}^{(p)}$.

Complexity:

$4KI(\tilde{L}+IN)$ for $\nabla_{\mathbf{A}}^{(p)}$ in case C1 and $16KI(\tilde{L}+IN)$ for $\nabla_{\mathbf{A}^*}^{(p)}$ in cases C2 and C3;

$I^2(2\tilde{L}K+\tilde{L}+N)$ for $\nabla_{\mathbf{D}}^{(p)}$ in case C1 and $4I^2(2\tilde{L}K+\tilde{L}+N)$ for $\nabla_{\mathbf{D}^*}^{(p)}$ in cases C2 and C3.

Step 2. Update search directions.

Step 2.1. Stabilized Polak-RibiÃÍre formula: $\beta^{(p)} = \max\left\{ \frac{\mathrm{Re}\left[\left(\mathrm{vec}\left(\nabla_{\mathbf{A}^*}^{(p)}\right)\right)^H \mathrm{vec}\left(\nabla_{\mathbf{A}^*}^{(p)}-\nabla_{\mathbf{A}^*}^{(p-1)}\right)+\left(\mathrm{vec}\left(\nabla_{\mathbf{D}^*}^{(p)}\right)\right)^H \mathrm{vec}\left(\nabla_{\mathbf{D}^*}^{(p)}-\nabla_{\mathbf{D}^*}^{(p-1)}\right)\right]}{\|\nabla_{\mathbf{A}^*}^{(p-1)}\|^2+\|\nabla_{\mathbf{D}^*}^{(p-1)}\|^2}, 0 \right\}$.

Step 2.2. Restart strategy: *if* $\frac{\left|\mathrm{Re}\left[\left(\mathrm{vec}\left(\nabla_{\mathbf{A}^*}^{(p)}\right)\right)^H \mathrm{vec}\left(\nabla_{\mathbf{A}^*}^{(p-1)}\right)+\left(\mathrm{vec}\left(\nabla_{\mathbf{D}^*}^{(p)}\right)\right)^H \mathrm{vec}\left(\nabla_{\mathbf{D}^*}^{(p-1)}\right)\right]\right|}{\|\nabla_{\mathbf{A}^*}^{(p)}\|^2+\|\nabla_{\mathbf{D}^*}^{(p)}\|^2} \geq \nu$ (e.g. $\nu=0.1$), *then* $\beta^{(p)}=0$.

Step 2.3.

$\mathbf{d}_{\mathbf{A}}^{(p)} \leftarrow -\nabla_{\mathbf{A}^*}^{(p)}+\beta^{(p)}\mathbf{d}_{\mathbf{A}}^{(p-1)}$.

$\mathbf{d}_{\mathbf{D}}^{(p)} \leftarrow -\nabla_{\mathbf{D}^*}^{(p)}+\beta^{(p)}\mathbf{d}_{\mathbf{D}}^{(p-1)}$

Complexity for *Step* 2: $4(IN+K\tilde{L})$ for C1 and $16(IN+K\tilde{L})$ for C2 and C3.

Step 3. Exact Line Search: $(\alpha_{\mathbf{A}}^{(p)}, \alpha_{\mathbf{D}}^{(p)}) \leftarrow \underset{\alpha_{\mathbf{A}},\alpha_{\mathbf{D}}}{\operatorname{argmin}} \phi_{\mathrm{LS}}(\mathbf{A}^{(p)}+\alpha_{\mathbf{A}}\mathbf{d}_{\mathbf{A}}^{(p)},\ \mathbf{D}^{(p)}+\alpha_{\mathbf{D}}\mathbf{d}_{\mathbf{D}}^{(p)})$.

Complexity for *Step* 3: $3I^2(4\tilde{L}K+14K+\tilde{L}+N)$ for C1 and $12I^2(4\tilde{L}K+14K+\tilde{L}+N)$ for C2 and C3.

Step 4. Update $\mathbf{A}$ and $\mathbf{D}$:

$\mathbf{D}^{(p+1)} \leftarrow \mathbf{D}^{(p)}+\alpha_{\mathbf{D}}^{(p)}\mathbf{d}_{\mathbf{D}}^{(p)}$

$\mathbf{A}^{(p+1)} \leftarrow \mathbf{A}^{(p)}+\alpha_{\mathbf{A}}^{(p)}\mathbf{d}_{\mathbf{A}}^{(p)}$

End

Total complexity: $\begin{array}{c|c} \text{C1} & 2I^2K(7\tilde{L}+2N+21)+4I^2(\tilde{L}+N)+4IK\tilde{L}+4IN+4K\tilde{L} \\ \hline \text{C2 and C3} & 8I^2K(7\tilde{L}+2N+21)+16I^2(\tilde{L}+N)+16IK\tilde{L}+16IN+16K\tilde{L} \end{array}$

minimized. This ELS strategy can be improved, without significant additional complexity, by seeking for two different optimal steps $\alpha_{\mathbf{A}} \in \mathbb{R}$ and $\alpha_{\mathbf{D}} \in \mathbb{R}$ in the search directions $\mathbf{d}_{\mathbf{A}}$ and $\mathbf{d}_{\mathbf{D}}$, respectively. Then, (27) becomes

$$\phi_{\mathrm{LS}}(\alpha_{\mathbf{A}},\alpha_{\mathbf{D}}) = \|((\mathbf{A}^*+\alpha_{\mathbf{A}}\mathbf{d}_{\mathbf{A}}^*)\odot(\mathbf{A}+\alpha_{\mathbf{A}}\mathbf{d}_{\mathbf{A}})) \cdot(\mathbf{D}+\alpha_{\mathbf{D}}\mathbf{d}_{\mathbf{D}})-\mathbf{X}\|_F^2. \quad (28)$$

Let us build the $KI^2 \times 3$ matrices $\mathbf{W}_1$ and $\mathbf{W}_0$ as follows:

$$\mathbf{W}_1 \overset{\mathrm{def}}{=} [\mathrm{vec}((\mathbf{d}_{\mathbf{A}}^*\odot\mathbf{d}_{\mathbf{A}})\mathbf{d}_{\mathbf{D}}), \mathrm{vec}((\mathbf{A}^*\odot\mathbf{d}_{\mathbf{A}})\mathbf{d}_{\mathbf{D}}+(\mathbf{d}_{\mathbf{A}}^*\odot\mathbf{A})\mathbf{d}_{\mathbf{D}}), \mathrm{vec}((\mathbf{A}^*\odot\mathbf{A})\mathbf{d}_D)], \quad (29)$$

$$\mathbf{W}_0 \overset{\mathrm{def}}{=} [\mathrm{vec}((\mathbf{d}_{\mathbf{A}}^*\odot\mathbf{d}_{\mathbf{A}})\mathbf{D}), \mathrm{vec}((\mathbf{A}^*\odot\mathbf{d}_{\mathbf{A}})\mathbf{D}+(\mathbf{d}_{\mathbf{A}}^*\odot\mathbf{A})\mathbf{D}), \mathrm{vec}((\mathbf{A}^*\odot\mathbf{A})\mathbf{D}-\mathbf{X})]. \quad (30)$$

It is a matter of standard algebraic manipulations to show that

$$\phi_{\mathrm{LS}}(\alpha_{\mathbf{A}},\alpha_{\mathbf{D}}) = \alpha_{\mathbf{D}}^2 Q_2(\alpha_{\mathbf{A}})+2\alpha_{\mathbf{D}}Q_1(\alpha_{\mathbf{A}})+Q_0(\alpha_{\mathbf{A}}) \quad (31)$$

where

$$\begin{cases} Q_2(\alpha_{\mathbf{A}}) \overset{\mathrm{def}}{=} \mathbf{u}(\alpha_{\mathbf{A}})^T\mathbf{W}_1^H\mathbf{W}_1\mathbf{u}(\alpha_{\mathbf{A}}) \\ Q_1(\alpha_{\mathbf{A}}) \overset{\mathrm{def}}{=} \mathbf{u}(\alpha_{\mathbf{A}})^T\mathrm{Re}\left[\mathbf{W}_1^H\mathbf{W}_0\right]\mathbf{u}(\alpha_{\mathbf{A}}) \\ Q_0(\alpha_{\mathbf{A}}) \overset{\mathrm{def}}{=} \mathbf{u}(\alpha_{\mathbf{A}})^T\mathbf{W}_0^H\mathbf{W}_0\mathbf{u}(\alpha_{\mathbf{A}}) \end{cases}$$

are polynomials of degree four in $\alpha_{\mathbf{A}}$ and

$$\mathbf{u}(\alpha_{\mathbf{A}}) \overset{\mathrm{def}}{=} \left[\alpha_{\mathbf{A}}^2, \alpha_{\mathbf{A}}, 1\right]^T. \quad (32)$$

Solving $\partial\phi_{\mathrm{LS}}(\alpha_{\mathbf{A}},\alpha_{\mathbf{D}})/\partial\alpha_{\mathbf{D}}=0$ yields

$$\alpha_{\mathbf{D}} = -\frac{Q_1(\alpha_{\mathbf{A}})}{Q_2(\alpha_{\mathbf{A}})} \quad (33)$$

and substitution of (33) in (31) yields

$$\phi_{\mathrm{LS}}(\alpha_{\mathbf{A}}) = -\frac{Q_1^2(\alpha_{\mathbf{A}})}{Q_2(\alpha_{\mathbf{A}})}+Q_0(\alpha_{\mathbf{A}}) \quad (34)$$

which depends on the variable $\alpha_{\mathbf{A}}$ only. Finally, $\alpha_{\mathbf{A}}$ is estimated as the real root of $\partial\phi_{\mathrm{LS}}(\alpha_{\mathbf{A}})/\partial\alpha_{\mathbf{A}}$ that minimizes $\phi_{\mathrm{LS}}(\alpha_{\mathbf{A}})$, after which $\alpha_{\mathbf{D}}$ is given by (33).

The resulting $\mathrm{JBD_{NCG}}$ algorithm is summarized in Algorithm 1, with the complexity associated to each step expressed in terms of Real FLoating point OPeration (flop) counts. For instance, the scalar product of d dimensional real, respectively, complex, vectors involves $2d$, respectively, $8d$, flops.

V. A Closed Form Solution to Exact JBD Problems

In this section, we propose a closed form solution to the exact (i.e., noise-free) JBD problem, for the exactly determined case $(I=N)$ and the overdetermined case $(I>N)$. Consider the exact JBD model

$$\mathbf{X}_k = \mathbf{A}\mathbf{D}_k\mathbf{A}^T \quad (35)$$

(the case C1 is considered but the method can be derived in the same way for cases C2 and C3). Assume that $\mathbf{A}$ is rank-N and

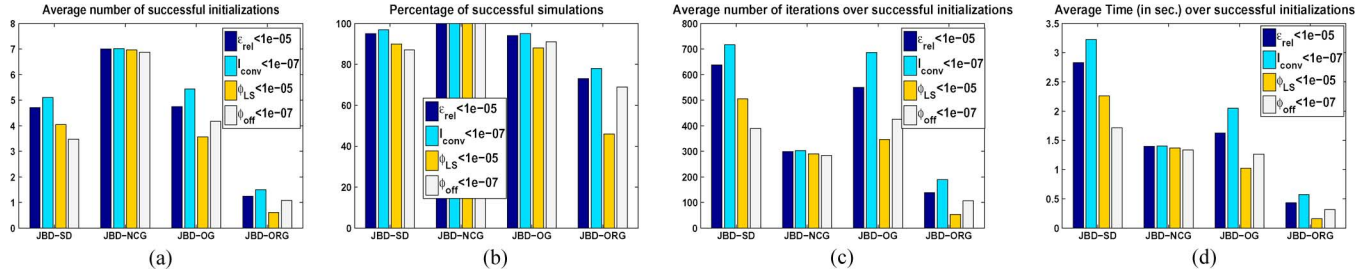

Fig. 2. Exactly determined case, $I = N = 9$, $R = 3$, $L_1 = L_2 = L_3 = 3$, $K = 30$, case C1. Criteria from left to right: $\epsilon_{\text{rel}} < 10^{-5}$, $\text{I}_{\text{conv}} < 10^{-7}$, $\phi_{\text{LS}} < 10^{-5}$, $\phi_{\text{off}} < 10^{-7}$. Algorithms from left to right: JBD_{SD}, JBD_{NCG}, JBD_{OG}, JBD_{ORG}. (a) Successful initializations. (b) Successful runs. (c) Number of iterations. (d) Time in sec.

that there exists at least two indices k_1 and k_2, such that $\mathbf{D}_{k_1}$ and $\mathbf{D}_{k_2}$ are nonsingular. Let us build the matrix

$$\mathbf{X}_{k_1 k_2} \stackrel{\text{def}}{=} \mathbf{X}_{k_1}\mathbf{X}_{k_2}^{\dagger} = \mathbf{A}\left(\mathbf{D}_{k_1}\mathbf{D}_{k_2}^{-1}\right)\mathbf{A}^{\dagger}. \tag{36}$$

The proposed technique relies on the generalized eigenvalue decomposition (GEVD) of $(\mathbf{X}_{k_1}, \mathbf{X}_{k_2})$. From (36), the subspace spanned by the columns of $\mathbf{A}$, denoted by $\text{Span}(\mathbf{A})$, consists of the N eigenvectors of $\mathbf{X}_{k_1}\mathbf{X}_{k_2}^{\dagger}$ associated to the N nonzero eigenvalues. Denoting by $\mathbf{E}$ the matrix formed by these eigenvectors, $\mathbf{A}$ can be written (up to the unresolvable JBD ambiguities) as

$$\mathbf{A} = \mathbf{E}\mathbf{\Pi} \tag{37}$$

where $\mathbf{\Pi}$ is an $N \times N$ a priori unknown permutation matrix that groups the eigenvectors L_r by L_r. Let us build the matrices

$$\mathbf{P}_k = \mathbf{E}^{\dagger}\mathbf{X}_k(\mathbf{E}^{\dagger})^T = \mathbf{\Pi}\mathbf{D}_k\mathbf{\Pi}^T, \quad k = 1, \ldots, K. \tag{38}$$

The matrix $\mathbf{\Pi}$ can be found by building $\mathbf{P} = (1/K)\sum_{k=1}^{K}|\mathbf{P}_k|$ and searching the position of the L nonzero elements of each normalized row of $\mathbf{P}$.

For approximate JBD problems, this closed form solution can be used as a good starting point of iterative JBD algorithms, as illustrated in Section VII. It suffices to select the eigenvectors of $\mathbf{X}_{k_1}\mathbf{X}_{k_2}^{\dagger}$ associated to the N most significant eigenvalues and to find the position of the L most significant values on each normalized row of $\mathbf{P}$.

VI. Performance Index

The performance index used in [12] is

$$\text{I}_{\text{conv}} \stackrel{\text{def}}{=} \frac{1}{R(R-1)}\left[\sum_{i=1}^{R}\left(\sum_{j=1}^{R}\frac{\|(\mathbf{G})_{ij}\|_F^2}{\max_l \|(\mathbf{G})_{il}\|_F^2} - 1\right) + \sum_{j=1}^{R}\left(\sum_{i=1}^{R}\frac{\|(\mathbf{G})_{ij}\|_F^2}{\max_l \|(\mathbf{G})_{lj}\|_F^2} - 1\right)\right] \tag{39}$$

where $\mathbf{G} = \mathbf{B}\hat{\mathbf{A}}$, $\hat{\mathbf{B}}$ is an estimate of $\mathbf{B}$ and $(\mathbf{G})_{ij}$ is the (i,j)-th square block matrix of $\mathbf{G}$. As explained in the introduction, combination of the optimization criterion (4) with the performance index I_{conv} may be misleading since it hides the possible loss of essential uniqueness.[2] For instance, the matrix $\mathbf{G}$ remains unchanged in the overdetermined case if $\hat{\mathbf{B}} = [\hat{\mathbf{B}}_1^T, \hat{\mathbf{B}}_2^T]^T$ is substituted by $\hat{\mathbf{B}} = [\hat{\mathbf{B}}_1^T + \mathbf{A}^{\perp}, \hat{\mathbf{B}}_2^T]^T$. Moreover, I_{conv} cannot be used in the underdetermined case.

In order to check that the optimization strategy provides an estimate of $\mathbf{A}$ (or its pseudo-inverse) only up to the ambiguities inherent to the JBD model, one can use another performance index, described in the following. Denote by $\hat{\mathbf{A}}$ an estimate of $\mathbf{A}$. Due to the JBD ambiguities, in case of perfect estimation, $\mathbf{A}$ and $\hat{\mathbf{A}}$ are linked as follows:

$$\mathbf{A} = \hat{\mathbf{A}}\mathbf{D}\mathbf{P}$$

where $\mathbf{D}$ is an unknown nonsingular block-diagonal matrix and $\mathbf{P}$ an unknown block-wise permutation matrix. The objective is to estimate $\mathbf{D}$ and $\mathbf{P}$ such that $\tilde{\mathbf{A}} \stackrel{\text{def}}{=} \hat{\mathbf{A}}\mathbf{D}\mathbf{P}$ "matches" $\mathbf{A}$, after which the relative error is defined by

$$\epsilon_{\text{rel}} = \frac{\|\mathbf{A} - \tilde{\mathbf{A}}\|_F}{\|\mathbf{A}\|_F}. \tag{40}$$

To estimate $\mathbf{P}$, one may proceed by deflation:

i) select the submatrix $\mathbf{A}_r$ with the minimal number of columns c_r from the set $\Omega_{\mathbf{A}} = \{\mathbf{A}_r, r = 1, \ldots, R\}$;
ii) for all possible submatrices $\hat{\mathbf{A}}_r$ of $\hat{\mathbf{A}}$ consisting of c_r consecutive columns, compute $\angle(\mathbf{A}_r, \hat{\mathbf{A}}_r)$, the angle between the subspaces $\text{Span}(\mathbf{A}_r)$ and $\text{Span}(\hat{\mathbf{A}}_r)$ [28];
iii) the matrix $\hat{\mathbf{A}}_r$ for which $\angle(\mathbf{A}_r, \hat{\mathbf{A}}_r)$ is minimal is paired to $\mathbf{A}_r$. Remove $\mathbf{A}_r$ from $\Omega_{\mathbf{A}}$ and $\hat{\mathbf{A}}_r$ from $\hat{\mathbf{A}}$ and go back to i) until each submatrix of $\mathbf{A}$ has been paired with a submatrix of $\hat{\mathbf{A}}$. The pairing indicates how to build $\mathbf{P}$.

Once $\mathbf{P}$ estimated, the R diagonal blocks of $\mathbf{D}$ are computed one by one in the least squares sense, $\mathbf{D}_r = \hat{\mathbf{A}}_r^{\dagger}\bar{\mathbf{A}}_r$, $r = 1, \ldots, R$, where $\bar{\mathbf{A}} \stackrel{\text{def}}{=} \mathbf{A}\mathbf{P}^{-1}$.

Note that, contrarily to I_{conv}, ϵ_{rel} can also be used in the underdetermined case $I < N$.

VII. Numerical Experiments

A. Noise-Free Exact JBD

In this first set of experiments (Figs. 2, 3, 4), we compare the performance of the following algorithms for noise-free exact JBD problems:

i) JBD_{OG}, the nonunitary JBD algorithm based on Gradient descent with Optimal step proposed in [12] to minimize ϕ_{off} defined in (4);
ii) JBD_{ORG}, the nonunitary JBD algorithm based on the Relative Gradient with Optimal step also proposed in [12] to minimize ϕ_{off};

[2] if the optimization strategy preserves essential uniqueness, such as the JBD_{NCG} algorithm, there is no ambiguity in the interpretation of I_{conv}

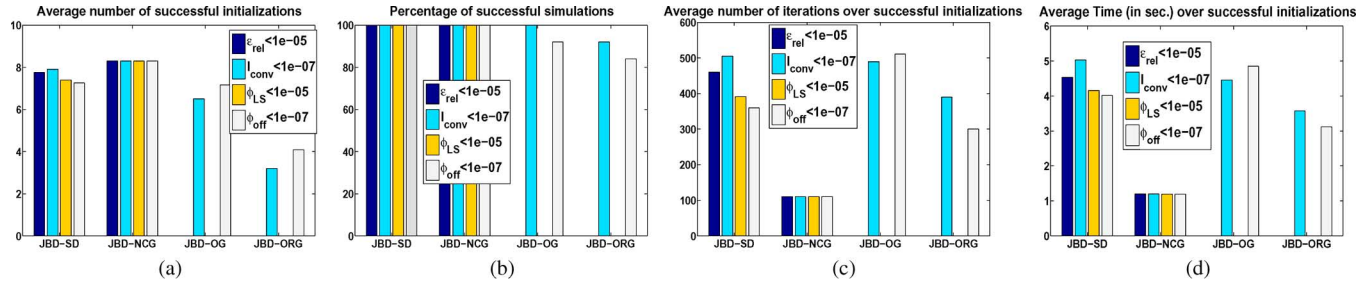

Fig. 3. Overdetermined case, $I = 15$, $N = 9$, $R = 3$, $L_1 = L_2 = L_3 = 3$, $K = 30$, case C2. (a) Successful initializations. (b) Successful runs. (c) Number of iterations. (d) Time in sec.

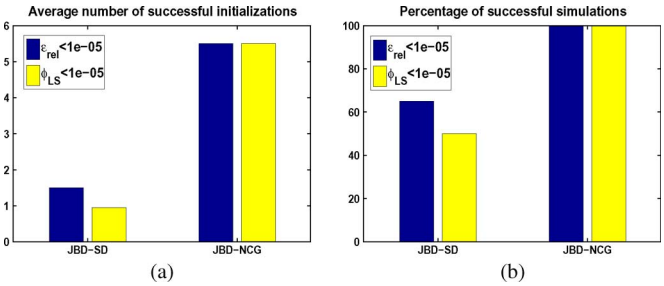

Fig. 4. Underdetermined case, $I = 6$, $N = 8$, $R = 4$, $L_1 = L_2 = L_3 = L_4 = 2$, $K = 30$, case C2. (a) Successful initializations. (b) Successful runs.

iii) JBD_{SD}, the Steepest Descent JBD algorithm given by Algorithm 1 to minimise ϕ_{LS}, where $\beta = 0$ is enforced to impose a steepest descent search direction at every step;
iv) JBD_{NCG}, the Nonlinear Conjugate Gradient JBD algorithm given by Algorithm 1 to minimize ϕ_{LS}.

For fixed dimensions and a chosen case (C1, C2, or C3), 100 exact JBD problems are generated. The matrices $\mathbf{A}$ and $\{\mathbf{D}_{kr}\}$, $k = 1, \ldots, K$, $r = 1, \ldots, R$, are randomly drawn for each problem, from a zero-mean unit-variance Normal distribution. For each of the 100 runs, 10 random starting points, generated with the same distribution as the true matrices, are used to initialize the four algorithms. For each starting point, the algorithms are stopped whenever one of the following criteria is satisfied

- (S1) $|\phi^{(p+1)} - \phi^{(p)}|/\phi^{(p)} < 10^{-8}$;
- (S2) $p = 2000$;
- (S3) $\phi < 10^{-8}$;

where ϕ stands for ϕ_{off} (algorithms JBD_{OG} and JBD_{ORG}) or ϕ_{LS} (algorithms JBD_{SD} and JBD_{NCG}).

In the exactly and overdertemined cases ($I \geq N$), we store the final values of ϕ_{LS}, ϕ_{off}, I_{conv} and ϵ_{rel} obtained after convergence for every run, every starting point and every algorithm (for the algorithms that minimize ϕ_{LS}, ϕ_{off} is computed afterwards from the final estimates and *vice-versa*). In the underdetermined case ($I < N$), only JBD_{SD} and JBD_{NCG} can be used, and their respective performance is assessed from the values of ϕ_{LS} and ϵ_{rel} obtained after convergence.

In Fig. 2, we focus on the exactly determined case $I = N = 9$. Fig. 2(a) represents the average number of successful initializations over the 100 runs, where an initialization is declared successful w.r.t. the following criteria: $\epsilon_{\text{rel}} < 10^{-5}$, $\text{I}_{\text{conv}} < 10^{-7}$, $\phi_{\text{LS}} < 10^{-5}$, $\phi_{\text{off}} < 10^{-7}$. Fig. 2(b) represents the percentage of successful runs, where a run is declared successful w. r. t. one criterion when at least one the ten starting points yields a final value that satisfies this criterion. Fig. 2(c) represents the number of iterations averaged over the successful initializations only whereas Fig. 2(d) represents the average running time per successful initialization. Comparison between JBD_{NCG} and JBD_{SD} confirms that using a conjugate gradient strategy rather than a steepest descent approach significantly improves the performance: JBD_{NCG} converges to the global minimum more frequently, see Fig. 2(a), (b) and faster, see Fig. 2(c), (d), than JBD_{SD}. On the same basis, it can be observed that JBD_{NCG} also outperforms JBD_{ORG} and JBD_{OG}.

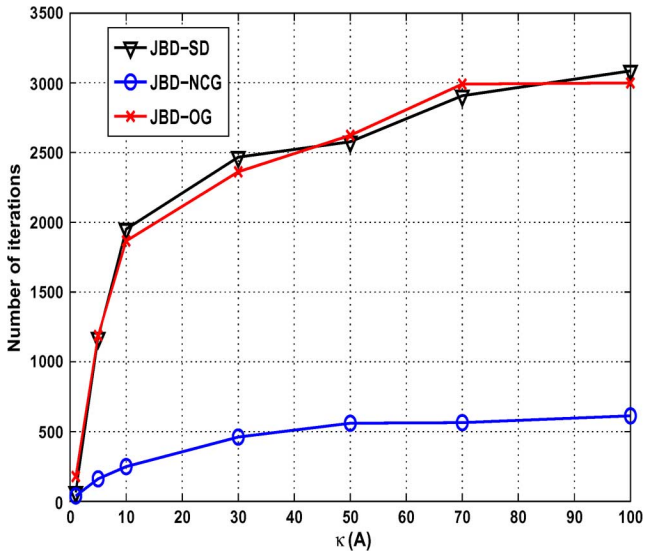

Fig. 5. Impact of the condition number $\kappa(\mathbf{A})$ on the number of iterations. Exactly determined case, $I = N = 9$, $R = 3$, $L_1 = L_2 = L_3 = 3$, $K = 30$, case C1.

In Fig. 3, we focus on the overdetermined case $I = 15$, $N = 9$. Fig. 3(a) shows that approximately eight initializations were successful in average for JBD_{SD} and JBD_{NCG}, for all criteria. Regarding the performance of JBD_{OG} and JBD_{ORG}, the criteria $\text{I}_{\text{conv}} < 10^{-7}$ and $\phi_{\text{off}} < 10^{-7}$ are satisfied by several initializations whereas the criteria $\epsilon_{\text{rel}} < 10^{-5}$ and $\phi_{\text{LS}} < 10^{-5}$ are never satisfied. This illustrates the analysis made in the Introduction to explain that minimization of ϕ_{off} by JBD_{OG} and

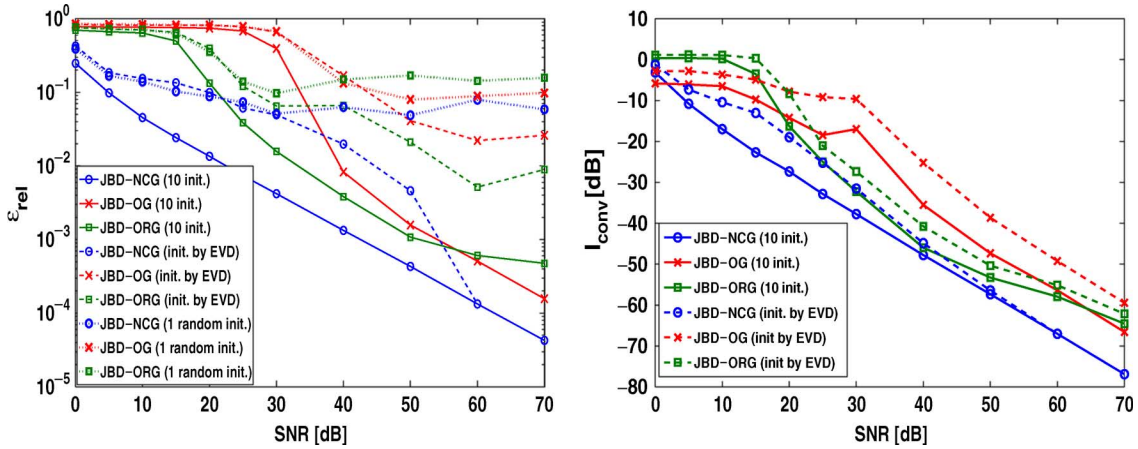

Fig. 6. Exactly determined case, $I = N = 9$, $R = 3$, $L_1 = L_2 = L_3 = 3$, $K = 30$, case C2. Left: $\epsilon_{\rm rel}$ versus SNR. Right: $I_{\rm conv}$ versus SNR.

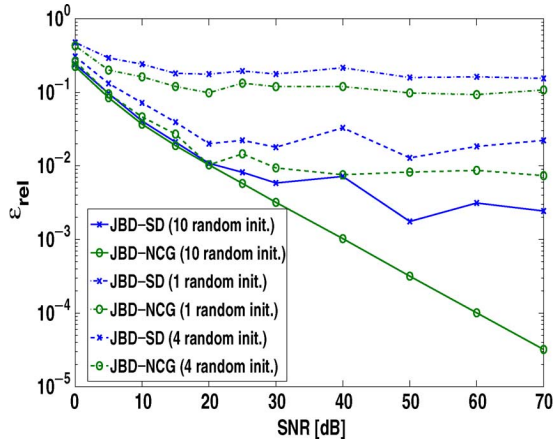

Fig. 7. Underdetermined case, $I = 6$, $N = 8$, $R = 4$, $K = 100$, $L_1 = L_2 = L_3 = L_4 = 2$, case C2.

JBD_{ORG} in the overdetermined case introduces ambiguities that breaks essential uniqueness.

In Fig. 4, we focus on the underdetermined case[3] $I = 6$, $N = 8$. An initialization is declared successful w. r. t. the criteria $\epsilon_{\rm rel} < 10^{-5}$, $\phi_{\rm LS} < 10^{-5}$ and it can be observed that JBD_{NCG} significantly outperforms JBD_{SD}. For instance, all runs were successful for JBD_{NCG}, versus 50 percent of the runs for JBD_{SD}.

In Fig. 5, we have fixed the matrices $\{\mathbf{D}_{kr}\}$, $k = 1, \ldots, K$, $r = 1, \ldots, R$, and for each value of the condition number of $\mathbf{A}$, $\kappa(\mathbf{A}) = \{1, 5, 10, 30, 50, 70, 100\}$, we test the JBD_{NCG}, JBD_{SD} and JBD_{OG} with 50 different random initializations. The condition number is imposed from an SVD of a randomly drawn matrix $\mathbf{A}$, $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^H$, after which $\mathbf{U}$ and $\mathbf{V}$ are kept fixed while $\mathbf{\Sigma}$ is changed so as to enforce the desired value of $\kappa(\mathbf{A})$. For each algorithm, the number of iterations is averaged over the successful initializations. The conjugate gradient-based algorithm JBD_{NCG} is far less sensitive to the value of $\kappa(\mathbf{A})$ than JBD_{SD} and JBD_{OG}.

[3]Although uniqueness of JBD is not covered by Theorem 1 in this case, preliminary experiments have been conducted with JBD_{NCG} to check that each time the global minimum is reached (with the threshold $\phi_{\rm LS} < 10^{-10}$), $\hat{\mathbf{A}}$ is equal to $\mathbf{A}$, only up to the model ambiguities (with the threshold $\epsilon_{\rm rel} < 10^{-8}$).

B. Monte Carlo Simulations

In this second set of experiments (Figs. 6 and 7), we compare the performance of the nonunitary JBD algorithms via Monte Carlo simulations, with the same stop criteria as before. From (7), the signal-to-noise ratio (SNR) is defined by $\text{SNR} = 10\log_{10}(\|\sum_{r=1}^{R} \mathcal{D}_r \bullet_1 \mathbf{A}_r \bullet_2 \mathbf{A}_r^*\|_F^2 / \|\mathcal{N}\|_F^2)$ dB, where the noise tensor $\mathcal{N}$ has symmetric slices and is generated from a zero-mean Normal distribution and variance $\sigma_{\mathcal{N}}$. For each value of the SNR, we conduct 200 Monte Carlo runs, where $\mathbf{A}$ and $\{\mathbf{D}_{kr}\}$, $k = 1, \ldots, K$, $r = 1, \ldots, R$ are randomly regenerated for each run, from a zero-mean unit-variance Normal distribution. For each value of the SNR, $\epsilon_{\rm rel}$ is averaged over the 200 runs. In Fig. 6, the exactly determined case $I = N = 9$ is considered and the performance of JBD_{NCG} is compared to that of JBD_{OG} and JBD_{ORG} for the three following settings: i) a single random initialization is used; ii) a single EVD-based initialization is used; iii) 10 initializations (one by EVD and nine random) are used and the best is selected as the one that yields the smallest final value of ϕ ($\phi_{\rm off}$ or $\phi_{\rm LS}$, depending on the algorithm). This experiment shows that the JBD_{NCG} algorithm proposed in this paper is significantly more accurate than the JBD_{OG} and JBD_{ORG} algorithms of [12], for both criteria $\epsilon_{\rm rel}$ and $I_{\rm conv}$. It can be observed that all algorithms substantially benefit from an EVD-based initialization, especially in the high SNR regime, as compared to a single random initialization. Moreover, coupling the EVD-based initialization with several additional random initializations significantly improves the performance.

In Fig. 7, the performance of JBD_{NCG} is compared to that of JBD_{SD} for the underdetermined case $I = 6$, $N = 9$. It can be observed that JBD_{NCG} outperforms JBD_{SD} and that the performance of both algorithms significantly improves when the number of random initializations increases, up to a certain point (the EVD-based initialization cannot be used in the underdetermined case).

C. Application: ISA

In this third set of experiments (Figs. 8 and 9), we use the JBD_{NCG} algorithm in the context of ISA, for SOS-based blind subspace separation. The mixing system is described in Fig. 8. We consider R primary discrete source signals $\{s_r[p]\}_{r=1}^{R}$. For the rth primary source $s_r[p]$, we generate a set of L_r discrete sequences $\{h_{rl}[p]\}_{l=1}^{L_r}$ and compute the set of L_r secondary

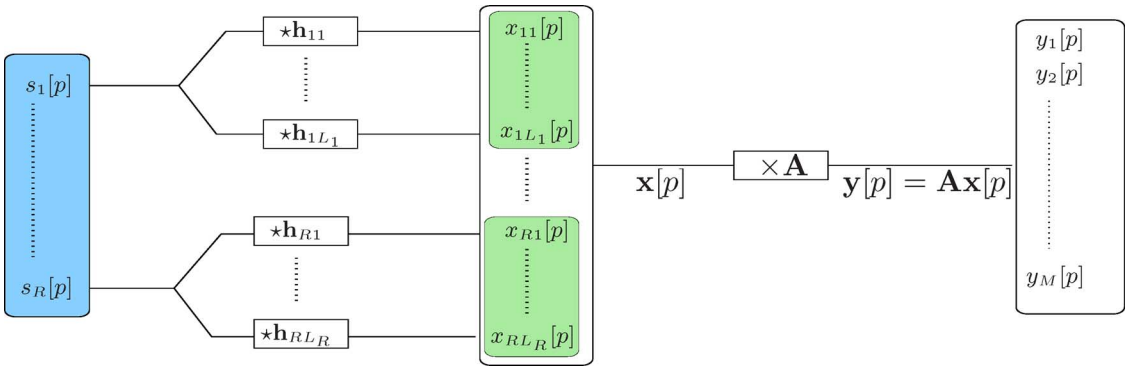

Fig. 8. Mixing system for the JBD-based ISA application. The primary sources $s_1, \ldots, s_R$ are assumed mutually uncorrelated. Each primary source is convolved with a subset of unknown FIR filters. The N secondary sources consist of R subsets; the secondary sources are mutually correlated within the same subset but mutually uncorrelated with sources from other subsets. The secondary sources are mixed through a linear instantaneous channel modeled by the matrix $\mathbf{A}$. Given the observations $y_1, \ldots, y_M$, the purpose is to estimate the primary sources.

sources $\{x_{rl}[p]\}_{l=1}^{L_r}$ associated to this primary source by $x_{rl}[p] = (h_{rl} \star s_r)[p]$, where $\star$ denotes the linear convolution operator. Let us stack the $N = \sum_{r=1}^{R} L_r$ secondary sources in the $N \times 1$ vector $\mathbf{x}[p]$. Then, we assume that the N secondary sources are linearly mixed through an instantaneous mixing channel, i.e., we get the following classical mixing model $\mathbf{y}[p] = \mathbf{A}\mathbf{x}[p]$, where $\mathbf{A}$ is the $M \times N$ unknown mixing matrix, M denotes the number of sensors and $\mathbf{y}[p]$ denotes the observed signal. Given $\mathbf{y}[p]$, the purpose is to estimate the primary sources.

To solve this BSS problem, we resort to SOS. Let us assume that the primary sources are zero-mean individually correlated in time but mutually uncorrelated. The set of secondary sources generated by the same primary source are assumed mutually correlated, whereas the secondary sources generated by different primary sources are assumed mutually uncorrelated. Under these assumptions, the spatial covariance matrices of the observations satisfy

$$\mathbf{R}_{\mathbf{yy}}(\tau_1) \stackrel{\text{def}}{=} E\left\{\mathbf{y}[p]\mathbf{y}^H[p+\tau_1]\right\} = \mathbf{A}\mathbf{D}_1\mathbf{A}^H$$
$$\vdots$$
$$\mathbf{R}_{\mathbf{yy}}(\tau_K) \stackrel{\text{def}}{=} E\left\{\mathbf{y}[p]\mathbf{y}^H[p+\tau_K]\right\} = \mathbf{A}\mathbf{D}_K\mathbf{A}^H$$

in which $\mathbf{D}_k \stackrel{\text{def}}{=} E\{\mathbf{x}[p]\mathbf{x}^H[p+\tau_k]\}$ is block-diagonal, $k = 1, \ldots, K$. Thus, an estimate $\hat{\mathbf{A}}$ of the mixing matrix $\mathbf{A}$ may be obtained by JBD of the set $\{\mathbf{R}_{\mathbf{yy}}(\tau_k)\}_{k=1}^{K}$.

For a practical illustration of this problem, we proceed as follows. Fig. 9(a), (b), (c) represents $R = 3$ speech signals used as primary sources, consisting of 64 000 samples each, sampled at 16 KHz. For each primary source, we set $L_r = 4$ and we randomly generate 12 sequences $\{h_{rl}[p]\}_{r=1,l=1}^{3,4}$, each of length 50, from a zero-mean unit-variance normal distribution. The 12 obtained secondary sources are then linearly mixed through a 14×12 mixing matrix $\mathbf{A}$, randomly generated from a zero-mean unit-variance normal distribution, i.e., we work in the overdetermined C1 case. Fig. 9(d), (e), (f) represent the first three mixtures. A set of $K = 21$ covariance matrices is computed, with a time lag τ_k taking linearly spaced values between 0 anf 200. Fig. 9(g) confirms that $\sum_{k=1}^{21} |\mathbf{R}_{\mathbf{yy}}(\tau_k)|$ is block-diagonal, i.e., the secondary sources generated by the same primary source are mutually correlated but not correlated to the other secondary sources. The separation task is achieved by JBD_{NCG}

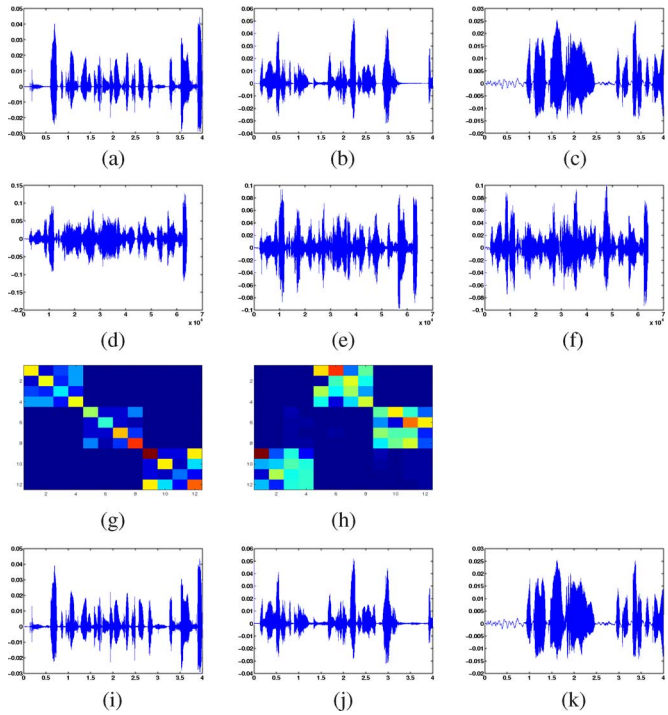

Fig. 9. Results of the JBD-based independent subspace analysis application. (a) s_1. (b) s_2. (c) s_3. (d) y_1. (e) y_2. (f) y_3. (g) $\sum_{k=1}^{K} |\mathbf{R}_{\mathbf{yy}}(\tau_k)|$. (h) $|\hat{\mathbf{A}}^\dagger\mathbf{A}|$. (i) $\hat{s}_1$. (j) $\hat{s}_2$. (k) $\hat{s}_3$.

initialized by the EVD-based technique of Section V. Fig. 9(h) shows that estimation of the mixing matrix is successful since $|\hat{\mathbf{A}}^\dagger\mathbf{A}|$ is a block-wise permuted block-diagonal matrix. Given $\hat{\mathbf{A}}$, the least squares estimate of the secondary sources is $\hat{\mathbf{x}}[p] = \hat{\mathbf{A}}^\dagger\mathbf{y}[p]$. Due to the inherent ambiguities of JBD, in case of perfect estimation, we get $\hat{\mathbf{x}}_{\tilde{r}}[p] = \mathbf{Z}_r\mathbf{x}_r[p]$, where $\hat{\mathbf{x}}_{\tilde{r}}[p] = [\hat{x}_{\tilde{r}1}[p], \ldots, \hat{x}_{\tilde{r}L_r}[p]]^T$ and $\mathbf{x}_r[p] = [x_{r1}[p], \ldots, x_{rL_r}[p]]^T$ denote the $\tilde{r}$th and rth group of L_r estimated and true secondary sources, respectively, that match together (block-wise permutation ambiguity), and $\mathbf{Z}_r$ denotes an unknown $L_r \times L_r$ arbitrary nonsingular matrix. Thus, we get

$$\hat{x}_{\tilde{r}1}[p] = (\bar{h}_{r1} \star s_r)[p]$$
$$\vdots$$
$$\hat{x}_{\tilde{r}L_r}[p] = \left(\bar{h}_{rL_r} \star s_r\right)[p] \qquad (41)$$

where $\bar{h}_{rl}[p] = [\mathbf{Z}_r]_{l,:}\mathbf{h}_r[p]$, $l = 1,\ldots,L_r$, and $\mathbf{h}_r[p] = [h_{r1}[p],\ldots,h_{rL_r}[p]]^T$. In other words, the estimates of the secondary sources from the same group are all filtered versions of the same unknown primary source. Thus, a blind SIMO (Single Input Multiple Output) system identification step is required to recover the primary sources instead of filtered versions of them. This can for instance be achieved via the subspace-based technique proposed in [29]. Comparison between the recovered primary sources—see Fig. 9(i), (j), (k)—with the true ones shows that the separation and reconstruction stages are successful.

VIII. Conclusion

In this paper, we have shown that nonunitary JBD can be regarded as a tensor decomposition. The latter can be computed by the minimization of a least squares subspace fitting criterion, for which we have proposed an efficient NCG algorithm. This tensor framework covers the exactly, over-, and underdetermined nonunitary JBD problems. In the exactly and overdetermined cases, we have shown that JBD algorithms can efficiently be initialized via an EVD-based closed form solution. Numerical experiments confirm that $\mathrm{JBD}_{\mathrm{NCG}}$ significantly outperforms existing nonunitary JBD algorithms and can successfully be applied to SOS-based ISA.

References

[1] P. Tichavsky and A. Yeredor, "Fast approximate joint diagonalization incorporating weight matrices," *IEEE Trans. Signal Process.*, vol. 57, no. 3, pp. 878–891, 2009.

[2] A. Yeredor, "Non-orthogonal joint diagonalization in the least-squares sense with application in blind source separation," *IEEE Trans. Signal Process.*, vol. 50, no. 7, pp. 1545–1553, 2002.

[3] L. De Lathauwer and J. Castaing, "Blind identification of underdetermined mixtures by simultaneous matrix diagonalization," *IEEE Trans. Signal Process.*, vol. 56, no. 3, pp. 1096–1105, 2008.

[4] R. A. Harshman, "Foundations of the PARAFAC procedure: Model and conditions for an 'explanatory' multi-mode factor analysis," *UCLA Working Papers in Phonet.*, vol. 16, pp. 1–84, 1970.

[5] R. Bro, "PARAFAC: Tutorial and applications," *Chemom. Intell. Lab. Syst.*, vol. 38, pp. 149–171, 1997.

[6] J. D. Carroll and J. Chang, "Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition," *Psychometrika*, vol. 35, no. 3, pp. 283–319, 1970.

[7] A. Stegeman, J. ten Berge, and L. De Lathauwer, "Sufficient conditions for uniqueness in Candecomp/Parafac and Indscal with random component matrices," *Psychometrika*, vol. 71, pp. 219–229, 2006.

[8] D. Nion, K. N. Mokios, N. D. Sidiropoulos, and A. Potamianos, "Batch and adaptive PARAFAC-based blind separation of convolutive speech mixtures," *IEEE Trans. Audio, Speech Lang. Process.*, vol. 18, no. 6, pp. 1193–1207, Aug. 2010.

[9] H. Bousbiah Salah, A. Belouchrani, and K. Abed Meraim, "Jacobi-like algorithm for blind signal separation of convolutive mixtures," *Electron. Lett.*, vol. 37, no. 16, pp. 1049–1050, Aug. 2001.

[10] C. Févotte and F. Theis, "Pivot selection strategies in Jacobi joint block-diagonalization," in *Proc. 7th Int. Conf. Independ. Compon. Anal. Signal Separat. (ICA'07)*, 2007, pp. 177–187.

[11] C. Févotte and C. Doncarli, "A unified presentation of blind source separation methods for convolutive mixtures using block-diagonalization," in *Proc. 4th Symp. Independ. Compon. Anal. Blind Source Separat. (ICA'03)*, 2003.

[12] H. Ghennioui, N. Thirion-Moreau, E. Moreau, and D. Aboutajdine, "Gradient-based joint block diagonalization algorithms: Application to blind separation of FIR convolutive mixtures," *Signal Process.*, vol. 90, no. 6, pp. 1836–1849, 2010.

[13] F. Theis, "Towards a general independent subspace analysis," in *Proc. NIPS 2006*, 2007.

[14] F. Theis, "Blind signal separation into groups of dependent signals using joint block diagonalization," in *Proc. ISCAS 2005*, Kobe, Japan, 2005, pp. 5878–5881.

[15] D. Nion and N. D. Sidiropoulos, "Tensor algebra and multi-dimensional harmonic retrieval in signal processing for MIMO radar," *IEEE Trans. Signal Process.*, vol. 58, no. 11, pp. 5693–5705, Nov. 2010.

[16] K. Abed Meraim and A. Belouchrani, "Algorithms for joint block diagonalization," in *Proc. (Eusipco'04)*, 2004, pp. 209–212.

[17] C. Févotte and F. Theis, Orthonormal Approximate Joint Block-Diagonalization GET/Télécom Paris, Tech. Rep. 2007D007, 2007.

[18] H. Ghennioui, E.-M. Fadailli, N. Thirion-Moreau, A. Adib, and E. Moreau, "A nonunitary joint block diagonalization algorithm for blind separation of convolutive mixtures of sources," *IEEE Signal Process. Lett.*, vol. 14, no. 11, pp. 860–863, Nov. 2007.

[19] X.-L. Li and X.-D. Zhang, "Nonorthogonal joint diagonalization free of degenerate solution," *IEEE Trans. Signal Process.*, vol. 55, no. 5, pp. 1803–1814, May 2007.

[20] L. De Lathauwer, "Decompositions of a higher-order tensor in block terms—Part I: Lemmas for partitioned matrices," *SIAM J. Matrix Anal. Appl.*, vol. 30, no. 3, pp. 1022–1032, 2008.

[21] L. De Lathauwer, "Decompositions of a higher-order tensor in block terms—Part II: Definitions and uniqueness," *SIAM J. Matrix Anal. Appl.*, vol. 30, no. 3, pp. 1033–1066, 2008.

[22] K. B. Petersen and M. S. Pedersen, The Matrix Cookbook Oct. 2008.

[23] H. Li and T. Adali, "Complex-valued adaptive signal processing using nonlinear functions," *EURASIP J. Adv. Signal Process.*, vol. 2008, pp. 1–9, 2008.

[24] A. Hjørungnes and D. Gesbert, "Complex-valued matrix differentiation: Techniques and key results," *IEEE Trans. Signal Process.*, vol. 55, no. 6, pp. 2740–2746, June 2007.

[25] W. W. Hager and H. Zhang, "A survey of nonlinear conjugate gradient methods," *Pacific J. Optimiz.*, vol. 2, pp. 35–58, 2006.

[26] J. Nocedal and S. J. Wright, *Numerical Optimization*. New York: Springer, 2000.

[27] P. Paatero, "The multilinear engine—A table-driven, least-squares program for solving multi-linear problems, including the n-way parallel factor analysis model," *J. Comput. Graph. Stat.*, vol. 8, no. 4, pp. 854–888, Dec. 1999.

[28] A. Bjorck and G. Golub, "Numerical methods for computing angles between linear subspaces," *Math. Comp.*, vol. 27, pp. 579–594, 1973.

[29] E. Moulines, P. Duhamel, J.-F. Cardoso, and S. Mayrargue, "Subspace methods for the blind identification of multichannel FIR filters," *IEEE Trans. Signal Process.*, vol. 43, pp. 516–525, 1995.

Dimitri Nion was born in Lille, France, on September 6, 1980. He received the Electronic Engineering degree from ISEN, Lille, France, in 2003, the M.S. degree from Queen Mary University, London, U.K., in 2003, and the Ph.D. degree in signal processing from the University of Cergy-Pontoise, France, in 2007.

He has been a Postdoctoral Fellow with the Technical University of Crete (2007–2008) and K.U. Leuven (2008–2010). His research interests include linear and multilinear algebra, blind source separation, signal processing for communications, and adaptive signal processing.