Joint EigenValue Decomposition Algorithms Based on First-Order Taylor Expansion

Remi Andre, Xavier Luciani, Eric Moreau

1

Joint EigenValue Decomposition Algorithms Based on First-Order Taylor Expansion

Rémi André, Xavier Luciani, and Eric Moreau, *Senior Member, IEEE*

***Abstract*—In this paper, we propose a new approach to compute the Joint EigenValue Decomposition (JEVD) of real or complex matrix sets. JEVD aims to find a common basis of eigenvectors to a set of matrices. JEVD problem is encountered in many signal processing applications. In particular, recent and efficient algorithms for the Canonical Polyadic Decomposition (CPD) of multiway arrays resort to a JEVD step. The suggested method is based on multiplicative updates. It is distinguishable by the use of a first-order Taylor Expansion to compute the inverse of the updating matrix. We call this approach Joint eigenvalue Decomposition based on Taylor Expansion (JDTE). This approach is derived in two versions based on simultaneous and sequential optimization schemes respectively. Here, simultaneous optimization means that all entries of the updating matrix are simultaneously optimized at each iteration. To the best of our knowledge, such an optimization scheme had never been proposed to solve the JEVD problem in a multiplicative update procedure. Our numerical simulations show that, in many situations involving complex matrices, the proposed approach improves the eigenvectors estimation while keeping a limited computational cost. Finally, these features are highlighted in a practical context of source separation through the CPD of telecommunication signals.**

***Keywords*—Joint eigenvalue decomposition, joint diagonalization, canonical polyadic decomposition, MIMO system.**

I. Introduction

Joint diagonalization problems have been at the heart of many source separation algorithms since the early nineties [1], [2], [3]. Joint diagonalization usually refers to the Joint Diagonalization by Congruence (JDC) [4],[5],[6]. However in many applications, the original separation problem can be rephrased as a joint diagonalization by similarity problem also called Joint EigenValue Decomposition (JEVD). For instance, JEVD has been used for direction of arrival estimation [7], joint angle-delay estimation [7], [8], multi-dimensional harmonic retrieval [9], Canonical Polyadic Decomposition (CPD) of tensors [10], [11], [12], [13], blind source separation [14], [15] and econometric [16].

JEVD consists in jointly diagonalizing a set of K non-defective real or complex matrices of size $N \times N$ sharing the same basis of eigenvectors. The matrices to diagonalize are then defined as

$$\mathbf{M}^{(k)} = \mathbf{A}\mathbf{D}^{(k)}\mathbf{A}^{-1}, \quad \forall k = 1, \ldots, K, \tag{1}$$

where $\mathbf{A} \in \mathbb{C}^{N\times N}$ is the matrix of eigenvectors and each diagonal matrix $\mathbf{D}^{(k)} \in \mathbb{C}^{N\times N}$ contains the eigenvalues of $\mathbf{M}^{(k)}$. In JDC, the inverse of $\mathbf{A}$ in equation (1) is replaced with the (Hermitian) transpose matrix. Hence, JDC and JEVD are equivalent if $\mathbf{A}$ is an unitary matrix. This is not necessary the case here. Another noteworthy difference between JDC and JEVD is the uniqueness condition of the solution. Let us define from the set of matrices $\mathbf{D}^{(k)}$ the $N \times K$ matrix

$$\mathbf{\Omega} = \begin{pmatrix} D_{11}^{(1)} & \cdots & D_{11}^{(K)} \\ \vdots & \cdots & \vdots \\ D_{NN}^{(1)} & \cdots & D_{NN}^{(K)} \end{pmatrix}. \tag{2}$$

Then, the JDC is essentially unique (*i.e.* matrix $\mathbf{A}$ is estimated up to permutation and column scaling) if and only if the rows of $\mathbf{\Omega}$ are non zero and non collinear, while the JEVD is essentially unique if and only if the rows of $\mathbf{\Omega}$ are distinct [10]. Thus JEVD uniqueness condition is less restrictive. Furthermore, most JDC problems can be rewritten as a JEVD problem while the opposite is not true [17].

A naive approach to solve the JEVD problem would be to compute a simple eigenvalue decomposition of any matrix $\mathbf{M}^{(k)}$ or a product of some of them. However, the joint decomposition allows to get better results in presence of noise and allows to solve the problem even if one or several eigenvalues are degenerated or very close. In the last decade, several algorithms were proposed to solve the JEVD problem. Most of them resort to the same general algorithmic scheme, usually referred to as Jacobi-like algorithm. This means that:

- The matrix set is jointly diagonalized by successive multiplicative updates that minimize a diagonalization criterion. At the end of the process, the diagonalizing matrix is then equal to the inverse of the matrix of eigenvectors up to scaling and permutation.
- The minimization is performed according to a block coordinate approach. Instead of computing all the parameters at each update, algorithms sequentially focus on a small subset of parameters. In the following, we will speak of sequential optimization scheme.

As a consequence, the main differences between JEVD algorithms are the parametrization of the updating matrix and/or the cost function used to estimate these parameters. In the first JEVD applications, authors were only interested in the eigenvalues. In this case, the matrix set can be triangularized by orthogonal or unitary matrices. Then, the eigenvalues are directly estimated from the diagonals of the triangular matrices [9], [18]. Rigorously speaking, these are not JEVD algorithms since the matrix of eigenvectors is not computed. Later, several authors proposed to compute the updating matrix in a polar decomposition form *i.e.* as the product of a symmetric (Hermitian in the complex case) matrix and an orthogonal

Rémi André, Xavier Luciani and Eric Moreau are with Université de Toulon, Aix Marseille Université, CNRS, LIS, Marseille, France.

(unitary) matrix. SH-RT [19], JUST [20], JDTM [13] and ESJD [21] algorithms have been developed according to this idea. These algorithms mainly differ in the cost functions used to compute the symmetric matrix at each update. SH-RT minimizes the off-diagonal entries of only one particular matrix of the matrix set. In JUST, all the off-diagonal entries of the matrix set are considered while JDTM targets two particular entries of each matrix. Finally, ESJD evaluates the most appropriate cost function between those of JUST and JDTM before each update. An alternative approach called JET has been proposed in [15]. It resorts to the LU factorization of the diagonalizing matrix and allows to reduce the numerical complexity. In this approach, the iterative procedure consists in estimating a lower triangular matrix that makes the matrix set as triangular as possible. This matrix is updated by minimizing a triangularization criterion. In JET-O, the exact criterion is used while JET-U resorts to an approximation. Finally, entries of the upper triangular matrix are directly estimated from analytical expressions. A perturbation analysis of the JEVD problem has been recently introduced and can be used to compare JEVD algorithms [22].

The previous algorithms require significant modifications to work with complex-valued data. It has been showed that these modifications degrade the performances [15]. Indeed, in the complex case, polar decomposition based algorithms are robust to the noise power but have a very high computational cost while algorithms of the JET family give poor results in the low Signal-to-Noise Ratio (SNR) range. That is why, we introduced in [23] an original approach called JAPAM. This family of algorithms can be directly applied to complex-valued matrices without any modification. Another characteristic of this approach is that different factorizations (LU, QR, Polar...) can be used to build the updating matrix. Moreover, both factorization matrices are jointly estimated. In the complex case, these algorithms significantly improve the estimation of the eigenvector matrix for any SNR while keeping a rather low computational cost. However, these algorithms are sensitive to the matrix size. A solution is then to initialize the algorithms by using the Generalized EigenValue Decomposition (GEVD) of two matrices. However, we are going to see that this solution may fail.

Thus, in this paper, we propose an alternative approach inspired by the NOODLES algorithm for JDC [24]. This approach has been briefly introduced in a previous conference paper [25]. In comparison with the other JEVD algorithms, we use a different parametrization of the updating matrix and a simplified cost function based on the first-order Taylor expansion of the matrix inverse. This approach has several advantages. First of all, as for the JAPAM approach, all the computations are performed in $\mathbb{C}$ so that no modification is required to deal with complex-valued matrix sets. Second, it can be applied to a simultaneous or a block coordinate (sequential) optimization scheme. In the simultaneous optimization scheme, all the parameters are jointly estimated. This is totally original in the context of JEVD based on multiplicative updates and this allows to decrease the numerical complexity.

The paper is organized as follows. In section II, we describe simultaneous and sequential optimization schemes to compute the JEVD by means of multiplicative updates. In section III-A, we deeply describe the first version of our approach based on a simultaneous optimization scheme and we introduce a first algorithm. Then in section III-B, we propose a second algorithm as a refinement of the first one. In section III-C, we show how to modify these two algorithms to derive two other algorithms based on a sequential optimization scheme. A short discussion about the algorithms convergence is addressed in section III-D. Section IV is dedicated to numerical simulations. First, we provide an empirical convergence study of the proposed algorithms. Then, these are compared with the reference algorithms from the literature according to different scenarios. Finally in section V, we show how our approach can help to compute the canonical polyadic decomposition in the context of the source separation of telecommunication signals.

Notations: Scalars are denoted by a lower case (a), vectors by a boldface lower case ($\mathbf{a}$) and matrices by a boldface upper case ($\mathbf{A}$). a_i is the i-th element of the vector $\mathbf{a}$ and A_{ij} is the (i,j)-th element of the matrix $\mathbf{A}$. $\mathbf{I}$ is the identity matrix. $\mathsf{Diag}\{\cdot\}$ is an operator which sets to zero the off-diagonal entries of the argument matrix. $\mathsf{ZDiag}\{\cdot\}$ is an operator which sets to zero the diagonal entries of the argument matrix. $\|\cdot\|$ is the Frobenius norm of the argument matrix or tensor. $\langle \mathbf{A}, \mathbf{B} \rangle = \text{trace}\{\mathbf{A}^H\mathbf{B}\}$ is the Frobenius inner product between the matrices $\mathbf{A}$ and $\mathbf{B}$. Finally, $\overline{z}$ and $|z|$ denote the complex conjugate and modulus of complex number z respectively.

II. Main structure of JEVD algorithms

JEVD algorithms aim to find an estimation of the matrix $\mathbf{A}^{-1}$ in equation (1) (up to the scale and permutation indeterminacy), denoted $\mathbf{B}$, such that matrices $\mathbf{N}^{(k)}$ defined by

$$\mathbf{N}^{(k)} = \mathbf{B}\mathbf{M}^{(k)}\mathbf{B}^{-1}, \ \forall k = 1, ..., K, \tag{3}$$

be as diagonal as possible in presence of noise. $\mathbf{B}$ is called the diagonalizing matrix. As mentioned in the introduction, $\mathbf{B}$ is estimated thanks to multiplicative updates. This means that, at each iteration, a matrix $\mathbf{X}$ is computed in order to update the current estimation of $\mathbf{B}$ and consequently the current matrix set as

$$\left\{ \begin{array}{ll} \mathbf{B} & \leftarrow \mathbf{X}\mathbf{B} \\ \mathbf{N}^{(k)} & \leftarrow \mathbf{X}\mathbf{N}^{(k)}\mathbf{X}^{-1}, \ \forall k = 1, ..., K. \end{array} \right. \tag{4}$$

The updating matrix $\mathbf{X}$ is estimated in order to make the matrices $\mathbf{N}^{(k)}$ as diagonal as possible. To this end, a classical cost function is the quadratic measure of diagonality:

$$C(\mathbf{X}) = \sum_{k=1}^{K} \|\mathsf{ZDiag}\{\mathbf{X}\mathbf{N}^{(k)}\mathbf{X}^{-1}\}\|^2. \tag{5}$$

Due to the scaling indeterminacy of the JEVD, the updating matrix $\mathbf{X}$ has N degrees of freedom. Thus, we only have to compute $N(N-1)$ parameters to build $\mathbf{X}$. In a simultaneous optimization scheme, at each iteration, a suitable approximation of C is minimized with respect to the whole set of unknown parameters with the conjecture that the algorithm will converge to a minimum of C. The framework of this scheme

is summarized in Algorithm 1. To the best of our knowledge, there is no JEVD algorithm based on this strategy. Indeed,

Algorithm 1 Framework of JEVD algorithms based on a simultaneous optimization scheme

Let S be a stopping criterion and it_{max} the maximal number of iterations
Initialize $\mathbf{B}$ with $\mathbf{I}$ or any clever choice
$\mathbf{N}^{(k)} \leftarrow \mathbf{M}^{(k)}$, $\forall k = 1, ..., K$
$it = 1$
while S is $false$ and $it < it_{max}$ **do**
  Estimate $\mathbf{X}$ as the matrix which minimizes an approximation of C
  Update $\mathbf{B} \leftarrow \mathbf{XB}$
  Update $\mathbf{N}^{(k)} \leftarrow \mathbf{X}\mathbf{N}^{(k)}\mathbf{X}^{-1}$, $\forall k = 1, ..., K$
  $it = it + 1$;
  Update S
end while
$\mathbf{D}^{(k)} \leftarrow \mathbf{N}^{(k)}$, $\forall k = 1, ..., K$

most JEVD algorithms of the literature resort to sequential optimization schemes. This consists in the sequential optimization of small subsets of parameters by using the following factorization of $\mathbf{X}$:

$$\mathbf{X} = \prod_{i=1}^{N-1} \prod_{j=i+1}^{N} \mathbf{X}^{(i,j)}. \tag{6}$$

Recalling that $\mathbf{X}$ is defined by $N(N-1)$ non free parameters, the main interest of this factorization is that the $N(N-1)/2$ matrices $\mathbf{X}^{(i,j)}$ depend on only two parameters. Thus, at each iteration, the diagonalizing matrix $\mathbf{B}$ and the matrix set $\mathbf{N}^{(k)}$ are updated $N(N-1)/2$ times by sweeping the couple indexes (i, j) as

$$\begin{cases} \mathbf{B} & \leftarrow \mathbf{X}^{(i,j)}\mathbf{B} \\ \mathbf{N}^{(k)} & \leftarrow \mathbf{X}^{(i,j)}\mathbf{N}^{(k)}\mathbf{X}^{(i,j)-1}, \ \forall k = 1, ..., K. \end{cases} \tag{7}$$

We speak of sweeping procedure. Matrices $\mathbf{X}^{(i,j)}$ are estimated by minimizing $C(\mathbf{X}^{(i,j)})$ (or an approximation of it) with respect to the two unknown parameters.

Usually, matrices $\mathbf{X}^{(i,j)}$ are equal to the identity matrix at the exception of entries $(i, i), (i, j), (j, i)$ and (j, j). Therefore, in the transformation $\mathbf{X}^{(i,j)}\mathbf{N}^{(k)}\mathbf{X}^{(i,j)-1}$ only the (i, j) and (j, i) entries of $\mathbf{N}^{(k)}$ are transformed by both the right and left multiplications. Hence, an alternative and simpler cost function that targets these two particular entries is

$$\begin{aligned} C_s(\mathbf{X}^{(i,j)}) = &\sum_{k=1}^{K} |(\mathbf{X}^{(i,j)}\mathbf{N}^{(k)}\mathbf{X}^{(i,j)-1})_{ij}|^2 \\ &+ \sum_{k=1}^{K} |(\mathbf{X}^{(i,j)}\mathbf{N}^{(k)}\mathbf{X}^{(i,j)-1})_{ji}|^2. \end{aligned} \tag{8}$$

Minimizing the cost function in equation (8) is generally not equivalent to minimize the cost function in equation (5) but it provides very good results in practice when N is not too large with respect to K. Moreover, it allows to rewrite the updating equations involving 2×2 matrices only. Practical and theoretical justifications of this approximation can be found in [21] and [26] in slightly different contexts. Another link between C and C_s will be highlighted in section III-C. The framework of JEVD algorithms based on a sequential optimization scheme is summarized in Algorithm 2.

Algorithm 2 Framework of JEVD algorithms based on a sequential optimization scheme

Let S be a stopping criterion and it_{max} the maximal number of iterations
Initialize $\mathbf{B}$ with $\mathbf{I}$ or any clever choice
$\mathbf{N}^{(k)} \leftarrow \mathbf{M}^{(k)}$, $\forall k = 1, ..., K$
$it = 1$
while S is $false$ and $it < it_{max}$ **do**
  for $i = 1$ to $N - 1$ **do**
    for $j = i + 1$ to N **do**
      Estimate $\mathbf{X}^{(i,j)}$ as the matrix minimizing C, C_s or another approximation of C.
      Update $\mathbf{B} \leftarrow \mathbf{X}^{(i,j)}\mathbf{B}$
      Update $\mathbf{N}^{(k)} \leftarrow \mathbf{X}^{(i,j)}\mathbf{N}^{(k)}\mathbf{X}^{(i,j)-1}$, $\forall k = 1, ..., K$
    end for
  end for
  $it = it + 1$;
  Update S
end while
$\mathbf{D}^{(k)} \leftarrow \mathbf{N}^{(k)}$, $\forall k = 1, ..., K$

The approach proposed in this paper allows both optimization schemes: simultaneous and sequential. Moreover switching from one scheme to another involves few algorithmic modifications.

Numerical complexity of main JEVD algorithms. The numerical complexity can be defined as the number of real multiplications that an algorithm computes during one iteration. We consider here that a complex multiplication is equivalent to four real multiplications (even if it can be done in only three multiplications). Numerical complexity of JEVD algorithms is clearly dominated by the updating step of the matrix set (equation (7)). Thus, numerical complexities of complex JEVD algorithms are about $44KN^3$ for JUST, $32KN^3$ for SH-RT and JDTM, $16KN^3$ for JAPAM, $8KN^3$ for JET-O and $4KN^3$ for JET-U.

III. JEVD ALGORITHMS BASED ON TAYLOR EXPANSION

In the first subsection, we introduce an original JEVD algorithm based on a simultaneous optimization scheme. This algorithm is called JDTE for Joint eigenvalue Decomposition based on Taylor Expansion. In the second subsection, we propose an improved version of this algorithm. In the last subsection, we develop two algorithms based on a sequential optimization scheme.

Before going further, we consider the following assumptions.

Assumption 1: At a given iteration, matrix $\mathbf{X}$ is close to a stationary point of the optimization process.

Assumption 2: At a given iteration, matrix $\mathbf{B}$ is close to the diagonalizing solution.

A stationary point is characterized by $\mathbf{X} = \mathbf{I}$, therefore Assumption 1 implies

$$\|\mathsf{ZDiag}\{\mathbf{X}\}\| \ll 1. \tag{9}$$

On the other hand, Assumption 2 naturally implies

$$\|\mathsf{ZDiag}\{\mathbf{N}^{(k)}\}\| \ll 1, \ \forall k = 1, ..., K. \tag{10}$$

A. Joint eigenvalue Decomposition based on Taylor Expansion (JDTE)

JDTE algorithm is based on the framework described in Algorithm 1. Since $\mathbf{X}$ has N degrees of freedom due to the scaling indeterminacy, we can impose

$$\mathbf{X} = \mathbf{I} + \mathbf{Z}, \tag{11}$$

where $\mathbf{Z} = \mathsf{ZDiag}\{\mathbf{X}\}$. According to Assumption 1, $\mathbf{X}$ is a strictly diagonally dominant matrix. Such a matrix is nonsingular. Thereby the updating scheme in equation (4) becomes

$$\left\{ \begin{array}{ll} \mathbf{B} & \leftarrow (\mathbf{I} + \mathbf{Z})\mathbf{B} \\ \mathbf{N}^{(k)} & \leftarrow (\mathbf{I} + \mathbf{Z})\mathbf{N}^{(k)}(\mathbf{I} + \mathbf{Z})^{-1}, \ \forall k = 1, ..., K. \end{array} \right. \tag{12}$$

By injecting (12) in (5), the cost function C only depends on matrix $\mathbf{Z}$ and can be rewritten as

$$C_{I+Z}(\mathbf{Z}) = \sum_{k=1}^{K} \|\mathsf{ZDiag}\{(\mathbf{I} + \mathbf{Z})\mathbf{N}^{(k)}(\mathbf{I} + \mathbf{Z})^{-1}\}\|^2. \tag{13}$$

The term $(\mathbf{I}+\mathbf{Z})^{-1}$ makes the minimization of the above cost function rather intricate. However, under Assumption 1, the first-order Taylor expansion of $(\mathbf{I}+\mathbf{Z})^{-1}$ gives us the following useful approximation for $\mathbf{X}^{-1}$:

$$(\mathbf{I} + \mathbf{Z})^{-1} \simeq \mathbf{I} - \mathbf{Z}. \tag{14}$$

The cost function (13) can thus be approximated as

$$C_{I+Z}(\mathbf{Z}) \simeq \sum_{k=1}^{K} \|\mathsf{ZDiag}\{(\mathbf{I} + \mathbf{Z})\mathbf{N}^{(k)}(\mathbf{I} - \mathbf{Z})\}\|^2. \tag{15}$$

By developing and neglecting the terms with an order in $\mathbf{Z}$ superior to one, we have for all $k = 1, ..., K$,

$$(\mathbf{I} + \mathbf{Z})\mathbf{N}^{(k)}(\mathbf{I} - \mathbf{Z}) = \mathbf{N}^{(k)} + \mathbf{Z}\mathbf{N}^{(k)} - \mathbf{N}^{(k)}\mathbf{Z} - \mathbf{Z}\mathbf{N}^{(k)}\mathbf{Z} \tag{16}$$

$$\simeq \mathbf{N}^{(k)} + \mathbf{Z}\mathbf{N}^{(k)} - \mathbf{N}^{(k)}\mathbf{Z}. \tag{17}$$

Let us now decompose the matrices $\mathbf{N}^{(k)}$ as

$$\mathbf{N}^{(k)} = \mathbf{\Lambda}^{(k)} + \mathbf{O}^{(k)}, \ \forall k = 1, ..., K, \tag{18}$$

where $\mathbf{\Lambda}^{(k)} = \mathsf{Diag}\{\mathbf{N}^{(k)}\}$ and $\mathbf{O}^{(k)} = \mathsf{ZDiag}\{\mathbf{N}^{(k)}\}$. Under Assumption 2, we have $||\mathbf{O}^{(k)}|| \ll 1$ for all k. Then, by neglecting the terms in $\mathbf{O}^{(k)}$ and $\mathbf{Z}$ with an order superior to one, equation (17) becomes

$$(\mathbf{I} + \mathbf{Z})\mathbf{N}^{(k)}(\mathbf{I} - \mathbf{Z}) \simeq \mathbf{\Lambda}^{(k)} + \mathbf{O}^{(k)} + \mathbf{Z}\mathbf{\Lambda}^{(k)} - \mathbf{\Lambda}^{(k)}\mathbf{Z}. \tag{19}$$

Finally, injecting (19) in (15), we obtain the approximated cost function (under Assumption 1 and Assumption 2)

$$C_{jdte}(\mathbf{Z}) = \sum_{k=1}^{K} \|\mathbf{O}^{(k)} + \mathbf{Z}\mathbf{\Lambda}^{(k)} - \mathbf{\Lambda}^{(k)}\mathbf{Z}\|^2. \tag{20}$$

Now, by rewriting this equation entrywise, it immediately appears that $C_{jdte}(\mathbf{Z})$ is separable as a sum of convex functions:

$$C_{jdte}(\mathbf{Z}) = \sum_{\substack{m,n=1 \\ m \neq n}}^{N} f_{mn}(Z_{mn}),$$

where

$$f_{mn}(Z_{mn}) = \sum_{k=1}^{K} |O_{mn}^{(k)} + Z_{mn}(\Lambda_{nn}^{(k)} - \Lambda_{mm}^{(k)})|^2.$$

As a consequence, minimizing $C_{jdte}(\mathbf{Z})$ is equivalent to independently minimize each $f_{mn}(Z_{mn})$. Therefore for each couple (m, n), $m \neq n$, the optimal value of Z_{mn} is given by $\frac{\partial f(Z_{mn})}{\partial Z_{mn}} = 0$. This yields

$$\sum_{k=1}^{K} (\bar{\Lambda}_{nn} - \bar{\Lambda}_{mm})(O_{mn}^{(k)} + Z_{mn}(\Lambda_{nn}^{(k)} - \Lambda_{mm}^{(k)})) = 0 \tag{21}$$

and we finally obtain

$$Z_{mn} = \frac{\sum_{k=1}^{K} (\bar{\Lambda}_{mm}^{(k)} - \bar{\Lambda}_{nn}^{(k)}) O_{mn}^{(k)}}{\sum_{k=1}^{K} |\Lambda_{nn}^{(k)} - \Lambda_{mm}^{(k)}|^2}, \ \forall (m, n), \ m \neq n. \tag{22}$$

Remark 1: If the algorithm is initialized with the identity matrix, Assumption 1 and Assumption 2 may not be true during the first iterations especially if N is large. A simple way to fulfill both assumptions is thus to initialize the algorithms by the eigenvectors obtained from the GEVD of two matrices $\mathbf{M}^{(k)}$. We are going to propose an alternative solution in the next subsection.

Remark 2: The denominator in equation (22) is null if and only if $\forall k, \Lambda_{nn}^{(k)} = \Lambda_{mm}^{(k)}$. If this particular case occurs for a given couple (m, n), with $m \neq n$, we can impose $Z_{mn} = 0$ in the algorithm at the current iteration. The following proposition provides a sufficient condition to ensure that there exists at least one couple (m, n), with $m \neq n$, for which the denominator is non zero (otherwise $\mathbf{X}$ will be the identity matrix and the algorithm will stop).

Proposition 1: Let $\mathbf{B}$ be the estimation of $\mathbf{A}^{-1}$ at the current iteration and $\mathbf{G} = \mathbf{B}\mathbf{A}$. If $\mathbf{G}$ is a strictly diagonally dominant matrix (up to permutation) and if the matrix $\mathbf{\Omega}$, defined in equation (2), has at least two non co-linear rows then there exists at least one couple (m, n) with $m \neq n$ such that the denominator in equation (22) is not null.

A proof is given in appendix A. The first part of the sufficient condition is ensured by Assumption 2. The second part is a very weak condition.

Remark 3: In the algorithmic procedure, we recommend to use $(\mathbf{I} + \mathbf{Z})^{-1}$ instead of $(\mathbf{I} - \mathbf{Z})$ to update the matrices $\mathbf{N}^{(k)}$. In doing so, we preserve the similarity between the matrices $\mathbf{N}^{(k)}$

and $\mathbf{D}^{(k)}$.

Numerical complexity. The costliest step of one iteration of JDTE is the update of the K matrices $\mathbf{N}^{(k)}$ by the matrix $\mathbf{I}+\mathbf{Z}$. Hence, in the complex case, the numerical complexity of JDTE is

$$\Gamma[JDTE] \simeq 8KN^3. \tag{23}$$

It is noteworthy that JDTE is one of the JEVD algorithms with the lowest numerical complexity.

B. Weighted JDTE (WJDTE)

We now propose a refinement of JDTE in order to mitigate the influence of Assumption 1 and Assumption 2. We call this new algorithm WJDTE for Weighted JDTE. Here, we decompose the updating matrix as

$$\mathbf{X} = \mathbf{I} + \mu_{\text{opt}}(\mathbf{Z})\mathbf{Z}, \tag{24}$$

where $\mu_{\text{opt}}(\mathbf{Z}) \in \mathbb{R}$. Matrix $\mathbf{Z}$ is computed as previously, entrywise from equation (22), we denote this matrix $\mathbf{Z}_{jdte}$. Then, we compute $\mu_{\text{opt}}(\mathbf{Z}_{jdte})$ by minimizing a cost function derived from equation (17) instead of equation (19):

$$\mu_{\text{opt}}(\mathbf{Z}_{jdte}) = \underset{\mu}{\text{argmin}} \sum_{k=1}^{K} ||\mathbf{O}^{(k)} + \mu\mathbf{C}^{(k)}||^2, \tag{25}$$

where $\mathbf{C}^{(k)} = \mathsf{ZDiag}\{\mathbf{Z}_{jdte}\mathbf{N}^{(k)} - \mathbf{N}^{(k)}\mathbf{Z}_{jdte}\}$. The cost function is thus a convex quadratic polynomial:

$$\sum_{k=1}^{K} ||\mathbf{O}^{(k)} + \mu\mathbf{C}^{(k)}||^2 = \sum_{k=1}^{K} ||\mathbf{O}^{(k)}||^2 + 2\mathsf{Re}\{\langle \mathbf{O}^{(k)}, \mathbf{C}^{(k)} \rangle\}\mu + ||\mathbf{C}^{(k)}||^2\mu^2. \tag{26}$$

So, we immediately obtain

$$\mu_{\text{opt}}(\mathbf{Z}_{jdte}) = -\frac{\sum_{k=1}^{K} \mathsf{Re}\{\langle \mathbf{O}^{(k)}, \mathbf{C}^{(k)} \rangle\}}{\sum_{k=1}^{K} ||\mathbf{C}^{(k)}||^2}. \tag{27}$$

Finally, we take $\mathbf{Z}_{wjdte} = \mu_{\text{opt}}(\mathbf{Z}_{jdte})\mathbf{Z}_{jdte}$.

Remark 1: It is worth mentioning that for $\mu = 0$, $||\mathbf{O}^{(k)} + \mu\mathbf{C}^{(k)}||^2 = ||\mathsf{ZDiag}\{\mathbf{N}^{(k)}\}||^2$. So an important result here is that matrix $\mathbf{Z}_{wjdte}$ cannot increase the approximated cost function

$$C_{wjdte}(\mathbf{Z}) = \sum_{k=1}^{K} \|\mathsf{ZDiag}\{\mathbf{N}^{(k)} + \mathbf{Z}\mathbf{N}^{(k)} - \mathbf{N}^{(k)}\mathbf{Z}\}\|^2. \tag{28}$$

Remark 2: We can easily show that $C_{wjdte}(\mathbf{Z})$ decreases for any value of μ taken in the interval $]0, 2\mu_{\text{opt}}[$ (if $\mu_{\text{opt}} > 0$) or $]2\mu_{\text{opt}}, 0[$ (otherwise). This means that in the algorithmic procedure, we can force $-1 \leq \mu_{\text{opt}} \leq 1$ in order to better respect Assumption 1. Moreover, at the convergence, if the denominator in equation (27) gets close to the machine precision, we force $\mu_{\text{opt}} = 1$.

Remark 3: Whatever the chosen matrix $\mathbf{Z}$, it is clear that matrix $\mu_{\text{opt}}(\mathbf{Z})\mathbf{Z}$ cannot increase C_{wjdte}. As a consequence, Assumption 2 is not required anymore and we can see $\mathbf{Z}_{jdte}$ as a smart choice for $\mathbf{Z}$.

Numerical complexity. The computation of the K matrices $\mathbf{C}^{(k)}$ involves about $8KN^3$ extra multiplications. So the numerical complexity of WJDTE is

$$\Gamma[WJDTE] \simeq 16KN^3. \tag{29}$$

C. Sweeping JDTE (SJDTE) and Weighted Sweeping JDTE (WSJDTE)

We remind that the sequential optimization scheme consists in updating the matrix set with $N(N-1)/2$ matrices $\mathbf{X}^{(i,j)}$ at each iteration, according to the framework described in Algorithm 2. We thus decompose matrices $\mathbf{X}^{(i,j)}$ as

$$\mathbf{X}^{(i,j)} = \mathbf{I} + \mathbf{Z}^{(i,j)}, \ \forall(i,j), \ i < j \tag{30}$$

where matrix $\mathbf{Z}^{(i,j)}$ has all its entries equal to zero except $Z_{ij}^{(i,j)}$ and $Z_{ji}^{(i,j)}$. The same reasoning as for JDTE immediately leads to

$$Z_{ij}^{(i,j)} = \frac{\sum_{k=1}^{K}(\bar{\Lambda}_{ii}^{(k)} - \bar{\Lambda}_{jj}^{(k)})O_{ij}^{(k)}}{\sum_{k=1}^{K}|\Lambda_{ii}^{(k)} - \Lambda_{jj}^{(k)}|^2}, \ \forall(i,j), \ i < j \tag{31}$$

and

$$Z_{ji}^{(i,j)} = \frac{\sum_{k=1}^{K}(\bar{\Lambda}_{jj}^{(k)} - \bar{\Lambda}_{ii}^{(k)})O_{ji}^{(k)}}{\sum_{k=1}^{K}|\Lambda_{ii}^{(k)} - \Lambda_{jj}^{(k)}|^2}, \ \forall(i,j), \ i < j. \tag{32}$$

We call this algorithm SJDTE for Sweeping JDTE.

It is noteworthy that in this case, the simplifications obtained from Assumption 1 and Assumption 2 lead to

$$C_{I+Z}(\mathbf{Z}^{(i,j)}) \simeq C_s(\mathbf{I} + \mathbf{Z}^{(i,j)}) + \alpha_{i,j}, \tag{33}$$

where C_s is the simplified cost function defined in equation (8) and where

$$\alpha_{i,j} = \sum_{k=1}^{K} \sum_{\substack{n=1 \\ n\neq i, n\neq j}}^{N} |O_{ni}^{(k)}|^2 + |O_{in}^{(k)}|^2 + |O_{jn}^{(k)}|^2 + |O_{nj}^{(k)}|^2. \tag{34}$$

Since $\alpha_{i,j}$ does not depend of $\mathbf{Z}^{(i,j)}$, C_s and C_{I+Z} are equivalent under Assumption 1 and Assumption 2.

Remarks done for JDTE hold for SJDTE. In particular, the denominators of $Z_{ij}^{(i,j)}$ and $Z_{ji}^{(i,j)}$ must not be equal to zero for all couples (i,j) with $i < j$. If it occurs for any couple (i,j), we set $Z_{ij}^{(i,j)} = Z_{ji}^{(i,j)} = 0$ so that $\mathbf{X}^{(i,j)}$ is the identity matrix. As a consequence, the sufficient condition given by the proposition 1 holds.

As it has been done for JDTE, we can derive a Weighted Sweeping JDTE algorithm (WSJDTE) by setting the updating matrix $\mathbf{X}^{(i,j)}$ as

$$\mathbf{X}^{(i,j)} = \mathbf{I} + \mu_{\text{opt}}(\mathbf{Z}^{(i,j)})\mathbf{Z}^{(i,j)}. \tag{35}$$

The weight $\mu_{\text{opt}}(\mathbf{Z}^{(i,j)})$ is then computed as $\mu_{\text{opt}}(\mathbf{Z})$ by taking $\mathbf{C}^{(k)} = \mathsf{ZDiag}\{\mathbf{Z}^{(i,j)}\mathbf{N}^{(k)} - \mathbf{Z}^{(i,j)}\mathbf{N}^{(k)}\}$ in equation (27).

Numerical complexity. The numerical complexity of SJDTE is dominated by the $N(N-1)/2$ updates by the

matrices $\mathbf{X}^{(i,j)}$. Each update has a numerical complexity equal to $24KN$ (indeed the inverse of $\mathbf{X}^{(i,j)}$ may have four entries different to 0 or 1). Therefore, we have

$$\Gamma[SJDTE] \simeq 12KN^3. \quad (36)$$

Now for WSJDTE, the computation of $\mu_{\text{opt}}(\mathbf{Z}^{(i,j)})$ involves $8KN^2 + 4KN$ extra multiplications. This yields

$$\Gamma[WSJDTE] \simeq 4KN^4 + 20KN^3. \quad (37)$$

In term of numerical complexity, SJDTE is thus a compromise between JDTE and WJDTE while WSJDTE has the highest numerical complexity.

D. Discussion about algorithms convergence

JDTE and SJDTE resort to different approximations of the cost function given in equation (13). We thus conjecture that if the approximations are good enough, then the original cost function C_{I+Z} and the approximated one will have the same local behavior, *i.e.* if an update decreases the approximated cost function, then it will also decrease the original one. Those approximations are justified by Assumption 1 and Assumption 2. We have proposed to resort to GEVD initialization in order to ensure both assumptions. We can show that this is not necessary for WJDTE (the same reasoning holds for WSJDTE).

We first recall that WJDTE only requires Assumption 1, that is $\|\mathbf{Z}\| \ll 1$. Under this assumption, C_{I+Z} can be rewritten thanks to Neumann series [27] as

$$C_{I+Z}(\mathbf{Z}) = \sum_{k=1}^{K} \|\mathsf{ZDiag}\{(\mathbf{I}+\mathbf{Z})\mathbf{N}^{(k)}(\mathbf{I}-\mathbf{Z}+\mathbf{S})\}\|^2, \quad (38)$$

where $\mathbf{S} = \sum_{n=2}^{\infty}(-1)^n\mathbf{Z}^n$. Developing equation (38) yields

$$C_{I+Z}(\mathbf{Z}) = \sum_{k=1}^{K} \|\mathbf{F}^{(k)}\|^2 + \sum_{k=1}^{K} \|\mathbf{G}^{(k)}\|^2 + 2\mathsf{Re}\{\langle \mathbf{F}^{(k)H}, \mathbf{G}^{(k)} \rangle\}, \quad (39)$$

where

$$\mathbf{F}^{(k)} = \mathsf{ZDiag}\{\mathbf{N}^{(k)} + \mathbf{Z}\mathbf{N}^{(k)} - \mathbf{N}^{(k)}\mathbf{Z}\}, \quad (40)$$

$$\mathbf{G}^{(k)} = \mathsf{ZDiag}\{-\mathbf{Z}\mathbf{N}^{(k)}\mathbf{Z} + \mathbf{N}^{(k)}\mathbf{S} + \mathbf{Z}\mathbf{N}^{(k)}\mathbf{S}\}. \quad (41)$$

Thus, we have

$$C_{I+Z}(\mathbf{Z}) = C_{wjdte}(\mathbf{Z}) + \sum_{k=1}^{K} \|\mathbf{G}^{(k)}\|^2 + 2\mathsf{Re}\{\langle \mathbf{F}^{(k)H}, \mathbf{G}^{(k)} \rangle\}. \quad (42)$$

Now, considering that $\mathbf{Z}_{wjdte} = \mu_{\text{opt}}\mathbf{Z}_{jdte}$, it is always possible to make $\sum_{k=1}^{K} \|\mathbf{G}^{(k)}\|^2 + 2\mathsf{Re}\{\langle \mathbf{F}^{(k)H}, \mathbf{G}^{(k)} \rangle\}$ negligible in equation (42) by choosing $|\mu_{\text{opt}}|$ small enough. As a consequence, if the original cost function C_{I+Z} increases for $\mathbf{Z} = \mathbf{Z}_{wjdte}$, we can still decrease $|\mu_{\text{opt}}|$ in order to ensure Assumption 1. This precaution also ensures that $\mathbf{X}^{-1}$ exists. However, in practice it is unnecessary and could slowdown the convergence. Therefore, we only impose $|\mu_{\text{opt}}| < 1$ and accept the small possible increases of the original cost function that can occur during the first iterations.

IV. Numerical simulations

In this section, we study the behavior and the performances of the proposed algorithms. The first subsection is a convergence study and the second one is a performance comparison with the existing JEVD algorithms. The matrix sets to diagonalize are built as follows. First, we randomly generate K diagonal matrices $\mathbf{D}^{(k)}$ and a matrix $\mathbf{A}$ of size $N \times N$ by using a complex standard normal distribution (unless otherwise indicated). Then, we compute matrices $\mathbf{M}^{(k)}$ as

$$\mathbf{M}^{(k)} = \frac{\mathbf{A}\mathbf{D}^{(k)}\mathbf{A}^{-1}}{||\mathbf{A}\mathbf{D}^{(k)}\mathbf{A}^{-1}||} + \sigma\frac{\mathbf{E}^{(k)}}{||\mathbf{E}^{(k)}||}, \ \forall k = 1, ..., K, \quad (43)$$

where $\mathbf{E}^{(k)}$ models a zero mean Gaussian complex-valued random noise and σ is a parameter allowing to set the SNR value: $\text{SNR} = -20\log(\sigma)$.

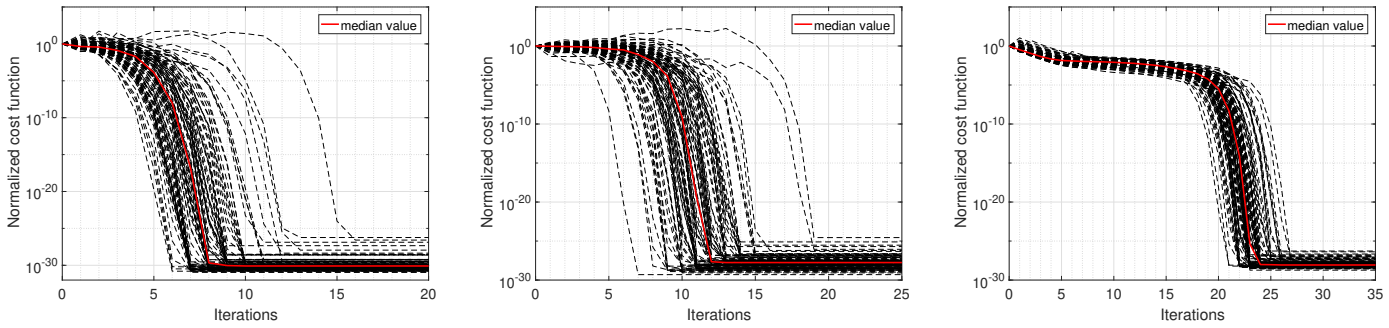
A. Convergence study

Algorithms are initialized with the identity matrix and are run on 100 different sets of 20 noise free matrices ($K = 20, \sigma = 0$). When the identity matrix is used for the initialization, Assumption 1 and Assumption 2 may not be satisfied during the first iterations of the algorithms, especially when N is large. As a consequence, the matrix size can have a significant impact on the convergence rate. However, we expect that this impact is limited for WJDTE and WSJDTE. We thus consider different matrix sizes for the different algorithms: $N = 5$ for JDTE, $N = 16$ for SJDTE and $N = 100$ for WJDTE and WSJDTE. We refer to sections IV-B2 and IV-B3 for a deeper study of the influence of the matrix size. Figure 1 shows the evolution of the cost function C, introduced in equation (5), with respect to the iteration number for JDTE, SJDTE and WJDTE and for the different matrix sets (convergence plots of WSJDTE are not shown since they are similar to those of WJDTE). The cost function is normalized here by $\sum_{k=1}^{K} ||\mathsf{ZDiag}\{\mathbf{M}^{(k)}\}||^2$. In each case, the method consistently converges after few iterations to a value close to the machine precision. These results indicate that the proposed approach is relevant in the general case and for a large range of matrix sizes.

We now consider a more specific case in which entries of matrix $\mathbf{A}$ are defined as $A_{ii} = 1, \ \forall i$ and $A_{ij} = 0.999$, $\forall (i,j), i \neq j$. Hence, the same ill-conditioned matrix $\mathbf{A}$ is used to build all the matrix sets. Convergence plots are shown in figure 2. Here again, the cost function consistently reaches a value close to the machine precision after few iterations.

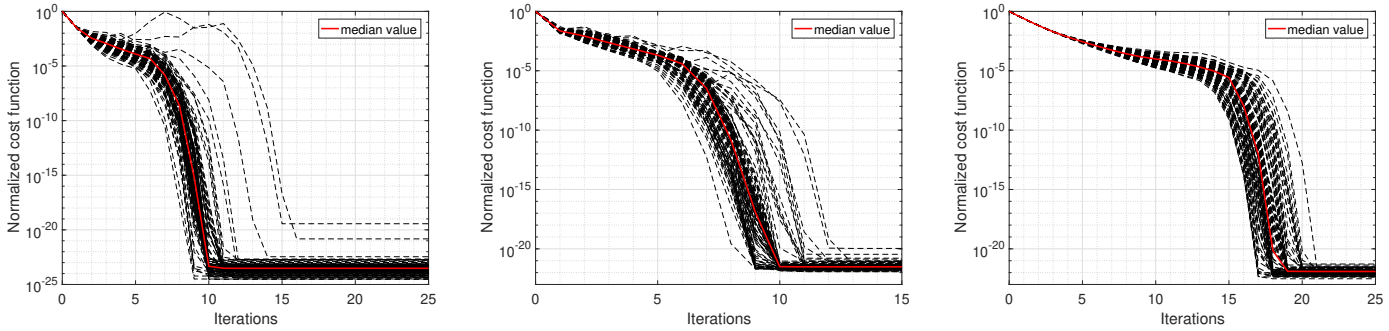
B. Comparison study

We now investigate practical advantages and limitations of the proposed algorithms in comparison with several algorithms of the literature: SH-RT [19], JUST [20], JDTM [13], JET-U [15] and JAPAM-5 [23]. Our first comparison criterion is an indicator that quantifies the relative deviation between the estimated matrix of eigenvectors ($\mathbf{B}^{-1}$) and the actual matrix of eigenvectors ($\mathbf{A}$). After removing scaling and permutation



(a) JDTE, $N = 5$ (b) SJDTE, $N = 16$ (c) WJDTE, $N = 100$

Fig. 1: Convergence plots with random matrices of eigenvectors.



(a) JDTE, $N = 5$ (b) SJDTE, $N = 16$ (c) WJDTE, $N = 100$

Fig. 2: Convergence plots with ill-conditioned matrix of eigenvectors.

indeterminacy, this indicator is defined as the Normalized Root Mean Square Error (NRMSE) between $\mathbf{A}$ and $\mathbf{B}^{-1}$:

$$r_A = \frac{||\mathbf{A} - \mathbf{B}^{-1}||}{||\mathbf{A}||}. \quad (44)$$

The second criterion, denoted Γ_{tot}, is the computational cost, defined as the product between the numerical complexity and the number of computed iterations to reach the stopping criterion. The stopping criterion is defined here as

$$\frac{|C(\mathbf{X}_{it}) - C(\mathbf{X}_{it-1})|}{|C(\mathbf{X}_{it-1})|} < 10^{-6} \text{ or } C(\mathbf{X}_{it}) > 10^5 C(\mathbf{X}_0) \quad (45)$$

where $\mathbf{X}_{it}$ is the updating matrix at iteration it. For each algorithm the maximal number of iterations is set to 500.

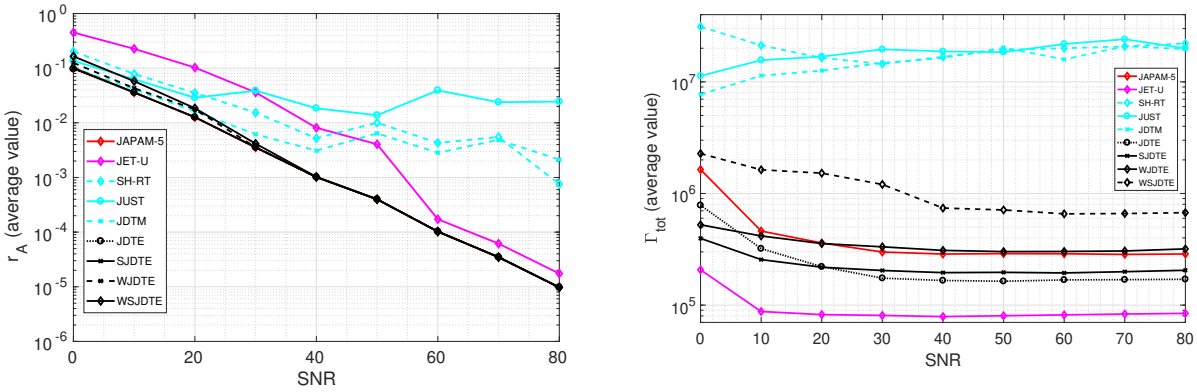
Algorithms are compared according to three scenarios. In the first scenario, we vary the SNR value. In the second one we vary the matrix size N. In the last one we vary the number of degenerated eigenvalues of two matrices $\mathbf{M}^{(k)}$. For each scenario, algorithm performances are evaluated by comparing the average values of r_A and Γ_{tot} computed from 100 Monte-Carlo (MC) runs. Each MC run corresponds to a new matrix set built as previously explained.

1) Scenario 1: We set the number of matrices to $K = 20$, the matrix size to $N = 5$ and we vary the SNR value from 0 dB to 80 dB by step of 10 dB. Algorithms are initialized with the identity matrix.

Results of this first scenario are plotted on figure 3. Concerning the average value of r_A (figure 3a), the four proposed algorithms clearly provide the best results along with JAPAM-5. Other polar decomposition based algorithms compete only for the lowest SNR value while JET-U is consistently outperformed. On the other hand, we can see that JET-U has the lowest computational cost for any SNR value (figure 3b). In contrast, SH-RT, JUST and JDTM are the most expensive. JDTE and SJDTE have a lower computational cost than JAPAM-5 for all the SNR values. WJDTE and JAPAM-5 have similar cost while WSJDTE is more expensive.

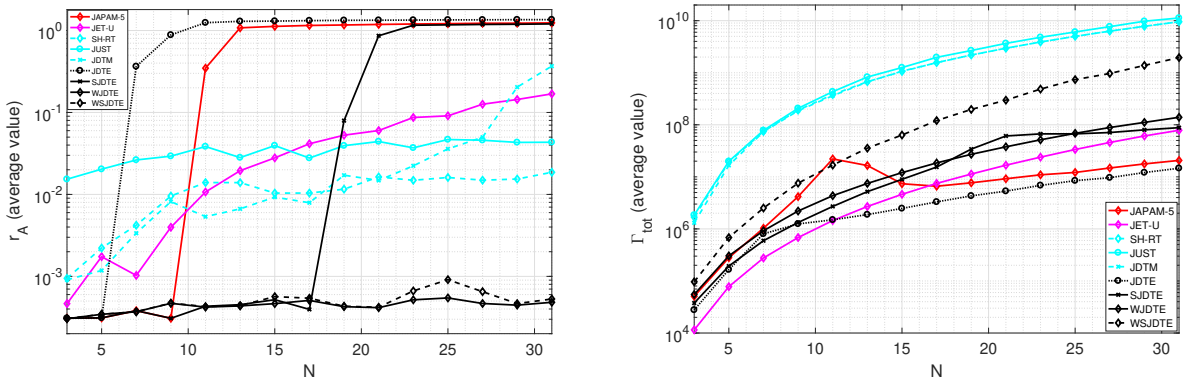
Thus, this scenario shows that for small matrix sizes, JDTE and SJDTE compete with the best algorithms of the literature for the estimation of the matrix of eigenvectors. Moreover, their computational costs are significantly lower, whatever the considered SNR value.

2) Scenario 2.a: We set the number of matrices to $K = 20$, the SNR value to 50 dB and we vary the matrix size from 3 to 31 by step of 2. Matrix $\mathbf{B}$ is still initialized with the identity



(a) r_A average value (b) Average computational cost

Fig. 3: Scenario 1, average r_A and Γ_{tot} versus the SNR.



(a) r_A average value (b) Average computational cost

Fig. 4: Scenario 2.a, average r_A and Γ_{tot} versus the matrix size.

matrix.

First of all, we see that the average value of r_A (figure 4a) obtained from JAPAM-5, JDTE and SJDTE dramatically increases above a critical value of N. Thereby, we can define an operating limit for these three algorithms. This is not surprising because these algorithms are based on assumptions and the validity of these assumptions decreases when the matrix size increases. This behavior has already been observed in [23] for the JAPAM algorithms. Here the operating limits of JAPAM-5, JDTE and SJDTE are $N = 13$, $N = 9$ and $N = 21$ respectively. Thus our simultaneous optimization scheme is more sensitive to the matrix size than the sequential one. It is noteworthy that performances of WJDTE and WSJDTE are not affected by the increase of the matrix size. Indeed these algorithms provide very good results whatever the value of N. This means that the weight mitigates the influence of Assumption 1 and Assumption 2 as it was expected. Finally, we can note that inside their operating ranges, JDTE and SJDTE provide similar results to WJDTE and WSJDTE. Average computational costs are plotted on figure 4b. JET-U is the least expensive algorithm from $N = 3$ to $N = 11$. JDTE and SJDTE have a lower average computational cost than JAPAM-5 in its operating range ($N < 13$). This is also the case of WJDTE for $N = 7$ to $N = 13$. WSJDTE is clearly more expensive than JAPAM-5 but it is still less expensive than SH-RT, JUST and JDTM, whatever the considered value of N.

From this scenario, we can conclude that WJDTE is clearly the best choice when dealing with large matrices ($N > 17$ for the settings considered here). For smaller matrices, we can recommend SJDTE because it is slightly less expensive. Finally, with this initialization, JDTE is suitable only for small matrices.

In [23], the sensitivity of JAPAM algorithms to the matrix size is mitigated by initializing the diagonalizing matrix $\mathbf{B}$ with a generalized eigenvalue decomposition. As previously mentioned, this kind of initialization allows to respect Assump-

tion 1 and Assumption 2 from the first iteration. Thus, in the next two scenarios, we investigate the behavior of the JEVD algorithms when GEVD initialization is used.

3) Scenario 2.b: We keep the same settings as scenario 2.a but here the matrix $\mathbf{B}$ is initialized by GEVD (of course the same GEVD is used for all the algorithms).

Results are plotted on figure 5. As expected, regarding the r_A criterion, algorithms are much less sensitive to the matrix size. Here, there is no difference between the four proposed algorithms and JAPAM-5. However, JDTE and SJDTE still have a significantly lower computational cost than JAPAM-5, whatever the matrix size. Here one should be careful of the logarithmic scale in figure 5b: the computational cost of JDTE is twice lower than the one of JAPAM-5. Thereby, JDTE appears here as a good option when it is well initialized. Indeed, it provides a good estimation of the eigenvectors with a rather low computational cost. However, it can happen that the GEVD does not provide a good initialization. We are going to study one of these cases in our last scenario.

4) Scenario 3: We set the number of matrices to $K = 20$, the SNR value to 50 dB and the matrix size to $N = 25$. Matrix $\mathbf{B}$ is initialized by GEVD but here we vary the algebraic multiplicity of the first eigenvalue of each of the two matrices used to compute the GEVD. We denote P this number and we vary it from 5 to 25.

Results are plotted on figure 6. Performances of JDTE, JAPAM-5 and SJDTE dramatically decrease above a critical value of P. This value is respectively equal to 7, 8 and 17. Conversely, WJDTE is not affected by this bad initialization and it consistently provides an optimal estimation of the matrix of eigenvectors. The same remark holds for WSJDTE but it is much more expensive. JDTE and SJDTE have a lower computational cost than JAPAM-5 inside its operating range. This is also the case of WJDTE for $P > 6$.

V. Application to Canonical Polyadic Decomposition and MIMO transmission system

JEVD can be used to compute the canonical polyadic decomposition of tensors. Such a decomposition has become an important tool for digital telecommunication signal processing. In particular, CPD can be used in MIMO systems where the source separation step at the receiver can be performed in a deterministic way. This approach has originally been proposed in [28] for Direct-Sequence Code Division Multiple Access (DS-CDMA) systems. Later, it has been generalized thanks to the concept of Khatri-Rao Space-Time (KRST) coding [29], [30], [31], [32]. At the transmission, KRST coding consists in spreading the symbols of the source signals by a different code sequence for each source signal. At the reception, mixed coded signals are gathered in a third order data tensor. Then a CPD of this tensor provides a deterministic estimation of the source signals, the mixing matrix and the code matrix. Another advantage of this approach is that it can deal with under-determined mixtures of source signals. We show here that the proposed JEVD algorithms help to improve the CPD of such data tensors.

In this purpose, we consider the following simple MIMO system: At the emission, N source signals are coded by a $Q \times N$ code matrix $\mathbf{C}$ (each source symbol is spread by a code sequence of length Q). The code matrix is chosen as a truncated discrete Fourier transform matrix as in [31]. Source signals consist in random sequences of P QPSK symbols gathered in matrix $\mathbf{S}$ ($P \times N$). At the reception, an array of R antennas receives R linear mixtures of the N emitted signals. These linear mixtures are modeled by a random mixing matrix $\mathbf{H}$ ($R \times N$) of complex numbers. A white Gaussian noise is added to the mixed signals. Thus each receiving antenna receives a sequence of PQ symbols. After shifting and downsampling operations, the collected symbols are gathered in a $P \times Q \times R$ complex-valued tensor $\boldsymbol{\mathcal{X}}$. Then, the rank N CPD of $\boldsymbol{\mathcal{X}}$ gives

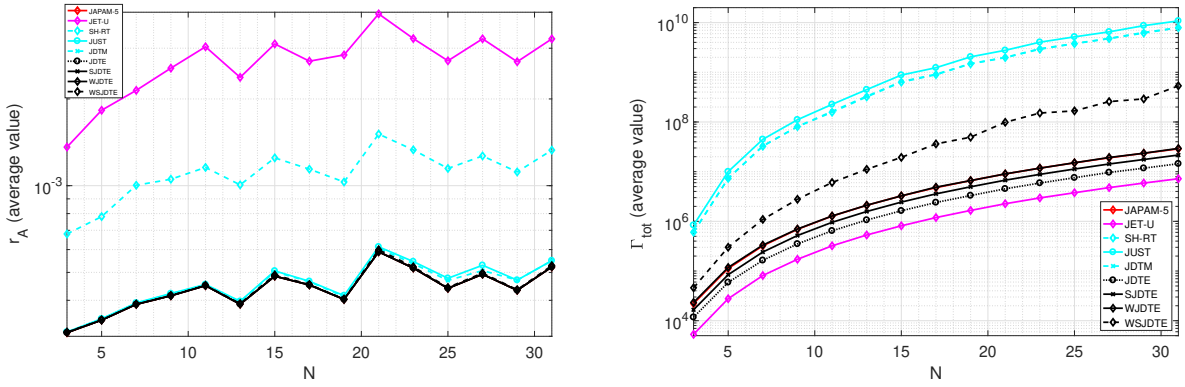
$$\mathcal{X}_{pqr} = \sum_{n=1}^{N} S_{pn} C_{qn} H_{rn} + \mathcal{E}_{pqr}, \ \forall (p,q,r) \tag{46}$$

where the tensor $\mathcal{E}$ models the noise. Here the CPD is computed using the DIAG algorithm [13]. The crucial step of this algorithm is the JEVD of $K = R(R-1)$ matrices of size N. In [23], we showed that in a similar context this approach significantly outperforms classical CPD algorithms such as the alternative least square. JAPAM-5 then appeared as the best solution for the JEVD step. As a consequence, we compare here the performances of JDTE, WJDTE and SJDTE with those of JAPAM-5.

We take $P = 64$, $Q = 16$, $R = 10$ and $N = 13$. Thereby, the JEVD problem involves 90 matrices of size 13. We have chosen a (quite) large number of source signals in order to emphasize the differences between JEVD algorithms and deal with under-determined mixtures. Furthermore, these (relatively) small tensor dimensions should increase the difficulty of the problem. We define here the SNR as

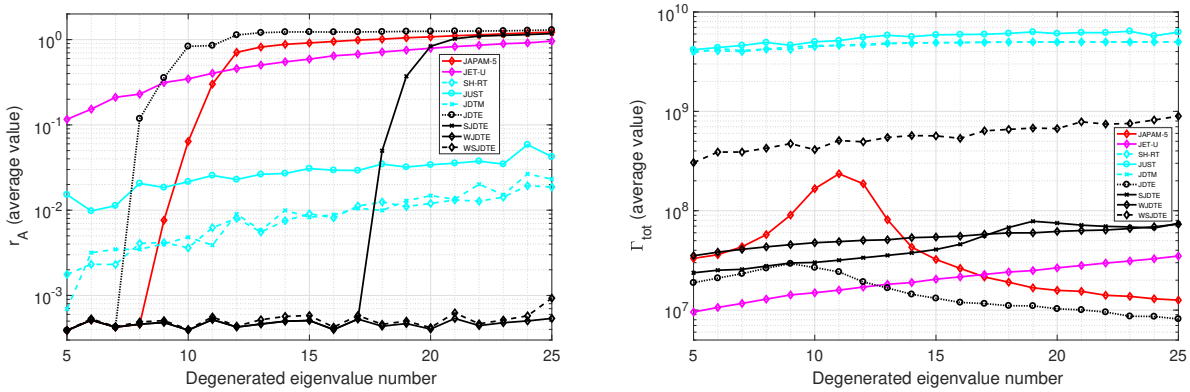
$$\text{SNR} = 20 \log \left(\frac{||\mathcal{X} - \mathcal{E}||}{||\mathcal{E}||} \right) \tag{47}$$

and we vary it from -4 to 8 dB by step of 1 dB. Comparisons are performed by means of MC simulations. For each SNR value, 500 data tensors are built according to equation (46) from random draws of $\mathbf{H}$ and $\boldsymbol{\mathcal{E}}$ while each column of $\mathbf{S}$ is built as the QPSK modulation of a random binary signal of length $2P$. JDTE, WJDTE and SJDTE are initialized by the identity matrix while two kinds of initializations are considered for JAPAM-5: identity matrix and GEVD. Comparison criteria are the Bit Error Rate (BER) computed from the estimated source matrix after demodulation, the average NRMSE between the estimated and the actual factor matrices (computed over the three matrices) and the computational cost Γ_{tot} of the JEVD step. Average values of the three criteria (computed over the MC runs) are plotted on figures 7a, 7b and 7c respectively. Regarding the BER criterion, SJDTE and WJDTE provide the best results and clearly outperform JAPAM even with the GEVD initialization. The gap with JAPAM increases as the SNR decreases. In the highest SNR range ([5;8] dB), WJDTE seems more stable than SJDTE. It is interesting to note that in spite of the large value of N, JDTE results are better than those of JAPAM initialized with the GEVD in



(a) r_A average value (b) Average computational cost

Fig. 5: Scenario 2.b, average r_A and Γ_{tot} versus the matrix size (GEVD initialization).



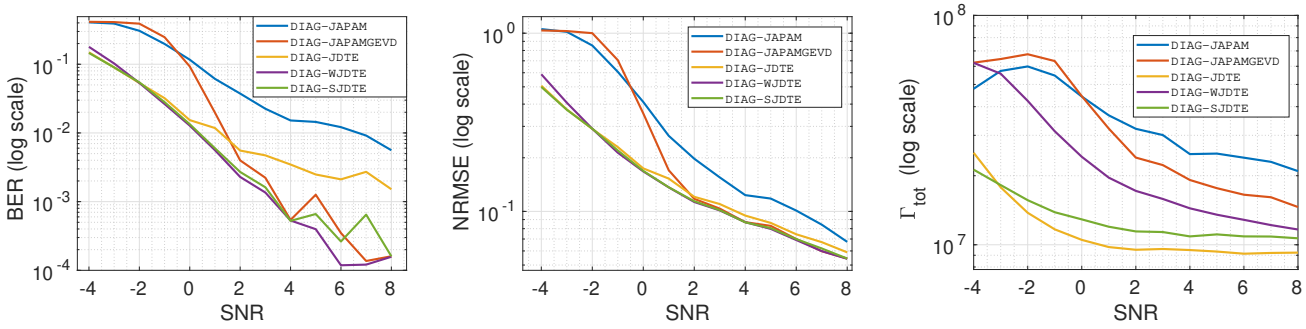
(a) r_A average value (b) Average computational cost

Fig. 6: Scenario 3, average r_A and Γ_{tot} versus the number of degenerated eigenvalues (GEVD initialization).

the range $[-4; 1]$ dB and are consistently better than those of JAPAM initialized with the identity. Now, regarding the NRMSE criterion, JDTE, SJDTE and WJDTE provide similar results and clearly improve JAPAM results whatever the used initialization in the range $[-4; 1]$ dB. For higher SNR, JAPAM-GEVD competes with the proposed algorithms. The good performances of JDTE can be explained here by the large value of the ratio K/N. Finally, SJDTE is less costly than WJDTE which is less costly than JAPAM-5 with GEVD initialization. However, JDTE remains the least costly algorithm (except at -4 dB). In conclusion, SJDTE appears here as the best choice followed by WJDTE. Nevertheless, if the computational cost is the most important criterion, one may consider JDTE as a good option.

VI. Conclusion

In the present paper, we have introduced an original approach to compute the joint eigenvalue decomposition of a set of matrices. This approach has two main advantages. First, it equally works in $\mathbb{R}$ and $\mathbb{C}$. Second, it allows to derive several iterative algorithms that are based on very simple analytical expressions of the updating parameters. The first one, called JDTE has a low numerical complexity and works very well for small matrix sizes. The counterpart is that it is quite sensitive to the matrix size in the absence of smart initialization. In order to deal with this issue, we have proposed an improved version of JDTE called WJDTE. The numerical complexity increases but we obtain the desired robustness with respect to the matrix size. Actually, in our numerical simulations, WJDTE has outperformed all the other algorithms in all the considered situations. It is also remarkably stable whatever the matrix size, the conditioning of the matrix of eigenvectors and the initialization. The third algorithm (SJDTE) is a sequential version of JDTE. SJDTE is a very versatile algorithm: its numerical complexity is between those of JDTE and WJDTE and it is as efficient as WJDTE for small to medium matrix



(a) BER average value (b) NRMSE average value (c) Average computational cost

Fig. 7: Algorithms comparison for source separation in a MIMO system context.

sizes. However, for larger matrix sizes, WJDTE should be preferred. The last algorithm (WSJDTE) is a sequential version of WJDTE. In all of our numerical simulations, WSJDTE and WJDTE have provided similar results. However, the computational cost of WSJDTE is significantly higher.

Another way to improve the performances of the algorithms with respect to the matrix size is to initialize the algorithms with a generalized eigenvalue decomposition. In this case, JDTE appears as the best trade-off between estimation performances and computational cost. However, GEVD does not always provide a good starting point and it appears that only WJDTE and WSJDTE are totally robust to a bad initialization.

Finally, we have shown that WJDTE and SJDTE outperform the reference JEVD algorithm for the canonical polyadic decomposition of complex-valued tensors in the context of source separation of digital telecommunications signals.

APPENDIX A
PROOF OF PROPOSITION 1

An important point of the proposed algorithms is that the denominator of Z_{mn} in equation (22) must not be equal to zero for all couples (m,n) with $m \neq n$. Proposition 1 gives us a sufficient condition to ensure this point. We give here a proof of this proposition. Let us first recall the proposition.

Proposition 1: Let $\mathbf{B}$ be the estimation of $\mathbf{A}^{-1}$ at the current iteration and $\mathbf{G} = \mathbf{BA}$. If $\mathbf{G}$ is a strictly diagonally dominant matrix (up to permutation) and if matrix $\mathbf{\Omega}$ defined in equation (2) has at least two non co-linear rows then there exists at least one couple (n,m) with $m \neq n$ such that the denominator of Z_{mn} in (22) is not null.

Proof:

Denominator of Z_{mn} is non null if and only if $\Lambda_{mm}^{(k)} \neq \Lambda_{nn}^{(k)}$ for all k. In other words, we want to show that the matrix

$$\mathbf{\Delta} = \begin{pmatrix} \Lambda_{11}^{(1)} & \cdots & \Lambda_{11}^{(K)} \\ \vdots & \cdots & \vdots \\ \Lambda_{NN}^{(1)} & \cdots & \Lambda_{NN}^{(K)} \end{pmatrix} \tag{48}$$

has at least two distinct rows.

First, let us establish the link between matrices $\mathbf{\Delta}$ and $\mathbf{\Omega}$. $\mathbf{G}$ is strictly diagonally dominant, this implies that both $\mathbf{G}$ and $\mathbf{B}$ are invertible. Thus at the current iteration, injecting (1) in (3) yields

$$\mathbf{N}^{(k)} = \mathbf{G}\mathbf{D}^{(k)}\mathbf{G}^{-1}. \tag{49}$$

Let $\mathrm{diag}\{\cdot\}$ be the operator that puts the diagonal entries of the matrix in argument in a column vector. This yields

$$\mathrm{diag}\{\mathbf{N}^{(k)}\} = (\mathbf{G} \boxdot \mathbf{G}^{-T})\ \mathrm{diag}\{\mathbf{D}^{(k)}\}, \tag{50}$$

where $\boxdot$ is the matrix Hadamard product. Finally, denoting $\mathbf{H} = \mathbf{G} \boxdot \mathbf{G}^{-T}$, we obtain

$$\mathbf{\Delta} = \mathbf{H}\mathbf{\Omega}. \tag{51}$$

The difference between two rows of $\mathbf{\Delta}$ is thus given by

$$\mathbf{\Delta}_{m.} - \mathbf{\Delta}_{n.} = (\mathbf{H}_{m.} - \mathbf{H}_{n.})\mathbf{\Omega}, \tag{52}$$

where $\mathbf{\Delta}_{m.}$, $\mathbf{\Delta}_{n.}$, $\mathbf{H}_{m.}$ and $\mathbf{H}_{n.}$ are the m^{th} and the n^{th} rows of matrices $\mathbf{\Delta}$ and $\mathbf{H}$ respectively. As a consequence,

$$\forall (m,n),\ m \neq n,\ \ \mathbf{\Delta}_{m.} - \mathbf{\Delta}_{n.} \neq \mathbf{0} \Leftrightarrow \mathbf{\Omega}^T(\mathbf{H}_{m.} - \mathbf{H}_{n.})^T \neq \mathbf{0} \tag{53}$$

where $\mathbf{0}$ is the null vector. Thus, we only need to show that there exists a couple (m,n), $m \neq n$ such that $(\mathbf{H}_{m.} - \mathbf{H}_{n.})^T$ is not in the kernel of $\mathbf{\Omega}^T$.

We first show that $\mathbf{H}$ is full rank. Because $\mathbf{G}$ is strictly diagonally dominant, $\mathbf{G}^{-T}$ is also strictly diagonally dominant (see [33] page 27). Thereby $\mathbf{H}$ is strictly diagonally dominant too and consequently it is nonsingular and full rank. Thus, on one hand the subspace spanned by the set of vectors $\{(\mathbf{H}_{m.} - \mathbf{H}_{n.})^T\}_{\substack{m,n \\ m \neq n}}$ is of dimension $N-1$. On the other hand, we have assumed that matrix $\mathbf{\Omega}^T$ has at least two non co-linear rows thereby the dimension of the space spanned by its columns is greater or equal to 2. Consequently, the dimension of $\mathrm{Ker}(\mathbf{\Omega}^T)$ is strictly lower than $N-1$.

Thus, there exists at least one couple (m,n), $m \neq n$ such that $(\mathbf{H}_{m.} - \mathbf{H}_{n.})^T$ is not in the kernel of $\mathbf{\Omega}^T$. ■

REFERENCES

[1] P. Comon, "Independent Component Analysis," in Higher Order Statistics, J-L. Lacoume, Ed., pp. 29–38. Elsevier, Amsterdam, London, 1992.

[2] J. F. Cardoso and A. Souloumiac, "Blind beamforming for non-gaussian signals," IEE Proceesings-F, vol. 140, no. 6, pp. 362–370, Dec. 1993.

[3] J. F. Cardoso and A. Souloumiac, "Jacobi angles for simultaneous diagonalization," SIAM Journal Matrix Analysis and Applications, vol. 17, no. 1, pp. 161–164, 1996.

[4] A. Souloumiac, "Nonorthogonal joint diagonalization by combining givens and hyperbolic rotations," IEEE Transactions on Signal Processing, vol. 57, no. 6, pp. 2222–2231, June 2009.

[5] A. Yeredor, "Non-orthogonal joint diagonalization in the least-squares sense with application in blind source separation," IEEE Transactions on Signal Processing, vol. 50, no. 7, pp. 1545–1553, jul 2002.

[6] Guang-Hui Cheng, Shan-Man Li, and Eric Moreau, "New jacobi-like algorithms for non-orthogonal joint diagonalization of hermitian matrices," Signal Processing, vol. 128, pp. 440 – 448, 2016.

[7] A. J. Van der Veen, P. B. Ober, and E. F. Deprettere, "Azimuth and elevation computation in high resolution doa estimation," IEEE Transactions on Signal Processing, vol. 40, no. 7, pp. 1828–1832, July 1992.

[8] A. N. Lemma, A. J. Van der Veen, and E. F. Deprettere, "Analysis of joint angle-frequency estimation using ESPRIT," IEEE Transactions on Signal Processing, vol. 51, no. 5, pp. 1264–1283, May 2003.

[9] M. Haardt and J.A. Nossek, "Simultaneous schur decomposition of several nonsymmetric matrices to achieve automatic pairing in multidimensional harmonic retrieval problems," IEEE Transactions on Signal Processing, vol. 46, no. 1, pp. 161–169, January 1998.

[10] L. De Lathauwer, B. De Moor, and J. Vandewalle, "Computation of the canonical decomposition by means of a simultaneous Schur decomposition," SIAM Journal on Matrix Analysis and Applications, vol. 26, no. 2, pp. 295–327, 2004.

[11] F. Roemer and M. Haardt, "A closed-form solution for multilinear parafac decompositions," in SAM 08, Fifth IEEE Sensor Array and Multichannel Signal Processing Workshop, july 2008, pp. 487–491.

[12] F. Roemer and M. Haardt, "A semi-algebraic framework for approximate cp decompositions via simultaneous matrix diagonalizations (secsi)," Signal Processing, vol. 93, no. 9, pp. 2722 – 2738, 2013.

[13] X. Luciani and L. Albera, "Canonical polyadic decomposition based on joint eigenvalue decomposition," Chemometrics and Intelligent Laboratory Systems, vol. 132, no. 0, pp. 152 – 167, 2014.

[14] A. Boudjellal, A. Mesloub, K. Abed-Meraim, and A. Belouchrani, "Separation of dependent autoregressive sources using joint matrix diagonalization," IEEE Signal Processing Letters, vol. 22, no. 8, pp. 1180–1183, Aug 2015.

[15] X. Luciani and L. Albera, "Joint eigenvalue decomposition of non-defective matrices based on the LU factorization with application to ICA," IEEE Transactions on Signal Processing, vol. 63, no. 17, pp. 4594–4608, Sept 2015.

[16] S. Bonhomme, K. Jochmans, and J.-M. Robin, "Nonparametric estimation of non-exchangeable latent-variable models," Journal of Econometrics, vol. 201, no. 2, pp. 237 – 248, 2017.

[17] A. Mesloub, K. Abed-Meraim, and A. Belouchrani, "A new algorithm for complex non-orthogonal joint diagonalization based on shear and givens rotations," IEEE Transactions on Signal Processing, vol. 62, no. 8, pp. 1913–1925, April 2014.

[18] P. Strobach, "Bi-iteration multiple invariance subspace tracking and adaptive esprit," IEEE Transactions on Signal Processing, vol. 48, pp. 442–456, 2000.

[19] T. Fu and X. Gao, "Simultaneous diagonalization with similarity transformation for non-defective matrices," in ICASSP 2006, 2006 IEEE International Conference on Acoustics Speech and Signal Processing, May 2006, vol. 4, pp. 1137–1140.

[20] R. Iferroudjene, K. Abed-Meraim, and A. Belouchrani, "A new jacobi-like method for joint diagonalization of arbitrary non-defective matrices," Applied Mathematics and Computation, vol. 211, no. 2, pp. 363–373, 2009.

[21] A. Mesloub, A. Belouchrani, and K. Abed-Meraim, "Efficient and stable joint eigenvalue decomposition based on generalized givens rotations," in 2018 26th European Signal Processing Conference (EUSIPCO), Sep. 2018, pp. 1247–1251.

[22] E. R. Balda, S. A. Cheema, A. Weiss, A. Yeredor, and M. Haardt, "Perturbation analysis of joint eigenvalue decomposition algorithms," in 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), March 2017, pp. 3101–3105.

[23] R. André, Xavier Luciani, and Eric Moreau, "A new class of block coordinate algorithms for the joint eigenvalue decomposition of complex matrices," Signal Processing, vol. 145, pp. 78 – 90, 2018.

[24] T. Trainini and E. Moreau, "A coordinate descent algorithm for complex joint diagonalization under hermitian and transpose congruences," IEEE Transactions on Signal Processing, vol. 62, no. 19, pp. 4974–4983, Oct 2014.

[25] R. André, T. Trainini, X. Luciani, and E. Moreau, "A fast algorithm for joint eigenvalue decomposition of real matrices," in EUropean SIgnal Processing COnference (EUSIPCO'2015), Nice, France, 2015.

[26] V. Maurandi and E. Moreau, "A decoupled jacobi-like algorithm for non-unitary joint diagonalization of complex-valued matrices," IEEE Signal Processing Letters, vol. 21, no. 12, pp. 1453–1456, Dec 2014.

[27] G. W. Stewart, "Matrix Algorithms: Volume 1, Basic Decompositions", Society for Industrial Mathematics, 1998.

[28] N. D. Sidiropoulos, G. B. Giannakis, and R. Bro, "Blind PARAFAC receivers for DS-CDMA systems," IEEE Transactions On Signal Processing, vol. 48, no. 8, pp. 810–823, March 2000.

[29] D. Nion and L. De Lathauwer, "A Block Component Model based Blind DS-CDMA Receiver," IEEE Trans. Signal Proc., vol. 56, no. 11, pp. 5567–5579, 2008.

[30] A. de Almeida, G. Favier, and J.C. Mota, "Space-time spreading mimo-cdma downlink systems using constrained tensor modeling," Signal Processing, vol. 88, no. 10, pp. 2403 – 2416, 2008.

[31] L. R. Ximenes, G. Favier, A. L. F. de Almeida, and Y. C. B. Silva, "Parafac-paratuck semi-blind receivers for two-hop cooperative mimo relay systems," IEEE Transactions on Signal Processing, vol. 62, no. 14, pp. 3604–3615, July 2014.

[32] L. R. Ximenes, G. Favier, and A. L. F. de Almeida, "Semi-blind receivers for non-regenerative cooperative mimo communications based on nested parafac modeling," IEEE Transactions on Signal Processing, vol. 63, no. 18, pp. 4985–4998, Sep. 2015.

[33] B. Li, "Generalizations of diagonal dominance in matrix theory", Ph.D. thesis, University of Regina, 1997.