New algorithms on Complex Joint Eigenvalue Decomposition Based on Generalized Givens Rotations

Ammar MESLOUB
Lab. Traitement du signal
Ecole Militaire Polytechnique
BP 17 Bordj El Bahri, Algeria
mesloub.a@gmail.com

Adel BELOUCHRANI
Electrical Engineering Dept.LDCCP
Ecole Nationale Polytechnique
16200 Algiers, Algeria
adel.belouchrani@enp.edu.dz

Karim ABED-MERAIM
PRISME Lab.
Univ. Orléans
45067 Orléans, France
karim.abed-meraim@univ-orleans.fr

***Abstract*—In this paper, new joint eigenvalue decomposition (JEVD) methods are developed by considering generalized Givens rotations. These algorithms deal with a set of square complex matrices sharing a same eigen-structure. Several existing methods, using or not generalized Givens rotations, have treated the aforementioned problem. To improve the JEVD solutions, we developed two methods, the first one is numerically stable and efficient but relatively expensive. The second one is developed by considering some justified approximations. Simulation results are provided to highlight the effectiveness and behaviour of the proposed techniques for different scenarios.**

***Index Terms*—Complex Joint EigenValue Decomposition (JEVD), Complex Efficient and Stable Joint eigenvalue Decomposition algorithm (CESJD), generalized Givens rotations, exact JEVD, approximative JEVD.**

I. Introduction

In this paper, we mainly propose new algorithms to solve the Joint EigenValue Decomposition (JEVD) of a set of complex non-defective matrices based on generalized Givens rotations. This JEVD problem plays an important role in several applications such as Canonical Polyadic Decomposition (CPD) of tensors [1], [2], multi-dimensional harmonic retrieval [3], joint angle-delay estimation [4], Direction of arrival estimation [5] and Blind Sources Separation (BSS) [6].

The JEVD problem is widely treated in the literature by using different schemes. In [7], [8], generalized Givens rotations have been used in the complex case and in [9], the LU decomposition is used.

The JEVD can be defined by considering K complex square matrices of dimension $N \times N$ sharing the following joint structure (exact JEVD case):

$$\mathbf{M}_k = \mathbf{A}\mathbf{D}_k\mathbf{A}^{-1} \tag{1}$$

Where $k \in \{1, ..., K\}$, K is the number of matrices, N is the matrix dimension. $\mathbf{A}$ is a square non defective matrix (referred to as mixing matrix in the BSS context) and $\mathbf{D}_k$ is the k^{th} diagonal matrix associated to the k^{th} matrix $\mathbf{M}_k$.
The problem consists of looking for $\{\mathbf{A}, \mathbf{D}_1, \cdots, \mathbf{D}_K\}$ by using only the set of the K complex matrices $\{\mathbf{M}_1, \cdots, \mathbf{M}_K\}$. It can be also defined by finding a matrix $\mathbf{V}$ which makes the set of matrices $\{\mathbf{V}\mathbf{M}_1\mathbf{V}^{-1}, \cdots, \mathbf{V}\mathbf{M}_K\mathbf{V}^{-1}\}$ as diagonal as possible specially in the approximate JEVD case (see section III for more details).

In this paper, we investigate generalized Givens rotations applied to the JEVD problem. The unknown matrix $\mathbf{A}$ is decomposed in a product of generalized Givens rotations, according to:

$$\mathbf{A} = \prod_{\#\text{sweeps}} \prod_{1 \le i < j \le N} \mathbf{S}_{ij}\mathbf{G}_{ij} \tag{2}$$

where $\#$sweeps represents the number of iterations, $\mathbf{S}_{ij}$ and $\mathbf{G}_{ij}$ are the elementary Shear and Givens rotations, respectively. The problem of JEVD reduces then in the estimation of these elementary rotations.

II. Proposed methods

The elementary rotations of equation (2) can be expressed according to two different schemes. The first one is based on sinus and hyperbolic sinus which needs some complicated developments leading to an efficient method. The second scheme allows appropriate approximations that lead to a computationally simplified method.

A. Complex JEVD method

The first scheme considers elementary rotations that are equal to the identity matrix except for $(i,i)^{th}$, $(i,j)^{th}$, $(j,i)^{th}$ and $(j,j)^{th}$ entries which are:

$$\begin{bmatrix} G_{ij}(i,i) & G_{ij}(i,j) \\ G_{ij}(j,i) & G_{ij}(j,j) \end{bmatrix} = \begin{bmatrix} \cos(\theta) & e^{-\jmath\varphi}\sin(\theta) \\ -e^{\jmath\varphi}\sin(\theta) & \cos(\theta) \end{bmatrix} \tag{3}$$

$$\begin{bmatrix} S_{ij}(i,i) & S_{ij}(i,j) \\ S_{ij}(j,i) & S_{ij}(j,j) \end{bmatrix} = \begin{bmatrix} \cosh(y) & e^{\jmath\alpha}\sinh(y) \\ e^{-\jmath\alpha}\sinh(y) & \cosh(y) \end{bmatrix} \tag{4}$$

θ and φ are the Givens rotation parameters while y and α represent the Shear rotations parameters.

The objective of this subsection is to generalize the method proposed in [10] to the complex case. Hence, only the Shear rotation is studied where the Givens rotation is obtained by applying the solution given in [6].

978-9-0827-9703-9/19/$31.00 ©2019 IEEE

Let us study the transformed matrices by the Shear transform. Each k^{th} transformed matrix is

$$\mathbf{M}'_k \leftarrow \mathbf{S}_{ij}(-y,\alpha)\mathbf{M}_k\mathbf{S}_{ij}(y,\alpha)$$

Where only i^{th}, j^{th} rows and columns are affected by the Shear transformation. Modified entries can be written as:

$$\begin{aligned} M'_k(i,j)e^{-\jmath\alpha} = \; & \tfrac{M_k(i,j)e^{-\jmath\alpha}+M_k(j,i)e^{\jmath\alpha}}{2} \\ & + \tfrac{M_k(i,j)e^{-\jmath\alpha}-M_k(j,i)e^{\jmath\alpha}}{2}\cosh(2y) \\ & + \tfrac{M_k(i,i)-M_k(j,j)}{2}\sinh(2y) \end{aligned} \tag{5}$$

$$\begin{aligned} M'_k(j,i)e^{\jmath\alpha} = \; & \tfrac{M_k(i,j)e^{-\jmath\alpha}+M_k(j,i)e^{\jmath\alpha}}{2} \\ & - \tfrac{M_k(i,j)e^{-\jmath\alpha}-M_k(j,i)e^{\jmath\alpha}}{2}\cosh(2y) \\ & - \tfrac{M_k(i,i)-M_k(j,j)}{2}\sinh(2y) \end{aligned} \tag{6}$$

$$\begin{aligned} M'_k(l,i) = \; & M_k(l,i)\cosh(y) + M_k(l,j)e^{-\jmath\alpha}\sinh(y) \\ M'_k(l,j) = \; & M_k(l,i)e^{\jmath\alpha}\sinh(y) + M_k(l,j)\cosh(y) \\ M'_k(i,l) = \; & M_k(i,l)\cosh(y) - M_k(j,l)e^{\jmath\alpha}\sinh(y) \\ M'_k(j,l) = \; & -M_k(i,l)e^{-\jmath\alpha}\sinh(y) + M_k(i,j)\cosh(y) \end{aligned} \tag{7}$$

Minimizing the sum of square modulus of the off-diagonal entries of matrices $\mathbf{M}'_k$, $k = 1,\cdots,K$ is equivalent to minimizing the following criterion

$$C_T(y,\alpha) = C_1(y,\alpha) + C_2(y,\alpha) \tag{8}$$

where the first term, $C_1(y,\alpha)$ corresponds to the $(i,j)^{th}$ and $(j,i)^{th}$ entries as

$$\begin{aligned} C_1(y,\alpha) \; & = \textstyle\sum_{k=1}^{K} |M'_k(i,j)|^2 + |M'_k(j,i)|^2 \\ & = \textstyle\sum_{k=1}^{K} |M'_k(i,j)e^{-\jmath\alpha}|^2 + |M'_k(j,i)e^{\jmath\alpha}|^2 \end{aligned} \tag{9}$$

And the second term, $C_2(y,\alpha)$, contains the others entries affected by the Shear rotation.

$$\begin{aligned} C_2(y,\alpha) = \textstyle\sum_{k=1}^{K}\sum_{l=1,l\neq i,j}^{N} \; & |M'_k(i,l)|^2 + |M'_k(j,l)|^2 + \\ & |M'_k(l,i)|^2 + |M'_k(l,i)|^2 \end{aligned} \tag{10}$$

Some workouts of the two above equations leads to:

$$\begin{aligned} C_1(y,\alpha) &= \boldsymbol{v}^T\mathbf{Q}(\alpha)\boldsymbol{v} + \beta_1 \\ C_2(y,\alpha) &= \boldsymbol{v}^T\mathbf{g}(\alpha) + \beta_2 \end{aligned} \tag{11}$$

Where

$$\boldsymbol{v} = \begin{bmatrix} \cosh 2y \\ \sinh 2y \end{bmatrix} \tag{12}$$

and

$$\begin{aligned} \mathbf{Q}(\alpha) &= \Re\left[\mathbf{C}(\alpha)^H\mathbf{C}(\alpha)\right] \\ \mathbf{C}(\alpha) &= \begin{bmatrix} \frac{M_1(i,j)e^{-\jmath\alpha}-M_1(j,i)e^{\jmath\alpha}}{2} & \frac{M_1(i,i)-M_1(j,j)}{2} \\ \vdots & \vdots \\ \frac{M_K(i,j)e^{-\jmath\alpha}-M_K(j,i)e^{\jmath\alpha}}{2} & \frac{M_K(i,i)-M_K(j,j)}{2} \end{bmatrix} \end{aligned} \tag{13}$$

and

$$\begin{aligned} &\mathbf{g}(\alpha) = \textstyle\sum_{k=1}^{K}\sum_{l=1,l\neq i,j}^{N} \\ &\begin{bmatrix} |M_k(i,l)|^2 + |M_k(j,l)|^2 + |M_k(l,i)|^2 + |M_k(l,i)|^2 \\ 2\Re\left(M_k(l,i)\bar{M}_k(l,j)e^{\jmath\alpha} + M_k(i,l)\bar{M}_k(j,l)e^{-\jmath\alpha}\right) \end{bmatrix} \end{aligned} \tag{14}$$

β_1 and β_2 are constants independent from considered Shear rotation parameter. The notation $\bar{a}$ denotes the complex conjugate of a.

Note that, minimizing (8) has no closed form solution. So, we tried several methods to get optimal parameters (y, α) and found out that a robust manner to do it is to introduce two rotations the real and imaginary rotations. Where in the real one (resp. imaginary one), α is set to zero (resp. $\frac{\pi}{2}$).

For a fixed α value, the problem can be summarized as an optimization under constraint problem:

$$\begin{cases} \boldsymbol{v}^T\mathbf{Q}(\alpha)\boldsymbol{v} + \boldsymbol{v}^T\mathbf{g}(\alpha) \;\; (\alpha = 0 \text{ or } \frac{\pi}{2}). \\ \boldsymbol{v}^T\mathbf{J}\boldsymbol{v} = 1 \text{ and } v(1) > 0 \end{cases} \tag{15}$$

and $\mathbf{J} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$. The solution to this problem is treated in [7], [8], [11] and we have opted for the iterative scheme given in [10].

This iterative scheme can be summarized as a constrained minimization problem with the following Lagrangian.

$$L(\boldsymbol{v},\lambda) = \boldsymbol{v}^T\mathbf{Q}(\alpha)\boldsymbol{v} + \lambda\left(\boldsymbol{v}^T\mathbf{J}\boldsymbol{v} - 1\right) + \boldsymbol{v}^T\boldsymbol{g}(\alpha) \tag{16}$$

where λ is the Lagrangian factor.

To get this iterative scheme, the Lagrangian is derived with respect to λ and $\boldsymbol{v}$. Then, the following equations are obtained:

$$\lambda = \boldsymbol{v}^T\mathbf{Q}(\alpha)\boldsymbol{v} + \boldsymbol{v}^T\boldsymbol{g}(\alpha)$$

and

$$\boldsymbol{v} = -\frac{1}{2}\left(\mathbf{Q}(\alpha) + \lambda\mathbf{J}\right)^{-1}\boldsymbol{g}(\alpha)$$

The initialization is done by computing the solution of $\boldsymbol{v}$ given in [8]. The rest of iterations is summarized in Table I.

TABLE I
ITERATIVE FUNCTION TO GET OPTIMAL SHEAR PARAMETER $\boldsymbol{v}$

Require : $\mathbf{Q}(\alpha)$, $\boldsymbol{g}(\alpha)$, N_{max}, μ and τ
Initialization: $\boldsymbol{v}_1$ is the generalized eigenvector of the positive eigenvalue associated to ($\mathbf{Q}$, $\mathbf{J}$).
$\boldsymbol{v}_n \leftarrow \boldsymbol{v}_1$, $n \leftarrow 1$.
If $\|g\| > \mu$ **Then** $\boldsymbol{v} \leftarrow \boldsymbol{v}_n$ and exit function.
$C_r = \tau + 1$
while $C_r > \tau$ and $n < N_{max}$
  $\lambda \leftarrow \boldsymbol{v}_n^T\mathbf{Q}(\alpha)\boldsymbol{v}_n + \boldsymbol{v}_n^T\boldsymbol{g}(\alpha)$.
  $\boldsymbol{v}_t \leftarrow -\frac{1}{2}\left[\mathbf{Q}(\alpha) + \lambda\mathbf{J}\right]^{-1}\boldsymbol{g}(\alpha)$.
  $C_r \leftarrow \|\boldsymbol{v}_t - \boldsymbol{v}_n\|$
  $\boldsymbol{v}_n \leftarrow \boldsymbol{v}_t$, $n \leftarrow n+1$
end while.
if $v_n(1) < 0$ **Then** $\boldsymbol{v} \leftarrow -\boldsymbol{v}_n$
  Else $\boldsymbol{v} \leftarrow \boldsymbol{v}_n$
End if

The overall algorithm can be summarized as follows. Once the target complex matrices are given, the algorithm applies an iterative scheme to get the matrix $\mathbf{V}$ which is the inverse of the matrix $\mathbf{A}$ given in equation (1). Each iteration is realized by successive unitary and Shear transformations. The unitary transformation is obtained by using the solution given in [6].

The latter, $\mathbf{G}(\theta,\varphi)$, is used to update the matrices according to:

$$\begin{array}{rcl}\mathbf{V} & \leftarrow & \mathbf{V}\mathbf{G}(\theta,\varphi) \\ \mathbf{M}_k & \leftarrow & \mathbf{G}(\theta,\varphi)^H\mathbf{M}_k\mathbf{G}(\theta,\varphi)\end{array} \quad (17)$$

Once done, the Shear transformation is realized by considering two rotations. The first one is called real rotation by setting the value of α to zero. The second rotation called imaginary one sets the value of α to $\frac{\pi}{2}$. The real rotation (resp. the imaginary rotation) starts by computing $\mathbf{Q}(0)$ and $\mathbf{g}(0)$ (resp. $\mathbf{Q}(\frac{\pi}{2})$ and $\mathbf{g}(\frac{\pi}{2})$). Hence, the optimal value of $\boldsymbol{\nu}$ is obtained by applying the iterative function given in Table I using $\mathbf{Q}(0)$ and $\mathbf{g}(0)$ (resp. $\mathbf{Q}(\frac{\pi}{2})$ and $\mathbf{g}(\frac{\pi}{2})$). Once optimal $\boldsymbol{\nu}$ is obtained, Shear parameters are estimated by:

$$\left\{\begin{array}{rcl}\cosh(y) & = & \sqrt{\frac{1}{2}(v(1)-1)} \\ \sinh(y) & = & \frac{v(2)}{2\cosh(y)}\end{array}\right. \quad (18)$$

Next, the real (resp. the imaginary) rotation is completely defined as $\mathbf{S}(y,0)$ (resp. $\mathbf{S}(y,\frac{\pi}{2})$) and, the matrix update process is realized as follows:

$$\begin{array}{rcl}\mathbf{V} & \leftarrow & \mathbf{V}\mathbf{S}(\alpha,y) \\ \mathbf{M}_k & \leftarrow & \mathbf{S}(\alpha,-y)\mathbf{M}_k\mathbf{S}(\alpha,y)\end{array} \quad (19)$$

where α is set to zero (resp. $\frac{\pi}{2}$) in real (resp. imaginary) rotation. Finally, the developed algorithm, referred to as Complex Efficient and numerically Stable Joint eigenvalue Decomposition (CESJD), is given in Table II.

TABLE II
CESJD ALGORITHM

Require : $\mathbf{M}_k,\ k=1,\cdots,K$, fixed threshold τ and maximum sweep number M_{it}.
Initialization: $\mathbf{V}=\mathbf{I}_N$ and $\mathbf{A}=\mathbf{I}_N$.
while $\max_{i,j}(|y|,|\theta|)>\tau$ and (#sweeps $< M_{it}$)
  for all $1\le i<j\le N$
    Unitary transform
      Estimate $\mathbf{G}_{ij}(\theta,\varphi)$ using solution given in [6].
      Updates matrices $\mathbf{V},\mathbf{M}_k$ using equation (17).
    Real Shear transform
      Compute $\mathbf{Q}(0)$ and $\mathbf{g}(0)$ using equations (13) and (14).
      Estimate optimal $\boldsymbol{\nu}$ using function given in Table I.
      Compute Shear parameter using equation (18).
      Matrices updates using equation (19)
    Imaginary Shear transform
      Compute $\mathbf{Q}(\frac{\pi}{2})$ and $\mathbf{g}(\frac{\pi}{2})$ using equations (13) and (14).
      Estimate optimal $\boldsymbol{\nu}$ using function given in Table I.
      Compute Shear parameter using equation (18).
      Matrices updates using equation (19)
  end for
end while.

B. The simplified complex JEVD method

In this simplified method, another form of the elementary rotations is considered. Shear and Givens rotations are equal to the identity matrix except for some entries which are given by:

$$\begin{bmatrix} G_{ij}(i,i) & G_{ij}(i,j) \\ G_{ij}(j,i) & G_{ij}(j,j)\end{bmatrix} = \lambda_\theta \begin{bmatrix} 1 & \bar{\theta} \\ -\theta & 1\end{bmatrix} \quad (20)$$

$$\begin{bmatrix} S_{ij}(i,i) & S_{ij}(i,j) \\ S_{ij}(j,i) & S_{ij}(j,j)\end{bmatrix} = \lambda_y \begin{bmatrix} 1 & \bar{y} \\ y & 1\end{bmatrix} \quad (21)$$

In this second scheme, θ and y are complex numbers associated to Givens and Shear parameters, respectively. $\lambda_\theta = \frac{1}{\sqrt{1+|\theta|^2}}$ and $\lambda_y = \frac{1}{\sqrt{1-|y|^2}}$. In order to simplify the matrix notation, we introduced $\mathbf{H}_{ij}(\theta,y)$ as product of Shear and Givens rotation ($\mathbf{H}_{ij}(\theta,y)=\mathbf{S}_{ij}(y)\mathbf{G}_{ij}(\theta)$).

To our best knowledge, these parameters are estimated separately like in subsection II-A and [7], [8], [11]. Hence, the simplified method estimates both parameters simultaneously by considering particular approximations.
For that, let's consider $\mathbf{M}''_k$ a k^{th} matrix affected by a generalized Givens transformation $\mathbf{H}_{ij}(\theta,y)$.

$$\mathbf{M}''_k \leftarrow \mathbf{H}_{ij}(\theta,y)\mathbf{M}_k\mathbf{H}_{ij}(-\theta,-y) \quad k\in\{1,\cdots,K\} \quad (22)$$

Entries of $\mathbf{M}''_k$, twice affected by the elementary rotation, can be expressed as follows:

$$\begin{aligned}M''_k(i,j) = \lambda_\theta^2\lambda_y^2 \quad & \big(M_k(i,j)+[M_k(j,j)-M_k(i,i)]\,(\bar{\theta}+\bar{y}) \\ & -M_k(j,i)(\bar{\theta}^2+\bar{y}^2)-2M_k(i,j)\theta\bar{y} \\ & +[M_k(j,j)-M_k(i,i)]\left(\bar{y}^2\theta+|\theta|^2\,\bar{y}\right) \\ & +M_k(i,j)\theta^2\bar{y}^2-2M_k(j,i)\bar{\theta}\bar{y}\big)\end{aligned} \quad (23)$$

$$\begin{aligned}M''_k(j,i) = \lambda_\theta^2\lambda_y^2 \quad & \big(M_k(j,i)+[M_k(j,j)-M_k(i,i)]\,(\theta-y) \\ & -M_k(i,j)(\theta^2+y^2)+2M_k(j,i)y\bar{\theta} \\ & +[M_k(j,j)-M_k(i,i)]\left(y^2\bar{\theta}+y\,|\theta|^2\right) \\ & +M_k(j,i)\bar{\theta}^2y^2+2M_k(j,i)y\bar{\theta}\big)\end{aligned} \quad (24)$$

Assuming that we are close enough to the diagonalizing solution, then both magnitudes of θ and y can be considered very small. Hence, higher order of the latter parameters can be neglected in equations (23) and (24). Therefore, the first order approximation is as follows:

$$\begin{array}{rcl}M''_k(i,j) & \approx & M_k(i,j)+[M_k(j,j)-M_k(i,i)]\,(\bar{\theta}+\bar{y}) \\ M''_k(j,i) & \approx & M_k(j,i)+[M_k(j,j)-M_k(i,i)]\,(\theta-y)\end{array} \quad (25)$$

Next, these approximated entries will be used to derive the optimal generalized Givens parameters θ, y. The second introduced approximation is the simplified criterion where only entries twice affected by the generalized Givens rotation are considered according to [10].

$$C_s(\theta,y)=\sum_{k=1}^{K}|M''_k(i,j)|^2+|M''_k(j,i)|^2 \quad (26)$$

Let us introduce the following notations:

$$\boldsymbol{\beta}' = \begin{bmatrix} M''_1(i,j) & \cdots & M''_K(i,j)\end{bmatrix} \approx \boldsymbol{\beta}+\boldsymbol{w}^H\boldsymbol{C_{a1}} \quad (27)$$

$$\boldsymbol{\gamma}' = \begin{bmatrix} M''_1(j,i) & \cdots & M''_K(j,i)\end{bmatrix}^T \approx \boldsymbol{\gamma}+\boldsymbol{C_{a2}}\boldsymbol{w} \quad (28)$$

where $\boldsymbol{\gamma} = \begin{bmatrix} M_1(i,j) & \cdots & M_K(i,j)\end{bmatrix}^T$, $\boldsymbol{\beta} = \begin{bmatrix} M_1(i,j) & \cdots & M_K(i,j)\end{bmatrix}$, $\boldsymbol{w} = \begin{bmatrix}\theta & y\end{bmatrix}^T$,
$\boldsymbol{C_{a1}} = \begin{bmatrix} M_1(j,j)-M_1(i,i) & \cdots & M_K(j,j)-M_K(i,i) \\ M_1(j,j)-M_1(i,i) & \cdots & M_K(j,j)-M_K(i,i)\end{bmatrix}$

and $\boldsymbol{C_{a2}} = \begin{bmatrix} M_1(j,j) - M_1(i,i) & M_1(i,i) - M_1(j,j) \\ \vdots & \vdots \\ M_K(j,j) - M_K(i,i) & M_K(i,i) - M_K(j,j) \end{bmatrix}$.
Hence, the simplified criterion can be written as

$$\begin{aligned} C_s(\boldsymbol{w}) &= \boldsymbol{\gamma'}^H \boldsymbol{\gamma'} + \boldsymbol{\beta'} \boldsymbol{\beta'}^H \\ &= \boldsymbol{\gamma}^H \boldsymbol{\gamma} + \boldsymbol{\beta} \boldsymbol{\beta}^H + \boldsymbol{w}^H \boldsymbol{g_a} + \boldsymbol{g_a}^H \boldsymbol{w} + \boldsymbol{w}^H \boldsymbol{Q_a} \boldsymbol{w} \end{aligned} \tag{29}$$

where

$$\boldsymbol{g_a} = \boldsymbol{C_{a1}} \boldsymbol{\beta}^H + \boldsymbol{C_{a2}}^H \boldsymbol{\gamma} \tag{30}$$

and

$$\boldsymbol{Q_a} = \boldsymbol{C_{a1}} \boldsymbol{C_{a1}}^H + \boldsymbol{C_{a2}}^H \boldsymbol{C_{a2}} \tag{31}$$

The optimization task is realized by annulling the derivation with respect to the complex vector $\boldsymbol{w}^H$. $\frac{\partial \boldsymbol{C_s}}{\partial \boldsymbol{w}^H} = 0 \Rightarrow \boldsymbol{g_a} + \boldsymbol{Q_a} \boldsymbol{w} = 0$ Then, the optimal value of $\boldsymbol{w}$ is expressed as

$$\boldsymbol{w}_{opt} = -\boldsymbol{Q_a}^{-1} \boldsymbol{g_a} \tag{32}$$

TABLE III
SIMPLE JOINT DIAGONALIZATION ALGORITHM (SJD)

Require : $\mathbf{M}_k,\ k = 1, \cdots, K$, fixed threshold τ
and maximum sweep number M_{it}.
Initialization: $\mathbf{V} = \mathbf{I}_N$ and $\mathbf{A} = \mathbf{I}_N$.
while $\max_{i,j}(|y|, |\theta|) > \tau$ and (#sweeps $< M_{it}$)
for all $1 \leq i < j \leq N$
Compute $\boldsymbol{Q_a}$ and $\boldsymbol{g_a}$ using equations (30) and (31).
Estimate $\boldsymbol{w}_{opt} \leftarrow \boldsymbol{Q_a}^{-1} \boldsymbol{g_a}$.
Construct $\boldsymbol{H}_{ij}(\theta, y)$ using (20) and (21).
Up date different matrices as:
$\mathbf{M}_k \leftarrow \mathbf{H}_{ij}(\theta, y) \mathbf{M}_k \mathbf{H}_{ij}(-\theta, -y)$
$\mathbf{V} \leftarrow \mathbf{H}_{ij}(\theta, y) \mathbf{V}$
$\mathbf{A} \leftarrow \mathbf{A} \mathbf{H}_{ij}(-\theta, -y)$
end for
end while.

The overall proposed algorithm, named Simplified Joint Diagonalization algorithm (SJD), is summarized in Table III.

III. SIMULATION, RESULTS AND DISCUSSIONS

In this section, we have tested the proposed algorithms and compared with respect to JDTM given in [8] for different scenarios. In the first scenario, the complex matrices to be tested satisfy exactly relation (1), this scenario is referred to as exact JEVD case. The aim of this simulation is to observe each algorithms' convergence rate.
In the second scenario, the complex matrices satisfy approximatively relation (1) leading to the approximative JEVD case. The aim of this scenario is to compare algorithms' robustness in case of noise corrupted data.
Another algorithm is constructed by combining the CESJD and SJD. This algorithm, referred to as Hybrid, uses three sweeps of CESJD (to get close enough to the desired solution) and continues the sweeps by using SJD.

The Performance Index (PI) used here is the same as in [7], [12] evaluated over 100 Monte-Carlo realisations. This PI defined as

$$\begin{aligned} \mathrm{PI}(\mathbf{T}) = \ & \frac{1}{2N(N-1)} \sum_{n=1}^{N} \left(\sum_{m=1}^{N} \frac{|T(n,m)|^2}{\max_k |T(n,k)|^2} - 1 \right) \\ & + \\ & \frac{1}{2N(N-1)} \sum_{n=1}^{N} \left(\sum_{m=1}^{N} \frac{|T(m,n)|^2}{\max_k |T(k,n)|^2} - 1 \right) \end{aligned} \tag{33}$$

where $\mathbf{T} = \hat{\mathbf{V}} \mathbf{A}$ is the global matrix. The closer the PI is to zero the better is the JEVD quality.

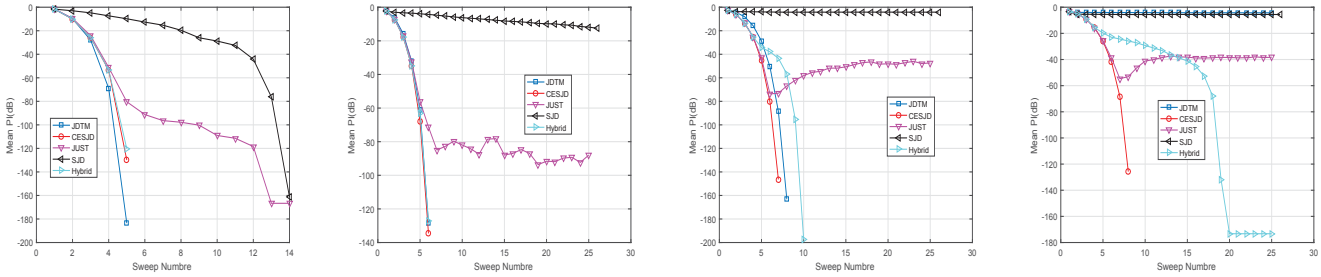
A. Exact JEVD case

In this case, all matrices $\mathbf{A}$ and $\{\mathbf{D}_k\}_{k=1,\ldots,K}$ are generated by considering complex, independent and normal distribution for all entries. Then, the complex matrices $\{\mathbf{M}_k\}_{k=1,\ldots,K}$ are computed by considering equation (1). The different algorithms are applied to these matrices to estimate the JEVD.

We have considered the JEVD of three matrices (which means that $K = 3$) and variate the matrix dimension N. N takes the following values $\{5, 50, 20, 50\}$. The obtained results are given in Figure 1. Note that as the ratio $\frac{K}{N}$ decreases, the convergence rate of different algorithms decreases. However, the considered algorithms are differently affected by decreasing $\frac{K}{N}$. CESJD and Hybrid are the less affected algorithms and present the best convergence rates especially in the difficult case ($K = 3$ and $N = 50$). The JDTM algorithm, as shown in Figures 1d and 2d, diverges completely when the ratio $\frac{K}{N}$ is less than 6% due to the approximated criterion considered by this algorithm as explained in [10]. Our proposed SJD starts to diverge when the ratio $\frac{K}{N}$ is less than 30%. This is due to the approximations introduced while developing SJD. Again JDTM, we have used two approximations, the first one is inside the considered criterion and the second one is in the generalized Givens rotations where only the first order is kept and other ones are neglected. Hence, SJD can be used in simple cases of JEVD where the ratio have to be higher than 30%. The JUST algorithm, as explained in [10], suffers from numerical instability which can be observed in Figure 1c and 1d. The other proposed algorithms CESJD and Hybrid are still converging fast especially in difficult cases (when the ratio $\frac{K}{N}$ is less than 6% in our context). CESJD has a high computation complexity but ensures faster convergence. For that, we have combined it with SJD to construct Hybrid algorithm in order to get an acceptable trade-off between convergence rate and computation complexity.

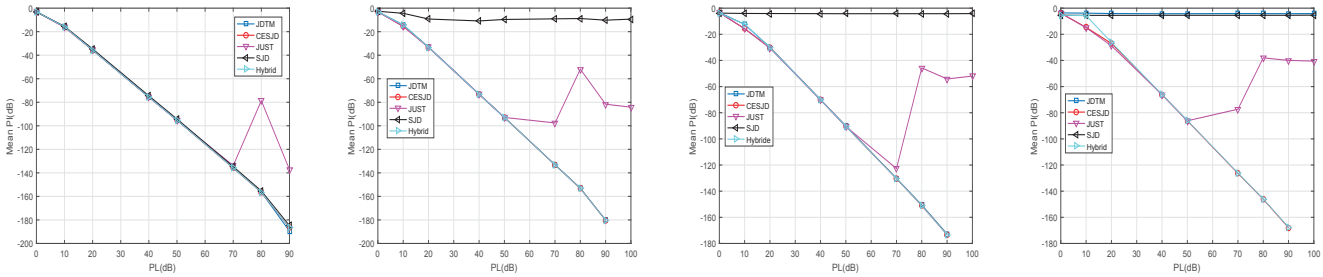
B. Approximative JEVD case

In this scenario, the complex matrices satisfy approximatively the equation given in (1) as: $\mathbf{M}_k = \mathbf{A} \mathbf{D}_k \mathbf{A}^{-1} + \mathbf{\Xi}_k$ where $\mathbf{\Xi}_k$ is a noise matrix. The level or perturbation level (PL) is measured by: $PL(dB) = \frac{\left\| \mathbf{A} \mathbf{D}_k \mathbf{A}^{-1} \right\|_F}{\left\| \mathbf{\Xi}_k \right\|_F}$. The noise matrix $\mathbf{\Xi}_k$ is generated as $\mathbf{\Xi}_k = \varrho_k \mathbf{\Upsilon}_k$ where $\mathbf{\Upsilon}_k$ is a random matrix (generated at each Monte Carlo run) and ϱ_k is a positive number allowing to fix the perturbation level, i.e. a kind of target SNR.
Algorithms' performance are evaluated according to the PL



(a) Case of $K = 3$ and $N = 5$ (b) Case of $K = 3$ and $N = 10$ (c) Case of $K = 3$ and $N = 20$ (d) Case of $K = 3$ and $N = 50$

Fig. 1. Mean PI versus sweep number in exact JEVD for different matrix dimensions



(a) Case of $K = 3$ and $N = 5$ (b) Case of $K = 3$ and $N = 10$ (c) Case of $K = 3$ and $N = 20$ (d) Case of $K = 3$ and $N = 50$

Fig. 2. Median PI versus perturbation level in approximate JEVD

for different $\frac{K}{N}$ ratios. Obtained results are given in Figure 2 where PI versus PL curves are drawn. In simple cases i.e. ratio $\frac{K}{N}$ is greater than 30%, all algorithms reach approximatively the same performance as it can be seen in Figures 2a and 2b. The convergence problem of SJD can be observed in Figures 2c and 2d and same remark is observed for JDTM algorithm in Figure 2d. On the other hand, CESJD and Hybrid are still reaching same good results especially in difficult cases where $\frac{K}{N}$ is less than 6%.

IV. Conclusion

In this paper, the JEVD problem is considered in the complex case. We have proposed two new algorithms based on generalized Givens rotations. The first one, referred to as CESJD, reaches the best results in term of convergence rate and robustness against noise but is relatively expensive. The second one, referred to as SJD, uses some justified approximations leading to a simple and low computational complexity algorithm with some convergence problem in difficult JEVD cases. The Hybrid method is constructed by combining both proposed methods in order to get an acceptable trade-off between convergence rate and computational complexity.

References

[1] L. De Lathauwer, B. De Moor, and J. Vandewalle, "Computation of the canonical decomposition by means of a simultaneous generalized Schur decomposition," *SIAM journal on Mat. Ana. and App.*, vol. 26, no. 2, 2004.

[2] X. Luciani and L. Albera, "Canonical polyadic decomposition based on joint eigenvalue decomposition," *Chemometrics and Intelligent Laboratory Systems*, vol. 132, 2014.

[3] M. Haardt and J. A Nossek, "Simultaneous Schur decomposition of several nonsymmetric matrices to achieve automatic pairing in multi-dimensional harmonic retrieval problems," *IEEE-Tr-SP*, vol. 46, no. 1, 1998.

[4] A. N. Lemma, A-J Van Der Veen, and E. F. Deprettere, "Analysis of joint angle-frequency estimation using ESPRIT," *IEEE-Tr-SP*, vol. 51, no. 5, 2003.

[5] A-J Van Der Veen, P. B. Ober, and E. F. Deprettere, "Azimuth and elevation computation in high resolution DOA estimation," *IEEE-Tr-SP*, vol. 40, no. 7, 1992.

[6] A. Belouchrani, K. Abed-Meraim, J.-F. Cardoso, and E. Moulines, "A blind source separation technique using second-order statistics," *IEEE-Tr-SP*, 1997.

[7] R. Iferroudjene, K. Abed-Meraim, and A. Belouchrani, "A new Jacobi-like method for joint diagonalization of arbitrary non-defective matrices," *Appl. Math. and Comp.*, vol. 211, no. 2, May 2009.

[8] X. Luciani and L. Albera, "Joint eigenvalue decomposition using polar matrix factorization," in *LVA/ICA*. Springer, 2010.

[9] 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, 2015.

[10] A. Mesloub, A. Belouchrani, and K. Abed-Meraim, "Efficient and stable joint eigenvalue decomposition based on generalized Givens rotations," in *EUSIPCO*, Italy, Sep. 2018.

[11] T. Fu and X. Gao, "Simultaneous diagonalization with similarity transformation for non-defective matrices," in *IEEE ICASSP Proceedings*, May 2006, vol. 4.

[12] A. Mesloub, K. Abed-Meraim, and A. Belouchrani, "A new algorithm for complex non-orthogonal joint diagonalization based on Shear and Givens rotations," *IEEE Tr-SP*, vol. 62, no. 8, April 2014.