A FAST ALGORITHM FOR JOINT EIGENVALUE DECOMPOSITION OF REAL MATRICES

Rémi André$^{(1,2)}$, *Tual Trainini*$^{(1,2)}$, *Xavier Luciani*$^{(1,2)}$, *Eric Moreau*$^{(1,2)}$

$^{(1)}$ Aix Marseille Université, CNRS, ENSAM, LSIS, UMR 7296, 13397 Marseille, France.
$^{(2)}$ Université de Toulon, CNRS, LSIS, UMR 7296, 83957 La Garde, France.

ABSTRACT

We introduce an original algorithm to perform the joint eigenvalue decomposition of a set of real matrices. The proposed algorithm is iterative but does not resort to any sweeping procedure such as classical Jacobi approaches. Instead we use a first order approximation of the inverse of the matrix of eigenvectors and at each iteration the whole matrix of eigenvectors is updated. This algorithm is called Joint eigenvalue Decomposition using Taylor Expansion and has been designed in order to decrease the overall numerical complexity of the procedure (which is a trade off between the number of iterations and the cost of each iteration) while keeping the same level of performances. Numerical comparisons with reference algorithms show that this goal is achieved.

Index Terms— Joint eigenvalue decomposition, joint diagonalization, canonical polyadic decomposition, ICA.

1. INTRODUCTION

In this paper, we consider the Joint EigenValue Decomposition (JEVD), also called joint diagonalization by similarity. The principle of JEVD is to diagonalize a set of K non-defective matrices, as the following way:

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

where the matrix of eigenvectors $\mathbf{A}$ and the K diagonal matrices $\mathbf{D}^{(k)}$ are unknown. We only consider here the real case for simplicity. Note that the matrix of eigenvectors $\mathbf{A}$ is estimated up to a permutation and scaling indeterminacy. This limitation is inherent to the JEVD problem. The uniqueness condition of the JEVD problem is given in [1]. Let the $N \times K$ matrix

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

be the matrix whose columns are the diagonals of each matrix $\mathbf{D}^{(k)}$. JEVD is unique if and only if the rows of $\mathbf{\Omega}$ are all distinct. The JEVD is useful in several contexts: direction of arrival estimation [2], joint angle-delay estimation [3], multi-dimensional harmonic retrieval [4], Canonical Polyadic Decomposition (CPD) of tensors [1], [5], [6], and Independent Component Analysis (ICA) [7], [8], [9], [10], [11]. Note that this problem is different than classical Joint Diagonalization by Congruence (JDC), for which the right matrix $\mathbf{A}^{-1}$ is replaced by $\mathbf{A}^t$ [12], [13], [14]. Of course, if $\mathbf{A}$ is orthogonal JEVD and JDC problems are equivalent.

JEVD algorithms usually resort of a Jacobi-like procedure using different matrix decompositions such as QR, LU and polar factorizations. Most algorithms are based on the polar decomposition: “sh-rt” algorithm [15] allows to diagonalize the K matrices correctly but it is not as good regarding the estimation of the matrix of eigenvectors, JUST algorithm [16] can sometimes improves these results. Before going further with this short JEVD historic, we would like to insist here that an important application of JEVD is the CPD of multiway arrays (also called PARAFAC). Indeed the CPD can be rewritten as a JEVD problem and this approach has given birth to very fast CPD algorithms such as CFS [5] and DIAG [6]. Thereby, in order to accentuate this strong point, it was important to develop fast JEVD algorithms. In this context, JDTM algorithm [6], still based on polar decomposition, provides satisfying results in most situations and requires only few iterations to converge. However it keeps a high computational cost per iteration. Conversely, JET-O algorithm [17], [18] based on the LU decomposition has low computational cost per iteration but it can require many more iterations to converge.

Therefore, we propose here a new approach able to decrease the overall numerical complexity of the process by keeping low both the cost per iteration and the average number of iterations to reach the convergence and the same performance level that JET-O and JDTM. This algorithm is called Joint eigenvalues Decomposition based on a Taylor Expansion (JDTE) and has been inspired by the algorithm H-NOODLES for JDC [19].

2. NOTATIONS

In this paper scalars are denoted by a lower case (a), vectors by a lower case boldface $(\mathbf{a})$ and matrices by an upper case boldface $(\mathbf{A})$. The $i-th$ element of the vector $\mathbf{a}$ is denoted by a_i and the $(i, j)-th$ element of a matrix $\mathbf{A}$ is denoted by A_{ij}. The identity matrix is denoted by $\mathbf{I}$. The operator $\mathsf{Diag}\{\cdot\}$ represents the diagonal matrix built from the diagonal of the

978-0-9928626-3-3/15/$31.00 ©2015 IEEE

matrix argument, the operator ZDiag$\{\cdot\}$ sets to zero the diagonal of the matrix argument and $||.||$ is Frobenius norm of the argument matrix.

3. A FAST METHOD FOR JEVD BASED ON TAYLOR EXPANSION

Here, we consider that all matrices are of size $N \times N$. To solve our problem we have to find an estimated matrix $\widehat{\mathbf{A}}$ such that, for all matrices $\mathbf{M}^{(k)}$, $\widehat{\mathbf{A}}^{-1}\mathbf{M}^{(k)}\widehat{\mathbf{A}}$ are as diagonal as possible. The matrix $\widehat{\mathbf{A}}$ is estimated iteratively by successive updates. At each iteration i a matrix $\mathbf{B}_i$ is computed in order to decrease a diagonalization criterion and the matrix set is updated as

$$\forall i = 1, ..., S, \forall k = 1, ..., K \quad \mathbf{T}_{i+1}^{(k)} = \mathbf{B}_i^{-1}\mathbf{T}_i^{(k)}\mathbf{B}_i, \quad (3)$$

with $\mathbf{T}_1^{(k)} = \mathbf{M}^{(k)}$, and S the number of iterations to reach the convergence. Thereby, if S iterations allow to achieve the JEVD, $\widehat{\mathbf{A}}$ will be equal to $\prod_{i=1}^{S}\mathbf{B}_i$ and all the $\mathbf{T}_S^{(k)}$ matrices are diagonal.

To process, a classical approach is to reduce at each iteration the following criterion based on a quadratic measure of diagonality:

$$C(\mathbf{B}_i) = \sum_{k=1}^{K} ||\mathsf{ZDiag}\{\mathbf{B}_i^{-1}\mathbf{T}_i^{(k)}\mathbf{B}_i\}||^2. \quad (4)$$

As said before, at each iteration we have to determine the N^2 elements of the matrix $\mathbf{B}_i$. Here, we simply decompose $\mathbf{B}_i$ as follow:

$$\mathbf{B}_i = (\mathbf{I} + \mathbf{Z}_i), \quad (5)$$

where $\mathbf{Z}_i$ is equal to ZDiag$\{\mathbf{B}_i\}$. Hence, at each iteration the updated data matrices become:

$$\forall k = 1, ..., K \quad \mathbf{T}_{i+1}^{(k)} = (\mathbf{I} + \mathbf{Z}_i)^{-1}\mathbf{T}_i^{(k)}(\mathbf{I} + \mathbf{Z}_i), \quad (6)$$

so that the criterion (4) depends only on $\mathbf{Z}_i$ and can be written as:

$$C(\mathbf{B}_i) = \tilde{C}(\mathbf{Z}_i) = \sum_{k=1}^{K} ||\mathsf{ZDiag}\{(\mathbf{I} + \mathbf{Z}_i)^{-1}\mathbf{T}_i^{(k)}(\mathbf{I} + \mathbf{Z}_i)\}||^2. \quad (7)$$

In the following, to simplify the notations, we drop the index i.

To solve this problem we aim to minimize the previous criterion, more precisely an approximation of this criterion. Indeed, it is compulsory to approximate our criterion if we want work on an analytic form of $\mathbf{Z}$. For this purpose, we assume that we are close to the solution and so that $||\mathbf{Z}|| \ll 1$ then we compute its first order Taylor expansion which yields:

$$\begin{aligned}(\mathbf{I} + \mathbf{Z})^{-1}\mathbf{T}^{(k)}(\mathbf{I} + \mathbf{Z}) &\approx (\mathbf{I} - \mathbf{Z})\mathbf{T}^{(k)}(\mathbf{I} + \mathbf{Z}) \\ &\approx \mathbf{T}^{(k)} - \mathbf{Z}\mathbf{T}^{(k)} + \mathbf{T}^{(k)}\mathbf{Z} - \mathbf{Z}\mathbf{T}^{(k)}\mathbf{Z} \\ &\approx \mathbf{T}^{(k)} - \mathbf{Z}\mathbf{T}^{(k)} + \mathbf{T}^{(k)}\mathbf{Z}. \quad (8)\end{aligned}$$

Moreover each matrix $\mathbf{T}^{(k)}$ can be decomposed as

$$\mathbf{T}^{(k)} = \mathbf{\Lambda}^{(k)} + \mathbf{O}^{(k)}, \quad (9)$$

where $\mathbf{\Lambda}^{(k)} = \mathsf{Diag}\{\mathbf{T}^{(k)}\}$ and $\mathbf{O}^{(k)} = \mathsf{ZDiag}\{\mathbf{T}^{(k)}\}$. Once again, considering that we are close to the solution, all the $\mathbf{T}^{(k)}$ matrices are almost diagonal and $||\mathbf{O}^{(k)}|| \ll 1$. This second approximation gives:

$$\mathbf{T}^{(k)} - \mathbf{Z}\mathbf{T}^{(k)} + \mathbf{T}^{(k)}\mathbf{Z} \approx \mathbf{\Lambda}^{(k)} + \mathbf{O}^{(k)} - \mathbf{Z}\mathbf{\Lambda}^{(k)} + \mathbf{\Lambda}^{(k)}\mathbf{Z} \quad (10)$$

and thus, we finally resort to the following approximated JEVD criterion:

$$\begin{aligned}C_a(\mathbf{Z}) = \sum_{k=1}^{K} ||\mathsf{ZDiag}\{\mathbf{O}^{(k)} - \mathbf{Z}\mathbf{\Lambda}^{(k)} + \mathbf{\Lambda}^{(k)}\mathbf{Z}\}||^2 \\ \approx \tilde{C}(\mathbf{Z}). \quad (11)\end{aligned}$$

Now, developing (11) yields:

$$\begin{aligned}C_a(\mathbf{Z}) &= \sum_{k=1}^{K} \sum_{\substack{m,n=1 \\ m \neq n}}^{N} (O_{mn}^{(k)} + Z_{mn}\Lambda_{mm}^{(k)} - Z_{mn}\Lambda_{nn}^{(k)})^2 \\ &= \sum_{\substack{m,n=1 \\ m \neq n}}^{N} f(Z_{mn}), \quad (12)\end{aligned}$$

where

$$f(Z_{mn}) = \sum_{k=1}^{K} (O_{mn}^{(k)} + (\Lambda_{mm}^{k} - \Lambda_{nn}^{k})Z_{mn})^2. \quad (13)$$

Since for all $m, n, f(Z_{mn})$ is non-negative, minimizing $C_a(\mathbf{Z})$ is equivalent to minimize all the $f(Z_{mn})$ independently. In order to find the elements of the updating matrix $\mathbf{Z}$, we look for the values of Z_{mn} which annul the derivative:

$$\forall m, n \quad m \neq n,$$

$$\frac{\partial f(Z_{mn})}{\partial Z_{mn}} = 2\sum_{k=1}^{K} (O_{mn}^{(k)} + (\Lambda_{mm}^{(k)} - \Lambda_{nn}^{(k)})Z_{mn})(\Lambda_{mm}^{(k)} - \Lambda_{nn}^{(k)}). \quad (14)$$

Finally, we deduce the analytic expression of Z_{mn}:

$$\forall m, n \quad m \neq n, Z_{mn} = -\frac{\sum_{k=1}^{K} O_{mn}^{(k)}(\Lambda_{mm}^{(k)} - \Lambda_{nn}^{(k)})}{\sum_{k=1}^{K} (\Lambda_{mm}^{(k)} - \Lambda_{nn}^{(k)})^2}. \quad (15)$$

Algorithm 1 The JDTE algorithm

```
Define a stopping criterion $crit$ and a maximal number of iterations $S_{max}$;
Initialize $\widehat{\mathbf{A}}$;
$i = 1$
while $crit$ is $false$ and $i \leq S_{max}$ do
  for $m = 1$ to $N$ do
    for $n = 1$ to $N$ do
      if $m \neq n$ then
        Compute $Z_{mn}$ using (15);
      end if
    end for
  end for
  Compute $\mathbf{B} = \mathbf{I} + \mathbf{Z}$ ;
  Compute $\mathbf{B}^{-1}$;
  for $k = 1$ to $K$ do
    $\mathbf{T}^{(k)} \leftarrow \mathbf{B}^{-1}\mathbf{T}^{(k)}\mathbf{B}$;
  end for
  $\widehat{\mathbf{A}} \leftarrow \widehat{\mathbf{A}}\mathbf{B}$;
  $i = i + 1$;
  update $crit$
end while
```

Algorithm 1 gives an algorithmic description of the method. Note that here, one iteration consists in only one update of matrices $\widehat{\mathbf{A}}$ and $\mathbf{T}^{(k)}$. Several stopping criteria can be used, for our simulations, algorithm are stopped as soon as the following condition is fulfilled:

$$\frac{|C(\mathbf{B}_i) - C(\mathbf{B}_{i-1})|}{C(\mathbf{B}_{i-1})} \leq \varepsilon. \tag{16}$$

4. NUMERICAL COMPLEXITIES OF JEVD ALGORITHMS

We define here the numerical complexity (denoted Γ) of an iterative algorithm as the minimal number of multiplications computed by this algorithm during each iteration. According to this definition the numerical complexity of the JDTE algorithm is given by:

$$\Gamma\{\text{JDTE}\} \;\simeq\; (N(N-1)(2K+1) + 2N^3 + 2KN^3), \tag{17}$$

this value has to be compared to the numerical complexities of reference JEVD algorithms. Hence we have:

$$\begin{aligned}
\Gamma\{\text{JET-O}\} &\simeq KN(N-1)(N+7)/6 + N^2(N+1)/2 \\
&\quad + (5K + 4KN + N/3)(N-1)N/2, && (18)\\
\Gamma\{\text{sh-rt}\} &\simeq (3K + 16KN + 14N)(N-1)N/2, && (19)\\
\Gamma\{\text{JDTM}\} &\simeq (6K + 16KN + 8N)(N-1)N/2, && (20)\\
\Gamma\{\text{JUST}\} &\simeq (32 - 6K + 22KN + 8N)(N-1)N/2. && (21)
\end{aligned}$$

Costs per iteration are thus dominated by a term in αKN^3, whatever the algorithm, and only the value of α depends on the considered algorithm. It then clearly appears that only JET-O can compete with JDTE in terms of cost per iteration. However the average number of required iterations to reach the stopping criterion varies from one algorithm to an other. It's why, we define the total numerical complexity as $\Gamma_{tot} = S\Gamma$. Therefore total numerical complexities of the different algorithms have to be compared empirically by means of numerical simulations.

5. NUMERICAL SIMULATIONS

Matrix sets to be diagonalized are created as follow: for given values of K and N we randomly draw (according to a standard normal distribution) a set of K diagonal matrices and a squared matrix $\mathbf{A}$ of size N. Then we compute the K matrices $\mathbf{M}^{(k)}$ as:

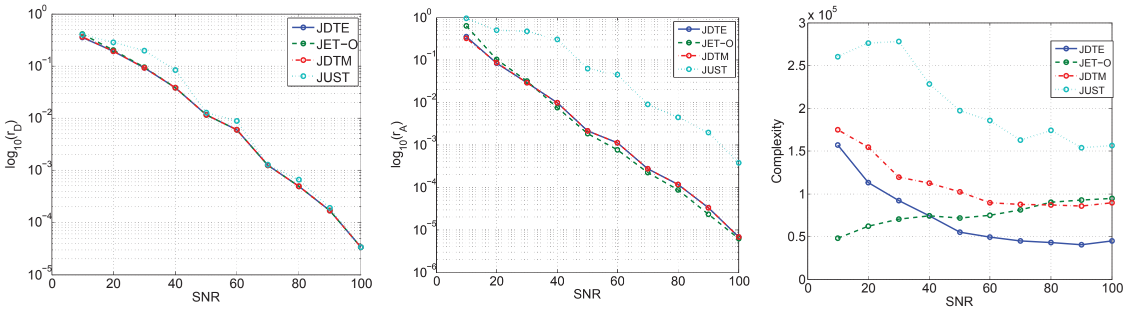
$$\forall k = 1, ..., K \quad \mathbf{M}^{(k)} = \frac{\mathbf{A}\mathbf{D}^{(k)}\mathbf{A}^{-1}}{\|\mathbf{A}\mathbf{D}^{(k)}\mathbf{A}^{-1}\|} + \sigma\frac{\mathbf{E}^{(k)}}{\|\mathbf{E}^{(k)}\|}. \tag{22}$$

Where $\mathbf{E}^k$ models a Gaussian random noise. Parameter σ allows to regulate the noise power and obtain the desired Signal to Noise Ratio (SNR) value which is then defined as : $-20\log_{10}(\sigma)$.

In order to validate the proposed approach, we have compared the performance of JDTE algorithm with those of JUST, JDTM and JET-O according two different scenarios. For each simulation *i.e.* each time we vary a simulation parameter, 100 Monte-Carlo runs are performed. For each MC run a new random matrix set is built and the three algorithms are applied to this new set. The same stopping criterion (16) is used for the four algorithms wiht $\varepsilon = 10^{-6}$. All the algorithms are initialized with the identity matrix. At the end of each run two indicators are computed for each algorithm along with the numerical complexities. Our first indicator quantifies the relative deviation between the eigenvalues estimated by a JEVD algorithm and the actual eigenvalues of matrices $\mathbf{M}^{(k)}$. In the same way, our second indicator quantifies the relative deviation between the estimated matrix of eigenvectors and the actual one. These indicators are denoted r_D and r_A respectively and are computed as follow, after removing scaling and permutation indeterminacy:

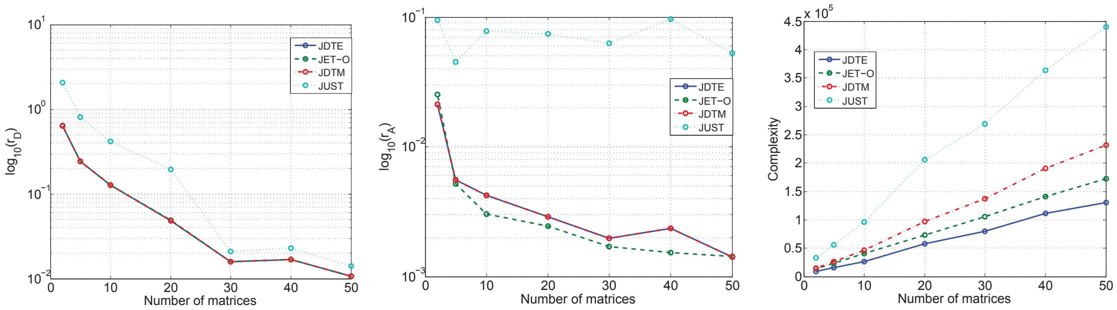
$$\begin{aligned}
r_D &= \frac{1}{K}\sum_{k=1}^{K}\frac{\|\mathbf{d}^{(k)} - \widehat{\mathbf{d}}^{(k)}\|}{\|\mathbf{d}^{(k)}\|}, && (23)\\
r_A &= \frac{\|\mathbf{A} - \widehat{\mathbf{A}}\|}{\|\mathbf{A}\|}, && (24)
\end{aligned}$$

where $\mathbf{d}^{(k)}$ and $\widehat{\mathbf{d}}^{(k)}$ are vectors containing actual and estimated eigenvalues of $\mathbf{M}^{(k)}$ respectively. Finally we resort to three criteria in order to assess algorithm performances: the



(a) *Diagonalization criterion (median value).* (b) *Estimation of the matrix of eigenvectors (median value).* (c) *Numerical complexity (average value).*

Fig. 1. Evolution of the three comparison criteria as a function of the SNR value (in dB) for the JDTM (dash-dot line), JET-O(dash line), JUST(dot line) algorithms and the proposed JDTE (solid line) method for 20 matrices of size 5×5.



(a) *Diagonalization criterion (median value).* (b) *Estimation of the matrix of eigenvectors (median value).* (c) *Numerical complexity (average value).*

Fig. 2. Evolution of the three comparison criteria as a function of the size of the number of matrices to be jointly diagonalized for the JDTM (dash-dot line), JET-O(dash line), JUST(dot line) algorithms and the proposed JDTE (solid line) method for matrices of size 5×5 and a SNR of 50 dB.

median values of r_D and r_A and the average value of Γ_{tot} over the 100 MC runs.

For our first scenario, we consider sets of 20 matrices of size 5 and we vary the SNR value from 10 to 100. Evolutions of the three comparison criteria as a function of the SNR are reported on figures 1(a), 1(b) and 1(c). It can be seen that the JDTE, JET-O and JDTM algorithms provide very similar performances in terms of estimation precision of the eigenvalues and of the matrix of eigenvectors while JUST is less efficient. As expected, the numerical complexity criterion allows to discriminate between these three algorithms. We observe that for the lowest SNR values JET-O is the less costly algorithm respectively followed by JDTE and JDTM. Note that if we consider that the median values of r_A must be lower than 10^{-1} no algorithm gives satisfying results under 30 dB. Then, above 40 dB, JDTE becomes clearly the less costly solution and this tendency increases with the SNR value.

The second scenario is performed in the same way but this time, we vary the size of the matrix sets from 2 to 50 while the SNR and the matrix size are set to 50 dB and 5 respectively. Results are plotted on figures 2(a), 2(b) and 2(c). Once again the diagonalization criterion does not allow to decide between JDTE, JDTM and JET-O and although JET-O algorithm provides now the best estimation of the matrix of eigenvectors, performances remains very close. Conversely, looking at the Γ_{tot} criterion, it appears clearly that if we are looking for rapidity JDTE is a recommended choice. Indeed it is consistently less costly than its competitors and it is noteworthy that the gap increases linearly with the size of the matrix set. This makes JDTE a suitable solution for the CP decomposition since in most CPD application the size of the matrix set can be very large. More precisely, the ratios $\Gamma_{tot}\{\text{JDTM}\}/\Gamma_{tot}\{\text{JDTE}\}$, $\Gamma_{tot}\{\text{JET-O}\}/\Gamma_{tot}\{\text{JDTE}\}$ and $\Gamma_{tot}\{\text{JUST}\}/\Gamma_{tot}\{\text{JDTE}\}$ remains stable around 1.7, 1.4 and 3.4 respectively. In conclusion, numerical simulations, we can show that the proposed method is the least costly while keeping the same performances as JDTM and JET-O.

6. CONCLUSION

We have introduced here a new algorithm to compute the JEVD of a set of real matrices. Contrary to classical approaches our algorithm does not resort to any sweeping procedure so that at each iteration all the element of the matrix of eigenvectors are directly computed. This approach is achieved by using a Taylor expansion of the diagonalization criterion and allows significantly to reduce the cost per iteration. Thus, we have shown that the numerical complexity of one JDTE iteration is much lower than the one of most other JEVD algorithms. Numerical simulations confirm that JDTE clearly appears as the quickest algorithm in many practical situations, notably when the signal to noise ratio is not too low. Moreover, it offers similar performances than reference JEVD algorithms in term of estimation precision of eigenvectors and eigenvalues. This makes JDTE a good candidate to perform the JEVD step included in recent fast CPD algorithms.

REFERENCES

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

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

[8] A. Belouchrani, K. Abed-Meraim, J. F. Cardoso, and E. Moulines, "A blind source separation technique using second-order statistics," *IEEE Transactions on Signal Processing*, vol. 45, no. 2, pp. 434–444, 1997.

[9] A. Ferreol, L. Albera, and P. Chevalier, "Fourth order blind identification of underdetermined mixtures of sources (FOBIUM)," *IEEE Transactions on Signal Processing*, vol. 53, pp. 1254–1271, April 2005.

[10] L. Albera, A. Ferreol, P. Chevalier, and P. Comon, "ICAR, a tool for blind source separation using fourth order statistics only," *IEEE Transactions on Signal Processing*, vol. 53, pp. 3633–3643, October 2005.

[11] L. De Lathauwer, J. Castaing, and J. F. Cardoso, "Fourth-order cumulant-based blind identification of underdetermined mixtures," *IEEE Transactions on Signal Processing*, vol. 55, no. 6, pp. 2965–2973, 2007.

[12] 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.

[13] 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, July 2002.

[14] G. Chabriel, M. Kleinsteuber, E. Moreau, H. Shen, P. Tichavsky, and A. Yeredor, "Joint Matrices Decompositions and Blind Source Separation. A Survey of Methods, Identification and Applications," *IEEE Signal Processing Magazine*, vol. 31, no. 3, pp. 34–43, May 2014.

[15] 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.

[16] 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.

[17] X. Luciani and L. Albera, "Semi-algebraic canonical decomposition of multi-way arrays and joint eigenvalue decomposition," in *ICASSP 11, 2011 IEEE International Conference on Acoustics Speech and Signal Processing*, Prague, Czech Republic, May 22-27 2011, pp. 4104–4107.

[18] X. Luciani and L. Albera, "Joint eigenvalue decomposition of non-defective matrices based on the lu factorization with application to ica," *to appear in IEEE Transactions on Signal Processing (already available on IEEE Xplore)*.

[19] 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, 2014.