FAST JACOBI ALGORITHM FOR NON-ORTHOGONAL JOINT DIAGONALIZATION OF NON-SYMMETRIC THIRD-ORDER TENSORS

Victor MAURANDI [1,2] *and 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.
Email : {maurandi, moreau}@univ-tln.fr

ABSTRACT

We consider the problem of non-orthogonal joint diagonalization of a set of non-symmetric real-valued third-order tensors. This appears in many signal processing problems and it is instrumental in source separation. We propose a new Jacobi-like algorithm based on an LU decomposition of the so-called diagonalizing matrices. The parameters estimation is done entirely analytically following a strategy based on a classical inverse criterion and a fully decoupled estimation. One important point is that the diagonalization is directly done on the set of third-order tensors and not on their unfolded version. Computer simulations illustrate the overall good performances of the proposed algorithm.

***Index Terms*—** Blind Source Separation, Independent Component Analysis, Joint Diagonalization, Third-Order Tensors.

1. INTRODUCTION

Joint diagonalization of sets of matrices or tensors is an important issue in source separation and independent component analysis (ICA) [1] [2]. It takes its origin in the two important papers [3] and [4] where the orthogonal joint diagonalization of matrices and the orthogonal diagonalization of a fourth-order tensor are introduced respectively in link with ICA. A generalization to statistics of any order greater than two is done in [5] and the specific case of the orthogonal joint diagonalization of third-order tensors is considered in [6].

Nowadays, attention has focused on the non-orthogonal joint diagonalization of matrices, see e.g. [7], [8] and references there in. A non-orthogonal diagonalization is important mainly because it allows to skip a first processing step (called whitening in source separation) that limits the performances in practice. Numerous such algorithms have been devised. Among them, the Jacobi-like ones have the main advantage to be simple to implement, especially when one is able to derive an analytical solution. These procedures allow to tackle large dimension matrices and offer very good performances, see e.g. [9] – [14]. Moreover these methods often allow a potential computational parallelism. Concerning tensors, Canonical Polyadic Decomposition (CPD) [15]- [17] is a well-known multi-way decomposition that can be interpreted as the diagonalization of a single tensor. The non-orthogonal joint diagonalization of tensors is an open problem as such. To our best knowledge, even if a link exists with CPD, it seems that no works has directly been devoted to.

In this paper, we propose a Jacobi-like algorithm for the non-orthogonal joint diagonalization of non-symmetric real-valued third-order tensors. To our best knowledge, this is one of the first algorithm for non-unitary simultaneous tensor diagonalization. It is based on an LU parameterization of the so-called diagonalizing matrices. Contrary to the diagonalization of a third-order tensor which is handled in the literature as a problem of joint diagonalization of matrices unfolding the target tensor, in this paper we directly work on the tensors in order to joint diagonalize them, thus avoiding potential preprocessing step. We present two approaches of the algorithm based on a fully decoupled parameters estimation that allows an entirely analytical resolution for each approach. Numerical simulations illustrate the algorithm performances for both approaches and provide a comparison with the CPD algorithm based on the Alternate Least Squares (ALS) developed in the Matlab toolbox [18].

2. PROBLEM FORMULATION

We consider a set of third-order cubic $N \times N \times N$, $N \in \mathbb{N}^* \setminus \{1\}$, real-valued tensors $\mathbf{T}(k) = \big(T_{i_1 i_2 i_3}(k)\big)$, $k = 1, \ldots, K$, $K \in \mathbb{N}^* \setminus \{1\}$ and $(i_1, i_2, i_3) \in \{1, \ldots, N\}^3$. The tensors $\mathbf{T}(k)$ are assumed to ideally follow the following component-wise decomposition

$$T_{i_1 i_2 i_3}(k) = \sum_{j_1, j_2, j_3 = 1}^{N} D_{j_1 j_2 j_3}(k) A_{1,i_1 j_1} A_{2,i_2 j_2} A_{3,i_3 j_3} \quad (1)$$

where $\mathbf{A}_x = \big(A_{x,ij}\big)$ for $x \in \{1, 2, 3\}$ are three $N \times N$ invertible matrices, called factor matrices, and $\mathbf{D}(k) = \big(D_{j_1 j_2 j_3}(k)\big)$ are K diagonal tensors of dimension $N \times N \times$

The authors thank the *Direction Générale de l'Armement* for its financial support.

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

N, *i.e.* $D_{j_1j_2j_3}(k) = D_{j_1j_2j_3}(k)\delta_{j_1j_2j_3}$ where $(j_1, j_2, j_3) \in \{1, \ldots, N\}^3$ and $\delta_{j_1j_2j_3} = 1$ if $j_1 = j_2 = j_3$ and 0 otherwise. In the following, the relation (1) will be denoted for all tensor components as

$$\mathbf{T}(k) = \mathbf{D}(k) \times_1 \mathbf{A}_1 \times_2 \mathbf{A}_2 \times_3 \mathbf{A}_3. \quad (2)$$

In practice, these tensors are typically obtained by some sample statistics through a finite number of data. The goal is, from only data tensors $\mathbf{T}(k)$, $k = 1, \ldots, K$, to identify the inverse matrices, denoted by $\mathbf{B}_x$, of matrices $\mathbf{A}_x$, for all $x \in \{1, 2, 3\}$. $\mathbf{B}_x$ are the so-called diagonalizing matrices. In order to do this, we introduce the transformed tensors defined as

$$\mathbf{R}(k) = \mathbf{T}(k) \times_1 \mathbf{B}_1 \times_2 \mathbf{B}_2 \times_3 \mathbf{B}_3. \quad (3)$$

If each $\mathbf{B}_x$ is equal to the inverse of the respective $\mathbf{A}_x$ (up to scaling factors and permutations) then it is easily seen that $\mathbf{R}(k)$ are diagonal tensors. Thus the purpose is to find the matrices $\mathbf{B}_x$ such that $\mathbf{R}(k)$ all are diagonal tensors.

In practice, this joint diagonalization of third-order tensors is done only in an approximate way because the set $\mathbf{T}(k)$ is estimated from statistics or other analysis operators. Classically, the approximation is measured using a criterion. In this paper, we propose to focus on a so-called inverse criterion defined as

$$\mathcal{J}(\mathbf{B}_x) = \sum_{k=1}^{K} \|\mathsf{ZTdiag}\{\mathbf{R}(k)\}\|^2 \quad (4)$$

where $\|\cdot\|$ is the Frobenius norm and $\mathsf{ZTdiag}\{\cdot\}$ is the zero diagonal tensor (with all zeros on the diagonal) built from the tensor argument. Once the criterion is fixed, the problem becomes an optimization one. Here one has to search for matrices $\mathbf{B}_x$ minimizing $\mathcal{J}$.

Remark: Because $\mathbf{D}(k)$ are diagonal tensors for all k, there is a link between the decomposition in (1) and the CPD of a fourth order tensors. Indeed, we have

$$T_{i_1i_2i_3}(k) = \sum_{j=1}^{N} D_{jjj}(k) A_{1,i_1j} A_{2,i_2j} A_{3,i_3j} \quad (5)$$

that is readily written as

$$T'_{i_1i_2i_3k} = \sum_{j=1}^{N} A_{4,kj} A_{1,i_1j} A_{2,i_2j} A_{3,i_3j} \quad (6)$$

where $A_{4,kj} = D_{jjj}(k)$ and $T'_{i_1i_2i_3k} = T_{i_1i_2i_3}(k)$ for all index values. This last writing is directly the CPD of the fourth-order tensor $\mathbf{T}' = \left(T'_{i_1i_2i_3k}\right)$.

3. PROPOSED ALGORITHM

3.1. Parameterization

An obvious solution minimizing the criterion $\mathcal{J}$ in (4) is $\mathbf{B}_x = \mathbf{0}$ for all $x \in \{1, 2, 3\}$ and where $\mathbf{0}$ is the zero matrix. Of course this solution is a degenerate one. Moreover we assumed that all diagonalizing matrices are invertible. That is why, it is important to consider a specific constraint on each matrix $\mathbf{B}_x$ allowing to find invertible solutions. For this, we constrain each matrix $\mathbf{B}_x$ to have a unit determinant. A way to do that consists of using an LU parameterization of each matrix $\mathbf{B}_x$. It is well-known that all square matrices can be decomposed as $\mathbf{DPLU}$, where $\mathbf{D}$ is a diagonal matrix, $\mathbf{P}$ is a permutation matrix and $\mathbf{L}$ and $\mathbf{U}$ are, respectively lower and upper triangular matrices with diagonal components all equal to 1. Thus, let us notice that $\det(\mathbf{L}) = 1$ and $\det(\mathbf{U}) = 1$ in such a way that $\mathbf{LU}$ has also a unit determinant. Since we are looking for matrices $\mathbf{B}_x$ up to scaling factors and permutations then $\mathbf{D}$ and $\mathbf{P}$ are useless. Then it remains to estimate the diagonalizing matrices all as $\mathbf{B}_x = \mathbf{L}_x\mathbf{U}_x$ for all $x \in \{1, 2, 3\}$. Following [9] and [14], we here propose to estimate matrices $\mathbf{L}_x$ and $\mathbf{U}_x$ in an alternate way, *i.e.* by deriving $\mathbf{U}_x$ while keeping fixed $\mathbf{L}_x$ and vice versa.

3.2. Jacobi procedure

The Jacobi procedure consists of decomposing each $\mathbf{U}_x$ and $\mathbf{L}_x$ matrices by a product of $\frac{N(N-1)}{2}$ elementary matrices as

$$\mathbf{U}_x = \prod_{i=1}^{N-1} \prod_{j=i+1}^{N} \mathbf{U}_x^{ij} \quad \text{and} \quad \mathbf{L}_x = \prod_{i=1}^{N-1} \prod_{j=i+1}^{N} \mathbf{L}_x^{ij}. \quad (7)$$

The elementary matrices $\mathbf{U}_x^{ij}$ and $\mathbf{L}_x^{ij}$ each correspond to an $N \times N$ identity matrix but by replacing the (i, j), $i < j$ component by u_x^{ij} for $\mathbf{U}_x^{ij}$ and by replacing the (j, i), $i < j$ component by ℓ_x^{ij} for $\mathbf{L}_x^{ij}$. We can notice that each elementary matrices $\mathbf{U}_x^{ij}$ (respectively $\mathbf{L}_x^{ij}$) depends only on one parameter u_x^{ij} (respectively ℓ_x^{ij}). Finally, for all $x \in \{1, 2, 3\}$, $\mathbf{B}_x$ is assumed to follow the decomposition

$$\mathbf{B}_x = \prod_{i=1}^{N-1} \prod_{j=i+1}^{N} \mathbf{L}_x^{ij} \prod_{i=1}^{N-1} \prod_{j=i+1}^{N} \mathbf{U}_x^{ij}. \quad (8)$$

3.3. Proposed algorithm

One of our main goal is to find a direct analytical minimizing solution for each parameter. For that, we consider the estimation of u_x^{ij} and we directly use the criterion $\mathcal{J}$ in (4) on $\mathbf{U}_x^{ij}$. This reads

$$\mathcal{J}(\mathbf{U}_x^{ij}) = \sum_{k=1}^{K} \left\|\mathsf{ZTdiag}\left\{\mathbf{R}^{ij}(k)\right\}\right\|^2 \quad (9)$$

where

$$\mathbf{R}^{ij}(k) = \mathbf{R}(k) \times_1 \mathbf{U}_1^{ij} \times_2 \mathbf{U}_2^{ij} \times_3 \mathbf{U}_3^{ij}. \quad (10)$$

Each matrix $\mathbf{U}_x^{ij}$ has a linear impact on a slice of $\mathbf{R}(k)$, $\mathbf{U}_1^{ij}$ on the slice defined by $\{(i, p, q) \,|\, (p, q) \in \{1, \ldots, N\}^2\}$, $\mathbf{U}_2^{ij}$

on the slice $\{(p,i,q)\,|\,(p,q) \in \{1,\ldots,N\}^2\}$ and $\mathbf{U}_3^{ij}$ on $\{(p,q,i)\,|\,(p,q) \in \{1,\ldots,N\}^2\}$. Thus, in (10), the components of $\mathbf{R}(k)$ located at the intersection of at least two of these slices are transformed by at least two matrices $\mathbf{U}_x^{ij}$. Thus, at these locations, the components of $\mathbf{R}^{ij}(k)$ are non-linear equations in at least two u_x^{ij}. If we directly minimize (9) for estimating all $\mathbf{U}_x^{ij}$, as in [9] for the matrix case, these non-linear terms do not allow an analytical solution.

Nevertheless, we can overcome this issue by estimating each $\mathbf{U}_x^{ij}$ in a decoupled way, *i.e.* estimating $\mathbf{U}_1^{ij}$ for fixed $\mathbf{U}_2^{ij}$ and $\mathbf{U}_3^{ij}$, then updating $\mathbf{R}(k)$ and $\mathbf{B}_1$ and then doing the same for the two other matrices recursively. This proposed algorithm is called T-ALUJA for Tensorial Alternate LU Jacobi Algorithm.

4. PARAMETERS DERIVATION

For a given (i,j) with $i<j$, we now derive the parameters minimizing $\mathcal{J}$ in (9). We only consider $\mathbf{U}_x^{ij}$ since for $\mathbf{L}_x^{ij}$, all follows the same lines.

In this section, we propose two different approaches. The first one is a classical one by using directly the criterion (9) for the parameters estimation. It just consists of taking into account all the off-diagonal components impacted by the transformation (10). The second approach is similar but adapted to the case where we are close to a diagonalizing solution. In the following, we only develop the derivations for $\mathbf{U}_1^{ij}$. For the other $\mathbf{U}_2^{ij}$ and $\mathbf{U}_3^{ij}$, all may be simply deduced by indexes permutations.

4.1. Classical approach

Let us derive the optimal solution for $\mathbf{U}_1^{ij}$ setting $\mathbf{U}_2^{ij}$ and $\mathbf{U}_3^{ij}$ to the identity. Hence (10) becomes

$$\mathbf{R}_1^{ij}(k) = \mathbf{R}(k) \times_1 \mathbf{U}_1^{ij}. \tag{11}$$

Let the sets $\mathcal{N} = \{1,\ldots,N\}$ and $\mathcal{P} = \mathcal{N} \setminus \{i\}$, the components of $\mathbf{R}_1^{ij}(k)$ can be read

$$\begin{aligned} R_{1,imn}^{ij}(k) &= R_{imn}(k) + R_{jmn}(k)u_1^{ij} \\ R_{1,pmn}^{ij}(k) &= R_{pmn}(k) \end{aligned} \tag{12}$$

with $(m,n) \in \mathcal{N}^2$ and $p \in \mathcal{P}$.
Let us define the sets $\mathcal{Q} = \{\big((i,b,c)\,|\,(b,c) \in \mathcal{N}^2\big) \setminus (i,i,i)\}$ and $\mathcal{S} = \{\big((a,b,c)\,|\,a \in \mathcal{P}, (b,c) \in \mathcal{N}^2\big) \setminus \{(p,p,p)\,|\,p \in \mathcal{P}\}\}$, we can also rewrite $\mathcal{J}$ in (9) as

$$\mathcal{J}(u_1^{ij}) = \sum_{k=1}^{K} \{ \sum_{(i,m,n)\in\mathcal{Q}} R_{1,imn}^{ij\,2}(k) + \sum_{(a,b,c)\in\mathcal{S}} R_{1,abc}^{ij\,2}(k) \}. \tag{13}$$

Now using (12) in (13) leads to

$$\begin{aligned} \mathcal{J}(u_1^{ij}) = \sum_{k=1}^{K} \{ &\sum_{(i,m,n)\in\mathcal{Q}} \big(R_{imn}(k) + R_{jmn}(k)u_1^{ij}\big)^2 \\ &+ \sum_{(a,b,c)\in\mathcal{S}} R_{abc}^2(k) \}. \end{aligned} \tag{14}$$

We now have to solve $\partial\mathcal{J}/\partial u_1^{ij} = 0$. After straightforward derivations, we obtain the following analytical optimal solution

$$u_1^{ij} = -\frac{\sum_{k=1}^{K} \sum_{(i,m,n)\in\mathcal{Q}} R_{imn}(k)R_{jmn}(k)}{\sum_{k=1}^{K} \sum_{(i,m,n)\in\mathcal{Q}} R_{jmn}^2(k)}. \tag{15}$$

4.2. Adapted approach

A close look at the above derivations shows that some transformed tensor components are more weighted than the others. Suppose that we are close to a diagonalizing solution in the sense that for all $k \in \{1,\ldots,K\}$, all $\mathbf{R}(k)$ off-diagonal components have a very small magnitude in comparison to the $\mathbf{R}(k)$ diagonal component with the smallest one, *i.e.* for all $(a,b,c) \in \{\mathcal{N}^3 \setminus \{(p,p,p)|p \in \mathcal{N}\}\}$, $|R_{abc}(k)| \ll 1$. It is then rather easy to show that $|u_x^{ij}| \ll 1$ for all $x \in \{1,2,3\}$. Thus in (12), in the first equation, for $(m,n) \in \{\mathcal{N}^2 \setminus (j,j)\}$, $|R_{jmn}(k)u_1^{ij}| \ll |R_{imn}(k)|$, so this equation can be approximated by

$$R_{1,imn}^{ij}(k) \approx R_{imn}(k). \tag{16}$$

Hence, in $\mathbf{R}_1^{ij}(k)$ after the transformation (11), it approximately remains only one non-constant off-diagonal component that is

$$R_{1,ijj}^{ij}(k) = R_{ijj}(k) + R_{jjj}(k)u_1^{ij}. \tag{17}$$

Finally, using (16) and (17) in (13), after straightforward derivation, (15) can be approximated by

$$u_1^{ij} \approx -\frac{\sum_{k=1}^{K} R_{ijj}(k)R_{jjj}(k)}{\sum_{k=1}^{K} R_{jjj}^2(k)}. \tag{18}$$

Comparing with the classical approach, the complexity of each parameter estimation is reduced. Moreover the estimations of u_1^{ij}, u_2^{ij} and u_3^{ij} are now independent. Indeed the $\mathbf{R}(k)$ update using this approximate solution have no influence on the estimation of u_2^{ij} and u_3^{ij}. Hence the estimation of these three parameters can be done all together in a same loop. This is a clear supplementary advantage.

5. COMPUTER SIMULATIONS

We now illustrate the performances of the proposed T-ALUJA algorithm by considering the two versions. The one using the classical approach is simply denoted by T-ALUJA and the one using the adapted approach is denoted by T-ALUJA-A. We also compare them with CPD-ALS algorithm given in [18].

In order to evaluate the algorithm performances, we use the performance index proposed in [5] [19] [20]. It compares each matrix $\mathbf{S}_x = \mathbf{B}_x \mathbf{A}_x = (S_{x,ij})$ to the product of a permutation matrix and a diagonal matrix. It is defined by $I(\mathbf{S}_x) = \dfrac{1}{2N(N-1)} I'(\mathbf{S}_x)$ where

$$I'(\mathbf{S}_x) = \sum_{i=1}^{N}\left(\sum_{j=1}^{N}\frac{|S_{x,ij}|^2}{S_{x,r}^2}-1\right)+\sum_{j=1}^{N}\left(\sum_{i=1}^{N}\frac{|S_{x,ij}|^2}{S_{x,c}^2}-1\right) \tag{19}$$

with $S_{x,r}^2 = \max_m |S_{x,im}|^2$ and $S_{x,c}^2 = \max_m |S_{x,mj}|^2$. This normalized index is zero if $\mathbf{S}_x$ satisfies $\mathbf{B}_x = \mathbf{DPA}_x^{-1}$. For the CPD-ALS algorithm, we first identify the three estimated matrices $\widetilde{\mathbf{A}}_x$ corresponding to each searched matrices $\mathbf{A}_x$ and we directly consider $\mathbf{S}_x = \widetilde{\mathbf{A}}_x^{-1}\mathbf{A}_x$. Finally we take the mean value of $I(\mathbf{S}_x)$ for $x \in \{1, 2, 3\}$.

We consider 10 real-valued tensors of size $10 \times 10 \times 10$ defined as $\mathbf{T}(k) + t\,\mathbf{N}(k)$ where $\mathbf{N}(k)$ are noise tensors and t represents the noise level. All components of matrices $\mathbf{A}_x$ and diagonal tensors $\mathbf{D}(k)$ follow a uniform law on $[-1, 1]$ whereas the components of $\mathbf{N}(k)$ follow a zero mean unit variance normal distribution. Each $\mathbf{B}_x$ is initialized randomly by using a zero mean unit variance normal distribution.

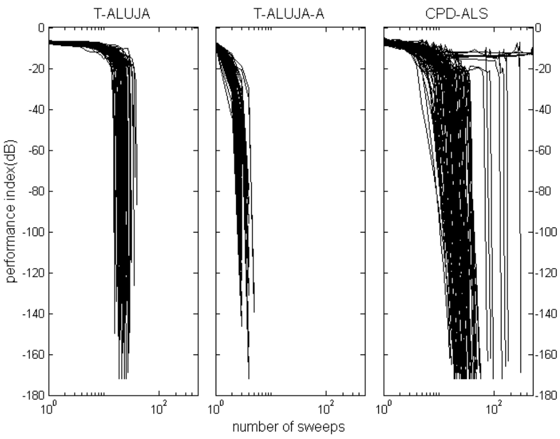

Fig. 1. Performance index versus the number of sweeps for 10 tensors of size $10 \times 10 \times 10$ in a noiseless context (superposition of 100 independent draws).

The Fig. 1 displays the index value *w.r.t.* the sweeps of the two proposed Jacobi-like algorithms and CPD-ALS one for 100 independent draws for $t = 0$. Even if the initialization of each $\mathbf{B}_x$ is done *a priori* far from any diagonalizing solution, T-ALUJA-A presents a much better convergence speed than T-ALUJA and CPD-ALS. T-ALUJA-A presents also a standard deviation that is also very good within the 100 draws by always converging between the fourth and the sixth sweep while for T-ALUJA it is between the seventeenth and the fortieth sweep whereas CPD-ALS diverges for four draws and gets a real large standard deviation for the converging ones (from the seventeenth sweep to the three hundred and sixth one).

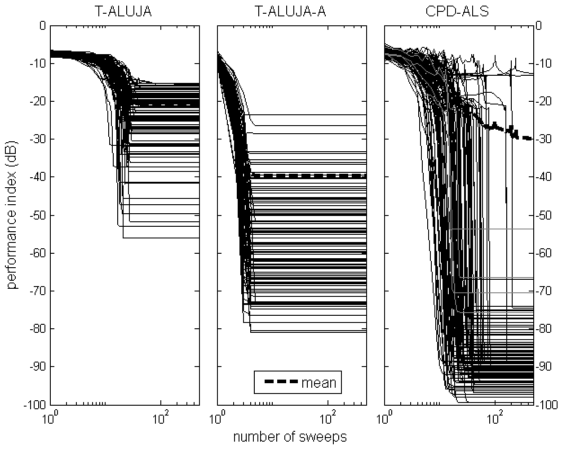

Fig. 2. Performance index versus the number of sweeps for 10 tensors of size $10 \times 10 \times 10$ in the noisy context $t = 10^{-4}$ (superposition of 100 independent draws).

The Fig. 2 displays the index value *w.r.t.* the sweeps of the two Jacobi-like algorithms and CPD-ALS for 100 independent draws for $t = 10^{-4}$. Once again, T-ALUJA-A exhibits a much better convergence speed than the two other algorithms. Moreover, T-ALUJA-A also reaches a better average level of performance after 500 sweeps with $-39.7dB$ against only $-29.9dB$ for CPD-ALS and $-20.8dB$ for T-ALUJA. Notice that, once again, the proposed algorithms are more robust than CPD-ALS of which two draws diverge. However, for the converging draws, CPD-ALS seems to often achieve a better level of performance than T-ALUJA-A.

Finally, the Fig. 3 displays the mean level of performance after 500 sweeps for 500 independent draws *w.r.t.* the level of noise t. T-ALUJA-A always reaches a better mean level of performance than T-ALUJA. CPD-ALS exhibits a good behavior in a very noisy context ($t \geq 10^{-3}$). Nevertheless, CPD-ALS gets an almost constant mean level of performance for each value of t. This is certainly due to some diverging draws. Our T-ALUJA approach seems more robust.

6. CONCLUSION

We proposed one of the first Jacobi-like algorithm for the non-orthogonal joint diagonalization of non-symmetric real-

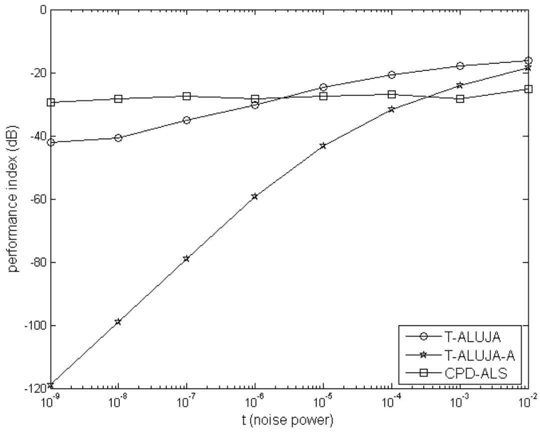

Fig. 3. Mean performance index obtained after 500 sweeps versus the level of noise (500 independent draws of sets of 10 tensors of size $10 \times 10 \times 10$ for each level of noise).

valued third-order tensors. It is based on an LU decomposition and a fully decoupled estimation of the diagonalizing matrices parameters. Two approaches of this algorithm have been tackled. Each one rely on the analytical optimizing solution using an inverse cost function. The first approach takes into account all terms involved by the criterion whereas the second one is based on a selection of involved terms by considering the proximity to a diagonalizing solution. The numerical simulation illustrates the good behavior of both algorithms. Nevertheless T-ALUJA-A provides a much better convergence speed and accuracy than T-ALUJA in the simulation framework. Moreover T-ALUJA-A compares favorably to a CPD-ALS algorithm in term of robustness and convergence speed.

REFERENCES

[1] P. Comon, C. Jutten, eds. *Handbook of Blind Source Separation, Independent Component Analysis and Applications*, Academic Press, 2010.

[2] E. Moreau and T. Adali, *Blind Identification and Separation of Complex-valued Signals*, FOCUS Digital Signal and Image Processing Series, Wiley-ISTE, 2013.

[3] J.-F. Cardoso and A. Souloumiac, "Blind Beamforming for Non Gaussian Signals", *IEE Proceedings-F*, Vol. 140, No. 6, pp. 362–370, 1993.

[4] P. Comon, "Independent Component Analysis, a New Concept?", *Signal processing, Elsevier*, Vol. 36, pp. 287–314, 1994.

[5] E. Moreau, "A Generalization of Joint-Diagonalization Criteria for Source Separation", *IEEE Trans. Signal Processing*, Vol. 49, No. 3, pp. 530–541, March 2001.

[6] L. De Lathauwer, B. De Moor and J. Vandewalle, "Independent Component Analysis and (Simultaneous) Third-Order Tensor Diagonalization", *IEEE Trans. Signal Processing*, Vol. 49, No. 10, pp. 2262–2271, October 2001

[7] T. Trainini and E. Moreau, "A Coordinate Descent Algorithm for Complex Joint Diagonalization Under Hermitian and Transpose Congruences", *IEEE Trans. Signal Processing*, Vol. 62, No. 19, pp. 4974–4983, October 2014.

[8] G. Chabriel, M. Kleinsteuber, E. Moreau, H. Shen, P. Tichavsky and A. Yeredor, "Joint Matrices Decompositions and Blind Source Separation: A survey of methods, identtification, and applications", *IEEE Signal Processing Magazine*, Vol. 31, No. 3, pp. 34–43, May 2014.

[9] B. Afsari, "Simple LU and QR Based Non-Orthogonal Matrix Joint Diagonalization", in *Proc. ICA'2006, Springer LNCS series*, pp. 1–7, 2006.

[10] A. Souloumiac, "Nonorthogonal Joint Diagonalization by Combining Givens and Hyperbolic Rotations", *IEEE Trans. Signal Processing*, Vol. 57, No. 6, pp. 2222–2231, June 2009.

[11] X. Guo, S. Zhu, S. Miron and D. Brie, "Approximate Joint Diagonalization by Nonorthogonal Nonparametric Jacobi Transformations", in *Proc. ICASSP'2010*, Dallas, USA, March 2010.

[12] L. Wang, L. Albera, A. Kachenoura, H. Shu and L. Senhadji, "Nonnegative Joint Diagonalization by Congruence Based on LU Matrix Factorization", *IEEE Signal Processing Letters*, Vol. 20, No. 8, pp. 807–810, August 2013.

[13] V. Maurandi, E. Moreau and C. De Luigi, "Jacobi Like Algorithms for Non-Orthogonal Joint Diagonalization of Hermitian Matrices", in *Proc. ICASSP'2014*, Florence, Italy, May 2014.

[14] 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, December 2014.

[15] T.-G. Kolda and B.-W. Bader, "Tensor decompositions and applications", *SIAM review*, Vol. 51, No. 3, pp. 455–500, August 2009.

[16] A. Cichocki, D. Mandic, A.-H. Phan, C. Caiafa, G. Zhou, Q. Zhao and L. De Lathauwer, "Tensor decompositions for signal processing applications from two-way to multiway component analysis", *arXiv preprint arXiv:1403.4462*, 2014.

[17] P. Comon, "Tensors: a brief introduction", *IEEE Signal Processing Magazine*, Vol. 31, No. 3, pp. 44–53, May 2014.

[18] L. Sorber, M. Van Barel and L. De Lathauwer, *Tensorlab v2.0*, Available online, January 2014. URL: http://www.tensorlab.net/.

[19] E. Moreau and O. Macchi, "A One Stage Self-Adaptive Algorithm for Source Separation", in *Proc. ICASSP'1994*, Vol. 3, pp. 49–52, April 1994.

[20] E. Moreau and O. Macchi, "High-Order Contrasts for Self-Adaptive Source Separation", *International Journal of Adaptive Control and Signal Processing*, Vol. 10, No. 1, pp. 19–46, January 1996.