A Combination of Parallel Factor and Independent Component Analysis

Maarten De Vos, *Student Member, IEEE*, Dimitri Nion, *Member, IEEE*, Sabine Van Huffel, *Fellow, IEEE* and Lieven De Lathauwer, *Senior Member, IEEE*

***Abstract*— Although CPA (Canonical/Parallel factor Analysis) provides a unique decomposition, the actual computation can be made more robust by incorporating extra constraints. In several applications, the factors in one mode are known to be statistically independent. On the other hand, in Independent Component Analysis (ICA), it often makes sense to impose a Khatri-Rao structure on the mixing vectors. In this paper, we propose a new algorithm to impose independence constraints in CPA. Our algorithm implements the algebraic CPA structure and the property of statistical independence simultaneously. Numerical experiments show that our method outperforms in several cases ordinary CPA and ICA. We also present a strategy for imposing (partial) symmetry in CPA.**

I. Introduction

One of the most important tools in linear algebra based signal processing is the Singular Value Decomposition (SVD) [11]. In the case of multi-way data, one of the higher-order generalizations of the SVD can be used. An example of such a multi-way decomposition method is CPA (also known as **C**anonical (CANDECOMP) [8] or **P**arallel Factor (PARAFAC) [12] **A**nalysis). On the other hand, Independent Component Analysis (ICA) can also be considered as a fine-tuning of SVD. With ICA, the data can be transformed in a set of statistically independent components instead of only uncorrelated components.

An important recent research trend concerns the incorporation of constraints or prior knowledge in existing analysis tools, see *e.g.* [47] and references herein. Recently, a new concept arose in the biomedical field. In [1], the idea of combining ICA and CPA was introduced. The authors proposed a trilinear decomposition with independence imposed in one mode. However, the multi-way structure was imposed after the computation of the independent components. A discussion of [1] can be found in [40]. In this paper, we propose an algorithm that imposes the CPA structure during the ICA computation. We will refer to it as the ICA-CPA algorithm.

The results are potentially useful for many applications. CP has found numerous applications in wireless communication [35], [36]. In these applications, it is natural to impose statistical independence of the transmitted signals. A biomedical application is discussed in [1].

We start with some basic definitions (§II), and with reviewing the concepts of ICA (§III-A) and CPA (§III-B). In §IV we derive a new algorithm for computing ICA-CPA. We perform a simulation study to investigate when the proposed method outperforms ordinary ICA and CPA (§V). We conclude with a discussion of the results (§VI). Our formulation is in terms of complex data. Wherever the results for real data cannot simply be obtained by dropping the conjugations, this will be explicitly mentioned.

Preliminary versions of §IV appeared in conference papers [45], [46]. However, this paper presents an improved algorithm.

Notation. Scalars are denoted by lower-case letters (a, b, ...), vectors are written in bold-face lower-case ($\mathbf{a}$, $\mathbf{b}$, ...), matrices correspond to bold-face capitals ($\mathbf{A}$, $\mathbf{B}$, ...) and tensors are written in calligraphic letters ($\mathcal{A}$, $\mathcal{B}$, ...). The ith column vector of a matrix $\mathbf{A}$ is denoted as $\mathbf{a}_i$, i.e., $\mathbf{A} = [\mathbf{a}_1 \mathbf{a}_2 \ldots]$. Italic capitals are also used to denote index upper bounds (*e.g.*, $i = 1, 2, \ldots, I$).

II. Basic Definitions

Definition 1: A third-order tensor $\mathcal{X}$ has rank one if it equals the outer product of three non-zero vectors $\mathbf{a}, textbfb, textbfc$: $x_{ijk} = a_i b_j c_k$ for all values of the indices. The outer product of $\mathbf{a}, \mathbf{b}$ and $\mathbf{c}$ is denoted by $\mathbf{a} \circ \mathbf{b} \circ \mathbf{c}$.

Definition 2: The rank of a tensor is defined as the minimal number of rank-1 terms in which the tensor can be decomposed.

Definition 3: The *Kruskal rank* or *k-rank* of a matrix is the maximal number r such that any set of r columns of the

This research is funded by a PhD grant of the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen). Research Council KUL: CIF1, STRT1/08/023, GOA-AMBioRICS, GOA MaNet, CoE EF/05/006 Optimization in Engineering (OPTEC), IDO 05/010 EEG-fMRI, IDO 08/013 Autism, IOF-KP06/11 FunCopt; Flemish Government: FWO: projects, G.0519.06 (Noninvasive brain oxygenation), FWO-G.0321.06 (Tensors/Spectral Analysis), G.0302.07 (SVM), G.0341.07 (Data fusion), research communities (ICCoS, ANMMM); IWT: TBM070713-Accelero, TBM070706-IOTA3; Belgian Federal Science Policy Office IUAP P6/04 (DYSCO, 'Dynamical systems, control and optimization', 2007-2011); EU: ETUMOUR (FP6-2002-LIFESCIHEALTH 503094), Healthagents (IST200427214), FAST (FP6-MC-RTN-035801), Neuromath (COST-BM0601) ESA: Cardiovascular Control (Prodex-8 C90242)

M. De Vos, L. De Lathauwer and S. Van Huffel are with the Department of Electrical Engineering (ESAT), Katholieke Universiteit Leuven, Leuven, Belgium.

L. De Lathauwer and D. Nion are with the Katholieke Universiteit Leuven Campus Kortrijk, Group Science, Engineering and Technology, Kortrijk, Belgium

contact:maarten.devos@esat.kuleuven.be

matrix is linearly independent. The *k-rank* of $\mathbf{A}$ is denoted by $rank_k(\mathbf{A})$.

Definition 4: The Frobenius norm of a tensor $\mathcal{X} \in \mathbb{C}^{I\times J\times K}$ is defined as

$$||\mathcal{X}||_F = (\sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K}|x_{ijk}|^2)^{\frac{1}{2}} \tag{1}$$

The basic higher-order statistics (HOS) are higher-order moments and cumulants. For random vectors these quantities are higher-order tensors. Cumulants are defined as follows:

Definition 5: For a complex zero-mean stochastic vector $\mathbf{s}$ the *cumulants* of order 2 and 4 are given by:

$$\begin{aligned}(\mathbf{C_s})_{i_1i_2} &= \mathrm{Cum}(s_{i_1}, s^*_{i_2}) \\ &\stackrel{\mathrm{def}}{=} \mathrm{E}\{s_{i_1}s^*_{i_2}\}, \qquad (2)\\ (\mathcal{C}^{(4)}_{\mathbf{s}})_{i_1i_2i_3i_4} &= \mathrm{Cum}(s_{i_1}, s^*_{i_2}, s_{i_3}, s^*_{i_4}) \\ &\stackrel{\mathrm{def}}{=} \mathrm{E}\{s_{i_1}s^*_{i_2}s_{i_3}s^*_{i_4}\} \\ &\quad -\mathrm{E}\{s_{i_1}s^*_{i_2}\}\mathrm{E}\{s_{i_3}s^*_{i_4}\} \\ &\quad -\mathrm{E}\{s_{i_1}s_{i_3}\}\mathrm{E}\{s^*_{i_2}s^*_{i_4}\} \\ &\quad -\mathrm{E}\{s_{i_1}s^*_{i_4}\}\mathrm{E}\{s^*_{i_2}s_{i_3}\}. \qquad (3)\end{aligned}$$

For every component s_i of $\mathbf{s}$ that has a non-zero mean, s_i has to be replaced in these formulas, by $s_i - \mathrm{E}\{s_i\}$.

The second-order cumulant is the covariance matrix. The definition for arbitrary cumulant orders can be found in [22], [28], [29].

The symbol $\otimes$ denotes the Kronecker product:

$$\mathbf{A}\otimes\mathbf{B} := [a_{ij}\mathbf{B}].$$

The symbol $\odot$ denotes the Khatri-Rao or column-wise Kronecker product:

$$\mathbf{A}\odot\mathbf{B} = (\mathbf{a}_1\otimes\mathbf{b}_1 \ldots \mathbf{a}_R\otimes\mathbf{b}_R).$$

We use the Tucker mode-n multiplication. Consider $\mathcal{X} \in \mathbb{C}^{I_1\times I_2\times I_3}$ and $\mathbf{A}\in\mathbb{C}^{J_1\times I_1}$, $\mathbf{B}\in\mathbb{R}^{J_2\times I_2}$, $\mathbf{C}\in\mathbb{C}^{J_3\times I_3}$, then the *Tucker mode-1 product* $\mathcal{X}\times_1\mathbf{A}$, *mode-2 product* $\mathcal{X}\times_2\mathbf{B}$ *and mode-3 product* $\mathcal{X}\times_3\mathbf{C}$ are defined by [43]:

$$\begin{aligned}(\mathcal{X}\times_1\mathbf{A})_{j_1i_2i_3} &= \sum_{i_1=1}^{I_1} x_{i_1i_2i_3}a_{j_1i_1}, \qquad \forall j_1,i_2,i_3,\\ (\mathcal{X}\times_2\mathbf{B})_{i_1j_2i_3} &= \sum_{i_2=1}^{I_2} x_{i_1i_2i_3}b_{j_2i_2}, \qquad \forall i_1,j_2,i_3,\\ (\mathcal{X}\times_3\mathbf{C})_{i_1i_2j_3} &= \sum_{i_3=1}^{I_3} x_{i_1i_2i_3}c_{j_3i_3}, \qquad \forall i_1,i_2,j_3.\end{aligned}$$

III. ICA AND CPA SEPARATELY

A. Independent Component Analysis (ICA)

Problem: We consider the basic linear statistical model

$$\mathbf{x} = \mathbf{M}\cdot\mathbf{s} \quad (+\mathbf{n}) \tag{4}$$

where $\mathbf{x}\in\mathbb{C}^{L\times 1}$ is called the observation vector, $\mathbf{s}\in\mathbb{C}^{R\times 1}$ the source vector and $\mathbf{n}\in\mathbb{C}^{L\times 1}$ additive noise. $\mathbf{M}\in\mathbb{C}^{L\times R}$ is the mixing matrix. We will not explicitly model the noise $\mathbf{n}$. L is the number of sensors and R is the number of sources.

The goal of ICA is to estimate the mixing matrix $\mathbf{M}$, and/or realizations of the source vector $\mathbf{s}$, given only different realizations of the observation vector $\mathbf{x}$. In this work, we assume $R \leqslant L$.

Blind identification of $\mathbf{M}$ in (4) is only possible when other assumptions about the sources are made. The crucial assumption underlying ICA is that the sources are mutually statistically independent, as well as independent from the noise [10], [13].

Algebraic equations: From an algebraic point of view, saying that the source signals contained in $\mathbf{s}$ are uncorrelated means that the covariance matrix of $\mathbf{s}$, $\mathbf{C_s}$, is a diagonal matrix. Saying that the signals in $\mathbf{s}$ are independent also implies that all the higher-order cumulants of $\mathbf{s}$, $\mathcal{C}^{(N)}_{\mathbf{s}}$, are diagonal. With the linear model from equation (7) and using the properties of cumulants, we can write the second- and fourth-order cumulants of $\mathbf{x}$ as:

$$\begin{aligned}\mathbf{C_x} &= \mathbf{C_s}\times_1\mathbf{M}\times_2\mathbf{M}^* + \mathbf{C_n} \qquad (5)\\ \mathcal{C}^{(4)}_{\mathbf{x}} &= \mathcal{C}^{(4)}_{\mathbf{s}}\times_1\mathbf{M}\times_2\mathbf{M}^*\times_3\mathbf{M}\times_4\mathbf{M}^* + \mathcal{C}^{(4)}_{\mathbf{n}} \qquad (6)\end{aligned}$$

in which $\mathbf{C_s}$ and $\mathcal{C}^{(4)}_{\mathbf{s}}$ are diagonal and $\mathcal{C}^{(4)}_{\mathbf{n}}$ drops if the noise is Gaussian. Different kinds of algebraic ICA algorithms can be distinguished, depending on how equations (5) and (6) are combined. One approach is to solve the ICA problem using only higher-order statistics. For algebraic techniques, we refer to [4], [21]. However, many algorithms start with a so-called pre-whitening step [22]. In this pre-whitening step, as much information as possible is extracted from equation (5). In this stage, the mixing matrix can be identified up to an orthogonal rotation matrix. Afterwards, (6) has to be solved in some approximative manner.

The prewhitening step has the disadvantage that the calculations are directly affected by additive Gaussian noise. Recall that (6) is blind to additive Gaussian noise. Errors introduced in the pre-whitening step cannot be fully compensated in the higher-order step [5], [24].

ICA based on soft whitening: A balanced way to combine equations (5) and (6), is called soft whitening [26], [48]. The problem of soft whitening can be formulated as the estimation of a matrix $\mathbf{M}\in\mathbb{C}^{L\times R}$ that minimizes the cost function

$$\begin{aligned}f(\mathbf{M},\mathbf{C_s},\mathcal{C}^{(4)}_{\mathbf{s}}) = &\quad (1-w)\frac{\|\mathbf{C_x}-\mathbf{C_s}\times_1\mathbf{M}\times_2\mathbf{M}^*\|^2}{\|\mathbf{C_x}\|^2}\\ &+w\quad \frac{\|\mathcal{C}^{(4)}_{\mathbf{x}}-\mathcal{C}^{(4)}_{\mathbf{s}}\times_1\mathbf{M}\times_2\mathbf{M}^*\times_3\mathbf{M}\times_4\mathbf{M}^*\|^2}{\|\mathcal{C}^{(4)}_{\mathbf{x}}\|^2}, \qquad (7)\end{aligned}$$

in which $\mathbf{C_s}\in\mathbb{C}^{R\times R}$ is an unknown diagonal matrix and $\mathcal{C}^{(4)}_{\mathbf{s}}\in\mathbb{C}^{R\times R\times\mathbb{R}\times R}$ an unknown diagonal tensor, and in which $\mathbf{C_x}$ and $\mathcal{C}^{(4)}_{\mathbf{x}}$ are the covariance matrix and the fourth-order cumulant tensor of $\mathbf{x}$. A positive value $0\leqslant w\leqslant 1$ is used to weigh the confidence in equations (8) and (9). A small value of w reflects that one has much confidence in the estimate $\mathcal{C}^{(4)}_{\mathbf{x}}$ and small confidence in the estimate $\mathcal{C}_{\mathbf{x}}$ (*e.g.*, in the case of short datasets). When one considers $\mathcal{C}^{(4)}_{\mathbf{x}}$ as more reliable than $\mathbf{C_x}$ (*e.g.*, in the presence of colored Gaussian noise),

one will increase the value of w.

An efficient way to minimize equation (7), is by rewriting the problem as a weighted simultaneous matrix diagonalization [14], [38]. In [7], it was derived that the matrix $\mathbf{M}$ that approximately minimizes the second part of equation (7), can be estimated as the matrix that simultaneously diagonalizes the R dominant eigenmatrices of $(\mathcal{C}_{\mathbf{x}}^{(4)})$. These eigenmatrices are defined as follows. Stack the elements of the fourth-order cumulant $\mathcal{C}_{\mathbf{x}}^{(4)}$ in a matrix $\mathbf{C}_{\mathbf{x}}^{(4)} \in \mathbb{C}^{L^2 \times L^2}$:

$$(\mathbf{C}_{\mathbf{x}}^{(4)})_{i+I(j-1),k+K(l-1)} = (\mathcal{C}_{\mathbf{x}}^{(4)})_{ijkl}. \tag{8}$$

The eigenvectors of $\mathbf{C}_{\mathbf{X}}^{(4)}$ can be stacked again in matrices $\in \mathbb{C}^{L \times L}$. These matrices are called the eigenmatrices of $\mathbf{C}_{\mathbf{X}}^{(4)}$. The R dominant eigenmatrices will be denoted by $\mathbf{U}_r, r = 1, \ldots, R$.
This can be written in another way as

$$\begin{aligned} \mathbf{U}_1 &= \mathbf{M} \cdot \mathbf{D}_1 \cdot \mathbf{M}^H \\ &\vdots \\ \mathbf{U}_R &= \mathbf{M} \cdot \mathbf{D}_R \cdot \mathbf{M}^H \end{aligned} \tag{9}$$

In order to minimize also the first part of eq. (7), we add to this set of matrices the covariance matrix

$$\mathbf{U}_{R+1} = \mathbf{C}_{\mathbf{X}} = \mathbf{M} \cdot \mathbf{D}_{R+1} \cdot \mathbf{M}^H. \tag{10}$$

When we normalize the set of equations (9) and (10) based on their Frobenius norm, and weigh them with respectively a factor w and $1-w$, the resulting simultaneous diagonalization will provide an estimate of the mixing matrix $\mathbf{M}$.

B. Canonical / Parallel factor Analysis (CPA)

Definition: CPA [8], [12], [15], [39] of a third-order tensor $\mathcal{X} \in \mathbb{C}^{I \times J \times K}$ writes $\mathcal{X}$ as a sum of rank-1 terms:

$$\mathcal{X} = \sum_{r=1}^{R} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{s}_r \ (+\mathcal{E}). \tag{11}$$

$\mathcal{E}$ represents a noise term. We assume in this paper that the number of terms is minimal. In that case, R is called the rank of $\mathcal{X}$. A pictorial representation of CPA for third-order tensors is given in Fig. 1.

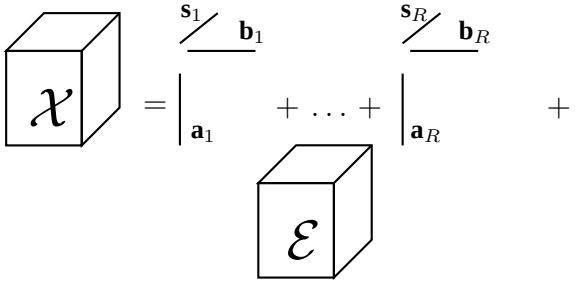

Fig. 1. Pictorial representation of CPA with R components.

Equation (11) can also be expressed as

$$x_{ijk} = \sum_{r=1}^{R} a_{ir} b_{jr} s_{kr} \ (+e_{ijk}), \qquad \forall i, j, k \tag{12}$$

in which $\mathbf{A} \in \mathbb{C}^{I \times R}$, $\mathbf{B} \in \mathbb{C}^{J \times R}$ and $\mathbf{C} \in \mathbb{C}^{K \times R}$. We will from now on drop the noise term e_{ijk} for convenience. Another equivalent and useful expression of the same CPA makes use of the Khatri-Rao product. We assume that $\min(IJ, K) \geqslant R$. Associate with $\mathcal{X}$ a matrix $\mathbf{X} \in \mathbb{C}^{IJ \times K}$ as follows:

$$(\mathbf{X})_{(i-1)J+j,k} = x_{ijk}. \tag{13}$$

This matrix has following structure:

$$\mathbf{X} = (\mathbf{A} \odot \mathbf{B}) \cdot \mathbf{S}^H. \tag{14}$$

Uniqueness: CPA has the advantage that it is unique under mild conditions [16], [37], [41]. When

$$\text{rank}_k(\mathbf{A}) + \text{rank}_k(\mathbf{B}) + \text{rank}_k(\mathbf{C}) \geqslant 2R + 2 \tag{15}$$

is satisfied, CPA is unique. If one of the dimensions is larger than R, a less restrictive condition was derived in [17]. Although this condition guarantees uniquess, it can in real-world applications still be useful to impose extra constraints to order to increase the robustness of the estimation. One could for instance impose non-negativity [9]. In this paper, we impose that the components of $\mathbf{S}$ are statistically independent.

Computation: Consider the minimization of the least-squares cost function associated with CPA:

$$\min_{\mathbf{A},\mathbf{B},\mathbf{S}} ||\mathbf{X} - (\mathbf{A} \odot \mathbf{B})\mathbf{S}^H||^2. \tag{16}$$

This cost function is classically minimized via an algorithm of the Alternating Least Squares type (ALS) [12], [39]. Assume that $\mathbf{A}$ and $\mathbf{B}$ are fixed, the optimal estimation of $\mathbf{C}$ takes the form of a linear least squares problem:

$$\min_{\mathbf{S}} ||\mathbf{X} - \mathbf{Z}\mathbf{S}^H||^2, \tag{17}$$

where $\mathbf{Z}$ equals $\mathbf{A} \odot \mathbf{B}$. Due to the symmetry of the model in the different modes, the updates for all the modes are essentially identical with the role of the different modes shifted. One then alternates between updates of $\mathbf{A}, \mathbf{B}, \mathbf{S}, \mathbf{A}, etc.$ until convergence.
Line search was proposed to speed up the convergence of the ALS algorithm. In [31], [33], the authors show how to compute the optimal step size for respectively real- and complex-valued tensors. This method is called ALS with enhanced line search (ELS).
Other computational schemes have also been proposed, see *e.g.* [17], [23], [27], [32], [34], [42], [44].

IV. Combination of ICA and CPA: ICA-CPA

In this section, we review the tensorial extension of ICA, called 'tensor pICA', as it was introduced in [1]. Then we present a new algorithm to compute CPA of a tensor $\mathcal{X}$ where independence is imposed on the factors in one mode.For notational convenience, we will restrict the exposition to the third-order case, but the generalization to higher orders is straightforward. In the following, we always assume that the

components in the third mode are independent. Due to the symmetric structure of CPA, similar equations can be derived for the other two modes.
In equation (4), we assume that we have K realizations of the observation vector, represented by a matrix $\mathbf{X}$ and that the mixing vector $\mathbf{M}$ has the Khatri-Rao structure $(\mathbf{A} \odot \mathbf{B})$. Alternatively, seen from equation (14), matrix $\mathbf{S}$ contains the independent sources and $(\mathbf{A} \odot \mathbf{B})$ corresponds to the mixing vector $\mathbf{M}$. The rank of $\mathbf{X}$ corresponds to the number of sources R.

A. Tensor pICA

In [1], the following algorithm was proposed.

1) Perform an iteration of the two-dimensional probabilistic ICA algorithm for the decomposition of the data in a mixing matrix $\mathbf{M} \in \mathbb{C}^{IJ \times R}$ and associated source signals $\mathbf{S} \in \mathbb{C}^{K \times R}$: $\mathbf{X} \in \mathbb{C}^{IJ \times K} = \mathbf{M}\mathbf{S}^H + \tilde{\mathbf{E}}_1$.
2) Decompose the estimated mixing matrix $\mathbf{M}$ such that $\mathbf{M} = \mathbf{A} \odot \mathbf{B} + \tilde{\mathbf{E}}_2$. Each column in $\mathbf{A} \odot \mathbf{B}$ is formed by J scaled repetitions of a single column from $\mathbf{A}$, the scaling factors being given by the corresponding column of $\mathbf{B}$. In order to obtain $\mathbf{A}$ and $\mathbf{B}$, we define matrices $\mathbf{G_1}, \ldots, \mathbf{G_R} \in \mathbb{C}^{I \times J}$ as
$$(\mathbf{G_r})_{ij} = m_{(i-1)J+j,r} \qquad \forall i, j, r \tag{18}$$
$\mathbf{a}_r$ and $\mathbf{b}_r$ can then be computed as the dominant left and right singular vector of $\mathbf{G_r}, 1 \leqslant r \leqslant R$.
3) iterate decomposition over steps 1 and 2 until convergence, i.e., until $||\mathbf{A}^{\text{new}} - \mathbf{A}^{\text{old}}||_F + ||\mathbf{B}^{\text{new}} - \mathbf{B}^{\text{old}}||_F + ||\mathbf{C}^{\text{new}} - \mathbf{C}^{\text{old}}||_F < \epsilon$. This third step is in reality redundant, as the decompositions in step 1 and 2 are deterministic and always give the same result.

This algorithm was called tensor pICA, but, roughly speaking it is ordinary ICA, with afterwards the best rank-1 approximation of the mixing vectors, interpreted as $(I \times J)$ matrices.

B. Imposing independence constraints in CPA: ICA-CPA

1) Formulation as a fifth-order CPA with partial symmetry: Recall that in soft whitened ICA, in order to estimate the mixing matrix, the R dominant eigenmatrices $\mathbf{U}_r$ of $\mathcal{C}_{\mathbf{x}}^{(4)}$ have to be jointly (approximately) diagonalized together with the covariance matrix. Stack these $R+1$ matrices $\mathbf{U}_r$ one after the other in a tensor $\mathcal{U} \in \mathbb{R}^{IJ \times IJ \times (R+1)}$. After diagonalization, we can write:
$$\mathbf{U}_r = \mathbf{M} \cdot \mathbf{D}_r \cdot \mathbf{M}^H, \qquad 1 \leqslant r \leqslant R+1 \tag{19}$$
with $\mathbf{D}_r$ a diagonal matrix. Stack these diagonal matrices $\mathbf{D}_r$ in a tensor $\mathcal{D} \in \mathbb{C}^{IJ \times IJ \times (R+1)}$ as follows:
$$(\mathcal{D})_{mpq} = (\mathbf{D}_q)_{mp}. \tag{20}$$
Define $\mathcal{U}$ as follows:
$$(\mathcal{U})_{mpq} = (\mathbf{U}_q)_{mp}. \tag{21}$$
Equation (19) is a standard formulation of the CPA of $\mathcal{U}$ [39]:
$$\mathcal{U} = \sum_{r=1}^{R} \mathbf{m}_r \circ \mathbf{m}_r \circ \mathbf{d}_r. \tag{22}$$
Equation (22) can be written in an element-wise manner as follows:
$$u_{mpq} = \sum_{r=1}^{R} m_{mr} m_{pr} d_{qr}. \tag{23}$$
Now stack $\mathcal{U}$ in a fifth-order tensor $\mathcal{U}^{(5)} \in \mathbb{C}^{I \times J \times I \times J \times (R+1)}$ as follows:
$$(\mathcal{U}^{(5)})_{ijkln} = \mathcal{U}_{(i-1)J+j,(k-1)L+l,n} \tag{24}$$
Due to the Khatri-Rao structure of the mixing matrix, namely $\mathbf{M} = \mathbf{A} \odot \mathbf{B}$, $\mathcal{U}^{(5)}$ has the following CPA structure:
$$\mathcal{U}^{(5)} = \sum_{r=1}^{R} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{d}_r. \tag{25}$$
Indeed, equation (25) can be written in an element-wise manner as follows:
$$(\mathcal{U}^{(5)})_{i,j,k,l,n} = \sum_{r=1}^{R} a_{ir} b_{jr} a_{kr} b_{lr} d_{nr}, \tag{26}$$
which is equivalent to equation (23).

The question is now how equation (25) can be computed, taking into account the partial symmetry.

2) Computing the fifth-order CPA with partial symmetry: It is still largely an open problem how CPA can be computed in the case of partial symmetry. In this section we present a solution. Namely, we explain that (partial) symmetry is naturally preserved in line search schemes.

We want to fit the decomposition in (25) in least squares sense, *i.e.*, we want to minimize the following cost function:
$$\phi(\mathbf{A}, \mathbf{B}, \mathbf{D}) = \|\mathcal{U}^{(5)} - \sum_{r=1}^{R} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{a}_r^* \circ \mathbf{b}_r^* \circ \mathbf{d}_r\|_F^2.$$
In a line search scheme, $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{D}$ are updated in the following manner:
$$\begin{aligned} \mathbf{A}_{p+1} &= \mathbf{A}_p + \lambda \mathbf{\Delta A}_p, \\ \mathbf{B}_{p+1} &= \mathbf{B}_p + \lambda \mathbf{\Delta B}_p, \\ \mathbf{D}_{p+1} &= \mathbf{D}_p + \lambda \mathbf{\Delta D}_p, \end{aligned} \tag{27}$$
in which $\mathbf{\Delta A}_p$, $\mathbf{\Delta B}_p$, $\mathbf{\Delta D}_p$ are the search directions after the pth iteration step for $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{D}$, respectively, and in which λ is the step size. If $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{D}$ are updated as in (27), then (partial) symmetry is naturally preserved. This is an advantage of line search schemes that is not fully appreciated in the literature. We mention here that the popular ALS scheme [12], [39] breaks the symmetry since different estimates of $\mathbf{A}$ are obtained in the first and third mode, and different estimates of $\mathbf{B}$ are obtained in the second and fourth mode.

The line search approach yields two subproblems: (i) the determination of good search directions and (ii) the determination of a good step size. These aspects will now subsequently be discussed.

It makes sense to search along the line that connects the current estimate of $\mathbf{A}$ ($\mathbf{B}$, $\mathbf{D}$) with the estimate $\mathbf{A}_{ALS}$ ($\mathbf{B}_{ALS}$,

$\mathbf{D}_{ALS}$) that would be obtained after one ALS step. That is, the search directions could be defined as follows:

$$\begin{aligned} \mathbf{\Delta A}_p &= \mathbf{A}_{ALS} - \mathbf{A}_p, \\ \mathbf{\Delta B}_p &= \mathbf{B}_{ALS} - \mathbf{B}_p, \\ \mathbf{\Delta D}_p &= \mathbf{D}_{ALS} - \mathbf{D}_p. \end{aligned} \tag{28}$$

In this way, the current estimate is obtained for $\lambda = 0$ and $\mathbf{A}_{ALS}$, $\mathbf{B}_{ALS}$, $\mathbf{D}_{ALS}$ are obtained for $\lambda = 1$. Since the line contains the current estimate, the cost function can be monotonically decreased if suitable values for λ are chosen. The Appendix contains the expressions for $\mathbf{A}_{ALS}$, $\mathbf{B}_{ALS}$ and $\mathbf{D}_{ALS}$ that go with decomposition (25).

In [12] the step size λ is given a fixed value between 1.2 and 1.3. In [3] λ is set to $p^{1/3}$ and the step along the line is accepted only if the cost function decreases. In [33] it was shown that the optimal real-valued step size can be computed by rooting a polynomial. This procedure was generalized to complex-valued λ in [31]. The technique was called Enhanced Line Search with Complex Step (ELSCS). In the Appendix we explain how one can work along the lines of [31], [33] in the specific case of decomposition (25).

V. Numerical experiments

In this section, we illustrate the performance of our algorithm by means of numerical experiments and compare it to standard ICA (as a variant of tensor pICA), and standard CPA. Our ICA-CPA algorithm is soft whitening based. To allow for a fair comparison, we report the performance of an ICA method that is also soft whitening based. The latter amounts to the computation of the CP decomposition (22) by means of the ALS+ELS algorithm [33]. (Tensor pICA itself was computed with pre-whitened fastICA.) For standard CPA, we fit decomposition (11) by means of the ALS+ELS algorithm, without taking into account that the vectors $\mathbf{s}_r$ represent statistically independent signals.

Rank-R tensors $\tilde{\mathcal{X}}$, of which the components in the different modes will be estimated afterwards, are generated in the following way:

$$\tilde{\mathcal{X}} = \sigma \cdot \frac{\mathcal{X}}{||\mathcal{X}||_F} + \frac{\mathcal{N}}{||\mathcal{N}||_F}, \tag{29}$$

in which $\mathcal{X}$ exactly satisfies the CPA model with $R = 4$ independent sources in the third mode (**S**), in which $\mathcal{N}$ represents noise, of which the characteristics will be specified later, and in which σ controls the signal-to-noise ratio.

We conduct 5 different experiments, covering very different conditions, with Monte Carlo simulations consisting of 250 runs.

In a first simulation, we generate $\mathcal{X} \in \mathbb{R}^{5\times4\times400}$ with all source distributions (in **S**) zero-mean and uniformly distributed between $-\sqrt{3}$ and $\sqrt{3}$. The matrices **A** and **B** have mutually orthonormal columns, so these matrices are optimally conditioned. The entries from these matrices are drawn from a zero-mean unit-variance gaussian distribution, and afterwards orthogonalized. The noise is colored. It was generated by multiplying the gaussian noise with a matrix, of which the entries were randomly taken from a uniform distribution. We estimate 4 sources, and stack the entries in $\hat{\mathbf{S}}$. The weighting factor w for soft whitening was chosen equal to 0.3. After estimation, the columns of $\hat{\mathbf{S}}$ are optimally ordered and scaled. We evaluate the performance of the different algorithms by means of the relative Frobenius norm of the difference between the estimated and the real sources:

$$error_S = \frac{||\mathbf{S} - \hat{\mathbf{S}}||_F}{||\mathbf{S}||_F}. \tag{30}$$

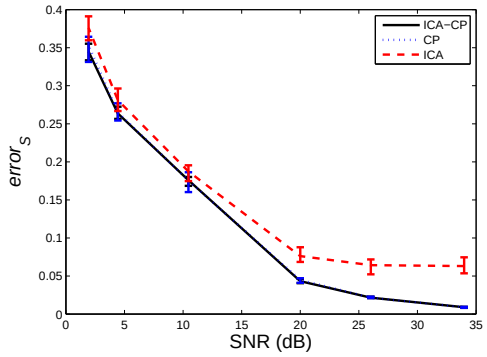

Fig. 2. The median value of $error_S$ as a function of the SNR for the algorithms ICA-CPA (solid), CPA (dotted) and soft whitened ICA (dashed). The 4 sources were uniformly distributed. The mixture ($\in \mathbb{R}^{5\times4\times400}$)was optimally conditioned. The noise was colored gaussian.

In Fig. 2, we plot the median value and the interquartile range of $error_C$ as a function of the SNR.
The solid, dashed and dash-dotted line represent respectively the accuracy of ICA-CPA, ICA and CPA. It can be seen that the estimation accuracy of standard ICA is worse than that of the two other algorithms.

In a second simulation, we assume that all source distributions are binary (1 or -1), with an equal probability of both values. Tensors $\mathcal{X} \in \mathbb{C}^{5\times4\times1000}$ are generated. The entries of the other two modes (**A** and **B**) are drawn from a complex zero-mean unit-variance gaussian distribution. The noise is colored in the same way as in the previous simulation. The decomposition with correct number of components ($R = 4$) is estimated. The weighting factor w for soft whitening was chosen equal to 0.5. We evaluate the performance of the different algorithms by means of the Bit Error Rate (BER). Fig. 3 shows the result of this experiment. It can be seen that for higher noise levels, ICA-CPA outperforms the other two algorithms. Also a high variability of the BER can be observed.

In a third simulation, we generate the 4 sources in **C** and the mixing matrix $\mathbf{A} \odot \mathbf{B}$ in the same way as in the second simulation, but we consider tensors $\mathcal{X} \in \mathbb{C}^{5\times2\times1000}$. Note that for these dimensions, the CPA decomposition is still unique. We consider white gaussian noise. The results are shown in Fig. 4. The figure shows that when one dimension of the tensor is smaller than the number of sources, a small improvement in accuracy can be obtained by incorporating

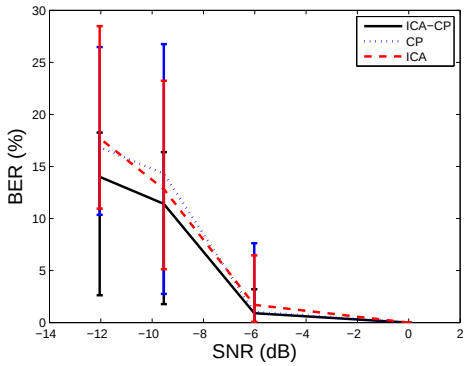

Fig. 3. The median value and interquartile range of BER as a function of the SNR for the algorithms ICA-CPA (solid), CPA (dotted) and ICA (dashed). The 4 sources were binary. The mixture ($\in \mathbb{C}^{5\times 4\times 1000}$) was not optimally conditioned. The noise was colored gaussian.

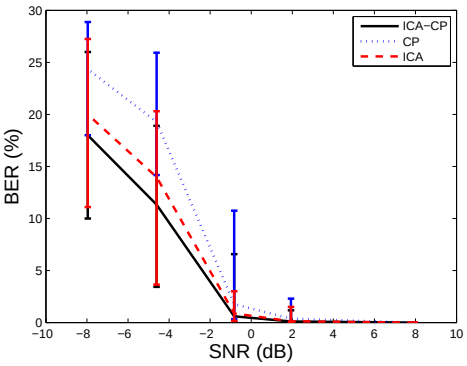

Fig. 5. The median value and interquartile range of BER as a function of the SNR for the algorithms ICA-CPA (solid), CPA (dotted) and ICA (dashed). The sources were binary. The noise was colored gaussian. The observed tensor $\mathcal{X}$ ($\in \mathbb{C}^{5\times 2\times 1000}$) contained contributions of 4 sources.

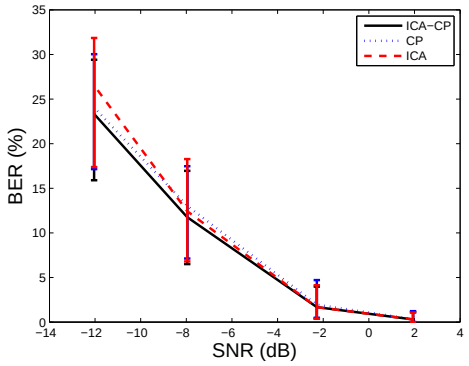

Fig. 4. The median value and interquartile range of BER as a function of the SNR for the algorithms ICA-CPA (solid), CPA (dotted) and ICA (dashed). The sources were binary. The noise was white gaussian. The mixture ($\in \mathbb{C}^{5\times 2\times 1000}$) contained 4 sources.

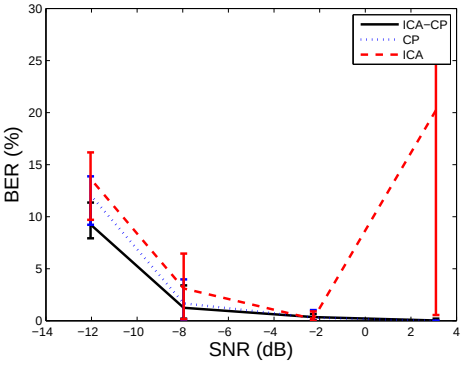

Fig. 6. The median value and interquartile range of BER as a function of the SNR for the algorithms ICA-CPA (solid), CPA (dotted) and ICA (dashed). The sources were binary. The noise was white gaussian. The observed tensor $\mathcal{X}$ ($\in \mathbb{C}^{5\times 4\times 1000}$) contained contributions of 4 sources. The number of components was overestimated by 1.

the rank-1 constraint in the ICA decomposition instead of estimating ordinary CPA.

The fourth simulation (Fig. 5) shows the result when the mixture ($\in \mathbb{C}^{5\times 2\times 1000}$) contained 4 sources and colored noise was present. The noise was generated in the same way as in the second experiment. It can be seen that a substantial increase in accuracy can be obtained by ICA-CPA compared to ordinary ICA and CPA.

The last simulation validates the robustness of the method to overestimation of the rank. Tensors $\mathcal{X} \in \mathbb{C}^{5\times 4\times 1000}$ are generated. $\mathbf{A}, \mathbf{B}$ and $\mathbf{C}$ are generated in the same way as the second simulation. We assume again white gaussian noise and we estimate 5 sources. So we overestimate the number of sources by 1. At the considered high noise levels, it is not always straightforward to estimate the correct number of sources as there is no gap between the contributions. The weighting factor w for soft whitening was again equal to 0.5. Fig. 6 shows the BER as a function of the SNR. It can be seen that ICA-CPA is most robust to overestimation of the rank. Ordinary ICA is not robust at all.

VI. Discussion

In [1], the tensor pICA method was introduced to impose independence constraints in one mode of the CPA. The method boils down to standard ICA, with afterwards the mixing vectors mapped to the closest Kronecker product. In this paper, we develop a new and improved algorithm for the same problem, which we call ICA-CPA. The algorithm simultaneously imposes the independence and the rank-1 structure constraints. We performed several simulations, covering different situations. The simulations were constructed to investigate the accuracy of the new method in (a) an ideal mixture and (b) less ideal mixtures. The latter mixtures were constructed by changing each time one parameter: adding colored noise, considering a tensor with one dimension lower than the number of sources, and overestimating the rank.

In all the experiments, our method outperformed standard ICA. This shows that adding the trilinear constraint improves the accuracy of the decomposition. In a mixture deviating from an ideal case, the increase in accuracy can be very substantial.

It is also interesting to look at the differences between ICA-CPA and standard CPA. In an ideal mixture or at low noise levels, the use of standard CPA is advisable, because it is faster than ICA-CPA (because no cumulant has to be computed) and has comparable accuracy. Although there is a high variability in results, the simulations show clearly that ICA-CPA is more robust than CPA with respect to colored noise. Also when one of the dimensions of the tensor is smaller than the number of components, fitting an ICA-CPA model should be advised. In [1], [40], it was shown that CPA suffered under specific conditions from interference between different components. In these papers, it was claimed that this could be due to overestimation of the number of components. We performed a simulation with white gaussian noise that also confirmed this statement.

It is still largely an open problem how CPA can be performed in the case of full or partial tensor symmetry. A side-result of this paper is that symmetry is naturally preserved in line search schemes. We provided explicit expressions for Enhanced Line Search in the case of real-valued or complex-valued tensor with a specific partial symmetry. The approach can be generalized to tensors of arbitrary order with any type of symmetry.

Our algorithm is based on soft whitening, which means that second and higher-order statistics are taken into account with a predetermined relative weight. It was shown in literature that this approach is sometimes better than pre-whitened based algorithms [6], [38], [48]. The weighting factor w can be optimized for a specific application.

In this paper, we considered an implementation of ICA that was based on the fourth-order cumulant. A similar algorithm can be developed for ICA based on sets of covariance matrices (SOBI) [2], see also [46]. A similar reasoning can also be followed to impose independence constraints in the higher-order block component model (BCM), which is more general than CP [18]–[20], [25], [30].

APPENDIX

Expressions for $\mathbf{A}_{ALS}$, $\mathbf{B}_{ALS}$, $\mathbf{D}_{ALS}$

Define the following matrix representations of $\mathcal{U}^{(5)}$:

$$\begin{aligned}(\mathbf{U}_1^{(5)})_{(j-1)IJ(R+1)+(k-1)J(R+1)+(l-1)(R+1)+n,i} &= (\mathcal{U}^{(5)})_{i,j,k,l,n},\\ (\mathbf{U}_2^{(5)})_{(k-1)IJ(R+1)+(l-1)I(R+1)+(n-1)I+i,j} &= (\mathcal{U}^{(5)})_{i,j,k,l,n},\\ (\mathbf{U}_5^{(5)})_{(i-1)IJ^2+(j-1)IJ+(k-1)J+l,n} &= (\mathcal{U}^{(5)})_{i,j,k,l,n}.\end{aligned}$$

Given $\mathbf{A}_p$, $\mathbf{B}_p$ and $\mathbf{D}_p$, an ALS update in the first mode yields:

$$\mathbf{A}_{ALS} = (\mathbf{U}_1^{(5)})^T \cdot (\mathbf{B}_p \odot \mathbf{A}_p^* \odot \mathbf{B}_p^* \odot \mathbf{D}_p)^{T,\dagger}. \tag{31}$$

An ALS update in the second mode yields:

$$\mathbf{B}_{ALS} = (\mathbf{U}_2^{(5)})^T \cdot (\mathbf{A}_p^* \odot \mathbf{B}_p^* \odot \mathbf{D}_p \odot \mathbf{A}_p)^{T,\dagger}. \tag{32}$$

An ALS update in the fifth mode yields:

$$\mathbf{D}_{ALS} = (\mathbf{U}_5^{(5)})^T \cdot (\mathbf{A}_p \odot \mathbf{B}_p \odot \mathbf{A}_p^* \odot \mathbf{B}_p^*)^{T,\dagger}. \tag{33}$$

Computation of λ

After vectorization, cost function ϕ can be written as

$$\begin{aligned}&\phi(\mathbf{A}_{p+1}, \mathbf{B}_{p+1}, \mathbf{D}_{p+1})\\ &= \|(\mathbf{A}_{p+1} \odot \mathbf{A}_{p+1}^* \odot \mathbf{B}_{p+1} \odot \mathbf{B}_{p+1}^* \odot \mathbf{D}_{p+1}) \cdot \mathbf{1}_R - \mathbf{u}^{(5)}\|_2^2,\end{aligned} \tag{34}$$

where $\mathbf{1}_R = [1, 1, \ldots, 1]^T \in \mathbb{R}^R$ and $\mathbf{u} \in \mathbb{C}^{I^2J^2(R+1)}$ is defined by

$$(\mathbf{u}^{(5)})_{(i-1)IJ^2(R+1)+(k-1)J^2(R+1)+(j-1)J(R+1)+(l-1)(R+1)+n} = (\mathcal{U}^{(5)})_{i,j,k,l,n}.$$

Substitution of (27) in (34) yields equation (35), given on top of the next page, with the vectors $\mathbf{t}_0$ to $\mathbf{t}_{11}$ given in equation (36), also given on the next page. (We dropped the subscripts for notational convenience.)

Let $\mathbf{h} = [\lambda^3(\lambda^2)^*, \lambda^2(\lambda^2)^*, \lambda^3\lambda^*, \lambda^3, \lambda^2\lambda^*, \lambda(\lambda^2)^*, \lambda^2, (\lambda^2)^*, \lambda\lambda^*, \lambda, \lambda^*, 1]^T \in \mathbb{C}^{12}$ and $\mathbf{T} = [\mathbf{t}_0, \ldots, \mathbf{t}_{11}] \in \mathbb{C}^{I^2J^2(R+1)\times 12}$. From (35) we have now that

$$\phi = \mathbf{h}^H \cdot \mathbf{T}^H \cdot \mathbf{T} \cdot \mathbf{h}. \tag{38}$$

From the parameterization $\lambda = a + jb$, we get

$$\begin{bmatrix} \lambda^3(\lambda^2)^* \\ \lambda^2(\lambda^2)^* \\ \lambda^3\lambda^* \\ \lambda^3 \\ \lambda^2\lambda^* \\ \lambda(\lambda^2)^* \\ \lambda^2 \\ (\lambda^2)^* \\ \lambda\lambda^* \\ \lambda \\ \lambda^* \\ 1 \end{bmatrix} = \begin{bmatrix} j & a & 2ja^2 & 2a^3 & ja^4 & a^5 \\ 0 & 1 & 0 & 2a^2 & 0 & a^4 \\ 0 & -1 & 2ja & 0 & 2ja^3 & a^4 \\ 0 & 0 & -j & -3a & 3ja^2 & a^3 \\ 0 & 0 & j & a & ja^2 & a^3 \\ 0 & 0 & -j & a & -ja^2 & a^3 \\ 0 & 0 & 0 & -1 & 2ja & a^2 \\ 0 & 0 & 0 & -1 & -2ja & a^2 \\ 0 & 0 & 0 & 1 & 0 & a^2 \\ 0 & 0 & 0 & 0 & j & a \\ 0 & 0 & 0 & 0 & -j & a \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} b^5 \\ b^4 \\ b^3 \\ b^2 \\ b \\ 1 \end{bmatrix} \stackrel{def}{=} \mathbf{P}_a \cdot \mathbf{b}$$

$$= \begin{bmatrix} 1 & jb & 2b^2 & 2jb^3 & b^4 & jb^5 \\ 0 & 1 & 0 & 2b^2 & 0 & b^4 \\ 0 & 1 & 2jb & 0 & 2jb^3 & -b^4 \\ 0 & 0 & 1 & 3jb & -3b^2 & -jb^3 \\ 0 & 0 & 1 & jb & b^2 & jb^3 \\ 0 & 0 & 1 & -jb & b^2 & -jb^3 \\ 0 & 0 & 0 & 1 & 2jb & -b^2 \\ 0 & 0 & 0 & 1 & -2jb & -b^2 \\ 0 & 0 & 0 & 1 & 0 & b^2 \\ 0 & 0 & 0 & 0 & 1 & jb \\ 0 & 0 & 0 & 0 & 1 & -jb \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a^5 \\ a^4 \\ a^3 \\ a^2 \\ a \\ 1 \end{bmatrix} \stackrel{def}{=} \mathbf{P}_b \cdot \mathbf{a}.$$

Substitution in (38) yields that the cost function can be written as

$$\begin{aligned}\phi &= \mathbf{b}^T \cdot (\mathbf{P}_a^H \cdot \mathbf{T}^H \cdot \mathbf{T} \cdot \mathbf{P}_a) \cdot \mathbf{b},\\ &= \mathbf{a}^T \cdot (\mathbf{P}_b^H \cdot \mathbf{T}^H \cdot \mathbf{T} \cdot \mathbf{P}_b) \cdot \mathbf{a}.\end{aligned}$$

For a fixed, ϕ is a real polynomial of degree 10 in the real variable b. Hence, given a, optimization over b is easy. One just has to compute the roots of the derivative of $\phi(b)$ and select the real root that corresponds to the global minimum. Similarly, for b fixed, optimization over a amounts to computing the roots of the derivative of $\phi(a)$ and selecting the real root that corresponds to the global minimum. We alternate between such conditional updates of a and b, starting with a search over the real line ($b = 0$). The purpose of this iteration is just to increase the convergence speed. It is not theoretically guaranteed that by alternating between updates of a and b, one will find the global optimum for λ. However, since we start from a search over the real line, the alternating updates in the

$$\phi(\lambda) = \|\lambda^3(\lambda^2)^*\mathbf{t}_0 + \lambda^2(\lambda^2)^*\mathbf{t}_1 + \lambda^3\lambda^*\mathbf{t}_2 + \lambda^3\mathbf{t}_3 + \lambda^2\lambda^*\mathbf{t}_4 + \lambda(\lambda^2)^*\mathbf{t}_5 + \lambda^2\mathbf{t}_6 + (\lambda^2)^*\mathbf{t}_7 + \lambda\lambda^*\mathbf{t}_8 + \lambda\mathbf{t}_9 + \lambda^*\mathbf{t}_{10} + \mathbf{t}_{11}\|_F^2, \tag{35}$$

$$\begin{aligned}
\mathbf{t}_0 &= (\boldsymbol{\Delta}\mathbf{A} \odot \boldsymbol{\Delta}\mathbf{A}^* \odot \boldsymbol{\Delta}\mathbf{B} \odot \boldsymbol{\Delta}\mathbf{B}^* \odot \boldsymbol{\Delta}\mathbf{D}) \cdot \mathbf{1}_R, \\
\mathbf{t}_1 &= (\mathbf{A} \odot \boldsymbol{\Delta}\mathbf{A}^* \odot \boldsymbol{\Delta}\mathbf{B} \odot \boldsymbol{\Delta}\mathbf{B}^* \odot \boldsymbol{\Delta}\mathbf{D} + \boldsymbol{\Delta}\mathbf{A} \odot \boldsymbol{\Delta}\mathbf{A}^* \odot \mathbf{B} \odot \boldsymbol{\Delta}\mathbf{B}^* \odot \boldsymbol{\Delta}\mathbf{D}) \cdot \mathbf{1}_R \\
&\quad + (\boldsymbol{\Delta}\mathbf{A} \odot \boldsymbol{\Delta}\mathbf{A}^* \odot \boldsymbol{\Delta}\mathbf{B} \odot \boldsymbol{\Delta}\mathbf{B}^* \odot \mathbf{D}) \cdot \mathbf{1}_R, \\
\mathbf{t}_2 &= (\boldsymbol{\Delta}\mathbf{A} \odot \boldsymbol{\Delta}\mathbf{A}^* \odot \boldsymbol{\Delta}\mathbf{B} \odot \mathbf{B}^* \odot \boldsymbol{\Delta}\mathbf{D} + \boldsymbol{\Delta}\mathbf{A} \odot \mathbf{A}^* \odot \boldsymbol{\Delta}\mathbf{B} \odot \boldsymbol{\Delta}\mathbf{B}^* \odot \boldsymbol{\Delta}\mathbf{D}) \cdot \mathbf{1}_R, \\
\mathbf{t}_3 &= (\boldsymbol{\Delta}\mathbf{A} \odot \mathbf{A}^* \odot \boldsymbol{\Delta}\mathbf{B} \odot \mathbf{B}^* \odot \boldsymbol{\Delta}\mathbf{D}) \cdot \mathbf{1}_R \\
\mathbf{t}_4 &= (\boldsymbol{\Delta}\mathbf{A} \odot \boldsymbol{\Delta}\mathbf{A}^* \odot \boldsymbol{\Delta}\mathbf{B} \odot \mathbf{B}^* \odot \mathbf{D} + \boldsymbol{\Delta}\mathbf{A} \odot \boldsymbol{\Delta}\mathbf{A}^* \odot \mathbf{B} \odot \mathbf{B}^* \odot \boldsymbol{\Delta}\mathbf{D} + \mathbf{A} \odot \boldsymbol{\Delta}\mathbf{A}^* \odot \boldsymbol{\Delta}\mathbf{B} \odot \mathbf{B}^* \odot \boldsymbol{\Delta}\mathbf{D}) \cdot \mathbf{1}_R \\
&+ (\boldsymbol{\Delta}\mathbf{A} \odot \mathbf{A}^* \odot \boldsymbol{\Delta}\mathbf{B} \odot \boldsymbol{\Delta}\mathbf{B}^* \odot \mathbf{D} + \boldsymbol{\Delta}\mathbf{A} \odot \mathbf{A}^* \odot \mathbf{B} \odot \boldsymbol{\Delta}\mathbf{B}^* \odot \boldsymbol{\Delta}\mathbf{D} + \mathbf{A} \odot \mathbf{A}^* \odot \boldsymbol{\Delta}\mathbf{B} \odot \boldsymbol{\Delta}\mathbf{B}^* \odot \boldsymbol{\Delta}\mathbf{D}) \cdot \mathbf{1}_R \\
\mathbf{t}_5 &= (\boldsymbol{\Delta}\mathbf{A} \odot \boldsymbol{\Delta}\mathbf{A}^* \odot \mathbf{B} \odot \boldsymbol{\Delta}\mathbf{B}^* \odot \mathbf{D} + \mathbf{A} \odot \boldsymbol{\Delta}\mathbf{A}^* \odot \boldsymbol{\Delta}\mathbf{B} \odot \boldsymbol{\Delta}\mathbf{B}^* \odot \mathbf{D} + \mathbf{A} \odot \boldsymbol{\Delta}\mathbf{A}^* \odot \mathbf{B} \odot \boldsymbol{\Delta}\mathbf{B}^* \odot \boldsymbol{\Delta}\mathbf{C}) \cdot \mathbf{1}_R, \\
\mathbf{t}_6 &= (\boldsymbol{\Delta}\mathbf{A} \odot \mathbf{A}^* \odot \boldsymbol{\Delta}\mathbf{B} \odot \mathbf{B}^* \odot \mathbf{D} + \boldsymbol{\Delta}\mathbf{A} \odot \mathbf{A}^* \odot \mathbf{B} \odot \mathbf{B}^* \odot \boldsymbol{\Delta}\mathbf{C} + \mathbf{A} \odot \mathbf{A}^* \odot \boldsymbol{\Delta}\mathbf{B} \odot \mathbf{B}^* \odot \boldsymbol{\Delta}\mathbf{D}) \cdot \mathbf{1}_R, \\
\mathbf{t}_7 &= (\mathbf{A} \odot \boldsymbol{\Delta}\mathbf{A}^* \odot \mathbf{B} \odot \boldsymbol{\Delta}\mathbf{B}^* \odot \mathbf{D}) \cdot \mathbf{1}_R, \\
\mathbf{t}_8 &= (\boldsymbol{\Delta}\mathbf{A} \odot \boldsymbol{\Delta}\mathbf{A}^* \odot \mathbf{B} \odot \mathbf{B}^* \odot \mathbf{D} + \mathbf{A} \odot \boldsymbol{\Delta}\mathbf{A}^* \odot \boldsymbol{\Delta}\mathbf{B} \odot \mathbf{B}^* \odot \mathbf{D} + \mathbf{A} \odot \boldsymbol{\Delta}\mathbf{A}^* \odot \mathbf{B} \odot \mathbf{B}^* \odot \boldsymbol{\Delta}\mathbf{D}) \cdot \mathbf{1}_R \\
&+ (\boldsymbol{\Delta}\mathbf{A} \odot \mathbf{A}^* \odot \mathbf{B} \odot \boldsymbol{\Delta}\mathbf{B}^* \odot \mathbf{C} + \mathbf{A} \odot \mathbf{A}^* \odot \boldsymbol{\Delta}\mathbf{B} \odot \boldsymbol{\Delta}\mathbf{B}^* \odot \mathbf{D} + \mathbf{A} \odot \mathbf{A}^* \odot \mathbf{B} \odot \boldsymbol{\Delta}\mathbf{B}^* \odot \boldsymbol{\Delta}\mathbf{D}) \cdot \mathbf{1}_R, \\
\mathbf{t}_9 &= (\boldsymbol{\Delta}\mathbf{A} \odot \mathbf{A}^* \odot \mathbf{B} \odot \mathbf{B}^* \odot \mathbf{D} + \mathbf{A} \odot \mathbf{A}^* \odot \boldsymbol{\Delta}\mathbf{B} \odot \mathbf{B}^* \odot \mathbf{C} + \mathbf{A} \odot \mathbf{A}^* \odot \mathbf{B} \odot \mathbf{B}^* \odot \boldsymbol{\Delta}\mathbf{D}) \cdot \mathbf{1}_R, \\
\mathbf{t}_{10} &= (\mathbf{A} \odot \mathbf{A}^* \odot \mathbf{B} \odot \Delta\mathbf{B}^* \odot \mathbf{D} + \mathbf{A} \odot \boldsymbol{\Delta}\mathbf{A}^* \odot \mathbf{B} \odot \mathbf{B}^* \odot \mathbf{D}) \cdot \mathbf{1}_R, \\
\mathbf{t}_{11} &= (\mathbf{A} \odot \mathbf{A}^* \odot \mathbf{B} \odot \mathbf{B}^* \odot \mathbf{D}) \cdot \mathbf{1}_R - \mathbf{u}.
\end{aligned} \tag{36}$$

$$\phi(\lambda) = \|\lambda^5\mathbf{t}_0 + \lambda^4(\mathbf{t}_1 + \mathbf{t}_2) + \lambda^3(\mathbf{t}_3 + \mathbf{t}_4 + \mathbf{t}_5) + \lambda^2(\mathbf{t}_6 + \mathbf{t}_7 + \mathbf{t}_8) + \lambda(\mathbf{t}_9 + \mathbf{t}_{10}) + \mathbf{t}_{11}\|_F^2. \tag{37}$$

complex plane can only improve the fit. Since the goal is only to increase the convergence speed, just a few updates suffice.

The matrix $\mathbf{T}^H \cdot \mathbf{T}$ can be computed in a relatively efficient way. Since it is Hermitian, only the coefficients of its upper triangular part have to be computed. The following property of the Khatri-Rao product can be used to compute these coefficients efficiently:

$$(\mathbf{A}_1 \odot \ldots \odot \mathbf{A}_N)^H \cdot (\mathbf{B}_1 \odot \cdots \odot \mathbf{B}_N) = (\mathbf{A}_1^H \cdot \mathbf{B}_1) \diamond \cdots \diamond (\mathbf{A}_N^H \cdot \mathbf{B}_N),$$

where $\diamond$ denotes the Hadamard (element-wise) product.

Finally, in the case of real-valued data, expression (35) simplifies to equation (37) given on top of this page. Let $\tilde{\mathbf{h}} = [\lambda^5, \lambda^4, \lambda^3, \lambda^2, \lambda, 1]^T \in \mathbb{R}^6$ and $\tilde{\mathbf{T}} = [\mathbf{t}_0, \mathbf{t}_1 + \mathbf{t}_2, \mathbf{t}_3 + \mathbf{t}_4 + \mathbf{t}_5, \mathbf{t}_6 + \mathbf{t}_7 + \mathbf{t}_8, \mathbf{t}_9 + \mathbf{t}_{10}, \mathbf{t}_{11}] \in \mathbb{R}^{I^2J^2(R+1)\times 6}$. From (37) we have now that

$$\phi = \tilde{\mathbf{h}}^T \cdot \tilde{\mathbf{T}}^T \cdot \tilde{\mathbf{T}} \cdot \tilde{\mathbf{h}}. \tag{39}$$

The cost function is a real polynomial of degree 10 in the real variable λ and can easily be minimized.

REFERENCES

[1] C.F. Beckmann and S.M. Smith. Tensorial extensions of independent component analysis for multisubject fMRI analysis. *Neuroimage*, 25:294–311, 2005.

[2] A. Belouchrani, K. Abed-Meraim, and Y. Hua. Jacobi-like algorithms for joint block diagonalization: Applications to source localization. In *Proc. IEEE Int Work on Intelligent Sig. Proc. and Comm. Sys. (ISPACS)*, Melbourne, Australia, 1998.

[3] R. Bro. *Multi-way analysis in the food industry: models, algorithms, and applications*. PhD thesis, University of Amsterdam, 1998.

[4] J.-F. Cardoso. Iterative techniques for blind source separation using only fourth-order cumulants. In *Proc. of the European Signal Processing Conference (EUSIPCO '92)*, pages 739–742, Brussels, Belgium, 1992.

[5] J.-F. Cardoso. On the performance of orthogonal source separation algorithms. In *Proc. EUSIPCO '94*, pages 776–779, Edinburgh, 1998.

[6] J.-F. Cardoso, S. Bosc, and S. Friedlander. On optimal source separation based on second and fourth order cumulants. In *Proceedings of the 8th IEEE Signal Processing Workshop on Statistical Signal and Array Processing*, 1996.

[7] J.-F. Cardoso and A. Souloumiac. Blind beamforming for non-gaussian signals. *IEE Proc. F*, 140:362–370, 1994.

[8] J.D. Carroll and J. Chang. Analysis of individual differences in multidimensional scaling via an n-way generalization of 'Eckart-Young' decomposition. *Psychometrika*, 35:283–319, 1970.

[9] A. Cichocki, R. Zdunek, A.H. Phan, and S.-I. Amari. *Nonnegative matrix and tensor factorizations*. John Wiley & Sons, 2009.

[10] P. Comon. Independent component analysis, a new concept? *Signal Processing*, 36:287–314, 1994.

[11] G.H. Golub and C.F. Van Loan. *Matrix computations*. Third Edition The Johns Hopkins University Press, 1996.

[12] R.A. Harshman. Foundations of the Parafac procedure: models and conditions for an 'explanatory' multi-modal factor analysis. *UCLA Working Papers in Phonetics*, 16, 1970.

[13] A. Hyvärinen, J. Karhunen and E. Oja. *Independent Component Analysis*. John Wiley & Sons, 2001.

[14] S.M. Kay. *Fundamentals of Statistical Signal Processing*. Prentice-Hall, 1993.

[15] P.M. Kroonenberg. *Applied Multiway Data Analysis*. John Wiley & Sons, 2008.

[16] J.B. Kruskal. Three-way arrays: rank and uniqueness of trilinear decompositions, with applications to arithmetic complexity and statistics. *Psychometrika*, 18:95–138, 1977.

[17] L. De Lathauwer. A link between the canonical decomposition in multilinear algebra and simultaneous matrix diagonalization. *SIAM J Matrix Anal. Appl.*, 28:642–666, 2006.

[18] L. De Lathauwer. Decompositions of a higher-order tensor in block terms — Part I: Lemmas for partitioned matrices. *SIAM J Matrix Anal. Appl.*, 30:1022–1032, 2008.

[19] L. De Lathauwer. Decompositions of a higher-order tensor in block terms — Part II: Definitions and uniqueness. *SIAM J Matrix Anal. Appl.*, 30:1033–1066, 2008.

[20] L. De Lathauwer and A. de Baynast. Blind deconvolution of DS-CDMA Signals by Means of Decomposition in Rank-$(1, L, L)$ Terms. *IEEE Trans. on Signal Processing*, 56:1562–1571, 2008.

[21] L. De Lathauwer, B. De Moor, and J. Vandewalle. Independent component analysis based on higher-order-only ICA. In *Proc. of 8th IEEE SP workshop on Statistical Signal and Array Processing*, pages 356–359, Corfu, Greece, 1996.

[22] L. De Lathauwer, B. De Moor, and J. Vandewalle. An introduction to independent component analysis. *Journal of Chemometrics*, 14:123–149, 2000.

[23] L. De Lathauwer, B. De Moor, and J. Vandewalle. Computation of the canonical decomposition by means of a simultaneous generalized Schur decomposition. *SIAM J Matrix Anal. Appl.*, 26:295–327, 2004.

[24] L. De Lathauwer, B. De Moor, and J. Vandewalle. A prewhitening-induced bound on the identification error in independent component analysis. *IEEE Trans. on Circuits and Systems*, 52:546–554, 2005.

[25] L. De Lathauwer and D. Nion. Decompositions of a higher-order tensor in block terms — Part III: Alternating least squares algorithms. *SIAM J Matrix Anal. Appl.*, 30:1067–1083, 2008.

[26] L. De Lathauwer and J. Vandewalle. Dimensionality reduction in higher-order signal processing and rank-$(R_1, R_2, \ldots, R_N)$ reduction in multilinear algebra. *Linear Algebra and its Applications*, 391:31–55, 2004.

[27] C. Navasca and L. De Lathauwer. Low multilinear rank tensor approximation via semidefinite programming. In *Proceedings 17th European Signal Processing Conference (EUSIPCO 2009)*, Seattle, USA, 2009.

[28] C.L. Nikias and J. Mendel. Signal processing with higher-order spectra. *IEEE Signal Processing Magazine*, pages 10–37, 1993.

[29] C.L. Nikias and A.P. Petropulu. *Higher-Order Spectra Analysis. A Nonlinear Signal Processing Framework.* Englewood Cliffs, N.J.: Prentice-Hall, 1993.

[30] D. Nion and L. De Lathauwer. Block Component Model Based Blind DS-CDMA Receivers. *IEEE Trans. on Signal Processing*, 56:5567–5579, 2008.

[31] D. Nion and L. De Lathauwer. An enhanced line search scheme for complex-valued tensor decompositions. Application in DS-CDMA. *Signal Processing*, 88:749–755, 2008.

[32] P. Paatero. The multilinear engine — a table-driven, least squares program for solving multilinear problems, including the n-way parallel factor analysis model. *Journal of Computational and Graphical Statistics*, 8:854–888, 1999.

[33] M. Rajih, P. Comon, and R.A. Harshman. Enhanced line search: a novel method to accelerate PARAFAC. *SIAM J Matrix Anal. Appl.*, 2007.

[34] N.D. Sidiropoulos A.B. Gershman S.A. Vorobyov, Y. Rong. Robust iterative fitting of multilinear models. *IEEE Transactions on Signal Processing*, 53:2678–2689, 2005.

[35] N. D. Sidiropoulos, R. Bro, and G. B. Giannakis. Parallel factor analysis in sensor array processing. *IEEE Trans. Signal Process.*, 48:23772388, 2000.

[36] N. D. Sidiropoulos, G. B. Giannakis, and R. Bro. Blind parafac receivers for DS-CDMA systems. *IEEE Trans. Signal Process.*, 48:810–823, 2000.

[37] N.D. Siridopoulos and R. Bro. On the uniqueness of multilinear decomposition of n-way arrays. *J Chemometrics*, 14:229–239, 2000.

[38] A. Smekhov and A. Yeredor. Optimization of JADE using a novel optimally weighted joint diagonalization approach. In *Proc. of EUSIPCO*, 2004.

[39] A. Smilde, R. Bro and P. Geladi. *Multi-way Analysis with applications in the Chemical Sciences*. John Wiley & Sons, 2004.

[40] A. Stegeman. Comparing independent component analysis and the parafac model for artificial multi-subject fmri data. *Technical Report, Heymans Institute, University of Groningen, the Netherlands*, 2007.

[41] A. Stegeman and N.D. Siridopoulos. On Kruskal's uniqueness condition for the candecomp/parafac decomposition. *Lin. Algebra and Applications*, 420:540–552, 2007.

[42] G. Tomasi and R. Bro. A comparison of algorithms for fitting the parafac model. *Journal of Computational and Graphical Statistics*, 50:1700–1734, 2006.

[43] L.R. Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31:279–311, 1966.

[44] A.-J. Van Der Veen and A. Paulraj. An analytical constant modulus algorithm. *IEEE Trans. Signal. Proc.*, 44:1136–1155, 1996.

[45] M. De Vos, L. De Lathauwer, and S. Van Huffel. Imposing independence constraints in the CP model. In *Proc. of 7th international conference of independent component analysis*, London, U.K., 2007.

[46] M. De Vos, L. De Lathauwer, and S. Van Huffel. Algorithm for imposing SOBI-type constraints on the CP model. In *Proc. of IEEE International Symposium on Circuits and Systems*, Seattle, USA, 2008.

[47] M. De Vos, L. De Lathauwer, and S. Van Huffel. Spatially constrained ICA algorithms with an application in EEG processing. *submitted*, 2009.

[48] A. Yeredor. Non-orthogonal joint diagonalization in the least-squares sense with application in blind source separation. *IEEE Trans. Signal Processing*, 50:1545–1553, 2002.