NOTICE: this is the author’s version of a work that was accepted for publication in Signal Processing.
Changes resulting from the publishing process, such as peer review, editing, corrections, structural formatting, and other quality control mechanisms may not be reflected in this document. Changes may have been made to this work since it was submitted for publication. A definitive version was subsequently published in Signal Processing, vol. 92, Jul. 2012, pp. 2990-2999.

# A Combination of Parallel Factor and Independent Component Analysis ☆

Maarten De Vos[a,b,d], Dimitri Nion[c], Sabine Van Huffel[a,d], Lieven De Lathauwer[a,c]

[a] *KU Leuven, Department of Electrical Engineering-ESAT, SCD-SISTA, Leuven, Belgium*
[b] *Neuropsychology, Dept. of Psychology, University of Oldenburg, Germany*
[c] *KULeuven Campus Kortrijk, Group Science, Engineering and Technology, Belgium*
[d] *IBBT Future Health Department*

**Abstract**

Although CPA (Canonical/Parallel factor Analysis) has a unique solution, 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 pure CPA, pure ICA, and tensor ICA, a previously proposed method for combining ICA and CPA. We also present a strategy for imposing full or

☆This research is funded by Research Council KUL: CIF1, STRT1/08/023, GOA MaNet, CoE EF/05/006 Optimization in Engineering (OPTEC), IDO 05/010 EEG-fMRI; FWO: projects, G.0519.06 (Noninvasive brain oxygenation), G.0427.10N (Integrated EEG-fMRI); Belgian Federal Science Policy Office IUAP P6/04 (DYSCO, 2007-2011); M De Vos is supported by an Alexander von Humboldt postdoctoral stipend.

*Email address:* maarten.devos@esat.kuleuven.be (Maarten De Vos )

partial symmetry in CPA.

*Keywords:* Independent Component Analysis, Parallel Factor Analysis, Canonical Decomposition, multiway

## 1. Introduction

Canonical or Parallel factor Analysis (CPA) writes multi-way data as a sum of rank-1 components [1, 2]. On the other hand, Independent Component Analysis (ICA) splits data into statistically independent components. In [3], an application was considered in which the data, on one hand, had a three-way CP structure but, on the other hand, could also be rearranged into a matrix that could be analyzed by means of ICA. Since both views made sense, the authors proposed to combine ICA and CPA, i.e., they considered a CP Decomposition (CPD) with independence imposed in one mode. The algorithm to compute the decomposition was called tensor (p)ICA. As we will discuss in section 4.1, the algorithm can be interpreted in two ways, one of which amounts to imposing the CPA structure after the computation of independent components. A discussion of [3] can be found in [4]. In this paper, we propose an algorithm that simultaneously imposes the CPA and ICA structure. We will refer to it as the ICA-CPA algorithm.

Besides being conceptually important, the results are potentially useful for many applications. CPA has found numerous applications in wireless communication [5, 6]. In such applications, it is natural to impose statistical independence of the transmitted signals. A biomedical application is discussed in [3].

We start with some basic definitions (§2), and with reviewing the concepts

of ICA (§3.1) and CPA (§3.2). In §4 we derive a new algorithm for computing ICA-CPA. We show the performance of the proposed method for simulated data. We investigate when the proposed method outperforms tensor (p)ICA and standard CPA (§5). We conclude with a discussion of the results (§6). Our formulation is in terms of complex data.

Preliminary versions of §4 appeared in conference papers [7, 8]. However, this paper presents an improved algorithm and a more detailed discussion.

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 $i$th 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 used to denote index upper bounds (*e.g.*, $i = 1, 2, \ldots, I$). The complex conjugate, conjugated transpose and pseudo-inverse are denoted by $\cdot^*, \cdot^H$ and $\cdot^\dagger$, respectively. The symbol $\otimes$ denotes the Kronecker product, i.e., $\mathbf{A} \otimes \mathbf{B} = [a_{ij}\mathbf{B}]$. The symbol $\odot$ denotes the Khatri-Rao or column-wise Kronecker product, i.e., $\mathbf{A} \odot \mathbf{B} = (\mathbf{a}_1 \otimes \mathbf{b}_1 \ \ldots \ \mathbf{a}_R \otimes \mathbf{b}_R)$. The symbol $*$ denotes the Hadamard or element-wise product. The inner product of matrices is given by $< \mathbf{A}, \mathbf{B} >= \text{trace}(\mathbf{A}^H \cdot \mathbf{B})$.

## 2. Basic Definitions

**Definition 1.** *An $N$th-order tensor $\mathcal{X}$ has rank one if it equals the outer product of $N$ non-zero vectors $\boldsymbol{a}_1$, $\boldsymbol{a}_2$, ..., $\boldsymbol{a}_N$: $x_{i_1 i_2 \ldots i_N} = (\boldsymbol{a}_1)_{i_1} (\boldsymbol{a}_2)_{i_2} \ldots (\boldsymbol{a}_N)_{i_N}$, for all values of the indices. The outer product of $\boldsymbol{a}_1$, $\boldsymbol{a}_2$, ..., $\boldsymbol{a}_N$ is denoted by $\boldsymbol{a}_1 \circ \boldsymbol{a}_2 \circ \ldots \circ \boldsymbol{a}_N$.*

**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 Frobenius norm of a tensor* $\mathcal{X} \in \mathbb{C}^{I_1 \times I_2 \times \ldots \times I_N}$ *is defined as*

$$||\mathcal{X}||_F = (\sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \ldots \sum_{i_n=1}^{I_N} |x_{i_1 i_2 \ldots i_N}|^2)^{\frac{1}{2}} \tag{1}$$

**Definition 4.** *The mode-n product* $\mathcal{X} \cdot_n \mathbf{A} \in \mathbb{C}^{I_1 \times \ldots \times J_n \times \ldots \times I_N}$ *of a tensor* $\mathcal{X} \in \mathbb{C}^{I_1 \times I_2 \times \ldots \times I_N}$ *and a matrix* $\mathbf{A} \in \mathbb{C}^{J_n \times I_n}$ *is defined by:*

$$(\mathcal{X} \cdot_n \mathbf{A})_{i_1 \ldots j_n \ldots i_N} = \sum_{i_n=1}^{I_n} x_{i_1 \ldots i_n \ldots i_N} a_{j_n i_n},$$

*for all index entries.*

We introduce a convenient notation for format changes and index permutations, which we explain by means of an example. Consider a fifth-order tensor $\mathcal{X} = \mathcal{X}_{[1;2;3;4;5]} \in \mathbb{C}^{I_1 \times I_2 \times I_3 \times I_4 \times I_5}$. Then the third-order tensor $\mathcal{X}_{[1,5;4;2,3]} \in \mathbb{C}^{I_1 I_5 \times I_4 \times I_2 I_3}$, the matrix $\mathbf{X}_{[3,1;2,5,4]} \in \mathbb{C}^{I_3 I_1 \times I_2 I_5 I_4}$ and the vector $\mathbf{x}_{[1,2,3,5,4]} \in \mathbb{C}^{I_1 I_2 I_3 I_5 I_4}$ are defined by

$$\begin{aligned} x_{i_1,i_2,i_3,i_4,i_5} &= \left(\mathcal{X}_{[1,5;4;2,3]}\right)_{(i_1-1)I_5+i_5, i_4, (i_2-1)I_3+i_3} \\ &= \left(\mathbf{X}_{[3,1;2,5,4]}\right)_{(i_3-1)I_1+i_1, (i_2-1)I_5 I_4+(i_5-1)I_4+i_4} \\ &= \left(\mathbf{x}_{[1,2,3,5,4]}\right)_{(i_1-1)I_2 I_3 I_5 I_4+(i_2-1)I_3 I_5 I_4+(i_3-1)I_5 I_4+(i_5-1)I_4+i_4}, \end{aligned}$$

for all index entries. Note that a semicolon indicates a new mode. Within a given mode, the indices indicate the order in which the entries are stacked.

## 3. ICA and CPA separately

### *3.1. Independent Component Analysis (ICA)*

We consider the basic linear statistical model

$$\mathbf{x} = \mathbf{M} \cdot \mathbf{s} \quad (+\mathbf{n}) \tag{2}$$

where $\mathbf{x} \in \mathbb{C}^L$ is called the observation vector, $\mathbf{s} \in \mathbb{C}^R$ the source vector and $\mathbf{n} \in \mathbb{C}^L$ additive noise. $\mathbf{M} \in \mathbb{C}^{L \times R}$ is the mixing matrix. $L$ 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 realizations of the observation vector $\mathbf{x}$. In this work, we assume that $\mathbf{M}$ has full column rank, with $R \leqslant L$. After an estimate $\hat{\mathbf{M}}$ of the mixing matrix has been obtained, we can estimate $\mathbf{s}$ as $\hat{\mathbf{s}} = \hat{\mathbf{M}}^{\dagger} \cdot \mathbf{x}$.

Blind identification of $\mathbf{M}$ in (2) is only possible when specific 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 (see [9, 10] and references herein). With the linear model from equation (2) and using the properties of cumulants, we can write the second-order circular cumulant (covariance matrix) and fourth-order circular cumulant (quadricovariance tensor) of $\mathbf{x}$ as:

$$\mathbf{C}_{\mathbf{x}} = \mathbf{C}_{\mathbf{s}} \cdot_1 \mathbf{M} \cdot_2 \mathbf{M}^* + \mathbf{C}_{\mathbf{n}} \tag{3}$$

$$\mathcal{C}^{(4)}_{\mathbf{x}} = \mathcal{C}^{(4)}_{\mathbf{s}} \cdot_1 \mathbf{M} \cdot_2 \mathbf{M}^* \cdot_3 \mathbf{M} \cdot_4 \mathbf{M}^* + \mathcal{C}^{(4)}_{\mathbf{n}} \tag{4}$$

in which $\mathbf{C}_{\mathbf{s}}$ and $\mathcal{C}^{(4)}_{\mathbf{s}}$ are diagonal, because of the statistical independence, and in which $\mathcal{C}^{(4)}_{\mathbf{n}}$ drops if the noise is Gaussian. Different kinds of algebraic ICA algorithms can be distinguished, depending on how equations (3) and (4)

are combined [9, 10]. Many algorithms start with a so-called pre-whitening step, in which as much information as possible is extracted from equation (3). In this paper we will deal with (3) and (4) in a more balanced way.

In [11], it was derived that the matrix $\mathbf{M}$ that satisfies (4), simultaneously diagonalizes the $R$ dominant eigenmatrices of $\mathcal{C}_{\mathbf{x}}^{(4)}$. These eigenmatrices are the eigenvectors of $\mathbf{C}_{\mathbf{x},[2,1;3,4]}^{(4)}$, stacked in matrices of size $L \times L$. The $R$ dominant eigenmatrices will be denoted by $\mathbf{U}_r, r = 1, \ldots, R$. As a heuristic rule, they can be weighted by the square root of the absolute value of the corresponding eigenvalue. The simultaneous matrix diagonalization can be written as

$$\begin{aligned} \mathbf{U}_1 &= \mathbf{M} \cdot \mathbf{D}_1 \cdot \mathbf{M}^H \\ &\vdots \\ \mathbf{U}_R &= \mathbf{M} \cdot \mathbf{D}_R \cdot \mathbf{M}^H, \end{aligned} \tag{5}$$

with $\mathbf{D}_1, \ldots, \mathbf{D}_R$ diagonal. In order to take also (3) into account, we add to this set of matrices the covariance matrix

$$\mathbf{U}_{R+1} = \mathbf{C}_{\mathbf{x}} = \mathbf{M} \cdot \mathbf{D}_{R+1} \cdot \mathbf{M}^H. \tag{6}$$

We will fit (5) and (6) in least-squares sense. In order to balance the fourth-order part (5) and the second-order part (6), we divide them by the associated Frobenius norm, weighting them with positive factors $w = \cos\theta$ and $\sqrt{1-w^2} = \sin\theta$, $0 \leqslant \theta \leqslant \frac{\pi}{2}$, respectively. The determination of the statistically optimal value of $w$ is out of the scope of this paper. However, it is possible to give some general guidelines. A small value of $w$ reflects that one has little confidence in the estimate $\mathcal{C}_{\mathbf{x}}^{(4)}$ and much confidence in the estimate

$\mathbf{C_x}$. This could for instance be the case when one disposes only of a short data set. When one considers $\mathcal{C}_{\mathbf{x}}^{(4)}$ as more reliable than $\mathbf{C_x}$, then one will take a large value for $w$. This could for instance be the case when the data are perturbed by colored Gaussian noise and one disposes of a long data set.

*3.2. Canonical / Parallel factor Analysis (CPA)*

CPA [2, 1, 12, 13] 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:

$$\mathcal{X} = \sum_{r=1}^{R} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{s}_r \ (+\mathcal{E}), \tag{7}$$

in which $\mathbf{a}_r \in \mathbb{C}^I, \mathbf{b}_r \in \mathbb{C}^J$ and $\mathbf{s}_r \in \mathbb{C}^K, r = 1, \ldots, R$. $\mathcal{E}$ represents a noise term. We assume in this paper that the number of terms is minimal. In that case, $R$ is the rank of $\mathcal{X}$. Equation (7) can also be expressed as

$$x_{ijk} = \sum_{r=1}^{R} a_{ir} b_{jr} s_{kr} \ (+e_{ijk}), \qquad \forall i, j, k \tag{8}$$

in which $\mathbf{A} \in \mathbb{C}^{I \times R}$, $\mathbf{B} \in \mathbb{C}^{J \times R}$ and $\mathbf{S} \in \mathbb{C}^{K \times R}$. We will from now on drop the noise term $e_{ijk}$ for convenience. A matrix representation of the CPD is:

$$\mathbf{X}_{[1,2;3]} = (\mathbf{A} \odot \mathbf{B}) \cdot \mathbf{S}^T. \tag{9}$$

The CPD has the advantage that it is essentially unique under mild conditions [14, 15]. This means that the terms in decomposition (7) are unique up to permutation and up to scaling/counterscaling of the factors within the same term. If the CP factors are known to have certain properties, then these could be imposed as constraints. Imposing constraints in certain decomposition techniques has been shown to improve accuracy of the estimated sources (see e.g. [16]). One could for instance impose non-negativity in tensors [17].

Also orthogonality has been proposed, see e.g. [18]. In this paper, we impose that the components of $\mathbf{S}$ are statistically independent.

The least-squares cost function associated with CPA,

$$\min_{\mathbf{A},\mathbf{B},\mathbf{S}} ||\mathbf{X} - (\mathbf{A} \odot \mathbf{B})\mathbf{S}^T||^2, \tag{10}$$

is classically minimized via an algorithm of the Alternating Least Squares type (ALS) [2, 12]. Assuming that $\mathbf{A}$ and $\mathbf{B}$ are fixed, the optimal estimation of $\mathbf{S}$ takes the form of a linear least squares problem:

$$\min_{\mathbf{S}} ||\mathbf{X} - \mathbf{Z}\mathbf{S}^T||^2, \tag{11}$$

where $\mathbf{Z}$ equals $\mathbf{A} \odot \mathbf{B}$. The updates in an other mode are analogous 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. A discussion of optimization-based methods in general can be found in [19]. Many algorithms compute the next iterate by taking a step of size $\lambda$ in a certain search direction (e.g. the direction defined by the ALS solution). In [2] the step size $\lambda$ is given a fixed value between 1.2 and 1.3. In [20] $\lambda$ is set equal to $p^{1/3}$, where $p$ is the iteration number, and the step along the line is accepted only if the cost function decreases. In [21] it was shown that the multilinearity of the CP problem allows the computation of the optimal $\lambda$, when the latter is real. The optimal step can be determined by computing the roots of a polynomial. This results in faster convergence and reduced risk of convergence to non-global minima. The procedure was called Exact Line Search (ELS). In [22] the result was generalized to line search with a complex parameter $\lambda$. In this case it is not guaranteed that the optimal complex step is found. However, the method finds a complex step that decreases the cost function at least as much as

the optimal real step. This method was called Enhanced Line Search with Complex Step (ELSCS).

## 4. Combination of ICA and CPA: ICA-CPA

In this section, we review the tensorial extension of ICA, called 'tensor (p)ICA', as it was introduced in [3]. Then we present a new algorithm to compute the CPD 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 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 (2), we assume that we have $K$ realizations of the observation vector, represented by a matrix $\mathbf{X} \in \mathbb{C}^{IJ \times K}$. We also assume that the mixing matrix $\mathbf{M}$ has the Khatri-Rao structure $\mathbf{A} \odot \mathbf{B}$. Alternatively, seen from equation (9), matrix $\mathbf{S}$ contains the independent sources and $\mathbf{A} \odot \mathbf{B}$ corresponds to the mixing matrix $\mathbf{M}$. The rank of $\mathbf{X}$ corresponds to the number of sources $R$ if $\mathbf{A} \odot \mathbf{B}$ and $\mathbf{S}$ both have full column rank.

### *4.1. Tensor pICA*

In [3], the following algorithm was proposed.

1. Perform an iteration of the pICA algorithm [23] for the decomposition of a data matrix $\mathbf{X} \in \mathbb{C}^{IJ \times K}$ in the product of a mixing matrix $\mathbf{M} \in \mathbb{C}^{IJ \times R}$ and associated source matrix $\mathbf{S} \in \mathbb{C}^{K \times R}$: $\mathbf{X} = \mathbf{M}\mathbf{S}^T (+\tilde{\mathbf{E}}_1)$.

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 $\mathbf{B}$. We define $\mathbf{G}_r = (\mathbf{m}_r)_{[1;2]} \in \mathbb{C}^{I \times J}$, $1 \leqslant r \leqslant R$. The vectors $\mathbf{a}_r$ and $\mathbf{b}_r$ can now be computed as the dominant left and right singular vectors of $\mathbf{G}_r, 1 \leqslant r \leqslant R$.
3. Iterate over steps 1 and 2 until convergence, i.e., until $||\mathbf{A}^{\text{new}} - \mathbf{A}^{\text{old}}||_F + ||\mathbf{B}^{\text{new}} - \mathbf{B}^{\text{old}}||_F + ||\mathbf{S}^{\text{new}} - \mathbf{S}^{\text{old}}||_F < \epsilon$.

Some comments are in order. The pICA algorithm in step 1 may in principle be replaced by any other ICA algorithm, so we will further refer to the overall algorithm as "tensor ICA".

Two variants of the algorithm can be considered. In the first variant, only one ICA iteration step is considered in step 1, and this iteration is initialized with the result of step 2. Step 1 works in the direction of independent sources. Step 2 works in the direction of a Khatri-Rao structured mixing matrix. Since these are two different objectives, the algorithm may fail to converge. This will be illustrated in section 5.

In the second variant, the ICA algorithm in the first step is run until convergence. The overall algorithm can then be iterative to the extent that the ICA algorithm in step 1 hits different local optima. Essentially, the algorithm boils down to computing ICA, with afterwards the best rank-1 approximation of the mixing vectors, interpreted as $(I \times J)$ matrices.

In section 4.2 we propose a new algorithm that imposes both constraints simultaneously.

*4.2. Imposing independence constraints in CPA: ICA-CPA*

*4.2.1. Formulation as a fifth-order CPA with partial symmetry*

Recall that, in the first version of soft whitened ICA in Section 3.1, the estimation of the mixing matrix involves the joint (approximate) diagonalization of the $R$ dominant eigenmatrices $\mathbf{U}_r$ of $\mathcal{C}_{\mathbf{x}}^{(4)}$ together with the covariance matrix. In the second-order variant, a set of covariance matrices has to be diagonalized. In both cases the joint diagonalization problem can be formalized as:

$$\mathbf{U}_k = \mathbf{M} \cdot \mathbf{D}_k \cdot \mathbf{M}^H, \qquad 1 \leqslant k \leqslant K, \tag{12}$$

with $\mathbf{D}_k$ diagonal. Stack the diagonal elements of the matrices $\mathbf{D}_k$ in a matrix $\mathbf{D} \in \mathbb{C}^{K \times R}$ as follows:

$$(\mathbf{D})_{kr} = (\mathbf{D}_k)_{rr} \quad \forall k, r. \tag{13}$$

We stack the $K$ matrices $\mathbf{U}_k$ in a tensor $\mathcal{U} \in \mathbb{C}^{I \times J \times I \times J \times K}$ as follows:

$$u_{i_1 j_1 i_2 j_2 k} = (\mathbf{U}_k)_{(i_1-1)J+j_1,(i_2-1)J+j_2}. \tag{14}$$

Equation (12) can be seen as a CPD of $\mathcal{U}_{[1,2;3,4;5]}$:

$$\mathcal{U}_{[1,2;3,4;5]} = \sum_{r=1}^{R} \mathbf{m}_r \circ \mathbf{m}_r^* \circ \mathbf{d}_r. \tag{15}$$

Due to the Khatri-Rao structure of the mixing matrix, namely $\mathbf{M} = \mathbf{A} \odot \mathbf{B}$, $\mathcal{U}$ has the CP structure

$$\mathcal{U} = \sum_{r=1}^{R} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{a}_r^* \circ \mathbf{b}_r^* \circ \mathbf{d}_r. \tag{16}$$

After obtaining estimates $\hat{\mathbf{a}}_r$ and $\hat{\mathbf{b}}_r$, the mixing matrix is estimated as $\hat{\mathbf{M}} = \hat{\mathbf{A}} \odot \hat{\mathbf{B}}$ and $\mathbf{S}$ is estimated via

$$\hat{\mathbf{S}}^T = \hat{\mathbf{M}}^\dagger \cdot \mathbf{X}. \tag{17}$$

The question is now how decomposition (16) can be computed, taking into account the partial symmetry.

*4.2.2. Computing the fifth-order CPD with partial symmetry*

It is still largely an open problem how a CPD can be computed in the case of full or partial symmetry. In this section we present a solution. Namely, we explain that symmetry is naturally preserved in line search schemes. We want to fit the decomposition in (16) in least squares sense, *i.e.*, we want to minimize the following cost function:

$$\phi(\mathbf{A},\mathbf{B},\mathbf{D}) = \|\mathcal{U} - \sum_{r=1}^{R} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{a}_r^* \circ \mathbf{b}_r^* \circ \mathbf{d}_r\|_F^2. \tag{18}$$

In a line search scheme, $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{D}$ are updated in the following manner:

$$\begin{aligned} \mathbf{A}_{p+1} &= \mathbf{A}_p + \lambda \mathbf{\Delta A}_p, \\ \mathbf{B}_{p+1} &= \mathbf{B}_p + \lambda \mathbf{\Delta B}_p, \\ \mathbf{D}_{p+1} &= \mathbf{D}_p + \lambda \mathbf{\Delta D}_p, \end{aligned} \tag{19}$$

in which $\mathbf{\Delta A}_p$, $\mathbf{\Delta B}_p$, $\mathbf{\Delta D}_p$ are the search directions after the $p$th iteration step for $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{D}$, respectively, and in which $\lambda$ is the step size. If $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{D}$ are updated as in (19), 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 [2, 12] 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 addressed.

In an ALS-type algorithm, we conduct a 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:

$$\begin{aligned} \mathbf{\Delta A}_p &= \mathbf{A}_{ALS} - \mathbf{A}_p, \\ \mathbf{\Delta B}_p &= \mathbf{B}_{ALS} - \mathbf{B}_p, \\ \mathbf{\Delta D}_p &= \mathbf{D}_{ALS} - \mathbf{D}_p. \end{aligned} \tag{20}$$

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 $\lambda$ are chosen. The Appendix contains explicit expressions for $\mathbf{A}_{ALS}$, $\mathbf{B}_{ALS}$ and $\mathbf{D}_{ALS}$ that go with decomposition (16). We also explain how ELS and ELSCS for the determination of the step size $\lambda$ work in the specific case of decomposition (16). In [24] we give expressions for the gradients of the cost function w.r.t. $\mathbf{A}$, $\mathbf{B}$, $\mathbf{D}$. These could be used in gradient and CG descent schemes.

Finally, we mention that, working as in [25, 26], an estimate of $\mathbf{M}$ may be obtained from a Generalized Eigenvalue Decomposition (GEVD) of two $(IJ \times IJ)$ matrix slices of $\mathcal{U}_{[1,2;3,4;5]}$ if $R \leqslant IJ$. Estimates of the matrices $\mathbf{A}$ and $\mathbf{B}$ may be obtained from the estimate of $\mathbf{M}$ as in section 4.1. These estimates are exact if the data are noise-free. When the noise level is not too high, the estimates are expected to be close to the global optimum. Hence, initializing the algorithm with these estimates may significantly reduce the computational cost. Analogous, estimates of $\mathbf{A}$ and $\mathbf{B}$ may be obtained directly from a GEVD of two $(I \times J)$ matrix slices of $\mathcal{U}_{[1;2;3,4,5]}$ if $R \leqslant \min(I, J)$.

*4.2.3. Pseudo-code ICA-CPA*

In summary, ICA-CPA can be computed as follows:

1. Compute the second-order covariance matrix and the dominant eigenmatrices of the fourth order cumulant of the data matrix $\mathbf{X} \in \mathbb{C}^{IJ \times K}$.
2. Rearrange this set of matrices into a fifth-order tensor as in (14).
3. Compute a fifth-order CP decomposition with partial symmetry. One can resort to a line search scheme (19) in which the search direction is taken equal to the ALS search direction (20), (28) and in which the complex step size is computed in an ELSCS manner (29).
4. The mixing matrix is estimated as $\hat{\mathbf{M}} = \hat{\mathbf{A}} \odot \hat{\mathbf{B}}$ and the source signals $\mathbf{S}$ are estimated via
$$\hat{\mathbf{S}}^T = \hat{\mathbf{M}}^\dagger \cdot \mathbf{X}. \tag{21}$$

## 5. Numerical experiments

In this section, we illustrate the performance of our algorithm by means of numerical experiments and compare it to tensor ICA and standard CPA. Our ICA-CPA algorithm is soft whitening based. To allow a fair comparison, we report the performance of tensor ICA based on an ICA method that is also soft whitening based. The latter amounts to the computation of CPA (15) by means of the ALS+ELSCS algorithm [22]. (Tensor pICA itself was computed with pre-whitened fastICA.) For standard CPA, we fit decomposition (7) by means of the ALS+ELSCS algorithm, without taking into account that the vectors $\mathbf{s}_r$ represent statistically independent signals. The search direction in the ICA-CPA algorithm was also taken equal to the ALS direction and the step size determined by means of ELSCS.

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

$$\tilde{\mathcal{X}} = \sigma \cdot \frac{\mathcal{X}}{||\mathcal{X}||_F} + \frac{\mathcal{N}}{||\mathcal{N}||_F}, \tag{22}$$

in which $\mathcal{X}$ exactly satisfies the CPA model with $R = 4$ components that are independent in the third mode ($\mathbf{S}$). $\mathcal{N}$ represents noise of which the characteristics will be specified later. The parameter $\sigma$ controls the signal-to-noise ratio (SNR). We conduct different experiments, covering very different conditions, with Monte Carlo simulations consisting of 250 runs. At every run, the algorithms are 5 times initialized and the best solution is kept. The first experiment shows the convergence of tensor pICA. The next three datasets mimic a DS-CDMA application (see [5]) and the last four datasets are synthetic and are meant to assess specific features of the method. We consider narrow-band sources received by a uniform circular array (UCA) of $I$ identical sensors of radius $P$. We assume free-space propagation. This means that the entries of $\mathbf{A}$ represent the gain between a transmitter and an antenna and are given by

$$a_{jr} = exp(2\pi\sqrt{-1}(x_j \cos(\theta_r)\cos(\phi_r) + y_j \cos(\theta_r)\sin(\phi_r)))$$

where $x_j = (\frac{P}{\mu})\cos(2\pi(j-1)/J)$ and $y_j = (\frac{P}{\mu})\cos(2\pi(j-1)/J)$. We have $\frac{P}{\mu} = 0.55$. We consider $I = 5$ sensors. The directions-of-arrival (DOAs) of the sources are given by $\theta_1 = \frac{3\pi}{10}, \theta_2 = \frac{2\pi}{10}, \theta_3 = \frac{2\pi}{10}, \theta_4 = \frac{3\pi}{10}, \theta_5 = \frac{2\pi}{10}$ and $\phi_1 = \frac{7\pi}{10}, \phi_2 = \frac{9\pi}{10}, \phi_3 = \frac{3\pi}{5}, \phi_4 = \frac{4\pi}{5}, \phi_5 = \frac{3\pi}{10}$. The sources are either BPSK ($\pm 1$) or QAM16 ($x + \sqrt{-1}y$, with $x, y \in \{\pm 1/3, \pm 1\}$). $\mathbf{B}$ contains the chips of the spreading codes for the different users.

As mentioned in section 4.1, we first show that the iterative tensor pICA method (iterating over ICA optimisation step and rank-1 decomposition step), does not necessarily converge. We generated a tensor $\mathcal{X} \in \mathbb{C}^{5\times4\times1000}$. The entries of $\mathbf{B}$ are drawn from a complex zero-mean unit-variance Gaussian distribution, and we estimated four components. Fig. 1 shows the evolution of cost function used in the stop criterion of the tensor ICA method ($||\mathbf{A}^{\text{new}} - \mathbf{A}^{\text{old}}||_F + ||\mathbf{B}^{\text{new}} - \mathbf{B}^{\text{old}}||_F + ||\mathbf{S}^{\text{new}} - \mathbf{S}^{\text{old}}||_F$) during the first 1000 iterations. It can be seen that the algorithm does not converge. In the rest of this section, we use the term "tensor ICA" to refer to the algorithm that determines a full ICA decomposition and afterwards imposes the rank-1 constraint. This is also how the results in [3] were obtained.

In a next experiment, tensors $\mathcal{X} \in \mathbb{C}^{5\times4\times1000}$ are generated. QAM16 sources are considered. The entries in $\mathbf{B}$, representing the 4 spreading codes, are drawn from a complex zero-mean unit-variance Gaussian distribution and orthogonalized. The noise is Gaussian with unknown color. It was generated by multiplying white Gaussian noise with a matrix, of which the entries were randomly drawn from a complex uniform distribution. The decomposition with correct number of components ($R = 4$) is estimated. The weighting factor $w$ for soft whitening was set equal to 0.6. The relatively high value of $w$ allows us to reduce the effect of the colored Gaussian noise. We evaluate the performance of the different algorithms by means of the Bit Error Rate (BER), defined as the percentage of incorrectly estimated bits. Fig. 2 shows the result of this experiment. The solid, dashed and dotted line represent respectively the accuracy of ICA-CPA, tensor ICA and CPA. It can be seen that mainly at higher noise levels, ICA-CPA outperforms the other

two algorithms. Also a high variability of the BER for all the methods can be observed. This variability is due to the condition of the mixing matrix, which is varied over the different runs. Looking at individual runs, ICA-CPA consistently outperforms ICA and CPA separately.

In a third experiment, we generate the mixing matrix $\mathbf{A} \odot \mathbf{B}$ in the same way as in the previous experiment, but we consider tensors $\mathcal{X} \in \mathbb{C}^{5\times 2\times 1000}$ containing 4 BPSK sources in $\mathbf{S}$. Note that for these low dimensions, the CPD is still unique. We consider white Gaussian noise. The weighting factor $w$ for soft whitening was chosen equal to 0.5. The results are shown in Fig. 3. The figure shows that when one dimension of the tensor is smaller than the number of sources, an improvement in accuracy can be obtained by incorporating the rank-1 constraint in the ICA decomposition in comparison to ordinary CPA.

The fourth experiment validates the robustness of the method to overestimation of the rank. Tensors $\mathcal{X} \in \mathbb{C}^{5\times 4\times 1000}$ are generated. $\mathbf{A}$ and $\mathbf{S}$ are generated in the same way as in the second experiment. This time, $\mathbf{B}$ is not orthogonalized. This is realistic, because in the case of multi-path propagation with reflections in the far field of the array, the spreading code will be convolved with the channel response [6]. We assume again white Gaussian noise. We estimate 5 sources, i.e., 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. The weighting factor $w$ for soft whitening was again equal to 0.5. Fig. 4 shows the BER as a function of the SNR. It can be seen that ICA-CPA is most robust to overestimation of the rank.

We also generated several synthetic datasets to assess other features of the newly proposed method. We first generate $\mathcal{X} \in \mathbb{R}^{5\times4\times400}$ with all source distributions (in $\mathbf{S}$) zero-mean and uniformly distributed between $-\sqrt{3}$ and $\sqrt{3}$. The matrices $\mathbf{A}$ and $\mathbf{B}$ have mutually orthonormal columns, so these matrices are optimally conditioned. The entries of these matrices are drawn from a zero-mean unit-variance Gaussian distribution, and afterwards orthogonalized. The noise is colored. It was generated by multiplying 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. The relatively low value of $w$ takes into account that the data set is quite short. 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:

$$error_S = \frac{||\mathbf{S} - \hat{\mathbf{S}}||_F}{||\mathbf{S}||_F}. \tag{23}$$

In Fig. 5, we plot the median value and the interquartile range of $error_S$ as a function of the SNR. It can be seen that the estimation accuracy of tensor ICA is worse than that of the two other algorithms.

One could argue that also orthogonality constraints could improve the estimation of CPA. In order to check if independence constraints can yield better results than orthogonality we perform a next simulation. For estimation of CPA with orthogonality constraints, we used the N-way toolbox, available at www.models.life.ku.dk/source/nwaytoolbox/. Fig. 6 shows the BER after estimation when the mixture $\mathcal{X}$ ($\in \mathbb{R}^{5\times2\times1000}$) contained 4 binary sources. The entries in $\mathbf{A}$ and $\mathbf{B}$ are drawn from a real zero-mean unit-

variance Gaussian distribution. The weight $w$ was taken equal to 0.5. Colored noise, generated in the same way as previously described, was present. It can be seen that imposing orthogonality constraints only slightly improves the estimation accuracy, while independence constraints yield a substantial decrease in BER.

It is known that ICA schemes based on covariance matrix and fourth-order cumulant tensor can handle at most one Gaussian source. One could wonder if ICA-CPA can handle more than one Gaussian source by exploiting the Khatri-Rao structure of the mixing matrix. We repeated the first experiment, but as sources considered two sources that uniformly distributed over the interval $-\sqrt{3}$ and $\sqrt{3}$ together with two zero-mean unit-variance Gaussian sources. The weight $w$ was taken equal to 0.3, to put more emphasis on the second-order statistics. Fig. 7 shows that ICA-CPA, like CPA, was able to separate all sources.

## 6. Discussion

In [3], the tensor ICA method was introduced to impose independence constraints in one mode of the CPD. The method boils down to standard ICA, with afterwards the mixing vectors mapped to the closest Kronecker product. We have developed a new and improved algorithm for the same problem, which we called ICA-CPA. The algorithm simultaneously imposes the independence and the rank-1 structure constraints. In this paper, we worked with the covariance matrix and fourth-order cumulant tensor, like the popular JADE algorithm [11]. It is also possible to work with covariance matrices associated with different time lags, like in the popular SOBI

algorithm [27]. The method is guaranteed to converge to at least a local minimum. It is based on exact/enhanced line search, which yields fast convergence and reduced risk of convergence to non-global optima.

We do not claim that ICA-CPA always performs better than pure ICA or pure CPA. However, our simulations show that in several cases a performance improvement can be expected. The approach seems to be of particular interest for problems that are in some sense difficult (e.g., high level of noise with unknown color, tensor with one dimension lower than the number of sources, overestimated rank, sources not observed in the cumulant tensor, ...). In almost all the experiments, ICA-CPA outperformed tensor ICA, i.e., jointly imposing the CP and independence constraints improves the accuracy. For difficult data sets the improvement can be very substantial.

Pure CPA has the advantage over ICA-CPA that no statistics need to be estimated. It seems that the extra computational cost cannot be justified by a gain in accuracy for well-conditioned mixtures at high SNR. Naturally, also when dependent or correlated sources are present, the traditional CPA will be more appropriate than ICA-CPA. However, ICA-CPA seems to be more robust when the data are corrupted by noise with unknown color. (If the noise covariance is known, one might use the algorithm in [28].) ICA-CPA is also of interest when one of the dimensions of the tensor is smaller than the number of sources. In [3, 4], it was shown that under specific conditions CPA suffers from interference between components. It was claimed that this could be due to overestimation of the number of components. We performed a simulation that confirmed this statement. We also showed that CPA with an independence constraint can yield better results than CPA with merely

an orthogonality constraint.

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 ELS(CS) in the case of a fifth-order tensor with a specific partial symmetry. We introduced a convenient notation for dealing with format changes and index permutations. The approach can be generalized to tensors of arbitrary order with any type of symmetry. In particular, line search schemes can be employed for joint nonunitary diagonalization of a set of symmetric matrices, which can be stacked in a third-order tensor with partial symmetry in its first two modes. This is a key problem in ICA [29, 30, 9].

Our results can be extended to impose independence constraints in Block Component Analysis (BCA), which is more general than CPA [31, 32, 33, 34, 35]. In the context of wireless communication, BCA allows one to deal with more complex propagation scenarios than the ones discussed in section 5 [21, 35].

## Appendix

In this appendix, we provide expressions for the ALS search direction and the optimal step size parameter $\lambda$ in Section 4.2.2. We make use of the following properties of Khatri-Rao products. Consider $\mathbf{x} \in \mathbb{C}^I$, $\mathbf{y} \in \mathbb{C}^J$, $\mathbf{z} \in \mathbb{C}^K$, $\mathbf{X}, \tilde{\mathbf{X}} \in \mathbb{C}^{I \times R}$, $\mathbf{Y}, \tilde{\mathbf{Y}} \in \mathbb{C}^{J \times R}$ and $\mathcal{A} \in \mathbb{C}^{I \times J \times K}$. Then we have $(\mathbf{X} \odot \mathbf{Y})^T \cdot (\tilde{\mathbf{X}} \odot \tilde{\mathbf{Y}}) = (\mathbf{X}^T \cdot \tilde{\mathbf{X}}) * (\mathbf{Y}^T \cdot \tilde{\mathbf{Y}})$ and$(\mathbf{x} \odot \mathbf{y} \odot \mathbf{z})^T \cdot \mathbf{a}_{[1,2,3]} = \mathcal{A}_{[1;2;3]} \cdot_1 \mathbf{x}^T \cdot_2 \mathbf{y}^T \cdot_3 \mathbf{z}^T$. If $(\mathbf{X}^H \cdot \mathbf{X}) * (\mathbf{Y}^H \cdot \mathbf{Y})$ is nonsingular, then we also

have

$$(\mathbf{X} \odot \mathbf{Y})^{\dagger} = ((\mathbf{X}^H \cdot \mathbf{X}) * (\mathbf{Y}^H \cdot \mathbf{Y}))^{-1} \cdot (\mathbf{X} \odot \mathbf{Y})^H. \tag{24}$$

As a consequence of [15, Lemma 2.2], the nonsingularity condition is generically satisfied if $R \leqslant IJ$. Cost function $\phi$ in (18) can equivalently be expressed as

$$\begin{aligned} \phi(\mathbf{A}, \mathbf{B}, \mathbf{D}) &= \|\mathbf{U}_{[1;2,3,4,5]} - \mathbf{A} \cdot (\mathbf{B} \odot \mathbf{A}^* \odot \mathbf{B}^* \odot \mathbf{D})^T\|^2 \\ &= \|\mathbf{U}_{[2;1,3,4,5]} - \mathbf{B} \cdot (\mathbf{A} \odot \mathbf{A}^* \odot \mathbf{B}^* \odot \mathbf{D})^T\|^2 \\ &= \|\mathbf{U}_{[5;1,2,3,4]} - \mathbf{D} \cdot (\mathbf{A} \odot \mathbf{B} \odot \mathbf{A}^* \odot \mathbf{B}^*)^T\|^2 \end{aligned}$$

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

$$\mathbf{A}_{ALS} = \mathbf{U}_{[1;2,3,4,5]} \cdot (\mathbf{B} \odot \mathbf{A}^* \odot \mathbf{B}^* \odot \mathbf{D})^{T,\dagger}. \tag{25}$$

An ALS update in the second mode yields:

$$\mathbf{B}_{ALS} = \mathbf{U}_{[2;1,3,4,5]} \cdot (\mathbf{A} \odot \mathbf{A}^* \odot \mathbf{B}^* \odot \mathbf{D})^{T,\dagger}. \tag{26}$$

An ALS update in the fifth mode yields:

$$\mathbf{D}_{ALS} = \mathbf{U}_{[5;1,2,3,4]} \cdot (\mathbf{A} \odot \mathbf{B} \odot \mathbf{A}^* \odot \mathbf{B}^*)^{T,\dagger}. \tag{27}$$

Expressions (25)–(27) can be efficiently computed as (we drop the subscript $p$ for notational convenience):

$$\begin{aligned} \mathbf{A}_{ALS} &= [\mathcal{U} \cdot_2 \mathbf{b}_1^H \cdot_3 \mathbf{a}_1^T \cdot_4 \mathbf{b}_1^T \cdot_5 \mathbf{d}_1^H \ldots \mathcal{U} \cdot_2 \mathbf{b}_R^H \cdot_3 \mathbf{a}_R^T \cdot_4 \mathbf{b}_R^T \cdot_5 \mathbf{d}_R^H] \\ &\quad \cdot((\mathbf{A}^H \cdot \mathbf{A}) * (\mathbf{B}^H \cdot \mathbf{B}) * (\mathbf{B}^H \cdot \mathbf{B}) * (\mathbf{D}^H \cdot \mathbf{D}))^{-1}, \\ \mathbf{B}_{ALS} &= [\mathcal{U}_{[2;1,3,4,5]} \cdot_2 \mathbf{a}_1^H \cdot_3 \mathbf{a}_1^T \cdot_4 \mathbf{b}_1^T \cdot_5 \mathbf{d}_1^H \ldots \mathcal{U}_{[2;1,3,4,5]} \cdot_2 \mathbf{a}_R^H \cdot_3 \mathbf{a}_R^T \cdot_4 \mathbf{b}_R^T \cdot_5 \mathbf{d}_R^H] \\ &\quad \cdot((\mathbf{A}^H \cdot \mathbf{A}) * (\mathbf{A}^H \cdot \mathbf{A}) * (\mathbf{B}^H \cdot \mathbf{B}) * (\mathbf{D}^H \cdot \mathbf{D}))^{-1}, \\ \mathbf{D}_{ALS} &= [\mathcal{U}_{[5;1,2,3,4]} \cdot_2 \mathbf{a}_1^H \cdot_3 \mathbf{b}_1^H \cdot_4 \mathbf{a}_1^T \cdot_5 \mathbf{b}_1^T \ldots \mathcal{U}_{[5;1,2,3,4]} \cdot_2 \mathbf{a}_R^H \cdot_3 \mathbf{b}_R^H \cdot_4 \mathbf{a}_R^T \cdot_5 \mathbf{b}_R^T] \\ &\quad \cdot((\mathbf{A}^H \cdot \mathbf{A}) * (\mathbf{A}^H \cdot \mathbf{A}) * (\mathbf{B}^H \cdot \mathbf{B}) * (\mathbf{B}^H \cdot \mathbf{B}))^{-1} \end{aligned} \tag{28}$$

$$
\begin{aligned}
\mathbf{t}_1 &= (\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}_2 &= (\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}_3 &= (\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}_4 &= (\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}_5 &= (\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}_6 &= (\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}_7 &= (\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}_8 &= (\mathbf{A} \odot \boldsymbol{\Delta}\mathbf{A}^* \odot \mathbf{B} \odot \boldsymbol{\Delta}\mathbf{B}^* \odot \mathbf{D}) \cdot \mathbf{1}_R, \\
\mathbf{t}_9 &= (\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}_{10} &= (\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}_{11} &= (\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}_{12} &= (\mathbf{A} \odot \mathbf{A}^* \odot \mathbf{B} \odot \mathbf{B}^* \odot \mathbf{D}) \cdot \mathbf{1}_R - \mathbf{u}_{[1,3,2,4,5]}. \qquad (31)
\end{aligned}
$$

---

In order to derive the optimal step size, we can rewrite the cost function $\phi$ after vectorization as

$$\phi(\mathbf{A}_{p+1}, \mathbf{B}_{p+1}, \mathbf{D}_{p+1}) = \|(\mathbf{A}_{p+1} \odot \mathbf{A}^*_{p+1} \odot \mathbf{B}_{p+1} \odot \mathbf{B}^*_{p+1} \odot \mathbf{D}_{p+1}) \cdot \mathbf{1}_R - \mathbf{u}_{[1,3,2,4,5]}\|^2_F, \qquad (29)$$

in which $\mathbf{1}_R = [1, 1, \ldots, 1]^T \in \mathbb{R}^R$. Substitution of (19) in (29) yields:

$$
\begin{aligned}
\phi(\lambda) = \|&\lambda^3(\lambda^2)^*\mathbf{t}_0 + \lambda^2(\lambda^2)^*\mathbf{t}_1 + \lambda^3\lambda^*\mathbf{t}_2 + \lambda^3\mathbf{t}_3 + \lambda^2\lambda^*\mathbf{t}_4 + \lambda(\lambda^2)^*\mathbf{t}_5 \\
&+\lambda^2\mathbf{t}_6 + (\lambda^2)^*\mathbf{t}_7 + \lambda\lambda^*\mathbf{t}_8 + \lambda\mathbf{t}_9 + \lambda^*\mathbf{t}_{10} + \mathbf{t}_{11}\|^2_F, \qquad (30)
\end{aligned}
$$

where the vectors $\mathbf{t}_1$ to $\mathbf{t}_{12}$ are given in equation (31). (Subscript $p$ has been dropped 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 $\mathbf{T} = [\mathbf{t}_1, \ldots, \mathbf{t}_{12}] \in \mathbb{C}^{I^2J^2(R+1)\times 12}$. From (30) we have now that

$$\phi = \mathbf{h}^H \cdot \mathbf{T}^H \cdot \mathbf{T} \cdot \mathbf{h}. \tag{32}$$

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

$$\begin{bmatrix} \lambda^3(\lambda^2)^* \\ \lambda^2(\lambda^2)^* \\ \lambda^3\lambda^* \\ \lambda^3 \\ \lambda^2\lambda^* \\ \lambda(\lambda^2)^* \\ \lambda^2 \\ (\lambda^2)^* \\ \lambda\lambda^* \\ \lambda \\ \lambda^* \\ 1 \end{bmatrix} = \begin{bmatrix} j & a & 2ja^2 & 2a^3 & ja^4 & a^5 \\ 0 & 1 & 0 & 2a^2 & 0 & a^4 \\ 0 & -1 & 2ja & 0 & 2ja^3 & a^4 \\ 0 & 0 & -j & -3a & 3ja^2 & a^3 \\ 0 & 0 & j & a & ja^2 & a^3 \\ 0 & 0 & -j & a & -ja^2 & a^3 \\ 0 & 0 & 0 & -1 & 2ja & a^2 \\ 0 & 0 & 0 & -1 & -2ja & a^2 \\ 0 & 0 & 0 & 1 & 0 & a^2 \\ 0 & 0 & 0 & 0 & j & a \\ 0 & 0 & 0 & 0 & -j & a \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} b^5 \\ b^4 \\ b^3 \\ b^2 \\ b \\ 1 \end{bmatrix} \stackrel{def}{=} \mathbf{P}_a \cdot \mathbf{b} \tag{33}$$

$$= \begin{bmatrix} 1 & jb & 2b^2 & 2jb^3 & b^4 & jb^5 \\ 0 & 1 & 0 & 2b^2 & 0 & b^4 \\ 0 & 1 & 2jb & 0 & 2jb^3 & -b^4 \\ 0 & 0 & 1 & 3jb & -3b^2 & -jb^3 \\ 0 & 0 & 1 & jb & b^2 & jb^3 \\ 0 & 0 & 1 & -jb & b^2 & -jb^3 \\ 0 & 0 & 0 & 1 & 2jb & -b^2 \\ 0 & 0 & 0 & 1 & -2jb & -b^2 \\ 0 & 0 & 0 & 1 & 0 & b^2 \\ 0 & 0 & 0 & 0 & 1 & jb \\ 0 & 0 & 0 & 0 & 1 & -jb \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a^5 \\ a^4 \\ a^3 \\ a^2 \\ a \\ 1 \end{bmatrix} \stackrel{def}{=} \mathbf{P}_b \cdot \mathbf{a}. \tag{34}$$

Substitution of (33-34) in (32) yields that the cost function can be written as

$$\phi = \mathbf{b}^T \cdot (\mathbf{P}_a^H \cdot \mathbf{T}^H \cdot \mathbf{T} \cdot \mathbf{P}_a) \cdot \mathbf{b}, \tag{35}$$

$$= \mathbf{a}^T \cdot (\mathbf{P}_b^H \cdot \mathbf{T}^H \cdot \mathbf{T} \cdot \mathbf{P}_b) \cdot \mathbf{a}. \tag{36}$$

Since the matrix $\mathbf{T}^H \cdot \mathbf{T}$ is Hermitean, it suffices to compute the entries of its upper triangular part. These entries can be computed efficiently, without forming the large matrix $\mathbf{T}$, by using the properties of the Khatri-Rao product given above.

For $a$ fixed, $\phi$ 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 to increase the convergence speed and to reduce the risk of convergence to a local non-global minimum. There is no theoretical guarantee that by alternating between updates of $a$ and $b$, one will find the global optimum for the complex $\lambda$. However, after having found the global optimum on the real line, the subsequent alternating updates in the complex plane can only lead to points that yield an even better fit. In practice just a few alternating updates suffice.

In the case of real-valued data, expression (30) simplifies to

$$\phi(\lambda) = \|\lambda^5\mathbf{t}_1+\lambda^4(\mathbf{t}_2+\mathbf{t}_3)+\lambda^3(\mathbf{t}_4+\mathbf{t}_5+\mathbf{t}_6)+\lambda^2(\mathbf{t}_7+\mathbf{t}_8+\mathbf{t}_9)+\lambda(\mathbf{t}_{10}+\mathbf{t}_{11})+\mathbf{t}_{12}\|_F^2. \tag{37}$$

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

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

The cost function is a real polynomial of degree 10 in the real variable $\lambda$ and can easily be minimised to obtain the optimal stepsize.

## References


[1] J. Carroll, J. Chang, Analysis of individual differences in multidimensional scaling via an n-way generalization of 'Eckart-Young' decomposition, Psychometrika 35 (1970) 283–319.

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

[3] C. Beckmann, S. Smith, Tensorial extensions of independent component analysis for multisubject fMRI analysis, Neuroimage 25 (2005) 294–311.

[4] 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.

[5] N. D. Sidiropoulos, G. B. Giannakis, R. Bro, Blind PARAFAC receivers for DS-CDMA systems, IEEE Trans. Signal Process. 48 (2000) 810–823.

[6] N. D. Sidiropoulos, R. Bro, G. B. Giannakis, Parallel factor analysis in sensor array processing, IEEE Trans. Signal Process. 48 (2000) 2377–2388.

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

[8] M. De Vos, L. De Lathauwer, 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.

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

[10] A. Cichocki, R. Zdunek, A.-H. Phan, S. Amari, Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis and Blind Source Separation., John Wiley, 2009.

[11] J.-F. Cardoso, A. Souloumiac, Blind beamforming for non-Gaussian signals, IEE Proc. F 140 (1994) 362–370.

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

[13] P. Kroonenberg, Applied Multiway Data Analysis., John Wiley & Sons, 2008.

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

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

[16] M. De Vos, L. De Lathauwer, S. Van Huffel, Spatially constrained ICA

algorithms with an application in EEG processing., Signal Processing 91 (2011) 1963–1972.

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

[18] M. Sørensen, L. De Lathauwer, P. Comon, S. Icart, L. Deneire, Computation of a canonical polyadic decomposition with a semi-unitary matrix factor, Tech. Report 11-95, ESAT-SISTA, K.U.Leuven (Leuven, Belgium).

[19] L. Sorber, M. Van Barel, L. De Lathauwer, Optimization-based algorithms for tensor decompositions: Canonical polyadic decomposition, decomposition in rank-$(lr, lr, 1)$ terms and a new generalization, Tech. Report 11-182, ESAT-SISTA, K.U.Leuven (Leuven, Belgium).

[20] R. Bro, Multi-way analysis in the food industry: models, algorithms, and applications, Ph.D. thesis, University of Amsterdam (1998).

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

[22] M. Rajih, P. Comon, R. Harshman, Enhanced line search: a novel method to accelerate PARAFAC, SIAM J. Matrix Anal. Appl. 30 (2007) 1128–1147.

[23] C. Beckmann, S. Smith, Probabilistic independent component analysis for functional magnetic resonance imaging, IEEE Trans Med Imag 23 (2004) 137–152.

[24] L. De Lathauwer, M. De Vos, Gradient-based ICA-CPA, Tech. Report 12-16, ESAT-SISTA, K.U.Leuven (Leuven, Belgium).

[25] S. Leurgans, R. Ross, R. Abel, A decomposition for three-way arrays, SIAM J. Matrix Anal. Appl. 14 (1993) 1064–1083.

[26] E. Sanchez, B. Kowalski, Tensorial resolution: a direct trilinear decomposition, J Chemometrics 4 (1990) 29–45.

[27] A. Belouchrani, K. Abed-Meraim, J.-F. Cardoso, E. Moulines, A blind source separation technique using second-order Statistics, IEEE Trans. Signal Processing 45 (1997) 434–444.

[28] R. Bro, N. Sidiropoulos, A. Smilde, Maximum likelihood fitting using ordinary least squares algorithms., J. of Chemom. 16 (2002) 387–400.

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

[30] L. De Lathauwer, J. Castaing, Blind identification of underdetermined mixtures by simultaneous matrix diagonalization, IEEE Trans. Signal Processing 56 (2008) 1096–1105.

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

[32] L. De Lathauwer, Decompositions of a higher-order tensor in block terms

— Part II: Definitions and uniqueness., SIAM J. Matrix Anal. Appl. 30 (2008) 1033–1066.

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

[34] L. De Lathauwer, 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 (2008) 1562–1571.

[35] D. Nion, L. De Lathauwer, Block Component Model Based Blind DS-CDMA Receivers., IEEE Trans. Sig. Proc. 56 (2008) 5567–5579.

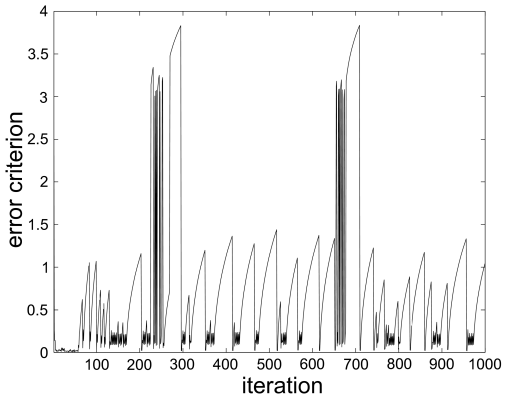


Figure 1: The evolution of the error criterion in tensor ICA over 1000 iterations.

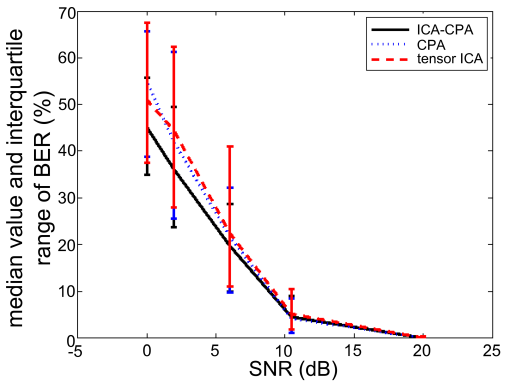


Figure 2: BER as a function of the SNR for the algorithms ICA-CPA (solid), CPA (dotted) and tensor ICA (dashed). The mixture $\mathcal{X}$ ($\in \mathbb{C}^{5\times 4\times 1000}$), containing four QAM16 sources, was not optimally conditioned. The noise was colored Gaussian.

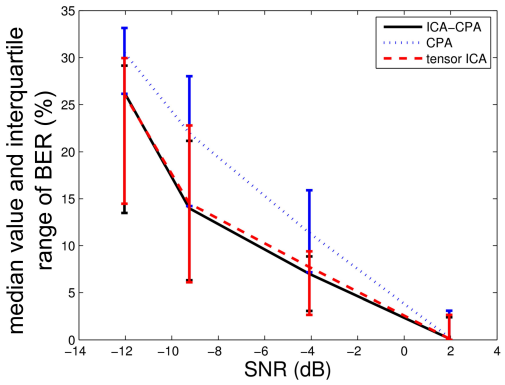


Figure 3: BER as a function of the SNR. The sources were binary. The noise was white Gaussian. The mixture $\mathcal{X}$ ($\in \mathbb{C}^{5\times 2\times 1000}$) contained 4 sources.

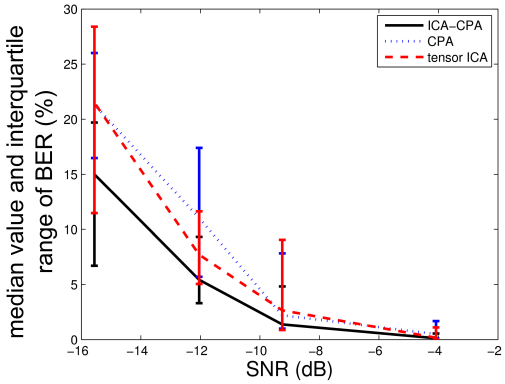


Figure 4: BER as a function of the SNR. The sources were binary. The noise was white Gaussian. The mixture $\mathcal{X}$ ($\in \mathbb{C}^{5\times 4\times 1000}$) contained contributions of 4 sources. The number of components was overestimated by 1.

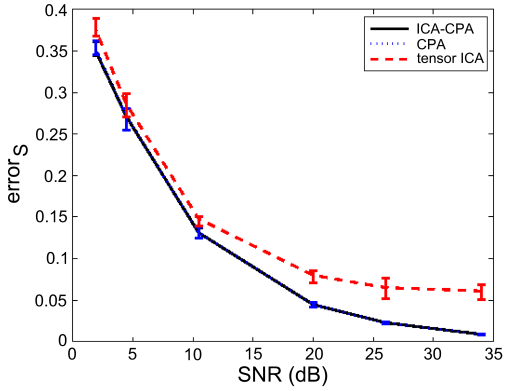


Figure 5: The median value and interquartile range of $error_S$ as a function of the SNR. The 4 sources were uniformly distributed. The mixture $\mathcal{X}$ ($\in \mathbb{R}^{5\times4\times400}$) was optimally conditioned. The noise was colored Gaussian.

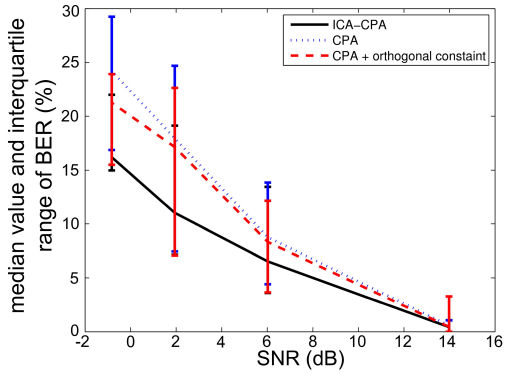


Figure 6: BER as a function of the SNR. The sources were binary. The noise was colored Gaussian. The observed tensor $\mathcal{X}$ ($\in \mathbb{R}^{5\times2\times1000}$) contained contributions of 4 sources.

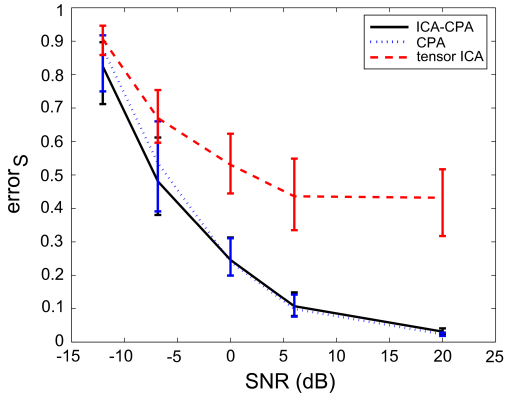


Figure 7: The median value and interquartile range of $error_S$ as a function of the SNR for the algorithms ICA-CPA (solid), CPA (dotted) and soft whitened ICA (dashed). The mixture $\mathcal{X}$ ($\in \mathbb{R}^{5\times4\times1000}$) contained two uniformly distributed and two Gaussian sources. The noise was colored Gaussian.