COMPUTATION OF A CANONICAL POLYADIC DECOMPOSITION WITH A SEMI-UNITARY MATRIX FACTOR

MIKAEL SØRENSEN [*‡], LIEVEN DE LATHAUWER [* ‡] , PIERRE COMON [†], SYLVIE ICART [†], AND LUC DENEIRE [†]

Abstract. Tensor decompositions such as the Canonical Polyadic Decomposition (CPD) have found many practical applications in statistics and signal processing. In certain applications such as blind identification and source separation one of the matrix factors of the CPD is constrained to be semi-unitary. It has been shown that optimal solutions for CPDs with a semi-unitary matrix factor always exist. This paper first explains that if one of the matrix factors of the CPD is semi-unitary, then this additionally leads to a more relaxed uniqueness condition. Second, this paper provides numerical algorithms for the computation of a CPD with a semi-unitary matrix factor. In particular, semi-unitary constrained versions of the CPD methods based on simultaneous matrix diagonalization, simultaneous generalized Schur decomposition and alternating least squares will be discussed. Numerical experiments will also be reported.

Key words. tensors, polyadic decomposition, canonical decomposition (CANDECOMP), parallel factor (PARAFAC), simultaneous matrix diagonalization, simultaneous generalized Schur decomposition, alternating least squares, semi-unitary matrix.

1. Introduction. Symmetric CPDs with unitary matrix factors have found applications in signal processing, in particular in algebraic methods for Independent Component Analysis (ICA) that involve a prewhitening [3], [7], [10]. In this problem the CPD is symmetric and the matrix factors are unitary. Numerically the unitary matrix factors can be found by various Jacobi-type sweeping procedures.

In image processing, unsymmetric CPDs with unitary matrix factors have found use in data representation via a joint Singular Value Decomposition (SVD) [23]. In this problem the CPD is unsymmetric and two of the matrix factors are constrained to be unitary. Again, the unitary matrix factors can be found by means of a Jacobi-type sweeping procedure.

More recently, two iterative methods to compute an unsymmetric CPD involving only orthogonal or semi-unitary matrix factors have been proposed in [22] and [6], respectively. However, no practical applications for these decompositions were presented.

On the other hand, several practical problems involve CPDs where only one of the matrix factors is constrained to be semi-unitary. This problem occurs in the blind identification of underdetermined mixtures [1], [14], in structured ICA problems [2], [26] and in the blind separation of statistically independent or uncorrelated signals in multiple access systems such as DS-CDMA [24].

Optimal solutions for CPDs with a semi-unitary matrix factor always exist [20]. This paper will explain that if one of the matrix factors of the CPD is semi-unitary, then this also leads to a new uniqueness condition for the CPD. In order to compute CPDs with a semi-unitary constrained matrix factor we discuss semi-unitary constrained

[*]Group Science, Engineering and Technology, K.U.Leuven - Kulak, E. Sabbelaan 53, 8500 Kortrijk, Belgium, {Mikael.Sorensen, Lieven.DeLathauwer}@kuleuven-kortrijk.be and K.U.Leuven - E.E. Dept. (ESAT) - SCD-SISTA, Kasteelpark Arenberg 10, B-3001 Leuven-Heverlee, Belgium.

[†]Les Algorithmes - Euclide-B, 06903 Sophia Antipolis, France, {pcomon, icart, deneire}@i3s.unice.fr.

[‡]Research supported by: (1) Research Council K.U.Leuven: GOA-Ambiorics, GOA-MaNet, CoE EF/05/006 Optimization in Engineering (OPTEC), CIF1, STRT1/08/23, (2) F.W.O.: (a) project G.0427.10N, (b) Research Communities ICCoS, ANMMM and MLDM, (3) the Belgian Federal Science Policy Office: IUAP P6/04 (DYSCO, Dynamical systems, control and optimization, 2007-2011), (4) EU: ERNSI.

versions of the Simultaneous matrix Diagonalization (SD-CP) [13], Simultaneous Generalized Schur Decomposition (SGSD-CP) [32], [12], [25], and the Alternating Least Squares (ALS-CP) [17], [28] methods.

The paper is organized as follows. The rest of the introduction will present our notation, followed by a quick review of the CPD. Next, in section 2 we discuss the uniqueness properties of a CPD with a semi-unitary matrix factor. Section 3 proposes a semi-unitary constrained version of the SD-CP method. Next, in section 4, we propose a semi-unitary constrained version of the SGSD-CP method. Semi-unitary constrained versions of the ALS-CP method will be discussed in section 5. In section 6 some numerical experiments are reported. We end the paper with a conclusion in section 7.

1.1. Notations. Vectors, matrices and tensors are denoted by lower case boldface, upper case boldface and upper case calligraphic letters, respectively. The symbol $\otimes$ denotes the Kronecker product

$$\mathbf{A} \otimes \mathbf{B} \triangleq \begin{bmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} & \dots \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} & \dots \\ \vdots & \vdots & \ddots \end{bmatrix}$$

and $\odot$ denotes the Khatri-Rao product

$$\mathbf{A} \odot \mathbf{B} \triangleq \begin{bmatrix} \mathbf{a}_1 \otimes \mathbf{b}_1 & \mathbf{a}_2 \otimes \mathbf{b}_2 & \dots \end{bmatrix},$$

where $\mathbf{a}_r$ and $\mathbf{b}_r$ denote the rth column vector of $\mathbf{A}$ and $\mathbf{B}$, respectively. Furthermore, the outer product of N vectors $\mathbf{a}^{(n)} \in \mathbb{C}^{I_n}$ is denoted by $\mathbf{a}^{(1)} \circ \mathbf{a}^{(2)} \circ \cdots \circ \mathbf{a}^{(N)} \in \mathbb{C}^{I_1 \times I_2 \times \cdots \times I_N}$ and it satisfies the relation

$$\left(\mathbf{a}^{(1)} \circ \mathbf{a}^{(2)} \circ \cdots \circ \mathbf{a}^{(N)}\right)_{i_1 i_2 \dots i_N} = a^{(1)}_{i_1} a^{(2)}_{i_2} \cdots a^{(N)}_{i_N}.$$

Further, $(\cdot)^T$, $(\cdot)^*$, $(\cdot)^H$, $(\cdot)^\dagger$, $\|\cdot\|_F$, $\mathrm{Col}\,(\cdot)$, $\mathrm{Re}\,\{\cdot\}$ and $\mathrm{Im}\{\cdot\}$ denote the transpose, conjugate, conjugate-transpose, Moore-Penrose pseudo-inverse, Frobenius norm, column space, real part and imaginary part of a matrix, respectively. Let $\mathrm{sign}\,(\cdot)$ denote the signum function of a scalar which is equal to

$$\mathrm{sign}\,(x) = \begin{cases} 1 & , x > 0 \\ 0 & , x = 0 \\ -1 & , x < 0 \end{cases}$$

The identity matrix is denoted by $\mathbf{I}_R \in \mathbb{C}^{R \times R}$. A matrix $\mathbf{A} \in \mathbb{C}^{I \times J}$ with $I \geq J$ is said to be semi-unitary if $\mathbf{A}^H \mathbf{A} = \mathbf{I}_J$. Matlab index notation will be used to denote submatrices of a given matrix. For example, $\mathbf{A}(1:k,:)$ denotes the submatrix of $\mathbf{A}$ consisting of the rows from 1 to k.

The notation $\mathrm{triu}\,(\cdot)$ and $\mathrm{diag}\,(\cdot)$ is used to denote the operator that sets the strictly lower triangular elements and off-diagonal elements of a matrix equal to zero, respectively. Let $\mathbf{A} \in \mathbb{C}^{J \times K}$ with $J, K \geq R$, then $\mathrm{triu}_R\,(\mathbf{A})$ is equal to

$$\mathrm{triu}_R\,(\mathbf{A}) = \begin{bmatrix} \mathrm{triu}\,(\mathbf{A}\,(1:R, 1:R)) & \mathbf{0}_{R,K-R} \\ \mathbf{0}_{J-R,R} & \mathbf{0}_{J-R,K-R} \end{bmatrix} \in \mathbb{C}^{J \times K},$$

where $\mathbf{0}_{M,N} \in \mathbb{C}^{M \times N}$ is an all-zero matrix.

Let $\mathbf{A} \in \mathbb{C}^{I \times J}$, then $\text{Vec}(\mathbf{A}) \in \mathbb{C}^{IJ}$ denotes the column vector defined by $(\text{Vec}(\mathbf{A}))_{i+(j-1)I} = (\mathbf{A})_{ij}$. Let $\mathbf{a} \in \mathbb{C}^{IJ}$, then the reverse operation is $\text{Unvec}(\mathbf{a}) = \mathbf{A} \in \mathbb{C}^{I \times J}$ such that $(\mathbf{a})_{i+(j-1)I} = (\mathbf{A})_{ij}$. Let $\mathbf{A} \in \mathbb{C}^{I \times I}$, then $\text{Vecd}(\mathbf{A}) \in \mathbb{C}^{I}$ denotes the column vector defined by $(\text{Vecd}(\mathbf{A}))_i = (\mathbf{A})_{ii}$. Let $\mathbf{A} \in \mathbb{C}^{I \times J}$, then $D_k(\mathbf{A}) \in \mathbb{C}^{J \times J}$ denotes the diagonal matrix holding row k of $\mathbf{A}$ on its diagonal.

The k-rank of a matrix $\mathbf{A}$ is denoted by $\text{k}(\mathbf{A})$. It is equal to the largest integer $\text{k}(\mathbf{A})$ such that every subset of $\text{k}(\mathbf{A})$ columns of $\mathbf{A}$ is linearly independent.

1.2. Canonical Polyadic Decomposition (CPD). A Nth-order rank-1 tensor $\mathcal{X} \in \mathbb{C}^{I_1 \times \cdots \times I_N}$ is defined as the outer product of non-zero vectors $\mathbf{a}^{(n)} \in \mathbb{C}^{I_n}$, $n \in [1, N]$, such that $\mathcal{X}_{i_1 \ldots i_N} = \prod_{n=1}^{N} a^{(n)}_{i_n}$. The rank of a tensor $\mathcal{X}$ is equal to the minimal number of rank-1 tensors that yield $\mathcal{X}$ in a linear combination. Assume that the rank of $\mathcal{X}$ is R, then it can be written as

$$\mathcal{X} = \sum_{r=1}^{R} \mathbf{a}_r^{(1)} \circ \cdots \circ \mathbf{a}_r^{(N)}, \tag{1.1}$$

where $\mathbf{a}_r^{(n)} \in \mathbb{C}^{I_n}$. This decomposition is sometimes referred to as the PARAllel FACtor (PARAFAC) [17] or the CANonical DECOMPosition (CANDECOMP) [5], but in this paper it will be referred to as the Canonical Polyadic Decomposition (CPD) of $\mathcal{X}$. Let us stack the vectors $\{\mathbf{a}_r^{(n)}\}$ into the matrices

$$\mathbf{A}^{(n)} = \left[\begin{array}{ccc} \mathbf{a}_1^{(n)}, & \cdots & , \mathbf{a}_R^{(n)} \end{array} \right] \in \mathbb{C}^{I_n \times R}, \quad n \in [1, N]. \tag{1.2}$$

The matrices $\mathbf{A}^{(n)}$ in (1.2) will be referred to as the matrix factors of the tensor $\mathcal{X}$ in (1.1). Consider $\mathcal{X} \in \mathbb{C}^{I_1 \times \cdots \times I_N}$ and let the row vectors $\mathbf{x}^{(i_1, \ldots, i_{N-1})} \in \mathbb{C}^{1 \times I_N}$ be constructed such that $\mathbf{x}^{(i_1 \ldots, i_{N-1})}_{i_N} = \mathcal{X}_{i_1, \ldots, i_N}$, then

$$\mathbf{x}^{(i_1, \ldots, i_{N-1})} = \sum_{i_N=1}^{I_N} \mathcal{X}_{i_1 \ldots i_N} \mathbf{e}^{(I_N)T}_{i_N} = \sum_{r=1}^{R} \prod_{n=1}^{N-1} \mathbf{A}^{(n)}_{i_n r} \mathbf{a}_r^{(N)}$$

where $\mathbf{e}^{(I_N)}_{i_N} \in \mathbb{C}^{I_N}$ is a unit vector with unit element at entry i_N and zero elsewhere. Stack the vectors $\{\mathbf{x}^{(i_1, \ldots, i_{N-1})}\}$ into the matrix

$$\mathbb{C}^{\prod_{n=1}^{N-1} I_n \times I_N} \ni \mathbf{X}_{[N]} \triangleq \left[\begin{array}{c} \mathbf{x}^{(1, \ldots, 1)} \\ \mathbf{x}^{(1, \ldots, 2)} \\ \vdots \\ \mathbf{x}^{(I_1, \ldots, I_{N-1})} \end{array} \right] = \left(\mathbf{A}^{(1)} \odot \cdots \odot \mathbf{A}^{(N-1)} \right) \mathbf{A}^{(N)T}. \tag{1.3}$$

The matrix representation (1.3) of a Nth-order CPD will be used throughout the paper. Furthermore, the following two matrix representations of a CPD of a third-order tensor $\mathcal{X} \in \mathbb{C}^{I_1 \times I_2 \times I_3}$ will be used throughout the paper. Let $\mathbf{X}^{(i_1 \cdot \cdot)} \in \mathbb{C}^{I_2 \times I_3}$ denote the matrix such that $\mathbf{X}^{(i_1 \cdot \cdot)}_{i_2 i_3} = \mathcal{X}_{i_1 i_2 i_3}$, then

$$\mathbf{X}^{(i_1 \cdot \cdot)} = \mathbf{A}^{(2)} D_{i_1}\left(\mathbf{A}^{(1)} \right) \mathbf{A}^{(3)T}$$

and

$$\mathbb{C}^{I_1 I_2 \times I_3} \ni \mathbf{X}_{(1)} \triangleq \left[\begin{array}{c} \mathbf{X}^{(1 \cdot \cdot)} \\ \vdots \\ \mathbf{X}^{(I_1 \cdot \cdot)} \end{array} \right] = \left[\begin{array}{c} \mathbf{A}^{(2)} D_1\left(\mathbf{A}^{(1)} \right) \\ \vdots \\ \mathbf{A}^{(2)} D_{I_1}\left(\mathbf{A}^{(1)} \right) \end{array} \right] \mathbf{A}^{(3)T} = \left(\mathbf{A}^{(1)} \odot \mathbf{A}^{(2)} \right) \mathbf{A}^{(3)T}.$$

Similarly, let the matrices $\mathbf{X}^{(\cdot i_2 \cdot)} \in \mathbb{C}^{I_3 \times I_2}$ be constructed such that $\mathbf{X}^{(\cdot i_2 \cdot)}_{i_3 i_1} = \mathcal{X}_{i_1 i_2 i_3}$, then

$$\mathbf{X}^{(\cdot i_2 \cdot)} = \mathbf{A}^{(3)} D_{i_2}\left(\mathbf{A}^{(2)}\right) \mathbf{A}^{(1)T}$$

and

$$\mathbb{C}^{=I_2 I_3 \times I_1} \ni \mathbf{X}_{(2)} \triangleq \begin{bmatrix} \mathbf{X}^{(\cdot 1 \cdot)} \\ \vdots \\ \mathbf{X}^{(\cdot I_2 \cdot)} \end{bmatrix} = \begin{bmatrix} \mathbf{A}^{(3)} D_1\left(\mathbf{A}^{(2)}\right) \\ \vdots \\ \mathbf{A}^{(3)} D_{I_2}\left(\mathbf{A}^{(2)}\right) \end{bmatrix} \mathbf{A}^{(1)T} = \left(\mathbf{A}^{(2)} \odot \mathbf{A}^{(3)}\right) \mathbf{A}^{(1)T}.$$

2. Essential Uniqueness of CPD with a Semi-unitary Matrix Factor. The CPD of $\mathcal{X}$ is said to be essentially unique if all the N-tuplets $\left(\overline{\mathbf{A}}^{(1)}, \ldots, \overline{\mathbf{A}}^{(N)}\right)$ satisfying (1.1) are related via

$$\overline{\mathbf{A}}^{(n)} = \mathbf{A}^{(n)} \Pi \Delta_{\mathbf{A}^{(n)}}, \quad \forall n \in [1, N],$$

where Π is a permutation matrix and $\{\Delta_{\mathbf{A}^{(n)}}\}$ are diagonal matrices satisfying $\prod_{n=1}^{N} \Delta_{\mathbf{A}^{(n)}} = \mathbf{I}_R$. A sufficient condition for essential uniqueness of the CPD of $\mathcal{X}$ is

$$\sum_{n=1}^{N} \mathrm{k}\left(\mathbf{A}^{(n)}\right) \geq 2R + (N-1), \tag{2.1}$$

where R denotes the rank of the tensor [21], [29]. However, this condition is not necessary [31], [13], [8]. For instance, in [13] it was shown that generically a third-order CPD is essentially unique when

$$2R(R-1) \leq I_1(I_1-1)I_2(I_2-1) \quad \text{and} \quad R \leq I_3. \tag{2.2}$$

The paper [13] also contains an algebraic (non-generic) version of this condition. When the matrix factor $\mathbf{A}^{(3)}$ is semi-unitary, then it also has full column rank ($R \leq I_3$). Hence, the generic essential uniqueness condition (2.2) is also valid for a CPD with a semi-unitary matrix factor.

We will now provide a new uniqueness condition for a complex CPD with a semi-unitary matrix factor. To simplify the discussion, let us consider the third order tensor $\mathcal{X} \in \mathbb{C}^{I_1 \times I_2 \times I_3}$ with matrix representation

$$\mathbb{C}^{I_1 \times I_2} \ni \mathbf{X}^{(i_1 \cdot \cdot)} = \mathbf{A}^{(2)} D_{i_1}\left(\mathbf{A}^{(1)}\right) \mathbf{A}^{(3)T}, \quad \forall i_3 \in [1, I_3],$$

where $\mathbf{A}^{(3)}$ is assumed to be semi-unitary. Since $\mathbf{A}^{(3)}$ is semi-unitary we can construct the fourth order tensor $\mathcal{Y} \in \mathbb{C}^{I_2 \times I_2 \times I_1 \times I_1}$ with matrix representations

$$\begin{aligned} \mathbb{C}^{I_2 \times I_2} \ni \mathbf{Y}^{(\cdot \cdot i_3, i_4)} &= \mathbf{X}^{(i_3 \cdot \cdot)} \mathbf{X}^{(i_4 \cdot \cdot)H} \\ &= \mathbf{A}^{(2)} D_{i_3}\left(\mathbf{A}^{(1)}\right) D_{i_4}\left(\mathbf{A}^{(1)*}\right) \mathbf{A}^{(2)H}, \quad \forall i_3, i_4 \in [1, I_1], \end{aligned}$$

and $\mathcal{Y}(:, :, i_3, i_4) = \mathbf{Y}^{(\cdot \cdot i_3, i_4)}$, $\forall i_3, i_4 \in [1, I_1]$. We have

$$\mathrm{Vec}\left(\mathbf{Y}^{(\cdot \cdot i_3, i_4)}\right) = \left(\mathbf{A}^{(2)*} \odot \mathbf{A}^{(2)}\right) \mathrm{Vecd}\left(D_{i_3}\left(\mathbf{A}^{(1)}\right) D_{i_4}\left(\mathbf{A}^{(1)*}\right)\right)$$

and

$$\begin{aligned}\mathbf{Y} &= \left[\mathrm{Vec}\left(\mathbf{Y}^{(\cdot\cdot 1,1)}\right), \mathrm{Vec}\left(\mathbf{Y}^{(\cdot\cdot 1,2)}\right), \cdots, \mathrm{Vec}\left(\mathbf{Y}^{(\cdot\cdot I_1,I_1)}\right)\right] \\ &= \left(\mathbf{A}^{(2)*} \odot \mathbf{A}^{(2)}\right)\left(\mathbf{A}^{(1)*} \odot \mathbf{A}^{(1)}\right)^T. \end{aligned} \tag{2.3}$$

The matrix $\mathbf{A}^{(n)*} \odot \mathbf{A}^{(n)}$ has generically full column rank when $I_n^2 \geq R$ [27]. Let $\mathbf{Y} = \mathbf{U}\Sigma\mathbf{V}^H$ denote the compact SVD of $\mathbf{Y}$ and assume that $I_1^2, I_2^2 \geq R$, then there generically exists a nonsingular matrix $\mathbf{F} \in \mathbb{C}^{R\times R}$ such that

$$\left(\mathbf{A}^{(2)*} \odot \mathbf{A}^{(2)}\right)\mathbf{F}^T = \mathbf{U}. \tag{2.4}$$

From the matrix representation (2.4) of a third order CPD and the generic uniqueness condition (2.2) we conclude that a complex third-order CPD with semi-unitary matrix factor $\mathbf{A}^{(3)}$ is also generically unique when

$$2R(R-1) \leq I_2^2(I_2-1)^2, \quad R \leq I_1^2 \quad \text{and} \quad R \leq I_3. \tag{2.5}$$

This result can be useful when I_1 is small and R is large. As an example, let $R = 7$, $I_1 = 3$ and $I_3 \geq R$, then condition (2.2) requires that $I_2 \geq 5$ while condition (2.5) only requires that $I_2 \geq 4$.

In order to compute the CPD of a tensor $\mathcal{X}$ satisfying the condition (2.5) we can first compute $\mathbf{A}^{(2)}$ from (2.4) by the use of the unconstrained SD-CP method which will be reviewed in section 3. Next, we find $\mathbf{A}^{(1)}$ from (2.3) via a set of R decoupled best rank-1 matrix approximation problems similar to what we will explain in subsection 3.2. Finally, we can obtain the semi-unitary matrix factor $\mathbf{A}^{(3)}$ by solving a unitary Procrustes-type problem as will also be explained in subsection 3.2. In the rest of the paper we will find the matrix factors directly from $\mathcal{X}$, instead of working with $\mathcal{Y}$.

3. SD-CPO: SD-CP with a semi-unitary matrix factor. In this section we explain how a semi-unitary constraint can be incorporated in the SD-CP method presented in [13]. The approach will be referred to as SD-CPO. The SD-CP method was originally developed to solve highly underdetermined mixture identification problems [14], [15]. Let $\mathcal{X} \in \mathbb{C}^{I_1\times I_2\times I_3}$ be a rank-R tensor with matrix representation

$$\mathbb{C}^{I_1 I_2\times I_3} \ni \mathbf{X}_{(1)} = \left(\mathbf{A}^{(1)} \odot \mathbf{A}^{(2)}\right)\mathbf{A}^{(3)T}. \tag{3.1}$$

Assume that the matrix $\mathbf{A}^{(1)} \odot \mathbf{A}^{(2)}$ has full column rank and that $\mathbf{A}^{(3)}$ is semi-unitary. Let $\mathbf{X}_{(1)} = \mathbf{U}\Sigma\mathbf{V}^H$ denote the compact SVD of $\mathbf{X}_{(1)}$, where $\mathbf{U} \in \mathbb{C}^{I_1 I_2\times R}$, $\mathbf{V} \in \mathbb{C}^{I_3\times R}$ are semi-unitary matrices and $\Sigma \in \mathbb{C}^{R\times R}$ is a positive diagonal matrix. Since $\mathrm{Col}\,(\mathbf{U}\Sigma) = \mathrm{Col}\left(\left(\mathbf{A}^{(1)} \odot \mathbf{A}^{(2)}\right)\mathbf{A}^{(3)T}\right) = \mathrm{Col}\left(\mathbf{A}^{(1)} \odot \mathbf{A}^{(2)}\right)$, there exists a nonsingular matrix $\mathbf{F} \in \mathbb{C}^{R\times R}$ such that

$$\mathbf{A}^{(1)} \odot \mathbf{A}^{(2)} = \mathbf{U}\Sigma\mathbf{F}. \tag{3.2}$$

Together with the relation $\mathbf{X}_{(1)} = \left(\mathbf{A}^{(1)} \odot \mathbf{A}^{(2)}\right)\mathbf{A}^{(3)T} = \mathbf{U}\Sigma\mathbf{V}^H$, this implies that

$$\mathbf{A}^{(3)T} = \mathbf{F}^{-1}\mathbf{V}^H. \tag{3.3}$$

In order to find $\mathbf{F}$ the SD-CP method was proposed in [13]. It was shown that, when $2R(R-1) \leq I_1(I_1-1)I_2(I_2-1)$ and $R \leq I_3$, there generically exist R complex symmetric

matrices $\mathbf{M}^{(r)} \in \mathbb{C}^{R\times R}$ and R diagonal matrices $\Lambda_r \in \mathbb{C}^{R\times R}$ satisfying

$$\begin{aligned} \mathbf{M}^{(1)} &= \mathbf{F}\Lambda_1\mathbf{F}^T \\ &\vdots \\ \mathbf{M}^{(R)} &= \mathbf{F}\Lambda_R\mathbf{F}^T. \end{aligned} \tag{3.4}$$

For the procedure to compute the matrices $\{\mathbf{M}^{(r)}\}$ we refer to [13]. The paper [13] also contains an algebraic (non-generic) version of this result.

When $\mathbf{A}^{(3)}$ is constrained to be semi-unitary we get $\mathbf{V}^H\mathbf{V} = \mathbf{F}\mathbf{A}^{(3)T}\mathbf{A}^{(3)*}\mathbf{F}^H = \mathbf{I}_R \Leftrightarrow \mathbf{F}\mathbf{F}^H = \mathbf{I}_R$ and hence $\mathbf{F}$ is a unitary matrix. This means that when the matrix factor $\mathbf{A}^{(3)}$ is semi-unitary, the SD-CP problem has been converted to a *unitary* Simultaneous Matrix Diagonalization (SMD) problem. Subsection 3.1 will discuss how the unitary matrix $\mathbf{F}$ can be computed from the matrices $\{\mathbf{M}^{(r)}\}$.

3.1. Computation of F. The unitary matrix $\mathbf{F}$ is equal to the joint diagonalizer in (3.4). In order to jointly diagonalize the set of complex symmetric matrices $\{\mathbf{M}^{(r)}\}$ in (3.4) by the unitary matrix $\mathbf{F}$, a Jacobi-type iteration similar to the JADE algorithm [3], [4] will be developed. Equation (3.4) can be interpreted as a simultaneous Takagi factorization [18]. The simultaneous Takagi factorization problem also pops up in the blind separation of non-circular sources [11]. Similarly to JADE, we determine the unitary matrix $\mathbf{F}$ that maximizes the objective function

$$f(\mathbf{F}) = \sum_{r=1}^{R} \left\| \operatorname{diag}\left(\mathbf{F}^H\mathbf{M}^{(r)}\mathbf{F}^*\right)\right\|_F^2.$$

This is done by means of a Jacobi-type algorithm. The idea behind the Jacobi procedure is that any unitary matrix $\mathbf{F} \in \mathbb{C}^{R\times R}$ with determinant equal to one can be parameterized as

$$\mathbf{F} = \prod_{p=1}^{R-1}\prod_{q=p+1}^{R} \mathbf{F}[p,q],$$

where $\mathbf{F}[p,q]$ is a Givens rotation matrix, defined by

$$\mathbf{F}[p,q]_{kl} = \begin{cases} 1 & \text{if } k = l \text{ and } k \notin \{p,q\} \\ \cos(\theta) & \text{if } k = l = p \\ \cos(\theta) & \text{if } k = l = q \\ \sin(\theta)e^{i\phi} & \text{if } k = p \text{ and } l = q \\ -\sin(\theta)e^{-i\phi} & \text{if } k = q \text{ and } l = p \\ 0 & \text{otherwise} \end{cases}$$

with $\theta, \phi \in \mathbb{R}$. An outline of the sweeping procedure for the simultaneous Takagi factorization is presented as Algorithm 1.

Let $\mathbf{F} = \mathbf{F}[p,q]$, then a technical derivation yields:

$$\begin{aligned} f(\mathbf{F}[p,q]) &= \sum_{r=1}^{R} \left\| \operatorname{diag}\left(\mathbf{F}[p,q]^H\mathbf{M}^{(r)}\mathbf{F}[p,q]^*\right)\right\|_F^2 \\ &= \mathbf{x}^T \begin{bmatrix} \alpha_1 & \beta_1/2 & \beta_2/2 \\ \beta_1/2 & \alpha_1/2 + \alpha_4/2 + \operatorname{Re}\{\alpha_5\} & \operatorname{Im}\{\alpha_5\} \\ \beta_2/2 & \operatorname{Im}\{\alpha_5\} & \alpha_1/2 + \alpha_4/2 - \operatorname{Re}\{\alpha_5\} \end{bmatrix} \mathbf{x} \\ &= \mathbf{x}^T\mathbf{B}\mathbf{x}, \end{aligned}$$

Algorithm 1 Outline of sweeping procedure for simultaneous Takagi factorization.

Initialize: $\mathbf{F}$
Repeat until convergence
for $p = 1$ to $R - 1$ **do**
  for $q = p + 1$ to R **do**
    calculate optimal $\mathbf{F}[p, q]$ (subsection 3.1)
    $\mathbf{F} \leftarrow \mathbf{F}\,\mathbf{F}[p, q]$
    $\mathbf{M}^{(r)} \leftarrow \mathbf{F}\,[p, q]^H\, \mathbf{M}^{(r)} \mathbf{F}\,[p, q]^*$
  end for
end for

where

$$\mathbf{x} = \left[\cos(2\theta), \sin(2\theta)\cos\left(\phi\right), \sin(2\theta)\sin\left(\phi\right)\right]^T$$

and

$$\alpha_1 = \sum_{r=1}^{R} \left|\mathbf{M}_{pp}^{(r)}\right|^2 + \left|\mathbf{M}_{qq}^{(r)}\right|^2 \qquad \alpha_2 = \sum_{r=1}^{R} \mathbf{M}_{qq}^{(r)*}\left(\mathbf{M}_{pq}^{(r)} + \mathbf{M}_{qp}^{(r)}\right)$$
$$\alpha_3 = \sum_{r=1}^{R} \mathbf{M}_{pp}^{(r)}\left(\mathbf{M}_{pq}^{(r)} + \mathbf{M}_{qp}^{(r)}\right)^* \quad \alpha_4 = \sum_{r=1}^{R} \left|\mathbf{M}_{pq}^{(r)} + \mathbf{M}_{qp}^{(r)}\right|^2$$
$$\alpha_5 = \sum_{r=1}^{R} \mathbf{M}_{qq}^{(r)*}\mathbf{M}_{pp}^{(r)}$$
$$\beta_1 = \mathrm{Re}\,\{\alpha_2\} + \mathrm{Re}\,\{\alpha_3\} \qquad \beta_2 = \mathrm{Im}\{\alpha_2\} - \mathrm{Im}\{\alpha_3\}.$$

Note that the unknown $\mathbf{x}$ that characterizes the Givens rotation has unit norm. The function $f\,(\mathbf{F}\,[p, q])$ can be maximized by taking $\mathbf{x}$ equal to the dominant eigenvector of $\mathbf{B}$. Hence, the problem reduces to finding the eigenvector that corresponds to the largest eigenvalue of the above matrix $\mathbf{B}$.

3.2. Computation of $\mathbf{A}^{(1)}$, $\mathbf{A}^{(2)}$ and $\mathbf{A}^{(3)}$. Assume that the unitary matrix $\mathbf{F}$ has been found, then due to relation (3.2) the matrices $\mathbf{A}^{(1)}$ and $\mathbf{A}^{(2)}$ can be found from a set of decoupled best rank-1 matrix approximation problems. Indeed, let $\mathbf{G} = \mathbf{U}\Sigma\mathbf{F}$, then

$$\left\|\mathrm{Unvec}\left(\mathbf{g}_r\right) - \mathrm{Unvec}\left(\mathbf{a}_r^{(1)} \otimes \mathbf{a}_r^{(2)}\right)\right\|_F^2 = \left\|\mathrm{Unvec}\left(\mathbf{g}_r\right) - \mathbf{a}_r^{(2)}\mathbf{a}_r^{(1)T}\right\|_F^2, \quad r \in [1, R],$$

where $\mathbf{g}_r$ and $\mathbf{a}_r^{(n)}$ denote the rth column of $\mathbf{G}$ and $\mathbf{A}^{(n)}$, respectively. Consequently, we can take $\mathbf{a}_r^{(1)} = \mathbf{v}_1^*$ and $\mathbf{a}_r^{(2)} = \sigma_1\mathbf{u}_1$, where σ_1 denotes the largest singular value of $\mathrm{Unvec}\left(\tilde{\mathbf{f}}_r\right)$ and $\mathbf{u}_1$ and $\mathbf{v}_1$ denote its dominant left and right singular vectors, respectively.

The matrix $\mathbf{A}^{(3)}$ can be found as the semi-unitary minimizer of the cost function

$$f\left(\mathbf{A}^{(3)}\right) = \left\|\mathbf{X}_{(1)} - \left(\mathbf{A}^{(1)} \odot \mathbf{A}^{(2)}\right)\mathbf{A}^{(3)T}\right\|_F^2. \tag{3.5}$$

The minimizer of (3.5) is equal to the minimizer of

$$g\left(\mathbf{A}^{(3)}\right) = \left\|\left(\mathbf{A}^{(1)} \odot \mathbf{A}^{(2)}\right)^H \mathbf{X}_{(1)} - \mathbf{A}^{(3)T}\right\|_F^2. \tag{3.6}$$

Let $\left(\mathbf{A}^{(1)} \odot \mathbf{A}^{(2)}\right)^H \mathbf{X}_{(1)} = \mathbf{U}\Sigma\mathbf{V}^H$ denote the SVD of $\left(\mathbf{A}^{(1)} \odot \mathbf{A}^{(2)}\right)^H \mathbf{X}_{(1)}$, then the optimal semi-unitary $\mathbf{A}^{(3)}$ is $\mathbf{A}^{(3)} = \mathbf{V}(:, 1:R)^* \mathbf{U}(:, 1:R)^T$. This can be understood as a variant of the unitary Procrustes problem [18] for the semi-unitary case.

In the presence of noise a refinement of the estimates is often necessary. This can be done by taking a few ALS steps, as explained in subsection 5.1.

3.3. SD-CPO for Nth-Order Tensors. In this subsection we generalize the above outlined method for computing a third order CPD with a semi-unitary matrix factor to the case of computing a Nth order CPD with a semi-unitary matrix factor. Consider $\mathcal{X} \in \mathbb{C}^{I_1 \times \cdots \times I_N}$ with rank R and matrix representation $\mathbf{X}_{[N]} = \left(\mathbf{A}^{(1)} \odot \cdots \odot \mathbf{A}^{(N-1)}\right) \mathbf{A}^{(N)T}$. Assume that the matrix $\mathbf{A}^{(1)} \odot \cdots \odot \mathbf{A}^{(N-1)}$ has full column rank and that $\mathbf{A}^{(N)}$ is semi-unitary. Let $\mathbf{X}_{[N]} = \mathbf{U}\Sigma\mathbf{V}^H$ denote the compact SVD of $\mathbf{X}_{[N]}$, then there exists a unitary matrix $\mathbf{F} \in \mathbb{C}^{R \times R}$ such that $\mathbf{A}^{(1)} \odot \cdots \odot \mathbf{A}^{(N-1)} = \mathbf{U}\Sigma\mathbf{F}$ and $\mathbf{A}^{(N)} = \mathbf{V}^*\mathbf{F}^*$. As mentioned in [13], the SD-CP method can be generalized to tensors of order higher than three, which means that that the Nth-order CPD problem can be converted to a SMD problem of the form (3.4). Thus, the unitary matrix $\mathbf{F}$ follows from solving a simultaneous Takagi factorization problem of the form (3.4) in the way explained in subsection 3.1. Next, the unconstrained matrix factors follow from the R decoupled rank-1 tensor approximation problems

$$\mathbf{U}\Sigma\mathbf{F} = \mathbf{A}^{(1)} \odot \cdots \odot \mathbf{A}^{(N-1)}. \tag{3.7}$$

The Khatri-Rao product indicates that every column of this matrix is the vector representation of a rank-1 term. In order to find the rank-1 terms we can apply the ALS method presented in [9]. Finally, the semi-unitary matrix $\mathbf{A}^{(N)}$ follows from a unitary Procrustes problem as explained above.

In the SD-CP method the most expensive step is the computation of the symmetric matrices $\{\mathbf{M}^{(r)}\}$. In order to reduce the computational burden, we present a SD-CP method that does not fully exploit the structure of the problem, but is must faster than the original SD-CP method. Let $\mathbb{C}^{\prod_{n=2}^{R} I_n \times R} \ni \mathbf{D} = \mathbf{A}^{(2)} \odot \cdots \odot \mathbf{A}^{(N-1)}$, then

$$\mathbf{X}_{[N]} = \left(\mathbf{A}^{(1)} \odot \mathbf{D}\right) \mathbf{A}^{(N)T}. \tag{3.8}$$

If we ignore the structure of $\mathbf{D}$, then we can treat (3.8) as a third-order CPD problem with a semi-unitary matrix factor $\mathbf{A}^{(N)}$. Hence, we first compute $\mathbf{A}^{(1)}$, $\mathbf{D}$ and $\mathbf{A}^{(N)}$ via the procedures described in subsection 3.1 and 3.2. Next, we compute the remaining matrix factors from the decoupled best rank-1 tensor approximation problems

$$\mathbf{D} = \mathbf{A}^{(2)} \odot \cdots \odot \mathbf{A}^{(N-1)}. \tag{3.9}$$

When $N = 4$, then the solution to (3.9) follows from R decoupled rank-1 matrix approximation problems.

4. SGSD-CPO: SGSD-CP with a semi-unitary matrix factor. In this section we explain how a semi-unitary constraint can be incorporated in the SGSD-CP method presented in [32], [12], [25] for computing a third-order CPD and the approach will be referred to as SGSD-CPO. The SGSD-CP method was originally developed to solve a SMD problem involving a joint congruence transformation in the context of blind separation of communication signals [32]. Consider the tensor $\mathcal{X} \in \mathbb{C}^{I_1 \times I_2 \times I_3}$ of rank R with matrix representations

$$\mathbf{X}^{(i_1 \cdot \cdot)} = \mathbf{A}^{(2)} D_{i_1}\left(\mathbf{A}^{(1)}\right) \mathbf{A}^{(3)T} \tag{4.1}$$

and

$$\mathbf{X}_{(1)} = \left(\mathbf{A}^{(1)} \odot \mathbf{A}^{(2)}\right)\mathbf{A}^{(3)T}.$$

The SGSD-CP method assumes that two of the matrix factors, say $\mathbf{A}^{(2)}$ and $\mathbf{A}^{(3)}$, have full column rank. In this section we assume that $\mathbf{A}^{(3)}$ is semi-unitary. Let $\mathbf{X}_{(1)} = \mathbf{U}\Sigma\mathbf{V}^H$ denote the compact SVD of $\mathbf{X}_{(1)}$, then it was shown in section 3 that there exists a unitary matrix $\mathbf{F}$ such that $\mathbf{A}^{(3)} = \mathbf{V}^*\mathbf{F}^*$.

The first step of the SGSD-CP method is to compute $\mathbf{V}$ from the compact SVD of $\mathbf{X}_{(1)}$ and to compute $\widetilde{\mathbf{X}}^{(i_1\cdot\cdot)} = \mathbf{X}^{(i_1\cdot\cdot)}\mathbf{V} = \mathbf{A}^{(2)}D_{i_1}\left(\mathbf{A}^{(1)}\right)\mathbf{F}^H$. Let $\mathbf{A}^{(2)} = \mathbf{Q}\mathbf{R}$ be the QR factorization of $\mathbf{A}^{(2)}$ where $\mathbf{Q} \in \mathbb{C}^{I_2\times I_2}$ is a unitary matrix and $\mathbf{R} \in \mathbb{C}^{I_2\times R}$ is an upper triangular matrix. From (4.1) we have that the unknowns can be computed by minimizing the cost function

$$\begin{aligned} f\left(\mathbf{Q},\mathbf{R},\mathbf{A}^{(1)},\mathbf{F}\right) &= \sum_{i_1=1}^{I_1}\left\|\widetilde{\mathbf{X}}^{(i_1\cdot\cdot)} - \mathbf{Q}\mathbf{R}D_{i_1}\left(\mathbf{A}^{(1)}\right)\mathbf{F}^H\right\|_F^2 \\ &= \sum_{i_1=1}^{I_1}\left\|\mathbf{Q}^H\widetilde{\mathbf{X}}^{(i_1\cdot\cdot)}\mathbf{F} - \mathbf{R}D_{i_1}\left(\mathbf{A}^{(1)}\right)\right\|_F^2. \end{aligned} \tag{4.2}$$

Note that the matrices $\mathbf{R}D_{i_1}\left(\mathbf{A}^{(1)}\right)$ in (4.2) are upper triangular. It can be shown that we can relax the structure of the upper triangular matrices $\{\mathbf{R}D_{i_1}\left(\mathbf{A}^{(1)}\right)\}$ and just consider them as independent upper triangular matrices without losing the essential uniqueness of the solution, see [25] for a further discussion. Hence, in the second step of the SGSD-CP method we compute the unitary matrices $\mathbf{Q}$ and $\mathbf{F}$ that make the matrices $\{\widetilde{\mathbf{X}}^{(i_1\cdot\cdot)}\}$ as upper triangular as possible by maximizing

$$g\left(\mathbf{Q},\mathbf{F}\right) = \sum_{i_1=1}^{I_1}\left\|\mathrm{triu}\left(\mathbf{Q}^H\widetilde{\mathbf{X}}^{(i_1\cdot\cdot)}\mathbf{F}\right)\right\|_F^2. \tag{4.3}$$

For the computation of $\mathbf{Q}$ and $\mathbf{F}$, we can resort to the extended QZ method developed in [32]. The discussion in [32] was limited to the case $I_2 = I_3 = R$. We will in subsection 4.1 generalize the algorithm to the case $I_2, I_3 \geq R$.

4.1. Computation of Q and F. In this subsection we generalize the extended QZ method from $I_2 = I_3 = R$ [32] to the case $I_2, I_3 \geq R$. Let $\widetilde{\mathbf{X}}^{(i_1\cdot\cdot)} \in \mathbb{C}^{I_2\times I_3}$, $i_1 \in [1, I_1]$, be the matrices that we try to make as upper triangular as possible by maximizing

$$h\left(\mathbf{Q},\mathbf{F}\right) = \sum_{i_1=1}^{I_1}\left\|\mathrm{triu}_R\left(\mathbf{Q}^H\widetilde{\mathbf{X}}^{(i_1\cdot\cdot)}\mathbf{F}\right)\right\|_F^2.$$

The unitary matrices $\mathbf{Q}$ and $\mathbf{F}$ are generated as

$$\mathbf{Q} = \prod_{s=1}^{S}\prod_{r=0}^{I_2-1}\mathbf{Q}[r,s] \quad , \quad \mathbf{F} = \prod_{s=1}^{S}\prod_{r=0}^{I_3-1}\mathbf{F}[r,s],$$

where S denotes the number of executed sweeps,

$$\mathbf{Q}[r,s]^H = \begin{bmatrix} \mathbf{I}_{r-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{U}_s \end{bmatrix} \in \mathbb{C}^{I_2 \times I_2} \quad , \quad \mathbf{F}[r,s] = \begin{bmatrix} \mathbf{A}_s & \mathbf{0} & \mathbf{C}_s \\ \mathbf{0} & \mathbf{I}_{R-r} & \mathbf{0} \\ \mathbf{B}_s & \mathbf{0} & \mathbf{D}_s \end{bmatrix} \in \mathbb{C}^{I_3 \times I_3},$$

$\mathbf{U}_s \in \mathbb{C}^{(I_2-r+1)\times(I_2-r+1)}$, $\mathbf{A}_s \in \mathbb{C}^{r\times r}$, $\mathbf{B}_s \in \mathbb{C}^{(I_3-R)\times r}$, $\mathbf{C}_s \in \mathbb{C}^{r\times(I_3-R)}$ and $\mathbf{D}_s \in \mathbb{C}^{(I_3-R)\times(I_3-R)}$. In the extended QZ method the unitary matrices $\mathbf{Q}$ and $\mathbf{F}$ are updated sequentially. The updating rules are $\widetilde{\mathbf{X}}^{(i_1\cdot\cdot)} \leftarrow \mathbf{Q}[r,s]^H \widetilde{\mathbf{X}}^{(i_1\cdot\cdot)}$, $\mathbf{Q} \leftarrow \mathbf{Q}\,\mathbf{Q}[r,s]$ and $\widetilde{\mathbf{X}}^{(i_1\cdot\cdot)} \leftarrow \widetilde{\mathbf{X}}^{(i_1\cdot\cdot)}\mathbf{F}[r,s]$, $\mathbf{F} \leftarrow \mathbf{F}\,\mathbf{F}[r,s]$.

The unitary matrix $\mathbf{Q}[r,s]^H$ is designed to minimize the elements below the diagonal entry of the rth column of the matrices $\{\widetilde{\mathbf{X}}^{(i_1\cdot\cdot)}\}$. Let

$$\overline{\mathbf{X}}^{(r)} = \left[\widetilde{\mathbf{X}}^{(1\cdot\cdot)}(r:I_2,r), \cdots, \widetilde{\mathbf{X}}^{(I_1\cdot\cdot)}(r:I_2,r)\right] \in \mathbb{C}^{(I_2-r+1)\times I_1}, \quad r \in [1, \min(R, I_2-1)],$$

and let $\overline{\mathbf{X}}^{(r)} = \overline{\mathbf{U}}\,\overline{\Sigma}\,\overline{\mathbf{V}}^H$ denote the compact SVD of $\overline{\mathbf{X}}^{(r)}$. A unitary matrix $\mathbf{U}_s$ that minimizes the Frobenius norm of $\left(\mathbf{U}_s\overline{\mathbf{X}}^{(r)}\right)(2:I_2-r+1,:)$ is $\mathbf{U}_s = \overline{\mathbf{U}}^H$.

Similarly, the unitary matrix $\mathbf{F}[r,s]$ is designed to minimize every element of the vectors $\{\widetilde{\mathbf{X}}^{(i_1\cdot\cdot)}(r,1:r-1)\}$ and $\{\widetilde{\mathbf{X}}^{(i_1\cdot\cdot)}(r,R+1:I_3)\}$. Let

$$\widehat{\mathbf{X}}^{(r)} = \begin{bmatrix} \widetilde{\mathbf{X}}^{(1\cdot\cdot)}(r,1:r)\,,\, \widetilde{\mathbf{X}}^{(1\cdot\cdot)}(r,R+1:I_3) \\ \vdots \\ \widetilde{\mathbf{X}}^{(I_1\cdot\cdot)}(r,1:r)\,,\, \widetilde{\mathbf{X}}^{(I_1\cdot\cdot)}(r,R+1:I_3) \end{bmatrix} \in \mathbb{C}^{I_1\times(r+I_3-R)}, \quad r \in [2-\mathrm{sign}(I_3-R)\,,R],$$

and let $\widehat{\mathbf{X}}^{(r)} = \widehat{\mathbf{U}}\,\widehat{\Sigma}\,\widehat{\mathbf{V}}^H$ denote the compact SVD of $\widehat{\mathbf{X}}^{(r)}$. A unitary matrix

$$\mathbf{F}_s = \begin{bmatrix} \mathbf{V}_s & \mathbf{X}_s \\ \mathbf{W}_s & \mathbf{Y}_s \end{bmatrix}$$

that minimizes the Frobenius norm of $\left(\widehat{\mathbf{X}}^{(r)}\begin{bmatrix} \mathbf{V}_s & \mathbf{X}_s \\ \mathbf{W}_s & \mathbf{Y}_s \end{bmatrix}\right)(:,[1:r-1,r+1:r+I_3-R])$ is $\mathbf{F}_s = \left[\widehat{\mathbf{V}}(:,2:r), \widehat{\mathbf{V}}(:,1), \widehat{\mathbf{V}}(:,r+1:r+I_3-R)\right]$.

An outline of the generalized version of the extended QZ procedure is presented as Algorithm 2.

4.2. Computation of $\mathbf{A}^{(1)}$, $\mathbf{A}^{(2)}$ and $\mathbf{A}^{(3)}$. Assume that the matrices $\mathbf{Q}$ and $\mathbf{F}$ have been found, then we obtain $\mathbf{A}^{(3)} = \mathbf{V}^*\mathbf{F}^*$. The upper triangular matrix $\mathbf{R}$ and the diagonal matrices $\{D_{i_1}\left(\mathbf{A}^{(1)}\right)\}$ can now be found from a set of decoupled best rank-1 matrix approximation problems, as discussed next. This is an important difference with the unconstrained case. Let $\mathbf{R}^{(i_1\cdot\cdot)} = \mathrm{triu}\left(\mathbf{Q}^H\widetilde{\mathbf{X}}^{(i_1\cdot\cdot)}\mathbf{F}\right)$, then the problem

Algorithm 2 Outline of the extended QZ method when $I_2, I_3 \geq R$.

Initialize: $\mathbf{Q}$ and $\mathbf{F}$
Repeat until convergence
for $r = 1$ to $\min(R, I_2 - 1)$ **do**
$\overline{\mathbf{X}}^{(r)} = \left[\widetilde{\mathbf{X}}^{(1\cdot\cdot)}(r : I_2, r), \cdots, \widetilde{\mathbf{X}}^{(I_1\cdot\cdot)}(r : I_2, r)\right]$
$\overline{\mathbf{U}}\,\overline{\Sigma}\,\overline{\mathbf{V}}^H = \mathrm{svd}\left(\overline{\mathbf{X}}^{(r)}\right)$
$\widetilde{\mathbf{X}}^{(i_1\cdot\cdot)}(r : I_2, :) \leftarrow \overline{\mathbf{U}}^H \widetilde{\mathbf{X}}^{(i_1\cdot\cdot)}(r : I_2, :)$
$\mathbf{Q}(:, r : I_2) \leftarrow \mathbf{Q}(:, r : I_2)\overline{\mathbf{U}}$
end for
for $r = R$ to $\max(1, R - I_3 + 2)$ **do**
$$\widehat{\mathbf{X}}^{(r)} = \begin{bmatrix} \widetilde{\mathbf{X}}^{(1\cdot\cdot)}(r, 1 : r), \widetilde{\mathbf{X}}^{(1\cdot\cdot)}(r, R+1 : I_3) \\ \vdots \\ \widetilde{\mathbf{X}}^{(I_1\cdot\cdot)}(r, 1 : r), \widetilde{\mathbf{X}}^{(I_1\cdot\cdot)}(r, R+1 : I_3) \end{bmatrix}$$
$\widehat{\mathbf{U}}\,\widehat{\Sigma}\,\widehat{\mathbf{V}}^H = \mathrm{svd}\left(\widehat{\mathbf{X}}^{(r)}\right)$
$\widehat{\mathbf{V}} \leftarrow \left[\widehat{\mathbf{V}}(:, 2 : r), \widehat{\mathbf{V}}(:, 1), \widehat{\mathbf{V}}(:, r+1 : r+I_3-R)\right]$
$\widetilde{\mathbf{X}}^{(i_1\cdot\cdot)}(:, [1 : r, R+1 : I_3]) \leftarrow \widetilde{\mathbf{X}}^{(i_1\cdot\cdot)}(:, [1 : r, R+1 : I_3])\widehat{\mathbf{V}}$
$\mathbf{F}(:, [1 : r, R+1 : I_3]) \leftarrow \mathbf{F}(:, [1 : r, R+1 : I_3])\widehat{\mathbf{V}}$
end for

of computing $\mathbf{R}$ and $\{D_{i_1}\left(\mathbf{A}^{(1)}\right)\}$ can be formulated as the minimization of

$$\begin{aligned} g\left(\mathbf{R}, \mathbf{A}^{(1)}\right) &= \sum_{i_1=1}^{I_1} \left\| \mathbf{R}^{(i_1\cdot\cdot)} - \mathbf{R} D_{i_1}\left(\mathbf{A}^{(1)}\right) \right\|_F^2 \\ &= \sum_{r=1}^{R} \sum_{i_1=1}^{I_1} \left\| \mathbf{r}_r^{(i_1)} - \mathbf{r}_r d_r^{(i_1)} \right\|_F^2 \\ &= \sum_{r=1}^{R} \left\| \left[\mathbf{r}_r^{(1)}, \cdots, \mathbf{r}_r^{(I_1)}\right] - \mathbf{r}_r \left[d_r^{(1)}, \cdots, d_r^{(I_1)}\right] \right\|_F^2 \\ &= \sum_{r=1}^{R} \left\| \overline{\mathbf{R}}^{(r)} - \mathbf{r}_r \overline{\mathbf{d}}^{(r)T} \right\|_F^2, \end{aligned} \tag{4.4}$$

where $\mathbf{r}_r^{(i_1)} \in \mathbb{C}^r$ is the rth column of the upper triangular part of $\mathbf{R}^{(i_1\cdot\cdot)}$, $\mathbf{r}_r \in \mathbb{C}^r$ is the rth column of the upper triangular part of $\mathbf{R}$, $d_r^{(i_1)} \in \mathbb{C}$ is the rth diagonal entry of the diagonal matrix $D_{i_1}\left(\mathbf{A}^{(1)}\right)$ and

$$\begin{aligned} \overline{\mathbf{R}}^{(r)} &= \left[\mathbf{r}_r^{(1)}, \cdots, \mathbf{r}_r^{(I_1)}\right] \in \mathbb{C}^{r \times I_1} \\ \overline{\mathbf{d}}^{(r)} &= \left[d_r^{(1)}, \cdots, d_r^{(I_1)}\right]^T \in \mathbb{C}^{I_1 \times 1}. \end{aligned}$$

From (4.4) it follows that we can take $\mathbf{r}_r = \sigma_1 \mathbf{u}_1$ and $\overline{\mathbf{d}}^{(r)} = \mathbf{v}_1^*$, where σ_1 denotes the largest singular value of $\overline{\mathbf{R}}^{(r)}$ and $\mathbf{u}_1$ and $\mathbf{v}_1$ denote its dominant left and right singular

vector, respectively. The matrix factor $\mathbf{A}^{(1)}$ follows directly from the minimizer of (4.4) while $\mathbf{A}^{(2)} = \mathbf{QR}$.

In the presence of noise a refinement of $\mathbf{A}^{(1)}$, $\mathbf{A}^{(2)}$ and $\mathbf{A}^{(3)}$ is often necessary. Hence, a few ALS iteration steps described in subsection 5.1 can be used to refine the estimates of $\mathbf{A}^{(1)}$, $\mathbf{A}^{(2)}$ and $\mathbf{A}^{(3)}$.

4.3. SGSD-CPO for Nth-Order Tensors. In this subsection we explain how the semi-unitary constrained SGSD-CP method can be extended to Nth-order tensors. Let $\mathcal{X} \in \mathbb{C}^{I_1 \times \cdots \times I_N}$ be a tensor of rank R constructed from the matrices $\mathbf{A}^{(n)} \in \mathbb{C}^{I_n \times R}$, $n \in [1, N]$, such that

$$\mathcal{X} = \sum_{r=1}^{R} \mathbf{a}_r^{(1)} \circ \cdots \circ \mathbf{a}_r^{(N)},$$

where $\mathbf{a}_r^{(n)}$ denotes the rth column vector of $\mathbf{A}^{(n)}$, and $\mathbf{A}^{(N)}$ is assumed to be a semi-unitary matrix. Let $\mathbf{X}^{(i_1,\ldots,i_{N-2})} \in \mathbb{C}^{I_{N-1} \times I_N}$ be constructed such that $\mathbf{X}^{(i_1,\ldots,i_{N-2})}_{i_{N-1},i_N} = \mathcal{X}_{i_1,\ldots,i_N}$, then

$$\mathbf{X}^{(i_1,\ldots,i_{N-2})} = \sum_{r=1}^{R} \prod_{n=1}^{N-2} \mathbf{A}^{(n)}_{i_n r} \mathbf{a}_r^{(N-1)} \mathbf{a}_r^{(N)T} = \mathbf{A}^{(N-1)} \prod_{n=1}^{N-2} D_{i_n}\left(\mathbf{A}^{(n)}\right) \mathbf{A}^{(N)T},$$

Denoting $\mathbf{D}^{(i_1,\ldots,i_{N-2})} = \prod_{n=1}^{N-2} D_{i_n}\left(\mathbf{A}^{(n)}\right) \in \mathbb{C}^{R \times R}$, we have

$$\mathbf{X}^{(i_1,\ldots,i_{N-2})} = \mathbf{A}^{(N-1)} \mathbf{D}^{(i_1,\ldots,i_{N-2})} \mathbf{A}^{(N)T}. \tag{4.5}$$

By ignoring the structural relations among the diagonal matrices $\{\mathbf{D}^{(i_1,\ldots,i_{N-2})}\}$, then in order to compute $\mathbf{A}^{(N-1)}$, $\mathbf{A}^{(N)}$ and $\{\mathbf{D}^{(i_1,\ldots,i_{N-2})}\}$ from (4.5) the semi-unitary constrained SGSD-CPO method discussed above can be applied.

Once $\mathbf{A}^{(N-1)}$, $\mathbf{A}^{(N)}$ and $\{\mathbf{D}^{(i_1,\ldots,i_{N-2})}\}$ have been obtained, the remaining matrix factors can be computed from a set of R decoupled best rank-1 tensor approximation problems. Indeed, we have that

$$\mathbb{C}^{\prod_{n=1}^{N-2} I_n \times R} \ni \mathbf{D} = \begin{bmatrix} \operatorname{Vecd}\left(\mathbf{D}^{(1,\ldots,1)}\right)^T \\ \operatorname{Vecd}\left(\mathbf{D}^{(1,\ldots,2)}\right)^T \\ \vdots \\ \operatorname{Vecd}\left(\mathbf{D}^{(I_1,\ldots,I_{N-2})}\right)^T \end{bmatrix} = \mathbf{A}^{(1)} \odot \mathbf{A}^{(2)} \odot \cdots \odot \mathbf{A}^{(N-2)}. \tag{4.6}$$

The Khatri-Rao product in the expression (4.6) indicates that every column of this matrix is the vector representation of a rank-1 term.

Remark that when $N = 4$, equation (4.6) reduces to $\mathbb{C}^{I_1 I_2 \times R} \ni \mathbf{D} = \mathbf{A}^{(1)} \odot \mathbf{A}^{(2)}$ and the matrix factors can be found from the set of R decoupled best rank-1 matrix approximation problems

$$\min_{\mathbf{a}_r^{(1)}, \mathbf{a}_r^{(2)}} \left\| \operatorname{Unvec}\left(\mathbf{d}_r\right) - \mathbf{a}_r^{(2)} \mathbf{a}_r^{(1)T} \right\|_F^2, \quad r \in [1, R],$$

where $\mathbf{d}_r$ and $\mathbf{a}_r^{(n)}$ denote the rth column vector of $\mathbf{D}$ and $\mathbf{A}^{(n)}$, respectively.

5. ALS-CPO: ALS-CP with semi-unitary matrix factor. Subsection 5.1 discusses how to incorporate a semi-unitary constraint into the ALS method for third-order tensors while subsection 5.2 discusses the extension to Nth-order tensors. The approach will be referred to as ALS-CPO.

5.1. ALS-CPO for Third Order Tensors. Recall that $\mathbf{X}_{(1)} = \left(\mathbf{A}^{(1)} \odot \mathbf{A}^{(2)}\right)\mathbf{A}^{(3)T}$ and assume that the matrix $\mathbf{A}^{(1)} \odot \mathbf{A}^{(2)}$ has full column rank and that $\mathbf{A}^{(3)}$ is semi-unitary. Let $\mathbf{X}_{(1)} = \mathbf{U}\Sigma\mathbf{V}^H$ denote the compact SVD of $\mathbf{X}_{(1)}$. Then $\mathbf{A}^{(3)T} = \mathbf{F}^T\mathbf{V}^H$ for some unitary matrix $\mathbf{F} \in \mathbb{C}^{R\times R}$. The first step of the semi-unitary constrained ALS-CP method is to compute $\widetilde{\mathbf{X}}_{(1)} = \mathbf{X}_{(1)}\mathbf{V} = \left(\mathbf{A}^{(1)} \odot \mathbf{A}^{(2)}\right)\mathbf{F}^T$. After this initial dimensionality reduction step, the matrices $\mathbf{A}^{(1)}$, $\mathbf{A}^{(2)}$ and $\mathbf{F}$ will be found by the unitary constrained ALS method presented in [19]. We recall this technique below for completeness. Related techniques have been proposed in [30], [6].

The ALS method is one of the most popular algorithms for the computation of a CPD. The ALS method attempts to minimize the cost function

$$f\left(\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \mathbf{F}\right) = \left\|\widetilde{\mathbf{X}}_{(1)} - \left(\mathbf{A}^{(1)} \odot \mathbf{A}^{(2)}\right)\mathbf{F}^T\right\|_F^2. \tag{5.1}$$

The idea is to update, in an alternating fashion, one of the matrices $\{\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \mathbf{F}\}$ while keeping the other two fixed. When $\mathbf{A}^{(2)}$ and $\mathbf{F}$ are fixed, the Least Squares (LS) solution of $\mathbf{A}^{(1)}$ will be used as the conditional update of $\mathbf{A}^{(1)}$. The computation of such a conditional update is just a linear LS problem. Due to the multilinearity, the computation of the conditional update of $\mathbf{A}^{(2)}$ is also just a linear LS problem. On the other hand, the conditional update of $\mathbf{F}$ corresponds to solving a unitary Procrustes problem. This is an important difference with the unconstrained case. Moreover, it turns out that, when $\mathbf{F}$ is unitary, the ALS method can be reduced to a cyclic *two*-step procedure.

First we consider an update of the unitary matrix $\mathbf{F}$, i.e., the matrices $\mathbf{A}^{(1)}$ and $\mathbf{A}^{(2)}$ are fixed. A LS estimate of $\mathbf{F}$ can be found by minimizing the expression

$$f\left(\mathbf{F}\right) = \left\|\widetilde{\mathbf{X}}_{(1)} - \left(\mathbf{A}^{(1)} \odot \mathbf{A}^{(2)}\right)\mathbf{F}^T\right\|_F^2.$$

This problem is a unitary Procrustes problem. Let $\left(\mathbf{A}^{(1)} \odot \mathbf{A}^{(2)}\right)^H \widetilde{\mathbf{X}}_{(1)} = \mathbf{U}\Sigma\mathbf{V}^H$ denote the compact SVD of $\left(\mathbf{A}^{(1)} \odot \mathbf{A}^{(2)}\right)^H \widetilde{\mathbf{X}}_{(1)}$, then the optimal unitary matrix $\mathbf{F}$ is $\mathbf{F} = \mathbf{V}^*\mathbf{U}^T$ [18].

Next, we consider the update of $\mathbf{A}^{(1)}$ and $\mathbf{A}^{(2)}$, i.e., we assume that $\mathbf{F}$ is fixed. The LS estimates of $\mathbf{A}^{(1)}$ and $\mathbf{A}^{(2)}$ can be obtained from the expression

$$f\left(\mathbf{A}^{(1)}, \mathbf{A}^{(2)}\right) = \left\|\widetilde{\mathbf{X}}_{(1)} - \left(\mathbf{A}^{(1)} \odot \mathbf{A}^{(2)}\right)\mathbf{F}^T\right\|_F^2 = \left\|\widetilde{\mathbf{X}}_{(1)}\mathbf{F}^* - \left(\mathbf{A}^{(1)} \odot \mathbf{A}^{(2)}\right)\right\|_F^2.$$

Indeed, let $\mathbf{Y} = \widetilde{\mathbf{X}}_{(1)}\mathbf{F}^*$, then

$$\mathrm{Unvec}\left(\mathbf{y}_r\right) = \mathrm{Unvec}\left(\mathbf{a}_r^{(1)} \otimes \mathbf{a}_r^{(2)}\right) = \mathbf{a}_r^{(2)}\mathbf{a}_r^{(1)T},$$

where $\mathbf{y}_r$ and $\mathbf{a}_r^{(n)}$ denote the rth column vector of $\mathbf{Y}$ and $\mathbf{A}^{(n)}$, respectively. Hence, we can take as $\mathbf{a}_r^{(1)} = \mathbf{v}_1^*$ and $\mathbf{a}_r^{(2)} = \sigma_1\mathbf{u}_1$, where σ_1 denotes the largest singular value

of Unvec$\left(\mathbf{y}_r\right)$ = and $\mathbf{u}_1$ and $\mathbf{v}_1$ denote its dominant left and right singular vector, respectively.

The unitary constrained ALS method repeats these two conditional updates until convergence. Once it has converged, we obtain the semi-unitary matrix factor as $\mathbf{A}^{(3)T} = \mathbf{F}^T\mathbf{V}^H$. This semi-unitary constrained ALS method will be referred to as ALS1-CPO.

On the other hand, if $\mathbf{A}^{(1)}$ and $\mathbf{A}^{(2)}$ are updated via the conventional conditional LS procedure like in [30], then the semi-unitary constrained ALS method will be referred to as ALS2-CPO.

5.2. ALS-CPO for Nth-Order Tensors. We now explain how the semi-unitary constrained ALS-CPO methods can be extended to Nth-order tensors. The extension of the ALS2-CPO method is obvious and will not explained. Let us now explain how to extend the ALS1-CPO method to Nth-order tensors. Consider the tensor $\mathcal{X} \in \mathbb{C}^{I_1\times\cdots\times I_N}$ with rank R and matrix representation $\mathbf{X}_{[N]} = \left(\mathbf{A}^{(1)} \odot \cdots \odot \mathbf{A}^{(N-1)}\right)\mathbf{A}^{(N)T}$. Assume that the matrix $\mathbf{A}^{(1)} \odot \cdots \odot \mathbf{A}^{(N-1)}$ has full column rank and that $\mathbf{A}^{(N)}$ is semi-unitary. Let $\mathbf{X}_{[N]} = \mathbf{U}\Sigma\mathbf{V}^H$ denote the compact SVD of $\mathbf{X}_{[N]}$, then $\mathbf{A}^{(N)T} = \mathbf{F}^T\mathbf{V}^H$ and $\widetilde{\mathbf{X}}_{[N]} = \mathbf{X}_{[N]}\mathbf{V} = \left(\mathbf{A}^{(1)} \odot \cdots \odot \mathbf{A}^{(N-1)}\right)\mathbf{F}^T$ for some unitary matrix $\mathbf{F} \in \mathbb{C}^{R\times R}$. The conditional update of $\mathbf{F}$ corresponds to solving a unitary Procrustes problem as explained above. Next, the Khatri-Rao product structure $\mathbf{A}^{(1)} \odot \cdots \odot \mathbf{A}^{(N-1)}$ indicates that each column of $\widetilde{\mathbf{X}}_{[N]}\mathbf{F}^*$ is the vector representation of a rank-1 tensor of order $N-1$. Hence, the problem of updating the involved matrix factors while $\mathbf{F}$ is fixed corresponds to solving a set of R decoupled rank-1 tensor approximation problems.

To summarize, after an initial dimension reduction step the semi-unitary constrained ALS1-CPO method for Nth-order tensors consists of alternating between solving a unitary Procrustes problem and solving a set of R decoupled rank-1 tensor approximation problems.

In the ALS1-CPO method for Nth-order tensors the most expensive step is the computation of the best rank-1 tensor approximation problems. In order to reduce the computational burden, the so-called ORBIT method was proposed in [19]. However, the ORBIT method does not fully exploit the structure of the problem Let $\mathbb{C}^{\prod_{n=2}^{R} I_n\times R} \ni \mathbf{D} = \mathbf{A}^{(2)} \odot \cdots \odot \mathbf{A}^{(N-1)}$, then

$$\mathbf{X}_{[N]} = \left(\mathbf{A}^{(1)} \odot \mathbf{D}\right)\mathbf{A}^{(N)T}. \tag{5.2}$$

If we ignore the structure of $\mathbf{D}$, then we can treat (5.2) as a third-order CPD problem with a semi-unitary matrix factor $\mathbf{A}^{(N)}$. Hence, the ORBIT method first computes $\mathbf{A}^{(1)}$, $\mathbf{D}$ and $\mathbf{A}^{(N)}$ via the procedures described in subsection 5.1. Next, it computes the remaining matrix factors from the decoupled best rank-1 tensor approximation problems

$$\mathbf{D} = \mathbf{A}^{(2)} \odot \cdots \odot \mathbf{A}^{(N-1)}. \tag{5.3}$$

When $N = 4$, then the solution to (5.3) follows from R decoupled rank-1 matrix approximation problems.

6. Numerical Experiments. We restrict the simulation study to real third and fourth order tensors. Let $\mathcal{T} \in \mathbb{R}^{I_1\times I_2\times I_3}$ or $\mathcal{T} \in \mathbb{R}^{I_1\times I_2\times I_3\times I_4}$ with rank R denote the structured tensor we attempt to estimate from the observed tensor $\mathcal{X} = \mathcal{T} + \beta\mathcal{N}$, where $\mathcal{N}$ is an unstructured perturbation tensor and $\beta \in \mathbb{R}$. The entries of the matrix factors

of $\mathcal{T}$ and the perturbation tensor $\mathcal{N}$ are randomly drawn from a uniform distribution with support $[-\frac{1}{2}, \frac{1}{2}]$. The following Signal-to-Noise Ratio (SNR) measure will be used

$$\text{SNR [dB]} = 10 \log \left(\frac{\|\mathcal{T}\|_F^2}{\left\|\beta \mathcal{N}\right\|_F^2} \right).$$

Furthermore, when $\mathbf{A}^{(n)} \in \mathbb{C}^{I_n \times R}$ with $I_n \geq R$, the following performance measure will also be used

$$\mathrm{P}\left(\mathbf{A}^{(n)}\right) = \min_{\Pi\Lambda} \frac{\left\|\mathbf{A}^{(n)} - \widehat{\mathbf{A}}^{(n)} \Pi\Lambda\right\|_F}{\left\|\mathbf{A}^{(n)}\right\|_F},$$

where $\widehat{\mathbf{A}}^{(n)}$ denotes the estimated matrix factor, Π denotes a permutation matrix and Λ denotes a diagonal matrix. When $\mathbf{A}^{(n)} \in \mathbb{C}^{I_n \times R}$ with $I_n < R \leq I_n^2$, we will use the following function

$$\mathrm{Q}\left(\mathbf{A}^{(n)}\right) = \min_{\Pi\Lambda} \frac{\left\|\mathbf{A}^{(n)} \odot \mathbf{A}^{(n)} - \left(\widehat{\mathbf{A}}^{(n)} \odot \widehat{\mathbf{A}}^{(n)}\right) \Pi\Lambda\right\|_F}{\left\|\mathbf{A}^{(n)} \odot \mathbf{A}^{(n)}\right\|_F},$$

as a performance measure. In order to find Π and Λ the greedy LS column matching algorithm between $\mathbf{A}^{(n)}$ and $\widehat{\mathbf{A}}^{(n)}$ proposed in [28] will be applied.

To measure the elapsed time in second used to execute the algorithms in MATLAB, the built-in functions *tic*(·) and *toc*(·) is used.

The ALS methods are randomly initialized and we decide that the ALS methods have converged when the applied cost function at iteration k and $k+1$ has changed less than $\epsilon_{\text{ALS}} = 10^{-8}$ or the number of iterations has exceeded 2000. Similarly, we decide that the extended QZ and the Jacobi iteration methods have converged when the applied cost function at iteration k and $k+1$ has changed less than $\epsilon = 10^{-6}$ or the number of iterations has exceeded 600.

6.1. Third order CPDs. In all the simulations in this subsection the matrix factor $\mathbf{A}^{(3)}$ is semi-unitary. The simulation cases are of increasing complexity. Case 1 and 2 deal with overdetermined problems ($I_1 = I_2 = I_3 \geq R$), case 3 and 4 deal with underdetermined problems ($I_2, I_3 > R$, $I_1 < R$) while case 5 deals with a highly underdetermined problem ($I_3 > R$ and $I_1, I_2 < R$).

Case 1: CPD with $I_1 = I_2 = I_3 = R$. We set $I_1 = I_2 = I_3 = R = 5$ and compare the performance of the semi-unitary constrained ALS1-CPO and ALS2-CPO, SD-CPO and SGSD-CPO methods with the unconstrained ALS-CP, SD-CP and SGSD-CP methods. The mean and median $\mathrm{P}\left(\mathbf{A}^{(n)}\right)$ and time values over 100 trials as a function of SNR can be seen in figure 7.1. We notice that the unconstrained methods perform worse than their semi-unitary counterparts. Moreover, we notice that the unconstrained ALS method is more costly than the semi-unitary constrained ALS method which is in turn more costly than the SD-CP(O) and SGSD-CP(O) methods. We also notice that above 20 dB SNR the ALS1-CPO, ALS2-CPO and SD-CPO methods perform about the same while below 20 dB SNR the SD-CPO method yields slightly worse $\mathrm{P}\left(\mathbf{A}^{(n)}\right)$ than the mentioned methods. The reason for this is that in the noise

free case SD-CPO yields the exact solution while at low SNR values the noise free assumption is violated. Finally, we notice that the SGSD-CPO method performs worse than the other semi-unitary constrained methods. This is due to the relaxation step taken by the SGSD-CPO method. Obviously, the situation for the unconstrained SGSD-CP method is even worse.

Case 2: CPD with $I_1, I_2, I_3 > R$. We set $I_1 = I_2 = I_3 = 8$, $R = 5$ and compare the performance of the semi-unitary constrained ALS1-CPO and ALS2-CPO, SD-CPO and SGSD-CPO methods with the unconstrained ALS-CP, SD-CP and SGSD-CP methods. The mean and median $\mathrm{P}\left(\mathbf{A}^{(n)}\right)$ and time values over 100 trials as a function of SNR can be seen in figure 7.2. Again we notice that the unconstrained methods perform worse and are more costly than their semi-unitary counterparts. In particular, the ALS-CP method has not always converged within 2000 iterations. We also notice that above 15 dB SNR the ALS1-CPO, ALS2-CPO , SD-CPO and SGSD-CPO methods perform about the same. The reason that the SGSD-CPO method performs about the same as the other semi-unitary constrained methods is that the CPD problem is highly overdetermined which implies that the relaxation step taken by the SGSD-CPO method has a smaller impact on the performance than in case 1. On the other hand, the unconstrained SGSD-CP method still suffers from it. The SGSD-CPO method is consistently faster than the ALS1-CPO and ALS2-CPO methods but this time the SD-CPO method is the slowest semi-unitary method. The reason SD-CPO is the slowest semi-unitary constrained method is that the number of rows of the matrix computed in the first step depends quadratically on I_1 and I_2 [13].

Case 3: CPD with $I_2, I_3 > R$ *and* $I_1 < R$. We set $I_1 = 3$, $I_2 = 8$, $I_3 = 8$, $R = 5$ and compare the performance of the semi-unitary constrained ALS1-CPO and ALS2-CPO, SD-CPO and SGSD-CPO methods with the unconstrained ALS-CP, SD-CP and SGSD-CP methods. The mean and median $\mathrm{Q}\left(\mathbf{A}^{(1)}\right)$, $\mathrm{P}\left(\mathbf{A}^{(2)}\right)$, $\mathrm{P}\left(\mathbf{A}^{(3)}\right)$ and time values over 100 trials as a function of SNR can be seen in figure 7.3. Again we notice that the unconstrained methods perform worse than their semi-unitary counterparts. We also notice that below 25 dB SNR the ALS1-CPO and ALS2-CPO methods perform better than the SD-CPO and SGSD-CPO methods. The reason for this is that in the noise free case the SD-CPO and SGSD-CPO yield the exact solution but at low SNR values the noise free assumption is violated. However, the SD-CPO and SGSD-CPO methods are consistently faster than the ALS1-CPO and ALS2-CPO methods. This suggests to use the SD-CPO or the SGSD-CPO method to initialize the ALS1-CPO method as illustrated in the next test case. We also notice that in the current case the ALS1-CPO method is faster than the ALS2-CPO method, which is in turn faster than the ALS-CP method. Hence, when not all matrix factors have full column rank, then the two-step ALS2-CPO updating procedure is indeed faster than the three-step ALS1-CPO updating procedure.

Case 4: CPD with $I_2, I_3 > R$, $I_1 < R$ *and Refinement Step*. We set $I_1 = 3$, $I_2 = 8$, $I_3 = 8$, $R = 5$. Moreover, we let the SD-CPO, SGSD-CPO, SD-CP and SGSD-CP methods be followed by at most 100 ALS1-CPO refinement steps. Consequently, we refer to them as SD-ALS-CPO, SGSD-ALS-CPO, SD-ALS-CP and SGSD-ALS-CP, respectively. We compare the performance of the semi-unitary constrained ALS1-CPO and ALS2-CPO, SD-ALS-CPO and SGSD-ALS-CPO methods with the unconstrained ALS-CP, SD-CP and SGSD-CP methods. The mean and median $\mathrm{Q}\left(\mathbf{A}^{(1)}\right)$, $\mathrm{P}\left(\mathbf{A}^{(2)}\right)$, $\mathrm{P}\left(\mathbf{A}^{(3)}\right)$ and time values over 100 trials as a function of SNR can be seen in figure 7.4. Again we notice that the unconstrained ALS-CP and SD-CP methods perform worse than their semi-unitary counterparts. Due to the refinement step we notice that the ALS1-CPO,

ALS2-CPO , SD-ALS-CPO, SGSD-ALS-CPO and SGSD-ALS-CP methods perform about the same while the SD-CP-ALS performs worse at low SNR. We also notice that the SD-ALS-CP(O) and SGSD-ALS-CP(O) methods are only slightly faster than the ALS1-CPO method. Hence, when a refinement step is applied and $I_2, I_3 > R$, $I_1 < R$, then all the semi-unitary constrained methods seem to yield a similar performance. This indicates in fact that too many refinement steps are taken, at least at high SNR, where the accuracy was already high.

Case 5: CPD with $I_3 > R$ and $I_1, I_2 < R$. We set $I_1 = 4$, $I_2 = 3$, $I_3 = 8$, $R = 5$ and compare the performance of the semi-unitary constrained ALS1-CPO, ALS2-CPO and SD-CPO methods with the unconstrained ALS-CP and SD-CP methods. In this case the SGSD-CP(O) method cannot be used. Moreover, we let the SD-CPO and SD-CP methods be followed by at most 200 ALS1-CPO refinement steps and hence we refer to them as SD-ALS-CPO and SD-ALS-CP, respectively. The mean and median $\mathrm{Q}\left(\mathbf{A}^{(1)}\right)$, $\mathrm{Q}\left(\mathbf{A}^{(2)}\right)$, $\mathrm{P}\left(\mathbf{A}^{(3)}\right)$ and time values over 100 trials as a function of SNR can be seen in figure 7.5. We notice that the unconstrained methods perform significantly worse than their semi-unitary counterparts. Hence, in this case there is a significant benefit in imposing the semi-unitary constraint. We also notice that above 25 dB SNR no significant difference in performance between the ALS1-CPO, ALS2-CPO, SD-CPO and SD-ALS-CPO methods is observed. Thus, above 25 dB SNR the SD-CPO method does not benefit from an additional refinement step. Below 25 dB SNR the ALS1-CPO and ALS2-CPO perform better than the SD-CPO method while the SD-ALS-CPO method performs as well as the ALS1-CPO and ALS2-CPO methods. The reason for this is that in the noise free case the SD-CPO yields the exact solution but at low SNR values the noise free assumption is violated. Hence, below 25 dB SNR the SD-CPO method does benefit from an additional refinement step. Finally, we notice that the SD-CPO and SD-ALS-CPO methods are consistently faster than the ALS1-CPO method which is in turn faster than the ALS2-CPO method. This shows the importance of initializing by means of SD-CPO.

6.2. Fourth order CPDs. In the simulations in this subsection the matrix factor $\mathbf{A}^{(4)}$ is semi-unitary. Moreover, we also let the ORBIT, SD-CPO and SGSD-CPO methods be followed by at most 200 ALS1-CPO refinement steps. Consequently, we refer to them as ALS-ORBIT, SD-ALS-CPO and SGSD-ALS-CPO, respectively.

Case 6: CPD with $I_3, I_4 > R$ and $I_1, I_2 < R$. We set $I_1 = I_2 = 3$, $I_3 = I_4 = 5$ and $R = 4$ and compare the performance of the ALS1-CPO, ORBIT, ALS-ORBIT, SD-CPO, SD-ALS-CPO, SGSD-CPO and SGSD-ALS-CPO methods. The mean and median $\mathrm{Q}\left(\mathbf{A}^{(1)}\right)$, $\mathrm{Q}\left(\mathbf{A}^{(2)}\right)$, $\mathrm{P}\left(\mathbf{A}^{(3)}\right)$, $\mathrm{P}\left(\mathbf{A}^{(4)}\right)$ and time values over 100 trials as a function of SNR can be seen in figure 7.6. We notice that the ALS1-CPO method has not always converged within 2000 iterations. Below 20 dB SNR a subsequent ALS refinement step seems to improve the performance of the SD-CPO and SGSD-CPO methods. Furthermore, below 30 dB SNR the ALS-ORBIT, SD-ALS-CPO and SGSD-ALS-CPO methods perform about the same while above 30 dB SNR the SD-ALS-CPO and SGSD-ALS-CPO methods seem to perform better than the ALS-ORBIT method. We also notice that the ALS1-CPO method is significantly more expensive and less accurate at high SNR than the other methods. SGSD-CPO is the cheapest method, but also the least accurate at low SNR. The SD-CPO and ORBIT methods cost about the same.

Case 7: CPD with $I_4 > R$ and $I_1, I_2, I_3 < R$. We set $I_1 = 3 = I_2 = 3 = I_3 = 3$, $I_4 = 15$ and $R = 6$. We compare the performance of the ALS1-CPO, ORBIT, ALS-ORBIT, SD-CPO and SD-ALS-CPO methods. The mean and median $\mathrm{Q}\left(\mathbf{A}^{(1)}\right)$, $\mathrm{Q}\left(\mathbf{A}^{(2)}\right)$, $\mathrm{Q}\left(\mathbf{A}^{(3)}\right)$,

$P\left(\mathbf{A}^{(4)}\right)$ and time values over 100 trials as a function of SNR can be seen in figure 7.7. We notice that the SD-ALS-CPO method seems to perform better than the other methods. Again the ALS1-CPO method is significantly more expensive than the other methods while the SD-CPO and ORBIT methods cost about the same.

7. Conclusion. In various practical problems one of the matrix factors of the CPD is constrained to be semi-unitary. For this CPD problem an optimal solution always exists. We explained that this semi-unitary constraint also leads to a new relaxed uniqueness condition of the CPD.

To numerically solve the semi-unitary constrained CPD problem we first presented a semi-unitary constrained version of the SD-CP method, which we called SD-CPO. This led to a simultaneous Takagi factorization problem which was solved by a Jacobi iteration scheme. Next, we presented a semi-unitary constrained version of the SGSD-CP method, which we called SGSD-CPO. An extension of the extended QZ method was derived to solve the problem. It was also explained how to extend the SD-CPO and SGSD-CPO methods for computing the CPD of Nth-order tensors.

In general, when one of the matrix factors of the CPD is semi-unitary, the problem of computing the CPD reduces from the search of nonsingular matrices to unitary matrices. This means that by taking this structural constraint into account we should get more accurate and computationally efficient methods.

Finally, numerical experiments were reported. We observed that by taking the semi-unitary constraint into account we indeed obtained more accurate and computationally efficient methods. We noticed that the ALS1-CPO method is cheaper than the ALS2-CPO method. At high SNR values the SD-CPO, SGSD-CPO and ORBIT methods are cheap and yield a good performance compared to the ALS method. At low SNR value the ALS-CPO method may yield better performance results than the SD-CPO, SGSD-CPO and ORBIT methods. The reason for this is that in the noise free case the SD-CPO, SGSD-CPO and ORBIT methods yield the exact solution but at low SNR values the noise free assumption is violated. However, the ALS methods can still be quite costly. Thus, at low SNR values it can still be useful to initialize ALS by the SD-CPO, SGSD-CPO and ORBIT methods.

REFERENCES

[1] L. Albera and A. Ferréol and P. Comon and P. Chevalier, *Blind Identification of Overomplete MixturEs of sources (BIOME)*, Linear Algebra and its Applications, 391(2004), pp. 3–30.

[2] C.F. Beckmann and S.M. Smith, *Tensorial Extension of Independent Component Analysis for Multisubject fMRI Analysis*, Neuroimage, 25(2005), pp. 294–311.

[3] J.-F. Cardoso and A. Souloumiac, *Blind Beamforming for non-Gaussian Signals*, IEEE Proceedings-F, 140(1993), pp. 362–370.

[4] J.-F. Cardoso and A. Souloumiac, *Jacobi Angles for Simultaneous Diagonalization*, SIAM J. Mat. Anal. Appl., 17(1996), pp. 161–164.

[5] J. Carroll and J. Chang, *Analysis of Individual Differences in Multidimensional Scaling via an N-way Generalization of "Eckart-Young" Decomposition*, Psychometrika, 9(1970), pp. 267-283.

[6] J. Chen and Y. Saad, *On the Tensor SVD and the Optimal Low Rank Orthogonal Approximation of Tensors*, SIAM J. Mat. Anal. Appl., 30(2009), pp. 1709–1734.

[7] P. Comon, *Independent Component Analysis, a new concept?*, Signal Processing, Elsevier, 36(1994), pp. 287–314.

[8] P. Comon and X. Luciani and A. L. F. De Almeida, *Tensor Decompositions, Alternating Least Squares and other Tales*, J. Chemometrics, 23(2009), pp. 393–405.

[9] L. De Lathauwer and B. De Moor and J. Vandewalle, *On the Best Rank-1 and Rank-(R_1, R_2, ..., R_N) Approximation of Higher-Order Tensors*, SIAM J. Matrix Anal. Appl., 21(2000), pp. 1324-1342.

[10] L. De Lathauwer and B. De Moor and J. Vandewalle, *Independent Component Analysis and (Simultaneous) Third-Order Diagonalization*, IEEE Trans. Signal Process., 49 (2001), pp. 2262-2271.

[11] L. De Lathauwer and B. De Moor, *On the Blind Separation of Non-circular Sources*, Proc. EUSIPCO, Sept. 3-6. 2002, Toulouse, France.

[12] L. De Lathauwer and 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 (2004), pp. 295–327.

[13] L. De Lathauwer, *A Link between the Canonical Decomposition in Multilinear Algebra and Simultaneous Matrix Diagonalization*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 642–666.

[14] L. De Lathauwer and J. Castaing and J.-F. Cardoso, *Fourth-Order Cumulant Based Identification of Underdetermined Mixtures*, IEEE Trans. Signal Process., 55 (2007), pp. 2965-2973.

[15] L. De Lathauwer and J. Castaing, *Blind Identification of Underdetermined Mixtures by Simultaneous Matrix Diagonalization*, IEEE Trans. Signal Process., 56 (2008), pp. 1096-1105.

[16] G. Golub and C. Van Loan, *Matrix Computations*, John Hopkins University Press, 1996.

[17] R. A. Harshman, *Foundations of the Parafac Procedure: Models and Conditions for an Explanatory Multimodal Factor Analysis*, UCLA Working Papers in Phonetics, 16 (1970), pp. 1–84.

[18] R. A. Horn and C. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, 1985.

[19] A. Karfoul and L. Albera and L. De Lathauwer, *Canonical Decomposition of Even Higher Order Cumulant Arrays for Blind Underdetermined Mixture Identification*, Proc. SAM, July 21-23, Darmstadt, Germany, 2008.

[20] W. P. Krijnen and T. K. Dijkstra and A. Stegeman, *On the Non-existence of Optimal Solutions and the Occurrence of "Degeneracy" in the CANDECOMP/PARAFAC Model*, Psychometrika, 73 (2008), pp. 431–439.

[21] J. B. Kruskal, *Thee-way Arrays: Rank and Uniqueness of Trilinear Decompositions, with Applications to Arithmetic Complexity and Statistics*, Linear Algebra and its Applications, 18 (1977), pp. 95–138.

[22] C. Martin and C. Van Loan, *A Jacobi-type Method for Orthogonal Tensor Decompositions*, SIAM J. Matrix Anal. Appl., 30 (2008), pp. 1219–1232.

[23] B. Pesquet-Popoescu and J.-C. Pesquet and A.P. Petropulu, *Joint Singular Value Decomposition - A New Tool for Separable Representation of Images*, Proc. ICIP, 2 (2001), pp. 569–572.

[24] M. Sørensen and L. De Lathauwer and L. Deneire, *PARAFAC with Orthogonality in One Mode and Applications in DS-CDMA Systems*, Proc. ICASSP, March 14-19, Dallas, USA, 2010.

[25] M. Sørensen and P. Comon and L. De Lathauwer and S. Icart and L. Deneire, *Simultaneous Generalized Schur Decomposition Methods for Computing the Canonical Polyadic Decomposition*, in preparation.

[26] M. Sørensen and L. De Lathauwer and S. Icart and L. Deneire, *Canonical Polyadic (CP) Structured ICA with Applications*, in preparation.

[27] M. Sørensen and L. De Lathauwer and P. Comon, *Cumulant Tensor based Identification of MIMO FIR Channels*, in preparation.

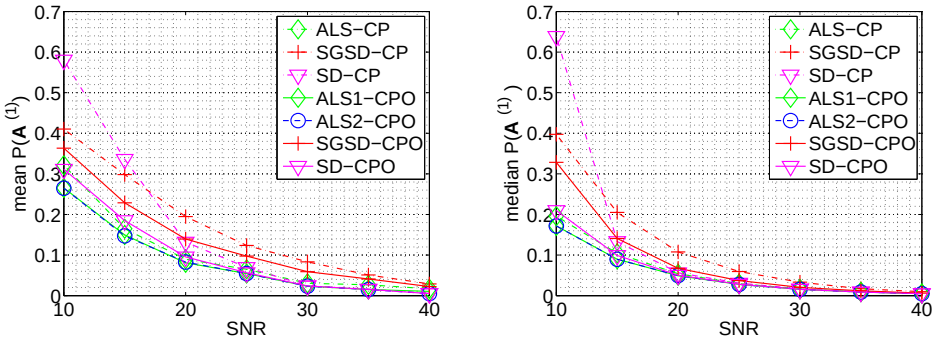
[28] N. D. Sidiropoulos and G. B. Giannakis and R.Bro, *Blind PARAFAC Receivers for DS-CDMA Systems*, IEEE Trans. Signal Process., 48 (2000), pp. 810-823.

[29] N. D. Sidiropoulos and R. Bro, *On the Uniqueness of Multilinear Decomposition of N-way Arrays*, J. Chemometrics, 14 (2000), pp. 229–239.

[30] J. M. F. Ten Berge, *Least Squares Optimization in Multivariate Analysis*, DSWO Press, Leiden, The Netherlands, 1993.

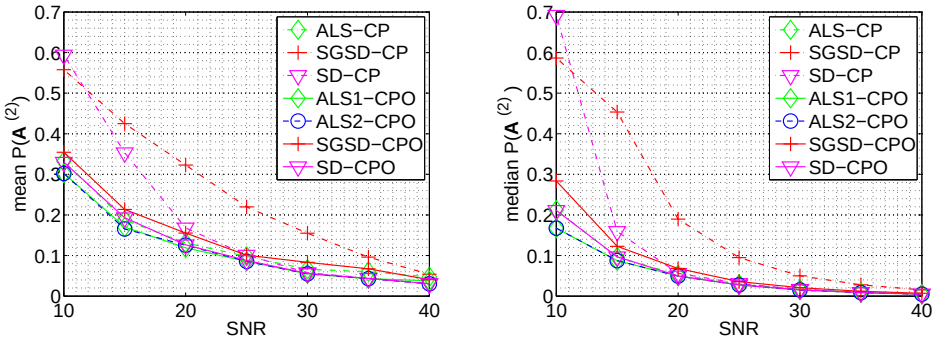
[31] J. M. F. Ten Berge and N. D. Sidiropoulos, *On Uniqueness in CANDECOMP/PARAFAC*, Psychometrika, 67(2002), pp. 399-409.

[32] A.-J. Van Der Veen and A. Paulraj, *Analytical Constant Modulus Algorithm*, IEEE Trans. Signal Process., 44 (1996), pp. 1136–1155.



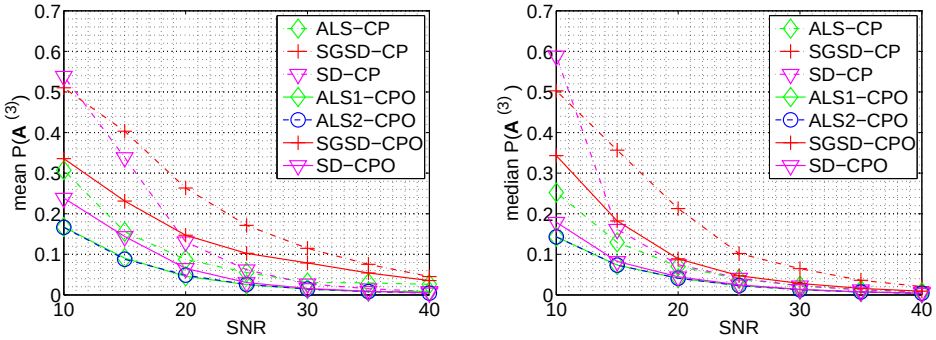
(a) Mean $P\left(\mathbf{A}^{(1)}\right)$.

(b) Median $P\left(\mathbf{A}^{(1)}\right)$.



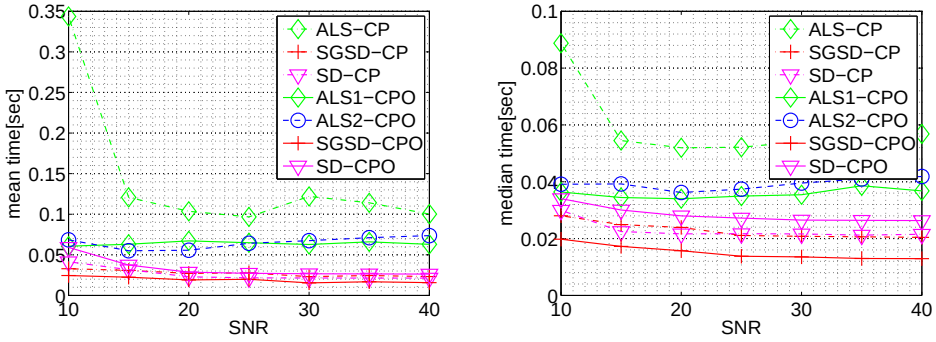
(c) Mean $P\left(\mathbf{A}^{(2)}\right)$.

(d) Median $P\left(\mathbf{A}^{(2)}\right)$.



(e) Mean $P\left(\mathbf{A}^{(3)}\right)$.

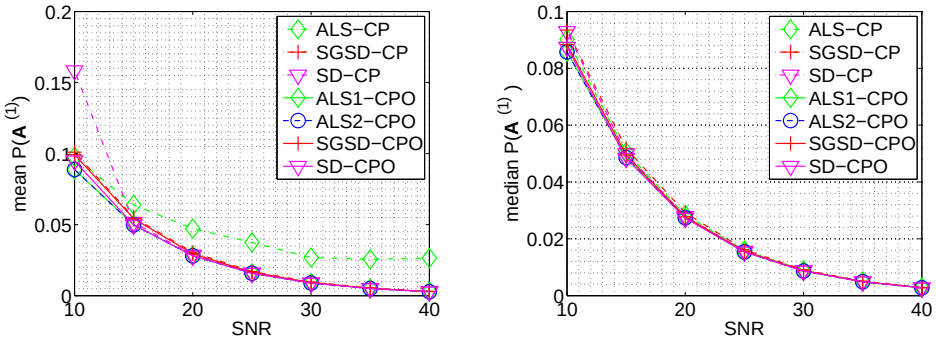
(f) Median $P\left(\mathbf{A}^{(3)}\right)$.



(g) Mean time.

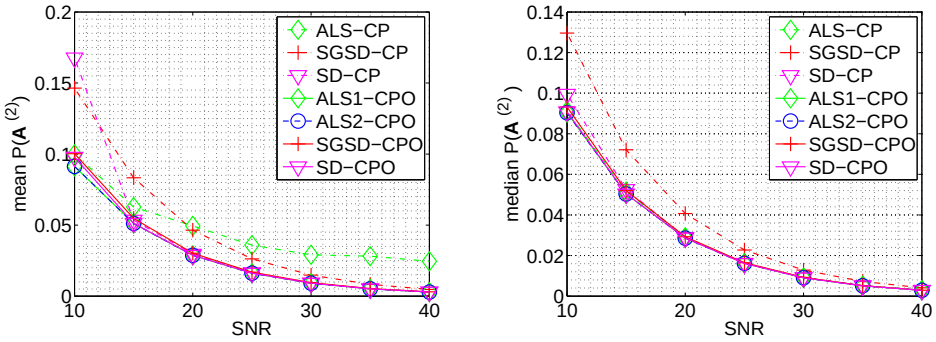
(h) Median time.

FIG. 7.1. *Mean and median* $P\left(\boldsymbol{A}^{(1)}\right)$, $P\left(\boldsymbol{A}^{(2)}\right)$, $P\left(\boldsymbol{A}^{(3)}\right)$ *and time values over 100 trials while SNR is varying from* 10 *to* 40 *dB for the real third order tensor simulation, case 1.*



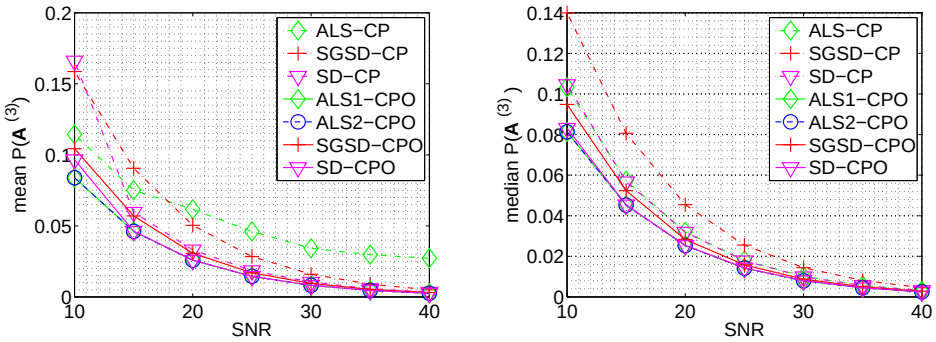
(a) Mean $P\left(\mathbf{A}^{(1)}\right)$.

(b) Median $P\left(\mathbf{A}^{(1)}\right)$.



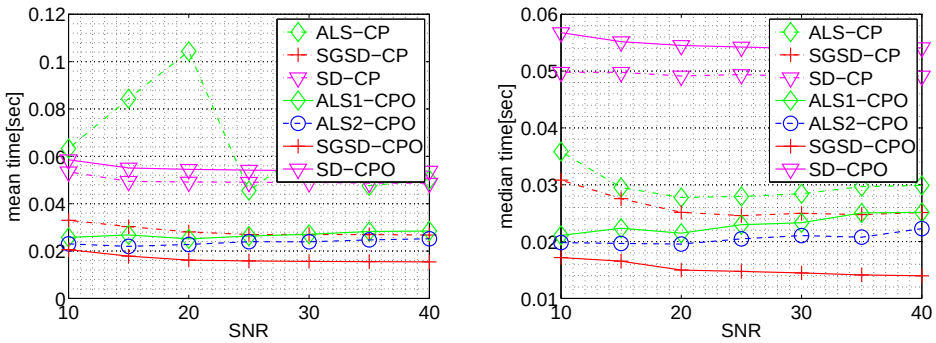
(c) Mean $P\left(\mathbf{A}^{(2)}\right)$.

(d) Median $P\left(\mathbf{A}^{(2)}\right)$.



(e) Mean $P\left(\mathbf{A}^{(3)}\right)$.

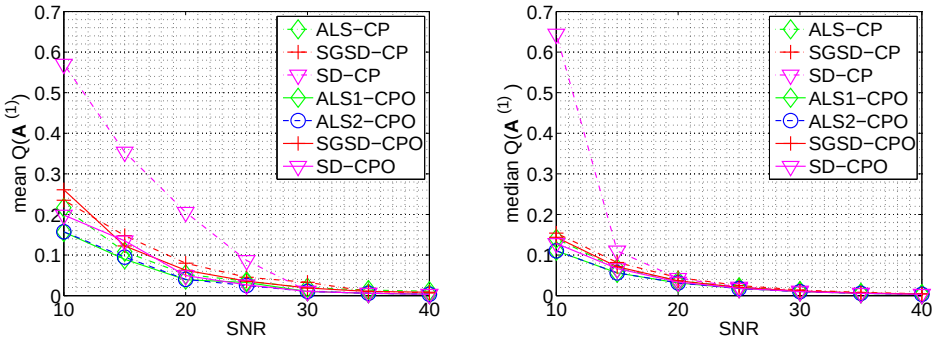
(f) Median $P\left(\mathbf{A}^{(3)}\right)$.



(g) Mean time.

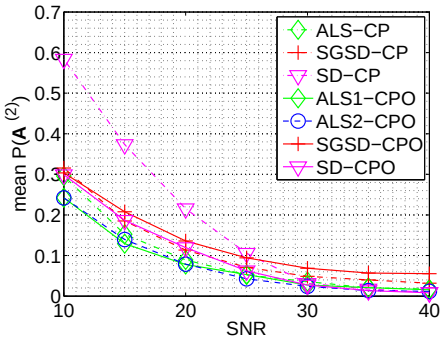
(h) Median time.

FIG. 7.2. *Mean and median* $P\left(\boldsymbol{A}^{(1)}\right)$, $P\left(\boldsymbol{A}^{(2)}\right)$, $P\left(\boldsymbol{A}^{(3)}\right)$ *and time values over 100 trials while SNR is varying from* 10 *to* 40 *dB for the real third order tensor simulation, case 2.*

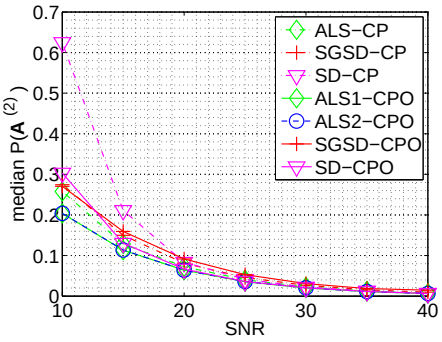


(a) Mean $Q\left(\mathbf{A}^{(1)}\right)$.

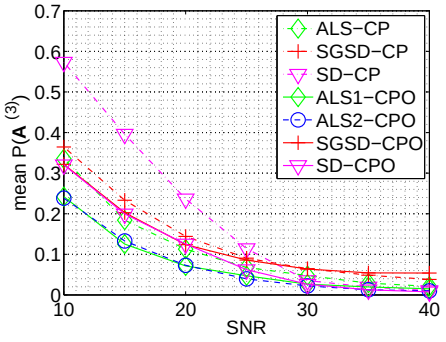
(b) Median $Q\left(\mathbf{A}^{(1)}\right)$.



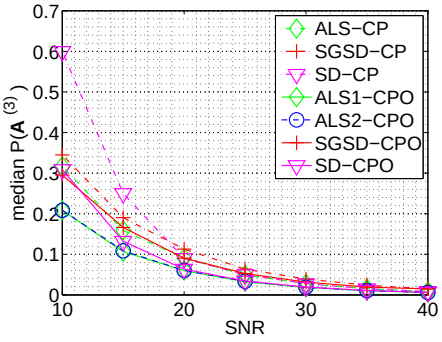
(c) Mean $P\left(\mathbf{A}^{(2)}\right)$.



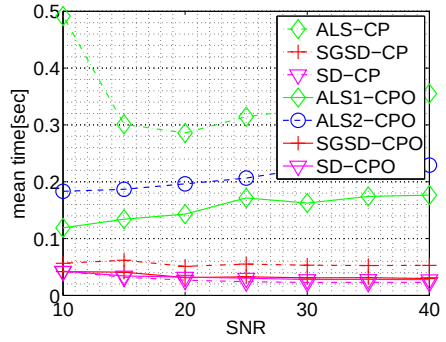
(d) Median $P\left(\mathbf{A}^{(2)}\right)$.



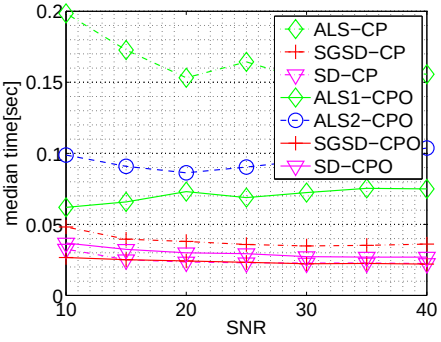
(e) Mean $P\left(\mathbf{A}^{(3)}\right)$.



(f) Median $P\left(\mathbf{A}^{(3)}\right)$.

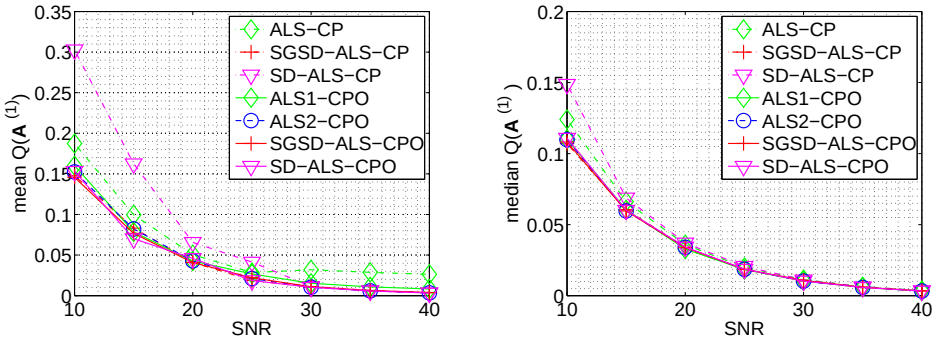


(g) Mean time.



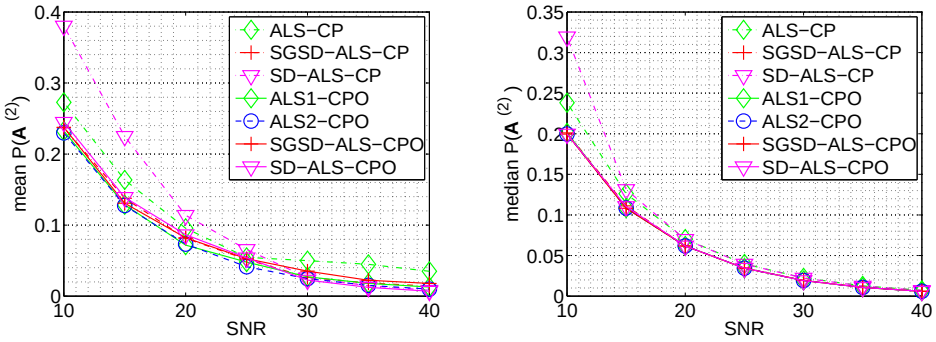
(h) Median time.

FIG. 7.3. *Mean and median* $Q\left(\boldsymbol{A}^{(1)}\right)$, $P\left(\boldsymbol{A}^{(2)}\right)$, $P\left(\boldsymbol{A}^{(3)}\right)$ *and time values over 100 trials while SNR is varying from* 10 *to* 40 *dB for the real third order tensor simulation, case 3.*



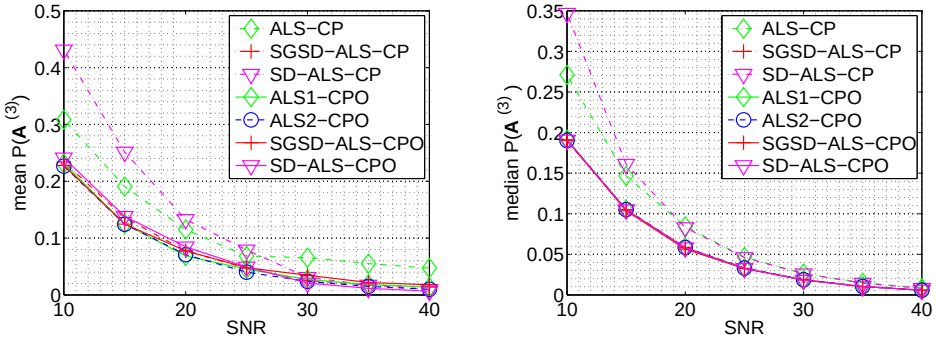
(a) Mean $Q\left(\mathbf{A}^{(1)}\right)$.

(b) Median $Q\left(\mathbf{A}^{(1)}\right)$.



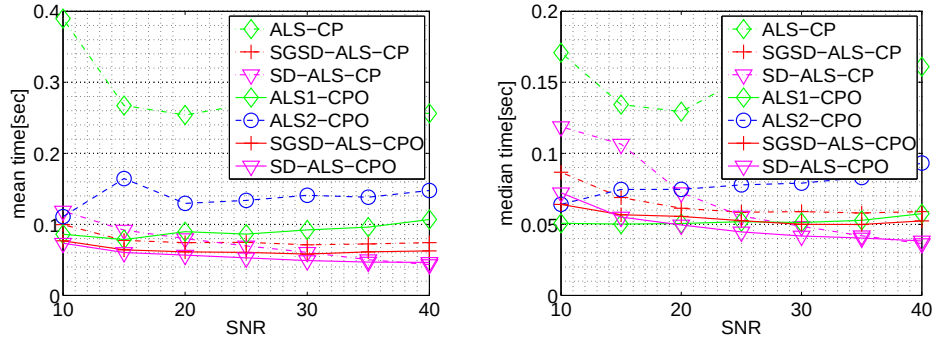
(c) Mean $P\left(\mathbf{A}^{(2)}\right)$.

(d) Median $P\left(\mathbf{A}^{(2)}\right)$.



(e) Mean $P\left(\mathbf{A}^{(3)}\right)$.

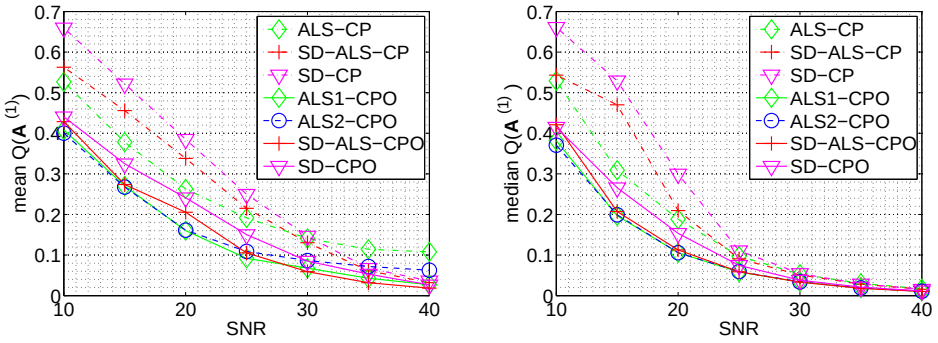
(f) Median $P\left(\mathbf{A}^{(3)}\right)$.



(g) Mean time.

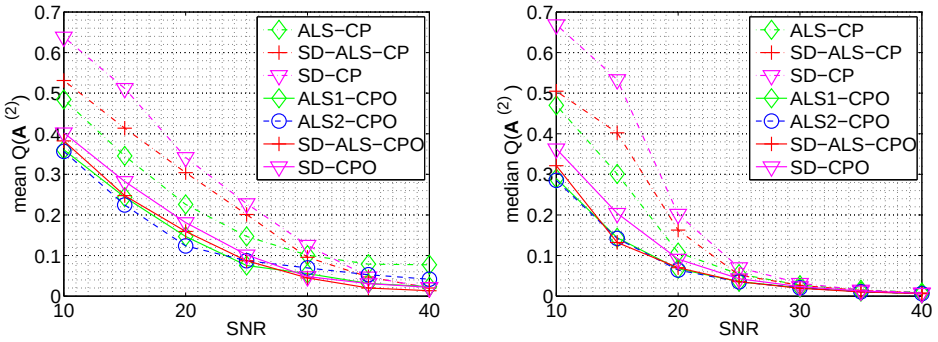
(h) Median time.

FIG. 7.4. *Mean and median* $Q\left(\boldsymbol{A}^{(1)}\right)$, $P\left(\boldsymbol{A}^{(2)}\right)$, $P\left(\boldsymbol{A}^{(3)}\right)$ *and time values over 100 trials while SNR is varying from* 10 *to* 40 *dB for the real third order tensor simulation, case 4.*



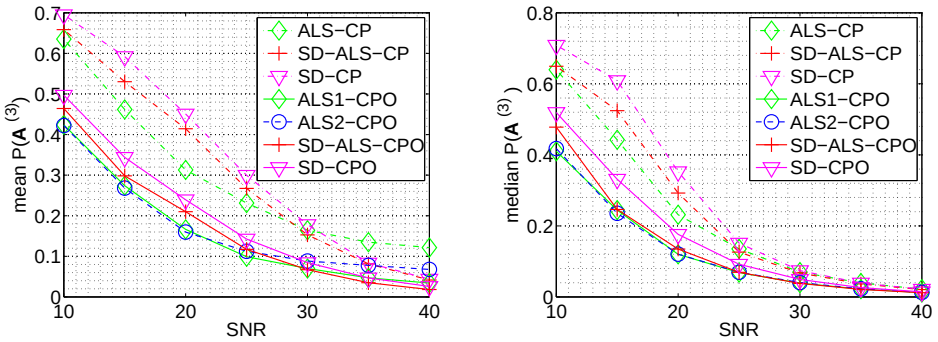
(a) Mean $\mathrm{Q}\left(\mathbf{A}^{(1)}\right)$.

(b) Median $\mathrm{Q}\left(\mathbf{A}^{(1)}\right)$.



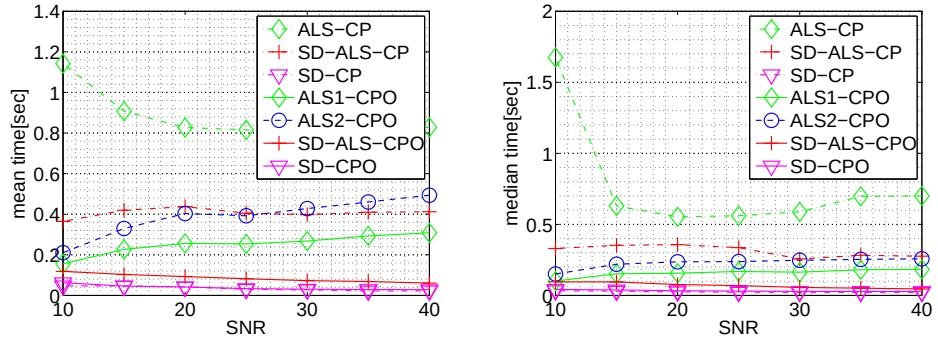
(c) Mean $\mathrm{Q}\left(\mathbf{A}^{(2)}\right)$.

(d) Median $\mathrm{Q}\left(\mathbf{A}^{(2)}\right)$.



(e) Mean $\mathrm{P}\left(\mathbf{A}^{(3)}\right)$.

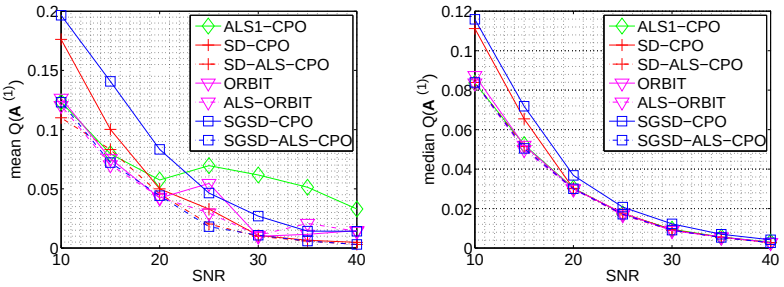
(f) Median $\mathrm{P}\left(\mathbf{A}^{(3)}\right)$.



(g) Mean time.

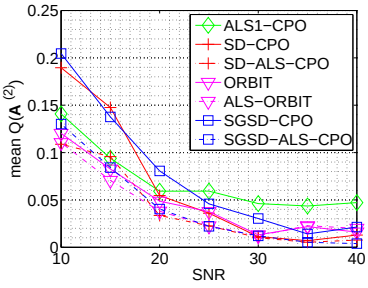
(h) Median time.

FIG. 7.5. *Mean and median* $Q\left(A^{(1)}\right)$, $Q\left(A^{(2)}\right)$, $P\left(A^{(3)}\right)$ *and time values over 100 trials while SNR is varying from* 10 *to* 40 *dB for the real third order tensor simulation, case 5.*

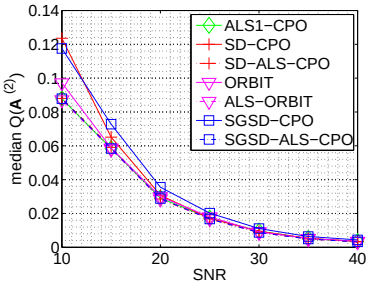


(a) Mean $Q\left(\mathbf{A}^{(1)}\right)$.

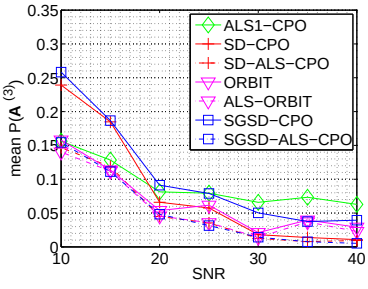
(b) Median $Q\left(\mathbf{A}^{(1)}\right)$.



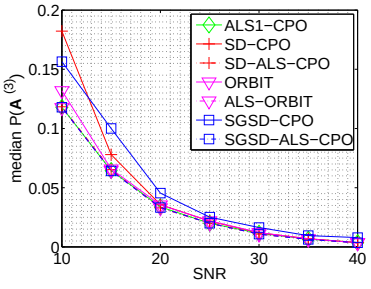
(c) Mean $Q\left(\mathbf{A}^{(2)}\right)$.



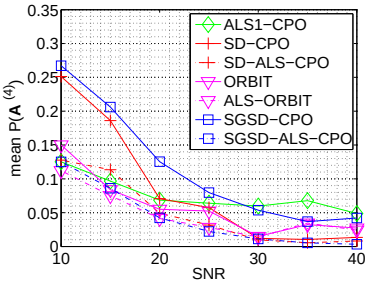
(d) Median $Q\left(\mathbf{A}^{(2)}\right)$.



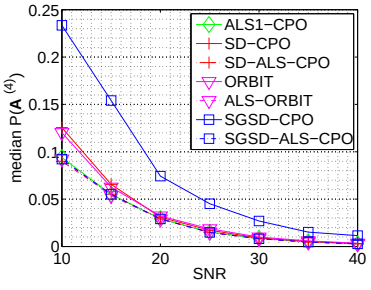
(e) Mean $P\left(\mathbf{A}^{(3)}\right)$.



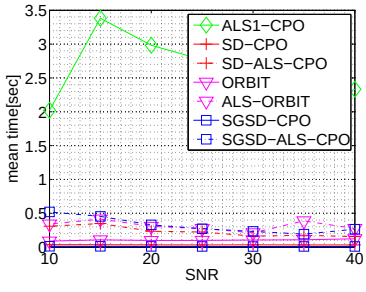
(f) Median $P\left(\mathbf{A}^{(3)}\right)$.



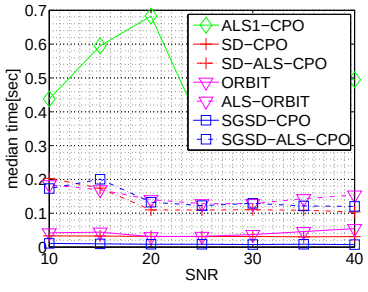
(g) Mean $P\left(\mathbf{A}^{(4)}\right)$.



(h) Median $P\left(\mathbf{A}^{(4)}\right)$.

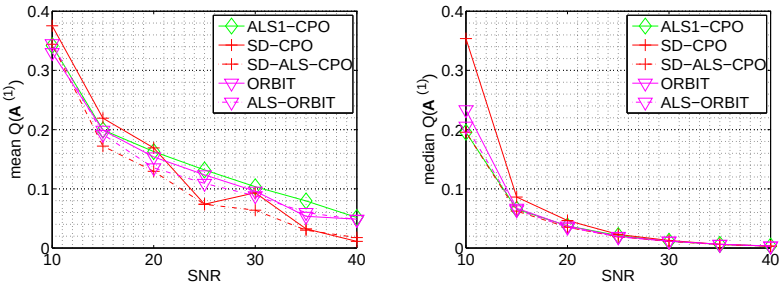


(i) Mean time.



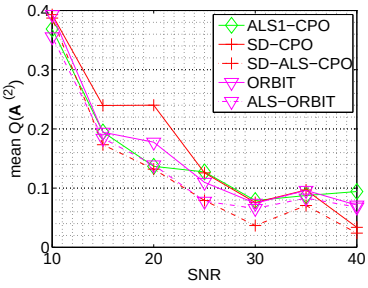
(j) Median time.

FIG. 7.6. *Mean and median $Q\left(\boldsymbol{A}^{(1)}\right)$, $Q\left(\boldsymbol{A}^{(2)}\right)$, $P\left(\boldsymbol{A}^{(3)}\right)$, $P\left(\boldsymbol{A}^{(3)}\right)$ and time values over 100 trials while SNR is varying from* 10 *to* 40 *dB for the real fourth order tensor simulation, case 6.*

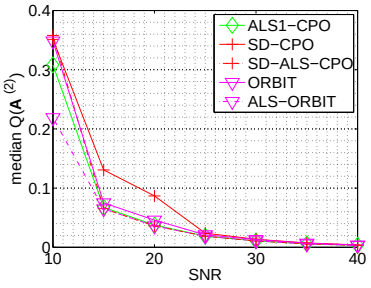


(a) Mean $Q\left(\mathbf{A}^{(1)}\right)$.

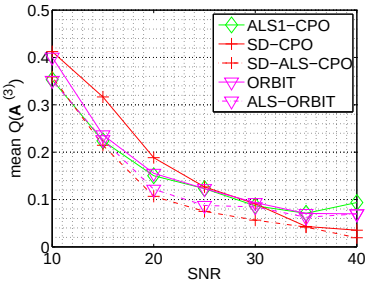
(b) Median $Q\left(\mathbf{A}^{(1)}\right)$.



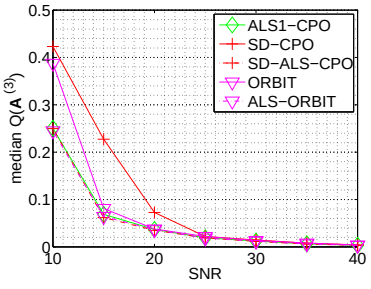
(c) Mean $Q\left(\mathbf{A}^{(2)}\right)$.



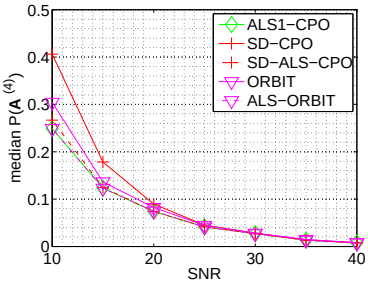
(d) Median $Q\left(\mathbf{A}^{(2)}\right)$.



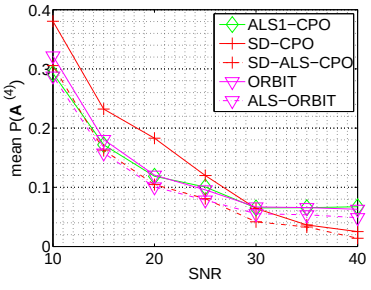
(e) Mean $Q\left(\mathbf{A}^{(3)}\right)$.



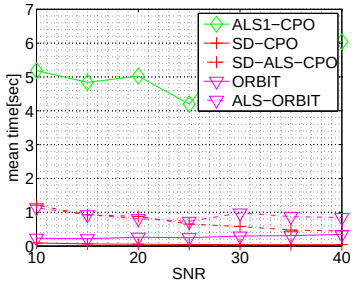
(f) Median $Q\left(\mathbf{A}^{(3)}\right)$.



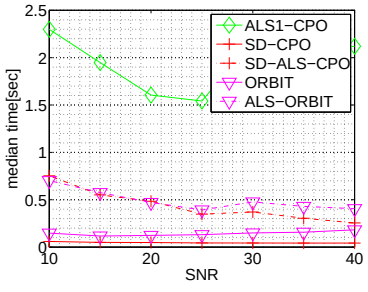
(g) Mean $P\left(\mathbf{A}^{(4)}\right)$.



(h) Median $P\left(\mathbf{A}^{(4)}\right)$.



(i) Mean time.



(j) Median time.

FIG. 7.7. *Mean and median* $Q\left(\boldsymbol{A}^{(1)}\right)$, $Q\left(\boldsymbol{A}^{(2)}\right)$, $Q\left(\boldsymbol{A}^{(3)}\right)$, $P\left(\boldsymbol{A}^{(3)}\right)$ *and time values over 100 trials while SNR is varying from* 10 *to* 40 *dB for the real fourth order tensor simulation, case 7.*