A SEMI-ALGEBRAIC FRAMEWORK FOR APPROXIMATE CP DECOMPOSITIONS VIA JOINT MATRIX DIAGONALIZATION AND GENERALIZED UNFOLDINGS

Florian Roemer, Carola Schroeter, and Martin Haardt

Ilmenau University of Technology, Communications Research Laboratory
P.O. Box 10 05 65, 98684 Ilmenau, Germany, http://tu-ilmenau.de/crl
{florian.roemer, martin.haardt}@tu-ilmenau.de, caroschroeter@gmx.de

***Abstract* — The Canonical Polyadic (CP) decomposition of R-way arrays is a powerful tool in multilinear algebra. Algorithms to compute an approximate CP decomposition from noisy observations are often based on Alternating Least Squares (ALS) which may require a large number of iterations to converge. To avoid this drawback we investigate semi-algebraic approaches that algebraically reformulate the CP decomposition into a set of simultaneous matrix diagonalization (SMD) problems.**

In particular, we propose a SEmi-algebraic framework for approximate CP decompositions via SImultaneous matrix diagonalization (SMD) and generalized unfoldings (SECSI-GU). SECSI-GU combines the benefits of two existing semi-algebraic approaches based on SMDs: the SECSI framework which selects the model estimate from multiple candidates obtained by solving multiple SMDs and the "Semi-Algebraic Tensor Decomposition" (SALT) algorithm which considers a "generalized" unfolding of the tensor in order to enhance the identifiability for tensors with $R > 3$ dimensions. The resulting SECSI-GU framework offers a large number of degrees of freedom to flexibly adapt the performance-complexity trade-off. As we show in numerical simulations, it outperforms SECSI and SALT for tensors with $R > 3$ dimensions.

1. INTRODUCTION

Tensor decompositions represent a powerful emerging tool in signal processing for multidimensional signals. In particular, the decomposition known as R-way Canonical Polyadic (CP) decomposition or Parallel Factor (PARAFAC) analysis [6] or Canonical Decomposition (CANDECOMP) [2] has found applications in areas ranging from psychometrics and chemometrics over communications and array signal processing to numerical mathematics or finance [7]. Its success is mainly due to its inherent essential uniqueness as well as the superior identifiability compared to 2-D (matrix-based) decompositions.

Computing a CP decomposition is a challenging task. In data-driven applications the signal of interest is typically observed under additive noise and hence, we can only compute an *approximate* CP decomposition. A popular class of algorithms for this task is based on the Alternating Least Squares (ALS) procedure [2, 6]. The main drawback of ALS-based algorithms is that the number of required iterations may be very large and they are sensitive to ill-conditioned data.

Therefore, we resort to a semi-algebraic approach where the CP decomposition is algebraically rephrased into a less complex, generic problem such as Simultaneous Matrix Diagonalizations (SMDs). The link between CP and SMDs was already pointed out in [3] and a **SE**mi-algebraic framework for approximate **C**P decompositions based on **SI**multaneous Matrix Diagonalizations (SECSI) was proposed in [12, 11]. SECSI constructs multiple SMDs by considering all R matrix unfoldings and therefore returns several independent estimates for the CP model. The final model estimate is then selected in a subsequent step.

In [9], the "Semi-Algebraic Tensor Decomposition" (SALT) was proposed. It also computes the CP decomposition based on SMDs. However, the main difference is that [9] proposes to use "generalized" matrix unfoldings where the dimensions of the tensor can be assigned to the rows and the columns of the matrix unfolding in an arbitrary manner. For $R > 3$ this approach allows the tensor rank to be much higher than in the SECSI approach from [12].

However, [9] only considers a single SMD constructed from one suitably selected generalized unfolding. Therefore, the main idea of the novel semi-algebraic framework for approximate CP decompositions we propose in this paper is to combine the generalized unfoldings inspired by SALT with the idea to obtain multiple candidate CP models by considering all possible unfoldings as in SECSI. The resulting framework is called SECSI via Generalized Unfoldings (SECSI-GU). SECSI-GU offers a rich flexibility in controlling the complexity-accuracy trade-off since we can freely choose how many SMDs to solve and how to select the final estimates from the multiple candidate solutions. To this end, some criteria are proposed which give rise to three exemplary algorithms inside the SECSI-GU framework. As we demonstrate in numerical simulations, these algorithms outperforms SECSI as well as SALT, in particular for $R > 3$. Note that an alternative and more general way of describing SALT compared to [9] which provides new insights is also developed in this paper.

2. NOTATION

In order to distinguish scalar quantities, vectors, matrices, and tensors, the following convention is used: Scalars are written as italic letters ($a, b, ...$), vectors as lower-case bold faced letters ($\boldsymbol{a}, \boldsymbol{b}, ...$), matrices as upper-case bold faced letters ($\boldsymbol{A}, \boldsymbol{B}, ...$), and tensors as bold faced calligraphic letters ($\boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{B}}, ...$).

A tensor is defined as an R-way array $\boldsymbol{\mathcal{X}} \in \mathbb{C}^{M_1 \times M_2 \ldots \times M_R}$ of size M_r in mode r. Its elements are referenced as $x_{i_1, i_2, \ldots, i_R}$ for $i_r = 1, 2, \ldots, M_r$ and $r = 1, 2, \ldots, R$. The vectors that are obtained by varying the r-th index i_r from 1 to M_r and keeping all other indices fixed are referred to as the r-mode vectors. Moreover, the space spanned by the r-mode vectors is termed r-space of $\boldsymbol{\mathcal{X}}$, the dimension of this vector space is the r-rank of $\boldsymbol{\mathcal{X}}$. Note that in general, the r-ranks (also referred to as the multilinear ranks) of a tensor $\boldsymbol{\mathcal{X}}$ can all be different. A matrix containing all the r-mode vectors is called r-mode (matrix) unfolding of $\boldsymbol{\mathcal{X}}$ and symbolized by $[\boldsymbol{\mathcal{X}}]_{(r)}$. The ordering the columns in $[\boldsymbol{\mathcal{X}}]_{(r)}$ is chosen according to [4]. A "generalized" matrix unfolding of the tensor $\boldsymbol{\mathcal{X}}$ is denoted by $[\boldsymbol{\mathcal{X}}]_{\boldsymbol{\alpha}^{(1)}, \boldsymbol{\alpha}^{(2)}}$, its definition is given in Section 5.

The r-mode product between a tensor $\boldsymbol{\mathcal{X}} \in \mathbb{C}^{M_1 \times M_2 \ldots \times M_R}$ and a matrix $\boldsymbol{U} \in \mathbb{C}^{P_r \times M_r}$ allows to define operations on the tensor that are linear in the r-th mode. It is denoted as $\boldsymbol{\mathcal{X}} \times_r \boldsymbol{U} \in \mathbb{C}^{M_1 \times M_2 \ldots \times P_r \ldots \times M_R}$ and obtained by multiplying all the r-mode vectors of $\boldsymbol{\mathcal{X}}$ from the left-hand side by the matrix $\boldsymbol{U}$, i.e., $[\boldsymbol{\mathcal{X}} \times_r \boldsymbol{U}]_{(r)} = \boldsymbol{U} \cdot [\boldsymbol{\mathcal{X}}]_{(r)}$.

An R-way tensor $\boldsymbol{\mathcal{X}}$ is rank-one if and only if it can be expressed as the outer product of R non-zero vectors, i.e., $\boldsymbol{\mathcal{X}} = \boldsymbol{x}_1 \circ \boldsymbol{x}_2 \circ \ldots \circ \boldsymbol{x}_R$, where $\circ$ represents the outer product operator. Finally, the tensor rank (also referred to as CP-rank or just rank) of $\boldsymbol{\mathcal{X}}$ is equal to r if and only if $\boldsymbol{\mathcal{X}}$ can be decomposed into the sum of r and not less than r rank-one tensors.

To select one element of a tensor, we use the notation $[\boldsymbol{\mathcal{X}}]_{(i,j,k)} = x_{i,j,k}$. Replacing a variable by a colon selects all elements in this mode. For instance, $[\boldsymbol{\mathcal{X}}]_{(:,:,k)}$ is the k-th three-mode slice of $\boldsymbol{\mathcal{X}}$ containing all elements of $\boldsymbol{\mathcal{X}}$ where the third index is equal to k. Likewise, $[\boldsymbol{A}]_{(n,:)} = \boldsymbol{a}_n^{\mathrm{T}}$ represents the n-th row of $\boldsymbol{A}$.

For matrices we use the superscripts $^{\mathrm{T}}, ^{\mathrm{H}}, ^{-1}, ^{+}$ for transposition, Hermitian transposition, matrix inversion, and the Moore-Penrose pseudo inverse, respectively. The transpose of an inverse may be written as $^{-\mathrm{T}}$. Moreover, the asterisk * represents complex conjugation. The Kronecker product between two matrices is expressed by $\boldsymbol{A} \otimes \boldsymbol{B}$, the Khatri-Rao (column-wise Kronecker) product by $\boldsymbol{A} \diamond \boldsymbol{B}$, the Schur (element-wise) product by $\boldsymbol{A} \odot \boldsymbol{B}$, and the inversr Schur (element-wise division) product by $\boldsymbol{A} \oslash \boldsymbol{B}$. Finally, the $\mathrm{vec}\{.\}$ operator rearranges the elements of a matrix or a tensor by stacking them into a long vector, starting with the first dimension and proceedings in ascending order.

3. DATA MODEL

We model our observed data tensor $\boldsymbol{\mathcal{X}} \in \mathbb{C}^{M_1 \times M_2 \ldots \times M_R}$ as a desired signal component $\boldsymbol{\mathcal{X}}_0$ superimposed by additive noise, i.e.,

$$\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{X}}_0 + \boldsymbol{\mathcal{N}}, \tag{1}$$

where the tensor $\boldsymbol{\mathcal{N}} \in \mathbb{C}^{M_1 \times M_2 \ldots \times M_R}$ contains samples of the zero mean additive noise. The rank of the desired signal component $\boldsymbol{\mathcal{X}}_0$ is denoted by d and assumed to be known. Since $\boldsymbol{\mathcal{X}}_0$ is rank-d, it can be decomposed into a sum of d rank-one tensors, i.e.,

$$\boldsymbol{\mathcal{X}}_0 = \sum_{n=1}^{d} \boldsymbol{f}_n^{(1)} \circ \boldsymbol{f}_n^{(2)} \circ \ldots \circ \boldsymbol{f}_n^{(R)}, \tag{2}$$

where $\boldsymbol{f}_n^{(r)} \in \mathbb{C}^{M_r \times 1}$ is the loading vector of the n-th component in the r-th mode. An alternative representation of the CP decomposition in (2) is given by

$$\boldsymbol{\mathcal{X}}_0 = \boldsymbol{\mathcal{I}}_{R,d} \times_1 \boldsymbol{F}^{(1)} \times_2 \boldsymbol{F}^{(2)} \ldots \times_R \boldsymbol{F}^{(R)} \tag{3}$$

where $\boldsymbol{F}^{(r)} = \left[\boldsymbol{f}_1^{(r)}, \ldots, \boldsymbol{f}_d^{(r)}\right] \in \mathbb{C}^{M_r \times d}$ is referred to as the loading matrix in the r-th mode, and $\boldsymbol{\mathcal{I}}_{R,d}$ is the R-way identity tensor of size $d \times d \times \ldots \times d$, which is equal to one if all R indices are equal and zero otherwise.

We refer to the CP decomposition as *degenerate* in the r-th mode if $p_r < d$, i.e., the r-rank of the desired signal component $\boldsymbol{\mathcal{X}}_0$ is less than the tensor rank d. An r-mode degeneracy occurs if the r-mode loading matrix $\boldsymbol{F}^{(r)}$ does not have full column rank. This is always the case for $M_r < d$. For $M_r \geq d$, degeneracies can be caused by linear dependencies among the columns of $\boldsymbol{F}^{(r)}$.

4. CP DECOMPOSITION VIA SMD

In this section we review the SECSI framework (which was called "Closed-Form PARAFAC" in the original conference publications [12, 11]) which exploits the link between the CP decomposition and Simultaneous Matrix Diagonalizations (SMDs).

An SMD problem, which is also called joint approximate eigendecomposition or joint diagonalization, is defined in the following fashion: Let $\boldsymbol{X}_n \in \mathbb{C}^{M \times M}$ be a set of N matrices $n = 1, 2, \ldots, N$. Find a full-rank matrix $\boldsymbol{T} \in \mathbb{C}^{M \times M}$, such that $\boldsymbol{X}_n \approx \boldsymbol{T} \cdot \boldsymbol{\Lambda}_n \cdot \boldsymbol{T}^{-1}$, i.e., all matrices $\boldsymbol{T}^{-1} \cdot \boldsymbol{X} \cdot \boldsymbol{T}$ are approximately diagonal. Efficient algorithms to solve this problem exist, e.g., the algorithm proposed in [5], the JUST algorithm from [10], or the JET algorithm found in [9]. To demonstrate the link between the CP decomposition and SMDs, let us consider the ($R = 3$)-way case for simplicity. It follows from (3) that the one-mode unfolding of $\boldsymbol{\mathcal{X}}_0$ can be written as

$$[\boldsymbol{\mathcal{X}}_0]_{(1)} = \boldsymbol{F}^{(1)} \cdot \left(\boldsymbol{F}^{(2)} \diamond \boldsymbol{F}^{(3)}\right)^{\mathrm{T}}. \tag{4}$$

On the other hand, the three-mode slices of $\boldsymbol{\mathcal{X}}_0$ (i.e., the matrices obtained by varying the first and the second index and fixing the third index) can be written as

$$[\boldsymbol{\mathcal{X}}_0]_{(:,:,k)} = \boldsymbol{F}^{(1)} \cdot \mathrm{diag}\left\{\boldsymbol{F}^{(3)}(k,:)\right\} \cdot \boldsymbol{F}^{(2)^{\mathrm{T}}}, \tag{5}$$

where $\mathrm{diag}\left\{\boldsymbol{F}^{(3)}(k,:)\right\}$ represents a diagonal matrix containing the k-th row of $\boldsymbol{F}^{(3)}$ on the main diagonal. Equation (5) shows the close connection between the CP decomposition and SMDs: all three-mode slices of $\boldsymbol{\mathcal{X}}_0$ (which are full matrices) are jointly diagonalized by $\boldsymbol{F}^{(1)}$ and $\boldsymbol{F}^{(2)}$. However, this SMD is not in the "canonical" form $\boldsymbol{X}_n \approx \boldsymbol{T} \cdot \boldsymbol{\Lambda}_n \cdot \boldsymbol{T}^{-1}$ which was mentioned above. To transform them into this form, two steps are required.

The first step is to transform $\boldsymbol{F}^{(1)}$ and $\boldsymbol{F}^{(2)}$ into square matrices. This is achieved by observing that $\boldsymbol{F}^{(1)} = \boldsymbol{U}_1^{[\mathrm{s}]} \cdot \boldsymbol{T}_1$ and $\boldsymbol{F}^{(2)} = \boldsymbol{U}_2^{[\mathrm{s}]} \cdot \boldsymbol{T}_2$, where $\boldsymbol{T}_1, \boldsymbol{T}_2 \in \mathbb{C}^{d \times d}$ and $\boldsymbol{U}_r^{[\mathrm{s}]} \in \mathbb{C}^{M_r \times d}$ contains an orthonormal basis for the column space of the r-mode unfolding of $\boldsymbol{\mathcal{X}}_0$ (which can be estimated via a Higher-Order SVD of $\boldsymbol{\mathcal{X}}$ [4]). Therefore, we can define a set of matrices $\boldsymbol{S}_{3,k} \in \mathbb{C}^{d \times d}$ via

$$\boldsymbol{S}_{3,k} = \boldsymbol{U}_1^{[\mathrm{s}]^{\mathrm{H}}} \cdot [\boldsymbol{\mathcal{X}}_0]_{(:,:,k)} \cdot \boldsymbol{U}_2^{[\mathrm{s}]^*} = \boldsymbol{T}_1 \cdot \mathrm{diag}\left\{\boldsymbol{F}^{(3)}(k,:)\right\} \cdot \boldsymbol{T}_2^{\mathrm{T}}.$$

The second step is to eliminate $\boldsymbol{T}_1$ or $\boldsymbol{T}_2$, which is easily achieved by multiplying all slices $\boldsymbol{S}_{3,k}$ with the inverse of one particular slice $\boldsymbol{S}_{3,p}$ from the left or the right. We obtain

$$\boldsymbol{S}_{3,k}^{\mathrm{rhs}} = \boldsymbol{S}_{3,k} \cdot \boldsymbol{S}_{3,p}^{-1} = \boldsymbol{T}_1 \cdot \mathrm{diag}\left\{\tilde{\boldsymbol{F}}^{(3)}(k,:)\right\} \cdot \boldsymbol{T}_1^{-1}$$
$$\boldsymbol{S}_{3,k}^{\mathrm{lhs}} = \left(\boldsymbol{S}_{3,p}^{-1} \cdot \boldsymbol{S}_{3,k}\right)^{\mathrm{T}} = \boldsymbol{T}_2 \cdot \mathrm{diag}\left\{\tilde{\boldsymbol{F}}^{(3)}(k,:)\right\} \cdot \boldsymbol{T}_2^{-1},$$

where $\tilde{\boldsymbol{F}}^{(3)} = \boldsymbol{F}^{(3)} \cdot \mathrm{diag}\left\{\boldsymbol{F}^{(3)}(p,:)\right\}^{-1}$. From the SMD of $\boldsymbol{S}_{3,k}^{\mathrm{rhs}}$ we can obtain an estimate for $\boldsymbol{F}^{(1)}$ (via $\boldsymbol{T}_1$) one for $\boldsymbol{F}^{(3)}$ (from the diagonals) and one for $\boldsymbol{F}^{(2)}$ (via a Least-Squares fit using the estimated $\boldsymbol{F}^{(1)}$ and $\boldsymbol{F}^{(3)}$). In a similar manner, we obtain an estimate for all three loading matrices from $\boldsymbol{S}_{3,k}^{\mathrm{lhs}}$. Since the same process can be applied to one- and two-mode slices as well, we obtain overall up to six estimates for all three loading matrices [12].

For $R > 3$ SECSI proceeds by joining tensor dimensions until the resulting tensor is a three-way tensor [11]. In particular, we pick two modes (k, ℓ) with $1 \leq k < \ell \leq R$ and join all modes except for k and ℓ. Each of the $(R-1) \cdot R/2$ combinations yields two SMDs so that the total number of possible SMDs becomes $N_{\mathrm{SMD}} = R(R-1)$.

If the CP is degenerate in one particular mode r then all SMDs connected with $\boldsymbol{F}^{(r)}$ cannot be solved anymore. As long as two non-degenerate modes exist we can select k and ℓ such that two SMDs connected with $\boldsymbol{F}^{(k)}$ and $\boldsymbol{F}^{(\ell)}$ yield two valid estimates of the CP model. The largest possible d is therefore limited to the largest value

d which satisfies $d \leq M_r$ for two values $r \in 1, 2, \ldots, R$. As we show in the next section, this limit can be raised significantly by considering generalized unfoldings for $R > 3$.

5. GENERALIZED UNFOLDINGS

In [9], the "Semi-Algebraic Tensor Decomposition" (SALT) was proposed. SALT also represents a semi-algebraic approach based on SMDs. However, it considers a "generalized" set of matrix unfoldings where the set of indices $(1, 2, \ldots, R)$ is divided into a P-dimensional subset $\boldsymbol{\alpha}^{(1)} = [\alpha_P, \alpha_{P-1}, \ldots, \alpha_1]$ and an $(R-P)$-dimensional subset $\boldsymbol{\alpha}^{(2)} = [\alpha_R, \alpha_{R-1}, \ldots, \alpha_{P+1}]$ where $1 \leq P < R$. Then, the first P indices are arranged into the rows and the remaining $R-P$ indices into the columns of the generalized unfolding represented by $[\boldsymbol{\mathcal{X}}]_{\boldsymbol{\alpha}^{(1)},\boldsymbol{\alpha}^{(2)}}$. It is easy to show that for the noise-free tensor $\boldsymbol{\mathcal{X}}_0$, this unfolding satisfies

$$[\boldsymbol{\mathcal{X}}_0]_{\boldsymbol{\alpha}^{(1)},\boldsymbol{\alpha}^{(2)}} = \left(\boldsymbol{F}^{(\alpha_1)} \diamond \boldsymbol{F}^{(\alpha_2)} \diamond \ldots \diamond \boldsymbol{F}^{(\alpha_P)}\right) \cdot \left(\boldsymbol{F}^{(\alpha_{P+1})} \diamond \boldsymbol{F}^{(\alpha_{P+2})} \diamond \ldots \diamond \boldsymbol{F}^{(\alpha_R)}\right)^{\mathrm{T}} \quad (6)$$

Note that the r-mode matrix unfoldings used in SECSI represent the special case where $P = 1$, $\alpha^{(1)} = r$ and $\boldsymbol{\alpha}^{(2)} = [r-1, \ldots, 1, R, \ldots, r+1]$. The generalized unfoldings differ from r-mode matrix unfoldings only for $R > 3$. Hence, SALT has a particular benefit compared to SECSI for $R > 3$.

Based on the generalized unfoldings, the loading matrices $\boldsymbol{F}^{(r)}$ can be permuted into an arbitrary order. In order to obtain estimates for $\boldsymbol{F}^{(r)}$ from the generalized unfoldings, the first step is to assign the different modes into three non-empty groups such that

$$\begin{aligned}\boldsymbol{F}_{\mathrm{A}} &= \boldsymbol{F}^{(\alpha_1)} \diamond \boldsymbol{F}^{(\alpha_2)} \diamond \ldots \diamond \boldsymbol{F}^{(\alpha_t)} \in \mathbb{C}^{M_{\mathrm{A}} \times d} \\ \boldsymbol{F}_{\mathrm{B}} &= \boldsymbol{F}^{(\alpha_{t+1})} \diamond \boldsymbol{F}^{(\alpha_{t+2})} \diamond \ldots \diamond \boldsymbol{F}^{(\alpha_P)} \in \mathbb{C}^{M_{\mathrm{B}} \times d} \\ \boldsymbol{F}_{\mathrm{C}} &= \boldsymbol{F}^{(\alpha_{P+1})} \diamond \boldsymbol{F}^{(\alpha_2)} \diamond \ldots \diamond \boldsymbol{F}^{(\alpha_R)} \in \mathbb{C}^{M_{\mathrm{C}} \times d},\end{aligned}$$

where $M_{\mathrm{A}} = \prod_{r=1}^{t} M_{\alpha_r}$, $M_{\mathrm{B}} = \prod_{r=t+1}^{P} M_{\alpha_r}$, $M_{\mathrm{C}} = \prod_{r=P+1}^{R} M_{\alpha_r}$, and we have $1 \leq t < P < R$. The generalized unfolding in (6) can then be written as

$$[\boldsymbol{\mathcal{X}}_0]_{\boldsymbol{\alpha}^{(1)},\boldsymbol{\alpha}^{(2)}} = (\boldsymbol{F}_{\mathrm{A}} \diamond \boldsymbol{F}_{\mathrm{B}}) \cdot \boldsymbol{F}_{\mathrm{C}}^{\mathrm{T}}. \quad (7)$$

Note that this partitioning is more general than the proposal in [9] where one of the three groups always contains only one mode ($r = Q$). Using the definition of the Khatri-Rao product, (7) can be rewritten as

$$[\boldsymbol{\mathcal{X}}_0]_{\boldsymbol{\alpha}^{(1)},\boldsymbol{\alpha}^{(2)}} = \begin{bmatrix} \boldsymbol{F}_{\mathrm{B}} \cdot \mathrm{diag}\left\{\boldsymbol{F}_{\mathrm{A}}(1,:)\right\} \\ \vdots \\ \boldsymbol{F}_{\mathrm{B}} \cdot \mathrm{diag}\left\{\boldsymbol{F}_{\mathrm{A}}(M_{\mathrm{A}},:)\right\} \end{bmatrix} \cdot \boldsymbol{F}_{\mathrm{C}}^{\mathrm{T}}. \quad (8)$$

In order to proceed, we need an estimate of the column space and the row space of $[\boldsymbol{\mathcal{X}}_0]_{\boldsymbol{\alpha}^{(1)},\boldsymbol{\alpha}^{(2)}}$. To this end, let the truncated SVD of $[\boldsymbol{\mathcal{X}}]_{\boldsymbol{\alpha}^{(1)},\boldsymbol{\alpha}^{(2)}}$ be given by

$$[\boldsymbol{\mathcal{X}}]_{\boldsymbol{\alpha}^{(1)},\boldsymbol{\alpha}^{(2)}} = \hat{\boldsymbol{U}}^{[\mathrm{s}]} \cdot \hat{\boldsymbol{\Sigma}}^{[\mathrm{s}]} \cdot \hat{\boldsymbol{V}}^{[\mathrm{s}]\mathrm{H}}, \quad (9)$$

where $\hat{\boldsymbol{U}}^{[\mathrm{s}]} \in \mathbb{C}^{M_{\mathrm{A}} \cdot M_{\mathrm{B}} \times d}$, $\hat{\boldsymbol{V}}^{[\mathrm{s}]} \in \mathbb{C}^{M_{\mathrm{C}} \times d}$, and $\hat{\boldsymbol{\Sigma}}^{[\mathrm{s}]} \in \mathbb{C}^{d \times d}$ Then, the column space of $\hat{\boldsymbol{U}}^{[\mathrm{s}]}$ is an estimate of the column space of $[\boldsymbol{\mathcal{X}}_0]_{\boldsymbol{\alpha}^{(1)},\boldsymbol{\alpha}^{(2)}}$. On the other hand, from (7) it is easy to see that $\boldsymbol{F}_{\mathrm{A}} \diamond \boldsymbol{F}_{\mathrm{B}} \in \mathbb{C}^{M_{\mathrm{A}} \cdot M_{\mathrm{B}} \times d}$ also provides a basis for the column space of $[\boldsymbol{\mathcal{X}}_0]_{\boldsymbol{\alpha}^{(1)},\boldsymbol{\alpha}^{(2)}}$. Therefore, we can write

$$\begin{aligned}\boldsymbol{F}_{\mathrm{A}} \diamond \boldsymbol{F}_{\mathrm{B}} &\approx \hat{\boldsymbol{U}}^{[\mathrm{s}]} \cdot \boldsymbol{T} \\ \Rightarrow (\boldsymbol{F}_{\mathrm{A}} \diamond \boldsymbol{F}_{\mathrm{B}}) \cdot \boldsymbol{T}^{-1} &\approx \hat{\boldsymbol{U}}^{[\mathrm{s}]} \quad (10)\end{aligned}$$

Inserting (8) into (10), we have

$$\begin{bmatrix} \boldsymbol{F}_{\mathrm{B}} \cdot \mathrm{diag}\left\{\boldsymbol{F}_{\mathrm{A}}(1,:)\right\} \cdot \boldsymbol{T}^{-1} \\ \vdots \\ \boldsymbol{F}_{\mathrm{B}} \cdot \mathrm{diag}\left\{\boldsymbol{F}_{\mathrm{A}}(M_{\mathrm{A}},:)\right\} \cdot \boldsymbol{T}^{-1} \end{bmatrix} \approx \hat{\boldsymbol{U}}^{[\mathrm{s}]} = \begin{bmatrix} \hat{\boldsymbol{U}}_1^{[\mathrm{s}]} \\ \vdots \\ \hat{\boldsymbol{U}}_{M_{\mathrm{A}}}^{[\mathrm{s}]} \end{bmatrix}. \quad (11)$$

In other words (11) shows that $\hat{\boldsymbol{U}}^{[\mathrm{s}]}$ can be partitioned into M_{A} blocks $\hat{\boldsymbol{U}}_m^{[\mathrm{s}]}$ of size $M_{\mathrm{B}} \times d$ for $m = 1, 2, \ldots, M_{\mathrm{A}}$ which are approximately jointly diagonalized by $\boldsymbol{F}_{\mathrm{B}}$ and $\boldsymbol{T}^{-1}$. In order to eliminate the matrix $\boldsymbol{F}_{\mathrm{B}} \in \mathbb{C}^{M_{\mathrm{B}} \times d}$, [9] proposes to compute a new set of matrices $\boldsymbol{\Gamma}_{p,m}$ defined as $\boldsymbol{\Gamma}_{p,m} = \hat{\boldsymbol{U}}_p^{[\mathrm{s}]^+} \cdot \hat{\boldsymbol{U}}_m^{[\mathrm{s}]}$. If $\boldsymbol{F}_{\mathrm{B}}$ has full column rank (which implies the condition $M_{\mathrm{B}} \geq d$) and the p-th row of $\boldsymbol{F}_{\mathrm{A}}$ does not contain any zeros this yields

$$\begin{aligned}\boldsymbol{\Gamma}_{p,m} &= \boldsymbol{T} \cdot \mathrm{diag}\left\{\boldsymbol{F}_{\mathrm{A}}(p,:)\right\}^{-1} \cdot \boldsymbol{F}_{\mathrm{B}}^{+} \cdot \boldsymbol{F}_{\mathrm{B}} \cdot \mathrm{diag}\left\{\boldsymbol{F}_{\mathrm{A}}(m,:)\right\} \cdot \boldsymbol{T}^{-1} \\ &= \boldsymbol{T} \cdot \mathrm{diag}\left\{\boldsymbol{F}_{\mathrm{A}}(m,:) \oslash \boldsymbol{F}_{\mathrm{A}}(p,:)\right\} \cdot \boldsymbol{T}^{-1}. \quad (12)\end{aligned}$$

Note that [9] does not further specify the choice of p and m. The largest possible set is given by considering all values $1 \leq p < m \leq M_{\mathrm{A}}$ for which $\hat{\boldsymbol{U}}_p^{[\mathrm{s}]}$ has full column rank This yields up to $M_{\mathrm{A}} \cdot (M_{\mathrm{A}} - 1)/2$ matrix slices.

After the joint diagonalization of $\boldsymbol{\Gamma}_{p,m}$ for all p, m we have an estimate for $\boldsymbol{T}$. Based on $\boldsymbol{T}$, we can obtain estimates for $\boldsymbol{F}_{\mathrm{A}} \diamond \boldsymbol{F}_{\mathrm{B}}$ and $\boldsymbol{F}_{\mathrm{C}}$, namely

$$\hat{\boldsymbol{U}}^{[\mathrm{s}]} \cdot \boldsymbol{T} \approx \boldsymbol{F}_{\mathrm{A}} \diamond \boldsymbol{F}_{\mathrm{B}} = \boldsymbol{F}^{(\alpha_1)} \diamond \ldots \diamond \boldsymbol{F}^{(\alpha_P)} \quad (13)$$

$$\hat{\boldsymbol{V}}^{[\mathrm{s}]^*} \cdot \hat{\boldsymbol{\Sigma}}^{[\mathrm{s}]} \cdot \boldsymbol{T}^{-\mathrm{T}} \approx \boldsymbol{F}_{\mathrm{C}} = \boldsymbol{F}^{(\alpha_{P+1})} \diamond \ldots \diamond \boldsymbol{F}^{(\alpha_R)}, \quad (14)$$

which is evident by comparing (7) and (9). Therefore, to obtain the loading matrices $\boldsymbol{F}^{(r)}$ from (13) and (14), we need to factorize a P-fold and an $(R-P)$-fold Khatri-Rao product. Therefore, we need to express each column as the Kronecker product of P and $(R-P)$ vectors, respectively. However, this is equivalent to finding a rank-one approximation of a P-way and an $(R-P)$-way array. For a two-way array, the truncated SVD provides the Least Squares optimal solution while for N-way arrays ($N > 2$), the truncated HOSVD can be used (which is not optimal but close to optimal).

Note that the proposed approach requires three conditions: (1) $\boldsymbol{F}_{\mathrm{A}}$ needs to possess at least one row which does not contain any zeros; (2) $\min\{M_{\mathrm{A}} \cdot M_{\mathrm{B}}, M_{\mathrm{C}}\} \geq d$ (for the rank-d SVD to be well-defined); (3) $M_{\mathrm{B}} \geq d$ (for $\boldsymbol{F}_{\mathrm{B}}$) to be invertible. The last two conditions can be summarized into one condition as $\min\{M_{\mathrm{B}}, M_{\mathrm{C}}\} \geq d$.

The original SALT algorithm in [9] can be summarized into the following steps:

1. Select one generalized unfolding $[\boldsymbol{\mathcal{X}}]_{\boldsymbol{\alpha}^{(1)},\boldsymbol{\alpha}^{(2)}}$. The authors in [9] propose to choose the one which maximizes $\min\{M_{\mathrm{B}}, M_{\mathrm{C}}\}$. Note that this implies that only one factor is associated to the group $\boldsymbol{F}_{\mathrm{A}}$.
2. Compute the rank-d truncated SVD of $[\boldsymbol{\mathcal{X}}]_{\boldsymbol{\alpha}^{(1)},\boldsymbol{\alpha}^{(2)}}$.
3. Partition the matrix of the d estimated dominant left singular vectors $\hat{\boldsymbol{U}}^{[\mathrm{s}]}$ into M_{A} blocks $\hat{\boldsymbol{U}}_m^{[\mathrm{s}]}$ of size $M_{\mathrm{B}} \times d$.
4. Compute the matrices $\boldsymbol{\Gamma}_{p,m} = \hat{\boldsymbol{U}}_p^{[\mathrm{s}]^+} \cdot \hat{\boldsymbol{U}}_m^{[\mathrm{s}]}$ for all $1 \leq p < m \leq M_{\mathrm{A}}$.
5. Obtain $\boldsymbol{T}$ via a simultaneous matrix diagonalization (SMD) of $\boldsymbol{\Gamma}_{p,m}$.
6. Recover estimates for the loading matrices $\boldsymbol{F}^{(r)}$ by performing a Least-Squares Khatri-Rao factorization [13] of $\hat{\boldsymbol{U}}^{[\mathrm{s}]} \cdot \boldsymbol{T}$ and $\hat{\boldsymbol{V}}^{[\mathrm{s}]^*} \cdot \hat{\boldsymbol{\Sigma}}^{[\mathrm{s}]} \cdot \boldsymbol{T}^{-\mathrm{T}}$ via a sequence of d rank-one approximations [8].

6. COMBINED APPROACH: SECSI-GU

The SECSI-GU framework we propose in this paper can be seen as an extension of the SALT scheme [9] discussed in Section 5 in various aspects by incorporating the concepts of the SECSI framework discussed in Section 4.

The first extension we propose is to consider not only one SMD based on one partitioning of the tensor modes into $\boldsymbol{F}_{\mathrm{A}}$, $\boldsymbol{F}_{\mathrm{B}}$, and $\boldsymbol{F}_{\mathrm{C}}$, but the set of all possible partitionings. To enumerate these, we have to find all possible partitions of a set of R numbers into three non-empty subsets. We do not count permutations inside the set multiple times, since these yield identical estimates for the loading matrices. The maximum number of such partitions can be computed as

$$N_{\mathrm{SMD}} = \sum_{k=1}^{R-2} \binom{R}{k} \cdot \sum_{\ell=1}^{R-k-1} \binom{R-k}{\ell} = 3^R - 3 \cdot 2^R + 3. \qquad (15)$$

Note that each partitioning we consider needs to satisfy the identifiability condition $\min\{M_{\mathrm{B}}, M_{\mathrm{C}}\} \geq d$. If a particular partitioning does not satisfy it, it is simply excluded from the set. The SECSI-GU framework remains applicable as long as there is at least one partitioning which satisfies $\min\{M_{\mathrm{B}}, M_{\mathrm{C}}\} \geq d$. Hence, it has the same identifiability limits as the SALT algorithm.

A second modification to [9] we propose is to apply pivoting when establishing the matrix slices $\boldsymbol{\Gamma}_{p,m}$ for the SMD. This means that we only consider one value of p so that we obtain only $M_{\mathrm{A}} - 1$ matrix slices for $m = 1, 2, \ldots, p-1, p+1, \ldots, M_{\mathrm{A}}$. Moreover, we choose p according to

$$p = \underset{n=1,2,\ldots,M_{\mathrm{A}}}{\arg\min} \ \mathrm{cond}\{\hat{\boldsymbol{U}}_n^{[\mathrm{s}]}\}, \qquad (16)$$

where $\mathrm{cond}\{\cdot\}$ represents the conditioning number. Note that this approach has a number of advantages. First of all, the number of matrix slices is smaller ($M_{\mathrm{A}} - 1$ compared to $M_{\mathrm{A}}(M_{\mathrm{A}} - 1)$) which lowers the computational complexity. Secondly, it avoids to invert matrix slices which are almost rank-deficient. Note that [9] argues that if $\boldsymbol{F}_{\mathrm{A}}$ contains zeros in the p-th row, the corresponding matrix slice $\hat{\boldsymbol{U}}_p^{[\mathrm{s}]}$ is rank-deficient (which is evident from (11)) and hence excluded from the set. This is true in the absence of noise, however, if additive noise is present the rank of $\hat{\boldsymbol{U}}_p^{[\mathrm{s}]}$ is full almost surely and it would not be easy to detect the fact that $\boldsymbol{F}_{\mathrm{A}}$ contains zeros in the p-th row. This is automatically taken care of by the proposed pivoting approach.

Therefore, the SECSI-GU framework allows us to establish between 1 and N_{SMD} SMDs and to obtain one estimate for all loading matrices from each SMD. As an additional extension to [9] we therefore propose to apply heuristic selection criteria on order to decide which SMDs to solve and how to select the final solution. Some examples for possible heuristics are the following:

- Use the problem dimensions. This is for instance done in SALT [9], where the SMD is chosen that maximizes $\min\{M_{\mathrm{B}}, M_{\mathrm{C}}\}$.
- Store the conditioning numbers (CON) of the pivot slices $\mathrm{cond}\{\hat{\boldsymbol{U}}_p^{[\mathrm{s}]}\}$. SMDs where the pivots have a higher conditioning number are expected to perform worse and hence, this information can even be used to exclude SMDs prior to solving them.
- The SMD provides an approximate joint diagonalization, and thus, the residual (RES) norm of all the remaining off-diagonal elements after the diagonalization can be used as a criterion to judge the reliability of an estimate.

Method	Fig. 1	Fig. 2
SECSI-GU CON	0.160 s	0.179 s
SECSI-GU RES	2.676 s	10.041 s
SECSI-GU REC	2.678 s	10.110 s
SECSI CON	0.106 s	1.310 s
SECSI RES	1.359 s	1.288 s
SECSI REC	1.363 s	1.288 s
ALS	44.621 s	0.512 s
SALT	0.017 s	0.062 s

Table 1. Mean run times

- After a complete set of loading matrices have been estimated, we can reconstruct (REC) the tensor and compare it to the original tensor. The norm of the difference between the two provides a quality measure as well, since it judges how well the estimated model explains the observed data tensor.

Note that these are only examples and many more heuristics that are specifically tailored to certain applications could easily be devised. Moreover, various heuristics can be combined, which results in a great flexibility to design various algorithms with a flexible control over the performance-accuracy trade-off.

7. SIMULATION RESULTS

In this section we demonstrate the SECSI-GU framework based on some numerical simulation results. As examples, we study three different heuristic algorithms:

- **CON**: CON solves only one SMD, namely, the one belonging to the slices with the best condition number.
- **RES**: RES solves all SMDs and then selects the final estimates from the SMD which yields the lowest residual after the diagonalization.
- **REC**: REC solves all SMDs and then selects the final estimates from the SMD which provides the lowest reconstruction error compared to the observed data tensor.

Note that REC only combines estimates that originate from the same SMD. We do not investigate combining estimates from different SMDs, such as the "best matching" scheme which was proposed for SECSI [12]. The reason is that with up to $3^R - 3 \cdot 2^R + 3$ estimates for each of the R loading matrices, testing all possible combinations requires to compute $\mathcal{O}(3^{R^2})$ recombinations ($1.7 \cdot 10^6$ for $R = 4$, $7.5 \cdot 10^{10}$ for $R = 5$, $2.5 \cdot 10^{16}$ for $R = 6$, and so on) which is computationally prohibitive.

Comparing the three heuristics we expect REC to yield the lowest reconstruction error while having the highest computational complexity. RES is slightly less complex (we do not have to compute the reconstructed tensors) and CON has the lowest complexity (since only one SMD is solved).

For the simulations we draw our loading matrices $\boldsymbol{F}^{(r)}$ randomly via $\boldsymbol{F}^{(r)} = \boldsymbol{F}^{(r)}_{\mathrm{white}} \cdot \boldsymbol{R}(\rho_r)$, where $\boldsymbol{F}^{(r)}_{\mathrm{white}}$ is a matrix containing zero mean circularly symmetric complex Gaussian (ZMCSCG) random variables with variance one. Moreover, $\boldsymbol{R}(\rho) = (1-\rho) \cdot \boldsymbol{I}_d + \frac{\rho}{d} \cdot \mathbf{1}_{d \times d}$ where $\mathbf{1}_{d \times d}$ is a $d \times d$ matrix filled with ones, i.e., $\boldsymbol{R}(\rho)$ is an identity matrix for $\rho = 0$ and its conditioning number increases as ρ approaches 1. Moreover, the observed data tensor $\boldsymbol{\mathcal{X}}$ is generated according to (1), where $\boldsymbol{\mathcal{N}}$ contains ZMCSCG random variables with variance σ_{n}^2.

For comparison we depict three algorithms from the SECSI framework, namely "CON", "RES", "REC", which correspond to similar heuristics as the ones proposed here. Finally, we depict an ALS-based scheme as a reference for which the "PARAFAC" as well as the "COMFAC" algorithms from the N-way toolbox v3.10 [1] are used.

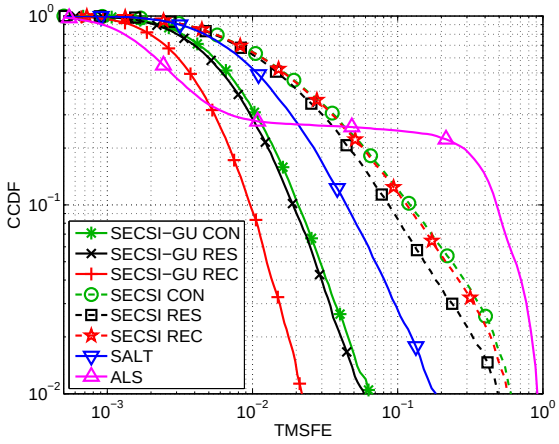

Fig. 1. CCDF of the TMSFE for a 4-D tensor of size $4 \times 7 \times 15 \times 6$, $d = 3$, and $\rho_1 = \rho_2 = 0.98$, $\rho_3 = \rho_4 = 0$.

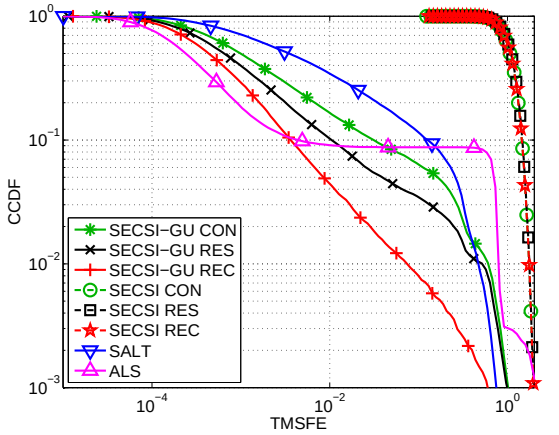

Fig. 2. CCDF of the TMSFE for a 5-D tensor of size $4 \times 6 \times 4 \times 6 \times 4$, $d = 6$, and $\rho_2 = \rho_4 = 0.98$, $\rho_1 = \rho_3 = \rho_5 = 0$.

To assess the estimation accuracy we compute the "total relative mean square factor error" (TMSFE) defined as

$$\text{TMSFE} = \mathbb{E}\left\{ \sum_{r=1}^{R} \min_{\boldsymbol{P}_r \in \mathcal{M}_{\text{PD}}(d)} \frac{\left\| \hat{\boldsymbol{F}}_r \cdot \boldsymbol{P}_r - \boldsymbol{F}^{(r)} \right\|_{\text{F}}^2}{\left\| \boldsymbol{F}^{(r)} \right\|_{\text{F}}^2} \right\},$$

where $\hat{\boldsymbol{F}}_r$ is the estimate of $\boldsymbol{F}_r$. Moreover, $\mathcal{M}_{\text{PD}}(d)$ denotes the set of permuted diagonal matrices which is needed to account for the permutation and scaling ambiguities in the estimates for $\boldsymbol{F}^{(r)}$.

For the first simulation result shown in Figure 1, we consider a 4-D tensor of size $4 \times 7 \times 15 \times 6$ with a model order of $d = 3$ and set $\sigma_{\text{n}} = 0.0178$. The coefficients ρ_r are chosen as $\rho_1 = \rho_2 = 0.98$, $\rho_3 = \rho_4 = 0$, i.e., the loading matrices in the first and the second mode are badly conditioned. Figure 1 shows an estimate for the complementary cumulative distribution function (CCDF) of the TMSFE. In Figure 2 we depict the simulation result for a 5-D tensor of size $4 \times 6 \times 4 \times 6 \times 4$ with a model order of $d = 6$, i.e., it is degenerate in the first, the third, and the fifth mode. Moreover, we set $\rho_2 = \rho_4 = 0.98$, $\rho_1 = \rho_3 = \rho_5 = 0$. The average run time[1]. is shown in Table 1. Both simulation results verify that ALS yields outliers which correspond to cases where a local minimum was reached. Moreover, while SALT is very fast, the estimation accuracy can be improved by considering a different heuristic (such as CON) or by solving multiple SMDs (as in RES and REC) and selecting the final estimate in a subsequent step. Finally, SECSI-GU clearly outperforms the SECSI approach which yields a higher TMSFE in all cases. Note that the heuristics presented here are only examples, further algorithms inside the SECSI-GU framework can be devised easily.

[1]Run times have been evaluated on an Intel X 5550 2.67 GHz machine running Linux (Red Hat 3.4.6-9, kernel version 2.6.9-67.0.4.ELsmp) and MATLAB R2012a (7.14.0.739) 64-bit. While absolute numbers will vary greatly from machine to machine, the relative comparison should remain about the same.

8. CONCLUSIONS

In this paper a novel framework for semi-algebraic approximate CP decompositions via matrix diagonalization is introduced. The framework combines the advantages of two existing approaches: the Semi-Algebraic CP decomposition based on Simultaneous Matrix Diagonalizations (SECSI) framework and the Semi-Algebraic Tensor Decomposition (SALT) algorithm. It is therefore called SECSI via Generalized Unfoldings (SECSI-GU). We extend SALT by considering multiple instead of one one generalized unfolding. Thereby, several candidate estimates can be found and the final estimate is selected in a subsequent step. Moreover, we introduce pivoting to construct the matrix slices for the SMD, which avoids inverting badly conditioned matrix slices and lowers the computational complexity.

Our SECSI-GU framework features a large flexibility in choosing which and how many SMDs to solve and how to select the final estimate. Thereby, we can design algorithms that are very accurate or have a very low computational complexity, depending on the requirements of the application. We have demonstrated this flexibility with three exemplary algorithms in numerical simulations. As we show, SECSI-GU outperforms SECSI and SALT for tensors with $R > 3$ dimensions.

REFERENCES

[1] C. A. Andersson and R. Bro, "The N-way toolbox for MATLAB", *Chemometrics and Intelligent Laboratory Systems*, vol. 52, no. 1, pp. 1–4, 2000.

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

[3] L. de Lathauwer, "A link between the canonical decomposition in multilinear algebra and simultaneous matrix diagonalizations", *SIAM Journal Matrix. Anal. Appl.*, vol. 28, no. 3, pp. 642–666, 2006.

[4] L. de Lathauwer, B. de Moor, and J. Vanderwalle, "A multilinear singular value decomposition", *SIAM J. Matrix Anal. Appl.*, vol. 21, no. 4, pp. 1253–1278, 2000.

[5] T. Fu and X. Gao, "Simultaneous diagonalization with similarity transformation for non-defective matrices", in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2006)*, Toulouse, France, May 2006.

[6] R. A. Harshman, "Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multi-modal factor analysis", *UCLA working papers in phonetics*, vol. 16, pp. 1–84, 1970.

[7] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications", *SIAM Review*, vol. 51, no. 3, pp. 455–500, Sept. 2009.

[8] C. F. Van Loan and N. Pitsianis, "Approximation with Kronecker products", *Linear Algebra for Large-Scale and Real-Time Applications*, pp. 293–314, 1993.

[9] X. Luciani and L. Albera, "Semi-algebraic canonical decomposition of multi-way arrays and joint eigenvalue decomposition", in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2011)*, Prague, Czech Republic, May 2011.

[10] K. Abed-Meraim R. Iferroudjenea and A. Belouchrani, "A new Jacobi-like method for joint diagonalization of arbitrary non-defective matrices", *Applied Mathematics and Computation*, vol. 211, no. 2, pp. 363–373, 2009.

[11] F. Roemer and M. Haardt, "A closed-form solution for multilinear PARAFAC decompositions", in *Proc. 5-th IEEE Sensor Array and Multich. Sig. Proc. Workshop (SAM 2008)*, pp. 487 – 491, Darmstadt, Germany, July 2008.

[12] F. Roemer and M. Haardt, "A closed-form solution for parallel factor (PARAFAC) analysis", in *Proc. IEEE Int. Conf. Acoustics, Speech and Sig. Proc. (ICASSP 2008)*, pp. 2365 – 2368, Las Vegas, NV, Apr. 2008.

[13] F. Roemer and M. Haardt, "Tensor-Based channel estimation (TENCE) and iterative refinements for Two-Way relaying with multiple antennas and spatial reuse", *IEEE Trans. Sig. Proc.*, vol. 58, pp. 5720 – 5735, Nov. 2010.