IET Signal Processing

Invited Paper

IET Journals
The Institution of Engineering and Technology

Tensor methods for multisensor signal processing

ISSN 1751-9675
Received on 6th October 2020
Revised 16th November 2020
Accepted on 24th November 2020
E-First on 2nd March 2021
doi: 10.1049/iet-spr.2020.0373
www.ietdl.org

Sebastian Miron[1] ✉, Yassine Zniyed[1], Rémy Boyer[2], André Lima Ferrer de Almeida[3], Gérard Favier[4], David Brie[1], Pierre Comon[5]

[1]*Université de Lorraine, CNRS, CRAN, F-54000 Nancy, France*
[2]*Université de Lille, CNRS, CRIStAL, 59655 Lille, France*
[3]*Department of Teleinformatics Engineering, Federal University of Ceará, Fortaleza, CE 60440–900, Brazil*
[4]*Université de Côte d'Azur, CNRS, I3S Laboratory, Sophia Antipolis, France*
[5]*Université Grenoble Alpes, CNRS, GIPSA-Lab, 38000 Grenoble, France*
✉ *E-mail: sebastian.miron@univ-lorraine.fr*

Abstract: Over the last two decades, tensor-based methods have received growing attention in the signal processing community. In this work, the authors proposed a comprehensive overview of tensor-based models and methods for multisensor signal processing. They presented for instance the Tucker decomposition, the canonical polyadic decomposition, the tensor-train decomposition (TTD), the structured TTD, including nested Tucker train, as well as the associated optimisation strategies. More precisely, they gave synthetic descriptions of state-of-the-art estimators as the alternating least square (ALS) algorithm, the high-order singular value decomposition (HOSVD), and of more advanced algorithms as the rectified ALS, the TT-SVD/TT-HSVD and the Joint dImensionally Reduction and Factor retrieval Estimator scheme. They illustrated the efficiency of the introduced methodological and algorithmic concepts in the context of three important and timely signal processing-based applications: the direction-of-arrival estimation based on sensor arrays, multidimensional harmonic retrieval and multiple-input–multiple-output wireless communication systems.

1 Introduction

Tensors are, roughly speaking, generalisations of matrices to more than two dimensions (orders larger than 2). Tensors present several advantages over matrices, such as uniqueness of rank-decomposition under mild conditions, or preservation of the local structure in N-way data processing. This is what motivated the advent of tensors in data analysis (phonetics [1], psychometrics [2]) in the early 1970s and later on, in the signal processing community. Over the last 15 years approximately, the computer science community actively participated in the popularisation of tensors as data analysis tools. Due to their ability to model highly structured data, tensors quickly became a staple of data mining and machine learning fields (see, e.g. [3, 4]).

In multisensor signal processing, tensors are being used for almost two decades in various applications such as direction of arrival (DOA) estimation [5], harmonic retrieval [6], communications systems [7] or radar [8]. However, despite a large number of publications on this topic, we were unable to find in the literature a comprehensive 'point of entry' article on the use of tensors in multisensor signal processing. The aim of this work is to fill this need. This paper does not intend to draw an exhaustive picture of the existing methods and models but rather provides an overview of the principles and main ideas used in tensor-based multisensor signal processing. We illustrate our presentation by three popular multisensor applications: DOA estimation, multidimensional harmonic retrieval (MHR) and multiple-input–multiple-output (MIMO) systems design.

The paper is organised as follows: in Section 2 we define the main tensor concepts and notations and in Section 3 we introduce some of the most commonly used tensor decompositions and algorithms. These tensor decompositions are then applied to the DOA estimation problem (Section 4), MHR (Section 5) and MIMO wireless communication system (Section 6). Conclusions are drawn in Section 7.

2 Tensor preliminaries: notations and operations

Tensors are *multilinear maps* between two sets of linear spaces [9]. Once these linear spaces are fixed as well as their bases, a tensor is characterised by its array of coordinates. In the present contribution, we shall assume that this is the case and shall mainly work on array coordinates. Hence, with the usual abuse of vocabulary, we shall assimilate *tensors* with their array representations. The number of dimensions (indices) of such an array is called the *order* of the tensor.

Tensors are typeset with a bold calligraphic font (e.g., $\mathcal{T}$), matrices in bold italic caps (e.g. $\boldsymbol{Q}$); the elements of tensors or matrices are typeset as $\mathcal{T}_{i,j,k}$ and $Q_{i,j}$, respectively; vectors are denoted with bold italic lower cases (e.g. $\boldsymbol{v}$), with entries v_i. The Frobenius norm of a matrix/tensor is denoted $\|\cdot\|_{\mathrm{F}}$; $(.)^{\mathrm{T}}$, $(.)^{\mathrm{H}}$ and $(.)^{\dagger}$ symbolise the matrix transpose, conjugate transpose and the Moore–Penrose inverse, respectively, and ' $\star$ ' denotes the convolution operator.

The tensor (outer) product between tensors, matrices, or vectors, is denoted by $\otimes$, as in e.g. $\mathcal{T} = \boldsymbol{v} \otimes \boldsymbol{M}$. The contraction on the pth index of a tensor is denoted as $\underset{p}{\bullet}$; by convention, when contracted with a matrix, the summation always applies on the first index of the matrix if it appears after, or to the second index if it appears first. In other words, the product $\boldsymbol{A}\boldsymbol{B}$ between two matrices can be written $\boldsymbol{A} \underset{1}{\bullet} \boldsymbol{B}$ as well, without further index specification.

For instance, the products $\mathcal{X}_{i,j,k} = \sum_q \mathcal{T}_{q,j,k} M_{i,q}$, $\mathcal{Y}_{i,j,k} = \sum_q \mathcal{T}_{i,q,k} M_{j,q}$ and $\mathcal{Z}_{i,j,k} = \sum_q \mathcal{T}_{i,j,q} M_{k,q}$ can be written in compact form as $\mathcal{X} = \mathcal{T} \underset{1}{\bullet} \boldsymbol{M}$, $\mathcal{Y} = \mathcal{T} \underset{2}{\bullet} \boldsymbol{M}$ and $\mathcal{Z} = \mathcal{T} \underset{3}{\bullet} \boldsymbol{M}$, respectively. Note that we also have $\mathcal{X} = \boldsymbol{M}^{\mathrm{T}} \underset{1}{\bullet} \mathcal{T}$.

The Kronecker product between two matrices is denoted $\boldsymbol{A} \boxtimes \boldsymbol{B}$, and is defined by the block matrix:

$$\boldsymbol{A} \boxtimes \boldsymbol{B} = \begin{bmatrix} A_{1,1}\boldsymbol{B} & A_{1,2}\boldsymbol{B} & \dots & A_{1,J}\boldsymbol{B} \\ A_{2,1}\boldsymbol{B} & A_{2,2}\boldsymbol{B} & \dots & A_{1,J}\boldsymbol{B} \\ \vdots & \vdots & \vdots & \vdots \\ A_{I,1}\boldsymbol{B} & A_{I,2}\boldsymbol{B} & \dots & A_{I,J}\boldsymbol{B} \end{bmatrix},$$

if the matrix $\boldsymbol{A}$ is of size $I \times J$. Once the bases of linear spaces are well defined, the tensor product between operators can be represented by Kronecker products [9]. The Khatri–Rao (column-wise Kronecker) product between two matrices with the same number J of columns is written as:

$$\boldsymbol{A} \odot \boldsymbol{B} = [\boldsymbol{a}_1 \boxtimes \boldsymbol{b}_1, \dots \boldsymbol{a}_2 \boxtimes \boldsymbol{b}_2, \dots \boldsymbol{a}_J \boxtimes \boldsymbol{b}_J],$$

if $\boldsymbol{a}_j$ and $\boldsymbol{b}_j$ represent the columns of $\boldsymbol{A}$ and $\boldsymbol{B}$, respectively.

Analogously to the Frobenius norm of a matrix, the *Frobenius norm* of a tensor $\mathcal{T}$ is defined as the square root of the sum of all the squares of its elements, i.e.:

$$\| \mathcal{T} \|_{\mathrm{F}} = \sqrt{\sum_i \sum_j \sum_k \mathcal{T}^2_{i,j,k}}.$$

Contraction between two tensors

The product $\overset{p}{\underset{q}{\bullet}}$ between two tensors $\mathcal{A}$ and $\mathcal{B}$ of size $I_1 \times \dots \times I_Q$ and $J_1 \times \dots \times J_P$, respectively, where $I_q = J_p$, is a tensor of order $(Q + P - 2)$ denoted by:

$$[\mathcal{A} \overset{p}{\underset{q}{\bullet}} \mathcal{B}]_{i_1,\dots,i_{q-1},i_{q+1},\dots,i_Q,j_1,\dots,j_{p-1},j_{p+1},\dots,j_P} = \sum_{k=1}^{I_q} [\mathcal{A}]_{i_1,\dots,i_{q-1},k,i_{q+1},\dots,i_Q} [\mathcal{B}]_{j_1,\dots,j_{p-1},k,j_{p+1},\dots,j_P}.$$

Tensor reshaping

Tensor reshaping transforms a Q-order tensor $\mathcal{T}$ of dimensions $I_1 \times \dots \times I_Q$ into a matrix $\boldsymbol{T}_{(q)}$ having the product of the first q dimensions of $\mathcal{T}$, say $I_1 \cdots I_q$, as the number of rows, and the product of the remaining dimensions, $I_{q+1} \cdots I_Q$, as the number of columns. In MATLAB, this reshaping can be obtained using the native `reshape` function, such that

$$\boldsymbol{T}_{(q)} = \mathrm{reshape}\left(\mathcal{T}, \prod_{s=1}^{q} I_s, \prod_{s=q+1}^{Q} I_s\right). \tag{1}$$

For example, let $\mathcal{T}$ be a fourth-order tensor of dimensions $2 \times 2 \times 2 \times 2$, defined by:

$$\mathcal{T}(:,:,1,1) = \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix}, \mathcal{T}(:,:,2,1) = \begin{bmatrix} 5 & 7 \\ 6 & 8 \end{bmatrix},$$
$$\mathcal{T}(:,:,1,2) = \begin{bmatrix} 9 & 11 \\ 10 & 12 \end{bmatrix}, \mathcal{T}(:,:,2,2) = \begin{bmatrix} 13 & 15 \\ 14 & 16 \end{bmatrix}.$$

Matrix $\boldsymbol{T}_{(2)}$ of dimensions 4×4 is then given by:

$$\boldsymbol{T}_{(2)} = \mathrm{reshape}(\mathcal{T}, 4, 4) = \begin{bmatrix} 1 & 5 & 9 & 13 \\ 2 & 6 & 10 & 14 \\ 3 & 7 & 11 & 15 \\ 4 & 8 & 12 & 16 \end{bmatrix}.$$

The so-called *flattening* or *unfolding* operation is a special case of the reshaping operation [9]. The most used flattenings are those keeping one of the original dimensions, say N_q, as the number of rows. Hence, the so-called qth matrix unfolding of $\mathcal{T}$ is of dimension $I_q \times I_1 \dots I_{q-1} I_{q+1} \dots I_Q$. Compared to the flattening operation, the `reshape` function can be considered as a more general flattening operation, in the sense that

$$\mathrm{unfold}_q \mathcal{T} = \mathrm{reshape}\left(\mathcal{T}', I_q, \frac{\prod_{s=1}^{Q} I_s}{I_q}\right), \tag{2}$$

with

$$\mathcal{T}' = \mathrm{permute}(\mathcal{T}, [q, 1, 2, \dots, q-1, q+1, q+2, \dots, Q]),$$

where `permute` is a native MATLAB function that rearranges the dimensions of the Q-order tensor $\mathcal{T}$.

3 Tensor decompositions and algorithms

We introduce hereafter the tensor decompositions (Section 3.1) used in the applications presented in Sections 4, 5 and 6, as well as some basic algorithms to compute these decompositions (Section 3.2).

The Tucker decomposition along with two of its variants (high-order SVD and partial Tucker) are introduced in Section 3.1.1. Tucker decomposition is a generic tensor tool that decomposes a tensor into a set of non-structured factor matrices and a core tensor. A simple algorithm for computing the high-order SVD (Tucker decomposition with semi-unitary factor matrices) for third-order tensors is given in Section 3.2.2. The canonical polyadic decomposition (CPD) (Section 3.1.2) – probably the most widely used tensor decomposition – can be seen as a Tucker decomposition with a diagonal core. Its existence and uniqueness issues are discussed in Section 3.1.3. The pseudo-code for the popular alternating least squares (ALSs) algorithm for estimating the CPD is given in Section 3.2.1. Section 3.1.4 introduces the tensor-train decomposition (TTD), a tool designed to efficiently handle high-order tensors (order higher than 3), by splitting them into an interconnected set ('train') of lower-order tensors. A particular class of TTD, the Structured Tensor-Train (STT) models, is presented in Section 3.1.6. The tensor-train SVD (TT-SVD) algorithm for computing TTD is illustrated in Section 3.2.3. A link between the TTD and the CPD of high-order tensors (TT-CPD) is developed in Section 3.2.4; the JIRAFE (Joint dImensionality Reduction And Factors rEtrieval) method for estimating the parameters of the TT-CPD model is also detailed. Two additional tools used in MIMO application (Section 6), the least squares Kronecker (Section 3.2.5) and least squares Khatri–Rao (Section 3.2.6) factorisations, conclude this section.

3.1 Tensor decompositions

Any matrix can always be diagonalised by the congruent transformation. In addition, the two linear transforms involved can be imposed to be unitary: this is the singular value decomposition (SVD). When we move from matrices to tensors, it is generally impossible to use unitary transforms and obtain a diagonal tensor. Consequently, depending on which property is desired, we end up with two different decompositions. This is what is explained in the next two subsections, where we limit ourselves to third-order tensors to ease the presentation.

3.1.1 Tucker decomposition, HOSVD, and multilinear rank: Given a tensor $\mathcal{T}$ of dimensions $I \times J \times K$, it is always possible to find three matrices $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$ of dimensions $I \times R_1$, $J \times R_2$ and $K \times R_3$, respectively, such that

$$\mathcal{T}_{i,j,k} = \sum_{m=1}^{R_1} \sum_{n=1}^{R_2} \sum_{p=1}^{R_3} A_{i,m} B_{j,n} C_{k,p}\, \mathcal{G}_{m,n,p}, \tag{3}$$

where R_1, R_2, R_3 are minimal and $R_1 \leq I$, $R_2 \leq J$, $R_3 \leq K$. Tensor $\mathcal{G}$, often called *core tensor*, is of dimensions $R_1 \times R_2 \times R_3$. The triplet of minimal values of $\{R_1, R_2, R_3\}$ forms the *multilinear rank* of $\mathcal{T}$. This is referred to as the *Tucker decomposition* of $\mathcal{T}$, and can be written in a more compact way as:

$$\mathcal{T} = [[\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}; \mathcal{G}]]. \tag{4}$$

Each R_ℓ is uniquely defined, and corresponds to the rank of the linear operator associated with the ℓth matrix unfolding of $\mathcal{T}$, unfold$_\ell\mathcal{T}$. The point is that the Tucker decomposition is not unique, for it is defined up to three invertible matrices $\{\boldsymbol{M}_1, \boldsymbol{M}_2, \boldsymbol{M}_3\}$, because:

$$[[\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}; \mathcal{G}]] = \left[\left[\boldsymbol{A}\boldsymbol{M}_1^{-1}, \boldsymbol{B}\boldsymbol{M}_2^{-1}, \boldsymbol{C}\boldsymbol{M}_3^{-1}; \mathcal{G}'\right]\right]$$

if $\mathcal{G}' = [[\boldsymbol{M}_1, \boldsymbol{M}_2, \boldsymbol{M}_3; \mathcal{G}]]$. Yet, as in the SVD of matrices, it is possible to impose that the core tensor is obtained via semi-unitary transforms $\{\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W}\}$ as:

$$\mathcal{T} = [[\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W}; \mathcal{G}]], \tag{5}$$

where $\boldsymbol{U}^{\mathrm{H}}\boldsymbol{U} = \boldsymbol{I}_{R_1}$, $\boldsymbol{V}^{\mathrm{H}}\boldsymbol{V} = \boldsymbol{I}_{R_2}$ and $\boldsymbol{W}^{\mathrm{H}}\boldsymbol{W} = \boldsymbol{I}_{R_3}$. Equation (5) defines the high-order SVD (HOSVD), sometimes referred to as *multilinear SVD* of tensor $\mathcal{T}$ [10].

Partial Tucker decomposition: We present next a variant of the Tucker decomposition for an Q-order tensor $\mathcal{X} \in \mathbb{C}^{I_1 \times \cdots \times I_Q}$, with factor matrices $\boldsymbol{A}^{(q)} \in \mathbb{C}^{I_q \times R_q}$ whose $Q - Q_1$ last ones are equal to identity matrices $\boldsymbol{I}_{I_q}$ of order I_q, for $q = Q_1 + 1, \cdots, Q$. This so-called Tucker-(Q_1, Q) decomposition is compactly written as $\left[\left[\boldsymbol{A}^{(1)}, \ldots, \boldsymbol{A}^{(Q_1)}, \boldsymbol{I}_{I_{Q_1+1}}, \ldots, \boldsymbol{I}_{I_Q}; \mathcal{G}\right]\right]$, where the core tensor $\mathcal{G}$ is of dimensions $R_1 \times \cdots \times R_{Q_1} \times I_{Q_1+1} \times \cdots \times I_Q$, which induces $R_q = I_q$ for $q = Q_1 + 1, \ldots, Q$. See [11] for more details. For instance, the standard Tucker2 and Tucker1 decompositions correspond to $(Q_1, Q) = (1, 3)$ and $(Q_1, Q) = (2, 3)$, respectively.

3.1.2 Exact CP decomposition, rank, uniqueness: Of course, there is no reason that tensor $\mathcal{G}$ in (5) be diagonal; this is clear by just looking at the number of degrees of freedom of both sides. On the other hand, if the unitary constraint of matrices $\{\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{W}\}$ is relaxed, it is possible to decompose any tensor $\mathcal{T}$ as:

$$\mathcal{T} = [[\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}; \mathcal{S}]], \tag{6}$$

where tensor $\mathcal{S}$ is diagonal. This is known as the *Canonical Polyadic* (CP [The acronym CP also stands for Candecomp/Parafac in the literature, due to two contributions [1, 2] in which this tensor decomposition has been independently rediscovered, years after the original publication [12].]) decomposition (CPD). Note that now, not only matrices $\{\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}\}$ are not semi-unitary, but their number of columns, R, may exceed the number of their rows. The minimal value of R such that (6) holds *exactly* is called the *rank* of the tensor $\mathcal{T}$. The explicit writing of (6) in terms of array entries is:

$$\mathcal{T}_{i,j,k} = \sum_{r=1}^{R} A_{i,r} B_{j,r} C_{k,r}\, \sigma(r), \tag{7}$$

if $\sigma(r)$'s denote the diagonal entries of $\mathcal{S}$. But clearer writing of this CPD is provided in terms of *decomposable tensors* [9] as follows:

$$\mathcal{T} = \sum_{r=1}^{R} \sigma_r \mathcal{D}(r), \tag{8}$$

where $\mathcal{D}(r) = \boldsymbol{a}(r) \otimes \boldsymbol{b}(r) \otimes \boldsymbol{c}(r)$, that is, $\mathcal{D}(r)_{i,j,k} = A_{i,r} B_{j,r} C_{k,r}$, $\| \mathcal{D}(r) \| = 1$, and $\sigma_r > 0$. In fact, decomposable tensors may be seen as rank-1 tensors. This writing is clearer because it does not depend on the way tensors $\mathcal{D}(r)$ are represented – and their representation is precisely rarely unique, which is *not related* to the uniqueness of (8). Note that the definition of tensor rank is pretty much the same as matrix rank. Hence, this CPD may be seen as another natural generalisation of the matrix SVD to tensors, the former being HOSVD seen in the previous section. Because the acronym CP is sometimes used to mean 'Completely Positive', it may sometimes be more suitable to call it *rank decomposition*, when the number of terms R is indeed minimal.

The uniqueness of the CPD (8) should not be confused with its array representation (7). In fact, the latter is never unique, since expressing a rank-1 tensor as the tensor product of N vectors is subject to $N - 1$ scaling indeterminacies. For instance, $\mathcal{D} = \boldsymbol{a} \otimes \boldsymbol{b} \otimes \boldsymbol{c}$ can also be written as $\alpha\boldsymbol{a} \otimes \beta\boldsymbol{b} \otimes \gamma\boldsymbol{c}$ provided $\alpha\beta\gamma = 1$. This is precisely the difference between a tensor space and a product of linear spaces [9, 13, 14]. This is the reason why the wording of 'essential uniqueness' is sometimes found in the literature; it expresses the fact that there are $N - 1$ scaling indeterminacies, and that the order of summation is subject to permutation because the addition is commutative.

With Definition (2), the CPD can be rewritten in matrix form. For a third-order rank-R tensor $\mathcal{T} = [[\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}; \mathcal{S}]]$ for example, the three flattened representations of the CPD are [15]:

$$\text{unfold}_1\mathcal{T} = \boldsymbol{A}\boldsymbol{S}(\boldsymbol{C} \odot \boldsymbol{B})^{\mathrm{T}}, \tag{9}$$

$$\text{unfold}_2\mathcal{T} = \boldsymbol{B}\boldsymbol{S}(\boldsymbol{C} \odot \boldsymbol{A})^{\mathrm{T}}, \tag{10}$$

$$\text{unfold}_3\mathcal{T} = \boldsymbol{C}\boldsymbol{S}(\boldsymbol{B} \odot \boldsymbol{A})^{\mathrm{T}}, \tag{11}$$

where $\boldsymbol{S}$ denotes the diagonal matrix with entries $\sigma(r) = \mathcal{S}_{r,r,r}$, $\forall 1 \leq r \leq R$.

One of the major interests in the CPD (8) lies in its uniqueness. In particular, it allows to relax the orthogonality constraint (necessary in the case of matrices to achieve some form of uniqueness), which is often incompatible with physical reality. More precisely, several uniqueness conditions have been derived in the literature. We shall quote two of them. A sufficient condition is that [16–18]:

$$2R \leq \text{krank}\{\boldsymbol{A}\} + \text{krank}\{\boldsymbol{B}\} + \text{krank}\{\boldsymbol{C}\} - 2, \tag{12}$$

where krank$\{\boldsymbol{A}\}$ denotes the so-called *Kruskal rank* of $\boldsymbol{A}$ [9, 16]; by definition [A matrix $\boldsymbol{M}$ has Kruskal rank R if *any* subset of R columns forms a full rank matrix. Remember that $\boldsymbol{M}$ has rank R if there exists *at least one* subset of R columns forming a full rank matrix.], the *krank* is always smaller than or equal to the usual matrix rank. If entries of $\boldsymbol{A}$ are seen as randomly drawn according to an absolutely continuous distribution, krank$\{\boldsymbol{A}\}$ can be replaced by the smallest of its two dimensions. In the latter case, the condition is referred to as *generic* [19, 20].

A second generic condition ensuring uniqueness of the CPD is given by the bound of the so-called *expected rank* [21–23]:

$$R \leq \frac{R_1 R_2 R_3}{R_1 + R_2 + R_3 - 2} - 1, \tag{13}$$

where R_ℓ are the multilinear ranks defined in Section 3.1.1. To sum up, the exact CP decomposition (8) is almost surely unique if the tensor rank is not too large.

3.1.3 Approximate CPD, existence, uniqueness: Things become more complicated when it is thought to *approximate* a tensor by another one of lower rank. To fix the ideas, suppose that our objective is to minimise:

$$\Upsilon = \| \mathcal{T} - \sum_{r=1}^{R_o} \sigma(r)\, \boldsymbol{a}(r) \otimes \boldsymbol{b}(r) \otimes \boldsymbol{c}(r) \|_F^2,$$

where $\sigma(r) > 0$ and $||\boldsymbol{a}(r)|| = ||\boldsymbol{b}(r)|| = ||\boldsymbol{c}(r)|| = 1$, $\forall r, 1 \leq r \leq R_o$. Unfortunately, it has been pointed out in e.g. [24] that this objective function may not always have a minimum (but only an infimum), which means that this low-rank approximation problem is ill-posed; there are ways to circumvent the difficulty, for instance by imposing extraneous angular constraints [25], which is meaningful in the context of DOA estimation for instance.

On the other hand, if all objects are real non-negative, then the problem becomes well-posed [26]. In the latter case, it is important to recall that imposing a non-negative constraint generally

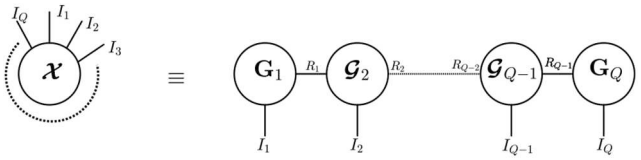

Fig. 1 *TT decomposition of a Q-order tensor*

increases the rank [9], even for matrices. This being said, it has been proved in [22] that the best non-negative low-rank tensor approximation of a non-negative tensor is almost surely unique. One can even look at the conditions under which the best low-rank approximation admits a unique CPD [27], but this is more involved.

To conclude, the *best low-rank approximate* of a tensor does not always exist, and this is still often ignored in the literature.

3.1.4 Tensor-train decomposition and TT-ranks: A Q-order tensor of size $I_1 \times \cdots \times I_Q$ admits a decomposition into a train of tensors [28] if

$$\mathcal{X} = \boldsymbol{G}^{(1)} \underset{2}{\overset{1}{\bullet}} \mathcal{G}^{(2)} \underset{3}{\overset{1}{\bullet}} \mathcal{G}^{(3)} \underset{4}{\overset{1}{\bullet}} \cdots \underset{Q-1}{\overset{1}{\bullet}} \mathcal{G}^{(Q-1)} \underset{Q}{\overset{1}{\bullet}} \boldsymbol{G}^{(Q)}, \tag{14}$$

where the TT-cores $\boldsymbol{G}^{(1)}$, $\mathcal{G}^{(q)} (2 \leq q \leq Q-1)$, and $\boldsymbol{G}^{(Q)}$ are, respectively, of size $I_1 \times R_1$, $R_{q-1} \times I_q \times R_q$ and $R_{Q-1} \times I_Q$. The $Q-1$ dimensions $\{R_1, , R_{Q-1}\}$ are referred to as the *TT-ranks* with boundary conditions $R_0 = R_Q = 1$.

The idea behind the TTD is to transform a high Q-order tensor into a set of much lower third-order tensors, which allows to break the 'curse of dimensionality' [29]. Indeed, just like the CPD, the storage cost of the TTD scales linearly with the order Q. A graph-based representation of the TTD of a Q-order tensor $\mathcal{X}$ is given in Fig. 1. It is straightforward to see that the TTD is not unique since [30]

$$[\mathcal{X}]_{i_1, \cdots, i_Q} = \boldsymbol{a}_1(i_1)^{\mathrm{T}} \boldsymbol{A}_2(i_2) \cdots \boldsymbol{A}_{Q-1}(i_{Q-1}) \boldsymbol{a}_Q(i_Q) \tag{15}$$

where

$$\boldsymbol{a}_1(i_1) = \boldsymbol{M}_1 [\boldsymbol{G}^{(1)}]^{\mathrm{T}}_{i_1, :} \ : \ R_1 \times 1, \tag{16}$$

$$\boldsymbol{a}_Q(i_Q) = \boldsymbol{M}_{Q-1}^{-1} [\boldsymbol{G}^{(Q)}]_{:, i_Q} \ : \ R_{Q-1} \times 1, \tag{17}$$

$$\boldsymbol{A}_q(i_q) = \boldsymbol{M}_{q-1}^{-T} [\mathcal{G}^{(q)}]_{:, i_q, :} \boldsymbol{M}_q \ : \ R_{q-1} \times R_q, \tag{18}$$

with $\boldsymbol{M}_q$, an invertible matrix of size $R_q \times R_q$. In [30], it is shown that, when we apply decomposition algorithms such as TT-SVD or TT-HSVD, the matrices $\boldsymbol{M}_q$ are *change-of-basis* matrices, that are mainly linked to the estimation of dominant subspaces by the SVD.

3.1.5 Model equivalence between TTD and CPD: An interesting model equivalence property between a Q-order CPD of rank-R and a train of $(Q-2)$ third-order CPD(s) of rank R and two matrices is briefly discussed in [28]. We can summarise this fundamental algebraic relation according to the following result.

Theorem 1: If the tensor $\mathcal{X}$ follows a Q-order CPD of rank-R with factors $\boldsymbol{P}_q$, then a TTD can be given by:

$$\boldsymbol{G}_1 = \boldsymbol{P}_1, \tag{19}$$

(see (20))

$$\boldsymbol{G}_Q = \boldsymbol{P}_Q^{\mathrm{T}}, \tag{21}$$

and the TT-ranks are all equal to the canonical rank R.

Proof: The TTD of the Q-order identity tensor $\mathcal{I}_{Q,R}$ is given by

$$\mathcal{I}_{Q,R} = \boldsymbol{I}_R \underset{2}{\overset{1}{\bullet}} \mathcal{I}_{3,R} \underset{3}{\overset{1}{\bullet}} \cdots \underset{Q-1}{\overset{1}{\bullet}} \mathcal{I}_{3,R} \underset{Q}{\overset{1}{\bullet}} \boldsymbol{I}_R. \tag{22}$$

Substituting the above TTD into the CPD, we get

$$\mathcal{X} = (\boldsymbol{I}_R \underset{2}{\overset{1}{\bullet}} \mathcal{I}_{3,R} \underset{3}{\overset{1}{\bullet}} \cdots \underset{Q}{\overset{1}{\bullet}} \boldsymbol{I}_R) \underset{1}{\bullet} \boldsymbol{P}_1 \ldots \underset{Q}{\bullet} \boldsymbol{P}_Q \tag{23}$$

by expressing the entries of $\mathcal{X}$ and reorganising them, we can equivalently write

$$\mathcal{X} = (\boldsymbol{P}_1 \boldsymbol{I}_R) \underset{2}{\overset{1}{\bullet}} (\mathcal{I}_{3,R} \underset{2}{\bullet} \boldsymbol{P}_2) \underset{3}{\overset{1}{\bullet}} \cdots \underset{Q}{\overset{1}{\bullet}} (\boldsymbol{I}_R \boldsymbol{P}_Q^{\mathrm{T}}). \tag{24}$$

By identifying the TT-cores in (24) with those in (14), we can deduce the relations (19), (20) and (21). Then, it is straightforward to conclude that the TT-ranks are all identical and equal to the canonical rank R. □

Thus, conditionally to the knowledge of the TT-cores, it is theoretically possible to recover the factors of the CPD by a one-to-one methodology. In Section 3.2.4, we present the so-called JIRAFE framework, used for that aim.

3.1.6 Structured TT models: In this subsection, we present a particular class of TT models, the so-called STT models. These models are composed of a train of third-order tensors, each tensor being represented by means of a tensor decomposition like CP, Tucker or generalised Tucker.

For a Q-order tensor $\mathcal{X} \in \mathbb{C}^{I_1 \times I_2 \cdots \times I_Q}$, an STT model can be written as:

$$\mathcal{X} = \mathcal{T}^{(1)} \underset{3}{\overset{1}{\bullet}} \mathcal{T}^{(2)} \underset{4}{\overset{1}{\bullet}} \cdots \underset{Q-1}{\overset{1}{\bullet}} \mathcal{T}^{(Q-2)}, \tag{25}$$

with $\mathcal{T}^{(1)} \in \mathbb{C}^{I_1 \times I_2 \times R_2}$, $\mathcal{T}^{(Q-2)} \in \mathbb{C}^{R_{2Q-6} \times I_{Q-1} \times I_Q}$, and $\mathcal{T}^{(q)} \in \mathbb{C}^{R_{2q-2} \times I_{q+1} \times R_{2q}}$ for $2 \leq q \leq Q-3$.

For instance, in the case of a fifth-order tensor $\mathcal{X} \in \mathbb{C}^{I_1 \times I_2 \cdots \times I_5}$, let us assume that $\mathcal{T}^{(q)}$ satisfies a Tucker-(1,3) decomposition for $q \in \{1, 2\}$ while $\mathcal{T}^{(3)}$ satisfies a Tucker-(2,3) decomposition such that [11]:

$$\mathcal{T}^{(1)} = \left[\left[\boldsymbol{A}^{(1)}, \boldsymbol{I}_{I_2}, \boldsymbol{I}_{R_2}; \mathcal{G}^{(1)}\right]\right] \in \mathbb{C}^{I_1 \times I_2 \times R_2} \tag{26}$$

$$\mathcal{T}^{(2)} = \left[\left[\boldsymbol{A}^{(2)}, \boldsymbol{I}_{I_3}, \boldsymbol{I}_{R_4}; \mathcal{G}^{(2)}\right]\right] \in \mathbb{C}^{R_2 \times I_3 \times R_4} \tag{27}$$

$$\mathcal{T}^{(3)} = \left[\left[\boldsymbol{A}^{(3)}, \boldsymbol{I}_{I_4}, \boldsymbol{A}^{(4)}; \mathcal{G}^{(3)}\right]\right] \in \mathbb{C}^{R_4 \times I_4 \times I_5} \tag{28}$$

with the core tensors $\mathcal{G}^{(1)} \in \mathbb{C}^{R_1 \times I_2 \times R_2}$, $\mathcal{G}^{(2)} \in \mathbb{C}^{R_3 \times I_3 \times R_4}$, $\mathcal{G}^{(3)} \in \mathbb{C}^{R_5 \times I_4 \times R_6}$, and the matrix factors $\boldsymbol{A}^{(1)} \in \mathbb{C}^{I_1 \times R_1}$, $\boldsymbol{A}^{(2)} \in \mathbb{C}^{R_2 \times R_3}$, $\boldsymbol{A}^{(3)} \in \mathbb{C}^{R_4 \times R_5}$, $\boldsymbol{A}^{(4)} \in \mathbb{C}^{I_5 \times R_6}$. We then obtain a structured Tucker train of order five, abbreviated as STuT(5), which can be written by means of the following scalar equation:

$$\begin{aligned} x_{i_1, i_2, i_3, i_4, i_5} = & \sum_{r_1=1}^{R_1} \sum_{r_2=1}^{R_2} \cdots \sum_{r_6=1}^{R_6} a^{(1)}_{i_1, r_1} g^{(1)}_{r_1, i_2, r_2} a^{(2)}_{r_2, r_3} g^{(2)}_{r_3, i_3, r_4} \\ & \times a^{(3)}_{r_4, r_5} g^{(3)}_{r_5, i_4, r_6} a^{(4)}_{i_5, r_6}, \end{aligned} \tag{29}$$

or compactly as:

$$\mathcal{G}_q = \mathcal{I}_{3,R} \underset{2}{\bullet} \boldsymbol{P}_q \quad (\text{third} - \text{order CPD}), \ \text{where } 2 \leq q \leq Q-1, \tag{20}$$

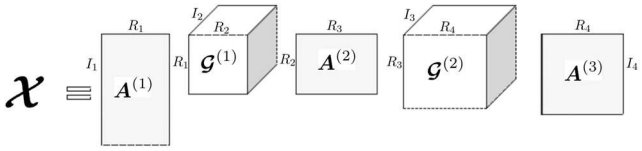

Fig. 2 *NTT(4) model for a fourth-order tensor*

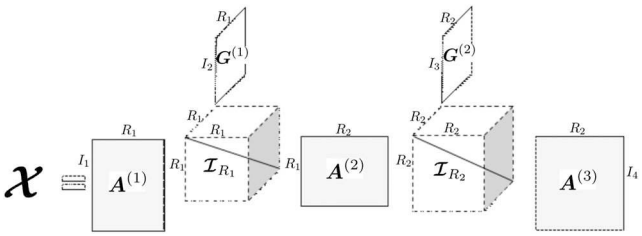

Fig. 3 *Nested CP model for a fourth-order tensor*

$$\mathcal{X} = \mathcal{T}^{(1)} \underset{3}{\bullet}{}^{1} \mathcal{T}^{(2)} \underset{4}{\bullet}{}^{1} \mathcal{T}^{(3)}. \tag{30}$$

Comparing (29) and (30) of the STuT(5) model with (14) of the TT model with $Q = 5$, we can conclude that the STuT model corresponds to a TT model for which each tensor of the train satisfies a Tucker-(1,3) or Tucker-(2,3) decomposition.

Such an STuT model was derived for the fourth-order tensor of received signals in a cooperative wireless MIMO communication system [31], and then generalised to a Q-order tensor in the case of multi-hop MIMO relay systems [32]. This model can also be interpreted as a nested Tucker train (NTT) model, with two successive tensors sharing a common factor.

Thus, in the case of a fourth-order tensor $\mathcal{X} \in \mathbb{C}^{I_1 \times I_2 \times I_3 \times I_4}$, as illustrated by means of Fig. 2, with $\boldsymbol{A}^{(1)} \in \mathbb{C}^{I_1 \times R_1}, \mathcal{G}^{(1)} \in \mathbb{C}^{R_1 \times I_2 \times R_2}, \boldsymbol{A}^{(2)} \in \mathbb{C}^{R_2 \times R_3}, \mathcal{G}^{(2)} \in \mathbb{C}^{R_3 \times I_3 \times R_4}, \boldsymbol{A}^{(3)}, \in \mathbb{C}^{I_4 \times R_4}$
the NTT(4) model can be described by means of the following scalar equation:

$$x_{i_1,i_2,i_3,i_4} = \sum_{r_1=1}^{R_1} \sum_{r_2=1}^{R_2} \sum_{r_3=1}^{R_3} \sum_{r_4=1}^{R_4} a^{(1)}_{i_1,r_1} g^{(1)}_{r_1,i_2,r_2} a^{(2)}_{r_2,r_3} g^{(2)}_{r_3,i_3,r_4} a^{(3)}_{i_4,r_4}. \tag{31}$$

Let us define the following third-order Tucker models:

$$\mathcal{T}^{(1)} = \left[\left[\boldsymbol{A}^{(1)}, \boldsymbol{I}_{I_2}, \boldsymbol{I}_{R_2}; \mathcal{G}^{(1)}\right]\right] \tag{32}$$

$$= \mathcal{G}^{(1)} \underset{1}{\bullet} \boldsymbol{A}^{(1)} \in \mathbb{C}^{I_1 \times I_2 \times R_2} \tag{33}$$

$$\mathcal{T}^{(2)} = \left[\left[\boldsymbol{A}^{(2)}, \boldsymbol{I}_{I_3}, \boldsymbol{A}^{(3)}; \mathcal{G}^{(2)}\right]\right] \tag{34}$$

$$= \mathcal{G}^{(2)} \underset{1}{\bullet} \boldsymbol{A}^{(2)} \underset{3}{\bullet} \boldsymbol{A}^{(3)} \in \mathbb{C}^{R_2 \times I_3 \times I_4} \tag{35}$$

$$\mathcal{T}^{(3)} = \left[\left[\boldsymbol{A}^{(1)}, \boldsymbol{I}_{I_2}, \boldsymbol{A}^{(2)}; \mathcal{G}^{(1)}\right]\right] \tag{36}$$

$$= \mathcal{G}^{(1)} \underset{1}{\bullet} \boldsymbol{A}^{(1)} \underset{3}{\bullet} \boldsymbol{A}^{(2)^{\mathrm{T}}} \in \mathbb{C}^{I_1 \times I_2 \times R_3} \tag{37}$$

$$\mathcal{T}^{(4)} = \left[\left[\boldsymbol{I}_{R_3}, \boldsymbol{I}_{I_3}, \boldsymbol{A}^{(3)}; \mathcal{G}^{(2)}\right]\right] \tag{38}$$

$$= \mathcal{G}^{(2)} \underset{3}{\bullet} \boldsymbol{A}^{(3)} \in \mathbb{C}^{R_3 \times I_3 \times I_4}. \tag{39}$$

These Tucker models can be written in an element-wise form as:

$$t^{(1)}_{i_1,i_2,r_2} = \sum_{r_1=1}^{R_1} g^{(1)}_{r_1,i_2,r_2} a^{(1)}_{i_1,r_1} \tag{40}$$

$$t^{(2)}_{r_2,i_3,i_4} = \sum_{r_3=1}^{R_3} \sum_{r_4=1}^{R_4} g^{(2)}_{r_3,i_3,r_4} a^{(2)}_{r_2,r_3} a^{(3)}_{i_4,r_4} \tag{41}$$

$$t^{(3)}_{i_1,i_2,r_3} = \sum_{r_1=1}^{R_1} \sum_{r_2=1}^{R_2} g^{(1)}_{r_1,i_2,r_2} a^{(1)}_{i_1,r_1} a^{(2)}_{r_2,r_3} \tag{42}$$

$$t^{(4)}_{r_3,i_3,i_4} = \sum_{r_4=1}^{R_4} g^{(2)}_{r_3,i_3,r_4} a^{(3)}_{i_4,r_4}. \tag{43}$$

The NTT(4) model (31) can be viewed as the nesting of the third-order Tucker models $\mathcal{T}^{(3)}$ and $\mathcal{T}^{(2)}$ which share the matrix factor $\boldsymbol{A}^{(2)}$. It can also be interpreted as a contraction of a Tucker-(1,3) model with a Tucker-(2,3) model, along their common mode:

$$\mathcal{X} = \mathcal{T}^{(1)} \underset{3}{\bullet}{}^{1} \mathcal{T}^{(2)} = \mathcal{T}^{(3)} \underset{3}{\bullet}{}^{1} \mathcal{T}^{(4)}. \tag{44}$$

These contraction operations correspond to summing the entries of the third-order tensors $\mathcal{T}^{(1)}$ and $\mathcal{T}^{(2)}$, or $\mathcal{T}^{(3)}$ and $\mathcal{T}^{(4)}$, along their common mode r_2, or r_3, as follows:

$$x_{i_1,i_2,i_3,i_4} = \sum_{r_2=1}^{R_2} t^{(1)}_{i_1,i_2,r_2} t^{(2)}_{r_2,i_3,i_4} \tag{45}$$

$$= \sum_{r_3=1}^{R_3} t^{(3)}_{i_1,i_2,r_3} t^{(4)}_{r_3,i_3,i_4}. \tag{46}$$

Defining the fourth-order tensor $\mathcal{C} \in \mathbb{C}^{R_1 \times I_2 \times I_3 \times R_4}$ such that:

$$c_{r_1,i_2,i_3,r_4} = \sum_{r_2=1}^{R_2} \sum_{r_3=1}^{R_3} g^{(1)}_{r_1,i_2,r_2} a^{(2)}_{r_2,r_3} g^{(2)}_{r_3,i_3,r_4}, \tag{47}$$

the NTT(4) model (31) can also be interpreted as the following Tucker-(2,4) model [33]:

$$\mathcal{X} = \mathcal{C} \underset{1}{\bullet} \boldsymbol{A}^{(1)} \underset{4}{\bullet} \boldsymbol{A}^{(3)}, \tag{48}$$

or in the scalar form:

$$x_{i_1,i_2,i_3,i_4} = \sum_{r_1=1}^{R_1} \sum_{r_4=1}^{R_4} c_{r_1,i_2,i_3,r_4} a^{(1)}_{i_1,r_1} a^{(3)}_{i_4,r_4}. \tag{49}$$

The nested CP model introduced in [34], and exploited in [35] in the context of MIMO relay systems, is a special case of the nested Tucker model (31), defined as follows:

$$x_{i_1,i_2,i_3,i_4} = \sum_{r_1=1}^{R_1} \sum_{r_2=1}^{R_2} a^{(1)}_{i_1,r_1} g^{(1)}_{i_2,r_1} a^{(2)}_{r_1,r_2} g^{(2)}_{i_3,r_2} a^{(3)}_{i_4,r_2}, \tag{50}$$

with $\boldsymbol{A}^{(1)} \in \mathbb{C}^{I_1 \times R_1}, \boldsymbol{G}^{(1)} \in \mathbb{C}^{I_2 \times R_1}, \boldsymbol{A}^{(2)} \in \mathbb{C}^{R_1 \times R_2}, \boldsymbol{G}^{(2)} \in \mathbb{C}^{I_3 \times R_2}, \boldsymbol{A}^{(3)}. \in \mathbb{C}^{I_4 \times R_2}$
The nested CP model is illustrated in Fig. 3 for a fourth-order tensor.

This nested CP model can be deduced from the nested Tucker model (31) using the following correspondences:

$$(r_1, r_2, r_3, r_4) \leftrightarrow (r_1, r_1, r_2, r_2) \tag{51}$$

$$(\boldsymbol{A}^{(1)}, \mathcal{G}^{(1)}, \boldsymbol{A}^{(2)}, \mathcal{G}^{(2)}, \boldsymbol{A}^{(3)}) \leftrightarrow (\boldsymbol{A}^{(1)}, \boldsymbol{G}^{(1)}, \boldsymbol{A}^{(2)}, \boldsymbol{G}^{(2)}, \boldsymbol{A}^{(3)}). \tag{52}$$

More generally, one can define a STT model (25) for which each elementary tensor of the train has a CP decomposition. Thus, for a Q-order tensor $\mathcal{X}$, (25) then becomes a structured CP train (SCPT) such that:

$$\mathcal{T}^{(1)} = \left[\left[\boldsymbol{A}^{(1)}, \boldsymbol{G}^{(1)}, \boldsymbol{A}^{(2)^{\mathrm{T}}}; \mathcal{J}_{R_1}\right]\right] \in \mathbb{C}^{I_1 \times I_2 \times R_2}$$

$$\mathcal{T}^{(q)} = \left[\left[\boldsymbol{A}^{(q)}, \boldsymbol{G}^{(q)}, \boldsymbol{A}^{(q+1)^{\mathrm{T}}}; \mathcal{J}_{R_q}\right]\right] \in \mathbb{C}^{R_{q-1} \times I_{q+1} \times R_{q+1}} \text{ for } 2 \le q \le Q-3$$

$$\mathcal{T}^{(Q-2)} = \left[\left[\boldsymbol{A}^{(Q-2)}, \boldsymbol{G}^{(Q-2)}, \boldsymbol{A}^{(Q-1)}; \mathcal{J}_{R_{Q-2}}\right]\right] \in \mathbb{C}^{R_{Q-2} \times I_{Q-1} \times I_Q},$$

with the matrix factors $\boldsymbol{A}^{(1)} \in \mathbb{C}^{I_1 \times R_1}, \boldsymbol{A}^{(q)} \in \mathbb{C}^{R_{q-1} \times R_q}$, for $2 \le q \le Q-1$, $\boldsymbol{A}^{(Q-1)} \in \mathbb{C}^{I_Q \times R_{Q-2}}$, and $\boldsymbol{G}^{(q)} \in \mathbb{C}^{I_{q+1} \times R_q}$, for $1 \le q \le Q-2$. The SCPT model of order Q can also be written in the following scalar form:

$$x_{i_1,\cdots,i_Q} = \sum_{r_1=1}^{R_1} \cdots \sum_{r_{Q-2}=1}^{R_{Q-2}} a^{(1)}_{i_1,r_1} g^{(1)}_{i_2,r_1} a^{(2)}_{r_1,r_2} g^{(2)}_{i_3,r_2} \times a^{(3)}_{r_2,r_3} g^{(3)}_{i_4,r_3} \cdots g^{(Q-2)}_{i_{Q-1},r_{Q-2}} a^{(Q-1)}_{i_Q,r_{Q-2}}.$$

3.2 Algorithms

3.2.1 ALSs algorithm: There are many algorithms to compute a CP decomposition. By far, the most used one is the ALSs. ALS was proposed in [1, 2], and is considered today as the 'workhorse' for CPD computation. It is a simple algorithm that fixes, iteratively, all but one factor, which is then updated by solving a linear least squares problem. In Algorithm 1 (Fig. 4), we present the ALS algorithm for a third-order tensor $\mathcal{T} \approx [\![\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}; \mathcal{J}_{3,R}]\!]$. The generalisation for a Q-order case is straightforward.

Generally `CritStop` is based on the evaluation of the fitting error $\| \mathcal{T} - [\![\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}; \mathcal{J}_{3,R}]\!] \|_{\mathrm{F}}$.

3.2.2 High-order SVD algorithm: Based on the definition of the HOSVD given in (5), an algorithm to compute the HOSVD is presented in Algorithm 2. This algorithm was proposed in [10], and it relies on the computation of the left singular vectors of the unfoldings $\mathrm{unfold}_q \mathcal{T}$. The truncated HOSVD has a moderate computing cost, and is not necessarily optimal, but it gives a good approximation, which can be used as initialisation for ALS-like algorithms. For ease of presentation, we derive the method only for a third-order tensor, but the generalisation to a Q-order tensor is straightforward.

Algorithm 2 HOSVD algorithm:

Input: Third-order tensor $\mathcal{T}$, multilinear rank $\{R_1, R_2, R_3\}$
Output: $\boldsymbol{U}$, $\boldsymbol{V}$, $\boldsymbol{W}$ and $\mathcal{G}$
(1) $\boldsymbol{U} \leftarrow R_1$ first left singular vectors of $\mathrm{unfold}_1 \mathcal{T}$
(2) $\boldsymbol{V} \leftarrow R_2$ first left singular vectors of $\mathrm{unfold}_2 \mathcal{T}$
(3) $\boldsymbol{W} \leftarrow R_3$ first left singular vectors of $\mathrm{unfold}_3 \mathcal{T}$
(4) $\mathcal{G} = \mathcal{T} \underset{1}{\bullet} \boldsymbol{U}^H \underset{2}{\bullet} \boldsymbol{V}^H \underset{3}{\bullet} \boldsymbol{W}^H$

3.2.3 TTD computation with the TT-SVD algorithm: The TT-SVD algorithm has been introduced in [28]. This algorithm minimises in a sequential way the following LS criterion:

$$\psi(\mathcal{X}) = \| \mathcal{X} - \boldsymbol{A}_1 \overset{1}{\underset{2}{\bullet}} \mathcal{A}_2 \overset{1}{\underset{3}{\bullet}} \cdots \overset{1}{\underset{Q-1}{\bullet}} \mathcal{A}_{Q-1} \overset{1}{\underset{Q}{\bullet}} \boldsymbol{A}_Q \|^2_{\mathrm{F}}. \tag{53}$$

In Fig. 5, we present in a schematic way the TT-SVD algorithm applied on a fourth-order tensor $\mathcal{X}$. As illustrated, the TT-SVD computes in a sequential way $Q-1$ SVDs on matrix reshapings of the Q-order original tensor. Note that at each step, we apply the truncated SVD on the reshaped matrices $\boldsymbol{V}^{(q)}_{(2)}$ of size $(R_q I_{q+1}) \times (I_{q+2} \cdots I_Q)$ to recover matrices $\boldsymbol{U}^{(q+1)}$ and $\boldsymbol{V}^{(q+1)}$, this latter containing the product of the pseudo-diagonal singular values matrix and the right singular vectors matrix.

Input: 3-order rank-R tensor $\mathcal{T}$, `CritStop`
Output: Estimated CPD factors: $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$

1: **Initialize** $\boldsymbol{B}$ and $\boldsymbol{C}$
2: **while** `CritStop` and maximum iterations are not reached **do**
3: $\quad \mathbf{A} = \mathrm{unfold}_1 \mathcal{T} \cdot (\boldsymbol{C} \odot \boldsymbol{B})^{\mathrm{T}\dagger}$
4: $\quad \mathbf{B} = \mathrm{unfold}_2 \mathcal{T} \cdot (\boldsymbol{C} \odot \boldsymbol{A})^{\mathrm{T}\dagger}$
5: $\quad \mathbf{C} = \mathrm{unfold}_3 \mathcal{T} \cdot (\boldsymbol{B} \odot \boldsymbol{A})^{\mathrm{T}\dagger}$
6: **end while**

Fig. 4 *Algorithm 1 ALS algorithm*

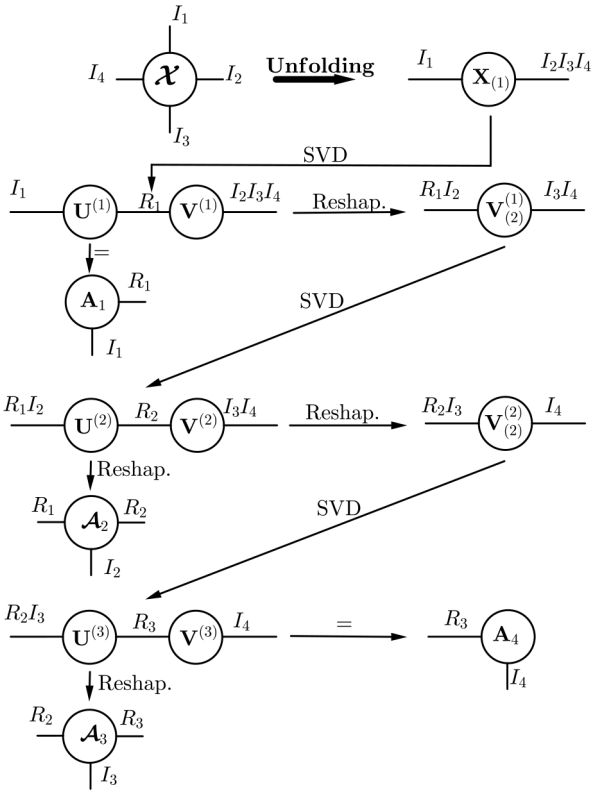

Fig. 5 *TT-SVD algorithm applied on a fourth-order tensor*

It is worth noting that the TT-SVD in its current state cannot be parallelised, which is a problem when we deal with very high order tensors. An alternative method to compute the TTD is to consider different reshapings than those considered in the TT-SVD. Indeed, the recently proposed TT-HSVD [30] algorithm, for tensor-train hierarchical SVD, is a hierarchical algorithm for TTD, that suggests to combine more than one mode in each dimension of the reshaped matrices, i.e. instead of considering the matrix $\boldsymbol{X}_{(1)}$ of size $I_1 \times (I_2 I_3 I_4)$, we can for example use as first unfolding the matrix $\boldsymbol{X}_{(2)}$ of size $(I_1 I_2) \times (I_3 I_4)$. Note that using this strategy allows to parallelise the decomposition, after each SVD, across several processors, which suits very well high order tensors. Moreover, the choice of the best reshaping strategy when the order is very high is discussed in [30] in terms of the algorithmic complexity. Indeed, [30] shows that reshapings with the 'most square' matrix (i.e. matrices with more balanced row and column dimensions), leads to the lower computational complexity.

Remark that from an estimation point of view, the TT-SVD and the TT-HSVD algorithms suffer from the lack of uniqueness of the TT-cores described in Section 3.1.4. Indeed, as the latent matrices $\{\boldsymbol{M}_1, \ldots, \boldsymbol{M}_{Q-1}\}$ are unknown, the true TT-cores remain unknown. In the next section, we propose a methodology to solve this problem in the important context where the observed tensor follows a Q-order CPD of rank R.

3.2.4 JIRAFE principle for CPD: In this section, we present a TT-based methodology [36], JIRAFE, for high-order CPD factors

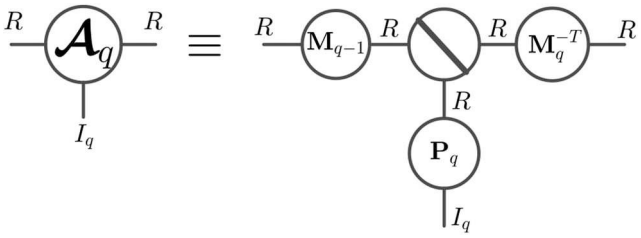

Fig. 6 *Third-order CPD of the qth TT-core*

Input: Q-order rank-R tensor $\boldsymbol{\mathcal{X}}$
Output: Estimated CPD factors: $\boldsymbol{P}_1, \cdots, \boldsymbol{P}_Q$.

1: Model reduction: Solve criterion (54) using TT-SVD or TT-HSVD, *i.e.*

$$\min_{\boldsymbol{A}_1, \boldsymbol{\mathcal{A}}_2, \ldots, \boldsymbol{\mathcal{A}}_{Q-1}, \boldsymbol{A}_Q} \psi(\boldsymbol{\mathcal{X}})$$

2: Factor retrieval:

$$\min_{\boldsymbol{M}_1, \boldsymbol{P}_2, \boldsymbol{M}_2} \phi(\boldsymbol{\mathcal{A}}_2).$$

3: **for** $k = 3 \cdots Q-1$ **do**
4: $\quad \min_{\boldsymbol{M}_q, \boldsymbol{P}_q} \phi(\boldsymbol{\mathcal{A}}_q | \boldsymbol{M}_{q-1})$
5: **end for**
6: $\boldsymbol{P}_1 = \boldsymbol{A}_1 \boldsymbol{M}_1$, and $\boldsymbol{P}_Q = \boldsymbol{A}_Q^{\mathsf{T}} \boldsymbol{M}_{Q-1}^{-\mathsf{T}}$

Fig. 7 *Algorithm 3 JIRAFE for CPD*

retrieval. The acronym JIRAFE stands for 'Joint dImensionality Reduction And Factors rEtrieval' and the method is described in [36, 37] by the following two-step procedure:

(i) Reduce the dimensionality of the original factor retrieval problem by breaking the difficult multidimensional optimisation problem into a collection of simpler optimisation problems on small-order tensors. This step is performed using the TT-SVD algorithm.
(ii) Design a factor retrieval strategy by exploiting (or not) the coupled existing structure between the first and third factors of two consecutive TT-cores. Here, the goal is to minimise a sum of coupled least-square (LS) criteria.

In [36], the structures of the TT-cores associated with the TT-SVD algorithm are described and a family of estimation algorithms is proposed. This methodology is based on the following result.

Theorem 2: If the data tensor follows a Q-order CPD of rank R parametrised by Q full column rank factors $\{\boldsymbol{P}_1, \ldots, \boldsymbol{P}_Q\}$. The TT-SVD algorithm recovers the TT-cores such that:

$$\boldsymbol{A}_1 = \boldsymbol{P}_1 \boldsymbol{M}_1^{-1}, \tag{54}$$

$$\mathscr{A}_q = \mathscr{T}_{3,R} \underset{1}{\bullet} \boldsymbol{M}_{q-1} \underset{2}{\bullet} \boldsymbol{P}_q \underset{3}{\bullet} \boldsymbol{M}_q^{\mathrm{T}}, \text{ where } 2 \le q \le Q-1 \tag{55}$$

$$\boldsymbol{A}_Q = \boldsymbol{M}_{Q-1} \boldsymbol{P}_Q^{\mathrm{T}}, \tag{56}$$

where $\boldsymbol{M}_q$ is a non-singular $R \times R$ change of basis matrix.

This means that if a Q-order tensor admits a rank-R CPD, then its TTD involves a train of $(Q-2)$ third-order CPD(s) and has all identical TT-ranks such as $R_1 = \cdots = R_{Q-1} = R$. The factors can be derived straightforwardly from the TT-cores up to two change-of-basis matrices.

Remark 1: Note that each TT-core for $2 \le q \le Q-1$ follows a CPD coupled with its adjacent TT-cores (see Fig. 6).

The above theorem and the remark allow us to derive the factor retrieval scheme, called JIRAFE and presented in Algorithm 3 (Fig. 7) as a pseudo-code which minimises the following criterion:

$$\min_{\mathbb{M}, \mathbb{P}} \Big\{ \| \boldsymbol{A}_1 - \boldsymbol{P}_1 \boldsymbol{M}_1^{-1} \|_{\mathrm{F}} + \| \boldsymbol{A}_Q - \boldsymbol{M}_{Q-1} \boldsymbol{P}_Q \|_{\mathrm{F}} \tag{57}$$

$$+ \sum_{q=2}^{Q-2} \phi(\mathscr{A}_q) \Big\}, \tag{58}$$

where $\mathbb{M} = \{\boldsymbol{M}_1, \ldots, \boldsymbol{M}_Q\}$ and $\mathbb{P} = \{\boldsymbol{P}_1, \ldots, \boldsymbol{P}_Q\}$ and

$$\phi(\mathscr{A}_q) = \| \mathscr{A}_q - \mathscr{T}_{3,R} \underset{1}{\bullet} \boldsymbol{M}_{q-1} \underset{2}{\bullet} \boldsymbol{P}_q \underset{3}{\bullet} \boldsymbol{M}_q^{\mathrm{T}} \|_{\mathrm{F}}^2 . \tag{59}$$

In Algorithm 3, we denote by $\phi(\mathscr{A}|\boldsymbol{M})$ the criterion $\phi(\mathscr{A})$ when factor $\boldsymbol{M}$ has been previously estimated. The JIRAFE approach breaks this delicate optimisation problem into a set of Q low-dimensional and coupled LS criterion. Indeed, the JIRAFE methodology is essentially based on a single tridimensional LS optimisation, $Q-3$ bidimensional LS optimisations and the computation of two inverse $R \times R$ matrices. It is worth noting that any processing applicable to CPDs can be done at the level of the estimated third-order CPD TT-cores, using the JIRAFE principle, with a much lower complexity. Moreover, the ambiguities of the CPD-train model, presented in Theorem 2, are the same as the ones of the CPD. Indeed, in [36], a proof of the following result is given:

Remark 2: Just like the CPD, the CPD-Train model, given in Theorem 2, is characterised by the following indeterminacies:

(i) A unique column permutation matrix denoted by $\boldsymbol{\Pi}$.
(ii) Diagonal scaling matrices satisfying the following relation:

$$\boldsymbol{\Lambda}_1 \boldsymbol{\Lambda}_2 \boldsymbol{\Lambda}_3 \cdots \boldsymbol{\Lambda}_{Q-1} \boldsymbol{\Lambda}_Q = \boldsymbol{I}_R \tag{60}$$

where $\boldsymbol{\Lambda}_k$ is the scaling ambiguity for the k-mode factor $\boldsymbol{P}_k$.

3.2.5 Least squares Kronecker factorisation: Consider the following minimisation problem

$$\min_{\boldsymbol{A}, \boldsymbol{B}} \| \boldsymbol{X} - \boldsymbol{A} \boxtimes \boldsymbol{B} \|_{\mathrm{F}}, \tag{61}$$

where $\boldsymbol{A} \in \mathbb{C}^{I_2 \times R_2}$, $\boldsymbol{B} \in \mathbb{C}^{I_1 \times R_1}$ and $\boldsymbol{X} = \boldsymbol{A} \otimes \boldsymbol{B} + \boldsymbol{V} \in \mathbb{C}^{I_1 I_2 \times R_1 R_2}$, and $\boldsymbol{V}$ represents a zero-mean uncorrelated noise term. The solution of the problem in (62), is based on a rank-one matrix approximation (via SVD) of $\overline{\boldsymbol{X}}$ (a permuted version of $\boldsymbol{X}$, the construction of which was proposed in [38]). The problem in (62) becomes

$$\min_{\boldsymbol{a}, \boldsymbol{b}} \| \bar{\boldsymbol{X}} - \boldsymbol{b} \otimes \boldsymbol{a} \|_{\mathrm{F}}, \tag{62}$$

meaning to find the nearest rank-one matrix to $\bar{\boldsymbol{X}}$, where $\boldsymbol{a} = \mathrm{vec}(\boldsymbol{A}) \in \mathbb{C}^{I_2 R_2 \times 1}$ and $\boldsymbol{b} = \mathrm{vec}(\boldsymbol{B}) \in \mathbb{C}^{I_1 R_1 \times 1}$. In [39], the authors proposed a solution generalising [38] to a Kronecker product involving N factor matrices. Let us consider the case $N = 3$, usually encountered in practice. The problem then becomes

$$\min_{\boldsymbol{A}, \boldsymbol{B}} \| \boldsymbol{X} - \boldsymbol{A} \boxtimes \boldsymbol{B} \boxtimes \boldsymbol{C} \|_{\mathrm{F}}, \tag{63}$$

where $\boldsymbol{A} \in \mathbb{C}^{I_3 \times R_3}$, $\boldsymbol{B} \in \mathbb{C}^{I_2 \times R_2}$ and $\boldsymbol{C} \in \mathbb{C}^{I_1 \times R_1}$. The problem in (64) now becomes

$$\min_{\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}} \| \bar{\mathcal{X}} - \boldsymbol{c} \otimes \boldsymbol{b} \otimes \boldsymbol{a} \|_{\mathrm{F}}, \tag{64}$$

where $\boldsymbol{a} = \mathrm{vec}(\boldsymbol{A}) \in \mathbb{C}^{I_3 R_3 \times 1}$, $\boldsymbol{b} = \mathrm{vec}(\boldsymbol{B}) \in \mathbb{C}^{I_2 R_2 \times 1}$, $\boldsymbol{c} = \mathrm{vec}(\boldsymbol{C}) \in \mathbb{C}^{I_1 R_1 \times 1}$. We have that $\bar{\boldsymbol{x}} = \mathrm{vec}(\bar{\boldsymbol{X}})$ and $\overline{\mathcal{X}} = \mathcal{T}\{\bar{\boldsymbol{x}}\} \in \mathbb{C}^{I_1 R_1 \times I_2 R_2 \times I_3 R_3}$, where the operator $\mathcal{T}\{\cdot\}$ maps the elements of $\bar{\boldsymbol{x}}$ into $\bar{\mathcal{X}}$, as follows:

$$\bar{\boldsymbol{x}}_{q_1+(q_2-1)Q_1+(q_3-1)Q_1Q_2} \underset{\mathcal{T}\{\cdot\}}{\rightarrow} \bar{\mathcal{X}}_{q_1,q_2,q_3} \tag{65}$$

where $q_i = \{1, \ldots, Q_i\}$ and $Q_i = I_i R_i$, with $i = \{1, 2, 3\}$. Otherwise stated, we have $\bar{\mathcal{X}} = \text{reshape}(\bar{\boldsymbol{x}}, Q_1, Q_2, Q_3)$. Hence, finding the matrix triplet $\{\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}\}$ that solves (63) is equivalent to finding the vector triplet $\{\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}\}$ that solves (64), i.e. the solution of a Kronecker approximation problem can be recast as the solution to a rank-one tensor approximation problem, for which effective algorithms exist in the literature (see, e.g. [40–42]).

3.2.6 Least squares Khatri–Rao factorisation: Consider the following minimisation problem

$$\min_{\boldsymbol{A},\boldsymbol{B}} \| \boldsymbol{X} - \boldsymbol{A} \odot \boldsymbol{B} \|_{\text{F}}, \tag{66}$$

where $\boldsymbol{A} \in \mathbb{C}^{I \times R}$, $\boldsymbol{B} \in \mathbb{C}^{J \times R}$ and $\boldsymbol{X} = \boldsymbol{A} \odot \boldsymbol{B} + \boldsymbol{V} \in \mathbb{C}^{JI \times R}$, and $\boldsymbol{V}$ represents the zero-mean uncorrelated noise term. Note that problem (66) can be rewritten as

$$\min_{\boldsymbol{a}_r, \boldsymbol{b}_r} \sum_{i=1}^{R} \| \boldsymbol{x}_r - \boldsymbol{a}_r \boxtimes \boldsymbol{b}_r \|_2, \tag{67}$$

or, equivalently,

$$\min_{\boldsymbol{a}_r, \boldsymbol{b}_r} \sum_{i=1}^{R} \| \bar{\boldsymbol{X}}_r - \boldsymbol{b}_r \otimes \boldsymbol{a}_r \|_{\text{F}}, \tag{68}$$

where $\bar{\boldsymbol{X}}_r \in \mathbb{C}^{J \times I}$. Defining $\boldsymbol{U}_r \boldsymbol{\Sigma}_r \boldsymbol{B}_r^{\text{H}}$ as the singular value decomposition (SVD) of $\boldsymbol{X}_r$, estimates for $\boldsymbol{a}_r \in \mathbb{C}^I$ and $\boldsymbol{b}_r \in \mathbb{C}^J$, $(r = 1, \ldots, R)$ can be obtained by truncating the SVD of $\boldsymbol{X}_r$ to its dominant eigenmodes, i.e. [43]

$$\hat{\boldsymbol{b}}_r = \sqrt{\sigma_r^{(1)}} \boldsymbol{u}_r^{(1)} \text{ and } \hat{\boldsymbol{a}}_r^{(1)} = \sqrt{\sigma_r^{(1)}} \boldsymbol{v}_r^{(1)*}, \tag{69}$$

where $\boldsymbol{u}_r^{(1)} \in \mathbb{C}^J$ and $\boldsymbol{v}_r^{(1)} \in \mathbb{C}^I$ are the dominant first left and right singular vectors of $\boldsymbol{U}_i$ and $\boldsymbol{V}_i$, respectively, while $\sigma_r^{(1)}$ denotes the largest singular value of $\bar{\boldsymbol{X}}_r$. Hence, estimates of the full matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ that minimise (67) are obtained by repeating such a rank-1 approximation problem R times in parallel, one for each column $\boldsymbol{x}_r$, $r = 1, \ldots, R$.

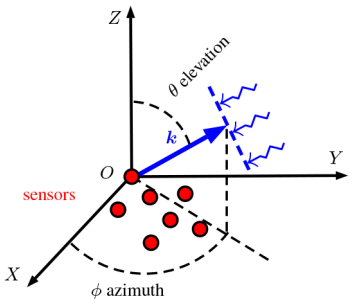

Fig. 8 *DOA estimation set-up*

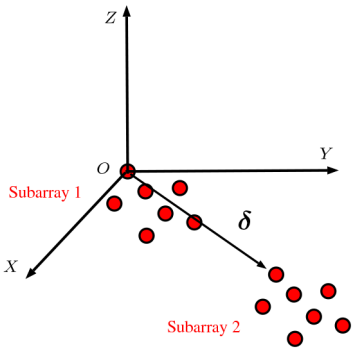

Fig. 9 *Acquisition set-up for ESPRIT*

4 Tensor-based DOA estimation with sensor arrays

A fundamental problem in array processing is the estimation of the DOAs for multiple sources impinging on an array of sensors. Assume that P narrow-band far-field sources are impinging on an array of L identical sensors ($P < L$, in general). The direction of arrival of a source p in a Cartesian coordinate system $OXYZ$ associated with the array is given by the unitary vector $\boldsymbol{k}_p = [\sin\theta_p \cos\phi_p \quad \sin\theta_p \sin\phi_p \quad \cos\theta_p]^{\text{T}}$, where θ and ϕ are the elevation and azimuth angles, respectively, as illustrated on Fig. 8.

With these notations, a snapshot of the array at time instant t is given by:

$$\boldsymbol{y}(t) = \sum_{p=1}^{P} \boldsymbol{a}(\boldsymbol{k}_p) s_p(t) + \boldsymbol{b}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{b}(t), \tag{70}$$

where $\boldsymbol{y}(t)$ is the $(L \times 1)$ output vector of the array, $\boldsymbol{A} = [\boldsymbol{a}(\boldsymbol{k}_1), \ldots, \boldsymbol{a}(\boldsymbol{k}_P)]$ is the $(L \times P)$ steering vectors matrix of the P sources, $\boldsymbol{x}(t) = [s_1(t), \ldots, s_P(t)]^{\text{T}}$ is the $(P \times 1)$ source signals vector at time instant t, and $\boldsymbol{b}(t)$ is a $(L \times 1)$ vector that accounts for the spatially and temporally white Gaussian noise on the array sensors. If K snapshots $t_1, t_2, \ldots, t_K$ are considered, the output of the array can be expressed in the matrix form as:

$$\begin{aligned} \boldsymbol{Y} &= [\boldsymbol{y}(t_1), \ldots, \boldsymbol{y}(t_K)] \\ &= \boldsymbol{A}(\boldsymbol{k}_1, \ldots, \boldsymbol{k}_P)[\boldsymbol{x}(t_1), \ldots, \boldsymbol{x}(t_K)] + \boldsymbol{B} \\ &= \boldsymbol{A}(\boldsymbol{k}_1, \ldots, \boldsymbol{k}_P)\boldsymbol{S}^{\text{T}} + \boldsymbol{B} \end{aligned} \tag{71}$$

with $\boldsymbol{S} = [\boldsymbol{s}_1, \ldots, \boldsymbol{s}_P]$, a $(K \times P)$ matrix gathering on its columns the K time samples for the P sources, $\boldsymbol{s}_p = [s_p(t_1), \ldots, s_p(t_K)]^{\text{T}}$, with $p = 1, \ldots, P$ and $\boldsymbol{B} = [\boldsymbol{b}(t_1), \ldots, \boldsymbol{b}(t_K)]$, a $(L \times K)$ noise matrix. The DOA estimation problem consists in finding $\boldsymbol{k}_1, \ldots, \boldsymbol{k}_P$ from the data $\boldsymbol{Y}$.

The use of multilinear algebra to solve this problem is inspired by the principle of ESPRIT algorithm introduced by Roy *et al.* in [44, 45]. The idea is to exploit the invariances in the array output data in order to create multilinear structures. These invariances can be intrinsic to the physics of the acquired signals (i.e. polarisation), to the acquisition setup (i.e. spatially shifted subarrays), or artificially created (i.e. matricisation/time–frequency transform of 1D signals).

The main idea of ESPRIT, when applied to DOA estimation is to employ two identical subarrays, the second one being spatially shifted compared to the first one by a known displacement vector $\boldsymbol{\delta}$, as illustrated in Fig. 9. The outputs of the first and the second subarrays, denoted by $\boldsymbol{y}_1(t)$ and $\boldsymbol{y}_2(t)$, respectively, can then be expressed as:

$$\boldsymbol{y}_1(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{b}_1(t), \tag{72}$$

$$\boldsymbol{y}_2(t) = \boldsymbol{A}\boldsymbol{\Phi}\boldsymbol{x}(t) + \boldsymbol{b}_2(t), \tag{73}$$

with $\boldsymbol{\Phi}$, a diagonal matrix having on its diagonals the phase-shifts between the two arrays for the P sources:

$$\boldsymbol{\Phi} = \begin{bmatrix} e^{j\frac{2\pi}{\lambda_1}\boldsymbol{k}_1^{\text{T}}\boldsymbol{\delta}} & & \\ & \ddots & \\ & & e^{j\frac{2\pi}{\lambda_P}\boldsymbol{k}_P^{\text{T}}\boldsymbol{\delta}} \end{bmatrix}.$$

In the expression of $\boldsymbol{\Phi}$, λ_p denotes the p th source wavelength. Thus, the DOA estimation problem comes down at estimating the $\boldsymbol{\Phi}$ matrix. To this end, the covariance matrix $\boldsymbol{R}_{yy}$ of the entire array:

$$\boldsymbol{y}(t) = \begin{bmatrix} \boldsymbol{y}_1(t) \\ \boldsymbol{y}_2(t) \end{bmatrix} = \begin{bmatrix} \boldsymbol{A} \\ \boldsymbol{A\Phi} \end{bmatrix} \boldsymbol{x}(t) + \begin{bmatrix} \boldsymbol{b}_1(t) \\ \boldsymbol{b}_2(t) \end{bmatrix}$$

is first computed as:

$$\boldsymbol{R}_{yy} = E\{\boldsymbol{y}(t)\boldsymbol{y}^{\mathrm{H}}(t)\} \simeq \frac{1}{K}\sum_{k=1}^{K} \boldsymbol{y}(t_k)\boldsymbol{y}^{\mathrm{H}}(t_k).$$

Then, we use the fact that the first P eigenvectors $[\boldsymbol{E}_1^{\mathrm{T}}, \boldsymbol{E}_2^{\mathrm{T}}]^{\mathrm{T}}$ of $\boldsymbol{R}_{yy}$ span the same subspace as the source steering vectors, i.e.

$$\begin{bmatrix} \boldsymbol{E}_1 \\ \boldsymbol{E}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{A} \\ \boldsymbol{A\Phi} \end{bmatrix} \boldsymbol{T} = \begin{bmatrix} \boldsymbol{AT} \\ \boldsymbol{A\Phi T} \end{bmatrix}, \tag{74}$$

where $\boldsymbol{T}$ is a non-singular matrix. As $\mathrm{span}\{\boldsymbol{A}\} = \mathrm{span}\{\boldsymbol{E}_1\} = \mathrm{span}\{\boldsymbol{E}_2\}$, there exist a non-singular matrix $\boldsymbol{\Psi}$ such that

$$\boldsymbol{E}_1\boldsymbol{\Psi} = \boldsymbol{E}_2. \tag{75}$$

From (74) and (75), it can be shown, after some algebraic manipulations (see e.g. [45]), that $\boldsymbol{\Phi}$ can be directly obtained by the diagonalisation of the matrix $\boldsymbol{\Psi} = \boldsymbol{E}_1^{\dagger}\boldsymbol{E}_2$. However, this least-squares (LS) estimate $\boldsymbol{\Psi}$ is biased because it assumes implicitly that $\boldsymbol{E}_1$ is perfectly known and the error is to be attributed only to $\boldsymbol{E}_2$. In practice, $\boldsymbol{E}_1$ and $\boldsymbol{E}_2$ are equally 'noisy' as they are both estimates of $\mathrm{span}\{\boldsymbol{A}\}$. A more appropriate criterion that takes into account noise on both $\boldsymbol{E}_1$ and $\boldsymbol{E}_2$ is the total least-squares (TLS) criterion. It can be formulated as finding two minimum Frobenius norm residual matrices $\boldsymbol{R}_1$ and $\boldsymbol{R}_2$, and a matrix $\boldsymbol{\Psi}$ satisfying $(\boldsymbol{E}_1 + \boldsymbol{R}_1)\boldsymbol{\Psi} = \boldsymbol{E}_2 + \boldsymbol{R}_2$. If we denote $\boldsymbol{B} = \boldsymbol{E}_1 + \boldsymbol{R}_1$, the TLS-ESPRIT problem can be expressed as the following minimisation problem:

$$\min_{\boldsymbol{\Psi}, \boldsymbol{B}} \left\| \begin{bmatrix} \boldsymbol{E}_1 \\ \boldsymbol{E}_2 \end{bmatrix} - \begin{bmatrix} \boldsymbol{B} \\ \boldsymbol{B\Psi} \end{bmatrix} \right\|_{\mathrm{F}}^2. \tag{76}$$

The solution of (77) is obtained by the eigenvalue decomposition of the $2P \times 2P$ matrix $[\boldsymbol{E}_1|\boldsymbol{E}_2]^{\mathrm{H}}[\boldsymbol{E}_1|\boldsymbol{E}_2]$.The ESPRIT algorithm presents several advantages over the 'classical' DOA estimation methods such as the beamforming or MUSIC [46, 47]. For example, it requires no calibration step and parameters estimates are obtained directly via two well-conditioned eigenproblems. Thus ESPRIT avoids the parameter grid search, inherent to MUSIC-like algorithms. However, in order to use ESPRIT one needs to estimate the data covariance matrix, which can be a difficult task in the case of highly correlated sources. Another major drawback of ESPRIT is the fact that it cannot handle more than two displaced subarrays.

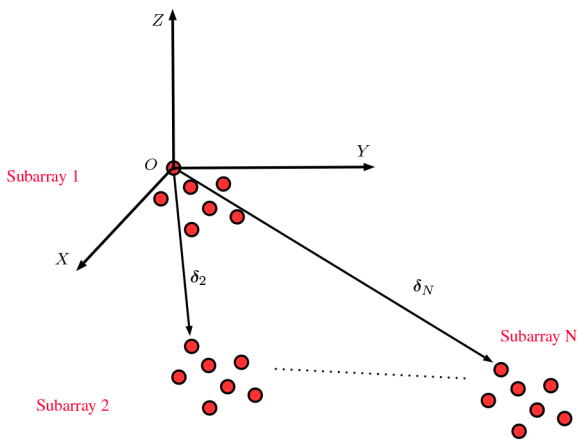

Fig. 10 *Acquisition set-up for the CP approach*

4.1 CPD-based array processing

To overcome these drawbacks, a DOA estimation approach was introduced in [5], capable of handling N arbitrarily displaced subarray, as illustrated in Fig. 10.

It is also the first time that a tensor-based algorithm is proposed for the DOA estimation problem. If $\boldsymbol{y}_1(t), \ldots, \boldsymbol{y}_N(t)$ are the outputs of the N subarrays and using notations similar to those used for ESPRIT, we have:

$$\begin{aligned}
\boldsymbol{y}_1(t) &= \sum_{p=1}^{P} \boldsymbol{a}(\boldsymbol{k}_p)s_p(t) + \boldsymbol{b}_1(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{b}_1(t), \\
\boldsymbol{y}_2(t) &= \sum_{p=1}^{P} \boldsymbol{a}(\boldsymbol{k}_p)\mathrm{e}^{j\frac{2\pi}{\lambda_p}\boldsymbol{k}_p^{\mathrm{T}}\boldsymbol{\delta}_2}s_p(t) + \boldsymbol{b}_2(t) = \boldsymbol{A}\boldsymbol{\Phi}_2\boldsymbol{x}(t) + \boldsymbol{b}_2(t), \\
&\vdots \\
\boldsymbol{y}_N(t) &= \sum_{p=1}^{P} \boldsymbol{a}(\boldsymbol{k}_p)\mathrm{e}^{j\frac{2\pi}{\lambda_p}\boldsymbol{k}_p^{\mathrm{T}}\boldsymbol{\delta}_N}s_p(t) + \boldsymbol{b}_2(t) = \boldsymbol{A}\boldsymbol{\Phi}_N\boldsymbol{x}(t) + \boldsymbol{b}_N(t),
\end{aligned}$$

where

$$\boldsymbol{\Phi}_n = \begin{bmatrix} \mathrm{e}^{j\frac{2\pi}{\lambda_1}\boldsymbol{k}_1^{\mathrm{T}}\boldsymbol{\delta}_n} & & \\ & \ddots & \\ & & \mathrm{e}^{j\frac{2\pi}{\lambda_P}\boldsymbol{k}_P^{\mathrm{T}}\boldsymbol{\delta}_n} \end{bmatrix}, \quad n = 2, \ldots, N.$$

If K snapshots are considered, the output of the entire array can be expressed as the $(NL \times K)$ matrix $\boldsymbol{Y}$:

$$\boldsymbol{Y} = \begin{bmatrix} \boldsymbol{Y}_1 \\ \boldsymbol{Y}_2 \\ \vdots \\ \boldsymbol{Y}_N \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_1 \\ \boldsymbol{A}_1\boldsymbol{\Phi}_2 \\ \vdots \\ \boldsymbol{A}_1\boldsymbol{\Phi}_N \end{bmatrix} \boldsymbol{S}^{\mathrm{T}} + \begin{bmatrix} \boldsymbol{B}_1 \\ \boldsymbol{B}_2 \\ \vdots \\ \boldsymbol{B}_N \end{bmatrix} = (\boldsymbol{A} \odot \boldsymbol{D})\boldsymbol{S}^{\mathrm{T}} + \boldsymbol{B}, \tag{77}$$

where $\boldsymbol{S} = [\boldsymbol{s}_1, \ldots, \boldsymbol{s}_P]$ is a $(K \times P)$ source matrix, $\boldsymbol{A} = [\boldsymbol{a}(\boldsymbol{k}_1), \ldots, \boldsymbol{a}(\boldsymbol{k}_P)]$ is the $(L \times P)$ steering vector matrix and

$$\boldsymbol{D} = [\boldsymbol{d}(\boldsymbol{k}_1), \ldots, \boldsymbol{d}(\boldsymbol{k}_P)] = \begin{bmatrix} \mathrm{diag}\{\boldsymbol{I}_P\}^{\mathrm{T}} \\ \mathrm{diag}\{\boldsymbol{\Phi}_2\}^{\mathrm{T}} \\ \vdots \\ \mathrm{diag}\{\boldsymbol{\Phi}_N\}^{\mathrm{T}} \end{bmatrix} \text{ is a } (N \times P)$$

matrix regrouping on its rows the diagonal elements of $\boldsymbol{\Phi}_n$, $\boldsymbol{B}$ is a noise matrix. In the light of the concepts introduced in Sections 3.1.2 and 3.1.3, (77) expresses a noisy CP decomposition of the sensor array output. Using the tensor formalism, (77) can be expressed as a $(L \times K \times N)$ tensor $\mathcal{Y}$:

$$\mathcal{Y} = \sum_{p=1}^{P} \boldsymbol{a}_p \otimes \boldsymbol{s}_p \otimes \boldsymbol{d}_p + \mathcal{B} = [[\boldsymbol{A}, \boldsymbol{S}, \boldsymbol{D}; \mathcal{I}_{3,P}]] + \mathcal{B}. \tag{78}$$

The tensor form (78) clearly highlights the three diversities/invariances of the array output: sensors – time – subarrays. To estimate the loading matrices $\boldsymbol{A}, \boldsymbol{S}, \boldsymbol{D}$, and consequently, the source DOAs, an approximate CP decomposition of the data tensor $\mathcal{Y}$ must be performed, using one of the many existing algorithms. Unlike the previously presented methods, the CP approach applies directly to the 'raw' data and does not require the estimation of the second-order statistics of the array output.

In [25], the ill-posedness of the tensor approximation problem induced by (78) is solved thanks to a constraint imposing that sources must be different, in the sense that either they should stem from different directions (e.g. if they are fully correlated like multipaths), or they should not be fully correlated, or both.

In [48], an approach that generalises the results given in [5] to an array that presents an arbitrary number of spatial invariances was proposed. Consider a subarray composed of L_1 isotropic

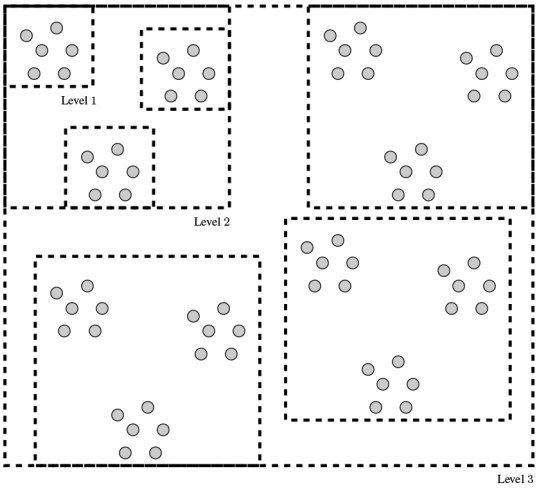

Fig. 11 *Multi-scale planar array with three hierarchical levels*

identical sensors indexed by $l_1 = 1, \ldots, L_1$. Consider then, L_2 identical replicas of this subarray, spatially translated to arbitrary locations. The L_2 distinct copies of the original subarray, indexed by $l_2 = 1, \ldots, L_2$, can now be seen as subarrays that together constitute a larger (higher-level) array. This proposed array structure can be further generalised by considering an additional level, composed of L_3 translated replicas of the previous sensor subarrays, indexed by $l_3 = 1, \ldots, L_3$. Let us generalise this scheme to a total of N such hierarchical levels, with the 'highest' level consisting of L_N subarrays indexed by $l_N = 1, \ldots, L_N$. It is worth noting that two different subarrays at a given level n need not be disjoint, i.e. they may have in common subarrays/sensors of the previous level $(n-1)$. However, if all subarrays at each level are disjoint, then the entire array will contain a total number of $L = L_1 L_2 \ldots L_N$ identical sensors. Fig. 11 illustrates a three-level array of co-planar sensors.

Consider first a narrow-band plane wave impinging on the array as described in the beginning of Section 4. Let us denote by $a_{l_1 l_2 \ldots l_N}$ its phase factor at the sensor indexed by $l_1, l_2, \ldots, l_N$ at the N various levels of the array. Define $\boldsymbol{d}_{l_n}^{(n)} = \left[x_{l_n}^{(n)}\ y_{l_n}^{(n)}\ z_{l_n}^{(n)}\right]^{\mathrm{T}}$, with $n = 1, \ldots, N$. With the notation introduced above, the spatial phase factor is given by:

$$a_{l_1 l_2 \ldots l_N}(\boldsymbol{k}) = \exp\left\{ j\frac{2\pi}{\lambda} \sum_{n=1}^{N} \boldsymbol{k}^{\mathrm{T}} \boldsymbol{d}_{l_n}^{(n)} \right\} = \prod_{n=1}^{N} \exp\left\{ j\frac{2\pi}{\lambda}\, \boldsymbol{k}^{\mathrm{T}} \boldsymbol{d}_{l_n}^{(n)} \right\}. \quad (79)$$

Thus, the array manifold for the entire sensor-array is

$$\boldsymbol{a}(\boldsymbol{k}) = \boldsymbol{a}_1(\boldsymbol{k}) \boxtimes \cdots \boxtimes \boldsymbol{a}_N(\boldsymbol{k}), \quad (80)$$

with

$$\boldsymbol{a}_n(\boldsymbol{k}) = \begin{bmatrix} \mathrm{e}^{\mathrm{j}(2\pi/\lambda)\boldsymbol{k}^{\mathrm{T}} \boldsymbol{d}_1^{(n)}} \\ \vdots \\ \mathrm{e}^{\mathrm{j}(2\pi/\lambda)\boldsymbol{k}^{\mathrm{T}} \boldsymbol{d}_{L_n}^{(n)}} \end{bmatrix} \quad (81)$$

an $L_n \times 1$ vector, $\forall n = 1, \ldots, N$.

Next, consider P narrow-band plane-waves, having traveled through a non-conductive homogeneous isotropic medium, impinging upon the array from directions $\boldsymbol{k}_p$, with $p = 1, \ldots, P$. Denote by $s_p(t)$ the time signal emitted by the pth narrow-band source. Then, the output at time t of the entire sensor-array can be expressed as an $L \times 1$ vector,

$$\boldsymbol{z}(t) = \sum_{p=1}^{P} (\boldsymbol{a}_1(\boldsymbol{k}_p) \boxtimes \cdots \boxtimes \boldsymbol{a}_N(\boldsymbol{k}_p)) s_p(t) + \boldsymbol{n}(t), \quad (82)$$

where $\boldsymbol{n}(t)$ is a complex-valued zero-mean additive white noise. Let us assume that we have at our disposal K snapshots at time instants, $t_1, t_2, \ldots, t_K$. Define the following $L_n \times P$ matrices:

$$\boldsymbol{A}_1 = [\boldsymbol{a}_1(\boldsymbol{k}_1), \ldots, \boldsymbol{a}_1(\boldsymbol{k}_P)] \quad (83)$$

$$\vdots$$

$$\boldsymbol{A}_N = [\boldsymbol{a}_N(\boldsymbol{k}_1), \ldots, \boldsymbol{a}_N(\boldsymbol{k}_P)], \quad (84)$$

and the $K \times P$ matrix:

$$\boldsymbol{S} = \begin{bmatrix} s_1(t_1) & s_2(t_1) & \ldots & s_P(t_1) \\ s_1(t_2) & s_2(t_2) & \ldots & s_P(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ s_1(t_K) & s_2(t_K) & \ldots & s_P(t_K) \end{bmatrix} = \left[\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_P\right]. \quad (85)$$

The collection of K snapshots of the array can then be organised into a $L \times K$ data matrix as

$$\boldsymbol{Z} = [\boldsymbol{z}(t_1), \ldots, \boldsymbol{z}(t_K)] = (\boldsymbol{A}_1 \odot \ldots \odot \boldsymbol{A}_N)\boldsymbol{S}^{\mathrm{T}} + \boldsymbol{N}, \quad (86)$$

where $\boldsymbol{N}$ is a $(L \times K)$ complex-valued matrix, modelling the noise on the entire array for all K temporal snapshots. Equation (86) reveals a $(N+1)$ -dimensional CPD structure. In the case where only one time-sample is available, i.e. matrix $\boldsymbol{S}$ is a $1 \times P$ vector, the data model given by (86) becomes

$$\boldsymbol{z} = (\boldsymbol{A}_1 \odot \ldots \odot \boldsymbol{A}_N)\boldsymbol{s} + \boldsymbol{n}, \quad (87)$$

with $\boldsymbol{z} = \boldsymbol{z}(t_1)$, $\boldsymbol{s} = \boldsymbol{s}(t_1) = (\boldsymbol{S}(1,:))^{\mathrm{T}}$ and $\boldsymbol{n} = \boldsymbol{N}(:,1)$. In the definitions above, we used the Matlab notations for columns and rows selection operators. Equation (87) is a vectorised representation of a N-dimensional CP data model. It is worth noting that if only one snapshot of the array is available, the $N+1$ CP model degenerates into an N -dimensional one. A discussion on the identifiability of these CPD models, for different scenarios can be found in [48].

For this multi-scale invariance array, a two stages DOA estimation procedure was also proposed in [48]. The first stage estimates the N steering vectors $\boldsymbol{a}_n(\boldsymbol{k}_p)$ $(n = 1, \ldots, N)$ for each of the P sources $(p = 1, \ldots, P)$, by exploiting the CPD structure (87) of the collected data. The second stage estimates the sources' direction-cosines $\boldsymbol{k}_p$, $p = 1, \ldots, P$ from the steering vectors obtained at the previous stage. To this end, the DOA estimation can be formulated as an optimisation problem, and a sequential procedure that exploits all the available information from the source's steering vectors encompassing all scale levels, is adopted.

Define the following cost functions:

$$\mathcal{J}_n(\boldsymbol{k}_p) = \| \hat{\boldsymbol{a}}_n^p - \boldsymbol{a}_n(\boldsymbol{k}_p) \|_2^2, \quad \text{with} \quad n = 1, \ldots, N, \quad (88)$$

where $\hat{\boldsymbol{a}}_n^p$ denotes the estimated steering vector at the nth level for the pth source. Estimating the DOAs for the pth source comes down to minimising the following criterion:

$$\mathcal{J}_N(\boldsymbol{k}_p) = \sum_{n=1}^{N} \mathcal{J}_n(\boldsymbol{k}_p). \quad (89)$$

This function is non-convex and highly non-linear with respect to the direction-cosines; hence, a direct local optimisation procedure would fail in most cases. In [48], a sequential strategy was adopted to minimise $\mathcal{J}_N(\boldsymbol{k}_p)$, progressing from one level to the next higher level, using iterative refinement of the direction-cosine estimates within each level. The method is based on the fact that, when noise-free, the N cost-functions in (89) have the same global minimum.

Assume that the level-1 subarrays' inter-sensor separations do not exceed half a wavelength. This assumption is essential to

obtaining a set of high-variance but unambiguous direction-cosine estimates. On the contrary, to achieve a practical advantage, it is important that the spatial displacement between any two subarrays of the highest level exceeds $\lambda/2$, where λ is the wavelength. This will produce estimates of lower variance but with cyclic ambiguity for the same set of direction-cosines. On the other hand, under the first assumption, the $\mathcal{J}_1(\boldsymbol{k}_p)$ function is unimodal on the support region of the DOAs. Therefore, any local optimisation procedure should converge towards the global minimum for the criterion. Thus, we obtain another set of estimates, now of high-variance but with no cyclic ambiguity, for the DOAs, to be denoted by $\boldsymbol{k}^*_{p,1}$ with $p = 1, \ldots, P$. These estimates will subsequently be used, in a second step, as the initial point for the minimisation of $\mathcal{J}_2(\boldsymbol{k}_p) = \mathcal{J}_1(\boldsymbol{k}_p) + \mathcal{J}_2(\boldsymbol{k}_p)$. As no assumption is made on the distances between the level-2 subarrays, $\mathcal{J}_2(\boldsymbol{k}_p)$ may present more than one local minimum. Hence, a good initial estimate is crucial for the optimisation procedure. The estimates obtained by the minimisation of $\mathcal{J}_2(\boldsymbol{k}_p)$, denoted by $\boldsymbol{k}^*_{p,2}$, are then used for the minimisation of $\mathcal{J}_3(\boldsymbol{k}_p) = \sum_{n=1}^{3} \mathcal{J}_n(\boldsymbol{k}_p)$, and so on, until the final estimates are obtained by the minimisation of $\mathcal{J}_N(\boldsymbol{k}_p)$. The proposed algorithm can be summarised as follows:

First stage:

• Estimate $\boldsymbol{A}_1, \ldots, \boldsymbol{A}_N$ by CP decomposition of the data $\boldsymbol{\mathcal{Z}}$ or $\boldsymbol{z}$ [see (86) or (87)].

Second stage:

• For $p = 1, \ldots, P$ and for $n = 1, \ldots, N$, compute

$$\boldsymbol{k}^*_{p,n} = \underset{\boldsymbol{k}_p}{\operatorname{argmin}}\ \mathcal{J}_n(\boldsymbol{k}_p) = \underset{\boldsymbol{k}_p}{\operatorname{argmin}} \sum_{i=1}^{n} \mathcal{J}_i(\boldsymbol{k}_p). \tag{90}$$

• *Output:* The estimated parameters for the P sources: $\hat{\boldsymbol{k}}_p = (\hat{u}_p, \hat{v}_p, \hat{w}_p) = \boldsymbol{k}^*_{p,N}$ with $p = 1, \ldots, P$.

Several CPD-based DOA estimation approaches, based on other diversity schemes have been proposed in the literature. In [49], polarisation of electromagnetic waves was exploited as an additional diversity, by using vector-sensor arrays capable of capturing the six components of the electromagnetic field. A coupled CPD approach for DOA estimation, using the multiple baselines in sparse arrays was introduced in [50]. High-order statistics (cumulants) were also used to create diversity in CPD based direction-finding algorithms [51].

While CPD remains the most commonly used decomposition in tensor-based array processing, other tensor decompositions made their way in. For example, the HOSVD has been used to develop multidimensional versions of the popular ESPRIT and MUSIC algorithms (see e.g. [52, 53]).

5 Tensor-based MHR

MHR [6, 54] is a classical signal processing problem that has found several applications in spectroscopy [55], radar communications [56], sensor array processing [5, 57], to mention a few. The multidimensional harmonic (MH) model can be viewed as the tensor-based generalisation of the one-dimensional harmonic one, resulting from the sampling process over a multidimensional regular grid. As a consequence, the Q-dimensional harmonic model needs the estimation of a large number (QR) of angular-frequencies of interest. We can easily note that the number of unknown parameters and the order of the associated data tensor grow with Q. Moreover, it is likely that the joint exploitation of multi-diversity/modality sensing technologies for data fusion [58–60] further increases the data tensor order. This trend is usually called the 'curse of dimensionality' [61–63] and the challenge here is to reformulate a high-order tensor as a set of low-order tensors over a graph. In this context, we observe an increasing interest for the tensor network theory (see [61] and references therein). Tensor network provides a useful and rigorous graphical representation of a high-order tensor into a factor graph where the nodes are low-order tensors, called cores, and the edges encode their dependencies, i.e. their common dimensions, often called 'rank'. In addition, tensor network allows to perform scalable/distributed computations over the cores [61]. In the tensor network framework, Hierarchical/tree Tucker [29, 64] and tensor train (TT) [28] are two popular representations of a high-order tensor into a graph-connected low-order (at most 3) tensors. In this section, we focus our effort on the TT formalism for its simplicity and compactness in terms of storage cost. Unlike the hierarchical Tucker model, TT is exploited in many practical and important contexts as, for instance, tensor completion [65], blind source separation [66], and machine learning [4], to mention a few. In the context of the MHR problem, this strategy has at least two advantages. First, it is well-known that the convergence of the ALSs algorithm becomes more and more difficult when the order increases [67–69]. To deal with this problem, applying ALS on lower-order tensors is preferable. The second argument is to exploit some latent coupling properties between the cores [4, 70] to propose new efficient estimators.

The maximum likelihood estimator (MLE) [71, 72] is the optimal choice from an estimation point of view since it is statistically efficient, i.e. its mean squared error (MSE) reaches the Cramér–Rao bound (CRB) [73] in the presence of noise. The main drawback of the MLE is its prohibitive complexity cost. This limitation is particularly severe in the context of a high-order data tensor. To overcome this problem, several low-complexity methods can be found in the literature. These methods may not reach, sometimes, the CRB, but they provide a significant gain in terms of the computational cost compared to the MLE. There are essentially two main families of methods. The first one is based on the factorisation of the data to estimate the well-known signal/noise subspace such as the estimation of signal parameters *via* rotational invariance techniques (ESPRIT) [45], the ND-ESPRIT [74], the improved multidimensional folding technique [75], and the CP-VDM [76]. The second one is based on the uniqueness property of the CPD. Indeed, factorising the data tensor thanks to the ALS algorithm [77] allows identification of the unknown parameters by Vandermonde-based rectification of the factor matrices.

5.1 Generalised Vandermonde canonical polyadic decomposition

The MH model assumes that the measurements can be modeled as the superposition of R undamped exponentials sampled on a Q-dimensional grid according to [6]

$$[\mathcal{X}]_{i_1 i_Q} = \sum_{r=1}^{R} \alpha_r \prod_{q=1}^{Q} z_{r,q}^{i_q - 1}, \quad 1 \le i_q \le I_q \tag{91}$$

in which the rth complex amplitude is denoted by α_r and the pole is defined by $z_{r,q} = \mathrm{e}^{\mathrm{j}\omega_{r,q}}$ where $\omega_{r,q}$ is the rth angular-frequency along the qth dimension, and we have $\boldsymbol{z}_q = \begin{bmatrix} z_{1,q} & z_{2,q} & \cdots & z_{R,q} \end{bmatrix}^{\mathrm{T}}$. Note that the tensor $\mathcal{X}$ is expressed as the linear combination of M rank-1 tensors, each of size $I_1 \times \times I_Q$ (the size of the grid), and follows a generalised Vandermonde CPD [78]:

$$\mathcal{X} = \mathcal{A} \underset{1}{\bullet} \boldsymbol{V}_1 \underset{2}{\bullet} \cdots \underset{Q}{\bullet} \boldsymbol{V}_Q \tag{92}$$

where $\mathcal{A}$ is a $R \times \cdots \times R$ diagonal tensor with $[\mathcal{A}]_{r,\ldots,r} = \alpha_r$ and

$$\boldsymbol{V}_q = \begin{bmatrix} \boldsymbol{v}(z_{1,q}) & \cdots & \boldsymbol{v}(z_{R,q}) \end{bmatrix}$$

is a $I_q \times R$ rank- R Vandermonde matrix, where

$$\boldsymbol{v}(z_{r,q}) = \begin{bmatrix} 1 & z_{r,q} & z_{r,q}^2 & \cdots & z_{r,q}^{I_q - 1} \end{bmatrix}^{\mathrm{T}}.$$

We define a noisy MH tensor model of order Q as:

$$\mathcal{Y} = \mathcal{X} + \sigma \mathcal{E}, \tag{93}$$

where $\sigma\mathcal{E}$ is the noise tensor, σ is a positive real scalar, and each entry $[\mathcal{E}]_{i_1 i_Q}$ follows an *i.i.d* circular Gaussian distribution $\mathcal{CN}(0, 1)$, and $\mathcal{X}$ has a canonical rank equal to R. The reader may

Require: $\mathcal{Y}, \boldsymbol{V}_1, \boldsymbol{V}_2, \{\mathbb{V}_1, \ldots, \mathbb{V}_K\}$, `CritStop`
Ensure: $\{z_{1,q}, \ldots, z_{R,q}\}$ for $1 \leq q \leq 3$
while `CritStop` and maximum iterations are not reached **do**
  $\boldsymbol{V}_3 = \text{unfold}_3\mathcal{Y} \cdot \left((\boldsymbol{V}_2 \odot \boldsymbol{V}_1)^{\mathsf{T}}\right)^{\dagger}$
  for $r = 1, \ldots, R$ **do**
    $\boldsymbol{v} := [\boldsymbol{V}_3]_r$
    while (`CritStop` is false) **do**
      $\boldsymbol{v} = \pi_{\mathbb{V}_K} \ldots \pi_{\mathbb{V}_1}(\boldsymbol{v})$
    end while
    $z_{r,3} = \min_z ||\boldsymbol{v}(z) - \boldsymbol{v}||^2$
  end for
  $\boldsymbol{V}_3 := \left[\boldsymbol{v}(z_{1,3}) \quad \ldots \quad \boldsymbol{v}(z_{R,3})\right]$
  $\boldsymbol{V}_2 = \text{unfold}_2\mathcal{Y} \cdot \left((\boldsymbol{V}_3 \odot \boldsymbol{V}_1)^{\mathsf{T}}\right)^{\dagger}$
  for $r = 1, \ldots, R$ **do**
    $\boldsymbol{v} := [\boldsymbol{V}_2]_r$
    while (`CritStop` is false) **do**
      $\boldsymbol{v} = \pi_{\mathbb{V}_K} \ldots \pi_{\mathbb{V}_1}(\boldsymbol{v})$
    end while
    $z_{r,2} = \min_z ||\boldsymbol{v}(z) - \boldsymbol{v}||^2$
  end for
  $\boldsymbol{V}_2 := \left[\boldsymbol{v}(z_{1,2}) \quad \ldots \quad \boldsymbol{v}(z_{R,2})\right]$
  $\boldsymbol{V}_1 = \text{unfold}_1\mathcal{Y} \cdot \left((\boldsymbol{V}_3 \odot \boldsymbol{V}_2)^{\mathsf{T}}\right)^{\dagger}$
  for $r = 1, \ldots, R$ **do**
    $\boldsymbol{v} := [\boldsymbol{V}_1]_r$
    while (`CritStop` is false) **do**
      $\boldsymbol{v} = \pi_{\mathbb{V}_K} \ldots \pi_{\mathbb{V}_1}(\boldsymbol{v})$
    end while
    $z_{r,1} = \min_z ||\boldsymbol{v}(z) - \boldsymbol{v}||^2$
  end for
  $\boldsymbol{V}_1 := \left[\boldsymbol{v}(z_{1,1}) \quad \ldots \quad \boldsymbol{v}(z_{R,1})\right]$
end while

Fig. 12 *Algorithm 4 Rectified ALS (RecALS)*

be referred to [74] for the case of damped signals, which is not addressed in this paper.

5.2 Algorithms

5.2.1 Vandermonde rectification of the ALS algorithm: Limit of the ALS algorithm for structured CPD: The CPD of any order-Q rank-R tensor $\mathcal{X}$ involves the estimation of Q factors $\boldsymbol{V}_q$ of size $I_q \times R$. As pointed out above, in the context of the MD-harmonic model, the factors $\boldsymbol{V}_q$ of the CPD are Vandermonde matrices. Consider, $\boldsymbol{Y}_q$, the qth mode unfolding [9] of tensor $\mathcal{Y}$, at the kth iteration with $1 \leq k \leq M$, M denoting the maximal number of iterations. The ALS algorithm solves alternatively for each of the Q dimensions the minimisation problem [23, 71]: $\min_{\boldsymbol{V}_q} \left|\left|\boldsymbol{Y}_q - \boldsymbol{V}_q\boldsymbol{S}_q\right|\right|^2$ where $\boldsymbol{S}_q^{\mathrm{T}} = \boldsymbol{V}_Q \odot \ldots \odot \boldsymbol{V}_{q+1} \odot \boldsymbol{V}_{q-1} \odot \ldots \odot \boldsymbol{V}_1$. It aims at approximating tensor $\mathcal{Y}$ by a tensor of rank R, hopefully close to $\mathcal{X}$. The LS solution conditionally to matrix $\boldsymbol{S}_q$ is given by $\boldsymbol{V}_q = \boldsymbol{Y}_q\boldsymbol{S}_q^{\dagger}$ where † stands for the pseudo-inverse. Now, remark that there is no reason that the above LS criterion promotes the Vandermonde structure in the estimated factors in the presence of noise. In other words, ignoring the structure in the CPD leads to estimate an excessive number of free parameters. This mismatched model dramatically decreases the estimation performance [79]. Hence there is a need to rectify the ALS algorithm to take into account the factor structure.

Rectified ALS (RecALS): The RecALS algorithm belongs to the family of lift-and-project algorithms [80, 81]. The optional lift step computes a low rank approximation and the projection step performs a rectification toward the desired structure. The RecALS algorithm is based on iterated projections and split LS criteria. Its algorithmic description is provided in Algorithm 4 for $Q = 3$. We insist that several iterations in the while loops are necessary, since restoring the structure generally increases the rank, and computing the low-rank approximation via truncated SVD generally destroys the structure (Fig. 12). The most intuitive way to exploit the Vandermonde structure is called column-averaging. Let $\omega = \frac{1}{i}\angle z^i$ where $\angle$ stands for the angle function. Define the sets

$$\mathbb{J} = \left\{\boldsymbol{v} = \frac{\boldsymbol{f}}{[\boldsymbol{f}]_1} : \boldsymbol{f} \in \mathbb{C}^I\right\} \tag{94}$$

$$\mathbb{A}_\ell = \left\{\boldsymbol{v}(z = e^{j\bar{\omega}}) : \bar{\omega} = \frac{1}{\ell}\sum_{i=1}^{\ell}\frac{1}{i}\angle[\boldsymbol{f}]_{i+1}\right\} \tag{95}$$

where $1 \leq \ell \leq I - 2$. This method exploits the Vandermonde structure in a heuristic way. So, the rectified strategy is to consider the iterated vector $\boldsymbol{f}(h) = \left(\pi_{\mathbb{A}_\ell}\pi_{\mathbb{J}}\right)^h(\boldsymbol{f})$. Another straightforward method is the periodogram maximisation. Under Gaussian noise and for a single tone $\alpha_r z_{r,q}^{i_q}$, the MLE is optimal and is given by the location of the maximal peak of the Fourier-periodogram [8, 82]. To increase the precision of the estimation, it is standard to use the well-known zero-padding technique at the price of an increase in computational cost. In [83], two strategies for Vandermonde-based rectification of the CP factors are provides but the RecALS principle is generic and flexible in the sense that any Vandermonde-based rectification methods can be used.

5.3 VTT-RecALS

In this section, the JIRAFE algorithm and the RecALS method are associated to solve the MHR problem. The VTT-RecALS estimator is based on the JIRAFE principle which is composed of two main steps.

(i) The first one is the computation of the TTD of the initial tensor. By doing this, the initial Q-order tensor is broken down into Q graph-connected third-order tensors, called TT-cores. This dimensionality reduction is an efficient way to mitigate the 'curse of dimensionality'. To reach this goal, the TT-SVD [28] is used as a first step.

(ii) The second step is dedicated to the factorisation of the TT-cores. Recall the main result given by Theorem 2, i.e. if the initial tensor follows a Q-order CPD of rank R, then the TT-cores for $2 \leq q \leq Q - 1$ follow coupled third-order CPD of rank R. Consequently, the JIRAFE minimises the following criterion over the physical quantities $\{\boldsymbol{V}_1, \ldots, \boldsymbol{V}_Q\}$ and over the latent quantities $\{\boldsymbol{M}_1, \ldots, \boldsymbol{M}_{Q-1}\}$:

$$\mathscr{C} = \left|\left|\boldsymbol{G}_1 - \boldsymbol{V}_1\boldsymbol{A}\boldsymbol{M}_1^{-1}\right|\right|_{\mathrm{F}}^2 + \left|\left|\boldsymbol{G}_Q - \boldsymbol{M}_{Q-1}\boldsymbol{V}_Q^{\mathrm{T}}\right|\right|_{\mathrm{F}}^2 \tag{96}$$

$$+ \sum_{q=2}^{Q-1}\left|\left|\mathcal{G}_q - \mathcal{I}_{3,R} \underset{1}{\bullet} \boldsymbol{M}_{q-1} \underset{2}{\bullet} \boldsymbol{V}_q \underset{3}{\bullet} \boldsymbol{M}_q^{-\mathrm{T}}\right|\right|_F^2, \tag{97}$$

where $\boldsymbol{A}$ is a $R \times R$ diagonal matrix with $[\boldsymbol{A}]_{r,r} = \alpha_r$.

The above cost function is the sum of coupled LS criteria. The aim of JIRAFE is to recover the original tensor factors using only the third-order tensors $\mathcal{G}_q$. Consequently, the JIRAFE approach adopts a straightforward sequential methodology, described in Fig. 13, to minimise the cost function $\mathscr{C}$.

We present in Algorithm 5, the pseudo-code of the VTT-RecALS algorithm, where RecALS_3 denotes the RecALS applied to a third-order tensor, while RecALS_2 denotes the RecALS applied to a third-order tensor using the knowledge of one factor. The VTT-RecALS algorithm is actually divided into two parts (Fig. 14). The first part is dedicated to dimensionality reduction, i.e. breaking the dimensionality of the high Q-order tensor into a set of third-order tensors using Theorem 2. The second part is dedicated to the factors retrieval from the TT-cores using the RecALS algorithm presented in the previous section. It is worth noting that the factors $\boldsymbol{V}_q$ are estimated up to a trivial (common) permutation ambiguity [84]. As noted in [37], since all the factors are estimated up to a unique column permutation matrix, the estimated angular-frequencies are automatically paired.

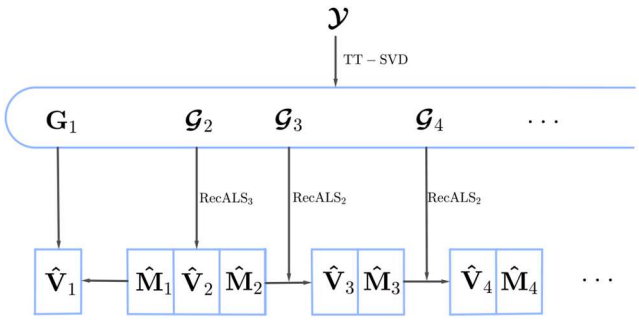

Fig. 13 *VTT-RecALS representation*

Input: $\mathcal{Y}$, R, `CritStop`
Output: Estimated parameters: $\{\mathbf{z}_1, \ldots, \mathbf{z}_Q\}$.

1: Dimensionality reduction:

$$[\boldsymbol{G}_1, \boldsymbol{\mathcal{G}}_2, \cdots, \boldsymbol{\mathcal{G}}_{Q-1}, \boldsymbol{G}_Q] = \text{TT-SVD}(\boldsymbol{\mathcal{X}}, R).$$

2: Factor retrieval:
3: For $q = 2$,

$$[\boldsymbol{M}_1, \boldsymbol{V}_2, \boldsymbol{M}_2, \mathbf{z}_2] = \text{RecALS}_3(\boldsymbol{\mathcal{G}}_2, R, \texttt{CritStop}).$$

4: **for** $q = 3 \cdots Q-1$ **do**
5: $\quad [\boldsymbol{V}_q, \boldsymbol{M}_q, \mathbf{z}_q] = \text{RecALS}_2(\boldsymbol{\mathcal{G}}_q, \boldsymbol{M}_{q-1}, M, \texttt{CritStop})$
6: **end for**
7: $\boldsymbol{V}_1 = \boldsymbol{G}_1 \boldsymbol{M}_1$, and $\boldsymbol{V}_Q = \boldsymbol{G}_Q^{\mathsf{T}} \boldsymbol{M}_{Q-1}^{-\mathsf{T}}$

Fig. 14 *Algorithm 5 VTT-RecALS*

6 Tensor-based MIMO wireless communication systems

As it is now well known, the use of multiple transmit and receive antennas, corresponding to a MIMO channel, allows to significantly improve the performance of wireless communication systems [85]. Such MIMO systems exploit space diversity to combat channel fading. Each information symbol is transmitted through different paths between the transmitter and the receiver, each path being characterised by an attenuation modeled by means of a fading coefficient of the channel. These channel coefficients are generally assumed to be independent and identically distributed (i.i.d.), which corresponds to an i.i.d. Rayleigh flat fading channel. In the presence of a slow fading environment, the channel coefficients are also assumed to be constant during P time slots, the duration of transmission of N data streams.

Another way to improve the performance consists in using space–time (ST) or space–time–frequency (STF) codings to get space, time and frequency diversities, corresponding to redundancies of the information symbols in each of these domains [33, 86, 87]. One objective of MIMO communication systems is to maximise the combined diversity gain while ensuring the best transmission rate possible, which corresponds to the fundamental diversity-rate tradeoff. Since the pioneering work [7], wireless communications have been a privileged field of application for tensors. Indeed, tensor-based approaches in the context of MIMO communication systems allow:

- to take simultaneously into account different diversities like space, time, frequency or polarisation, leading to high order tensor models for the signals received at destination;
- to design tensor coding like tensor space–time (TST) or tensor space–time–frequency (TSTF) codings;
- to develop semi-blind receivers for jointly estimating the information symbols and the channel, by exploiting uniqueness properties of tensor models, which is an advantage of tensor-based systems over matrix-based ones;
- to derive closed-form receivers under a priori knowledge of some transmitted symbols (semi-blind systems);
- to deal with point-to-point systems as well as multi-hop systems with relays.

A possible classification of MIMO wireless communication systems can be made according to:

- the considered channel access and multiplexing technology like CDMA (code-division multiple access), TDMA (time-division multiple access), FDMA (frequency-division multiplexing access), OFDMA (orthogonal frequency-division multiplexing access), or yet hybrid technologies like CDMA–OFDM or TDMA–FDMA;
- the assumptions on the channel: flat fading versus frequency-selective fading channel, according to whether all the frequency components of the transmitted signals are attenuated by the same fading or not; in this last case, the frequency dependency of the channel coefficients leads to a third-order MIMO channel tensor;
- the consideration or not of a sparsely scattering environment characterised by a multipath assumption with DoD (direction-of-departure) and DoA (direction-of-arrival) angles;
- the consideration or not of resource allocation under the form of matrix or tensor allocation;
- the type of coding (matrix/tensor), and the diversities taken into account (ST, STF codings);
- the type of communication system (supervised versus semi-blind, depending on the use or not of a training sequence);
- the type of receiver (iterative versus closed-form);
- the type of optimisation algorithm used at the receiver (ALS, Levenberg–Marquardt...);
- the type of communication: half-duplex (HD) versus full-duplex (FD) depending on the possibility or not of a simultaneous communication between two users, meaning that the communication can be in only one direction or in both directions, respectively;
- the existence or not of relays corresponding to multi-hop and single-hop communication systems, respectively;
- the relaying protocol used at the relays (amplify-and-forward (AF) versus decode-and-forward (DF) protocols).

Note that the DF protocol means that the signals received at the relay are decoded and then re-encoded before to be forwarded by the relay, whereas in the case of the AF protocol, the received signals are only amplified and forwarded by the relay. When two end nodes are communicating with each other through a relay, via bidirectionnal links, one can distinguish one-way and two-way relaying depending on the way the data streams are exchanged between the end nodes via the relay. With two-way relaying, the end nodes transmit simultaneously their data to the relay, and then the relay combines this information before broadcasting them to the end nodes, which allows to increase the system throughput (in bits/channel use).

A specificity of tensor-based communication systems lies in the development of new tensor models through the design of new systems. For instance, in the case of point-to-point systems, CONFAC (constrained factors) [88], generalised PARAFAC-Tucker (PARATUCK) [86], and nested PARAFAC models [34] were proposed. Most of these models can be viewed as constrained PARAFAC decompositions. See [11, 89, 90] for overviews of such models and corresponding point-to-point MIMO systems. In the context of cooperative or relay-assisted communications, several papers have proposed tensor-based models and algorithms for the supervised channel estimation problem [91, 92], considering two-hop AF relays and two-way MIMO relaying [93]. Tensor-based semi-blind receivers for the uplink multiuser cooperative scenario were first proposed in [94, 95]. For the two-hop MIMO AF relaying case, tensor modeling was exploited in several papers to derive semi-blind joint channel and symbol estimation algorithms [35, 96, 97]. More recently, tensor-based receivers have been

developed for multi-hop relaying systems, by exploiting the nested Tucker and coupled nested Tucker models [31, 98].

In this section, we make a brief presentation of three point-to-point systems and of one relaying system to illustrate the tensor-based approach in the context of wireless communication systems.

6.1 Khatri–Rao ST coding

Let us consider a MIMO wireless system with M transmit and K receive antennas. The information symbols to be transmitted by the M transmit antennas define a symbol matrix $\boldsymbol{S} \in \mathbb{C}^{N\times M}$, composed of N data streams, each one containing M symbols. These symbols are coded using a ST coding matrix $\boldsymbol{C} \in \mathbb{C}^{P\times M}$ such that the coded symbols to be transmitted are given by:

$$\boldsymbol{U} = \boldsymbol{C} \odot \boldsymbol{S} \in \mathbb{C}^{PN\times M}, \tag{98}$$

or equivalently:

$$u_{(p-1)N+n,m} = c_{p,m}s_{n,m}. \tag{99}$$

The coding matrix $\boldsymbol{C}$ introduces a time diversity in the sense that every symbol $s_{n,m}$ is duplicated P times, with P corresponding to the number of time slots, and the channel being assumed to be constant during the transmission, i.e. for $p \in [1, P]$. The coding (98) can be viewed as a simplified version of the Khatri–Rao space–time (KRST) coding proposed in [99].

The MIMO channel is assumed to be i.i.d. Rayleigh flat fading and defined as $\boldsymbol{H} \in \mathbb{C}^{K\times M}$, where $h_{k,m}$ is the complex fading coefficient between the mth transmit antenna and the kth receive antenna. The transmitter transmits the coded signals via the communication channel, so that the signal received by the kth receive antenna, during the pth time slot, and associated with the nth data stream, can be written as:

$$\begin{aligned} x_{k,n,p} &= \sum_{m=1}^{M} h_{k,m}u_{(p-1)N+n,m} + b_{k,n,p} \\ &= \sum_{m=1}^{M} h_{k,m}s_{n,m}c_{p,m} + b_{k,n,p}. \end{aligned} \tag{100}$$

These received signals form a third-order tensor $\mathcal{X} \in \mathbb{C}^{K\times N\times P}$, which satisfies a CPD model, with rank M and factor matrices $(\boldsymbol{H}, \boldsymbol{S}, \boldsymbol{C})$, and $\mathcal{B} \in \mathbb{C}^{K\times N\times P}$ represents the additive noise tensor. These matrices can be estimated using the tri-ALS algorithm. Assuming the coding matrix known at the receiver, it is possible to use the bi-ALS algorithm for jointly estimating the channel and symbol matrices. Another solution consists in applying the Khatri–Rao factorisation (KRF) algorithm [43] after a LS estimation of the Khatri–Rao product $\boldsymbol{H} \odot \boldsymbol{S}$ as:

$$\widehat{\boldsymbol{H} \odot \boldsymbol{S}} = \boldsymbol{X}_{KN\times P}(\boldsymbol{C}^{\mathrm{T}})^{\dagger}, \tag{101}$$

where $(.)^{\dagger}$ denotes the matrix right inverse, and $\boldsymbol{X}_{KN\times P}$ represents a tall mode-3 matrix unfolding of the received signals tensor $\mathcal{X} \in \mathbb{C}^{K\times N\times P}$ (see Section 3.2.6). It is worth noting that choosing a column-orthonormal coding matrix allows to simplify the right-inverse of $\boldsymbol{C}^{\mathrm{T}}$ as $(\boldsymbol{C}^{\mathrm{T}})^{\dagger} = \boldsymbol{C}^{*}(\boldsymbol{C}^{\mathrm{T}}\boldsymbol{C}^{*})^{-1} = \boldsymbol{C}^{*}$, with the necessary condition $M \leq P$ for uniqueness of this inverse.

6.2 Tensor space–time coding (TSTC)

We now present another MIMO system which can be viewed as a simplified version of the TST one [33] in the sense that no resource allocation matrix is considered. A TST coding $\mathcal{C} \in \mathbb{C}^{M\times R\times P}$ is used to code the symbol matrix $\boldsymbol{S} \in \mathbb{C}^{N\times R}$, composed of N data streams, each one containing R symbols.

The coded signals to be transmitted are given by:

$$\mathcal{U} = \mathcal{C} \underset{2}{\bullet} \boldsymbol{S} \in \mathbb{C}^{M\times N\times P} \tag{102}$$

or equivalently:

$$u_{m,n,p} = \sum_{r=1}^{R} c_{m,r,p}s_{n,r}, \tag{103}$$

which can be viewed as a linear combination of the R information symbols belonging to the nth data stream, with the coding coefficients $c_{m,r,p}$ as coefficients of the linear combination.

After transmitting the coded signals from the M transmit antennas via the channel $\boldsymbol{H} \in \mathbb{C}^{K\times M}$, the signals received at destination satisfy the following equation:

$$\begin{aligned} \mathcal{X} &= \mathcal{U} \underset{1}{\bullet} \boldsymbol{H} + \mathcal{B} \in \mathbb{C}^{K\times N\times P} \\ &= \mathcal{C} \underset{1}{\bullet} \boldsymbol{H} \underset{2}{\bullet} \boldsymbol{S} + \mathcal{B} \end{aligned} \tag{104}$$

or in scalar form:

$$\begin{aligned} x_{k,n,p} &= \sum_{m=1}^{M} u_{m,n,p}h_{k,m} + b_{k,n,p} \\ &= \sum_{m=1}^{M}\sum_{r=1}^{R} c_{m,r,p}h_{k,m}s_{n,r} + b_{k,n,p}, \end{aligned} \tag{105}$$

which defines a third-order tensor satisfying a Tucker-(2,3) model [11], with the coding tensor $\mathcal{C}$ as core tensor, and the factor matrices $(\boldsymbol{H}, \boldsymbol{S}, \boldsymbol{I}_P)$, where $\boldsymbol{I}_P$ is the identity matrix of order P. One factor matrix being an identity matrix, such a Tucker model is unique up to two non-singular matrices $(\boldsymbol{\Delta H}, \boldsymbol{\Delta S})$. If the coding tensor is assumed to be known at the receiver, the model ambiguity becomes a scalar factor related to a Kronecker product. This scalar ambiguity can be eliminated assuming the a priori knowledge of only one symbol, leading to a semi-blind Kronecker receiver [90].

Then, the channel and symbol matrices can be jointly estimated using the Kronecker factorisation (KF) algorithm (see Section 3.2.5), after the LS estimation of the Kronecker product $\boldsymbol{H} \boxtimes \boldsymbol{S}$ as:

$$\widehat{\boldsymbol{H} \boxtimes \boldsymbol{S}} = \boldsymbol{X}_{KN\times P}\boldsymbol{C}^{\dagger}_{MR\times P}, \tag{106}$$

where $\boldsymbol{C}_{MR\times P}$ denotes a tall mode-3 matrix unfolding of the coding tensor. Choosing this coding tensor such that its matrix unfolding $\boldsymbol{C}_{MR\times P}$ is row-orthonormal allows to simplify the right-inverse as $\boldsymbol{C}^{\dagger}_{MR\times P} = \boldsymbol{C}^{*}_{P\times MR}[\boldsymbol{C}_{MR\times P}(\boldsymbol{C}^{*}_{P\times MR}]^{-1} = \boldsymbol{C}^{*}_{P\times MR}$, with the necessary condition $MR \leq P$ for uniqueness of this inverse.

Comparing the two presented MIMO systems which are simplified versions of the KRST and TST coding-based systems ([99] and [33], respectively), one can conclude that both systems lead to a third-order tensor for the received signals, with the same dimensions. This tensor satisfies a PARAFAC model, in the first case, while it satisfies a Tucker-(2,3) model in the second case, which induces different ambiguities. Another important difference is that with the TST coding, the number (R) of symbols per data stream can be different from the number (M) of transmit antennas, which is not the case with the KRST coding.

In a practical situation where the coding used at the transmitter is known at the receiver, one can draw the supplementary conclusion that both systems offer the possibility of jointly and semi-blindly estimating the channel and symbol matrices by means of a closed-form receiver. However, in order to eliminate model ambiguities, the Kronecker factorisation algorithm used with TST coding presents the advantage over the Khatri–Rao factorisation algorithm used with KRST coding, to require the a priori knowledge of only one symbol for the Kronecker model while the a priori knowledge of M symbols is needed for the PARAFAC model.

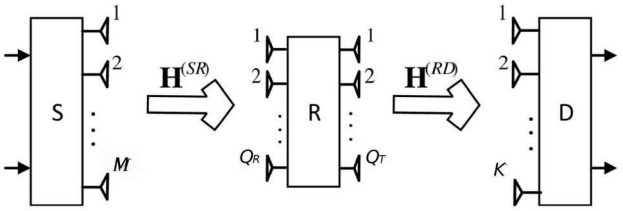

Fig. 15 *Block diagram of a one-way two-hop MIMO relay system*

6.3 TSTF coding

A more general and flexible tensor-based transceiver scheme for space, time and frequency transmit signalling was proposed in [86], referred to as TSTF coding. By relying on a new class of tensor models, namely, the generalised PARATUCK-(N_1, N) and the Tucker-(N_1, N) models, the TSTF system combines a fifth-order coding tensor with a fourth-order allocation tensor. It can also be seen as a generalisation of three previous tensor-based ST/TST/STF coding schemes, offering new performance/complexity tradeoffs and space, time and frequency allocation flexibility.

Herein, the transmission time is decomposed into P time-slots (data blocks) of N symbol periods, each one being composed of J chips. At each symbol period n of the pth block, the transceiver transmits a linear combination of the nth symbols of certain data streams, using a set of transmit antennas and of sub-carriers. The coding is carried out by means of a fifth-order code tensor $\mathscr{W} \in \mathbb{C}^{M\times R\times F\times P\times J}$ whose dimensions are the numbers of transmit antennas (M), data streams (R), sub-carriers (F), time blocks (P), and chips (J). An allocation tensor $\mathscr{C} \in \mathbb{C}^{M\times R\times F\times P}$ determines the transmit antennas and the sub-carriers that are used, as well as the data streams transmitted in each block p. As an example, $c_{m,r,f,p} = 1$ means that the data stream r is transmitted using the transmit antenna m, with the sub-carrier f, during the time block p.

The TSTF-coded signals to be transmitted are given by:

$$\mathscr{U} = \mathscr{G} \underset{1}{\bullet} \boldsymbol{S}, \text{ with } \mathscr{G} = \mathscr{W} \underset{\{m,r,f,p\}}{\star} \mathscr{C}, \tag{107}$$

or, alternatively, in scalar form

$$u_{m,n,f,p,j} = \sum_{r=1}^{R} w_{m,r,f,p,j} s_{n,r} c_{m,r,f,p}, \tag{108}$$

where $\mathscr{G} \in \mathbb{C}^{M\times R\times F\times P\times J}$ is the effective code-allocation tensor, given by the Hadamard product between the code tensor $\mathscr{W} \in \mathbb{C}^{M\times R\times F\times P\times J}$ and the allocation tensor $\mathscr{C} \in \mathbb{R}^{M\times R\times F\times P}$ along their common modes, i.e.

$$g_{m,r,f,p,j} = w_{m,r,f,p,j} c_{m,r,f,p}.$$

Assuming a flat Rayleigh fading propagation channel, the discrete-time baseband-equivalent model for the signal received at the kth receive antenna during the jth chip period of the nth symbol period of the pth block, and associated with the fth sub-carrier, is given by:

$$x_{k,n,f,p,j} = \sum_{m=1}^{M}\sum_{r=1}^{R} h_{k,m,f} s_{n,r} w_{m,r,f,p,j} c_{m,r,f,p} + b_{k,n,f,p,j}.$$

The above equation can be written as the following generalised Tucker-(2,5) model [11]

$$\mathscr{X} = \mathscr{G} \underset{1}{\bullet} \mathscr{H} \underset{2}{\bullet} \boldsymbol{S}. \tag{109}$$

Joint channel and symbol estimation is carried out by exploiting the following unfolding of the received signal tensor:

$$\boldsymbol{X}_{NK\times JPF} = (\boldsymbol{S} \boxtimes \boldsymbol{H}_{K\times MF})\boldsymbol{G}_{RMF\times JPF}, \tag{110}$$

where $\boldsymbol{G}_{RMF\times JPF}$ is a matrix unfolding of the code-allocation tensor defined in (108). By properly designing this tensor to ensure its right-invertibility, the channel and symbol matrices are estimated in closed form from a LS estimation of the Kronecker product

$$\widehat{\boldsymbol{S} \boxtimes \boldsymbol{H}}_{K\times MF} = \boldsymbol{X}_{NK\times JPF} \boldsymbol{G}^{\dagger}_{RMF\times JPF}. \tag{111}$$

The use of a pilot-assisted channel estimation and of a semi-blind ALSs-based receiver are discussed in [86].

6.4 MIMO relaying systems with tensor coding

The benefits of TSTC have been extended to MIMO relaying systems in [31], where semi-blind receivers have been derived for the joint estimation of the source-relay and relay-destination channels, in addition to symbol estimation. Consider a one-way two-hop MIMO relaying communication system shown in Fig. 15, where M, and K denote the numbers of antennas at the source (S), and destination (D) nodes, respectively, while Q_R and Q_T represent the numbers of receive and transmit antennas at the relay, respectively. The source–relay channel $\boldsymbol{H}^{(SR)} \in \mathbb{C}^{Q_R\times M}$ and the relay–destination channel $\boldsymbol{H}^{(RD)} \in \mathbb{C}^{K\times Q_T}$ are assumed to be flat fading and quasi-static, i.e. constant during the transmission protocol which is divided into two consecutive phases. Assume that the direct link between the source and destination is absent or negligible. During the first hop, the transmission is composed of N time-blocks associated with N symbol periods, R data streams being transmitted per time-block. The transmitted symbol matrix $\boldsymbol{S} \in \mathbb{C}^{N\times R}$ contains R data streams composed of N symbols each. During the time-block n, each antenna m of the source node transmits a combination of R information symbols $s_{n,r}, r = 1, \ldots, R$, to the relay, after a TSTC [31] by means of a third-order tensor $\mathscr{C}^{(S)} \in \mathbb{C}^{M\times P\times R}$, which introduces ST redundancies, since each symbol $s_{n,r}$ is repeated P times over the M transmit antennas, P being the source code length. During the second hop, the source remains silent and the relay uses a second three dimensional coding $\mathscr{C}^{(R)} \in \mathbb{C}^{Q_T\times J\times Q_R}$ before forwarding the Q_R received signals to the destination using Q_T transmit antennas. Not that this second tensor coding consists in repeating the signals received at the relay J times, over the Q_T transmit antennas, J being the relay code length. The tensor $\mathscr{T} \in \mathbb{C}^{Q_T\times J\times P\times N}$ of signals transmitted by the relay is defined as

$$t_{q_T,j,p,n} = \sum_{q_R=1}^{Q_R}\sum_{m=1}^{M}\sum_{r=1}^{R} c^{(R)}_{q_T,j,q_R} h^{(SR)}_{q_R,m} c^{(S)}_{m,p,r} s_{n,r}. \tag{112}$$

After transmission through the relay–destination channel, the noiseless signals received by K antennas at destination satisfies the following equation:

$$\begin{aligned} x_{k,j,p,n} &= \sum_{q_T=1}^{Q_T} h^{(RD)}_{k,q_T} t_{q_T,j,p,n} \\ &= \sum_{q_T=1}^{Q_T}\sum_{q_R=1}^{Q_R}\sum_{m=1}^{M}\sum_{r=1}^{R} h^{(RD)}_{k,q_T} c^{(R)}_{q_T,j,q_R} h^{(SR)}_{q_R,m} c^{(S)}_{m,p,r} s_{n,r}. \end{aligned} \tag{113}$$

Comparing (113) with (31), we can conclude that the tensor $\mathscr{X} \in \mathbb{C}^{K\times J\times P\times N}$ of noiseless signals received at destination satisfies a NTT(4) model. Fig. 16 illustrates the NTT(4) for the noiseless signal tensor $\mathscr{X}$ received at the destination. We have to note that the four dimensions of the tensor $\mathscr{X}$ are associated with four signal diversities (space k, relay code j, source code p, time n).

A special case of this model is the one presented in [35, 96], where the coding tensors $\mathscr{C}^{(S)}$ and $\mathscr{C}^{(R)}$ have a special structure where their frontal slices are diagonal matrices, with $M = R$ and $Q_R = Q_T$. Physically, this means that the source and the relay transmits a single data stream, in contrast to the NTT(4) model in (114) where R and Q_R signals are combined at the source and relay,

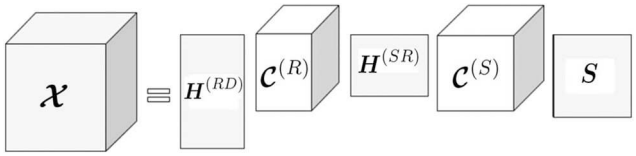

Fig. 16 *NTT(4) model for a one-way two-hop MIMO relaying system with tensor coding*

Table 1 Summary of the tensor decompositions and algorithms used in the applications

Application	Tensor decompositions	Algorithms
DOA	Canonical polyadic decomposition	alternating least squares
MHR	generalised Vandermonde CPD	rectified ALS (RecALS), Vandermonde tensor-train RecALS (VTT-RecALS)
MIMO	CPD, Tucker-(N_1, N), nested tucker train (NTT)	Kronecker factorisation (KF), Khatri–Rao factorisation (KRF)

respectively, before transmission. In this case, the noiseless received signal tensor at the destination node reduces to

$$x_{k,j,p,n} = \sum_{q_{\mathrm{T}}=1}^{Q_T} \sum_{m=1}^{M} h^{(\mathrm{RD})}_{k,q_{\mathrm{T}}} c^{(\mathrm{R})}_{q_{\mathrm{T}},j} h^{(\mathrm{SR})}_{q_{\mathrm{R}},m} c^{(\mathrm{S})}_{m,p} s_{n,r}, \tag{114}$$

which corresponds to a nested CP model. This can be concluded by comparing (114) with (50).

Assuming that the coding tensors $\mathscr{C}^{(\mathrm{S})}$ and $\mathscr{C}^{(\mathrm{R})}$ are known at the destination node (i.e. by the receiver), semi-blind receivers for jointly estimating the symbol matrix ($\boldsymbol{S}$) and the individual channels $\left(\boldsymbol{H}^{(\mathrm{SR})}, \boldsymbol{H}^{(\mathrm{RD})}\right)$ are proposed in [31]. We refer the interested reader to this reference for further details on the receiver algorithms.

We summarised in Table 1 the different tensor decompositions and algorithms used in the three applications presented in this paper.

7 Conclusion

Collecting data from multisensor signal processing can be naturally interpreted in the multilinear algebra context. In this work, three important array signal processing-based applications are exposed in which the tensor-based framework has proven to be useful. Namely, the problems of interest are (i) array signal processing for localisation, (ii) MHR and (iii) MIMO processing for wireless communications. In this overview paper, we present systematically in a synthetic way the necessary mathematical material to ensure a good understanding of the problems of interest.

8 References

[1] Harshman, R.A.: 'Foundations of the parafac procedure: models and conditions for an explanatory multimodal factor analysis', *UCLA Work. Pap. Phonetics*, 1970, **16**, pp. 1–84

[2] Carroll, J.D., Chang, J.J.: 'Analysis of individual differences in multidimensional scaling via N-way generalization of Eckart-Young decomposition', *Psychometrika*, 1970, **35**, (3), pp. 283–319

[3] Acar, E., Yener, B.: 'Unsupervised multiway data analysis: a literature survey', *IEEE Trans. Knowl. Data Eng.*, 2008, **21**, (1), pp. 6–20

[4] Sidiropoulos, N., De Lathauwer, L., Fu, X.*, et al.*: 'Tensor decomposition for signal processing andmachine learning', *IEEE Trans. Signal Process.*, 2017, **65**, pp. 3551–3582

[5] Sidiropoulos, N.D., Bro, R., Giannakis, G.B.: 'Parallel factor analysis in sensor array processing', *IEEE Trans. Signal Process.*, 2000, **48**, (8), pp. 2377–2388

[6] Jiang, T., Sidiropoulos, N.D., Berge, J.M.F.T.: 'Almost-sure identifiability of multidimensional harmonic retrieval', *IEEE Trans. Signal Process.*, 2001, **49**, (9), pp. 1849–1859

[7] Sidiropoulos, N.D., Bro, R., Giannakis, G.B.: 'Blind PARAFAC receivers for DSCDMA systems', *IEEE Trans. Signal Process.*, 2000, **48**, (3), pp. 810–823

[8] Nion, D., Sidiropoulos, N.D.: 'Tensor algebra and multidimensional harmonic retrieval in signal processing for mimo radar', *IEEE Trans. Signal Process.*, 2010, **58**, (11), pp. 5693–5705

[9] Comon, P.: 'Tensors: a brief introduction', *IEEE Signal. Proc. Mag.*, 2014, **31**, (3), pp. 44–53, special issue on BSS

[10] De Lathauwer, L., Moor, B.D., Vandewalle, J.: 'A multilinear singular value decomposition', *SIAM J. Matrix Anal. Appl.*, 2000, **21**, (4), pp. 1253–1278

[11] Favier, G., de Almeida, A.: 'Overview of constrained PARAFAC models', *EURASIP J. Adv. Signal Process.*, 2014, **5**, pp. 1–41

[12] Hitchcock, F.L.: 'The expression of a tensor or a polyadic as a sum of products', *J. Math Phys.*, 1927, **6**, (1), pp. 165–189

[13] Landsberg, J.M.: 'Tensors: geometry and applications', in David, Cox (Ed.): '*Graduate studies in mathematics*' (American Mathematical Society, Providence, Rhode Island, USA 2012), vol. **128**

[14] Hackbusch, W.: 'Tensor spaces and numerical tensor Calculus', in Randolph E., Bank (Ed.): '*Series in computational mathematics*' (Springer, Berlin, Heidelberg, 2012)

[15] Kolda, T.G., Bader, B.W.: 'Tensor decompositions and applications', *SIAM Rev.*, 2009, **51**, (3), pp. 455–500

[16] Kruskal, J.B.: 'Three-way arrays: rank and uniqueness of trilinear decompositions', *Linear Algebra Appl.*, 1977, **18**, pp. 95–138

[17] Sidiropoulos, N.D., Bro, R.: 'On the uniqueness of multilinear decomposition of N-way arrays', *J. Chemo*, 2000, **14**, pp. 229–239

[18] Stegeman, A., Sidiropoulos, N.: 'On Kruskal's uniqueness condition for the CP decomposition', *Linear Algebra Appl.*, 2007, **420**, (2–3), pp. 540–552

[19] Comon, P., Mourrain, B., Lim, L.H.*, et al.*: 'Genericity and rank deficiency of high order symmetric tensors'. 2006 IEEE Int. Conf. on Acoustics Speech and Signal Processing Proc., Toulouse, 2006

[20] Comon, P., Berge, J.M.F.T., De Lathauwer, L.*, et al.*: 'Generic and typical ranks of multi-way arrays', *Linear. Algebra. Appl.*, 2009, **430**, (11–12), pp. 2997–3007

[21] Chiantini, L., Ottaviani, G., Vannieuwenhoven, N.: 'An algorithm for generic and low-rank specific identifiability of complex tensors', *SIAM J. matrix Anal. Appl.*, 2014, **35**, (4), pp. 1265–1287

[22] Qi, Y., Comon, P., Lim, L.H.: 'Uniqueness of nonnegative tensor approximations', *IEEE Trans. Inf. Theory*, 2016, **62**, (4), pp. 2170–2183, arXiv:1410.8129

[23] Comon, P., Luciani, X., De Almeida, A.L.F.: 'Tensor decompositions, alternating least squares and other tales', *J. Chemometrics*, 2009, **23**, (7–8), pp. 393–405

[24] Lim, L.H., Comon, P.: 'Blind multilinear identification', *IEEE Trans. Inf. Theory*, 2014, **60**, (2), pp. 1260–1280

[25] Sahnoun, S., Comon, P.: 'Joint source estimation and localization', *IEEE Trans. Signal. Proc.*, 2015, **63**, (10), pp. 2485–2495. hal-01005352

[26] Lim, L.H., Comon, P.: 'Nonnegative approximations of nonnegative tensors', *J. Chemometrics*, 2009, **23**, pp. 432–441

[27] Qi, Y., Comon, P., Lim, L.H.: 'Semialgebraic geometry of nonnegative tensor rank', *SIAM J. Matrix Anal. Appl.*, 2016, **37**, (4), pp. 1556–1580. hal-01763832

[28] Oseledets, I.V.: 'Tensor-train decomposition', *SIAM J Sci. Comput.*, 2011, **33**, (5), pp. 2295–2317

[29] Oseledets, I.V., Tyrtyshnikov, E.E.: 'Breaking the curse of dimensionality, or how to use SVD in many dimensions', *SIAM J. Sci. Comput.*, 2009, **31**, pp. 3744–3759

[30] Zniyed, Y., Boyer, R., de Almeida, A.L.F.*, et al.*: 'A TT-based hierarchical framework for decomposing high-order tensors', *SIAM J. Sci. Comput.*, 2020, **42**, (2), pp. A822–A848

[31] Favier, G., Fernandes, C., de Almeida, A.: 'Nested Tucker tensor decomposition with application to MIMO relay systems using tensor space-time coding (TSTC)', *Elsevier Signal Process.*, 2016, **128**, (4), pp. 318–331

[32] Rocha, D., Favier, G., Fernandes, C.: 'Closed-form receiver for multi-hop MIMO relay systems with tensor space-time coding', *J. Commun. Inf. Syst.*, 2019, **34**, (1), pp. 50–54

[33] Favier, G., da Costa, M.N., de Almeida, A.*, et al.*: 'Tensor space-time (TST) coding for (MIMO) wireless communication systems', *Elsevier Signal Process.*, 2012, **92**, (4), pp. 1079–1092

[34] de Almeida, A., Favier, G.: 'Double Khatri-Rao space-time-frequency coding using semi-blind PARAFAC based receiver', *IEEE Signal Process. Lett.*, 2013, **20**, (5), pp. 471–474

[35] Ximenes, L., Favier, G., de Almeida, A.: 'Closed-form semi-blind receiver for MIMO relay systems using double Khatri-Rao space-time coding', *IEEE Signal Process. Lett.*, 2016, **23**, (3), pp. 316–320

[36] Zniyed, Y., Boyer, R., de Almeida, A.L.F.*, et al.*: 'High-order tensor estimation via trains of coupled third-order CP and Tucker decompositions', *Linear Algebr. Appl.*, 2020, **588**, pp. 304–337

[37] Zniyed, Y., Boyer, R., de Almeida, A.L.F.*, et al.*: 'High-order CPD estimation with dimensionality reduction using a tensor train model'. Proc. of the 26th European Signal Processing Conf., Rome, Italy, 2018

[38] Van.Loan, C.F., Pitsianis, N.: 'Approximation with Kronecker products', Moonen, M.S., Golub, G.H., De.Moor, B.L.R. (Eds).: '*Linear algebra for large scale and real-time applications*'. (Springer Netherlands, Dordrecht, 1993), pp. 293–314

[39] Wu, K.K., Yam, Y., Meng, H.*, et al.*: 'Kronecker product approximation with multiple factor matrices via the tensor product algorithm'. Proc. IEEE Int. Conf. on Systems, Man, and Cybernetics (SMC 2016), Budapest, Hungary, 2016, pp. 004277–004282

[40] Zhang, T., Golub, G.H.: 'Rank-one approximation to high order tensors', *SIAM J. Matrix Anal. Appl.*, 2001, **23**, (2), pp. 534–550

[41] Kofidis, E., Regalia, P.A.: 'On the best rank-1 approximation of higher-order supersymmetric tensors', *SIAM J. Matrix Anal. Appl.*, 2002, **23**, (3), pp. 863–884

[42] da Silva, A.P., Comon, P., de Almeida, A.L.F.: 'A finite algorithm to compute rank-1 tensor approximations', *IEEE Signal Process. Lett.*, 2016, **23**, (7), pp. 959–963

[43] Kibangou, A.Y., Favier, G.: 'Non-iterative solution for PARAFAC with a Toeplitz matrix factor'. Proc. EUSIPCO, Glasgow Scotland, 2009

[44] Roy, R., Paulraj, A., Kailath, T.: 'ESPRIT–a subspace rotation approach to estimation of parameters of cisoids in noise', *IEEE Trans. Acoust. Speech Signal Process.*, 1986, **34**, (5), pp. 1340–1342

[45] Roy, R., Kailath, T.: 'ESPRIT-estimation of signal parameters via rotational invariance techniques', *IEEE Trans. Acoust. Speech Signal Process.*, 1989, **37**, (7), pp. 984–995

[46] Bienvenu, G., Kopp, L.: 'Principe de la goniométrie passive adaptative'. 1979 7ème Colloque sur le traitement du signal et des images (GRETSI). (Groupe d'Etudes du Traitement du Signal et des Images, 1979

[47] Schmidt, R.O.: 'A signal subspace approach to multiple emitter location and spectral estimation', Stanford Univ, Stanford, CA, 1981

[48] Miron, S., Song, Y., Brie, D.*, et al.*: 'Multilinear direction finding for sensor-array with multiple scales of invariance', *IEEE Trans. Aerosp. Electron. Syst.*, 2015, **51**, (3), pp. 2057–2070

[49] Guo, X., Miron, S., Brie, D.*, et al.*: 'A CANDECOMP/PARAFAC perspective on uniqueness of DOA estimation using a vector sensor array', *IEEE Trans. Signal Process.*, 2011, **59**, (7), pp. 3475–3481

[50] Sørensen, M., De Lathauwer, L.: 'Multiple invariance ESPRIT for nonuniform linear arrays: a coupled canonical polyadic decomposition approach', *IEEE Trans. Signal Process.*, 2016, **64**, (14), pp. 3693–3704

[51] Liang, J., Yang, S., Zhang, J.*, et al.*: '4D near-field source localization using cumulant', *EURASIP J. Adv. Signal Process.*, 2007, **2007**, pp. 1–10

[52] Haardt, M., Roemer, F., Del.Galdo, G.: 'Higher-order SVD-based subspace estimation to improve the parameter estimation accuracy in multidimensional harmonic retrieval problems', *IEEE Trans. Signal Process.*, 2008, **56**, (7), pp. 3198–3213

[53] Boizard, M., Ginolhac, G., Pascal, F.*, et al.*: 'Numerical performance of a tensorMUSIC algorithm based on HOSVD for a mixture of polarized sources'. 2013 21st European Signal Processing Conf. (EUSIPCO), Marrakech Morocco, 2013

[54] Sidiropoulos, N.D.: 'Generalizing carathéodory's uniqueness of harmonic parameterization to n dimensions', *IEEE Trans. Inf. Theory*, 2001, **47**, pp. 1687–1690

[55] Li, Y., Razavilar, J., Liu, K.J.R.: 'A high-resolution technique for multidimensional NMR spectroscopie', *IEEE Trans. Biomed. Eng.*, 1998, **45**, pp. 78–86

[56] Nion, D., Sidiropoulos, N.D.: 'Tensor algebra and multidimensional harmonic retrieval in signal processing for MIMO radars', *IEEE Trans. Signal Process.*, 2010, **58**, pp. 5693–57057

[57] Zoltowski, M.D., Haardt, M., Mathews, C.P.: 'Closed-form 2D angle estimation with rectangular arrays in element space or beamspace via unitary ESPRIT', *IEEE Trans. Signal Process.*, 1996, **44**, pp. 316–328

[58] Acar, E., Bro, R., Smilde, A.K.: 'Data fusion in metabolomics using coupled matrix and tensor factorizations', *Proc. IEEE*, 2015, **103**, pp. 1602–1620

[59] Farias, R.C., Cohen, J.E., Comon, P.: 'Exploring multimodal data fusion through joint decompositions with flexible couplings', *IEEE Trans. Signal Process.*, 2016, **64**, pp. 4830–4844

[60] Lahat, D., Adali, T., Jutten, C.: 'Multimodal data fusion: an overview of methods, challenges and prospects', *Proc. IEEE*, 2015, **103**, pp. 1449–1477

[61] Cichocki, A.: 'Era of big data processing: A new approach via tensor networks and tensor decompositions'. Int. Workshop on Smart Info-Media Systems in Asia, Nagoya, Japan, 2013

[62] Phan, A., Cichocki, A., Uschmajew, A.*, et al.*: 'Tensor networks for latent variable analysis. Part i: algorithms for tensor train decomposition', *IEEE Trans. Neural Netw. Learn. Syst.*, 2020, **31**, (11), pp. 4622–4636

[63] Vervliet, N., Debals, O., Sorber, L.*, et al.*: 'Breaking the curse of dimensionality using decompositions of incomplete tensors: tensor-based scientific computing in big data analysis', *SIAM J. Sci. Comput.*, 2014, **31**, pp. 71–79

[64] Hackbusch, W., Kuhn, S.: 'A new scheme for the tensor representation', *J. Fourier Anal. Appl.*, 2009, **15**, pp. 706–722

[65] Kressner, D., Steinlechner, M., Vandereycken, B.: 'Low-rank tensor completion by Riemannian optimization', *BIT Numerical Math.*, 2014, **54**, pp. 447–468

[66] Bousse, M., Debals, O., De Lathauwer, L.: 'A tensor-based method for large-scale blind source separation using segmentation', *IEEE Trans. Signal Process.*, 2016, **65**, pp. 346–358

[67] Roemer, F., Haardt, M.: 'A closed-form solution for parallel factor (PARAFAC) analysis'. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Las Vegas, USA, 2008

[68] Bro, R.: 'PARAFAC: tutorial and applications', *Chemometr. Intell. Lab. Syst.*, 1997, **38**, pp. 149–171

[69] Li, N., Kindermann, S., Navasca, C.: 'Some convergence results on the regularized alternating least-squares method for tensor decomposition', *Linear Algebr. Appl.*, 2013, **438**, pp. 796–812

[70] Cichocki, A. N., Lee, I.V.O., Phan, A.H.*, et al.*: 'Low-rank tensor networks for dimensionality reduction and large-scale optimization problems: perspectives and challenges', *Foundations and Trends in Mach. Learn.*, 2016, **9**, pp. 249–420

[71] Kroonenberg, P.M., de Leeuw, J.: 'Principal component analysis of three-mode data by means of alternating least squares algorithms', *Psychometrika*, 1980, **45**, pp. 69–97

[72] Clark, M.P., Scharf, L.L.: 'Two-dimensional modal analysis based on maximum likelihood', *IEEE Trans. Signal Process.*, 1994, **42**, pp. 1443–1452

[73] Boyer, R.: 'Deterministic asymptotic Cramér-Rao bound for the multidimensional harmonic model', *Signal Process.*, 2008, **88**, pp. 2869–2877

[74] Sahnoun, S., Usevich, K., Comon, P.: 'Multidimensional ESPRIT for damped and undamped signals: algorithm, computations and perturbation analysis', *IEEE Trans. Signal. Proc.*, 2017, **65**, (22), pp. 5897–5910. hal-01360438

[75] Liu, J., Liu, X.: 'An eigenvector-based approach for multidimensional frequency estimation with improved identifiability', *IEEE Trans. Signal Process.*, 2006, **54**, pp. 4543–4556

[76] Sorensen, M., De Lathauwer, L.: 'Blind signal separation via tensor decomposition with vandermonde factor: canonical polyadic decomposition', *IEEE Trans. Signal Process.*, 2013, **61**, pp. 5507–5519

[77] Bro, R., Sidiropoulos, N.D., Giannakis, G.B.: 'A fast least squares algorithm for separating trilinear mixtures'. Int. Workshop on Independent Component Analysis and Blind Separationg, Aussois, France, Jan. 11–15, 1999

[78] Papy, J.M., De Lathauwer, L., Huffel, S.V.: 'Exponential data fitting using multilinear algebra: the single-channel and multi-channel case', *Wiley Online Library*, 2005, **12**, pp. 809–826

[79] Goulart, J.H., Boizard, M., Boyer, R.*, et al.*: 'Tensor CP decomposition with structured factor matrices: algorithms and performance', *IEEE. J. Sel. Top. Signal. Process.*, 2016, **10**, (4), pp. 757–769

[80] Markovsky, I.: '*Low rank approximation: algorithms, implementation, applications*' (Springer Science & Business Media, 2011)

[81] Boyle, J.P., Dykstra, R.L.: 'A method for finding projections onto the intersection of convex sets in hilbert spaces'. *Advances in order restricted statistical inference*, Springer, New York, NY, USA, 1986, pp. 28–47

[82] Rife, D.C., Boorstyn, R.R.: 'Single tone parameter estimation from discrete-time observations', *IEEE Trans. Inf. Theory*, 1974, **20**, (5), pp. 591–598

[83] Boyer, R., Comon, P.: 'Rectified ALS algorithm for multidimensional harmonic retrieval'. Sensor Array and Multichannel Signal Processing Workshop (SAM), Rio de Janeiro, Brazil, 2016

[84] Sørensen, M., De Lathauwer, L.: 'New uniqueness conditions for the canonical polyadic decomposition of third-order tensors', *SIAM J.Matrix Anal. Appl.*, 2015, **36**, pp. 1381–1403

[85] Zheng, L., Tse, D.N.C.: 'Diversity and multiplexing: A fundamental tradeoff in multiple-antenna channels', *IEEE Trans. Signal Process.*, 2003, **49**, (5), pp. 1073–1096

[86] Favier, G., de Almeida, A.: 'Tensor space-time-frequency coding with semiblind receivers for MIMO wireless communication systems', *IEEE Trans. Signal Process.*, 2014, **62**, (22), pp. 5987–6002

[87] Liu, K., da Costa, J.P.C., So, H.*, et al.*: 'Semi-blind receivers for joint symbol and channel estimation in space-time-frequency MIMO-OFDM systems', *IEEE Trans. Signal Process.*, 2013, **61**, (21), pp. 5444–5457

[88] de Almeida, A., Favier, G., Mota, J.C.: 'A constrained factor decomposition with application to MIMO antenna systems', *IEEE Trans. Signal Process.*, 2008, **56**, (6), pp. 2429–2442

[89] de Almeida, A., Favier, G., Mota, J.C.: 'PARAFAC-based unified tensor modeling for wireless communication systems with application to blind multiuser equalization', *Signal Process.*, 2007, **87**, (2), pp. 337–351

[90] da Costa, M.N., Favier, G., Romano, J.M.T.: 'Tensor modelling of MIMO communication systems with performance analysis and kronecker receivers', *Elsevier Signal Process.*, 2018, **145**, (4), pp. 304–316

[91] Rong, Y., Khandaker, M.R.A., Xiang, Y.: 'Channel estimation of dual-hop MIMO relay system via parallel factor analysis', *IEEE Trans. Wirel. Commun.*, 2012, **11**, (6), pp. 2224–2233

[92] Cavalcante, I.V., de Almeida, A.L.F., Haardt, M.: 'Tensor-based approach to channel estimation in amplify-and-forward MIMO relaying systems'. Proc. IEEE 8th Sensor Array and Multichannel Signal Processing Workshop (SAM 2014), Coruna, Spain, 2014, pp. 445–448

[93] Roemer, F., Haardt, M.: 'Tensor-based channel estimation and iterative refinements for two-way relaying with multiple antennas and spatial reuse', *IEEE Trans. Signal Process.*, 2010, **58**, (11), pp. 5720–5735

[94] Fernandes, C.A.R., de Almeida, A.L.F., da Costa, D.B.: 'Unified tensor modeling for blind receivers in multiuser uplink cooperative systems', *IEEE Signal Process. Lett.*, 2012, **19**, (5), pp. 247–250

[95] de Almeida, A.L.F., Fernandes, C.A.R., da Costa, D.B.: 'Multiuser detection for uplink DS-CDMA amplify-and-forward relaying systems', *IEEE Signal Process. Lett.*, 2013, **20**, (7), pp. 697–700

[96] Ximenes, L.R., Favier, G., de Almeida, A.L.F.: 'Semi-blind receivers for non-regenerative cooperative MIMO communications based on nested PARAFAC modeling', *IEEE Trans. Signal Process.*, 2015, **63**, (18), pp. 4985–4998

[97] Sokal, B., de Almeida, A.L.F., Haardt, M.: 'Semi-blind receivers for MIMO multi-relaying systems via rank-one tensor approximations', *Signal Process.*, 2020, **166**, p. 107254

[98] Rocha, D.S., Fernandes, C., Favier, G.: 'MIMO multi-relay systems with tensor space-time coding based on coupled nested tucker decomposition', *Digit. Signal Process.*, 2019, **89**, (3), pp. 170–185

[99] Sidiropoulos, N.D., Budampati, R.: 'Khatri-Rao space-time codes', *IEEE Trans. Signal Process.*, 2002, **50**, (10), pp. 2396–2407