International Journal of Sciences and Techniques of Automatic control & computer engineering
IJ-STA, Volume 3, N° 1, July 2009, pp. 840–869.

Tensor-based methods for system identification
Part 1: Tensor tools

Gérard Favier[1] and Alain Y. Kibangou[2]

[1] Laboratoire I3S, University of Nice Sophia Antipolis, CNRS, Les Algorithmes- Bât. Euclide B, 2000 Route des lucioles, B.P. 121 - 06903 Sophia Antipolis Cedex, France
favier@i3s.unice.fr
[2] LAAS-CNRS, University of Toulouse, 7 avenue Colonel Roche, 31077 Toulouse, France
akibango@laas.fr

Abstract. This paper first aims at introducing the main notions relative to tensors, also called multi-way arrays, with a particular emphasis on tensor models, uniqueness properties, tensor matricization (also known as unfolding) and the alternating least squares (ALS) algorithm that is the most used method to estimate the matrix components of a tensor. The interest for tensors in signal processing will be illustrated by means of some examples (tensors of statistics, Volterra kernels, tensors of signals). In a second part, we will show how a tensor approach allows to solve the parameter estimation problem for both linear and nonlinear models, by means of deterministic or stochastic methods.

Key words: Tensor models; System identification; Blind identification; Volterra systems; Wiener-Hammerstein systems.

1 Brief historic review and motivations

The word *tensor* was introduced in 1846 by William Rowan Hamilton, irish mathematician famous for his discovery of quaternions and very well known in the automatic control community for his theorem (Cayley-Hamilton's theorem) stating that any square matrix cancels its characteristic polynomial. The tensor notion and the tensor analysis began to play an important role in Physics with the introduction of the theory of relativity by Albert Einstein, around 1915, then in mechanics for representing the constraint and deformation state of a volume subject to forces by means of a stress tensor and a deformation tensor respectively. Tensor decompositions were introduced by Hitchcock in 1927 [1], then

This paper was recommended for publication in revised form by the editor Staff.
Edition: CPU of Tunis, Tunisia, ISSN: 1737-7749

developed and applied in psychometrics by Tucker [2] in 1966, and Carroll and Chang [3] and Harshman [4] in 1970, who introduced the most common tensor models, respectively called Tucker, CANDECOMP and PARAFAC models. Such tensor models became very popular tools in chemometrics in the nineties [5].

Tensors, also called multi-way arrays, are very useful for modeling and interpreting multidimensional data. Applications of tensor tools can now be found in many scientific areas, including image processing, computer vision, numerical analysis, data mining, neuroscience and many others.

Tensors first appeared in signal processing (SP) applications in the context of the blind source separation problem solved in using cumulant tensors [6, 7]. Indeed, moments and cumulants of random variables or stochastic processes can be viewed as tensors [8], as will be illustrated later on by means of some examples. First SP applications of the PARAFAC model were made by Sidiropoulos and his co-workers in the context of sensor array processing [9] and wireless communication systems [10], in 2000. The motivation for using tensor tools in the SP community was first connected to the development of SP methods based on the use of high-order statistics (HOS). Another motivation follows from the multidimensional nature of signals, as it is the case, for instance, with the emitted and received signals in wireless communication systems [11]. Another illustration is provided by the kernels of a Volterra model.

The tensor algebra, also called multilinear algebra, constitutes a generalization of linear algebra. Whereas this last one is based on the concepts of vectors and vector spaces, the multilinear algebra relies upon the notions of tensors and tensor spaces. As matrices arise for representing linear applications, the tensors are associated with the representation of multilinear applications.

Nowadays, there exist several mathematical approaches for defining a tensor. In a simplified way, a tensor can be viewed as a mathematical object described by means of a finite number of indices, this number being called the tensor order, and satisfying the multilinearity property. So, a tensor $\mathbb{H} \in \mathscr{K}^{I_1 \times I_2 \times \cdots \times I_n}$, of order n, with elements in the field $\mathscr{K} = \Re$ or $\mathscr{C}$, depending on whether the tensor elements are real-valued or complex-valued, and dimensions $(I_1, I_2, \cdots, I_n)$, will have for entries $h_{i_1 i_2 \cdots i_n}$ with $i_j = 1, 2, \cdots, I_j$, $j = 1, 2, \cdots, n$. Each index i_j is as-

sociated with a coordinate axis, also called a mode or a way. The tensor $\mathbb{H}$ is characterized by $\prod_{j=1}^{n} I_j$ elements, also called coefficients or scalar components.

Particular cases

- A scalar h is a tensor of order zero (without index).
- A vector $\mathbf{h} \in \mathscr{K}^I$ is a first-order tensor, with coefficients h_i, $i = 1, \cdots, I$, and a single index.
- A matrix $\mathbf{H} \in \mathscr{K}^{I \times J}$ is a second-order tensor, with coefficients h_{ij} and two indices.
- A third-order tensor, also called a three-way array, $\mathbb{H} \in \mathscr{K}^{I \times J \times K}$, of dimensions $I \times J \times K$, with entries h_{ijk}, can be viewed as a hyper rectangle, the indices i, j, and k being associated with the horizontal, lateral, and frontal coordinate axes (modes) respectively (see Fig. 1). Fig. 1 represents a third-order tensor with its decompositions in matrix slices (two-dimensional sections of the tensor, also called slabs) obtained by slicing the tensor along each mode, i.e. by fixing an index and varying the two other ones.

  The horizontal, lateral, and frontal slices of a third-order tensor $\mathbb{H}$ are denoted by $\mathbf{H}_{i..}$, $\mathbf{H}_{.j.}$, and $\mathbf{H}_{..k}$ respectively.

In the sequel, vectors and matrices are represented by boldface lowercase ($\mathbf{x}$) and boldface capital ($\mathbf{X}$) letters, respectively, whereas tensors are denoted by capital blackboard letters ($\mathbb{X}$).

Symmetric and diagonal tensors

A third-order tensor $\mathbb{X}$ is said to be square if every mode has the same dimension I, i.e. $\mathbb{X} \in \mathscr{K}^{I \times I \times I}$.

A square third-order tensor $\mathbb{X} \in \mathscr{K}^{I \times I \times I}$ is said to be symmetric if its elements do not change under any permutation of its indices, i.e.:

$$x_{ijk} = x_{ikj} = x_{jik} = x_{jki} = x_{kij} = x_{kji}\,,\ \forall\ i, j, k = 1, ..., I.$$

A third-order tensor $\mathbb{X}$ can be partially symmetric in two modes. For instance, $\mathbb{X}$ is symmetric in the modes i and j if its frontal slices $\mathbf{X}_{..k}$ are symmetric matrices.

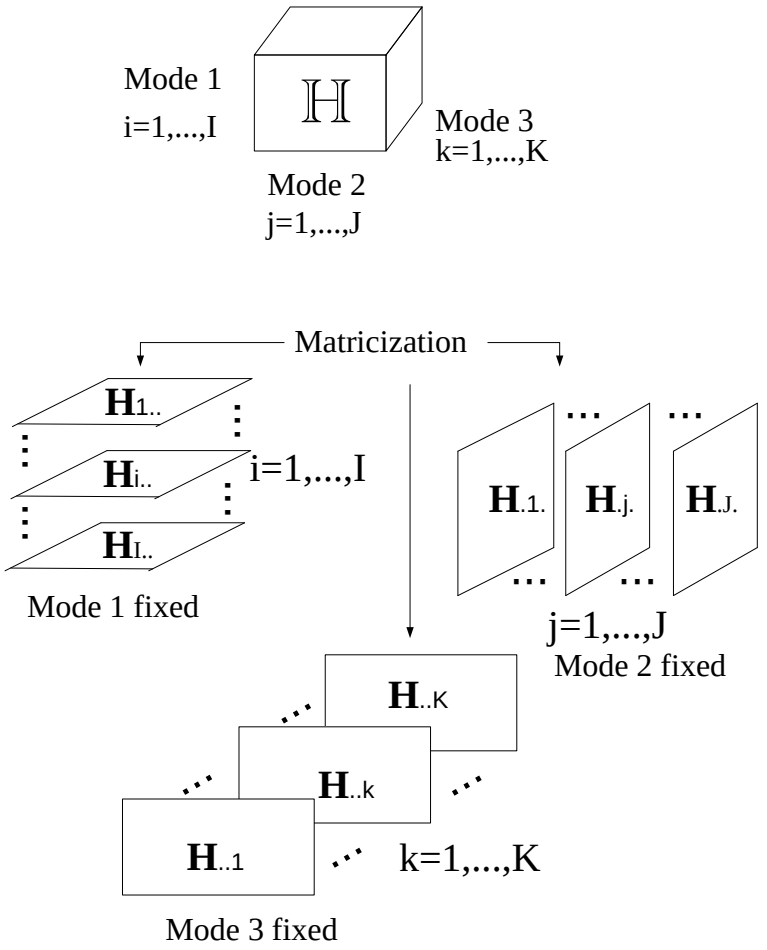

Fig. 1. Geometrical representation and slice decompositions of a third-order tensor.

A third-order tensor $\mathbb{X}$ is diagonal if $x_{ijk} \neq 0$ only if $i = j = k$. The diagonal tensor with ones on its superdiagonal is called the identity tensor.

The above definitions can be easily extended to tensors of any order $n > 3$.

2 Examples of tensors in SP

2.1 Tensors of statistics

Third-order cumulants of three real vector random variables

Let $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$ be three real-valued vector random variables, of respective dimensions M_1, M_2, and M_3. The cumulant $C_{\mathbf{x},\mathbf{y},\mathbf{z}} = cum(\mathbf{x},\mathbf{y},\mathbf{z})$ is a third-order tensor, with elements $[C_{\mathbf{x},\mathbf{y},\mathbf{z}}]_{m_1,m_2,m_3} = cum(x_{m_1}, y_{m_2}, z_{m_3})$. The modes 1, 2, and 3 correspond to the components of each vector $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$ respectively.

Fourth-order cumulants of a real scalar stochastic process

Let $x(k)$ be a discrete-time, stationary, real-valued, scalar stochastic process. Its fourth-order cumulants define a third-order tensor, with entries $[C_{4x}]_{\tau_1,\tau_2,\tau_3} = cum(x(k), x(k-\tau_1), x(k-\tau_2), x(k-\tau_3))$. The modes 1, 2, and 3 are associated with the time lags τ_1, τ_2, and τ_3.

Spatio-temporal covariances of a real-valued vector stochastic process

Let $\mathbf{x}(t) \in \Re^M$ be the vector of signals received on an M sensor array, at the time instant t. The spatio-temporal covariance matrices $C_{\mathbf{xx}}(k) = E\left[\mathbf{x}(t+k)\mathbf{x}^T(t)\right]$, $k = 1, \cdots, K$, define a third-order $M \times M \times K$ tensor, with entries $[C_{\mathbf{xx}}]_{i,j,k} = E[x_i(t+k)x_j(t)]$. The two first modes i and j are spatial (sensor numbers) whereas the third mode k is temporal (covariance time lag).

Third-order space cumulants of a real-valued vector stochastic process

Let $\mathbf{x}(t) \in \Re^M$ be the vector of signals received on an M sensor array, at the time instant t. The third-order space cumulants $cum(\mathbf{x},\mathbf{x},\mathbf{x})$ define a third-order $M \times M \times M$ tensor $\mathbb{C}_{\mathbf{xxx}}$, with elements $[C_{\mathbf{xxx}}]_{i,j,k} = cum(x_i(t), x_j(t), x_k(t))$. The three modes i, j, and k are spatial, the signals being considered at the same time instant t.

2.2 Kernels of Volterra models

SISO Volterra models

A P-th order Volterra model for a single-input single-output (SISO) system, is given by:

$$y(n) = h_0 + \sum_{p=1}^{P} \sum_{m_1=1}^{M_p} \cdots \sum_{m_p=1}^{M_p} h_p(m_1, \cdots, m_p) \prod_{j=1}^{p} u(n-m_j)$$

where $u(n)$ and $y(n)$ denote respectively the input and output signals, P is the nonlinearity degree of the Volterra model, M_p is the memory of the pth-order homogeneous term, and $h_p(m_1, \cdots, m_p)$ is a coefficient of the pth order kernel. This kernel can be viewed as an pth order $M_p \times M_p \times \cdots \times M_p$ tensor.

MIMO Volterra models

In the case of a multi-input multi-output (MIMO) Volterra model, with $\mathbf{u}(n) \in \Re^{n_I}$ and $\mathbf{y}(n) \in \Re^{n_O}$, where n_I and n_O denote respectively the number of inputs and outputs, the input-output relationship, for $s = 1, ...n_o$, is given by:

$$y_s(n) = h_0^s + \sum_{p=1}^{P} \sum_{j_1=1}^{n_I} \cdots \sum_{j_p=1}^{n_I} \sum_{m_1=1}^{M_p} \cdots \sum_{m_p=1}^{M_p} h^s_{j_1,\cdots,j_p}(m_1, \cdots, m_p) \prod_{k=1}^{p} u_{j_k}(n-m_k)$$

where $h^s_{j_1,\cdots,j_p}(m_1, \cdots, m_p)$ is the kernel of the sth output, acting on the product of p delayed input signals (with time delays m_1, $\cdots$, m_p), these signals being possibly associated with different inputs (j_1, $\cdots$, j_p). This Volterra kernel is now of order $2p$ and dimensions $M_p \times \cdots \times M_p \times n_I \times \cdots \times n_I$ with p temporal modes corresponding to the delays m_1, $\cdots$, m_p and p space modes associated with inputs j_1, $\cdots$, j_p.

2.3 Tensors of signals

Signals received by a communication system oversampled at the receiver

The signals received on an M antennas array, with an oversampling rate equal to P, and over a time duration of N symbol periods, constitute a tensor $\mathbb{X} \in \mathscr{C}^{M \times N \times P}$, with entries $x_{m,n,p}$, $m = 1, \cdots, M$, $n = 1, \cdots, N$, $p = 1, \cdots, P$, the three modes being associated with an antenna number (m), a symbol period (n),

and an oversampling period (p).

Signals received by a CDMA (Code Division Multiple Access) system

At the transmitter, each symbol to be transmitted is spread by means of a spreading code of length J, with a period $T_c = T/J$, where T is the symbol period. So, each symbol is converted into J chips. A block of N signals received by M antennas can then be viewed as a third-order tensor $\mathbb{X} \in \mathscr{C}^{M\times N\times J}$, with entries $x_{m,n,j}$, $m = 1,\cdots,M$, $n = 1,\cdots,N$, $j = 1,\cdots,J$.

Signals received by an OFDM (Orthogonal Frequency Division Multiplexing) system

In an OFDM system, the symbol sequence to be transmitted is organized into blocks of F symbols in the frequency domain. In this case, the received signals constitute a third-order tensor $\mathbb{X} \in \mathscr{C}^{M\times N\times F}$, with entries $x_{m,n,f}$, $m = 1,\cdots,M$, $n = 1,\cdots,N$, $f = 1,\cdots,F$.

We can conclude that these three received signals have two modes in common (m and n), and differ from one another in the third mode (p,j, or f). An unified tensor model of PARAFAC type was put in evidence for these three communication systems [11].
For illustrating these tensors of received signals, let us consider the case of a CDMA system [10]. In presence of Q users and in absence of noise, the signal received by the m-th antenna ($m = 1,\cdots,M$), at the n-th symbol period ($n = 1,\cdots,N$), can be written as:

$$x_{m,n,j} = \sum_{q=1}^{Q} a_{mq} s_{nq} c_{jq} \tag{1}$$

where a_{mq} denotes the fading coefficient of the channel between the q-th user and the m-th receive antenna, s_{nq} is the symbol transmitted by the q-th user, at the n-th symbol period and using the j-th spreading code c_{jq}. The set of signals received by the M receive antennas, during N symbol periods and using J spreading codes per user, constitutes a third-order tensor $\mathbb{X} \in \mathscr{C}^{M\times N\times J}$. Equation (1) corresponds to a PARAFAC model (See Section 4.1).

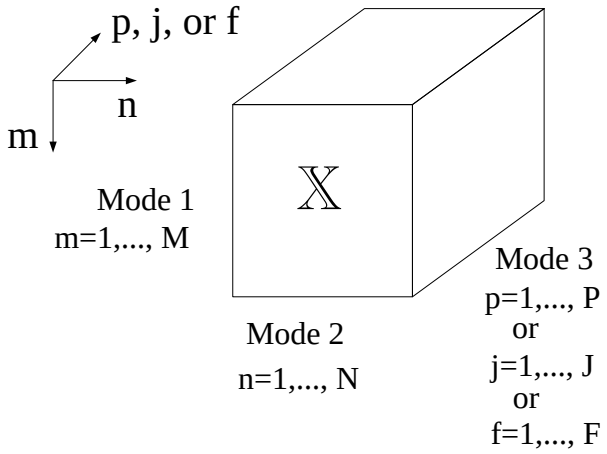

Fig. 2. Tensors of signals received by the three communication systems.

3 Some recalls on matrices and matrix decompositions

In this section, we recall some definitions relative to matrices and some matrix decompositions.

For a matrix $\mathbf{A} \in \mathscr{C}^{I \times J}$, we denote by $\mathbf{A}_{i.}$ and $\mathbf{A}_{.j}$ its i-th row vector and j-th column vector respectively.

3.1 Column-rank and row-rank

For a matrix $\mathbf{A} \in \Re^{I \times J}$, the number of linearly independent rows (columns) of $\mathbf{A}$ is called the row-rank (column-rank) of $\mathbf{A}$. The row-rank and the column-rank of a matrix are equal. This number, denoted by $r(\mathbf{A})$, is called the rank of $\mathbf{A}$. We have:

$$r(\mathbf{A}) \leq \min(I,J)$$

- $\mathbf{A}$ is said to be full rank if $r(\mathbf{A}) = \min(I,J)$ and rank-deficient if $r(\mathbf{A}) < \min(I,J)$.
- $\mathbf{A}$ is said to be full column-rank if $r(\mathbf{A}) = J$, i.e. if the J columns are independent.

– Analogously, $\mathbf{A}$ is said to be full row-rank if $r(\mathbf{A}) = I$, i.e. if the I rows are independent.

For a square matrix $\mathbf{A} \in \Re^{I \times I}$, if $r(\mathbf{A}) < I$, then $det(\mathbf{A}) = 0$ and $\mathbf{A}$ is called singular; if $\mathbf{A}$ is full-rank, then $det(\mathbf{A}) \neq 0$ and $\mathbf{A}$ is called nonsingular.

For $\mathbf{A} \in \Re^{I \times J}$, we have:

$$r(\mathbf{A}\mathbf{A}^T) = r(\mathbf{A}^T\mathbf{A}) = r(\mathbf{A})$$

For any full column-rank matrix $\mathbf{C}$ and full row-rank matrix $\mathbf{B}$, with appropriate dimensions, we have :

$$r(\mathbf{A}) = r(\mathbf{A}\mathbf{B}) = r(\mathbf{C}\mathbf{A}) = r(\mathbf{C}\mathbf{A}\mathbf{B}).$$

3.2 The rank theorem

For any matrix $\mathbf{A} \in \Re^{I \times J}$, we have:

$$dim\mathscr{R}(\mathbf{A}) + dim\mathscr{N}(\mathbf{A}) = J$$

where $\mathscr{R}(\mathbf{A}) = \{\mathbf{A}\mathbf{x}/\mathbf{x} \in \Re^J\}$ is the column space, also called the range space, of $\mathbf{A}$, and $\mathscr{N}(\mathbf{A}) = \{\mathbf{x}/\mathbf{A}\mathbf{x} = 0\}$ is the nullspace of $\mathbf{A}$, with $dim\mathscr{R}(\mathbf{A}) = r(\mathbf{A})$. From this theorem, we can conclude that:

– If $\mathbf{A}$ is full column-rank ($dim\mathscr{R}(\mathbf{A}) = J$), we have $dim\mathscr{N}(\mathbf{A}) = 0$ and consequently $\mathscr{N}(\mathbf{A}) = \{0\}$.
– If $\mathbf{A}$ is rank-deficient ($dim\mathscr{R}(\mathbf{A}) < J$), we have $dim\mathscr{N}(\mathbf{A}) > 0$, which implies $\mathscr{N}(\mathbf{A}) \neq \{0\}$.

3.3 k-rank (or Kruskal-rank)

The notion of k-rank was introduced by Kruskal [12] for studying the uniqueness of the PARAFAC decomposition of a given tensor.

Consider a matrix $\mathbf{A} \in \Re^{I \times J}$ of rank $r(\mathbf{A})$. The k-rank of $\mathbf{A}$ is the largest integer k such that any set of k columns of $\mathbf{A}$ is independent. It is denoted by $k_{\mathbf{A}}$. The k-rank is more constraining than the column-rank in the sense that $k_{\mathbf{A}} = k$ means that any set of k columns of $\mathbf{A}$ is independent, whereas $r(\mathbf{A}) = k$ implies that there exists at least one set of k independent columns and any set of $k+1$ columns is dependent. So, we have: $k_{\mathbf{A}} \leq r(\mathbf{A})$.

- If $\mathbf{A} \in \Re^{I \times J}$ is full column-rank, then $k_{\mathbf{A}} = J$.
- If $\mathbf{A} \in \Re^{I \times J}$ has no all-zero column, but contains at least two proportional columns, then $k_{\mathbf{A}} = 1$.
- If $\mathbf{A} \in \Re^{I \times J}$ has an all-zero column, then $k_{\mathbf{A}} = 0$.

Examples

$$\mathbf{A}_1 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \end{pmatrix} \Rightarrow\ r(\mathbf{A}_1) = 2,\ k_{\mathbf{A}_1} = 1$$

$$\mathbf{A}_2 = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{pmatrix} \Rightarrow\ r(\mathbf{A}_2) = k_{\mathbf{A}_2} = 2$$

$$\mathbf{A}_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \Rightarrow\ r(\mathbf{A}_3) = 2,\ k_{\mathbf{A}_3} = 0$$

3.4 The vectorization operator vec(.)

The vec-operator put the matrix $\mathbf{A} \in \mathscr{K}^{I \times J}$ in the form of a column vector of dimension IJ, denoted by $vec(\mathbf{A})$, by stacking all the column vectors of $\mathbf{A}$:

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}_{.1} & \cdots & \mathbf{A}_{.J} \end{pmatrix} \in \mathscr{K}^{I \times J} \Rightarrow vec(\mathbf{A}) = \begin{pmatrix} \mathbf{A}_{.1} \\ \vdots \\ \mathbf{A}_{.J} \end{pmatrix} \in \mathscr{K}^{IJ}$$

The matrix $\mathbf{A}$ is said to be vectorized, and we have:

$$\mathbf{v} = vec(\mathbf{A}) \Leftrightarrow v_{(i-1)J+j} = a_{ij}.$$

3.5 Kronecker and Khatri-Rao products

These matrix products are used for matricizing tensors.

The Kronecker product of two matrices $\mathbf{A} \in \mathscr{K}^{I \times J}$ and $\mathbf{B} \in \mathscr{K}^{K \times L}$, denoted by $\mathbf{A} \otimes \mathbf{B}$, is the partitioned matrix $\mathbf{C} \in \mathscr{K}^{IK \times JL}$ defined as:

$$\mathbf{C} = \mathbf{A} \otimes \mathbf{B} = \begin{pmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} & \cdots & a_{1J}\mathbf{B} \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} & \cdots & a_{2J}\mathbf{B} \\ \vdots & \vdots & \ddots & \vdots \\ a_{I1}\mathbf{B} & a_{I2}\mathbf{B} & \cdots & a_{IJ}\mathbf{B} \end{pmatrix}$$

The Khatri-Rao product of two matrices $\mathbf{A} \in \mathscr{K}^{I \times J}$ and $\mathbf{B} \in \mathscr{K}^{K \times J}$ having the same number of columns, is denoted by $\mathbf{A} \odot \mathbf{B}$ and defined as:

$$\mathbf{A} \odot \mathbf{B} = \left(\mathbf{A}_{.1} \otimes \mathbf{B}_{.1} \ \mathbf{A}_{.2} \otimes \mathbf{B}_{.2} \ \cdots \ \mathbf{A}_{.J} \otimes \mathbf{B}_{.J} \right) \tag{2}$$

This product is also called the column-wise Kronecker product. It can be written in the form of I row-blocks:

$$\mathbf{A} \odot \mathbf{B} = \begin{pmatrix} \mathbf{B} D_1^{\mathbf{A}} \\ \mathbf{B} D_2^{\mathbf{A}} \\ \vdots \\ \mathbf{B} D_I^{\mathbf{A}} \end{pmatrix} \tag{3}$$

where $D_i^{\mathbf{A}}$ is a diagonal matrix, with the elements of the i-th row of $\mathbf{A}$ on its main diagonal: $D_i^{\mathbf{A}} = diag\left(a_{i1} \ a_{i2} \ \cdots \ a_{iJ} \right)$.

We have the following properties: for $\mathbf{v} \in \mathscr{K}^J$, $\mathbf{A} \in \mathscr{K}^{I \times J}$, $\mathbf{B} \in \mathscr{K}^{J \times K}$, $\mathbf{C} \in \mathscr{K}^{K \times L}$,

$$vec(\mathbf{ABC}) = (\mathbf{C}^T \otimes \mathbf{A}) vec(\mathbf{B}) \tag{4}$$

$$vec(\mathbf{A} diag(\mathbf{v}) \mathbf{B}) = (\mathbf{B}^T \odot \mathbf{A}) \mathbf{v} \tag{5}$$

$$(\mathbf{A} diag(\mathbf{v})) \odot \mathbf{B}^T = \mathbf{A} \odot \left(\mathbf{B}^T diag(\mathbf{v}) \right) = \left(\mathbf{A} \odot \mathbf{B}^T \right) diag(\mathbf{v}) \tag{6}$$

where the operator diag(.) forms a diagonal matrix from its vector argument.

For $\mathbf{A} \in \mathscr{K}^{I \times R}$ and $\mathbf{B} \in \mathscr{K}^{J \times R}$, we have:

$$r(\mathbf{A} \odot \mathbf{B}) \leq r(\mathbf{A} \otimes \mathbf{B}) = r(\mathbf{A}) r(\mathbf{B})$$

and if neither $\mathbf{A}$ nor $\mathbf{B}$ contains a zero column [13]:

$$k_{\mathbf{A} \odot \mathbf{B}} \geq \min(k_{\mathbf{A}} + k_{\mathbf{B}} - 1, R)$$

As a consequence, we have [9]:

$\mathbf{A} \odot \mathbf{B}$ is full column-rank if $k_{\mathbf{A}} + k_{\mathbf{B}} \geq R + 1$.

We have to note that:

- This condition is satisfied, in particular, if one matrix is full column-rank R and the other one contains no zero column.
- This condition implies $I + J \geq R + 1$ and therefore $IJ \geq R$, i. e. $\mathbf{A} \odot \mathbf{B}$ is tall.

3.6 Moore-Penrose matrix pseudo-inverse

The Moore-Penrose pseudo-inverse of a matrix $\mathbf{A} \in \mathscr{K}^{I\times J}$ is a $J \times I$ matrix, denoted by $\mathbf{A}^{\dagger}$, that satisfies the four following conditions:

$$\mathbf{A}\mathbf{A}^{\dagger}\mathbf{A} = \mathbf{A}, \quad \mathbf{A}^{\dagger}\mathbf{A}\mathbf{A}^{\dagger} = \mathbf{A}^{\dagger}, \quad (\mathbf{A}\mathbf{A}^{\dagger})^T = \mathbf{A}\mathbf{A}^{\dagger}, \quad (\mathbf{A}^{\dagger}\mathbf{A})^T = \mathbf{A}^{\dagger}\mathbf{A}.$$

The Moore-Penrose pseudo-inverse always exists and is unique. See its expression in terms of the SVD in the next section.

If $\mathbf{A} \in \mathscr{C}^{I\times J}$ is full column rank, its Moore-Penrose pseudo-inverse is given by:

$$\mathbf{A}^{\dagger} = (\mathbf{A}^H\mathbf{A})^{-1}\mathbf{A}^H.$$

The calculation of $\mathbf{A}^{\dagger}$ needs to inverse a $J \times J$ matrix.

If $\mathbf{A}$ is a nonsingular matrix, then $\mathbf{A}^{\dagger} = \mathbf{A}^{-1}$.

3.7 Principal component analysis (PCA)

Consider a matrix $\mathbf{A} \in \Re^{I\times J}$. If $r(\mathbf{A}) = R$, then there exists a set of R pairs of vectors $\left\{\mathbf{u}_r \in \Re^I, \mathbf{v}_r \in \Re^J;\ r = 1,\cdots,R\right\}$ such that:

$$\mathbf{A} = \sum_{r=1}^{R} \mathbf{u}_r\mathbf{v}_r^T \tag{7}$$

or

$$\mathbf{A} = \sum_{r=1}^{R} \mathbf{u}_r \circ \mathbf{v}_r$$

where the symbol $\circ$ denotes the vector outer product defined as follows:

$$\mathbf{u}_r \in \Re^I,\ \mathbf{v}_r \in \Re^J \Rightarrow \mathbf{u}_r \circ \mathbf{v}_r \in \Re^{I\times J} \Leftrightarrow (\mathbf{u}_r \circ \mathbf{v}_r)_{ij} = u_{ir}v_{jr}.$$

The scalar writing of this matrix decomposition is the following :

$$a_{ij} = \sum_{r=1}^{R} u_{ir}v_{jr}, \quad i = 1,\cdots,I,\ j = 1,\cdots,J. \tag{8}$$

This writing corresponds to a bilinear decomposition of $\mathbf{A}$ by means of R components, each term $\mathbf{u}_r\mathbf{v}_r^T$, called a dyad, being a rank-one matrix. That means

that any matrix of rank R can be written as a sum of R rank-one matrices. The decomposition (7) is called a PCA model of $\mathbf{A}$. It can also be written as:

$$\mathbf{A} = \mathbf{U}\mathbf{V}^T = \begin{pmatrix}\mathbf{u}_1 & \cdots & \mathbf{u}_R\end{pmatrix}\begin{pmatrix}\mathbf{v}_1 & \cdots & \mathbf{v}_R\end{pmatrix}^T, \text{ with } \mathbf{U} \in \Re^{I\times R},\ \mathbf{V} \in \Re^{J\times R}. \quad (9)$$

The set of vectors $\{\mathbf{u}_r, \mathbf{v}_r;\ r = 1, \cdots, R\}$ is not unique. Indeed, for any nonsingular matrix $\mathbf{\Lambda}$, the matrices $\bar{\mathbf{U}} = \mathbf{U}\mathbf{\Lambda}$ and $\bar{\mathbf{V}} = \mathbf{V}\mathbf{\Lambda}^{-T}$ are such that:

$$\bar{\mathbf{U}}\bar{\mathbf{V}}^T = \mathbf{U}\mathbf{\Lambda}\left(\mathbf{V}\mathbf{\Lambda}^{-T}\right)^T = \mathbf{U}\mathbf{V}^T.$$

The only uniqueness case (up to a scaling factor) occurs when $\mathbf{A}$ is of rank one. Uniqueness can be obtained in imposing structural constraints on the matrix factors U and V (orthogonality, Toeplitz, Vandermonde, constant modulus, finite alphabet).

Link with the singular value decomposition (SVD)

Given matrix $\mathbf{A} \in \Re^{I\times J}$ with rank R, its reduced SVD is:

$$\mathbf{A} = \mathbf{U}_R\mathbf{S}_R\mathbf{V}_R^T,$$

where $\mathbf{U}_R = \begin{pmatrix}\mathbf{u}_1 & \cdots & \mathbf{u}_R\end{pmatrix} \in \Re^{I\times R}$ and $\mathbf{V}_R = \begin{pmatrix}\mathbf{v}_1 & \cdots & \mathbf{v}_R\end{pmatrix} \in \Re^{J\times R}$ are column-orthonormal matrices and $\mathbf{S}_R \in \Re^{R\times R}$ is diagonal:

$$\mathbf{U}_R^T\mathbf{U}_R = \mathbf{V}_R^T\mathbf{V}_R = \mathbf{I}_R \quad ; \quad \mathbf{S}_R = diag\begin{pmatrix}\sigma_1 & \cdots & \sigma_R\end{pmatrix}$$

where the $\sigma_i > 0$, $i = 1, \cdots, R$, are the nonzero singular values of $\mathbf{A}$, that are the positive square roots of the nonzero eigenvalues of $\mathbf{A}^T\mathbf{A}$ (and also of $\mathbf{A}\mathbf{A}^T$), the columns $\mathbf{u}_i$ of $\mathbf{U}_R$ are the eigenvectors of $\mathbf{A}\mathbf{A}^T$, also called the left singular vectors of $\mathbf{A}$, whereas the columns $\mathbf{v}_i$ of $\mathbf{V}_R$ are the eigenvectors of $\mathbf{A}^T\mathbf{A}$, also called the right singular vectors of $\mathbf{A}$. Equivalently, we have:

$$\mathbf{A} = \sum_{r=1}^{R} \sigma_r \mathbf{u}_r \mathbf{v}_r^T, \quad (10)$$

which shows that the SVD decomposes $\mathbf{A}$ into R rank-1 matrices. Consequently, the rank of $\mathbf{A}$ is the smallest number R of rank-1 matrices whose the sum equals $\mathbf{A}$.
By defining $\mathbf{U} = \mathbf{U}_R\mathbf{S}_R$ and $\mathbf{V} = \mathbf{V}_R$, we find again the PCA decomposition (9) with the column orthogonality constraint on the factors.

The Moore-Penrose pseudo-inverse of $\mathbf{A}$ is given by:

$$\mathbf{A}^\dagger = \mathbf{V}_R\mathbf{S}_R^{-1}\mathbf{U}_R^T.$$

3.8 LS solutions of a set of linear equations

The least squares (LS) solutions of the set of linear equations $\mathbf{A}\mathbf{x} = \mathbf{y}$, with $\mathbf{x} \in \Re^J$, $\mathbf{y} \in \Re^I$, $\mathbf{A} \in \Re^{I \times J}$, are such that $\|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2$ is minimized with respect to $\mathbf{x}$. The LS solutions, denoted by $\hat{\mathbf{x}}_{LS}$, are solutions of the normal equations:

$$\mathbf{A}^T\mathbf{A}\hat{\mathbf{x}}_{LS} = \mathbf{A}^T\mathbf{y}.$$

It is important to mention that, even if the set of equations $\mathbf{A}\mathbf{x} = \mathbf{y}$ is inconsistent, i.e. it admits no exact solutions, which corresponds to $\mathbf{y} \notin \mathscr{R}(\mathbf{A})$, the set of normal equations $\mathbf{A}^T\mathbf{A}\mathbf{x} = \mathbf{A}^T\mathbf{y}$ is always consistent, i.e. it always admits at least one solution [14]. Indeed, $\mathbf{A}^T\mathbf{y} \in \mathscr{R}(\mathbf{A}^T) = \mathscr{R}(\mathbf{A}^T\mathbf{A})$ implies that the second member of the normal equations belongs to the column space of its coefficient matrix ($\mathbf{A}^T\mathbf{A}$), which is the condition to satisfy for ensuring the consistency of a system of linear equations.

The LS solution is unique if and only if $\mathbf{A}$ is full column-rank. In this case, we have:

$$\hat{\mathbf{x}}_{LS} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{y} = \mathbf{A}^\dagger\mathbf{y} \tag{11}$$

When $\mathbf{A}$ is rank-deficient, there exists an infinite number of LS solutions and the general solution of the normal equations is given by:

$$\hat{\mathbf{x}}_{LS} = \mathbf{A}^\dagger\mathbf{y} + (\mathbf{I} - \mathbf{A}^\dagger\mathbf{A})\mathbf{z} \tag{12}$$

where $\mathbf{z}$ is any vector of $\Re^J$. The second term of this sum belongs to the nullspace of $\mathbf{A}$, which means $\mathscr{R}(\mathbf{I} - \mathbf{A}^\dagger\mathbf{A}) = \mathscr{N}(\mathbf{A})$. Indeed, we have:

$$\mathbf{A}(\mathbf{I} - \mathbf{A}^\dagger\mathbf{A})\mathbf{z} = (\mathbf{A} - \mathbf{A}\mathbf{A}^\dagger\mathbf{A})\mathbf{z} = 0$$

and consequently:

$$\mathbf{A}^T\mathbf{A}\hat{\mathbf{x}}_{LS} = \mathbf{A}^T\mathbf{A}\left(\mathbf{A}^\dagger\mathbf{y} + \left(\mathbf{I} - \mathbf{A}^\dagger\mathbf{A}\right)\mathbf{z}\right) = \mathbf{A}^T\mathbf{A}\mathbf{A}^\dagger\mathbf{y} = \mathbf{A}^T\mathbf{A}\mathbf{A}^\dagger\mathbf{A}\mathbf{x} = \mathbf{A}^T\mathbf{A}\mathbf{x} = \mathbf{A}^T\mathbf{y}$$

which shows that $\hat{\mathbf{x}}_{LS}$ defined in (12) satisfies the normal equations.

We have to note that $\hat{\mathbf{x}}_{LS} = \mathbf{A}^\dagger\mathbf{y}$ is the LS solution of minimal Euclidean norm.

3.9 Low-rank approximation of a matrix using its SVD[15]

Let us consider a matrix $\mathbf{A} \in \Re^{I \times J}$ of rank R, admitting $\mathbf{U}_R\mathbf{S}_R\mathbf{V}_R^T$ as reduced SVD, with the singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_R > \sigma_{R+1} = \cdots = \sigma_{\min(I,J)} = 0$

and denote by $\|\mathbf{A}\|_F^2$ its Frobenius norm[3]. Its best rank-K approximation ($K \leq R$), in the sense $\min_{\mathbf{B}/r(\mathbf{B})=K} \|\mathbf{A}-\mathbf{B}\|_F^2$, is given by the sum of the K rank-one matrices of its SVD (10) associated with the K largest singular values:

$$\mathbf{A}_K = \sum_{k=1}^{K} \sigma_k \mathbf{u}_k \mathbf{v}_k^T$$

and we have $\|\mathbf{A}-\mathbf{A}_K\|_F^2 = \sum_{k=K+1}^{R} \sigma_k^2$.

This property is not valid for higher-order tensors.

4 Main tensor models

It is important to notice that, in SP applications, tensor models are generally deduced from the underlying structure of the system under study and not from algebraic transforms applied to the data tensors. So, in the sequel, we will use the terms tensor model (also called multiway model or multilinear model) instead of tensor decomposition.

We first present the most well-known and used tensor model, the so called PARAFAC (Parallel Factor) model. Then, we will briefly describe several other tensor models (Tucker, PARATUCK2, CONFAC).

4.1 The PARAFAC model

The following decomposition of the third-order tensor $\mathbb{X} \in \Re^{I \times J \times K}$:

$$x_{ijk} = \sum_{r=1}^{R} a_{ir} b_{jr} c_{kr}, \quad i = 1, \cdots, I,\ j = 1, \cdots, J,\ k = 1, \cdots, K, \qquad (13)$$

is called the PARAFAC decomposition of $\mathbb{X}$. Equation (13) is a generalization of equation (8) to the third-order. It is a trilinear decomposition with R components that can be also written as:

$$\mathbb{X} = \sum_{r=1}^{R} \mathbf{A}_{.r} \circ \mathbf{B}_{.r} \circ \mathbf{C}_{.r}, \text{ with } \mathbf{A} \in \Re^{I \times R},\ \mathbf{B} \in \Re^{J \times R},\ \mathbf{C} \in \Re^{K \times R}. \qquad (14)$$

[3] The Frobenius norm of $\mathbf{A} \in \Re^{I \times J}$ is defined as: $\|\mathbf{A}\|_F^2 = \sum_{i,j} a_{ij}^2 = trace(A^T A)$.

Each term of the sum, called a triad, is a third-order rank-one tensor, i.e. the outer product of three vectors, also called an indecomposable tensor. So, PARAFAC corresponds to a decomposition of the tensor into a sum of R rank-one tensors. This is illustrated in Fig.3 .

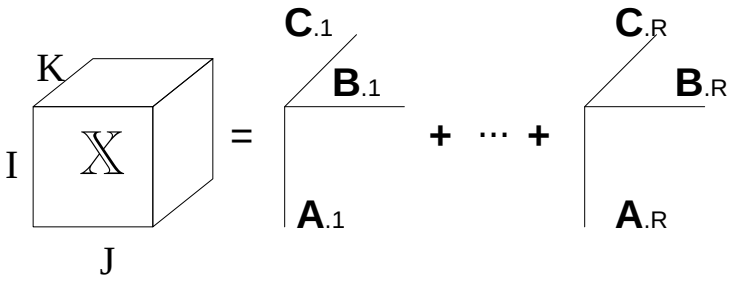

Fig. 3. PARAFAC decomposition of a third-order tensor.

The column vectors $\mathbf{A}_{.r}$, $\mathbf{B}_{.r}$, and $\mathbf{C}_{.r}$ are of respective dimensions I, J, and K. They contain the coefficients a_{ir}, b_{jr}, and c_{kr} of the three matrix factors (also called loadings or components) $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ of the PARAFAC decomposition.

The PARAFAC model can be easily extended to any order $N > 3$:

$$x_{i_1 i_2 \ldots i_N} = \sum_{r=1}^{R} \prod_{n} a_{i_n r}^{(n)}, \; i_n = 1, \ldots, I_n, \; n = 1, \ldots N, \tag{15}$$

or equivalently

$$\mathbb{X} = \sum_{r=1}^{R} \mathbf{A}_{.r}^{(1)} \circ \mathbf{A}_{.r}^{(2)} \circ \cdots \circ \mathbf{A}_{.r}^{(N)}, \text{ with } \mathbf{A}^{(n)} \in \mathscr{C}^{I_n \times R}, \; n = 1, \cdots, N.$$

In certain applications, the matrix factors are used to interpret the underlying information contained in the data tensors. In other applications, these factors contain the information of interest on the system under study.

So, in wireless communication applications, tensors of transmitted and received signals are decomposed into matrix factors that contain both the structure (design) parameters and the information of interest on the communication system.

For instance, with future multiuser MIMO systems where a base station will simultaneously transmit data via multiple transmit antennas to several users equipped with multiple receive antennas, tensor modeling can be used for block space-time spreading, i.e. each data stream is spread across both space (antennas) and time (symbols) dimensions [16]. In this case, the design parameters corresponding to the tensor dimensions, are the number of transmit/receive antennas, the temporal spreading factor (number of spreading codes), the number of data streams transmitted per block, and the number of symbol periods, whereas the information of interest to be recovered concerns the channel, the transmitted symbols and eventually the spreading codes.

In the case of tensor-based blind identification methods for FIR linear systems, like the ones described in the second part of the paper, that are based on the use of fourth-order output cumulants, the cumulant tensor admits a PARAFAC model that is directly obtained from the transformation of the output signals into fourth-order cumulants.

In the same way, with the tensor-based approach for Wiener-Hammerstein nonlinear system identification, the tensor corresponding to the third-order Volterra kernel associated with the Wiener-Hammerstein system to be identified, admits a PARAFAC model that directly results from the underlying structure of the original nonlinear system.

It is important to notice that the PARAFAC model captures the trilinear structure in the data, but this model is linear in its parameters, i.e. the coefficients of each matrix factor. This property is exploited in the ALS algorithm for estimating the PARAFAC parameters (see section 4.4).

In practice, a tensor of data is always represented as the sum of two terms : a structural term corresponding to the tensor model and a residual term containing both the measurement noise and the modeling error, which means that (13) is in general replaced by:

$$x_{ijk} = \sum_{r=1}^{R} a_{ir} b_{jr} c_{kr} + e_{ijk}, \quad i =, \cdots, I,\ j = 1, \cdots, J,\ k = 1, \cdots, K,$$

where e_{ijk} denotes the components of the residual tensor $\mathbb{E}$. An analysis of these residuals allows to quantify the fit of the tensor model with the data. The sum of squares of these residuals ($\|\mathbb{E}\|_F^2 = \sum_{i,j,k} e_{ijk}^2$) is often used as an objective

function for estimating the PARAFAC parameters, and as a test for stoping the ALS algorithm.

Tensor rank

The rank of a third-order tensor $\mathbb{X} \in \Re^{I \times J \times K}$ is the smallest number R of trilinear factors in its PARAFAC decomposition, i.e. the smallest number R such that (13) is exactly satisfied.
Other definitions exist for the rank of a tensor: symmetric rank, maximum rank, typical rank, generic rank.

Any symmetric tensor $\mathbb{X} \in \mathscr{C}^{I \times I \times \cdots \times I}$ of order N can be decomposed as a sum of R symmetric rank-1 tensors [17]:

$$\mathbb{X} = \sum_{r=1}^{R} \mathbf{A}_{.r} \circ \mathbf{A}_{.r} \circ \cdots \circ \mathbf{A}_{.r}, \tag{16}$$

with the matrix factor $\mathbf{A} \in \mathscr{C}^{I \times R}$.

The symmetric rank (over $\mathscr{C}$) is defined as the minimum number R of symmetric rank-1 factors in the decomposition (16).

The maximum rank is defined as the largest achievable rank for a given set of tensors. For instance, the set of all third-order tensors $\mathbb{X} \in \Re^{I \times J \times K}$ has a maximum rank, denoted by $R_{max}(I,J,K)$, such that [18]:

$$\max(I,J,K) \leq R_{max}(I,J,K) \leq \min(IJ,IK,JK).$$

Typical and generic ranks of tensors are more abstract concepts that are outside the scope of this paper (See [19]).

Remarks

- Unlike matrices, the rank of a tensor can be larger than its largest dimension, i.e. R can be larger than $\max(I,J,K)$.
- Kruskal [18] showed that the maximum possible rank of a third-order $2 \times 2 \times 2$ tensor is equal to 3.
- The maximum possible rank of a third-order $3 \times 3 \times 3$ tensor is equal to 5.
- From a practical point of view, the rank of a tensor is difficult to determine.

Uniqueness properties

Uniqueness is a key property of the PARAFAC decomposition. Four kinds of uniqueness can be defined [20]:

- *Strict uniqueness*: The matrix factors can be obtained without any ambiguity.
- *Quasi-strict uniqueness*: The matrix factors can be obtained up to column scaling.
- *Essential uniqueness*: The matrix factors can be obtained up to column permutation and scaling.
- *Partial uniqueness*: One or two matrix factors are essentially unique, while the other(s) matrix factor(s) is (are) not.

Consider the third-order tensor $\mathbb{X} \in \Re^{I \times J \times K}$. The PARAFAC decomposition of $\mathbb{X}$ is said to be essentially unique if two sets of matrix factors ($\mathbf{A}$,$\mathbf{B}$,$\mathbf{C}$) and ($\bar{\mathbf{A}}$, $\bar{\mathbf{B}}$, $\bar{\mathbf{C}}$) of the PARAFAC decomposition are linked by the following relations:

$$\bar{\mathbf{A}} = \mathbf{A}\boldsymbol{\Pi}\boldsymbol{\Delta}_1, \quad \bar{\mathbf{B}} = \mathbf{B}\boldsymbol{\Pi}\boldsymbol{\Delta}_2, \quad \bar{\mathbf{C}} = \mathbf{C}\boldsymbol{\Pi}\boldsymbol{\Delta}_3,$$

where $\boldsymbol{\Pi}$ is a permutation matrix, and $\boldsymbol{\Delta}_i = diag\left(d_{11}^i \ \cdots \ d_{RR}^i\right)$, $i = 1,2,3$, are diagonal matrices such that:

$$\boldsymbol{\Delta}_1\boldsymbol{\Delta}_2\boldsymbol{\Delta}_3 = \mathbf{I}_R, \text{ i.e. } d_{rr}^1 d_{rr}^2 d_{rr}^3 = 1, \ r = 1, \cdots, R.$$

These permutation and scaling ambiguities are evident from the writings (13) or (14). The permutation matrix $\boldsymbol{\Pi}$ is associated with a reordering of the rank-one component tensors, whereas the diagonal matrices $\boldsymbol{\Delta}_i$, $i = 1,2,3$, correspond to a scaling of the columns $\mathbf{A}_{.r}$, $\mathbf{B}_{.r}$, and $\mathbf{C}_{.r}$ by the diagonal elements d_{rr}^1 , d_{rr}^2, and d_{rr}^3 respectively, so that the PARAFAC decomposition can be rewritten as:

$$\mathbb{X} = \sum_{r=1}^{R} (d_{rr}^1 \mathbf{A}_{.r}) \circ (d_{rr}^2 \mathbf{B}_{.r}) \circ (d_{rr}^3 \mathbf{C}_{.r}).$$

The first uniqueness results are due to Harshman [4, 21]. The most general sufficient condition for essential uniqueness is due to Kruskal [12, 18] and depends on the concept of k-rank.

Kruskal's theorem

A sufficient condition for essential uniqueness of the PARAFAC decomposition (13), with (**A**, **B**, **C**) as matrix factors, is:

$$k_{\mathbf{A}} + k_{\mathbf{B}} + k_{\mathbf{C}} \geq 2R + 2. \tag{17}$$

Remarks

- Condition (17) does not hold when $R = 1$. Uniqueness was proved by Harshman [21] in this particular case.
- Condition (17) is sufficient but not necessary for essential uniqueness. However, it has recently been proved in [22] that (17) is also a necessary condition for $R = 2$ and $R = 3$.
- Other sufficient uniqueness conditions for the case where one of the matrix factors is full column rank are given in [23].
- The Kruskal's condition was extended to complex-valued tensors in [10] and to N-way tensors (N>3) in [24]. The PARAFAC model (15) of order N, with matrix factors $\mathbf{A}^{(n)}$, $n = 1, \cdots, N$, is essentially unique if:

$$\sum_{n} k_{\mathbf{A}^{(n)}} \geq 2R + N - 1.$$

4.2 Other third-order tensor models

Tucker models [2]

The Tucker model decomposes a tensor into a core tensor multiplied by a matrix along each mode. For a third-order tensor $\mathbb{X} \in \Re^{I \times J \times K}$, we have:

$$x_{ijk} = \sum_{p=1}^{P} \sum_{q=1}^{Q} \sum_{r=1}^{R} g_{pqr} a_{ip} b_{jq} c_{kr},\ i = 1, \cdots, I,\ j = 1, \cdots, J,\ k = 1, \cdots, K. \tag{18}$$

g_{pqr} being an element of the third-order core tensor $\mathbb{G} \in \Re^{P \times Q \times R}$.
The Tucker model can also be written as :

$$\mathbb{X} = \sum_{p=1}^{P} \sum_{q=1}^{Q} \sum_{r=1}^{R} g_{pqr} \mathbf{A}_{.p} \circ \mathbf{B}_{.q} \circ \mathbf{C}_{.r}$$

showing that $\mathbb{X}$ is decomposed into a weighted sum of PQR outer products, each weight corresponding to an element of the core tensor. This kind of tensor

model is more flexible due to the fact that it allows interactions between the different matrix factors, the weights g_{pqr} representing the amplitudes of these interactions.
When P, Q , and R are smaller than I, J, and K, the core tensor $\mathbb{G}$ can be viewed as a compressed version of $\mathbb{X}$.

The Tucker model can be viewed as the generalization of SVD to higher-order tensors, and a Tucker model with orthogonal matrix factors is called an higher-order SVD (HOSVD) [25]. For an Nth-order tensor, the computation of its HOSVD leads to the calculation of N matrix SVDs of unfolded matrices.

It is important to notice that the PARAFAC model with orthogonal matrix factors occurs only if the tensor is diagonalizable, but, in general, tensors cannot be diagonalized [26].

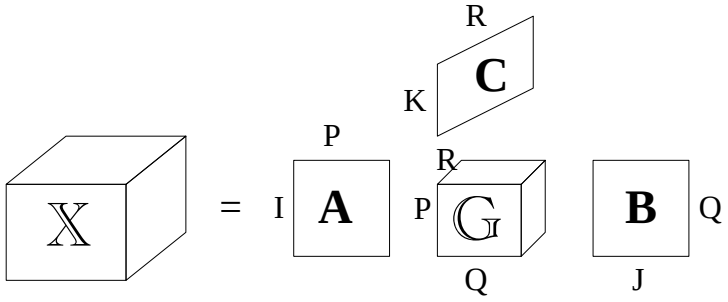

Fig. 4. Representation of the Tucker model for a third-order tensor.

The PARAFAC model is a special Tucker model corresponding to the case where $P = Q = R$ and the core tensor is diagonal ($g_{pqr} = 1$ if $p = q = r$ and $= 0$ otherwise).

The Tucker model can be easily extended to any order $N > 3$:

$$x_{i_1,\cdots,i_N} = \sum_{r_1=1}^{R_1} \cdots \sum_{r_N=1}^{R_N} g_{r_1,\cdots,r_N} \prod_{j=1}^{N} a_{i_j r_j}, \quad i_j = 1,\cdots,I_j, \quad j = 1,\cdots,N.$$

From equation (18), it is possible to deduce two simplified Tucker models:

Tucker 2 model

$$x_{ijk} = \sum_{p=1}^{P} \sum_{q=1}^{Q} h_{pqk} a_{ip} b_{jq},\ i = 1, \cdots, I,\ j = 1, \cdots, J,\ k = 1, \cdots, K. \quad (19)$$

Comparing (19) with (18), we can conclude that a Tucker2 model is a Tucker model with one matrix factor equals to the identity matrix ($\mathbf{C} = \mathbf{I}_K$) and $R = K$.

Tucker 1 model

$$x_{ijk} = \sum_{p=1}^{P} h_{pjk} a_{ip},\ i = 1, \cdots, I,\ j = 1, \cdots, J,\ k = 1, \cdots, K. \quad (20)$$

Tucker1 model is a Tucker model with $\mathbf{B} = \mathbf{I}_J$, $\mathbf{C} = \mathbf{I}_K$, $J = Q$ and $R = K$. In this case, two of the matrix factors are equal to the identity matrix.

It is important to notice that, unlike PARAFAC, Tucker models are not unique in the sense that each matrix factor can be determined only up to a rotation matrix. Indeed, if a matrix factor is multiplied by a rotation matrix, application of the inverse of this rotation matrix to the core tensor gives the same tensor model.

PARATUCK2 model

The PARATUCK2 model of a third-order tensor $\mathbb{X} \in \Re^{I \times J \times K}$is given by [27]:

$$x_{ijk} = \sum_{p=1}^{P} \sum_{q=1}^{Q} a_{ip} c^{A}_{kp} g_{pq} c^{B}_{kq} b_{jq},\ i = 1, \cdots, I,\ j = 1, \cdots, J,\ k = 1, \cdots, K. \quad (21)$$

This model allows interactions between the columns of the matrix factors $\mathbf{A}$ and $\mathbf{B}$, along the third-mode, through the interaction matrices $\mathbf{C}^A \in \Re^{K \times P}$ and $\mathbf{C}^B \in \Re^{K \times Q}$, the matrix $\mathbf{G} \in \Re^{P \times Q}$ defining the amplitude of these interactions.

The PARATUCK2 model was recently used for blind joint identification and equalization of Wiener-Hammerstein communication channels [28] and for space-time spreading-multiplexing in the context of MIMO wireless communication systems [29].

CONFAC model

For a third-order tensor $\mathbb{X} \in \Re^{I \times J \times K}$, the CONFAC model is given by [30]

$$x_{ijk} = \sum_{n=1}^{N} \sum_{p=1}^{P} \sum_{q=1}^{Q} \sum_{r=1}^{R} g_{pqr}^{(n)} a_{ip} b_{jq} c_{kr},\ i = 1, \cdots, I,\ j = 1, \cdots, J,\ k = 1, \cdots, K. \tag{22}$$

with $N \geq \max(P,Q,R)$ and $g_{pqr}^{(n)} = \psi_{pn} \phi_{qn} \omega_{rn}$.
The matrices $\boldsymbol{\Psi} \in \Re^{P \times N}$, $\boldsymbol{\Phi} \in \Re^{Q \times N}$, and $\boldsymbol{\Omega} \in \Re^{R \times N}$, with respective elements ψ_{pn}, ϕ_{qn}, and ω_{rn}, are called constraint matrices. They define the existence or not of interaction (coupling) between the different modes.

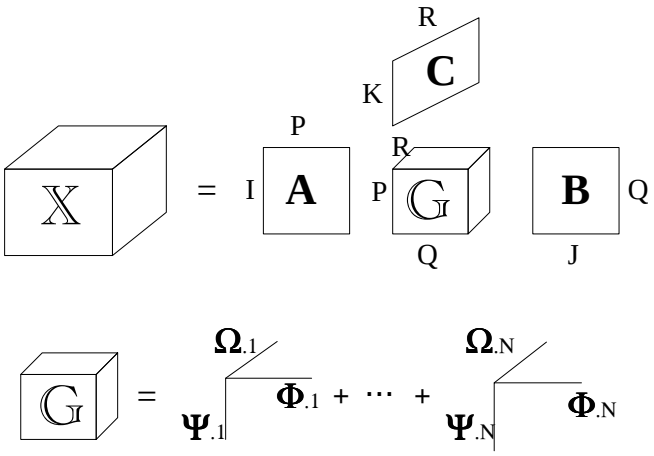

Fig. 5. Representation of the CONFAC model for a third-order tensor.

The constraint matrices are chosen such that:

- Their elements ψ_{pn}, ϕ_{qn}, and ω_{rn} are equal to 0 or 1, the value 1 implying the existence of an interaction whereas the value 0 implies no interaction.
- The columns of $\boldsymbol{\Psi}$, $\boldsymbol{\Phi}$, and $\boldsymbol{\Omega}$, are canonical basis vectors of the Euclidean spaces $\Re^P$, $\Re^Q$, and $\Re^R$ respectively.

– $\boldsymbol{\Psi}$, $\boldsymbol{\Phi}$, and $\boldsymbol{\Omega}$ are full rank, which implies that each basis vector is contained at least once in each constraint matrix.

The CONFAC model illustrated in Fig. 5, can be viewed as a constrained Tucker model the core tensor of which admits a PARAFAC decomposition. Indeed, equation (22) is identical to (18) with: $g_{pqr} = \sum_n g_{pqr}^{(n)} = \sum_n \psi_{pn}\phi_{qn}\omega_{rn}$, which corresponds to a PARAFAC model for the core tensor that depends on the constraint matrices $\boldsymbol{\Psi}$, $\boldsymbol{\Phi}$, and $\boldsymbol{\Omega}$.

The CONFAC model was used for designing new MIMO communication systems [30]. Another constrained tensor model that can be viewed as a simplified version of the CONFAC tensor model, is proposed in [31].

4.3 Matricization of a third-order tensor

The transformation that consists in putting a tensor of order larger than two under the form of a matrix is called matricization and the tensor is then said to be matricized.

Slice matrix representations

As illustrated in Fig. 1, a third-order $I \times J \times K$ tensor $\mathbb{H}$, with entries h_{ijk}, can be matricized in matrix slices along each mode. Each matrix slice, denoted by $\mathbf{H}_{i..}$, $\mathbf{H}_{.j.}$, and $\mathbf{H}_{..k}$, contains the elements of a horizontal, lateral and frontal slice, respectively. These matrices can be explicited as follows:

$$\mathbf{H}_{i..} = \begin{pmatrix} h_{i11} & h_{i12} & \cdots & h_{i1K} \\ h_{i21} & h_{i22} & \cdots & h_{i2K} \\ \vdots & \vdots & \ddots & \vdots \\ h_{iJ1} & h_{iJ2} & \cdots & h_{iJK} \end{pmatrix} \in \mathscr{K}^{J\times K}, \quad \mathbf{H}_{.j.} = \begin{pmatrix} h_{1j1} & h_{2j1} & \cdots & h_{Ij1} \\ h_{1j2} & h_{2j2} & \cdots & h_{Ij2} \\ \vdots & \vdots & \ddots & \vdots \\ h_{1jK} & h_{2jK} & \cdots & h_{IjK} \end{pmatrix} \in \mathscr{K}^{K\times I}$$

$$\mathbf{H}_{..k} = \begin{pmatrix} h_{11k} & h_{12k} & \cdots & h_{1Jk} \\ h_{21k} & h_{22k} & \cdots & h_{2Jk} \\ \vdots & \vdots & \ddots & \vdots \\ h_{I1k} & h_{I2k} & \cdots & h_{IJk} \end{pmatrix} \in \mathscr{K}^{I\times J}.$$

In the case of the PARAFAC model with matrix factors ($\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$), these matrix slices are given by:

$$\mathbf{H}_{i..} = \mathbf{B}D_i^{\mathbf{A}}\mathbf{C}^T, \quad \mathbf{H}_{.j.} = \mathbf{C}D_j^{\mathbf{B}}\mathbf{A}^T, \quad \mathbf{H}_{..k} = \mathbf{A}D_k^{\mathbf{C}}\mathbf{B}^T, \tag{23}$$

where $D_i^{\mathbf{A}}$, $D_j^{\mathbf{B}}$, $D_k^{\mathbf{C}}$ are the following diagonal matrices of dimension $R \times R$:

$$D_i^{\mathbf{A}} = diag\left(a_{i1}\ a_{i2}\ \cdots\ a_{iR}\right) = diag(\mathbf{A}_{i.})$$
$$D_j^{\mathbf{B}} = diag\left(b_{j1}\ b_{j2}\ \cdots\ b_{jR}\right) = diag(\mathbf{B}_{j.})$$
$$D_k^{\mathbf{C}} = diag\left(c_{k1}\ c_{k2}\ \cdots\ c_{kR}\right) = diag(\mathbf{C}_{k.})$$

Unfolded matrix representations

By stacking the matrices associated with a same type of slices, we get three different unfolded matrix representations of the tensor $\mathbb{H}$. These unfolded matrices, denoted by $\mathbf{H}_1$, $\mathbf{H}_2$, and $\mathbf{H}_3$, are defined as follows:

$$\mathbf{H}_1 = \begin{pmatrix} \mathbf{H}_{..1} \\ \vdots \\ \mathbf{H}_{..K} \end{pmatrix} \in \mathscr{K}^{IK \times J},\ \mathbf{H}_2 = \begin{pmatrix} \mathbf{H}_{1..} \\ \vdots \\ \mathbf{H}_{I..} \end{pmatrix} \in \mathscr{K}^{JI \times K},\ \mathbf{H}_3 = \begin{pmatrix} \mathbf{H}_{.1.} \\ \vdots \\ \mathbf{H}_{.J.} \end{pmatrix} \in \mathscr{K}^{KJ \times I}. \tag{24}$$

The three unfolded matrix representations of $\mathbb{H}$ are different from the point of view of the organization of their elements but they are equivalent in terms of information they contain since each one contains all the elements of the tensor.

In the case of the PARAFAC model, applying formula (3) to equations (23) and (24) leads to:

$$\mathbf{H}_1 = (\mathbf{C} \odot \mathbf{A})\mathbf{B}^T, \quad \mathbf{H}_2 = (\mathbf{A} \odot \mathbf{B})\mathbf{C}^T, \quad \mathbf{H}_3 = (\mathbf{B} \odot \mathbf{C})\mathbf{A}^T. \tag{25}$$

Another way for unfolding the tensor consists in columnwise stacking the matrix slices, which leads to the following unfolded matrix representations:

$$\mathbf{H}_1 = \left(\mathbf{H}_{..1} \cdots \mathbf{H}_{..K}\right) = \mathbf{A}(\mathbf{C} \odot \mathbf{B})^T \in \mathscr{K}^{I \times JK},$$
$$\mathbf{H}_2 = \left(\mathbf{H}_{1..} \cdots \mathbf{H}_{I..}\right) = \mathbf{B}(\mathbf{A} \odot \mathbf{C})^T \in \mathscr{K}^{J \times KI},$$
$$\mathbf{H}_3 = \left(\mathbf{H}_{.1.} \cdots \mathbf{H}_{.J.}\right) = \mathbf{C}(\mathbf{B} \odot \mathbf{A})^T \in \mathscr{K}^{K \times IJ}.$$

In the case of the CONFAC model with matrix factors $(\mathbf{A}, \mathbf{B}, \mathbf{C})$ and constraint matrices $(\boldsymbol{\Psi}, \boldsymbol{\Phi}, \boldsymbol{\Omega})$, we have:

$$\mathbf{H}_1 = (\mathbf{C}\boldsymbol{\Omega} \odot \mathbf{A}\boldsymbol{\Psi})\,(\mathbf{B}\boldsymbol{\Phi})^T,\ \mathbf{H}_2 = (\mathbf{A}\boldsymbol{\Psi} \odot \mathbf{B}\boldsymbol{\Phi})\,(\mathbf{C}\boldsymbol{\Omega})^T,\ \mathbf{H}_3 = (\mathbf{B}\boldsymbol{\Phi} \odot \mathbf{C}\boldsymbol{\Omega})\,(\mathbf{A}\boldsymbol{\Psi})^T. \tag{26}$$

Comparing (26) with (25), we can conclude that the CONFAC model can be viewed as a constrained PARAFAC model in the sense that the factors $(\mathbf{A},\mathbf{B},\mathbf{C})$ are replaced by the constrained factors $(\mathbf{A\Psi},\mathbf{B\Phi},\mathbf{C\Omega})$. Indeed, (22) can be rewritten as:

$$x_{ijk}=\sum_n\left(\sum_p a_{ip}\psi_{pn}\right)\left(\sum_q b_{jq}\phi_{qn}\right)\left(\sum_r c_{kr}\omega_{rn}\right)=\sum_n(\mathbf{A\Psi})_{in}(\mathbf{B\Phi})_{jn}(\mathbf{C\Omega})_{kn},$$

which is a PARAFAC model with matrix factors $(\mathbf{A\Psi},\mathbf{B\Phi},\mathbf{C\Omega})$.

4.4 Alternating Least Squares (ALS) algorithm

Identifying a PARAFAC model consists in estimating its matrix factors $(\mathbf{A},\mathbf{B},\mathbf{C})$ from the knowledge of the data tensor $\mathbb{H}$ or equivalently its unfolded matrix representations. This parameter estimation problem can be solved in the LS sense, i.e. in minimizing the Frobenius norm of the residual tensor $\mathbb{E}$, or equivalently the Frobenius norm of one of its unfolded matrices deduced from (25). So, for instance, the LS cost function to be minimized can be written as :

$$\min_{\mathbf{A},\mathbf{B},\mathbf{C}}\|\mathbf{E}_3\|_F^2=\min_{\mathbf{A},\mathbf{B},\mathbf{C}}\left\|\mathbf{H}_3-(\mathbf{B}\odot\mathbf{C})\mathbf{A}^T\right\|_F^2 \tag{27}$$

The ALS algorithm originally proposed in [4] and [21], consists in replacing the minimization of the LS criterion (27) by an alternating minimization of three conditional LS cost functions built from the three unfolded representations (25). Each cost function is minimized with respect to one matrix factor conditionally to the knowledge of the two other matrix factors, this knowledge being provided first by the initialization and then by the estimates obtained at the previous iteration :

$$\min_{\mathbf{A}}\left\|\mathbf{H}_3-(\mathbf{B}_{t-1}\odot\mathbf{C}_{t-1})\mathbf{A}^T\right\|_F^2\Rightarrow\mathbf{A}_t$$

$$\min_{\mathbf{B}}\left\|\mathbf{H}_1-(\mathbf{C}_{t-1}\odot\mathbf{A}_t)\mathbf{B}^T\right\|_F^2\Rightarrow\mathbf{B}_t$$

$$\min_{\mathbf{C}}\left\|\mathbf{H}_2-(\mathbf{A}_t\odot\mathbf{B}_t)\mathbf{C}^T\right\|_F^2\Rightarrow\mathbf{C}_t.$$

So, the initially trilinear LS problem that needs to use a nonlinear optimization method, is transformed into three linear LS problems that are successively solved by means of the standard LS solution. As we recalled in Section 3.8, these three LS problems admit an unique solution if anf only if $\mathbf{A}\odot\mathbf{B}$, $\mathbf{B}\odot\mathbf{C}$,

and $\mathbf{C} \odot \mathbf{A}$ are full column-rank, which directly leads to the following LS solution uniqueness condition [32]:

$$\min \{r(\mathbf{A} \odot \mathbf{B}), r(\mathbf{B} \odot \mathbf{C}), r(\mathbf{C} \odot \mathbf{A}\} = R.$$

We have to note that the sufficient Kruskal's condition (17) implies the above condition.

The ALS algorithm is summarized as follows:

1. Initialize $\mathbf{B}_0$ and $\mathbf{C}_0$ and set $t = 0$.
2. Increment t and compute:
   (a) $\mathbf{A}_t = \left((\mathbf{B}_{t-1} \odot \mathbf{C}_{t-1})^{\dagger} \mathbf{H}_3 \right)^T$.
   (b) $\mathbf{B}_t = \left((\mathbf{C}_{t-1} \odot \mathbf{A}_t)^{\dagger} \mathbf{H}_1 \right)^T$.
   (c) $\mathbf{C}_t = \left((\mathbf{A}_t \odot \mathbf{B}_t)^{\dagger} \mathbf{H}_2 \right)^T$.
3. Return to step 2 until convergence.

So, ALS is an iterative algorithm that successively estimates, at each iteration, one matrix factor in keeping the two other ones fixed to their previous estimated value. The computation loop is repeated until a convergence test is satisfied. This test is built either from the estimated parameters, or using a criterion based on the tensor reconstructed with the estimated parameters. It generally consists in detecting if an estimated parameter variation between two consecutive iterations or a model fit error becomes smaller than a predefined threshold.

The ALS algorithm can be easily extended to both higher-order PARAFAC models and to Tucker models with orthogonality constraints on the matrix factors.

The main advantage of the ALS algorithm is its simplicity. However, its convergence can be very slow and convergence towards the global minimum is not guaranteed. That is much depending on the initialization. Different solutions exist for improving the ALS algorithm in terms of speed of convergence and stability [33–35].

5 Conclusion

In this first part of the paper, the main definitions and properties relative to tensors have first been recalled before giving a brief review of the most used tensor

models. Some examples of tensors encountered in SP have been described for signal and system modeling.
Tensors constitute a very interesting research area, with very promising applications in various domains. However, fundamental properties of tensors in terms of rank and uniqueness remain to be better understood. New tensor models and tensor decompositions are to be studied, and more robust parameter estimation algorithms are necessary to render tensor-based SP methods even more efficient and attractive. Low-rank approximation of tensors of rank larger than one is also a very important open problem with the objective of tensor dimensionality reduction, which is very useful for tensor object feature extraction and classification. More generally, tensor simplification, like diagonalization for instance, by means of transforms, is also an open issue. Finally, other applications of tensor tools can be considered as, for instance, for multidimensional filtering with application to noise reduction in color images and seismic data [36] and also to channel equalization.
In the second part of this paper, we present three examples of tensor-based methods for system identification.

References

1. Hitchcock, F.: The expression of a tensor or a polyadic as a sum of products. J. Math. Phys. Camb. **6** (1927) 164–189
2. Tucker, L.: Some mathematical notes of three-mode factor analysis. Psychometrika **31**(3) (1966) 279–311
3. Caroll, J., Chang, J.: Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition. Psychometrika **35** (1970) 283–319
4. Harshman, R.: Foundation of the PARAFAC procedure: models and conditions for an "explanatory" multimodal factor analysis. UCLA working papers in phonetics **16** (1970) 1–84
5. Bro, R.: PARAFAC. Tutorial and applications. Chemometrics and Intelligent Laboratory Systems **38** (1997) 149–171
6. Cardoso, J.F.: Eigen-structure of the fourth-order cumulant tensor with application to the blind source separation problem. In: Proc. Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP), Albuquerque, New Mexico, USA (1990) 2655–2658
7. Cardoso, J.F.: Super-symmetric decomposition of the fourth-order cumulant tensor. Blind identification of more sources than sensors. In: Proc. Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP), Toronto, Ontario, Canada, (1991) 3109–3112

8. McCullagh, P.: Tensor methods in statistics. Chapman and Hall (1987)
9. Sidiropoulos, N., Bro, R., Giannakis, G.: Parallel factor analysis in sensor array processing. IEEE Trans. on Signal Processing **48**(8) (2000) 2377–2388
10. Sidiropoulos, N., Giannakis, G., Bro, R.: Blind PARAFAC receivers for DS-CDMA systems. IEEE Trans. on Signal Processing **48**(3) (2000) 810–823
11. de Almeida, A.L.F., Favier, G., Mota, J.C.M.: PARAFAC-based unified tensor modeling for wireless communication systems with application to blind multiuser equalization. Signal Processing **87**(2) (2007) 337–351
12. Kruskal, J.: Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics. Linear Algebra Applicat. **18** (1977) 95–138
13. Sidiropoulos, N., Liu, X.: Identifiability results for blind beamforming in incoherent multipath with small delay spread. IEEE Trans. on Signal Processing **49**(1) (2001) 228–236
14. Meyer, C.: Matrix analysis and applied linear algebra. SIAM (2000)
15. Eckart, C., Young, G.: The approximation of one matrix by another of lower rank. Psychometrika **1** (1936) 211–218
16. de Almeida, A.L.F., Favier, G., Mota, J.C.M.: Multiuser MIMO system using block space-time spreading and tensor modeling. Signal Processing **88**(10) (2008) 2388–2402
17. Comon, P., Golub, G., Lim, L.H., Mourrain, B.: Symmetric tensors and symmetric tensor ranks. SCCM technical report 06-02, Stanford University (2006)
18. Kruskal, J.: Rank, decomposition and uniqueness for 3-way and n-way arrays. In Coppi, R., Bolasco, S., eds.: Multiway data analysis. Elsevier, Amsterdam (1989) 8–18
19. Comon, P., ten Berge, J.: Generic and typical ranks of three-way arrays. Technical report, I3S (2006)
20. Kibangou, A., Favier, G.: Blind equalization of nonlinear channels using tensor decompositions with code/space/time diversities. Signal Processing **89**(2) (2009) 133–143
21. Harshman, R.: Determination and proof of minimum uniqueness conditions for PARAFAC 1. UCLA working papers in phonetics **22** (1972) 111–117
22. ten Berge, J., Sidiropoulos, N.: On uniqueness in CANDECOMP/PARAFAC. Psychometrika **67**(3) (2002) 399–409
23. Jiang, T., Sidiropoulos, N.: Kruskal's permutation lemma and the identification of CANDECOMP/PARAFAC and bilinear models with constant modulus constraints. IEEE Trans. on Signal Processing **52**(9) (2004) 2625–2636
24. Sidiropoulos, N., Bro, R.: On the uniqueness of multilinear decomposition of n-way arrays. Journal of Chemometrics **14** (2000) 229–239
25. De Lathauwer, L., De Moor, B., Vandevalle, J.: A multilinear singular value decomposition. SIAM J. Matrix Anal. Appl. **21** (2000) 1253–1278
26. Kolda, T.: Orthogonal tensor decompositions. SIAM J. Matrix Anal. Appl. **23**(1) (2001) 243–255

27. Harshman, R., Lundy, M.: Uniqueness proof for a family of models sharing features of Tucker's three-mode factor analysis and PARAFAC/CANDECOMP. Psychometrika **61**(1) (1996) 133–154
28. Kibangou, A., Favier, G.: Blind joint identification and equalization of Wiener-Hammerstein communication channels using PARATUCK-2 tensor decomposition. In: Proc. EUSIPCO, Poznan, Poland (2007)
29. de Almeida, A.L.F., Favier, G., Mota, J.C.M.: Space-time spreading-multiplexing for MIMO antenna systems using the PARATUCK2 tensor decomposition. In: Proc. EUSIPCO, Lausanne, Switzerland (2008)
30. de Almeida, A.L.F., Favier, G., Mota, J.C.M.: A constrained factor decomposition with application to MIMO antenna systems. IEEE Trans. on Signal Processing **56**(6) (2008) 2429–2442
31. de Almeida, A.L.F., Favier, G., Mota, J.C.M.: Constrained tensor modeling approach to blind multiple-antenna CDMA schemes. IEEE Trans. on Signal Processing **56**(6) (2008) 2417–2428
32. Liu, X., Sidiropoulos, N.: Cramer-Rao lower bounds for low-rank decomposition of multidimensional arrays. IEEE Trans. on Signal Processing **49**(9) (2001) 2074–2086
33. Bro, R.: Multi-way analysis in the food industry. Models, algorithms and applications. PhD thesis, University of Amsterdam, Netherlands (1998)
34. Tomasi, G., Bro, R.: A comparison of algorithms for fitting the PARAFAC model. Comp. Stat. Data Anal. **50**(7) (2006) 1700–1734
35. Rajih, M., Comon, P., Harshman, R.: Enhanced line search : a novel method to accelerate PARAFAC. (SIAM J. Matrix Anal. (To appear))
36. Muti, D., Bourennane, S.: Survey on tensor signal algebraic filtering. Signal Processing **87**(2) (2007) 237–249