Provided for non-commercial research and educational use only.
Not for reproduction or distribution or commercial use.



This article was originally published in a journal published by Elsevier, and the attached copy is provided by Elsevier for the author's benefit and for the benefit of the author's institution, for non-commercial research and educational use including without limitation use in instruction at your institution, sending it to specific colleagues that you know, and providing a copy to your institution's administrator.

All other uses, reproduction and distribution, including without limitation commercial reprints, selling or licensing copies or access, or posting on open internet sites, your personal or institution's website or repository, are prohibited. For exceptions, permission may be sought for such use through Elsevier's permissions site at:

http://www.elsevier.com/locate/permissionusematerial

Available online at www.sciencedirect.com
ScienceDirect
Signal Processing 87 (2007) 322–336
SIGNAL PROCESSING
www.elsevier.com/locate/sigpro

Tensor-based techniques for the blind separation of DS–CDMA signals

Lieven De Lathauwer*,[1], Joséphine Castaing[2]

ETIS, UMR 8051 (CNRS, ENSEA, UCP), 6, avenue du Ponceau, BP 44, F 95014 Cergy-Pontoise Cedex, France

Received 22 March 2005; accepted 19 December 2005
Available online 12 June 2006

Abstract

In this paper we present new deterministic tensor-based techniques for the blind separation of a mixture of DS–CDMA signals received by an antenna array. First, we show that the blind receiver follows from a simultaneous matrix decomposition. We present a new, relaxed, bound on the number of users that can be allowed at the same time. We further derive two algorithms that jointly exploit the CDMA structure and the constant modulus property of the transmitted signals.
© 2006 Elsevier B.V. All rights reserved.

Keywords: Code division multiple access; Signal separation; Blind technique; Higher-order tensor; Multi-linear algebra; Constant modulus

1. Introduction

This paper deals with the problem of multi-user separation in direct sequence–code division multiple access (DS–CDMA) systems. The techniques that will be presented are blind, i.e., no training sequences are required. The techniques are also deterministic, i.e., they do not require the estimation of statistics and they do not assume that the transmitted sequences are statistically independent. Instead, the algebraic structure of the data is exploited. As a consequence, the algorithms work for small sample sizes, or, equivalently, for channels that are fast varying.

It was shown in [1] that DS–CDMA data received by an antenna array can be arranged in a three-way array or third-order tensor that follows a so-called parallel factor (PARAFAC) model [2–4], where, under some conditions, each term consists of the signal transmitted by a different user (cf. below). If the PARAFAC of the data tensor is unique (up to trivial indeterminacies), then its computation separates the different users. Uniqueness is guaranteed if the number of terms (users) is below the so-called Kruskal bound (cf. below) [5,1]. The PARAFAC solution is usually computed by means of an alternating least squares (ALS) algorithm [1,4].

In [6] we developed a new approach to PARAFAC. It was shown that, under some conditions, the PARAFAC components can be obtained from a simultaneous matrix decomposition. Simultaneous

*Corresponding author. Tel.: +33 1 30 73 66 10; fax: +33 1 30 73 62 82.
E-mail addresses: delathau@ensea.fr (L. De Lathauwer), castaing@ensea.fr (J. Castaing).
[1]Lieven De Lathauwer holds a permanent research position with the CNRS, France; he also holds a honorary research position with the K.U.Leuven, Belgium.
[2]Joséphine Castaing is supported by a DGA/CNRS Ph.D. grant.

0165-1684/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.sigpro.2005.12.015

Author's personal copy

matrix decompositions have become popular tools in blind signal separation [7–12]. In this paper we apply the results of [6] to blind DS–CDMA signal separation. Apart from a new algorithm, this leads to a new bound on the number of users that is significantly more relaxed than the Kruskal bound. In a second part of the paper we additionally assume that the transmitted signals are constant modulus (CM). Exploiting this property leads to another set of simultaneous matrix decompositions, as explained in [13]. This set is coupled with the set that represents the PARAFAC structure constraint. We derive algorithms that take advantage of both constraints.

The paper is organized as follows. In Section 2 we recall the model that applies to DS–CDMA data received by an antenna array and we specify our working hypotheses. In Section 3 we follow the reasoning developed in [1] and look at the data model from a multi-linear algebraic perspective. We explain why PARAFAC can blindly extract the sequences transmitted by the different users. In Section 4 we show that the PARAFAC components can be computed from a simultaneous matrix decomposition and we present a new bound on the number of simultaneous users. Section 5 reviews two existing techniques for solving a set of simultaneous matrix decompositions. In Section 6 we jointly exploit the CDMA structure and the CM property by means of appropriate generalizations of the algorithms discussed in Section 5. Section 7 illustrates the performance of the new techniques by means of some simulations. Section 8 is the conclusion.

To conclude the introduction, let us introduce some notations. A calligraphic letter $\mathscr{Y}$ denotes a third-order tensor. A bold-face capital denotes a matrix ($\mathbf{Y}$). Vectors are written in italic capitals (U) and Y_k indicates the kth column of matrix $\mathbf{Y}$. Scalars are lower-case letters (a). The scalar a_i indicates the ith element of vector A, and the scalar a_{ij} denotes the element on the ith row and the jth column of matrix $\mathbf{A}$. Italic capitals are also used to denote index upper bounds ($i = 1, 2, \ldots, I$). The transpose, complex conjugate and complex conjugate transpose are denoted by $\cdot^{\mathrm{T}}$, $\cdot^{*}$, $\cdot^{\mathrm{H}}$, respectively. The norm $\|\cdot\|$ is the Frobenius norm. Furthermore, the operator vec($\cdot$) builds a vector from a matrix by stacking the columns of this matrix one above the other; more specifically, element a_{ij} of the $(I \times J)$ matrix $\mathbf{A}$ becomes the element at position $i + (j-1)I$ of the vector vec($\mathbf{A}$). Unvec($\cdot$) is the inverse operation of vec($\cdot$). The operator diag($\cdot$) stacks its vector argument in a diagonal matrix. The operator vecdiag($\cdot$) extracts the diagonal of its matrix argument and stacks it in a column vector. The $(N \times N)$ identity matrix is represented by $\mathbf{I}_N$. The symbol $\otimes$ denotes the Kronecker product,

$$\mathbf{A} \otimes \mathbf{H} \stackrel{\text{def}}{=} \begin{pmatrix} a_{11}\mathbf{H} & a_{12}\mathbf{H} & \ldots \\ a_{21}\mathbf{H} & a_{22}\mathbf{H} & \ldots \\ \vdots & \vdots & \end{pmatrix},$$

and $\odot$ represents the Khatri–Rao or column-wise Kronecker product:

$$\mathbf{A} \odot \mathbf{H} \stackrel{\text{def}}{=} (A_1 \otimes H_1 \;\; A_2 \otimes H_2 \;\; \ldots).$$

2. Signal model

The jth chip transmitted in the kth symbol period by user r, $j \in [1, J]$, $k \in [1, K]$, $r \in [1, R]$, is given by

$$x_{kjr} = s_{kr} c_{jr}, \tag{1}$$

in which s_{kr} is the kth symbol transmitted by user r and in which c_{jr} is the jth chip of user r's spreading code. We do not suppose that spreading codes are known nor that they are orthogonal. However, we assume that the spreading factor J is known or has been estimated. We also assume that the different user sequences have been synchronized at the symbol level. Let us first consider the case without inter-chip-interference (ICI), i.e., the mixture induced by the channel propagation can be modeled as instantaneous. Then the baseband output of antenna i, $i \in [1, I]$, for chip j and symbol k, can be written as the sum over all the users of the product of the fading factor between user r and antenna i (a_{ir}) and the signal value transmitted by user r (x_{kjr}):

$$y_{ijk} = \sum_{r=1}^{R} a_{ir} x_{kjr}, \tag{2}$$

$$= \sum_{r=1}^{R} a_{ir} s_{kr} c_{jr}, \tag{3}$$

in which R is the number of users. (For convenience, Eq. (3) does not contain a noise term. The same holds for Eqs. (4), (5), (12) and (45) below. Noise results in a perturbation of the equations.) When the propagation channel for each user can be modeled as a finite impulse response (FIR) filter, inter-symbol-interference (ISI) can be avoided by adopting a "guard chips" or "discard prefix" strategy. If moreover multi-path effects are in the far field, then

Author's personal copy

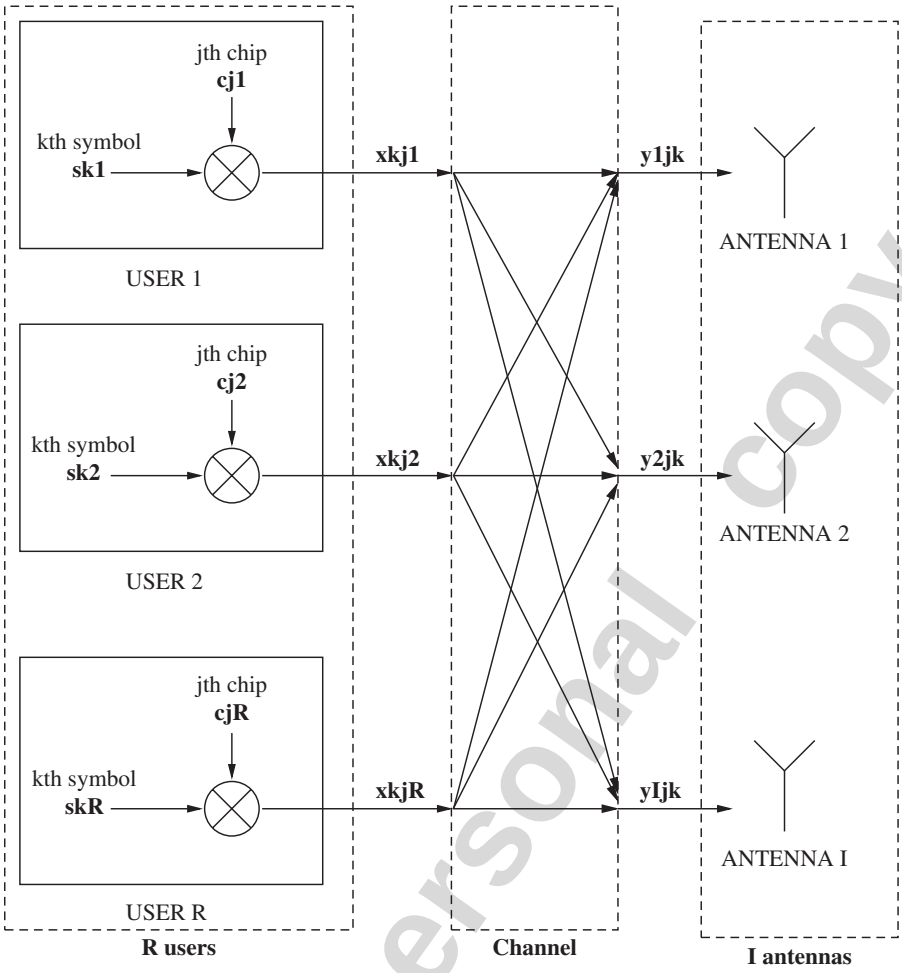


Fig. 1. Schematic representation of the CDMA system.

the antenna array still receives an instantaneous mixture of the (convolved) signals of the different users. Despite the presence of ICI in this case, the received data have a similar structure as before [1]. One just needs to replace in (3) c_{jr}, the jth element of user r's spreading code, by h_{jr}, the jth element of the convolution between user r's spreading code and the impulse response of the rth propagation channel:

$$y_{ijk} = \sum_{r=1}^{R} a_{ir} h_{jr} s_{kr}, \quad i \in [1, I], \ j \in [1, J], \ k \in [1, K]. \tag{4}$$

A schematic representation of the system is given in Fig. 1.

3. DS–CDMA signal separation by means of PARAFAC analysis

The element y_{ijk} can be seen as the element at position (i,j,k) of an $(I \times J \times K)$ tensor $\mathscr{Y}$. Let us call A_r, H_r, S_r the three vectors of size I, J, and K, containing, respectively, the fading coefficients a_{ir}, the channel coefficients h_{jr}, and the information symbols s_{kr} for user r. Eq. (4) can be formally written in a tensor format as

$$\mathscr{Y} = \sum_{r=1}^{R} A_r \circ H_r \circ S_r, \tag{5}$$

where $\circ$ denotes the outer product, defined by $(U \circ V)_{ij} = u_i v_j$. Tensors consisting of the outer product of a number of vectors, are called rank-1. (A matrix that equals the outer product of two vectors is also rank-1.) Eq. (5) thus shows that the data tensor $\mathscr{Y}$ consists of a sum of rank-1 terms, where each term is associated with a different user. The decomposition of a tensor in rank-1 terms is called PARAFAC model [2,3], or canonical decomposition (CANDECOMP) [14]. A didactical explanation of the different aspects of PARAFAC is given in [4].

The power of PARAFAC stems from its uniqueness properties. Recall that the decomposition of a

Authors personal copy

matrix in rank-1 terms is not unique at all. Usually one imposes orthogonality on the components in order to obtain a decomposition that is unique up to trivial indeterminacies. The singular value decomposition (SVD) and the symmetric eigenvalue decomposition (EVD) can be seen as ways to decompose a matrix in mutually orthogonal rank-1 terms. However, the orthogonality conditions may not be satisfied by the actual underlying components, so that these decompositions do not reveal the factors of interest.

Let us now explain why PARAFAC is a more interesting tool in this respect. We first note that the tensor $\mathscr{Y}$ in (5) stays the same if the triplet (A_r, H_r, S_r) is replaced by $(\alpha_r A_r, \beta_r H_r, \gamma_r S_r)$, with $\alpha_r\beta_r\gamma_r = 1$. Secondly, the order of the rank-1 terms is arbitrary. If the decomposition is unique up to these trivial indeterminacies, then it is called essentially unique. Next, let us introduce a variant of the rank of a matrix. The *Kruskal rank* of matrix $\mathbf{A}$, $k(\mathbf{A})$, is defined as the maximal number k such that *any* set of k columns of $\mathbf{A}$ is linearly independent [5]. This implies that the Kruskal rank of a matrix is always smaller than or equal to its rank. Using this variant of the rank, Kruskal was able to show that the PARAFAC is essentially unique if [5]

$$k(\mathbf{A}) + k(\mathbf{H}) + k(\mathbf{S}) \geqslant 2(R+1). \tag{6}$$

In this equation, $\mathbf{A} \in \mathbb{C}^{I\times R}$ (resp. $\mathbf{H} \in \mathbb{C}^{J\times R}$, $\mathbf{S} \in \mathbb{C}^{K\times R}$) are the matrices obtained by stacking the vectors A_r (resp. H_r, S_r) one after the other. The original proof by Kruskal only applied to real-valued tensors. A concise proof that also holds in the complex case, was given in [1]. The result was generalized to tensors of arbitrary order in [15]. Eq. (6) should be seen as a bound on R guaranteeing that decomposition (5), as opposed to matrix rank-1 expansions, is unique and can in principle be computed. No orthogonality constraints are involved. Bound (6) rather depends on the linear (in)dependence of the columns of $\mathbf{A}$, $\mathbf{H}$ and $\mathbf{S}$, as this affects the Kruskal rank.

If the entries of a matrix can be considered drawn from continuous distributions, then this matrix is full rank and full Kruskal rank with probability one. At least for $\mathbf{A}$ and $\mathbf{H}$ this is the case. The entries of $\mathbf{S}$ may belong to a finite alphabet. However, if this alphabet is sufficiently rich, then the probability that also $\mathbf{S}$ is full rank and full Kruskal rank, converges to one as the number of samples increases. If all three matrices are full Kruskal rank, then Eq. (6) reduces to

$$\min(I,R) + \min(J,R) + \min(K,R) \geqslant 2(R+1). \tag{7}$$

More specifically, if $R \leqslant K$, then the Kruskal bound implies that PARAFAC of $\mathscr{Y}$ reveals the sequences transmitted by

$$R \leqslant \min(I,R) + \min(J,R) - 2 \tag{8}$$

simultaneous users. If $R \geqslant \max(I,J)$, then the condition becomes $R \leqslant I + J - 2$.

PARAFAC is usually solved by means of an ALS algorithm [1,4]. This means that the least-squares cost function associated with (5), namely

$$f(\mathbf{A},\mathbf{H},\mathbf{S}) = \|\mathscr{Y} - \sum_{r=1}^{R} A_r \circ H_r \circ S_r\|^2, \tag{9}$$

$$\stackrel{\text{def}}{=} \sum_{ijk} \left| y_{ijk} - \sum_r a_{ir} s_{kr} c_{jr} \right|^2 \tag{10}$$

is minimized by means of alternating updates of one of its matrix arguments, keeping the other two matrices fixed. Because PARAFAC is a *multi-linear* decomposition, each update just amounts to solving a classical linear least-squares problem. Explicit expressions are given in Section 5.1. The convergence is monotonic because each update improves the fit in (9). Convergence may be local; to increase the probability that the global minimum is found, the algorithm may be reinitialized a couple of times. In general, the number of users R has to be estimated by trial-and-error. The direct application of the ALS algorithm to the data tensor $\mathscr{Y}$ will be called direct ALS (DALS) in this paper.

We conclude that it is in fact very natural to cast the DS–CDMA separation problem in a tensor framework. In this framework, PARAFAC is the appropriate tool to obtain the solution.

4. A new PARAFAC-based approach

In this paper we start from the weak assumption that

$$R \leqslant \min(IJ, K). \tag{11}$$

It turns out that in this case a bound on the number of users can be derived that is much weaker than Kruskal's. Moreover, the solution can be computed from a simultaneous matrix decomposition. For exact data, the solution follows from an EVD. The algebraic aspects of this new way to deal with PARAFAC are described in detail in [6]. The derivation builds upon results obtained in the

Authors personal copy

context of independent component analysis [16]. In this section we sketch the main line of reasoning, thereby showing how the set of matrices that have to be diagonalized, is derived from the data tensor $\mathscr{Y}$. For an in-depth discussion of the technique, the reader is referred to [6].

4.1. Reformulation of the problem

We first stack the entries of tensor $\mathscr{Y}$ in an $(IJ \times K)$ matrix $\mathbf{Y}$ as follows:

$$(\mathbf{Y})_{(i-1)J+j,k} = y_{ijk}, \quad i \in [1, I], \ j \in [1, J], \ k \in [1, K].$$

Eq. (5) can be written in a matrix format as

$$\mathbf{Y} = (\mathbf{A} \odot \mathbf{H})\mathbf{S}^{\mathrm{T}}. \tag{12}$$

Under condition (11) matrix $\mathbf{A} \odot \mathbf{H}$ is full column rank with probability one [6]. For the same reasons as in Section 3, we assume that also $\mathbf{S}$ is full column rank. This implies that the number of active users R is simply equal to the rank of $\mathbf{Y}$. Instead of determining it by trial-and-error, as in Section 3, it can be estimated as the number of significant singular values of $\mathbf{Y}$. Let the "economy size" SVD of $\mathbf{Y}$ be given by

$$\mathbf{Y} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{H}}, \tag{13}$$

in which $\mathbf{U} \in \mathbb{C}^{IJ \times R}$ and $\mathbf{V} \in \mathbb{C}^{K \times R}$ are column-wise orthonormal matrices and in which $\mathbf{\Sigma} \in \mathbb{C}^{R \times R}$ is positive diagonal.

We deduce from Eqs. (12) and (13) that there exists an a priori unknown matrix $\mathbf{F} \in \mathbb{C}^{R \times R}$ that satisfies

$$\mathbf{A} \odot \mathbf{H} = \mathbf{U}\mathbf{\Sigma}\mathbf{F}, \tag{14}$$

$$\mathbf{S}^{\mathrm{T}} = \mathbf{F}^{-1}\mathbf{V}^{\mathrm{H}}. \tag{15}$$

It is sufficient to estimate the matrix $\mathbf{F}$ to find the matrix of fading coefficients $\mathbf{A}$, the matrix of (convolved) spreading codes $\mathbf{H}$ and the matrix of symbols $\mathbf{S}$. Obviously, $\mathbf{S} = \mathbf{V}^{*}\mathbf{F}^{-\mathrm{T}}$. Furthermore, the columns N_r of $\mathbf{U}\mathbf{\Sigma}\mathbf{F} = \mathbf{A} \odot \mathbf{H}$ correspond to rank-1 $(I \times J)$ matrices $\mathbf{N}_r$:

$$\mathbf{N}_r \stackrel{\text{def}}{=} \mathrm{unvec}(N_r) = \mathrm{unvec}(A_r \otimes H_r) = H_r A_r^{\mathrm{T}}, \quad r \in [1, R].$$

This means that H_r can, up to an irrelevant scaling factor, be determined as the left singular vector associated with the largest singular value of $\mathbf{N}_r$ and that A_r corresponds to the complex conjugate of the associated right singular vector.

4.2. Estimation of $\mathbf{F}$

The aim is now to find a matrix $\mathbf{F}$ that satisfies (14) and to evaluate under which conditions such a matrix is essentially unique.

Let $\mathbf{E}_r$ be an $(I \times J)$ matrix in which the rth column of matrix $\mathbf{U}\mathbf{\Sigma}$ is stacked. We have

$$\begin{aligned}\mathbf{E}_r &= \mathrm{unvec}((\mathbf{U}\mathbf{\Sigma})_r) \\ &= \mathrm{unvec}(((\mathbf{A} \odot \mathbf{H})\mathbf{F}^{-1})_r) \\ &= \sum_{k=1}^{R} (H_k A_k^{\mathrm{T}})(\mathbf{F}^{-1})_{kr}.\end{aligned}$$

This means that the matrices $\mathbf{E}_r$ consist of linear combinations of the rank-1 matrices $H_r A_r^{\mathrm{T}}$ and that the linear combinations are the entries of the non-singular matrix $\mathbf{F}^{-1}$. Turned the other way around, we would like to find linear combinations of the matrices $\mathbf{E}_r$ that yield rank-1 matrices, because the coefficients of the linear combination may yield the matrix $\mathbf{F}$ we are looking for. To solve this problem, we need a tool that allows us to determine whether a matrix is rank-1 or not. Such a tool is offered by the following theorem [16,6].

Theorem 1. *Consider the mapping* Φ: $(\mathbf{X}, \mathbf{Y}) \in \mathbb{C}^{I \times J} \times \mathbb{C}^{I \times J} \longmapsto \Phi(\mathbf{X}, \mathbf{Y}) = \mathscr{P} \in \mathbb{C}^{I \times J \times I \times J}$ *defined by*

$$p_{ijkl} = x_{ij}y_{kl} + y_{ij}x_{kl} - x_{il}y_{kj} - y_{il}x_{kj}$$

for all index values. Given $\mathbf{X} \in \mathbb{C}^{I \times J}$, $\Phi(\mathbf{X}, \mathbf{X}) = 0$ *if and only if the rank of* $\mathbf{X}$ *is at most one.*

From the matrix $\mathbf{U}\mathbf{\Sigma}$ in SVD (13) we construct the following set of R^2 tensors $\{\mathscr{P}_{rs}\}_{r,s \in [1,R]}$:

$$\begin{aligned}\mathscr{P}_{rs} &= \Phi(\mathbf{E}_r, \mathbf{E}_s) \\ &= \Phi\left(\sum_{t=1}^{R} H_t A_t^{\mathrm{T}} (\mathbf{F}^{-1})_{tr}, \sum_{u=1}^{R} H_u A_u^{\mathrm{T}} (\mathbf{F}^{-1})_{us}\right).\end{aligned}$$

Due to the bilinearity of Φ, we have

$$\mathscr{P}_{rs} = \sum_{t,u=1}^{R} (\mathbf{F}^{-1})_{tr}(\mathbf{F}^{-1})_{us} \Phi(H_t A_t^{\mathrm{T}}, \ H_u A_u^{\mathrm{T}}). \tag{16}$$

Assume at this point that there exists a symmetric matrix $\mathbf{B} \in \mathbb{C}^{R \times R}$ satisfying (we will justify this assumption below):

$$\sum_{r,s=1}^{R} \mathscr{P}_{rs} b_{rs} = 0. \tag{17}$$

Author's Personal Copy

By substitution of (16) we obtain

$$\sum_{r,s=1}^{R}\sum_{t,u=1}^{R}(\mathbf{F}^{-1})_{tr}(\mathbf{F}^{-1})_{us}\Phi(H_tA_t^{\mathrm{T}},H_uA_u^{\mathrm{T}})b_{rs}=0.$$

According to Theorem 1, we have $\Phi(H_tA_t^{\mathrm{T}},H_tA_t^{\mathrm{T}})=0$ for all $t\in[1,R]$. Hence

$$\sum_{r,s=1}^{R}\sum_{\substack{t,u=1\\ t\neq u}}^{R}(\mathbf{F}^{-1})_{tr}(\mathbf{F}^{-1})_{us}b_{rs}\Phi(H_tA_t^{\mathrm{T}},\ H_uA_u^{\mathrm{T}})=0.$$

Furthermore, due to the symmetry of Φ and $\mathbf{B}$ we have

$$\sum_{r,s=1}^{R}\sum_{\substack{t,u=1\\ t<u}}^{R}(\mathbf{F}^{-1})_{tr}(\mathbf{F}^{-1})_{us}b_{rs}\Phi(H_tA_t^{\mathrm{T}},\ H_uA_u^{\mathrm{T}})=0. \tag{18}$$

Let us now make the crucial assumption that the tensors $\Phi(H_tA_t^{\mathrm{T}},H_uA_u^{\mathrm{T}})$, $t<u$, are linearly independent. (As will be explained in Section 4.3, this implies an upper-bound on the number of users. Below the bound, independence is induced by differences between the users in code and multipath.) Then (18) implies

$$\sum_{r,s=1}^{R}(\mathbf{F}^{-1})_{tr}(\mathbf{F}^{-1})_{us}b_{rs}=\lambda_{tu}\delta_{tu} \tag{19}$$

in which δ is the Kronecker delta ($\delta_{tu}=1$ if $t=u$, $\delta_{tu}=0$ if $t\neq u$). Eq. (19) can be rewritten as

$$\mathbf{B}=\mathbf{F}\mathbf{\Lambda}\mathbf{F}^{\mathrm{T}}, \tag{20}$$

in which $\mathbf{\Lambda}$ is a diagonal matrix whose diagonal elements are λ_{tt}, $t\in[1,R]$. It can be verified that the reverse of the reasoning above holds also true. Namely, any matrix $\mathbf{B}$ of the form (20), with $\mathbf{\Lambda}$ an arbitrary diagonal matrix, satisfies (17), so that it was justified to make the assumption leading to (17). Eq. (17) is just a set of linear equations, of which the coefficients are given by the entries of the tensors $\mathscr{P}_{rs}$ and of which the unknowns are the entries of $\mathbf{B}$. Linearly independent choices of $\mathbf{\Lambda}$ correspond to linearly independent solutions of (17). We conclude that the kernel of (17) yields R linearly independent matrices $\mathbf{B}_r$, which can all be decomposed as in (20).

The kernel matrices can be computed in a numerically reliable way as follows. Due to the symmetry of Φ and $\mathbf{B}$ we can rewrite (17) as

$$\sum_{\substack{r,s=1\\ r<s}}^{R}\mathscr{P}_{rs}b_{rs}+\frac{1}{2}\sum_{r=1}^{R}\mathscr{P}_{rr}b_{rr}=0. \tag{21}$$

After stacking the $(I\times J\times I\times J)$ tensors $\mathscr{P}_{rs}$ in I^2J^2-dimensional vectors P_{rs}, we solve the classical set of linear equations

$$(P_{11},P_{12},\ldots,P_{RR})\cdot\begin{pmatrix}x_{11}\\ x_{12}\\ \vdots\\ x_{RR}\end{pmatrix}=\begin{pmatrix}0\\ 0\\ \vdots\\ 0\end{pmatrix}. \tag{22}$$

The least-squares solution is given by the R right singular vectors of the coefficient matrix, associated with the smallest singular values. After stacking these vectors in R upper triangular matrices $\mathbf{X}_r$, the matrices $\mathbf{B}_r$ are given by $\mathbf{B}_r=\mathbf{X}_r+\mathbf{X}_r^{\mathrm{T}}$.

After the computation of the matrices $\mathbf{B}_r$, the unknown matrix $\mathbf{F}$ can be found from the following simultaneous decomposition:

$$\begin{cases}\mathbf{B}_1=\mathbf{F}\mathbf{\Lambda}_1\mathbf{F}^{\mathrm{T}},\\ \mathbf{B}_2=\mathbf{F}\mathbf{\Lambda}_2\mathbf{F}^{\mathrm{T}},\\ \vdots\\ \mathbf{B}_R=\mathbf{F}\mathbf{\Lambda}_R\mathbf{F}^{\mathrm{T}}\end{cases} \tag{23}$$

in which $\mathbf{\Lambda_1},\mathbf{\Lambda_2},\ldots,\mathbf{\Lambda_R}$ are diagonal.

The general outline of the technique is given in Table 1. Specific algorithms for solving (23) are given in Section 5. In the following subsection, we give a simple bound on the number of users R that can be allowed.

Remark. Not all the matrices $\mathbf{B}_r$ are necessarily equally accurate. In particular matrices that correspond to smaller singular values of the coefficient matrix in (22) are likely to be more accurate. Denote the singular value corresponding to $\mathbf{B}_r$ by $\sigma_{B,r}$, $r\in[1,R]$. Then, as a heuristic rule, we may weight $\mathbf{B}_r$ inversely proportional to $\sigma_{B,r}$.

4.3. Bound on the number of users

In the previous section we have shown that, under condition (11), the blind DS–CDMA separation problem can be solved if the tensors $\Phi(H_pA_p^{\mathrm{T}},H_qA_q^{\mathrm{T}})$ are linearly independent (Eq. (19)). This sufficient condition has also been found, in a different way, in [17]. It actually leads to a new bound on the number of users. It can be proven that, if the entries of $\mathbf{A}$ and $\mathbf{H}$ can be considered drawn from continuous distributions, the tensors $\Phi(H_pA_p^{\mathrm{T}},H_qA_q^{\mathrm{T}})$ are linearly independent with probability one if

$$R(R-1)\leqslant\tfrac{1}{2}(I^2-I)(J^2-J). \tag{24}$$

Author's personal copy

Table 1
Summary of the algorithm for the blind separation of DS–CDMA signals by simultaneous matrix diagonalization (SD)

- Stack $\mathscr{Y}$ in an $(IJ \times K)$ matrix $\mathbf{Y}$
- Compute SVD $\mathbf{Y} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{H}}$
- For $r \in [1, R]$, stack rth column of $\mathbf{U}\mathbf{\Sigma}$ in an $(I \times J)$ matrix $\mathbf{E}_r$
- For $r, s \in [1, R]$, $r<s$, construct the $(I \times J \times I \times J)$ tensor $\mathscr{P}_{rs} = \Phi(\mathbf{E}_r, \mathbf{E}_s)$ and stack it in an I^2J^2-dimensional vector P_{rs}
- Construct the $(I^2J^2 \times R(R+1)/2)$ matrix $(P_{11}, P_{12}, \ldots, P_{RR})$ and compute its R right singular vectors associated with the R lowest singular values
- Stack each of these vectors in an $(R \times R)$ upper triangular matrix $\mathbf{X}_r$. Compute $\mathbf{B}_r = \mathbf{X}_r + \mathbf{X}_r^{\mathrm{T}}$
- Obtain the matrix $\mathbf{F}$ by means of simultaneous diagonalization:

$$\begin{cases} \mathbf{B}_1 = \mathbf{F}\mathbf{\Lambda}_1\mathbf{F}^{\mathrm{T}} \\ \mathbf{B}_2 = \mathbf{F}\mathbf{\Lambda}_2\mathbf{F}^{\mathrm{T}} \\ \vdots \\ \mathbf{B}_R = \mathbf{F}\mathbf{\Lambda}_R\mathbf{F}^{\mathrm{T}} \end{cases}$$

- Estimate $\mathbf{S}$ as $\mathbf{V}^*\mathbf{F}^{-\mathrm{T}}$

The proof is technical; we refer to [6]. Note that bound (24) is much more relaxed than the Kruskal bound (8). In particular, for I and J large, the new bound on R depends on the product of I and J, rather than on their sum.

5. Solution of the simultaneous diagonalization problem

In this section we will explain how Eq. (23) can be solved. Let us stack the matrices $\mathbf{B}_1, \ldots, \mathbf{B}_R$ in an $(R \times R \times R)$ tensor $\mathscr{B}$ as follows:

$$b_{ijr} = (\mathbf{B}_r)_{ij}. \tag{25}$$

Let us further stack the diagonals of $\mathbf{\Lambda}_1$, …, $\mathbf{\Lambda}_R$ in an $(R \times R)$ matrix $\mathbf{D}$:

$$d_{ij} = (\mathbf{\Lambda}_j)_{ii}. \tag{26}$$

Now it can be seen that, actually, (23) is itself a PARAFAC:

$$\mathscr{B} = \sum_{r=1}^{R} F_r \circ F_r \circ D_r. \tag{27}$$

The main difference between the PARAFAC of $\mathscr{B}$ and the original PARAFAC of $\mathscr{Y}$ is that the number of rank-1 terms in (27) does not exceed the dimension R, which makes it easier to find the solution. As a matter of fact, in the absence of noise the unknown matrix $\mathbf{F}$ follows from an ordinary EVD [18]:

$$\mathbf{B}_1\mathbf{B}_2^{-1} = \mathbf{F}\mathbf{\Lambda}_1\mathbf{\Lambda}_2^{-1}\mathbf{F}^{-1}. \tag{28}$$

In practice it is preferable to take all matrices into account. In Section 5.1 we briefly describe the standard ALS approach to solve (27) [1,4]. In Section 5.2 we summarize the solution proposed in [11]. Background material can be found in [19].

5.1. ALS iteration

The solution of (27) can be computed using the standard ALS algorithm, briefly introduced in Section 3, if we ignore the symmetry. Namely, instead of solving (23), we solve the set of equations

$$\begin{cases} \mathbf{B}_1 = \mathbf{F}\mathbf{\Lambda}_1\tilde{\mathbf{F}}, \\ \mathbf{B}_2 = \mathbf{F}\mathbf{\Lambda}_2\tilde{\mathbf{F}}, \\ \vdots \\ \mathbf{B}_R = \mathbf{F}\mathbf{\Lambda}_R\tilde{\mathbf{F}} \end{cases} \tag{29}$$

by means of an ALS iteration. The symmetry $\tilde{\mathbf{F}} = \mathbf{F}^{\mathrm{T}}$ is restored upon convergence. Conditional updates are based on the application of

$$\mathrm{vec}(\mathbf{XYZ}) = (\mathbf{Z}^{\mathrm{T}} \otimes \mathbf{X})\mathrm{vec}(\mathbf{Y}) \tag{30}$$

in which $\mathbf{X}$, $\mathbf{Y}$, $\mathbf{Z}$ are matrices of compatible dimensions.

The ALS iteration consists of alternating between the following substeps:

(1) *Updating the estimate of* $\mathbf{\Lambda}_r$.

Applying (30) to (29), we obtain

$$\mathrm{vec}(\mathbf{B}_r) = (\tilde{\mathbf{F}}^{\mathrm{T}} \otimes \mathbf{F})\mathrm{vec}(\mathbf{\Lambda}_r),$$

i.e.,

$$\mathrm{vec}(\mathbf{B}_r) = (\tilde{\mathbf{F}}^{\mathrm{T}} \odot \mathbf{F})\mathrm{vecdiag}(\mathbf{\Lambda}_r). \tag{31}$$

Author's personal copy

Matrix $\mathbf{\Lambda}_r$, $r \in [1, R]$, follows from this set of linear equations.

(2) *Updating the estimate of* $\mathbf{F}$.

Define $\Delta_1 = [\mathbf{\Lambda}_1\tilde{\mathbf{F}}, \mathbf{\Lambda}_2\tilde{\mathbf{F}}, \ldots, \mathbf{\Lambda}_R\tilde{\mathbf{F}}]$ and $\Delta_2 = [\mathbf{B}_1, \mathbf{B}_2, \ldots, \mathbf{B}_R]$. Eq. (29) can be written as $\Delta_2 = \mathbf{F}\Delta_1 = \mathbf{I}_R\mathbf{F}\Delta_1$. Applying (30), we obtain

$$\mathrm{vec}(\Delta_2) = (\Delta_1^{\mathrm{T}} \otimes \mathbf{I}_R)\mathrm{vec}(\mathbf{F}). \tag{32}$$

From this set of linear equations matrix $\mathbf{F}$ can be computed.

(3) *Updating the estimate of* $\tilde{\mathbf{F}}$.

Define $\Delta_3 = [(\mathbf{F}\mathbf{\Lambda}_1)^{\mathrm{T}}, (\mathbf{F}\mathbf{\Lambda}_2)^{\mathrm{T}}, \ldots, (\mathbf{F}\mathbf{\Lambda}_R)^{\mathrm{T}}]^{\mathrm{T}}$ and $\Delta_4 = [\mathbf{B}_1^{\mathrm{T}}, \mathbf{B}_2^{\mathrm{T}}, \ldots, \mathbf{B}_R^{\mathrm{T}}]^{\mathrm{T}}$. Eq. (29) can be rewritten as $\Delta_4 = \Delta_3\tilde{\mathbf{F}} = \Delta_3\tilde{\mathbf{F}}\mathbf{I}_R$. Applying (30), we obtain:

$$\mathrm{vec}(\Delta_4) = (\mathbf{I}_R \otimes \Delta_3)\mathrm{vec}(\tilde{\mathbf{F}}), \tag{33}$$

from which $\tilde{\mathbf{F}}$ follows.

We refer to this algorithm as simultaneous diagonalization by ALS (SD-ALS). As initial values of $\mathbf{F}$ and $\tilde{\mathbf{F}}$, we can take the eigenmatrix in (28) and its transpose, respectively. We decide that the algorithm has converged when the Frobenius norm of the difference between the estimates of $\mathbf{F}$ at iteration steps k and $k+1$ is smaller than a certain tolerance $\varepsilon_{\text{SD-ALS}}$.

5.2. Extended QZ iteration

van der Veen and Paulraj solved the joint diagonalization problem (23) by turning it into a simultaneous triangularization involving unitary matrices [11]. The latter problem was solved by means of a multi-matrix extension of the *QZ*-iteration for the computation of the generalized Schur decomposition of two matrices [20]. We will denote their algorithm as simultaneous diagonalization by extended *QZ*-iteration (SD-QZ).

Let us briefly explain how SD-QZ works. Substitution in (23) of the *QR* decomposition of $\mathbf{F}$ and the *RQ* decomposition of $\mathbf{F}^{\mathrm{T}}$,

$$\mathbf{F} = \mathbf{Q}^{\mathrm{H}}\mathbf{R}', \tag{34}$$

$$\mathbf{F}^{\mathrm{T}} = \mathbf{R}''\mathbf{Z}^{\mathrm{H}}, \tag{35}$$

yields

$$\begin{cases} \mathbf{Q}\mathbf{B}_1\mathbf{Z} = \mathbf{R}_1 \overset{\text{def}}{=} \mathbf{R}'\mathbf{\Lambda}_1\mathbf{R}'', \\ \mathbf{Q}\mathbf{B}_2\mathbf{Z} = \mathbf{R}_2 \overset{\text{def}}{=} \mathbf{R}'\mathbf{\Lambda}_2\mathbf{R}'', \\ \vdots \\ \mathbf{Q}\mathbf{B}_R\mathbf{Z} = \mathbf{R}_R \overset{\text{def}}{=} \mathbf{R}'\mathbf{\Lambda}_R\mathbf{R}'' \end{cases} \tag{36}$$

in which $\mathbf{Q}, \mathbf{Z} \in \mathbb{C}^{R\times R}$ are unitary and in which the matrices $\mathbf{R}_r \in \mathbb{C}^{R\times R}$ are upper triangular.

The aim is now to find two unitary matrices $\mathbf{Q}$ and $\mathbf{Z}$ such that $\mathbf{Q}\mathbf{B}_r\mathbf{Z}$, for all $r \in [1, R]$, are jointly as upper triangular as possible. The solution can be obtained by means of an extended *QZ* iteration, alternating between updates of $\mathbf{Q}$ and $\mathbf{Z}$.

Let us first consider an update of $\mathbf{Q}$. Let us assume that after iteration step k, we have

$$\mathbf{R}_r^{(k)} = \mathbf{Q}^{(k)}\mathbf{B}_r\mathbf{Z}^{(k)}, \quad r \in [1, R]. \tag{37}$$

The goal is to find a unitary matrix $\tilde{\mathbf{Q}}$ such that the products $\tilde{\mathbf{Q}}\mathbf{R}_r^{(k)}$ are jointly more upper triangular than the matrices $\mathbf{R}_r^{(k)}$. The matrix $\tilde{\mathbf{Q}}$ is constructed as a product of unitary matrices that impose the upper triangular structure on the first, second, ... columns of $\mathbf{R}_r^{(k)}$, respectively,

$$\mathbf{Q}^{(k+1)} = \tilde{\mathbf{Q}}\mathbf{Q}^{(k)} = \left[\begin{array}{c|c} \mathbf{I}_{R-2} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{H}_{R-1} \end{array}\right] \cdots \left[\begin{array}{c|c} 1 & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{H}_2 \end{array}\right] \mathbf{H}_1\mathbf{Q}^{(k)}. \tag{38}$$

in which the matrices $\mathbf{H}_r \in \mathbb{C}^{(R-r+1)\times(R-r+1)}$ are unitary. The matrix $\mathbf{H}_1$, for instance, makes the subdiagonal entries of the first column of the matrices $\mathbf{R}_r^{(k)}$ small (Fig. 2). It is sufficient to find its first row; for the other rows one can take any orthonormal basis of the orthogonal complement. The first row, denoted by V^{H}, is determined as the vector that maximizes

$$f(V) = V^{\mathrm{H}}\mathbf{W}_1\mathbf{W}_1^{\mathrm{H}}V, \tag{39}$$

in which $\mathbf{W}_1 \in \mathbb{C}^{R\times R}$ is the matrix in which the first columns of the $\mathbf{R}_r^{(k)}$'s are stacked one after the other. The optimal vector V is the first left singular vector of $\mathbf{W}_1$. Hence, $\mathbf{H}_1$ can be taken equal to the Hermitian transpose of the matrix of left singular vectors of $\mathbf{W}_1$.

Matrix $\mathbf{H}_2$ subsequently minimizes the subdiagonal entries in the second column of the matrices $\mathbf{H}_1\mathbf{R}_r^{(k)}$, and so on.

Updating $\mathbf{Z}^{(k)}$ is analogous. We have

$$\mathbf{Z}^{(k+1)} = \mathbf{Z}^{(k)}\mathbf{G}_1 \left[\begin{array}{c|c} 1 & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{G}_2 \end{array}\right] \cdots \left[\begin{array}{c|c} \mathbf{I}_{R-2} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{G}_{R-1} \end{array}\right], \tag{40}$$

Authors personal copy

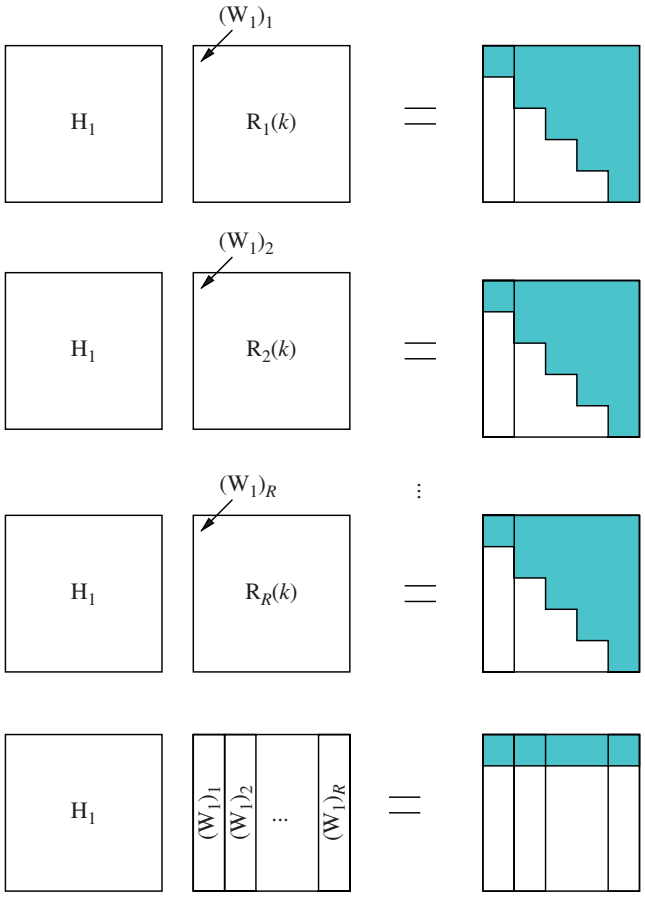


Fig. 2. Construction of matrix $\mathbf{W}_1$.

in which $\mathbf{G}_1$, $\mathbf{G}_2$, ... now subsequently impose the upper triangular structure on the Rth, $(R-1)$th, ... rows, respectively. For instance, if the last column of $\mathbf{G}_1$ is denoted by Z, the computation of $\mathbf{G}_1$ is based on the maximization of

$$\tilde{f}(Z) = Z^{\mathrm{H}}\tilde{\mathbf{W}}_1^{\mathrm{H}}\tilde{\mathbf{W}}_1 Z, \tag{41}$$

in which $\tilde{\mathbf{W}}_1$ is a matrix in which the last rows of the matrices $\mathbf{Q}^{(k+1)}\mathbf{R}_r^{(k)}$ are stacked one above the other. The factor $\mathbf{G}_1$ is consequently taken equal to the matrix of right singular vectors of $\tilde{\mathbf{W}}_1$, with the singular values in increasing order of magnitude. The matrix $\mathbf{G}_2$ is obtained in the same way from the matrices $\mathbf{Q}^{(k+1)}\mathbf{R}_r^{(k)}\mathbf{G}_1$, of which the last column and row have been peeled off, and so on.

The SD-ALS algorithm can be initialized by means of the generalized Schur decomposition of the pair $(\mathbf{B}_1, \mathbf{B}_2)$ [20]. The iteration is stopped when the Frobenius norm $\|\mathbf{Q}^{(k+1)} - \mathbf{Q}^{(k)}\|$ is smaller than a certain tolerance $\varepsilon_{\mathrm{SD-QZ}}$.

After calculating $\mathbf{Q}$, $\mathbf{Z}$ and $\mathbf{R}_r$, $r \in [1, R]$, the matrix $\mathbf{F}$ can be estimated as follows. Define

$$\mathbf{D}_1 = \begin{pmatrix} \mathrm{vecdiag}(\mathbf{R}_1)^{\mathrm{T}} \\ \vdots \\ \mathrm{vecdiag}(\mathbf{R}_R)^{\mathrm{T}} \end{pmatrix}. \tag{42}$$

It can be proved [11] that in the absence of noise

$$\sum_{i=1}^{R} (\mathbf{D}_1^{-1})_{ri}\mathbf{B}_i = \alpha_r F_r F_r^{\mathrm{T}} \tag{43}$$

in which α_r is an irrelevant scaling factor. In practice, the matrix in (43) is not exactly rank-1, and F_r is estimated as its dominant left singular vector.

6. Combination with the CM constraint

In the preceding sections we have exploited the structure of the columns of $\mathbf{A} \odot \mathbf{H}$ in Eq. (12). In the terminology of [13], this is a column span method. On the other hand, row span methods exploit the properties of the matrix $\mathbf{S}^{\mathrm{T}}$ in (12). In this section we will derive combined column/row span algorithms, which take advantage of both. More precisely, we will show how the CM property can be combined with the PARAFAC structure.

According to Eq. (15), the matrix $\mathbf{V}$ of right singular vectors of $\mathbf{Y}$ satisfies

$$\mathbf{V}^{\mathrm{H}} = \mathbf{F}\mathbf{S}^{\mathrm{T}}. \tag{44}$$

This is the classical expression of an $(R \times R)$ instantaneous mixture of source signals. It is shown in [11] that, if the transmitted sequences are CM, the demixing matrix may be found from the following simultaneous matrix decomposition:

$$\begin{cases} \mathbf{C}_1 = \mathbf{F}^{-\mathrm{H}}\mathbf{\Omega}_1\mathbf{F}^{-1}, \\ \mathbf{C}_2 = \mathbf{F}^{-\mathrm{H}}\mathbf{\Omega}_2\mathbf{F}^{-1}, \\ \vdots \\ \mathbf{C}_R = \mathbf{F}^{-\mathrm{H}}\mathbf{\Omega}_R\mathbf{F}^{-1}, \end{cases} \tag{45}$$

where matrices $\mathbf{\Omega}_r$ are diagonal and where the $\mathbf{C}_r$'s are obtained from $\mathbf{V}$. This technique is called analytical constant modulus algorithm (ACMA). For the computation of $\mathbf{C}_r$, we refer to [11]. (Like the matrices $\mathbf{B}_r$ in Section 4.2, $\mathbf{C}_r$ are obtained from an overdetermined set of linear equations. Hence they may also be weighted inversely proportional to the corresponding singular values $\sigma_{C,r}$ of the coefficient matrix.) Because this set of equations is

Author's personal copy

very similar to the set obtained from the CDMA structure constraint (23), they can be solved jointly.

Actually, jointly solving (23) and (45) can be interpreted as computing two coupled tensor decompositions. Define $\mathbf{G} \stackrel{\text{def}}{=} \mathbf{F}^{-\mathrm{H}}$. Let us stack the matrices $\mathbf{C}_1, \ldots, \mathbf{C}_R$ in an $(R \times R \times R)$ tensor $\mathscr{C}$ as follows:

$$c_{ijr} = (\mathbf{C}_r)_{ij}. \tag{46}$$

Let us further stack the diagonals of $\mathbf{\Omega}_1, \ldots, \mathbf{\Omega}_R$ in an $(R \times R)$ matrix $\mathbf{D}'$:

$$d'_{ij} = (\mathbf{\Omega}_j)_{ii}. \tag{47}$$

In analogy with Eq. (27) it can be seen that (45) is itself a PARAFAC:

$$\mathscr{C} = \sum_{r=1}^{R} G_r \circ G_r^* \circ D'_r. \tag{48}$$

Eqs. (27) and (48) have to be solved under the constraint $\mathbf{G} = \mathbf{F}^{-\mathrm{H}}$. We will derive appropriate generalizations of the SD-ALS and SD-QZ algorithms in Sections 6.1 and 6.2, respectively.

6.1. Generalized ALS iteration

As in Section 5.1, $\mathbf{F}^{\mathrm{T}}$ is replaced with $\tilde{\mathbf{F}}$. Eqs. (27) and (48) then become

$$\begin{cases} \mathbf{B}_1 = \mathbf{F}\mathbf{\Lambda}_1\tilde{\mathbf{F}}, \\ \mathbf{B}_2 = \mathbf{F}\mathbf{\Lambda}_2\tilde{\mathbf{F}}, \\ \vdots \\ \mathbf{B}_R = \mathbf{F}\mathbf{\Lambda}_R\tilde{\mathbf{F}}, \\ \mathbf{C}_1 = \tilde{\mathbf{F}}^{-*}\mathbf{\Omega}_1\mathbf{F}^{-1}, \\ \mathbf{C}_2 = \tilde{\mathbf{F}}^{-*}\mathbf{\Omega}_2\mathbf{F}^{-1}, \\ \vdots \\ \mathbf{C}_R = \tilde{\mathbf{F}}^{-*}\mathbf{\Omega}_R\mathbf{F}^{-1}. \end{cases} \tag{49}$$

With this set of equations, the following cost function can be associated:

$$f(\mathbf{F}, \tilde{\mathbf{F}}, \{\mathbf{\Lambda}_r\}, \{\mathbf{\Omega}_r\}) = \sum_{r=1}^{R} (\|\mathbf{B}_r - \mathbf{F}\mathbf{\Lambda}_r\tilde{\mathbf{F}}\|^2 + \|\tilde{\mathbf{F}}^*\mathbf{C}_r\mathbf{F} - \mathbf{\Omega}_r\|^2). \tag{50}$$

The generalized ALS iteration consists of alternating between the following substeps:

(1) *Updating the estimate of* $\mathbf{\Lambda}_r$ *and* $\mathbf{\Omega}_r$.

Equations $\mathbf{B}_r = \mathbf{F}\mathbf{\Lambda}_r\tilde{\mathbf{F}}$ and $\mathbf{C}_r = \tilde{\mathbf{F}}^{-*}\mathbf{\Omega}_r\mathbf{F}^{-1}$ can be rewritten as

$$\mathrm{vec}(\mathbf{B}_r) = (\tilde{\mathbf{F}}^{\mathrm{T}} \odot \mathbf{F})\mathrm{vecdiag}(\mathbf{\Lambda}_r) \tag{51}$$

and

$$\mathrm{vecdiag}(\mathbf{\Omega}_r) = (\mathbf{F}^{\mathrm{T}} \odot \tilde{\mathbf{F}}^*)\mathrm{vec}(\mathbf{C}_r). \tag{52}$$

Matrices $\mathbf{\Lambda}_r$ and $\mathbf{\Omega}_r$, $r \in [1, R]$, follow from these linear equations.

(2) *Updating the estimate of* $\mathbf{F}$.

Define $\mathit{\Delta}_1 = [\mathbf{\Lambda}_1\tilde{\mathbf{F}}, \mathbf{\Lambda}_2\tilde{\mathbf{F}}, \ldots, \mathbf{\Lambda}_R\tilde{\mathbf{F}}]$, $\mathit{\Delta}_2 = [\mathbf{B}_1, \mathbf{B}_2, \ldots, \mathbf{B}_R]$, $\mathit{\Gamma}_1 = [(\tilde{\mathbf{F}}^*\mathbf{C}_1)^{\mathrm{T}}, (\tilde{\mathbf{F}}^*\mathbf{C}_2)^{\mathrm{T}}, \ldots, (\tilde{\mathbf{F}}^*\mathbf{C}_R)^{\mathrm{T}}]^{\mathrm{T}}$ and $\mathit{\Gamma}_2 = [\mathbf{\Omega}_1^{\mathrm{T}}, \mathbf{\Omega}_2^{\mathrm{T}}, \ldots, \mathbf{\Omega}_R^{\mathrm{T}}]^{\mathrm{T}}$. In terms of these matrices, (49) can be rewritten as

$$\begin{cases} \mathit{\Delta}_2 = \mathbf{F}\mathit{\Delta}_1 = \mathbf{I}_R\mathbf{F}\mathit{\Delta}_1, \\ \mathit{\Gamma}_2 = \mathit{\Gamma}_1\mathbf{F} = \mathit{\Gamma}_1\mathbf{F}\mathbf{I}_R. \end{cases} \tag{53}$$

By means of (30) we obtain an overdetermined set of equations from which $\mathbf{F}$ can be computed:

$$\begin{bmatrix} \mathrm{vec}(\mathit{\Delta}_2), \\ \mathrm{vec}(\mathit{\Gamma}_2). \end{bmatrix} = \begin{bmatrix} \mathit{\Delta}_1^{\mathrm{T}} \otimes \mathbf{I}_R \\ \mathbf{I}_R \otimes \mathit{\Gamma}_1 \end{bmatrix} \mathrm{vec}(\mathbf{F}). \tag{54}$$

(3) *Updating the estimate of* $\tilde{\mathbf{F}}$.

Define $\mathit{\Delta}_3 = [(\mathbf{F}\mathbf{\Lambda}_1)^{\mathrm{T}}, (\mathbf{F}\mathbf{\Lambda}_2)^{\mathrm{T}}, \ldots, (\mathbf{F}\mathbf{\Lambda}_R)^{\mathrm{T}}]^{\mathrm{T}}$, $\mathit{\Delta}_4 = [\mathbf{B}_1^{\mathrm{T}}, \mathbf{B}_2^{\mathrm{T}}, \ldots, \mathbf{B}_R^{\mathrm{T}}]^{\mathrm{T}}$, $\mathit{\Gamma}_3 = [\mathbf{C}_1\mathbf{F}, \mathbf{C}_2\mathbf{F}, \ldots, \mathbf{C}_R\mathbf{F}]$ and $\mathit{\Gamma}_4 = [\mathbf{\Omega}_1, \mathbf{\Omega}_2, \ldots, \mathbf{\Omega}_R]$. In terms of these matrices, (49) can be rewritten as

$$\begin{cases} \mathit{\Delta}_4 = \mathit{\Delta}_3\tilde{\mathbf{F}} = \mathit{\Delta}_3\tilde{\mathbf{F}}\mathbf{I}_R, \\ \mathit{\Gamma}_4^* = \tilde{\mathbf{F}}\mathit{\Gamma}_3^* = \mathbf{I}_R\tilde{\mathbf{F}}\mathit{\Gamma}_3^*. \end{cases} \tag{55}$$

By means of (30) we obtain an overdetermined set of equations from which $\tilde{\mathbf{F}}$ can be computed:

$$\begin{bmatrix} \mathrm{vec}(\mathit{\Delta}_4) \\ \mathrm{vec}(\mathit{\Gamma}_4^*) \end{bmatrix} = \begin{bmatrix} \mathbf{I}_R \otimes \mathit{\Delta}_3 \\ \mathit{\Gamma}_3^{\mathrm{H}} \otimes \mathbf{I}_R \end{bmatrix} \mathrm{vec}(\tilde{\mathbf{F}}). \tag{56}$$

We refer to this algorithm as coupled simultaneous diagonalization by ALS (CSD-ALS). Like in Section 5.1, $\mathbf{F}$ can be initialized with the eigenmatrix of $\mathbf{B}_1\mathbf{B}_2^{-1}$ and $\tilde{\mathbf{F}}$ with its transpose. We decide that the algorithm has converged when the Frobenius norm of the difference between the estimates of $\mathbf{F}$ at iteration steps k and $k+1$ is smaller than a certain tolerance $\varepsilon_{\text{CSD-ALS}}$.

6.2. Generalized QZ iteration

We will now derive a generalization of SD-QZ, to which we will refer as Coupled Simultaneous Diagonalization by generalized *QZ* iteration (CSD-QZ).

In accordance with (34)–(35), we have also

$$\mathbf{F}^{-\mathrm{H}} = \mathbf{Q}^{\mathrm{H}}(\mathbf{R}')^{-\mathrm{H}}, \tag{57}$$

$$\mathbf{F}^{-1} = (\mathbf{R}'')^{-\mathrm{T}}\mathbf{Z}^{\mathrm{T}}. \tag{58}$$

Author's personal copy

Define $\mathbf{L}_r = (\mathbf{R}')^{-\mathrm{H}}\mathbf{\Omega}_r(\mathbf{R}'')^{-\mathrm{T}}$, $r \in [1, R]$. Substitution of (34), (35), (57), (58) in (23) and (45) yields

$$\begin{cases} \mathbf{Q}\mathbf{B}_1\mathbf{Z} = \mathbf{R}_1, \\ \mathbf{Q}\mathbf{B}_2\mathbf{Z} = \mathbf{R}_2, \\ \vdots \\ \mathbf{Q}\mathbf{B}_R\mathbf{Z} = \mathbf{R}_R, \\ \mathbf{Q}\mathbf{C}_1\mathbf{Z}^* = \mathbf{L}_1, \\ \mathbf{Q}\mathbf{C}_2\mathbf{Z}^* = \mathbf{L}_2, \\ \vdots \\ \mathbf{Q}\mathbf{C}_R\mathbf{Z}^* = \mathbf{L}_R. \end{cases} \tag{59}$$

in which $\mathbf{Q}, \mathbf{Z} \in \mathbb{C}^{R\times R}$ are unitary and in which the matrices $\mathbf{R}_r \in \mathbb{C}^{R\times R}$ are upper triangular and the matrices $\mathbf{L}_r \in \mathbb{C}^{R\times R}$ are lower triangular. This coupled set of matrix decompositions is visualized in Fig. 3.

As in Section 5.2, we alternate between updates of $\mathbf{Q}$ and $\mathbf{Z}$, which take again the form (38) and (40), respectively. The matrix $\mathbf{H}_1$ imposes the upper triangular structure on the first columns of $\mathbf{R}_r^{(k)}$ and the lower triangular structure on the second columns of $\mathbf{L}_r^{(k)}$. Next, the matrix $\mathbf{H}_2$ imposes the upper triangular structure on the second columns of $\mathbf{R}_r^{(k)}$ and the lower triangular structure on the third columns of $\mathbf{L}_r^{(k)}$, and so on. Likewise, the matrix $\mathbf{G}_1$ imposes the upper triangular structure on the Rth rows of $\mathbf{Q}\mathbf{R}_r^{(k)}$ and the lower triangular structure on the $(R-1)$th rows of $\mathbf{Q}\mathbf{L}_r^{(k)}$. Next, the matrix $\mathbf{G}_2$ imposes the upper triangular structure on the $(R-1)$th rows of $\mathbf{Q}\mathbf{R}_r^{(k)}$ and the lower triangular structure on the $(R-2)$th rows of $\mathbf{Q}\mathbf{L}_r^{(k)}$, and so on.

Let us consider in detail the computation of $\mathbf{H}_1$. This matrix has to make the subdiagonal entries of the first column of the matrices $\mathbf{R}_r^{(k)}$ small, while making the superdiagonal entries of the second column of the matrices $\mathbf{L}_r^{(k)}$ big. Let us denote the first row of $\mathbf{H}_1$ by V^{H}. If we would only impose the PARAFAC structure, i.e., if we would only use the matrices $\mathbf{B}_r$, then V^{H} would follow from maximization of the function defined in (39). On the other hand, if we would only impose the CM constraint, i.e., if we would only use the matrices $\mathbf{C}_r$, then V^{H} would minimize the sum of squared moduli of the entries at position $(1,2)$ of $\mathbf{H}_1\mathbf{L}_r^{(k)}$. Formally, V^{H} would minimize $V^{\mathrm{H}}\mathbf{W}_1'\mathbf{W}_1'^{\mathrm{H}}V$, in which $\mathbf{W}_1' \in \mathbb{C}^{R\times R}$ is a matrix in which the second columns of the $\mathbf{L}_r^{(k)}$'s are stacked one after the other. For the combined set of equations (59), we compute the vector V^{H} that *maximizes*

$$f(V) = V^{\mathrm{H}}\mathbf{W}_1\mathbf{W}_1^{\mathrm{H}}V - V^{\mathrm{H}}\mathbf{W}_1'\mathbf{W}_1'^{\mathrm{H}}V + \|\mathbf{W}_1'\|^2 \tag{60}$$

in which the regularization term $\|\mathbf{W}_1'\|^2$ makes the function f always positive. The optimal vector V is the dominant eigenvector of $\mathbf{W}_1\mathbf{W}_1^{\mathrm{H}} - \mathbf{W}_1'\mathbf{W}_1'^{\mathrm{H}} + \|\mathbf{W}_1'\|^2\mathbf{I}_R$. Hence, the matrix $\mathbf{H}_1$ can be taken equal to the Hermitian transpose of the unitary eigenmatrix of $\mathbf{W}_1\mathbf{W}_1^{\mathrm{H}} - \mathbf{W}_1'\mathbf{W}_1'^{\mathrm{H}} + \|\mathbf{W}_1'\|^2\mathbf{I}_R$, with the eigenvalues in decreasing order of magnitude. The matrix $\mathbf{H}_2$ is then derived in the same way from the matrices $\mathbf{H}_1\mathbf{R}_r^{(k)}$ and $\mathbf{H}_1\mathbf{L}_r^{(k)}$, of which the first column and row are peeled off, and so on.

Similarly, the matrix $\mathbf{G}_1$ has to minimize the below-diagonal entries of the last rows of the matrix $\mathbf{Q}^{(k+1)}\mathbf{R}_r^{(k)}$ while maximizing the below-diagonal entries of the $(R-1)$th rows of $\mathbf{Q}^{(k+1)^*}\mathbf{L}_r^{(k)^*}$. Let $\tilde{\mathbf{W}}_1$ be the matrix in which the last rows of the matrices $\mathbf{Q}^{(k+1)}\mathbf{R}_r^{(k)}$ are stacked one above the other and let $\tilde{\mathbf{W}}_1'$ be the matrix in which the $(R-1)$th rows of the matrices $\mathbf{Q}^{(k+1)}\mathbf{L}_r^{(k)}$ are stacked one above the other. Denoting the last column of $\mathbf{G}_1$ by Z, we will now maximize

$$\tilde{f}(Z) = Z^{\mathrm{H}}\tilde{\mathbf{W}}_1^{\mathrm{H}}\tilde{\mathbf{W}}_1 Z - Z^{\mathrm{H}}(\tilde{\mathbf{W}}_1'^{\mathrm{H}}\tilde{\mathbf{W}}_1')^* Z + \|\tilde{\mathbf{W}}_1'\|^2. \tag{61}$$

The optimal vector Z is the eigenvector of $\tilde{\mathbf{W}}_1^{\mathrm{H}}\tilde{\mathbf{W}}_1 - (\tilde{\mathbf{W}}_1'^{\mathrm{H}}\tilde{\mathbf{W}}_1')^* + \|\tilde{\mathbf{W}}_1'\|^2\mathbf{I}_R$, corresponding to the largest eigenvalue. The matrix $\mathbf{G}_1$ is consequently taken equal to the unitary eigenmatrix of $\tilde{\mathbf{W}}_1^{\mathrm{H}}\tilde{\mathbf{W}}_1 - (\tilde{\mathbf{W}}_1'^{\mathrm{H}}\tilde{\mathbf{W}}_1')^* + \|\tilde{\mathbf{W}}_1'\|^2\mathbf{I}_R$, with the eigenvalues sorted in increasing order of magnitude. The matrix $\mathbf{G}_2$ is then derived in the same way from the matrices $\mathbf{Q}^{(k+1)}\mathbf{R}_r^{(k)}\mathbf{G}_1$ and $\mathbf{Q}^{(k+1)}\mathbf{L}_r^{(k)}\mathbf{G}_1$, of which the last column and row have been peeled off, and so on.

The CSD-QZ iteration is stopped when the Frobenius norm $\|\mathbf{Q}^{(k+1)} - \mathbf{Q}^{(k)}\|$ is smaller than a certain tolerance $\varepsilon_{\mathrm{CSD-QZ}}$.

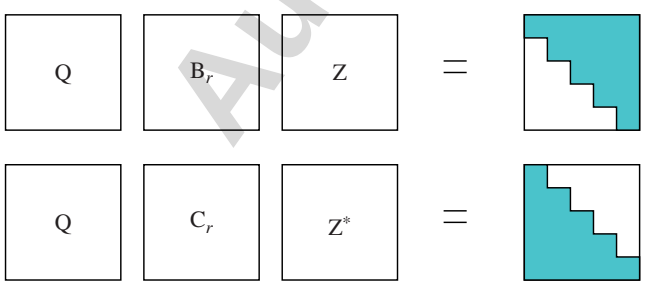


Fig. 3. Visualization of coupled set of matrix decompositions (59).

Author's personal copy

After having estimated $\mathbf{Q}$ and $\mathbf{Z}$, two estimates of $\mathbf{F}$ can be obtained from (59). A first estimate follows from the matrices $\mathbf{B}_r$, as in Section 5.2. An estimate of $\mathbf{F}^{-\mathrm{H}}$, and hence a second estimate of $\mathbf{F}$, is obtained from the matrices $\mathbf{C}_r$. In analogy with (42), the diagonals of the matrices $\mathbf{L}_r$ are stacked in a matrix $\mathbf{D}_2$. The columns of $\mathbf{F}^{-\mathrm{H}}$ are then obtained in analogy with (43). From the structure of (59) we have that the columns of the two estimates of $\mathbf{F}$ are automatically paired, i.e., they are in the same order. The final estimate of $\mathbf{F}$ can then simply be obtained as the average of the two subsolutions, possibly weighted relative to the supposed accuracy of (23) and (45) (cf. following subsection).

6.3. Weighting the equations

When combining sets (23) and (45), we have to take into account their relative accuracy. However, the prediction of the accuracy of the individual sets is hard and depends on the characteristics of channel and noise, the number of samples, etc. For CM-based signal separation, some partial results are given in [21]. Here we will follow a more heuristic approach. The precision of the individual sets (23) and (45) will be evaluated by a posteriori inspection of the least-squares error resulting from each set separately.

Assume that solving (23) yields

$$\mathbf{B}_r = \hat{\mathbf{F}}_B \hat{\mathbf{\Lambda}}_r \hat{\mathbf{F}}_B^{\mathrm{T}}, \quad i \in [1, R], \tag{62}$$

where the matrices $\hat{\mathbf{\Lambda}}_r$ are not necessarily fully diagonal. Imposing the diagonality constraint yields approximations of the matrices $\mathbf{B}_r$:

$$\hat{\mathbf{B}}_r = \hat{\mathbf{F}}_B \,\mathrm{diag}(\mathrm{vecdiag}(\hat{\mathbf{\Lambda}}_r))\hat{\mathbf{F}}_B^{\mathrm{T}}, \tag{63}$$

where diag(vecdiag($\mathbf{A}$)) is the matrix whose diagonal elements are the diagonal elements of $\mathbf{A}$ and whose off-diagonal elements are zero. The overall relative error associated with (23) is then given by

$$e_B^2 = \frac{\sum_{r=1}^{R} \|\mathbf{B}_r - \hat{\mathbf{B}}_r\|^2}{\sum_{r=1}^{R} \|\mathbf{B}_r\|^2}. \tag{64}$$

In a similar way, the overall relative error associated with (45) is given by

$$e_C^2 = \frac{\sum_{r=1}^{R} \|\mathbf{C}_r - \hat{\mathbf{C}}_r\|^2}{\sum_{r=1}^{R} \|\mathbf{C}_r\|^2} \tag{65}$$

with

$$\hat{\mathbf{C}}_r = \hat{\mathbf{F}}_C^{-\mathrm{H}} \,\mathrm{diag}(\mathrm{vecdiag}(\hat{\mathbf{\Omega}}_r))\hat{\mathbf{F}}_C^{-1}. \tag{66}$$

The combined problem (23), (45) can then be solved as follows. First solve (23) and (45) separately, whereby the result of one of both can be used to initialize the other. Then evaluate e_B and e_C. If one of the two sets turns out to be much more accurate than the other, then we simply retain the corresponding estimate of $\mathbf{F}$. In this case, combining the two sets will not substantially increase the precision. However, if e_B and e_C are of the same order of magnitude, then combining (23) and (45) may be useful. In that case, we divide the matrices $\mathbf{B}_r$ and $\mathbf{C}_r$ by e_B and e_C, respectively, and we apply the generalized ALS algorithm derived in Section 6.1 or the generalized extended QZ algorithm derived in Section 6.2. The latter algorithms may be initialized with the result of the most reliable subset, (23) or (45).

7. Simulation results

A first simulation illustrates the performance of the techniques developed in Sections 3–5. We consider the transmission of $K = 200$ QPSK-symbols by $R = 7$ simultaneous users in a system with $I = 4$ antennas and spreading gain $J = 4$. Note that the number of users is above the Kruskal bound (8). The entries of $\mathbf{A}$ and $\mathbf{H}$ are drawn from a Gaussian zero-mean unit-variance distribution. Note that $\mathbf{A}$ and $\mathbf{H}$ may be ill-conditioned. A Monte Carlo experiment was carried out, in which we averaged over 100 independent trials. The results were computed using Matlab 6.1.

We compared the performance of (1) SD-QZ with tolerance $\varepsilon_{\text{SD-QZ}} = 10^{-1}$, (2) SD-ALS with tolerance $\varepsilon_{\text{SD-ALS}} = 10^{-1}$, (3) DALS with tolerance $\varepsilon_{\text{DALS}} = 10^{-7}$, starting from three random initial values, and (4) the minimum mean square error (MMSE) filter. We did not weight the matrices $\mathbf{B}_r$ because the associated singular values $\sigma_{B,r}$ were typically of the same order of magnitude. To save computations, DALS was applied to the $(I \times J \times IJ)$ core of the Tucker model of the data tensor [22,23], and the solution was backtransformed to the original dimensions. The results are presented in Figs. 4–6.

In Fig. 4 the median symbol error rate (SER) is plotted vs the signal-to-noise ratio (SNR). The three algorithms yield similar curves, which are quite close to the non-blind MMSE reference. Actually,

Author's personal copy

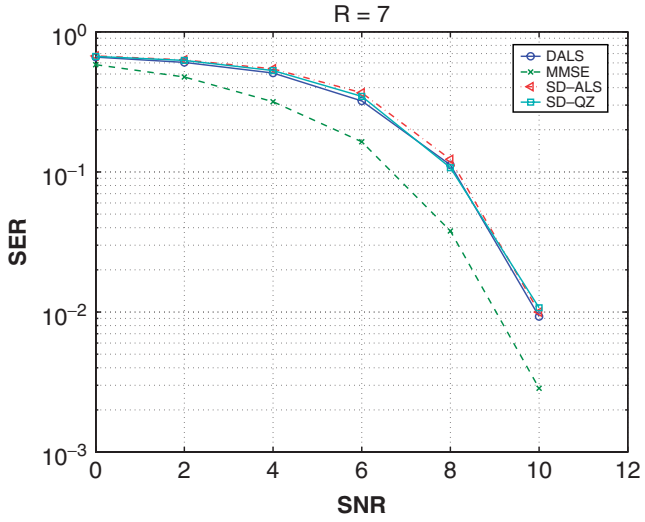


Fig. 4. Median of the SER vs SNR in the first simulation ($I = J = 4$, $K = 200$, $R = 7$).

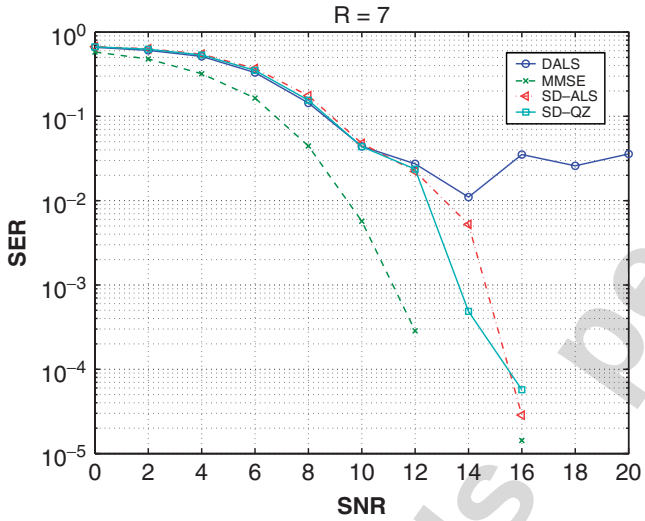


Fig. 5. Mean of the SER vs SNR in the first simulation ($I = J = 4$, $K = 200$, $R = 7$).

the tolerance for DALS has to be taken as small as 10^{-7}; increasing the tolerance increases the median SER. In Fig. 5 we plot the mean SER. The difference between the median and the mean DALS curve comes from the fact that DALS, for SNR $\geqslant 10$ dB, has not yet converged or has not found the global optimum in about 10% of the trials. This percentage also increases when $\varepsilon_{\text{DALS}}$ is increased. In Fig. 6 we plot the mean computation time. For DALS we plotted the time needed by the ALS iteration starting for the best initial value. Hence the global computation time, taking into account the three initializations, is at least a factor 3 higher. DALS is experiencing problems with the fact that

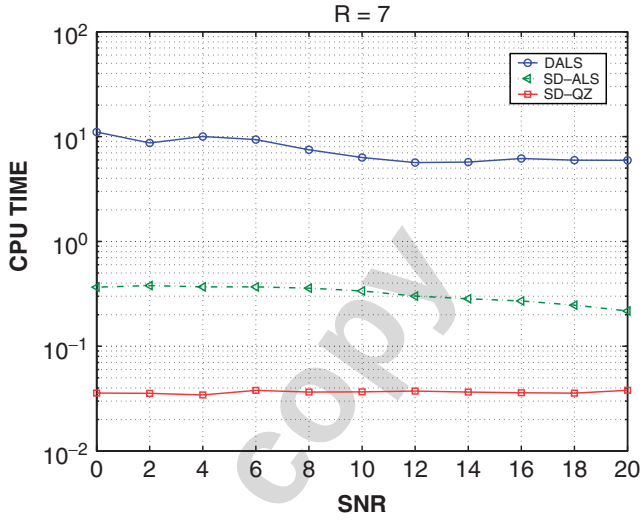


Fig. 6. Average computation time vs SNR in the first simulation ($I = J = 4$, $K = 200$, $R = 7$).

the number of users is high, compared to the number of antennas and the spreading gain. Similar results have been obtained for other choices of the parameters, in which $R > \max(I, J)$. However, if

$$R \leqslant \max(\min(I, J), \min(I, K), \min(J, K)),$$

then DALS can also be initialized by means of an EVD [4,18], which considerably improves the convergence. As a matter of fact, the original PARAFAC model (5) can be directly interpreted as a simultaneous matrix decomposition in this case, and DALS is just one of the techniques to compute the factors [19]. It then depends on the data which particular technique is to be preferred, as illustrated by some numerical experiments in [19].

We conclude that, for a relatively high number of users, SD-ALS and SD-QZ are more reliable and computationally much less demanding than DALS. In general, SD-QZ is more efficient than SD-ALS.

A second simulation illustrates the performance of the techniques developed in Section 6. We consider the transmission of only $K = 50$ QPSK-symbols by $R = 6$ simultaneous users in a system with $I = 4$ antennas and spreading gain $J = 4$. The entries of $\mathbf{A}$ and $\mathbf{H}$ are again drawn from a Gaussian zero-mean unit-variance distribution. The Monte Carlo experiment again consisted of 100 independent trials.

We compared the performance of (1) SD-QZ with tolerance $\varepsilon_{\text{SD-QZ}} = 10^{-1}$, imposing the PARAFAC structure constraint, (2) ACMA, imposing the CM constraint [11], (3) CSD-ALS with tolerance $\varepsilon_{\text{CSD-ALS}} = 10^{-1}$, imposing both constraints and (4)

Author's personal copy

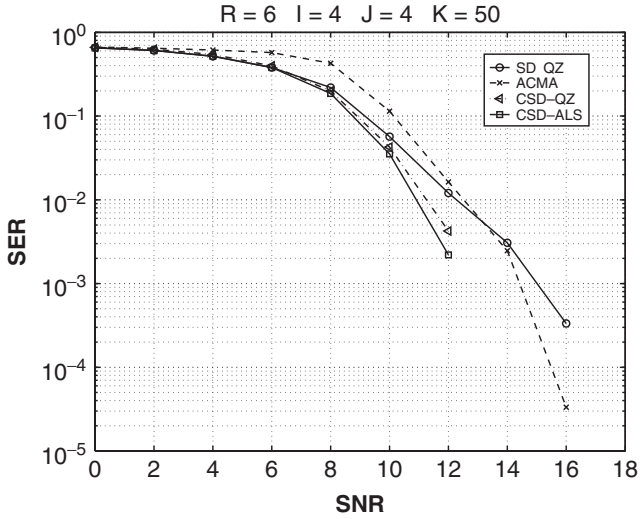


Fig. 7. Mean of the SER vs SNR in the second simulation ($I = J = 4$, $K = 50$, $R = 6$).

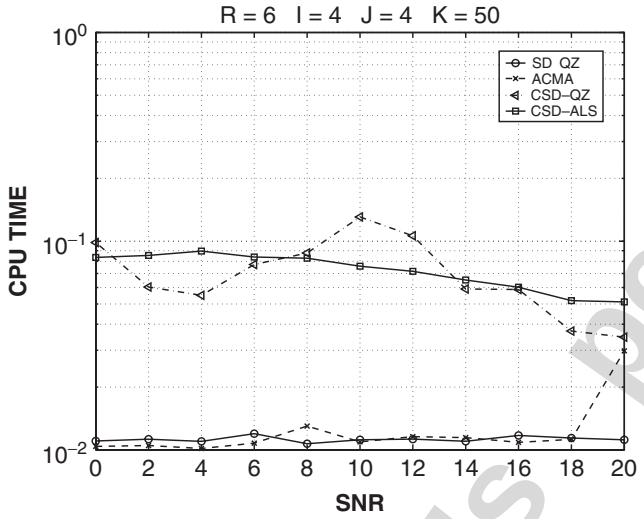


Fig. 8. Average computation time vs SNR in the second simulation ($I = J = 4$, $K = 50$, $R = 6$).

CSD-QZ, with tolerance $\varepsilon_{\text{CSD-QZ}} = 10^{-1}$, also imposing both constraints. The matrices $\mathbf{B}_r$ and $\mathbf{C}_r$ were scaled with $\sigma_{B,r}^{-1}$ and $\sigma_{C,r}^{-1}$, respectively. The results are presented in Figs. 7 and 8. In Fig. 7 the mean SER is plotted vs the SNR. In Fig. 8 we plot the mean computation time. We conclude that combining the two constraints indeed decreases the SER at a moderate additional computational cost.

8. Conclusion

In this paper we presented a new tensor-based mechanism for the blind separation of DS–CDMA signals impinging on an antenna array. The approach led to a new upper bound on the number of users that is quadratically more relaxed than the Kruskal bound. The number of active users can be estimated as the rank of the data matrix. We obtained an explicit expression for the noise-free receiver in terms of an EVD. In the presence of noise the signals can be separated by means of a simultaneous matrix decomposition. The new technique is, for a relatively high number of users, more accurate and computationally less demanding than the ALS algorithm for the computation of the blind receiver.

In a second part of the paper the new algorithm was combined with ACMA. We derived both a computation scheme based on a generalized ALS iteration as a scheme based on a generalized QZ iteration. Jointly exploiting the CDMA structure and the CM property, in a weighted manner, increases the robustness. Moreover, when the individual ACMA and CDMA receiver yield comparable results, the combined receiver is more accurate.

The principles described in this paper are also useful for other telecommunication applications in which PARAFAC plays a role, e.g. [24,25]. In [24] PARAFAC was used to solve the (multi-parameter) direction of arrival problem. In this context, array invariance leads to a third tensor dimension in much the same way as spreading does for CDMA. In [25] PARAFAC was used for the development of a "Khatri–Rao" space–time coding technique for wireless communication systems employing multiple transmit and receive antennas. Current research includes generalizing our results to long code DS–CDMA.

Author's personal copy

Acknowledegments

Part of this research was supported by the Research Council K.U.Leuven (GOA-AMBioRICS, CoE EF/05/006 Optimization in Engineering), the Flemish Government (F.W.O. Project G.0321.06, F.W.O. Research Communities ICCoS and ANMMM, Tournesol) and the Belgian Federal Government (IUAP V-22). The scientific responsibility is assumed by the authors.

References

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

[2] R.A. Harshman, Foundations of the PARAFAC procedure: model and conditions for an "explanatory" multi-mode factor analysis, UCLA Working Papers in Phonetics, vol. 16, 1970, pp. 1–84.
[3] R.A. Harshman, M.E. Lundy, The PARAFAC model for three-way factor analysis and multidimensional scaling, in: H.G. Law, C.W. Snyder, J.A. Hattie, R.P. McDonald (Eds.), Research Methods for Multimode Data Analysis, Praeger, NewYork, 1984, pp. 122–215.
[4] A. Smilde, R. Bro, P. Geladi, Multi-way Analysis. Applications in the Chemical Sciences, Wiley, Chichester, UK, 2004.
[5] J.B. Kruskal, Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics, Linear Algebra Appl. 18 (1977) 95–138.
[6] L. De Lathauwer, A link between the canonical decomposition in multilinear algebra and simultaneous matrix diagonalization, SIAM J. Matrix Anal. Appl. 28 (2006) 642–666.
[7] A. Belouchrani, K. Abed-Meraim, J.-F. Cardoso, E. Moulines, A blind source separation technique using second-order statistics, IEEE Trans. Signal Process. 45 (1997) 434–444.
[8] J.-F. Cardoso, A. Souloumiac, Blind beamforming for non-Gaussian signals, IEEE Proc.-F 140 (1994) 362–370.
[9] M. Haardt, J.A. Nossek, Simultaneous Schur decomposition of several non-symmetric matrices to achieve automatic pairing in multidimensional harmonic retrieval problems, IEEE Trans. Signal Process. 46 (1998) 161–169.
[10] D.-T. Pham, J.-F. Cardoso, Blind separation of instantaneous mixtures of non-stationary sources, IEEE Trans. Signal Process. 49 (2001) 1837–1848.
[11] A.J. van der Veen, A. Paulraj, An analytical constant modulus algorithm, IEEE Trans. Signal Process. 44 (5) (1996) 1136–1155.
[12] A. Yeredor, Non-orthogonal joint diagonalization in the least-squares sense with application in blind source separation, IEEE Trans. Signal Process. 50 (2002) 1545–1553.
[13] A.J. van der Veen, Algebraic methods for deterministic blind beamforming, Proc. IEEE 86 (10) (1998) 1987–2008.
[14] J. Carroll, J. Chang, Analysis of individual differences in multidimensional scaling via an *N*-way generalization of "Eckhardt–Young" decomposition, Psychometrika 9 (1970) 267–283.
[15] N. Sidiropoulos, R. Bro, On the uniqueness of multilinear decomposition of *N*-way arrays, J. Chemometrics 14 (2000) 2377–2388.
[16] J.-F. Cardoso, Super-symmetric decomposition of the fourth-order cumulant tensor, Proceedings of the IEEE Conference on Acoustics, Speech, and Signal Processing (ICASSP-91), vol. 5, Toronto, Canada, 1991, pp. 3109–3112.
[17] T. Jiang, N.D. Sidiropoulos, Kruskal's permutation lemma and the identification of CANDECOMP/PARAFAC and bilinear models with constant modulus constraints, IEEE Trans. Signal Process. 52 (9) (2004) 2625–2636.
[18] S.E. Leurgans, R.T. Ross, R.B. Abel, A decomposition for three-way arrays, SIAM J. Matrix Anal. Appl. 14 (1993) 1064–1083.
[19] L. De Lathauwer, B. De Moor, J. Vandewalle, Computation of the canonical decomposition by means of a simultaneous generalized Schur decomposition, SIAM J. Matrix Anal. Appl. 26 (2) (2004) 295–327.
[20] G.H. Golub, C.F. Van Loan, Matrix Computations, third ed., Johns Hopkins University Press, Baltimore, MD, 1996.
[21] A.J. van der Veen, Statistical performance analysis of the algebraic constant modulus algorithm, IEEE Trans. Signal Process. 50 (12) (2003) 3083–3097.
[22] L. De Lathauwer, B. De Moor, J. Vandewalle, A multilinear singular value decomposition, SIAM J. Matrix Anal. Appl. 21 (2000) 1253–1278.
[23] L.R. Tucker, The extension of factor analysis to three-dimensional matrices, in: H. Gulliksen, N. Frederiksen (Eds.), Contributions to Mathematical Psychology Holt, Rinehart and Winston, New York, 1964, pp. 109–127.
[24] N. Sidiropoulos, R. Bro, G. Giannakis, Parallel factor analysis in sensor array processing, IEEE Trans. Signal Process. 48 (8) (2000) 2377–2388.
[25] N. Sidiropoulos, R. Budampati, Khatri–Rao space–time codes, IEEE Trans. Signal Process. 50 (10) (2002) 2396–2407.

Author's personal copy