LINE SEARCH COMPUTATION OF THE BLOCK FACTOR MODEL FOR BLIND MULTI-USER ACCESS IN WIRELESS COMMUNICATIONS

Dimitri Nion, Lieven De Lathauwer

ETIS,UMR 8051 (CNRS, ENSEA, UCP)
6, avenue du Ponceau, BP 44, F 95014 Cergy-Pontoise Cedex, France
email: nion@ensea.fr, delathau@ensea.fr

ABSTRACT

In this paper, we present a technique for the blind separation of DS-CDMA signals received on an antenna array, for a multi-path propagation scenario that generates Inter-Symbol-Interference. Our method relies on a new third-order tensor decomposition, which is a generalization of the parallel factor model. We start from the observation that the temporal, spatial and spectral diversities give a third-order tensor structure to the received data. This tensor is then decomposed in a sum of contributions, where each contribution fully characterizes one user. We also present a Line Search scheme that greatly improves the convergence speed of the alternating least squares algorithm previously used.

1. INTRODUCTION

Let us consider R users transmitting frames of J symbols at the same time within the same bandwidth towards an array of K antennas. We denote by I the spreading factor, i.e., the CDMA code of each user is a vector of length I. In a direct-path only propagation scenario, the assumption that the channel is noiseless / memoryless leads to the following data model:

$$y_{ijk} = \sum_{r=1}^{R} h_{ir} s_{jr} a_{kr}, \qquad (1)$$

where y_{ijk} is the output of the kth antenna for chip i and symbol j. The scalar a_{kr} is the gain between user r and antenna element k, s_{jr} is the jth symbol transmitted by user r and h_{ir}, for varying i and fixed r contains the spreading sequence of user r. Note that this model can also include memory effects, provided that a discard prefix or guard chips are employed to avoid Inter-Symbol-Interference (ISI) [1]. For background material on algebraic solutions to this problem, we refer to [2]. In this article, we focus on the more complex situation where multi-path propagation leads to ISI. We also assume that the reflections can both occur in the far and close fields of the antenna array so that each path is characterized by its own delay τ_p, angle of arrival θ_p and attenuation α_p, where

p denotes the path index. Under these assumptions, our objective is to estimate the symbols transmitted by every user in a blind way, without using prior knowledge on the propagation parameters or the spreading codes. Our approach consists of collecting the received data in a third-order tensor and to express this tensor as a sum of R contributions by means of a new tensor decomposition: the Block Factor Model introduced in [3, 4].

In section 2, we introduce some multilinear algebra prerequisites. In section 3, we discuss the PARAFAC decomposition, which has been used to implement a blind receiver for the model of equation (1) [1]. In section 4, we discuss the Block Factor Model, which is a generalization of the PARAFAC model. In section 5, we propose a new Line Search scheme which greatly improves the performance of the Alternating Least Squares algorithm derived in [3].

2. MULTILINEAR ALGEBRA PREREQUISITES

A multi-way array of which the elements are addressed by N indices is an Nth-order tensor. Signal processing based on multilinear algebra is discussed in [5].

Definition 1. (Mode-n product) *The mode-1 product of a third-order tensor $\mathcal{Y} \in \mathbb{C}^{L \times M \times N}$ by a matrix $\mathbf{A} \in \mathbb{C}^{I \times L}$, denoted by $\mathcal{Y} \times_1 \mathbf{A}$, is an $(I \times M \times N)$-tensor with elements defined, for all index values, by*

$$(\mathcal{Y} \times_1 \mathbf{A})_{imn} = \sum_{l=1}^{L} y_{lmn} a_{il}$$

Similarly, the mode-2 product by a matrix $\mathbf{B} \in \mathbb{C}^{J \times M}$ and the mode-3 product by $\mathbf{C} \in \mathbb{C}^{K \times N}$ are the $(L \times J \times N)$ and $(L \times M \times K)$ tensors respectively, with elements defined by

$$(\mathcal{Y} \times_2 \mathbf{B})_{ljn} = \sum_{m=1}^{M} y_{lmn} b_{jm}$$

$$(\mathcal{Y} \times_3 \mathbf{C})_{lmk} = \sum_{n=1}^{N} y_{lmn} c_{kn}$$

In this notation, the matrix product $\mathbf{Y} = \mathbf{U}.\mathbf{S}.\mathbf{V}^T$ takes the form of $\mathbf{Y} = \mathbf{S} \times_1 \mathbf{U} \times_2 \mathbf{V}$.

Definition 2. (Rank-1 Tensor) *The third-order tensor $\mathcal{Y} \in \mathbb{R}^{I \times J \times K}$ is rank-1 if its elements can be written as $y_{ijk} = \mathbf{a}(i)\mathbf{b}(j)\mathbf{c}(k)$, where $\mathbf{a} \in \mathbb{C}^{I \times 1}$, $\mathbf{b} \in \mathbb{C}^{J \times 1}$ and $\mathbf{c} \in \mathbb{C}^{K \times 1}$.* This definition generalizes the definition of a rank-1 matrix: $\mathbf{A} \in \mathbb{C}^{I \times J}$ has rank 1 if $\mathbf{A} = \mathbf{a}.\mathbf{b}^T$.

Definition 3. (Tensor Rank) *The rank of $\mathcal{Y}$ is defined as the minimum number of rank-1 tensors that yield $\mathcal{Y}$ in a linear combination.*

Definition 4. (Frobenius Norm) *The Frobenius Norm of the tensor $\mathcal{Y} \in \mathbb{R}^{I \times J \times K}$ is defined by*

$$\|\mathcal{Y}\|_2 = \sqrt{\sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K}|y_{ijk}|^2}.$$

Definition 5. (Khatri-Rao Product) *The Khatri-Rao product of two matrices $\mathbf{A} \in \mathbb{C}^{I \times R}$ and $\mathbf{B} \in \mathbb{C}^{J \times R}$ that have the same number of columns, denoted by $\mathbf{A} \odot \mathbf{B}$, is an $(IJ \times R)$-matrix with elements defined, for all index values, by*

$$(\mathbf{A} \odot \mathbf{B})_{(i-1)J+j,r} = a_{i,r}b_{j,r}.$$

The Khatri-Rao product is also referred to as to column-wise Kronecker product: $(\mathbf{A} \odot \mathbf{B}) = [\mathbf{a}_1 \otimes \mathbf{b}_1 \ldots \mathbf{a}_R \otimes \mathbf{b}_R]$, where $\mathbf{a}_r$ and $\mathbf{b}_r$ denote the rth column of $\mathbf{A}$ and $\mathbf{B}$ respectively and $\otimes$ the Kronecker product.

Definition 6. (Partition-wise Kronecker product) *The partition-wise Kronecker product of two matrices $\mathbf{A} \in \mathbb{C}^{I \times RM}$ and $\mathbf{B} \in \mathbb{C}^{J \times RN}$, consisting of R submatrices $\mathbf{A}_r$ and $\mathbf{B}_r$ of size $I \times M$ and $J \times N$ respectively, denoted $\mathbf{A} \odot_R \mathbf{B}$, is an $(IJ \times RMN)$ matrix defined by*

$$\mathbf{A} \odot_R \mathbf{B} = [\mathbf{A}_1 \otimes \mathbf{B}_1 \ldots \mathbf{A}_R \otimes \mathbf{B}_R].$$

3. PARAFAC DECOMPOSITION

Parallel Factor Analysis (PARAFAC) was introduced by Harshman in [6]. It is a powerful technique to decompose a rank-R tensor in a linear combination of R rank-1 tensors. Let $\mathcal{Y}$ be an $(I \times J \times K)$ tensor, with elements denoted by y_{ijk}. The PARAFAC decomposition of $\mathcal{Y}$ can be written as

$$y_{ijk} = \sum_{r=1}^{R} \mathbf{a}_r(i)\mathbf{b}_r(j)\mathbf{c}_r(k), \qquad (2)$$

where $\mathbf{a}_r$, $\mathbf{b}_r$, $\mathbf{c}_r$ are the rth columns of matrices $\mathbf{A} \in \mathbb{C}^{I \times R}$, $\mathbf{B} \in \mathbb{C}^{J \times R}$ and $\mathbf{C} \in \mathbb{C}^{K \times R}$ respectively, and where i, j and k denote the row index. It now appears that the model for a memoryless channel (1) can be seen as a PARAFAC decomposition of the observation tensor $\mathcal{Y}$. Sidiropoulos *et al.* were the first to use this multilinear algebra technique in the context of wireless communications [1]. The algorithm commonly used to calculate the PARAFAC decomposition is an Alternating Least Squares (ALS) algorithm. Given only $\mathcal{Y}$,

it consists of alternating conditional updates of the unknown matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$. Though easy to implement, the convergence of this algorithm is occasionally slow.

It was noticed through simulations that, when the convergence is slow, $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ are gradually incremented along fixed directions. In [7], Line Search (LS) was proposed to speed up the convergence of ALS. The new values of $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ are sought on the line through their current estimates and the ALS updates. In [7], the step size is heuristic. In [8], the optimal step size is determined and the new method is called "Enhanced Line Search" (ELS). These methods were proposed for real data that fit the PARAFAC model. In this paper we will generalize the results to Block Factor Model. Since in the CDMA context the data are complex, we will look for the optimal step in $\mathbb{C}$.

4. BLOCK FACTOR MODEL

4.1. Data Model: Analytic Form

For the propagation scenario that takes into account multipath and ISI, a more general algebraic model has been introduced in [3], and is referred to as Block Factor Model (BFM). Let us start with a single source transmitting J symbols along P paths towards K antennas. These paths can be considered as channels with memory, leading to ISI, and are assumed to be stationary over J symbols. Let L be the maximum channel length at the symbol rate, meaning that interference is occurring over maximally L symbols. The coefficients resulting from the convolution between the channel impulse response for the pth path and the spreading sequence of the user under consideration are collected in a vector h_p of size LI. So $h_p(i + (l-1)I)$ is the coefficient of the overall impulse response corresponding to the ith chip and the lth symbol. We denote by $x_p(i,j)$ the ith chip of the signal received from the pth path during the jth symbol period. We have:

$$x_p(i,j) = \sum_{l=1}^{L} h_p(i + (l-1)I)\, s_{j-l+1}. \qquad (3)$$

Let $a_k(\theta_p)$ be the response of the kth antenna to the signal coming from the pth path with an angle of arrival θ_p, where we assume that the path loss is combined with the antenna gain. The model defined in (3) then yields:

$$x_p(i,j,k) = a_k(\theta_p) \sum_{l=1}^{L} h_p(i + (l-1)I)\, s_{j-l+1}, \quad (4)$$

where $x_p(i,j,k)$ denotes the ith chip of the jth symbol of the signal received by the kth antenna. We now write the overall received signal by summing the contributions of the P paths and the R users:

$$y_{ijk} = \sum_{r=1}^{R}\sum_{p=1}^{P} a_k(\theta_{rp}) \sum_{l=1}^{L} h_{rp}(i + (l-1)I)\, s_{j-l+1}^{(r)}, \quad (5)$$
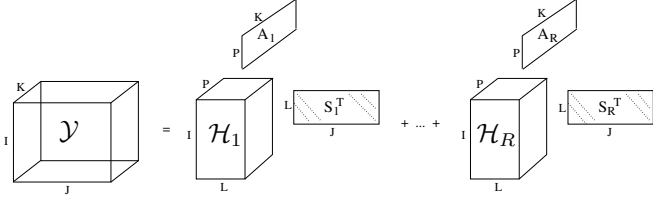
Fig. 1. Schematic representation of the BFM

where y_{ijk} denotes the ith chip of the jth symbol of the signal received by the kth antenna, and in which r, p and l are the user, path and interfering symbol index respectively.

4.2. Data Model: Algebraic Form

We have established in [3] that, algebraically, (5) can be expressed as:

$$\mathcal{Y} = \sum_{r=1}^{R} \mathcal{H}_r \times_2 \mathbf{S}_r \times_3 \mathbf{A}_r. \tag{6}$$

This BFM is represented in Figure 1. Each term of the sum in (6) contains the information related to one particular user. The global channel is characterized by the tensor $\mathcal{H}_r \in \mathbb{C}^{I \times L \times P}$, where each slice $\mathcal{H}_r(:,:,p)$ collects $I \times L$ samples of the vector resulting from the convolution between the spreading sequence of the rth user and the overall impulse response of the channel corresponding to the pth path. The antenna array response is given by $\mathbf{A}_r \in \mathbb{C}^{K \times P}$, where each column-vector represents the response of the K antennas to the pth path. The J transmitted symbols are collected in a matrix $\mathbf{S}_r$, which has a Toeplitz structure.

The BFM defined in (6) is intrinsically indeterminate as follows:

$$\mathcal{Y} = \sum_{r=1}^{R} (\alpha_r \mathcal{H}_r \times_3 \mathbf{U}_r) \times_2 (\alpha_r^{-1} \mathbf{S}_r) \times_3 (\mathbf{U}_r^{-1} \mathbf{A}_r), \quad (7)$$

where the scalar α_r and the non-singular matrix $\mathbf{U}_r$ represent the indeterminacy in modes two and three respectively. Note that the indeterminacy in the second mode involves a scalar rather than a matrix due to the Toeplitz structure of $\mathbf{S}_r$.

4.3. Uniqueness of the BFM

If the BFM (6) is unique (up to the trivial indeterminacies), then its computation allows for the separation of the different user signals and the estimation of the transmitted sequences. We call a property generic when it holds everywhere, except for a set of Lebesgue measure 0. A generic condition for uniqueness has been derived in [4]:

$$min\left(\left\lfloor \frac{J}{L} \right\rfloor, R\right) + min\left(\left\lfloor \frac{K}{P} \right\rfloor, R\right) + min\left(\left\lfloor \frac{I}{max(L,P)} \right\rfloor, R\right) \geq 2R+2, \tag{8}$$

if $I > L + P - 2$. If $I \leq L + P - 2$, then some additional conditions apply. This result implies an upper bound on the number of users that can be allowed at the same time. The maximal number of simultaneous users correspond to the maximal value R that satisfies (8).

4.4. ALS Computation of the BFM

Given only $\mathcal{Y}$, we want to estimate $\mathcal{H}_r$, $\mathbf{S}_r$, and $\mathbf{A}_r$ for each user. We denote by $\mathbf{A}$ and $\mathbf{S}$ the $K \times RP$ and $J \times RL$ matrices that result from the concatenation of the R matrices $\mathbf{A}_r$ and $\mathbf{S}_r$ respectively. Let $\mathbf{H}$ be an $RLP \times I$ matrix in which the entries of the tensors $\mathcal{H}_r$ are stacked in the following way: $[\mathbf{H}]_{(r-1)LP+(l-1)P+p,i} = \mathcal{H}_r(i,l,p)$. We define by $\mathbf{Y}^{(JK \times I)}$ the $JK \times I$ matrix representation of $\mathcal{Y}$, obtained from $\left[\mathbf{Y}^{(JK \times I)}\right]_{(j-1)J+k,i} = y_{ijk}$. This matrix can be considered as the result of row-wise concatenation of the J transposed left-right slices of $\mathcal{Y}$. Note the order in which the entries are stacked, with the left index (j here) varying more slowly than the right one. We denote by $\mathcal{Y}^{(n)}$ an estimation of $\mathcal{Y}$ at the nth iteration, built from the updated factors $\mathbf{A}^{(n)}$, $\mathbf{S}^{(n)}$ and $\mathbf{H}^{(n)}$. The ALS algorithm derived in [3] exploits the multilinearity of model (6) to alternate between conditional least-squares updates of the unknowns $\mathbf{A}$, $\mathbf{S}$ and $\mathbf{H}$ in each iteration. The cost function that is minimized, is given by:

$$\begin{aligned} \phi_{ALS} &= \left\| \mathcal{Y} - \mathcal{Y}^{(n)} \right\|_2^2 \\ &= \left\| \mathbf{Y}^{(JK \times I)} - (\mathbf{S}^{(n)} \odot_R \mathbf{A}^{(n)})\mathbf{H}^{(n)} \right\|_2^2. \end{aligned} \tag{9}$$

Explicit expressions for $\mathbf{A}^{(n)}$, $\mathbf{S}^{(n)}$ and $\mathbf{H}^{(n)}$ are given in [3].

5. NEW COMPUTATION SCHEME FOR THE BFM

5.1. Line Search procedure

Though easy to compute, the ALS algorithm can be slow. In particular, it is sensitive to swamps (i.e., several iterations with convergence speed almost null after which convergence resumes). In this section, we propose a new Line Search computation scheme that improves the way the unknowns are updated.

The Line Search procedure consists of the prediction of the unknown factors a certain number of iterations ahead from the following linear regression:

$$\begin{cases} \mathbf{A}^{(new)} &= \mathbf{A}^{(n-2)} + \rho_A \left(\mathbf{A}^{(n-1)} - \mathbf{A}^{(n-2)}\right) \\ \mathbf{S}^{(new)} &= \mathbf{S}^{(n-2)} + \rho_S \left(\mathbf{S}^{(n-1)} - \mathbf{S}^{(n-2)}\right) \\ \mathbf{H}^{(new)} &= \mathbf{H}^{(n-2)} + \rho_H \left(\mathbf{H}^{(n-1)} - \mathbf{H}^{(n-2)}\right) \end{cases}, \tag{10}$$

where $\mathbf{A}^{(n-1)}$, $\mathbf{S}^{(n-1)}$ and $\mathbf{H}^{(n-1)}$ are the estimates of $\mathbf{A}$, $\mathbf{S}$ and $\mathbf{H}$ respectively, obtained in the $(n-1)^{th}$ ALS iteration. The matrices $\mathbf{G}_A^{(n)} = \left(\mathbf{A}^{(n-1)} - \mathbf{A}^{(n-2)}\right)$, $\mathbf{G}_S^{(n)} = \left(\mathbf{S}^{(n-1)} - \mathbf{S}^{(n-2)}\right)$, and $\mathbf{G}_H^{(n)} = \left(\mathbf{H}^{(n-1)} - \mathbf{H}^{(n-2)}\right)$ represent the search directions in the nth iteration and ρ_A, ρ_S and

ρ_H are the relaxation factors, i.e., the step size in the search directions. The matrices $\mathbf{A}^{(new)}$, $\mathbf{S}^{(new)}$ and $\mathbf{H}^{(new)}$ are then used to start the nth iteration of the ALS.

It is in principle possible to consider different relaxation factors for the different modes, but this makes the computation more expensive. In this article, we consider the same relaxation factor for the three modes: $\rho_A = \rho_S = \rho_H = \rho$.

The optimal relaxation factor is found by the minimization of:

$$
\begin{aligned}
\phi_{ELS}^{(n)} &= \left\| (\mathbf{S}^{(new)} \odot_R \mathbf{A}^{(new)}) \mathbf{H}^{(new)} - \mathbf{Y}^{(JK \times I)} \right\|_2^2 \\
&= \left\| \left((\mathbf{S}^{(n-2)} + \rho \mathbf{G}_S^{(n)}) \odot_R (\mathbf{A}^{(n-2)} + \rho \mathbf{G}_A^{(n)}) \right) \right. \\
&\quad \left. \left(\mathbf{H}^{(n-2)} + \rho \mathbf{G}_H^{(n)} \right) - \mathbf{Y}^{(JK \times I)} \right\|_2^2.
\end{aligned}
\tag{11}
$$

This equation can be written as follows:

$$
\phi_{ELS}^{(n)} = \left\| \rho^3 \mathbf{T}_3 + \rho^2 \mathbf{T}_2 + \rho \mathbf{T}_1 + \mathbf{T}_0 \right\|_2^2,
\tag{12}
$$

in which the $JK \times I$ matrices $\mathbf{T}_3$, $\mathbf{T}_2$, $\mathbf{T}_1$ and $\mathbf{T}_0$ are defined as:

$$
\left\{
\begin{array}{rl}
\mathbf{T}_3 =& (\mathbf{G}_S \odot_R \mathbf{G}_A) \mathbf{G}_H \\
\mathbf{T}_2 =& (\mathbf{G}_S \odot_R \mathbf{G}_A) \mathbf{H} + (\mathbf{S} \odot_R \mathbf{G}_A + \mathbf{G}_S \odot_R \mathbf{A}) \mathbf{G}_H \\
\mathbf{T}_1 =& (\mathbf{S} \odot_R \mathbf{A}) \mathbf{G}_H + (\mathbf{S} \odot_R \mathbf{G}_A + \mathbf{G}_S \odot_R \mathbf{A}) \mathbf{H} \\
\mathbf{T}_0 =& (\mathbf{S} \odot_R \mathbf{A}) \mathbf{H} - \mathbf{Y}^{(JK \times I)}
\end{array}
\right. ,
$$

where the superscripts n and $n-2$ have been omitted for convenience of notation. Denote by *Vec* the operator that writes a matrix $\mathbf{A} \in \mathbb{C}^{I \times J}$ in vector format by concatenation of the columns such that $\mathbf{A}(i,j) = [Vec(\mathbf{A})]_{i+(j-1)I}$.

Eq. (12) is then equivalent to:

$$
\phi_{ELS}^{(n)} = \| \mathbf{T} \cdot \mathbf{u} \|_2^2 = \mathbf{u}^H \cdot \mathbf{T}^H \cdot \mathbf{T} \cdot \mathbf{u},
\tag{13}
$$

where the matrix $\mathbf{T} = [Vec(\mathbf{T}_3)|Vec(\mathbf{T}_2)|Vec(\mathbf{T}_1)|Vec(\mathbf{T}_0)]$ of size $IJK \times 4$ is obtained by column-wise concatenation of the vector representation of $\mathbf{T}_3$, $\mathbf{T}_2$, $\mathbf{T}_1$ and $\mathbf{T}_0$ respectively, where $\mathbf{u} = [\rho^3, \rho^2, \rho, 1]^T$ is a vector of size 4×1 and $.^H$ denotes the Hermitian transpose. The 4×4 matrix $\Delta = \mathbf{T}^H \cdot \mathbf{T}$ has complex elements defined by $[\Delta]_{m,n} = \alpha_{m,n} + j\beta_{m,n}$. Since Δ is Hermitian, we have: $\alpha_{m,n} = \alpha_{n,m}$, $\beta_{m,n} = -\beta_{n,m}$ and $\beta_{m,m} = 0$.

For real-valued data, (13) reduces to a polynomial of degree 6 w.r.t. the real variable ρ and can thus easily be minimized. The case of complex-valued data is more difficult. We write the relaxation factor as $\rho = r.e^{i\theta}$, where r is the modulus of ρ and θ its argument, and propose an iterative scheme that minimizes $\phi_{ELS}^{(n)}$ alternately w.r.t. r and θ. The complexity of the sub-steps is low. On the other hand, it is not necessary to compute the minimum of $\phi_{ELS}^{(n)}$ with high precision, as the goal is only to accelerate the ALS algorithm. As a result, for typical data dimensionalities, the cost of estimating ρ is negligible w.r.t. the cost of the ALS sub-step.

Enhanced Line Search Scheme:

1. Partial minimization of $\phi_{ELS}^{(n)}$ w.r.t. r.

The partial derivative of $\phi_{ELS}^{(n)}$ w.r.t. r can be expressed as:

$$
\frac{\delta \phi_{ELS}^{(n)}(r)}{\delta r} = \sum_{p=0}^{5} c_p r^p,
\tag{14}
$$

where the real coefficients c_p are given in Appendix and only depend on θ, $\alpha_{m,n}$ and $\beta_{m,n}$. Given θ, this step consists of finding the real roots of a polynomial of degree 5 and selecting the root that minimizes $\phi_{ELS}^{(n)}(r)$.

2. Partial minimization of $\phi_{ELS}^{(n)}$ w.r.t. θ.

After a change of variable, $t = tan(\frac{\theta}{2})$, the partial derivative of $\phi_{ELS}^{(n)}$ w.r.t. t can be expressed as:

$$
\frac{\delta \phi_{ELS}^{(n)}(t)}{\delta t} = \frac{\sum_{p=0}^{6} d_p t^p}{(1 + t^2)^3},
\tag{15}
$$

where the real coefficients d_p are given in Appendix and only depend on r, $\alpha_{m,n}$ and $\beta_{m,n}$. Given r, this step consists of finding the real roots of a polynomial of degree 6 and selecting the root that minimizes $\phi_{ELS}^{(n)}(t)$.

3. Repeat from step 1 until
$\left\| \phi_{ELS}^{(n)} - \phi_{ELS}^{(n-1)} \right\|_2^2 < \eta$ (e.g. $\eta = 10^{-1}$)

This ELS scheme is inserted in the standard ALS algorithm scheme as follows:

Algorithm 1 *Summary of the ALS+ELS algorithm:*

1- Initialize $\mathbf{A}^{(n-2)}$, $\mathbf{S}^{(n-2)}$, $\mathbf{H}^{(n-2)}$, $\mathbf{G}_A^{(n-2)}$, $\mathbf{G}_S^{(n-2)}$, $\mathbf{G}_H^{(n-2)}$, $n = 2$.
2- <u>ELS Scheme:</u>
- Find the optimal value of ρ from (14) and (15).
- Build $\mathbf{A}^{(new)}$, $\mathbf{S}^{(new)}$ and $\mathbf{H}^{(new)}$ from (10).
3- <u>ALS Steps:</u>
- Find $\mathbf{A}^{(n)}$ from $\mathbf{S}^{(new)}$ and $\mathbf{H}^{(new)}$.
- Find $\mathbf{S}^{(n)}$ from $\mathbf{A}^{(n)}$ and $\mathbf{H}^{(new)}$.
- Find $\mathbf{H}^{(n)}$ from $\mathbf{A}^{(n)}$ and $\mathbf{S}^{(n)}$.
4- Repeat from 2 until $c(n) < \epsilon$ (e.g. $\epsilon = 10^{-5}$),
where $c(n) = \left\| \mathcal{Y}^{(n)} - \mathcal{Y}^{(n-1)} \right\|_2^2$.
- Increase n to $n + 1$

5.2. Results of simulations

In this section, we illustrate the performance of the ALS+ELS algorithm for the calculation of the BFM and we compare with the standard ALS algorithm presented in [3]. We assume the presence of Additive White Gaussian Noise (AWGN) so that the observed tensor is given by $\mathcal{Y}_{obs} = \mathcal{Y} + \mathcal{N}$, where $\mathcal{Y}$ is the tensor that contains the data to be estimated (Eq. 6) and $\mathcal{N}$ contains noise with variable variance. The following simulation shows the result obtained from 1000 Monte-Carlo trials with spreading codes of length $I = 6$, a short frame of $J = 50$ QPSK-symbols, $K = 6$ antennas, $L = 2$ interfering symbols, $P = 2$ paths per user and $R = 4$ users, which means that we are on the uniqueness bound defined in (8).

In Fig 2(a), we show the accuracy of the BFM calculated either by ALS or ALS+ELS in terms of the Bit Error Rate (BER), and we compare to the performance of the
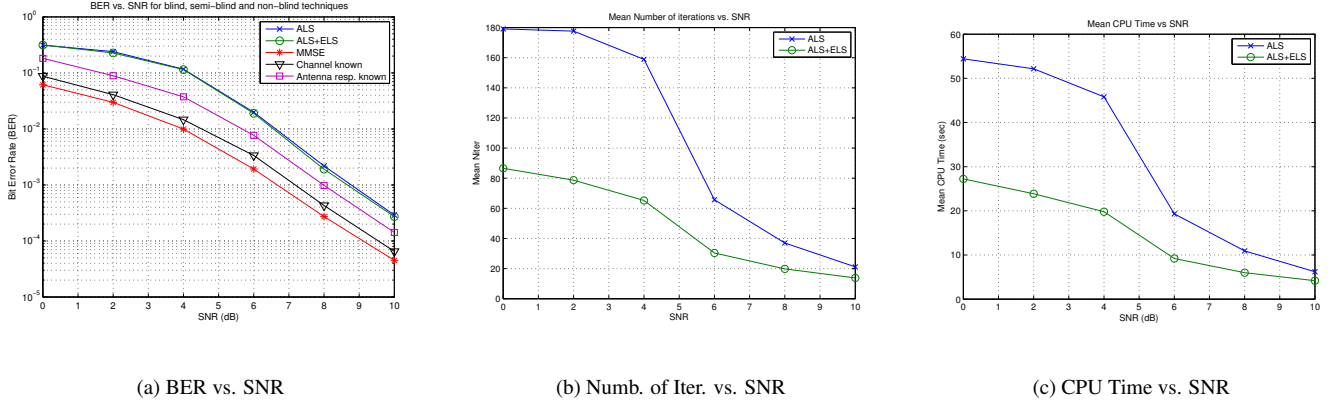
| (a) BER vs. SNR | (b) Numb. of Iter. vs. SNR | (c) CPU Time vs. SNR |

Fig. 2. Performance of ALS and LM in presence of AWGN.

MMSE (Minimum Mean-Square Error) estimator, which assumes perfect knowledge of the channel (tensors $\mathcal{H}_r$ known) and the antenna array response (matrices $\mathbf{A}_r$ known). We also plot the performance of two semi-blind techniques (either $\mathcal{H}_r$ or $\mathbf{A}_r$ known). It turns out that the performance of the blind receiver based on BFM is close to the MMSE (the gap between the two curves reduces for increasing values of SNR). The ALS and ALS+ELS algorithms give the same curve which was expected since these methods both reduce the same cost function.

In Fig 2(b) and 2(c), we compare the mean number of iterations and the mean CPU time required by standard ALS and by ALS+ELS for the 1000 runs. It is clear that the ELS scheme allowed to considerably reduce the number of iterations (e.g. gain of 59 percent for SNR=6dB) and that the cost of computing the step size is negligible w.r.t. the cost of the ALS sub-step (the gain of time is 57 percent for SNR=6dB).

6. CONCLUSION

In this paper, we have shown how Block Factor Analysis of a third-order tensor leads to a powerful blind receiver for multi-user access in wireless communications. The tensor model takes both ISI and multi-path propagation aspects into account, which was not the case for the blind PARAFAC receiver in [1]. The method works for very short data sequences, or, equivalently, for channels that are fast varying and our model can be applied to other systems where three diversities are available. The computation strategy for the calculation of the BFM decomposition is an important issue. It turns out that an ELS scheme greatly improves the convergence speed of the ALS algorithm.

APPENDIX

Coefficients c_p in equation (14):

$$\begin{cases} c_5 &= 6\alpha_{11} \\ c_4 &= 10(\alpha_{12}cos(\theta) + \beta_{12}sin(\theta)) \\ c_3 &= 4(\alpha_{22} + 2\alpha_{13}cos(2\theta) + 2\beta_{13}sin(2\theta)) \\ c_2 &= 6(\alpha_{14}cos(3\theta) + \alpha_{23}cos(\theta) + \beta_{14}sin(3\theta) + \beta_{23}sin(\theta)) \\ c_1 &= 2(\alpha_{33} + 2\alpha_{24}cos(2\theta) + 2\beta_{24}sin(2\theta)) \\ c_0 &= 2\alpha_{34}cos(\theta) + 2\beta_{34}sin(\theta) \end{cases}$$

Coefficients d_p in equation (15):

$$\begin{cases} d_6 &= -2\beta_{12}r^5 + 4\beta_{13}r^4 - 2(\beta_{23} + 3\beta_{14})r^3 + 4\beta_{24}r^2 - 2\beta_{34}r \\ d_5 &= -4\alpha_{12}r^5 + 16\alpha_{13}r^4 - 4(\alpha_{23} + 9\alpha_{14})r^3 + 16\alpha_{24}r^2 - 4\alpha_{34}r \\ d_4 &= -2\beta_{12}r^5 - 20\beta_{13}r^4 - 2(\beta_{23} - 45\beta_{14})r^3 - 20\beta_{24}r^2 - 2\beta_{34}r \\ d_3 &= -8\alpha_{12}r^5 - 8(\alpha_{23} - 15\alpha_{14})r^3 - 8\alpha_{34}r \\ d_2 &= 2\beta_{12}r^5 - 20\beta_{13}r^4 + 2(\beta_{23} - 45\beta_{14})r^3 - 20\beta_{24}r^2 + 2\beta_{34}r \\ d_1 &= -4\alpha_{12}r^5 - 16\alpha_{13}r^4 - 4(\alpha_{23} + 9\alpha_{14})r^3 - 16\alpha_{24}r^2 - 4\alpha_{34}r \\ d_0 &= 2\beta_{12}r^5 + 4\beta_{13}r^4 + 2(\beta_{23} + 3\beta_{14})r^3 + 4\beta_{24}r^2 + 2\beta_{34}r \end{cases}$$

7. REFERENCES

[1] N.D. Sidiropoulos, "Blind parafac receivers for DS-CDMA systems," in *IEEE Trans. Signal Proc.*, vol. 48, pp. 810–823. 2000.

[2] A.-J. van der Veen, "Algebraic methods for deterministic blind beamforming," in *Proc. IEEE*, vol. 86, pp. 1987–2008. 1998.

[3] D. Nion and L. De Lathauwer, "A block factor analysis based receiver for blind multi-user access in wireless communications," in *ICASSP06*, Accepted.

[4] L. De Lathauwer, "Decomposing a tensor in rank-(R_1,R_2,R_3) terms," Tech. Rep., ETIS Lab., Cergy-Pontoise, France, 2006, in preparation.

[5] L. De Lathauwer, *Signal Processing based on Multilinear Algebra*, Ph.D. thesis, Faculty of Engineering, K.U. Leuven, Belgium, 1997.

[6] R. Harshman, "Foundations of the parafac procedure: Model and conditions for an 'explanatory' multi-mode factor analysis," in *UCLA Working Papers in Phonetics*, vol. 16, pp. 1–84. 1970.

[7] R. Bro, *Multi-way Analysis in the Food Industry: Models, Algorithms, and Applications*, Ph.D. thesis, University of Amsterdam, Amsterdam, 1998.

[8] M. Rajih and P. Comon, "Enhanced line search: A novel method to accelerate parafac," in *Eusipco'05*. 2005.