

# LEVENBERG-MARQUARDT 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 with Inter-Symbol-Interference. Our method relies on a new third-order tensor decomposition, which is a generalization of the parallel factor model. We start with 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 an algorithm of the Levenberg-Marquardt type for the calculation of this decomposition. This method is faster than 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:

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

where  $y_{ijk}$  is the output of the  $k^{\text{th}}$  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  $j^{\text{th}}$  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  $\tau_p$ , angle of arrival  $\theta_p$  and attenuation  $\alpha_p$ , 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 Levenberg-Marquardt algorithm to compute the decomposition of the Block Factor Model and we compare its performance to the Alternating Least Squares algorithm used in [3].

## 2. MULTILINEAR ALGEBRA PREREQUISITES

A multi-way array of which the elements are addressed by  $N$  indices is an  $N$ th-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  $\mathbf{Y} = \mathbf{U} \cdot \mathbf{S} \cdot \mathbf{V}^T$  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  $\mathbf{A} = \mathbf{a} \cdot \mathbf{b}^T$ .

**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.*

Part of this research was supported by: (1) the Flemish Government: (a) Research Council K.U.Leuven (GOA-AMBioRICS, CoE EF/05/006 Optimization in Engineering), (b) F.W.O. project G.0321.06, (c) F.W.O. Research Communities ICCoS and ANMMM, (d) Tournesol 2005, (2) the Belgian Federal Science Policy Office: IUAP P5/22.

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

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

### 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

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

where  $\mathbf{a}_r$ ,  $\mathbf{b}_r$ ,  $\mathbf{c}_r$  are the  $r^{\text{th}}$  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. In [7], it has been established that derivative based methods often show better convergence properties than ALS, which has been confirmed by our own experiments on the PARAFAC model.

### 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  $p^{\text{th}}$  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  $i^{\text{th}}$  chip and the  $l^{\text{th}}$  symbol. We denote by  $x_p(i, j)$  the  $i^{\text{th}}$  chip of the signal received from the  $p^{\text{th}}$  path during the  $j^{\text{th}}$  symbol period. We have:

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

Let  $a_k(\theta_p)$  be the response of the  $k^{\text{th}}$  antenna to the signal coming from the  $p^{\text{th}}$  path with an angle of arrival  $\theta_p$ , where

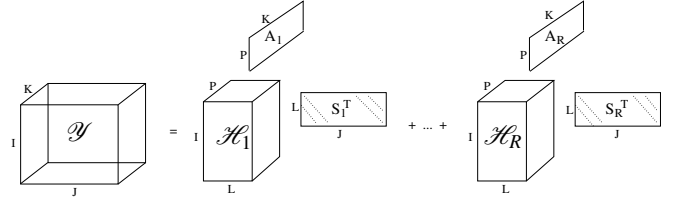


Figure 1: Schematic representation of the BFM

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  $i^{\text{th}}$  chip of the  $j^{\text{th}}$  symbol of the signal received by the  $k^{\text{th}}$  antenna. We now write the overall received signal by summing the contributions of the  $P$  paths and the  $R$  users:

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

where  $y_{ijk}$  denotes the  $i^{\text{th}}$  chip of the  $j^{\text{th}}$  symbol of the signal received by the  $k^{\text{th}}$  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:

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

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 front-back slice  $\mathcal{H}_r(:, :, p)$  collects  $I \times L$  samples of the vector resulting from the convolution between the spreading sequence of the  $r^{\text{th}}$  user and the overall impulse response of the channel corresponding to the  $p^{\text{th}}$  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  $p^{\text{th}}$  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  $\alpha_r$  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]:

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

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).

## 5. NEW COMPUTATION SCHEME FOR THE BFM

### 5.1 Levenberg Marquardt Algorithm

Given only  $\mathcal{Y}$ , we want to estimate  $\mathcal{H}_r$ ,  $\mathbf{S}_r$  and  $\mathbf{A}_r$  for each user. The algorithm developed in [3] was based on an ALS technique which consists of alternating updates of these unknowns at each iteration. However, this algorithm has at most linear convergence and is sensitive to swamps (i.e., several iterations with convergence speed almost null after which convergence resumes), while a Gauss-Newton based method provides, at least in theory, superlinear or even quadratic convergence. In this section, we build a new update strategy for the computation of the BFM, based on the Levenberg-Marquardt (LM) method [8].

Denote by  $\hat{\mathcal{Y}}$  an estimation of the model defined in (6). Each element of this tensor can be written as:

$$\hat{y}_{ijk} = \sum_{r=1}^R \sum_{l=1}^L \sum_{p=1}^P \mathcal{H}_r(i, l, p) \hat{\mathbf{S}}_r(j, l) \hat{\mathbf{A}}_r(k, p), \quad (9)$$

where the  $R$  estimated matrices  $\hat{\mathbf{A}}_r$  and  $\hat{\mathbf{S}}_r$  and the  $R$  estimated tensors  $\mathcal{H}_r$  yield  $\hat{\mathcal{Y}}$ .

We have the following cost function:

$$\begin{aligned} \phi(\mathcal{H}_r, \hat{\mathbf{A}}_r, \hat{\mathbf{S}}_r) &= \frac{1}{2} \sum_{i=1}^I \sum_{j=1}^J \sum_{k=1}^K |y_{ijk} - \hat{y}_{ijk}|^2 \\ &= \frac{1}{2} \sum_{i=1}^I \sum_{j=1}^J \sum_{k=1}^K |r_{ijk}|^2, \end{aligned} \quad (10)$$

where  $r_{ijk}$  are the residuals.

Denote by  $\mathbf{p}$  a vector that contains the elements of the  $R$  matrices  $\hat{\mathbf{A}}_r$ , the  $R$  tensors  $\mathcal{H}_r$  and the  $R$  generator vectors of the Toeplitz matrices  $\hat{\mathbf{S}}_r$ . This vector is of length  $F = R(KP + LLP + (J + L - 1))$ , where  $F$  represents the number of unknowns.

The cost function of equation (10) can now be written as:

$$\phi(\mathbf{p}) = \frac{1}{2} \sum_{m=1}^M |y_m - \hat{y}_m(\mathbf{p})|^2 = \frac{1}{2} \sum_{m=1}^M |r_m(\mathbf{p})|^2 = \frac{1}{2} \mathbf{r}(\mathbf{p})^H \mathbf{r}(\mathbf{p}), \quad (11)$$

where  $m = (i - 1)JK + (j - 1)K + k$ ,  $m = 1, \dots, M = IJK$ , is a super-index that combines  $i, j, k$ , where  $\mathbf{r} = [r_1 \dots r_M]^T$  is the vector of residuals and  $\cdot^H$  denotes the Hermitian transpose. Note the order in which the entries are stacked, with the left index ( $i$  here) varying more slowly than the right one. The problem now consists of minimizing the cost function with respect to the parameter vector  $\mathbf{p}$ , which is a non-linear

least squares problem that can be solved by a Gauss-Newton based method [9]. This method assumes that the residuals in the neighborhood of a point  $\mathbf{p}_0$  can be approximated by a Taylor expansion truncated after the linear term:

$$\mathbf{r}(\mathbf{p}) \cong \mathbf{r}(\mathbf{p}_0) + \mathbf{J}(\mathbf{p}_0)(\mathbf{p} - \mathbf{p}_0) \equiv \tilde{\mathbf{r}}(\Delta\mathbf{p}), \quad (12)$$

where  $\Delta\mathbf{p} = \mathbf{p} - \mathbf{p}_0$  and  $\mathbf{J}(\mathbf{p}_0)$  is the Jacobian matrix of size  $M \times F$  of which the elements  $j_{mf}$  are defined as  $j_{mf} = \frac{\delta r_m(\mathbf{p}_0)}{\delta p_f} = -\frac{\delta y_m(\mathbf{p}_0)}{\delta p_f}$ .

Using this linear approximation of  $\mathbf{r}(\mathbf{p})$ , the cost function (11) can be expressed in terms of  $\Delta\mathbf{p}$ :

$$\tilde{\phi}(\Delta\mathbf{p}) \equiv \frac{1}{2} \tilde{\mathbf{r}}(\Delta\mathbf{p})^H \tilde{\mathbf{r}}(\Delta\mathbf{p}) = \frac{1}{2} \|\mathbf{r}(\mathbf{p}_0) + \mathbf{J}(\mathbf{p}_0)\Delta\mathbf{p}\|_2^2. \quad (13)$$

Given an approximation  $\mathbf{p}^{(n)}$  of the parameter vector at iteration  $n$ , the approximation of  $\mathbf{p}$  at iteration  $(n + 1)$  is then given by  $\mathbf{p}^{(n+1)} = \mathbf{p}^{(n)} + \Delta\mathbf{p}^{(n)}$ .

The correction  $\Delta\mathbf{p}^{(n)}$  that minimizes  $\tilde{\phi}(\Delta\mathbf{p})$  is the solution to the linear least-squares problem:

$$\min_{\{\Delta\mathbf{p}^{(n)}\}} \frac{1}{2} \|\mathbf{r}(\mathbf{p}^{(n)}) + \mathbf{J}(\mathbf{p}^{(n)})\Delta\mathbf{p}^{(n)}\|_2^2.$$

The solution is obtained by solving the system of normal equations:

$$(\mathbf{J}^H \mathbf{J}) \Delta\mathbf{p}^{(n)} = -\mathbf{g}, \quad (14)$$

where  $\mathbf{g}$  is the gradient of  $\phi(\mathbf{p}^{(n)})$ . The matrix  $\mathbf{J}^H \mathbf{J}$  is in fact an approximation of the Hessian matrix [9]. The Gauss-Newton update of equation (14) requires the Jacobian to be full-rank in all steps.

However, the indeterminacies of the BFM (7) lead to rank deficiency of the Jacobian, which has at least  $R(P^2 + 1)$  vanishing singular values. Because of this over-parameterization, the Gauss-Newton update of equation (14) has to be modified. One solution to successfully handle the singularity of  $(\mathbf{J}^H \mathbf{J})$  has been proposed by Levenberg and Marquardt [8] and is also known as ‘‘damped Gauss-Newton Method’’. It consists of updating  $\Delta\mathbf{p}^{(n)}$  from the modified normal equations:

$$\left(\mathbf{J}^H \mathbf{J} + \lambda^{(n)} \mathbf{I}_F\right) \Delta\mathbf{p}^{(n)} = -\mathbf{g}. \quad (15)$$

The damping parameter  $\lambda^{(n)}$  makes the matrix  $\left(\mathbf{J}^H \mathbf{J} + \lambda^{(n)} \mathbf{I}_F\right)$  non-singular. This parameter has several effects:

- For large values of  $\lambda$ , (15) gives  $\Delta\mathbf{p}^{(n)} \simeq -\frac{1}{\lambda} \mathbf{g}$ , i.e., a short step in the steepest descent direction.
- For small values of  $\lambda$ , (15) reduces to a Gauss-Newton update.

The updating strategy for  $\lambda^{(n)}$  is thoroughly described in [9] and is based on the gain ratio  $\rho$  between the actual variation of the loss function  $(\Delta\phi)$  and the decrease of its estimate, obtained from the linear approximation (12)  $(\Delta\tilde{\phi})$ .

We finally build a Levenberg-Marquardt type algorithm for the computation of the BFM (6). We define the following stop criterion:

$$c^{(n)} = \|\hat{\mathcal{Y}}^{(n)} - \hat{\mathcal{Y}}^{(n-1)}\|_2^2. \quad (16)$$

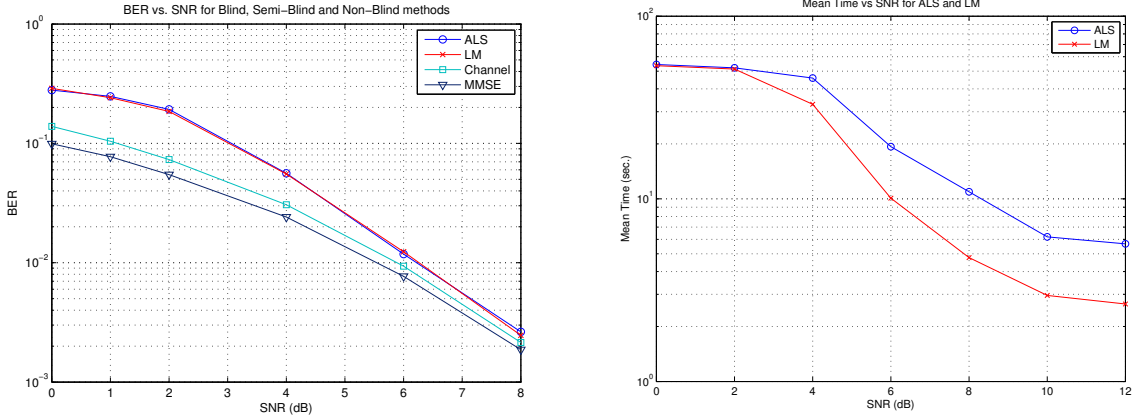


Figure 2: Performance of ALS and LM in presence of AWGN.

*Summary of the algorithm:*

- 1- Initialize  $\mathbf{p}$  randomly
- 2- Solve  $(\mathbf{J}^H \mathbf{J} + \lambda \mathbf{I}) \Delta \mathbf{p} = -\mathbf{g}$
- 3- Update  $\mathbf{p}$ :  $\mathbf{p}_{new} = \mathbf{p} + \Delta \mathbf{p}$
- 4- Update  $\lambda$  taking into account the gain ratio  $\rho$
- 5- Repeat from 2 until  $c^{(n)} < \varepsilon$  (e.g.  $\varepsilon = 10^{-5}$ )

## 5.2 Results of simulations

In this section, we illustrate the performance of the LM 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 400 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  $R = 3$  users.

In Fig 2, the curves on the left show the accuracy of the BFM calculated either by ALS or LM in terms of the Bit Error Rate (BER). We also plot the performance of the 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) and that of a semi-blind technique which only assumes that the channel is 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 LM algorithms give the same curve which was expected since these methods reduce the same cost function. The right figure compares the mean CPU time required by both techniques for the 400 runs. For each run we considered the (minimal) CPU time where the BER reached its final value. Thanks to the quadratic convergence property of the Levenberg-Marquardt algorithm, the computation cost has been considerably reduced (e.g. gain of 50 percent for SNR=6 dB).

## 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. Our model can be applied to other systems where three diversities are available (e.g. temporal oversampling instead of code diversity). The computational strategy for the calculation of the BFM decomposition is an important issue. It turns out that a derivative-based technique adapted of the LM type converges much faster than the ALS.

## REFERENCES

- [1] N.D. Sidiropoulos, G.B. Giannakis, and R. Bro, "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, 2005, 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] G. Tomasi and R. Bro, "A comparison of algorithms for fitting the parafac model," in *Comp. Stat. Data Anal.* 2005, to appear.
- [8] D. Marquardt, "An algorithm for least-squares estimation of non-linear parameters," in *SIAM J. Appl. Math.*, vol. 11, pp. 431–441. 1963.
- [9] K. Madsen, H.B. Nielsen, and O. Tingleff, "Methods for non-linear least squares problems," Technical University of Denmark, 2004, second ed.