Trilinear Wiener Filtering: Application to Equalization Problems

Lucas N. Ribeiro, João C. M. Mota, André L. F. de Almeida

***Abstract*— This paper presents a trilinear equalizer that is based on the classical Wiener filter. We review the theory of multiway arrays and then apply it in constructing a model of operation for the filter. However, the cost function of the estimation problem in the model is nonlinear. This problem overcame by using a property of the tensor versus vector product, allowing us to divide the nonlinear optimization problem in three linear problems. Then we develop an iterative algorithm for obtaining the optima filters for the problems. The performance of the filter is evaluated in two applications problems.**

***Keywords*— Wiener Filtering, Multilinear Algebra, Alternating Least-Squares, PARAFAC.**

I. INTRODUCTION

In the past years, multidimensional models have been used to solve signal processing problems applied in many fields like psychometrics [7], data analysis [11], chemometrics [16], educametrics [15] and communications systems [5]. Sidiropoulos on his paper [14] described a blind PARAFAC receiver that explored the diversity of DS-CDMA systems for estimating the factor matrices of the transmitted datacubes. In his paper [13], Muti et al. considered multiway filtering, using the subspace method and multimode Principal Component Analysis (PCA). His second approach on filtering was extending classical Wiener to tensor data. He applied the multiway Wiener filter on digital image processing and multicomponent seismic data.

In his application of the multiway Wiener filter, a colored image is modelled as a third order tensor $\mathcal{X}$. Given a received tensor $\mathcal{R}$, an estimation $\hat{\mathcal{X}}$ is generated by successively filtering the received tensor by N n-mode filters. The filtering is performed by multiplying a filter matrix by each slice of the input tensor. The optimization criterion used to determine the optimal n-mode filters is the minimization of the mean-squared error (MSE) between the desired signal $\mathcal{X}$ and the estimation $\hat{\mathcal{X}}$.

O. Filiz [6] proposed an alternating adaptative algorithm for estimating rank constrained spatial-temporal filters, since the solution for the optimization problem has not closed form. Using this same idea of dividing a complex optimization problem in simple problems, we propose a trilinear filter that instead of estimating the factor matrices of information about the channel, we devise the optimum trilinear filter that minimizes the error in the mean square sense. The three factors vectors of the filter will be obtained by solving the classical Wiener-Hopf equations, as we will see in the development.

Lucas N. Ribeiro, João C. M. Mota, André L. F. de Almeida are with Departament of Teleinformatics Engineering, Federal University of Ceará, Fortaleza-CE, Brazil. E-mails: {nogueira,mota,andre}@gtel.ufc.br. This work was supported by CNPq and FUNCAP.

The paper is organized as follows: section II presents some concepts about tensors, in section III, we describe a model for transmitting signal throughout trilinear systems and noise. Section IV describes the algebric development to obtain the Wiener solutions and V describes the trilinear alternating least-squares algorithm for obtaining the optimal solutions. Finally, section VI shows some applications and we conclude the paper on section VII.

II. MULTI-WAY ARRAYS

A Nth order tensor is a multidimensional array whose entries are accessed via N indices [13]. It can be represented by $\mathcal{U} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ for example. If $\mathcal{U}$ has rank one, it can be written as the outer product "$\circ$" of N vectors, i.e.:

$$\mathcal{U} = \mathbf{u}_1 \circ \mathbf{u}_2 \circ \ldots \circ \mathbf{u}_N \tag{1}$$

where $\mathbf{u}_1 \in \mathbb{R}^{I_1}$, $\mathbf{u}_2 \in \mathbb{R}^{I_2}$, ... , $\mathbf{u}_N \in \mathbb{R}^{I_N}$ are its factor vectors. The rank of a tensor $\mathcal{X}$ is defined as the smallest number of rank-one tensors that generates $\mathcal{X}$ as their sum [9]. For example, consider that $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ has rank R, then we can decompose it as:

$$\mathcal{X} = \sum_{r=1}^{R} \mathbf{x}_{1,r} \circ \mathbf{x}_{2,r} \circ \mathbf{x}_{3,r} \tag{2}$$

where R is a positive integer and $\mathbf{x}_{1,r}$, $\mathbf{x}_{2,r}$, $\mathbf{x}_{3,r}$ are the factor vectors of the rth rank-one tensor. The equation (2) is known as the PARAFAC (parallel factors) decomposition [2].

Hereafter, the notation for the nth element of a certain vector $\mathbf{z}$, will be denoted by $z^{(n)}$. Then, the element of a rank R third order tensor $\mathcal{X}$ indexed by (i, j, k) is:

$$x_{ijk} = \sum_{r=1}^{R} x_{1,r}^{(i)} x_{2,r}^{(j)} x_{3,r}^{(k)} \tag{3}$$

A tensor can be also decomposed in slices, which are bidimensional sections defined by fixing all but two indices. Consider a third order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ that can be composed like this: $\mathcal{X} = \mathbf{x}_1 \circ \mathbf{x}_2 \circ \mathbf{x}_3$. Its slices through the three modes are: $\mathbf{X}_{i\cdot\cdot} \in \mathbb{R}^{I_2 \times I_3}$, $\mathbf{X}_{\cdot j\cdot} \in \mathbb{R}^{I_3 \times I_1}$ and $\mathbf{X}_{\cdot\cdot k} \in \mathbb{R}^{I_1 \times I_2}$, where $\mathbf{X}_{i\cdot\cdot}$ denotes the ith slice of the first mode, $\mathbf{X}_{\cdot j\cdot}$ is the jth slice of the second mode and $\mathbf{X}_{\cdot\cdot k}$ is the kth slice of the third mode. In this paper, we will build the unfolded of a third order tensor, in the following way:

$$\mathbf{X}_{(1)} = [\mathbf{X}_{1\cdot\cdot}, \ldots, \mathbf{X}_{I_1\cdot\cdot}]^T \in \mathbb{R}^{I_1 I_2 \times I_3} \tag{4}$$

$$\mathbf{X}_{(2)} = [\mathbf{X}_{\cdot 1\cdot}, \ldots, \mathbf{X}_{\cdot I_2\cdot}]^T \in \mathbb{R}^{I_2 I_3 \times I_1} \tag{5}$$

$$\mathbf{X}_{(3)} = [\mathbf{X}_{\cdot\cdot 1}, \ldots, \mathbf{X}_{\cdot\cdot I_3}]^T \in \mathbb{R}^{I_3 I_1 \times I_2} \tag{6}$$

where $\mathbf{X}_{(n)}$ is the tensor $\mathcal{X}$ unfolded in the mode-n. These unfolded matrices can be represented by the following equations:

$$\begin{aligned} \mathbf{X}_{(1)} &= (\mathbf{x}_1 \otimes \mathbf{x}_2)\mathbf{x}_3^T && (7)\\ \mathbf{X}_{(2)} &= (\mathbf{x}_2 \otimes \mathbf{x}_3)\mathbf{x}_1^T && (8)\\ \mathbf{X}_{(3)} &= (\mathbf{x}_3 \otimes \mathbf{x}_1)\mathbf{x}_2^T && (9) \end{aligned}$$

where "$\otimes$" denotes the Kronecker product.

The n-mode product of a Nth order tensor $\mathcal{R} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ with a vector $\mathbf{v} \in \mathbb{R}^{I_n}$, denoted by $\mathcal{R} \times_n \mathbf{v}$, results into a tensor $\mathcal{Y}$ of order $N-1$ whose size is $I_1 \times \ldots \times I_{n-1} \times I_{n+1} \times \ldots \times I_N$ [1]. Its elementwise representation is:

$$y_{i_1 \ldots i_{n-1} i_{n+1} \ldots i_N} = \sum_{i_n=1}^{I_N} r_{i_1 i_2 \ldots i_N} v_{i_n} \tag{10}$$

In this paper, we will utilize the consecutive n-mode vector product between a rank R third order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ and three vectors $\mathbf{v}_1 \in \mathbb{R}^{I_1}$, $\mathbf{v}_2 \in \mathbb{R}^{I_2}$ and $\mathbf{v}_3 \in \mathbb{R}^{I_3}$. Since each n-mode vector product decreases one order of the resulting tensor, multiplying a tensor by three vectors results into a zeroth order tensor, i.e. a scalar. This consecutive product is written as:

$$y = \mathcal{X} \times_1 \mathbf{v}_1 \times_2 \mathbf{v}_2 \times_3 \mathbf{v}_3 \tag{11}$$

where y is a real scalar. Using equation (10), we can write the elementwise representation of x_{ijk}, equation (3):

$$y = \sum_{i=1}^{I_1} \sum_{j=1}^{I_2} \sum_{k=1}^{I_3} v_1^{(i)} v_2^{(j)} v_3^{(k)} x_{ijk} \tag{12}$$

Substituting the equation of the elementwise representation fo $\mathcal{X}$, equation 3, into equation (12):

$$\begin{aligned} y &= \sum_{r=1}^{R} \sum_{i=1}^{I_1} \sum_{j=1}^{I_2} \sum_{k=1}^{I_3} v_1^{(i)} v_2^{(j)} v_3^{(k)} x_{1,r}^{(i)} x_{2,r}^{(j)} x_{3,r}^{(k)} && (13)\\ &= \sum_{r=1}^{R} (\mathbf{v}_1^T \mathbf{x}_{1,r})(\mathbf{v}_2^T \mathbf{x}_{2,r})(\mathbf{v}_3^T \mathbf{x}_{3,r}) && (14) \end{aligned}$$

Applying the commutative property of the scalar product between the vectors and rearranging the factors:

$$y = \sum_{r=1}^{R} (\mathbf{v}_3^T \mathbf{U}_{1,r} \mathbf{v}_2)(\mathbf{x}_{1,r}^T \mathbf{v}_1) \tag{15}$$

where $\mathbf{U}_{1,r} = \mathbf{x}_{3,r}\mathbf{x}_{2,r}^T$. Applying the $vec(\cdot)$ operator on equation (15), and knowing that [12]:

$$vec(\mathbf{ABC}) = (\mathbf{C}^T \otimes \mathbf{A}) vec(\mathbf{B}) \tag{16}$$

we have that equation (15) becomes:

$$y = \sum_{r=1}^{R} vec(\mathbf{v}_3^T \mathbf{U}_{1,r} \mathbf{v}_2)(\mathbf{x}_{1,r}^T \mathbf{v}_1) \tag{17}$$

Since $vec(\mathbf{U}_{1,r}) = \mathbf{x}_{2,r} \otimes \mathbf{x}_{3,r}$, equation (17) turns into:

$$y = \sum_{r=1}^{R} (\mathbf{v}_2 \otimes \mathbf{v}_3)^T (\mathbf{x}_{2,r} \otimes \mathbf{x}_{3,r}) \mathbf{x}_{1,r}^T \mathbf{v}_1 \tag{18}$$

substituting equation (8), we obtain:

$$y = \sum_{r=1}^{R} (\mathbf{v}_2 \otimes \mathbf{v}_3)^T \mathbf{X}_{(2)}^{(r)} \mathbf{v}_1 \tag{19}$$

where $\mathbf{X}_{(2)}^{(r)}$ is the mode 2 unfolding of the rth rank one tensor of the PARAFAC decomposition of $\mathcal{X}$. The equation (19) brings an important result, for it is the matricial product version of the n-mode vector product. If we initiate this development isolating either $(\mathbf{x}_2^T \mathbf{v}_2)$ or $(\mathbf{x}_3^T \mathbf{v}_3)$ on equation (14) and repeating the development, we will obtain respectively these equations:

$$y = \sum_{r=1}^{R} (\mathbf{v}_3 \otimes \mathbf{v}_1)^T \mathbf{X}_{(3)}^{(r)} \mathbf{v}_2 \tag{20}$$

$$y = \sum_{r=1}^{R} (\mathbf{v}_1 \otimes \mathbf{v}_2)^T \mathbf{X}_{(1)}^{(r)} \mathbf{v}_3 \tag{21}$$

The equations (19), (20) and (21) will be important in section IV.

III. The trilinear filters processing model

Consider that we would like to transmit the real set of sequences $\{s_1(n), s_2(n), \ldots, s_R(n)\}$ through the rank-one trilinear systems $\{\mathcal{H}_1, \mathcal{H}_2, \ldots, \mathcal{H}_R\}$ where $\mathcal{H}_r \in \mathbb{R}^{I_1 \times I_2 \times I_3} \forall r$. Since each tensor system has unitary rank, it can be decomposed as:

$$\mathcal{H}_r = \mathbf{h}_{a,r} \circ \mathbf{h}_{b,r} \circ \mathbf{h}_{c,r} \tag{22}$$

where $\mathbf{h}_{a,r} \in \mathbb{R}^{I_1}$, $\mathbf{h}_{b,r} \in \mathbb{R}^{I_2}$ and $\mathbf{h}_{c,r} \in \mathbb{R}^{I_3}$ are the factor vectors of $\mathcal{H}_r$ and R denotes the number of sources. The output of the rth trilinear system is:

$$\mathcal{U}_r(n) = \mathcal{H}_r s_r(n) = (\mathbf{h}_{a,r} \circ \mathbf{h}_{b,r} \circ \mathbf{h}_{c,r})\, s_r(n) \tag{23}$$

The operation on equation (23) is a product between a scalar and a tensor, resulting into a tensorial sequence $\mathcal{U}_r(n) \in \mathbb{R}^{I_1 \times I_2 \times I_3}$.

Let us consider an additive white, independent-from-signal and Gaussian noise tensor $\mathcal{B} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$. The white Gaussian noise assumption can be formulated as

$$E[b_{i_1 i_2 i_3} b_{j_1 j_2 j_3}] = \delta_{i_1 j_1} \ldots \delta_{i_3 j_3} \tag{24}$$

where i_k and $j_k \in \{1, \ldots, I_k\}$, $k \in \{1, 2, 3\}$ and δ is the Kronecker delta. The outputs of the trilinear systems are combined with the noise tensor $\mathcal{B}$. Then, the received signal $\mathcal{X}(n) \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ is:

$$\mathcal{X}(n) = \sum_{r=1}^{R} s_r(n) \mathcal{H}_r + \mathcal{B} \tag{25}$$

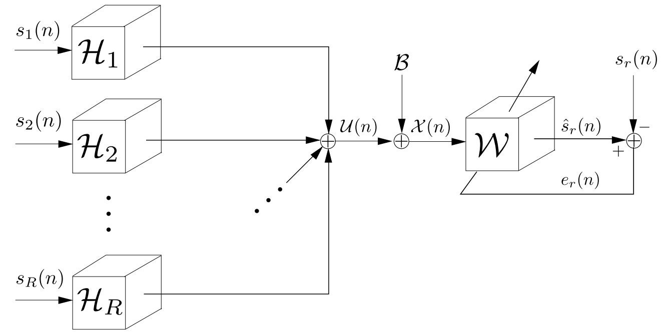

Fig. 1. Block diagram for the model.

Observe that now R represents both the number of sources and the rank of the tensor $\mathcal{X}$, whose elementwise representation is:

$$x_{ijk}(n) = \sum_{r=0}^{R} h_{a,r}^{(i)} h_{b,r}^{(j)} h_{c,r}^{(k)} s_r(n) + b_{ijk} \tag{26}$$

where b_{ijk} is the elementwise representation of $\mathcal{B}$. We desire to recover the real sequence $s_r(n)$, then we will filter the signal $\mathcal{X}(n)$ with a rank-1 trilinear system

$$\mathcal{W} = \mathbf{w}_a \circ \mathbf{w}_b \circ \mathbf{w}_c \tag{27}$$

where $\mathbf{w}_a \in \mathbb{R}^{I_1}$, $\mathbf{w}_b \in \mathbb{R}^{I_2}$ and $\mathbf{w}_c \in \mathbb{R}^{I_3}$ are its factor vectors. It produces an output $\hat{s}_r(n) \in \mathbb{R}$ that is compared to the desired signal $s_r(n)$, generating an estimation error $e_r(n)$. The coefficients of the factor vectors of $\mathcal{W}$ are tuned according to an algorithm that minimizes an optimization criterion based on $e_r(n)$.

Since $\mathcal{W}$ has unitary rank, we define the output of the filter as the n-mode vector product between the input tensor and the factor vectors of the filter, resulting into the estimated signal:

$$\hat{s}_r(n) = \mathcal{X}(n) \times_1 \mathbf{w}_a \times_2 \mathbf{w}_b \times_3 \mathbf{w}_c \tag{28}$$

We define the cost function J of the algorithm as the MSE of the estimation error and its parameters are the factor vectors $\mathbf{w}_a$, $\mathbf{w}_b$ and $\mathbf{w}_c$:

$$\begin{aligned} J &= E[|s_r(n) - \hat{s}_r(n)|^2] && (29) \\ &= E[|s_r(n) - \mathcal{X}(n) \times_1 \mathbf{w}_a \times_2 \mathbf{w}_b \times_3 \mathbf{w}_c|^2] && (30) \end{aligned}$$

In order to obtain the optima filters that minimize the MSE, we need to solve the following problem:

$$\min_{\mathbf{w}_a, \mathbf{w}_b, \mathbf{w}_c} E[|s_r(n) - \mathcal{X}(n) \times_1 \mathbf{w}_a \times_2 \mathbf{w}_b \times_3 \mathbf{w}_c|^2] \tag{31}$$

According to [8], the MSE will attain its minimum value when the gradient vector $\nabla J = \mathbf{0}$. However, the problem (31) is not linear, for there is an explicit product of parameters of the problem. Therefore, we can not apply the classical optimization techniques. To solve this problem, we will use the fact that the n-mode vector product can be written in three equivalent different ways, as we have seen through equations (19), (20) and (21).

IV. The trilinear Wiener filters

The equation (28) can be expressed using the equations (19), (20) and (21), resulting into:

$$\begin{aligned} \hat{s}_r(n) &= \sum_{r=1}^{R} (\mathbf{w}_b \otimes \mathbf{w}_c)^T \mathbf{X}_{(2)}^{(r)}(n) \mathbf{w}_a && (32) \\ &= \sum_{r=1}^{R} (\mathbf{w}_c \otimes \mathbf{w}_a)^T \mathbf{X}_{(3)}^{(r)}(n) \mathbf{w}_b && (33) \\ &= \sum_{r=1}^{R} (\mathbf{w}_a \otimes \mathbf{w}_b)^T \mathbf{X}_{(1)}^{(r)}(n) \mathbf{w}_c && (34) \end{aligned}$$

We can use the equations (32), (33) and (34) to obtain three linear problems from the problem (31). It can be done by fixing the two parameters in the Kronecker product. Since we fix two parameters, we affirm that the following vectors are constant:

$$\begin{aligned} \mathbf{u}_a(n) &= \sum_{r=1}^{R} \left(\mathbf{X}_{(2)}^{(r)}\right)^T (\mathbf{w}_b \otimes \mathbf{w}_c) && (35) \\ \mathbf{u}_b(n) &= \sum_{r=1}^{R} \left(\mathbf{X}_{(3)}^{(r)}\right)^T (\mathbf{w}_c \otimes \mathbf{w}_a) && (36) \\ \mathbf{u}_c(n) &= \sum_{r=1}^{R} \left(\mathbf{X}_{(1)}^{(r)}\right)^T (\mathbf{w}_a \otimes \mathbf{w}_b) && (37) \end{aligned}$$

Rewriting the equations (32), (33) and (34) using the equations (35), (36) and (37), we can obtain three LS estimations:

$$\hat{\mathbf{w}}_a = \arg\min_{\mathbf{w}_a} E[|s_r(n) - \mathbf{u}_a^T \mathbf{w}_a|^2] \tag{38}$$

$$\hat{\mathbf{w}}_b = \arg\min_{\mathbf{w}_b} E[|s_r(n) - \mathbf{u}_b^T \mathbf{w}_b|^2] \tag{39}$$

$$\hat{\mathbf{w}}_c = \arg\min_{\mathbf{w}_c} E[|s_r(n) - \mathbf{u}_c^T \mathbf{w}_c|^2] \tag{40}$$

It happens that the equations (38), (39) and (40) are classical least squares (LS) estimations and its solutions are [8]:

$$\hat{\mathbf{w}}_a = \mathbf{R}_a^{-1} \mathbf{p}_a \tag{41}$$

$$\hat{\mathbf{w}}_b = \mathbf{R}_b^{-1} \mathbf{p}_b \tag{42}$$

$$\hat{\mathbf{w}}_c = \mathbf{R}_c^{-1} \mathbf{p}_c \tag{43}$$

where

$$\begin{aligned} \mathbf{R}_a &= E[\mathbf{u}_a(n) \mathbf{u}_a(n)^T] && (44) \\ \mathbf{R}_b &= E[\mathbf{u}_b(n) \mathbf{u}_b(n)^T] && (45) \\ \mathbf{R}_c &= E[\mathbf{u}_c(n) \mathbf{u}_c(n)^T] && (46) \end{aligned}$$

are the covariance matrices of the input filters $\mathbf{w}_a$, $\mathbf{w}_b$, $\mathbf{w}_c$ respectively and

$$\begin{aligned} \mathbf{p}_a &= E[\mathbf{u}_a s_r(n)] && (47) \\ \mathbf{p}_b &= E[\mathbf{u}_b s_r(n)] && (48) \\ \mathbf{p}_c &= E[\mathbf{u}_c s_r(n)] && (49) \end{aligned}$$

where $\mathbf{p}_a$, $\mathbf{p}_b$ and $\mathbf{p}_c$ are the cross correlation vectors between the input of the filters $\mathbf{w}_a$, $\mathbf{w}_b$, $\mathbf{w}_c$ and the desired signal $s_r(n)$.

V. Trilinear Alternating Least-Squares (TALS)

In order to obtain the optimum filters, we solve, in an alternate way, each problem using the LS estimates (38), (39) and (40), fixing all parameters but the argument. This way, the sequences $\{\hat{\mathbf{w}}_a^k\}_{k\in\mathbb{N}}$, $\{\hat{\mathbf{w}}_b^k\}_{k\in\mathbb{N}}$ e $\{\hat{\mathbf{w}}_c^k\}_{k\in\mathbb{N}}$ may converge to its Wiener solution after k TALS steps. The condition for convergence is that the norm of the vector Δ is less then a certain positive value ε. The TALS algorithm is summarized in the following steps:

1) Initialize $\hat{\mathbf{w}}_a(0)$, $\hat{\mathbf{w}}_b(0)$, $\hat{\mathbf{w}}_c(0)$ and $k = 1$.
2) ALS loop: while $||\Delta|| > \varepsilon$
   a) Using $\hat{\mathbf{w}}_b(k)$,$\hat{\mathbf{w}}_c(k)$, estimate $\hat{\mathbf{w}}_a(k) = \hat{\mathbf{R}}_a^{-1}\hat{\mathbf{p}}_{da}$
   b) Using $\hat{\mathbf{w}}_c(k)$,$\hat{\mathbf{w}}_a(k)$, estimate $\hat{\mathbf{w}}_b(k) = \hat{\mathbf{R}}_b^{-1}\hat{\mathbf{p}}_{db}$
   c) Using $\hat{\mathbf{w}}_a(k)$,$\hat{\mathbf{w}}_b(k)$, estimate $\hat{\mathbf{w}}_c(k) = \hat{\mathbf{R}}_c^{-1}\hat{\mathbf{p}}_{dc}$
   d) Calculate $\Delta = \mathcal{W}(k) - \mathcal{W}(k-1)$
   e) $k = k + 1$

where $\mathcal{W}$ is given by equation (27). In order to calculate the covariance matrices and crosscorrelation vector estimates, the input vector is calculated (equations 35, 36, 37), and then the covariance matrix (equations 44, 45, 46) and the cross-correlation vector are calculated (equations 47, 48, 49).

VI. Applications

In this section, we expose two applications of the trilinear Wiener filtering in signal processing problems. The SNR is defined as

$$SNR = 10log\left(\frac{E_s}{||\mathcal{B}||_F^2}\right) \tag{50}$$

where E_s is the energy of the transmitted signal and $||\mathcal{B}||_F^2$ is the Frobenius norm of the noise tensor. The first application is the recovery of a sinusoid using the model described in the section III and the TALS algorithm (section V) with $\varepsilon = 10^{-4}$. In the first simulation, each source r transmits a sinusoid whose frequency is $(10.r)$ Hz. The recovery was done using a $4 \times 4 \times 4$ trilinear filter in a multi-source environment of 4 sources. We have recovered the sinusoid of 30 Hz as depicted in figure (2). In the second simulation, we have analyzed the MSE vs. SNR performance of three filters, as shown on figure 3. It can be seen that as we increase the order of the filter, it performs better, since it can process more information in a single run.

The second application is the transmission of BPSK signals by $R = 4$ sources. The equalizer will try to estimate the transmitted signals using the TALS algorithm for $\varepsilon = 10^{-4}$ and we will analyze the SNR vs. BER performance. Like in the first application, as we increase the order of the filter, it performs better in the SNR vs. BER. Through figure 5, we observe that as we increase the number of transmitting sources, the interefence increases, then the filter performs worse.

VII. Conclusions

Multilinear algebra applied to signal processing has been a promising tool and has allowed the exploration of a new field of research and applications. In this paper, we managed to mix the classical Wiener filter with a multilinear structure. Instead of solving a non-linear problem, we have divided the problem in three linear problems that were solved using LS estimators. The trilinear Wiener filter works under the assumption that the received signal has been corrupted by a trilinear system and summed up to a trilinear noise. Under these assumptions, we have shown through simulations that it performs satisfactorily. As we increase the order of the equalizer, it performs better and better. But as we increase the rank of the received tensor signal, the equalizer needs to deal with more interference. There can be some improvements to the trilinear filter like increasing the rank of the equalizer filter and deriving an adaptative algorithm for the model.

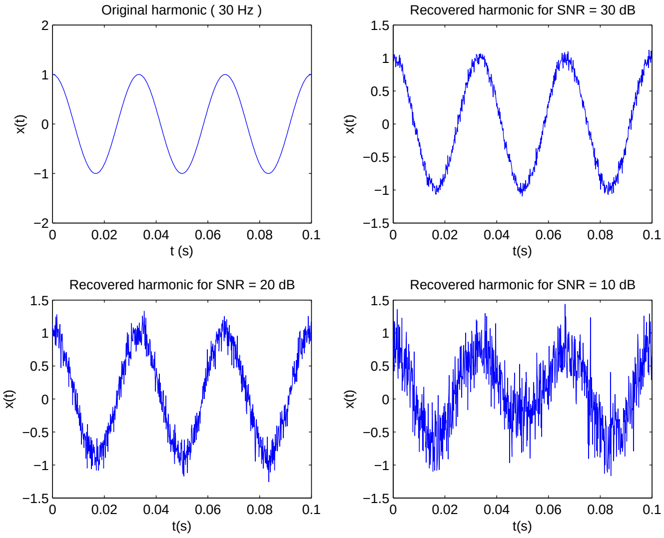

Fig. 2. Simulation 1: recovery of a 30 Hz harmonic using a $4 \times 4 \times 4$ trilinear filter.

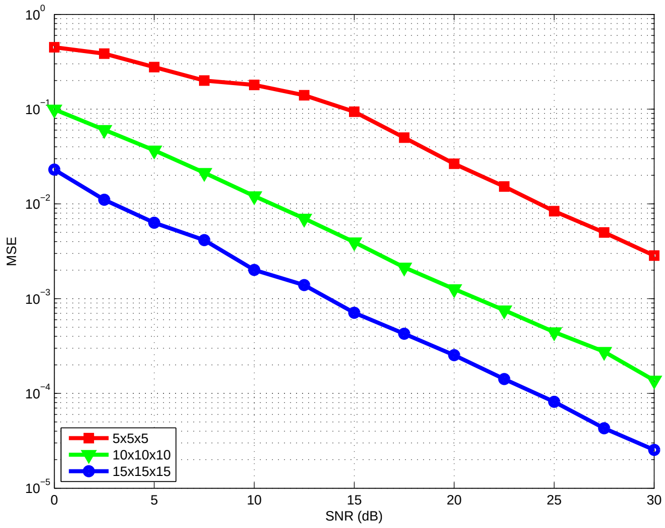

Fig. 3. Simulation 2: MSE vs SNR performance for filters of order $5 \times 5 \times 5$, $10 \times 10 \times 10$ and $15 \times 15 \times 15$ and R = 4 sources.

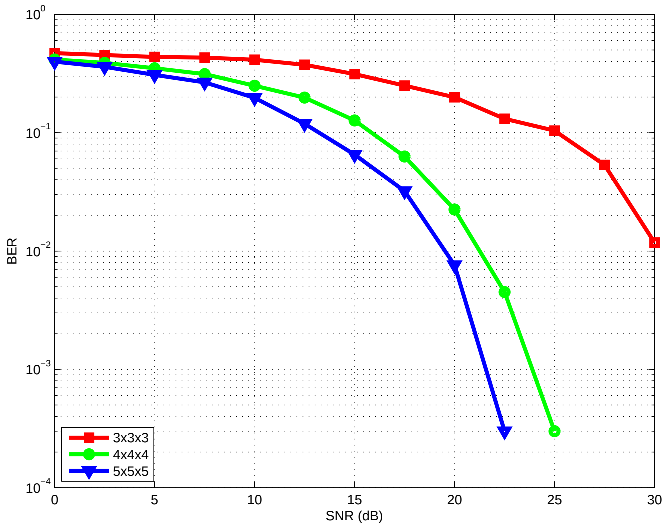

Fig. 4. Simulation 3: BER vs. SNR performance for 4 users.

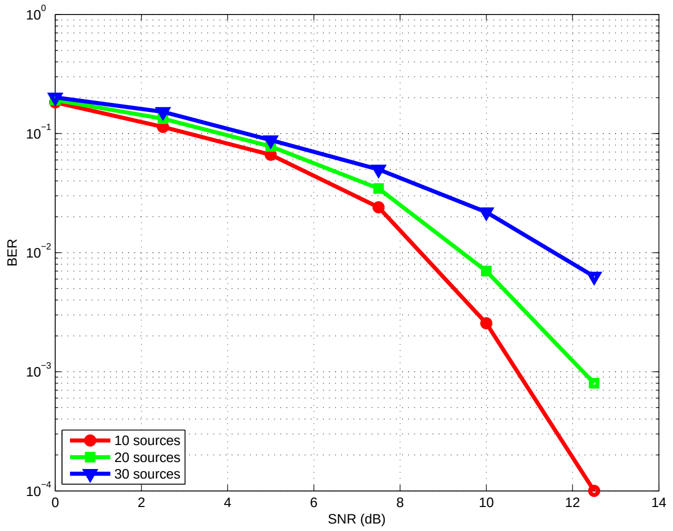

Fig. 5. Simulation 4: BER vs. SNR performance for a $10 \times 10 \times 10$ filter.

REFERENCES

[1] B. W. Bader and T. G. Kolda, "Algorithm 862: Matlab tensor classes for fast algorithm prototyping," *ACM Trans. Math. Softw.*, vol. 32, no. 4, pp. 635–653, Dec. 2006.

[2] J. Carroll and J. J. Chang, "Analysis of individual differences in multidimensional scaling via an n-way generalization of 'Eckart-Young' decomposition," *Psychometrika*, vol. 35, no. 3, pp. 283–319, Sept. 1970.

[3] A. L. F. De Almeida, G. Favier, C. Fernandes, and J. C. M. Mota, "A trilinear decompositiond approach fof space-time-frequency multiple-access wireless systems," in *Signal Processing Advances in Wireless Communications, 2007. SPAWC 2007. IEEE 8th Workshop on*, 2007, pp. 1–5.

[4] A. L. F. De Almeida, G. Favier, and J. C. M. Mota, "Trilinear space-time-frequency codes for broadband MIMO-OFDM systems," in *Telecommunications Symposium, 2006 International*, 2006, pp. 764–770.

[5] A. L. F. de Almeida, G. Favier, and J. C. M. Mota, "Space-time spreading-multiplexing for MIMO wireless communication systems using the PARATUCK-2 tensor model," *Signal Process.*, vol. 89, no. 11, pp. 2103–2116, 2009.

[6] O. Filiz and A. Yener, "Rank constrained temporal-spatial matrix filters for CDMA systems," *Trans. Wireless. Comm.*, vol. 3, no. 6, pp. 1974–1979, 2004.

[7] R. A. Harshman, "Foundations of the PARAFAC procedure: Models and conditions for an 'explanatory' multi-modal factor analysis," *UCLA Working Papers in Phonetics*, vol. 16, no. 1, p. 84, 1970.

[8] S. Haykin, *Adaptive filter theory*, ser. Prentice-Hall information and system sciences series. Prentice Hall, 2002.

[9] T. D. Howell, "Global properties of tensor rank," *Linear Algebra and its Applications*, vol. 22, no. 0, pp. 9 – 23, 1978.

[10] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," Sandia National Laboratories, Albuquerque, NM and Livermore, CA, Tech. Rep. SAND2007-6702, November 2007.

[11] H. Lu, K. Plataniotis, and A. Venetsanopoulos, "MPCA: Multilinear principal component analysis of tensor objects," *IEEE Transactions on Neural Networks*, vol. 19, no. 1, pp. 18–39, 2008.

[12] J. Magnus and H. Neudecker, *Matrix Differential Calculus with Applications in Statistics and Econometrics*, ser. Wiley series in probability and mathematical statistics. Wiley, 1999.

[13] D. Muti and S. Bourennane, "Multidimensional filtering based on a tensor approach," *Signal Process.*, vol. 85, no. 12, pp. 2338–2353, Dec. 2005. [Online]. Available: http://dx.doi.org/10.1016/j.sigpro.2004.11.029

[14] N. D. Sidiropoulos, G. B. Giannakis, and R. Bro, "Blind parafac receivers for DS-CDMA systems," *IEEE TRANS. SIGNAL PROCESSING*, vol. 48, no. 3, pp. 810–823, 2000.

[15] T. Silva, F. Vasconcelos, A. Almeida, and J. C. M. Mota, "Multivariate analysis for students' evaluation of teaching effectiveness in teleinformatics engineering," in *Teaching, Assessment and Learning for Engineering (TALE), 2012 IEEE International Conference on*, 2012, pp. H1A–1–H1A–6.

[16] A. Smilde, R. Bro, and P. Geladi, *Multi-way Analysis: Applications in the Chemical Sciences*. Wiley, 2004.