



HAL
open science

Distributed PARAFAC based DS-CDMA blind receiver for wireless sensor networks

Alain Kibangou, André L. F. de Almeida

► **To cite this version:**

Alain Kibangou, André L. F. de Almeida. Distributed PARAFAC based DS-CDMA blind receiver for wireless sensor networks. SPAWC 2010 - 11th IEEE International Workshop on Signal Processing Advances for Wireless Communications, Jun 2010, Marrakech, Morocco. pp.n.c. <hal-00480584>

HAL Id: hal-00480584

<https://hal.science/hal-00480584v1>

Submitted on 4 May 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

DISTRIBUTED PARAFAC BASED DS-CDMA BLIND RECEIVER FOR WIRELESS SENSOR NETWORKS

A. Y. Kibangou

GIPSA-Lab/UJF/CNRS
961 Rue de la Houille Blanche,
B.P. 46, F-38402 Grenoble, France
alain.kibangou@ujf-grenoble.fr

A.L.F. de Almeida

GTEL-Wireless Telecom Research Group
Federal University of Ceará
CP 6005, 60455-760, Fortaleza, Brazil
andre@gtel.ufc.br

ABSTRACT

In this paper, we consider a collaborative scheme in wireless sensor networks where the multiple access protocol is a DS-CDMA one. When the receiver is equipped with an antenna array, it has been shown that efficient blind receivers can be derived using the PARAFAC tensor model. In general, the parameters of the PARAFAC model are fitted using an alternating least squares algorithm. Herein, we consider the case where each receiver has a single antenna. Therefore, by allowing collaboration in a predefined neighborhood, we derive a distributed alternating least squares algorithm including some average consensus steps.

1. INTRODUCTION

In recent years, the deployment of sensors for monitoring, collaborative information processing and control has drawn a considerable attention. In particular, wireless sensor networks that can operate autonomously, i.e. without a fusion center collecting and processing all measurements, exhibit desirable properties such as robustness against node failure [1]. The coordinated action of different sensors requires local exchange of information. In this paper, we consider the exchange of information between sensors grouped in two clusters. The data are transmitted over multiple access fading channels. Data are recovered in a totally blind way.

Multiple access protocols such as Time Division Multiple Access (TDMA) can induce a latency that can be damaging for control purposes for example. In this case, Direct-Sequence Code Division Multiple Access (DS-CDMA) is certainly well indicated. In the last decade, by exploiting several diversities, new signal processing techniques based on tensor modeling have been developed. With a very high efficiency, they allow the blind estimation of transmitted information sequences [2, 3, 4, 5]. In general, these works are devoted to communication systems with an antenna array at the receiver. However, many wireless devices are limited by size, hardware complexity or other constraints to a single antenna. The powerful tensor based methods cannot be applied for such nodes. In this paper, by resorting to the idea of collaborative signal processing, we show how estimating the channel, symbols, and codes in a distributed way when each node in the network has a single antenna. The received data samples can be stored in a three-way array, or a third-order tensor, admitting a PARAFAC model [6]. In general, the parameters of the PARAFAC model are estimated using an Alternating Least Squares (ALS) algorithm. In this paper, we derive a distributed version of ALS using average consensus iterations. Average consensus is an important problem in algorithm design for distributed computing. In its distributed

framework, it has been extensively studied in computer science (distributed agreement and synchronization problems for example). It is a central topic for load balancing (with divisible tasks) in parallel computers and also has found application in distributed coordination of mobile autonomous agents, distributed data fusion in sensor networks, and distributed estimation and control [7].

Notations: Vectors are written as boldface lower-case letters ($\mathbf{a}, \mathbf{b}, \dots$) and matrices as boldface capitals ($\mathbf{A}, \mathbf{B}, \dots$). $\mathbf{A}_{i \cdot}$ and $\mathbf{A}_{\cdot j}$ denote respectively the i th row and the j th column of the $I \times J$ matrix \mathbf{A} . \mathbf{A}^T stands for the transpose of \mathbf{A} whereas \mathbf{A}^H stands for its complex conjugate. $\text{diag}(\cdot)$ is the operator that forms a diagonal matrix from its vector argument whereas $\text{vec}(\cdot)$ forms a vector by stacking the columns of its matrix argument. For $\mathbf{X} \in \mathbb{C}^{I \times R}$, and $\mathbf{Y} \in \mathbb{C}^{J \times R}$, the Khatri-Rao product, denoted by \odot , is defined as follows:

$$\mathbf{X} \odot \mathbf{Y} = \begin{pmatrix} \mathbf{Y} \text{diag}(\mathbf{X}_1) \\ \mathbf{Y} \text{diag}(\mathbf{X}_2) \\ \vdots \\ \mathbf{Y} \text{diag}(\mathbf{X}_I) \end{pmatrix} \in \mathbb{C}^{IJ \times R}. \quad (1)$$

In the sequel, we will make use the properties of the Khatri-Rao product and the Frobenius norm given below:

$$\text{vec}(\mathbf{X} \text{diag}(\mathbf{z}) \mathbf{Y}^T) = (\mathbf{Y} \odot \mathbf{X}) \mathbf{z}, \quad (2)$$

$$\mathbf{X} \odot \mathbf{Y} = \mathbf{\Pi} (\mathbf{Y} \odot \mathbf{X}), \quad (3)$$

$$\|\mathbf{X}\|_F^2 = \|\mathbf{\Pi} \mathbf{X}\|_F^2, \quad (4)$$

$$\|(\mathbf{X}_1 \ \cdots \ \mathbf{X}_M)\|_F^2 = \sum_{m=1}^M \|\mathbf{X}_m\|_F^2, \quad (5)$$

$$\|(\mathbf{x}_1 \ \cdots \ \mathbf{x}_M)\|_F^2 = \sum_{m=1}^M \|\mathbf{x}_m\|_2^2, \quad (6)$$

$\mathbf{\Pi}$, a permutation matrix, and \mathbf{X} being matrices with compatible dimensions.

2. PRELIMINARIES AND PROBLEM STATEMENT

Let us consider a wireless sensor network composed of smart devices that can sense, compute and exchange information with their neighbors. The sensors are grouped in two clusters A and B, where the Q sensors in cluster A transmit their information to the K sensors in cluster B using a synchronous DS-CDMA protocol through a flat fading channel. Each of the sensors spreads its information sequence $s_{n,q}$, $n = 1, \dots, N$, and encodes it using a code $\{c_{p,q}\}$ of length P before transmission through an unknown channel characterized by a

fading gain $a_{k,q}$, for a transmission between the q th sensor of cluster A and the k th sensor of cluster B. The baseband received signal of each sensor is sampled at the chip rate and decomposed into its polyphase components. So, in the noiseless case, the signal received by the k th sensor of cluster B, for the n th symbol and the p th chip is given by:

$$x_{k,p,n} = \sum_{q=1}^Q a_{k,q} c_{p,q} s_{n,q}. \quad (7)$$

If a central node collects the received signal samples $x_{k,p,n}$, we can build a third-order tensor $\mathcal{X} \in \mathbb{C}^{K \times P \times N}$ from these samples. The data collected from a given sensor k can be cast into a $P \times N$ matrix

$$\mathbf{X}_{k..} = \begin{pmatrix} x_{k,1,1} & \cdots & x_{k,1,N} \\ \vdots & \ddots & \vdots \\ x_{k,P,1} & \cdots & x_{k,P,N} \end{pmatrix} = \mathbf{C} \text{diag}(\mathbf{A}_k) \mathbf{S}^T \quad (8)$$

that can be viewed as a slice of \mathcal{X} , where $\mathbf{A} = [a_{k,q}] \in \mathbb{C}^{K \times Q}$, $\mathbf{C} = [c_{p,q}] \in \mathbb{C}^{P \times Q}$, and $\mathbf{S} = [s_{n,q}] \in \mathbb{C}^{N \times Q}$. In order to recover the information symbols $s_{n,q}$ solely from $\mathcal{X} \in \mathbb{C}^{K \times P \times N}$, we can then make use of tensor-based signal processing methods developed in the last decade [2].

2.1. PARAFAC model

From (7), one can note that \mathcal{X} admits a PARAFAC model [6], (also called CANDECOMP [8]), in which the tensor is completely characterized by the three loading, or factor, matrices \mathbf{A} , \mathbf{C} , and \mathbf{S} . PARAFAC is certainly the most popular tensor model that can be found in the literature. That is surely due to its essential uniqueness, which means that each factor matrix can be determined up to column scaling and permutation, i.e. two sets of matrices $\{\mathbf{A}, \mathbf{C}, \mathbf{S}\}$ and $\{\tilde{\mathbf{A}}, \tilde{\mathbf{C}}, \tilde{\mathbf{S}}\}$ giving rise to the same tensor \mathcal{X} are linked by the following relations $\tilde{\mathbf{A}} = \mathbf{A} \mathbf{\Pi} \mathbf{\Delta}_A$, $\tilde{\mathbf{C}} = \mathbf{C} \mathbf{\Pi} \mathbf{\Delta}_C$, $\tilde{\mathbf{S}} = \mathbf{S} \mathbf{\Pi} \mathbf{\Delta}_S$, with $\mathbf{\Delta}_A \mathbf{\Delta}_C \mathbf{\Delta}_S = \mathbf{I}_Q$, where $\mathbf{\Pi}$ is a $Q \times Q$ permutation matrix whereas $\mathbf{\Delta}_A, \mathbf{\Delta}_C$, and $\mathbf{\Delta}_S$ are $Q \times Q$ diagonal matrices. A sufficient condition for such an uniqueness, called Kruskal's condition, states that the PARAFAC decomposition (7) is essentially unique if [9, 2]

$$k_{\mathbf{A}} + k_{\mathbf{C}} + k_{\mathbf{S}} \geq 2(Q + 1), \quad (9)$$

where $k_{\mathbf{A}}$ denotes the Kruskal-rank, also called k-rank, of \mathbf{A} , i.e. the greatest integer $k_{\mathbf{A}}$ such that any set of $k_{\mathbf{A}}$ columns of \mathbf{A} is independent. The rank and the Kruskal-rank of \mathbf{A} are linked by the following inequality $k_{\mathbf{A}} \leq \text{rank}(\mathbf{A})$.

Another feature of PARAFAC is to provide a simple link between the unfolded forms of a tensor and its factor matrices. Unfolded forms of a tensor are obtained by concatenating tensor slices along the same mode. Indeed, as defined in (8) for the first mode, we can define slices along the second and the third modes respectively as follows:

$$\mathbf{X}_{..p} = \begin{pmatrix} x_{1,p,1} & \cdots & x_{K,p,1} \\ \vdots & \ddots & \vdots \\ x_{1,p,N} & \cdots & x_{K,p,N} \end{pmatrix} = \mathbf{S} \text{diag}(\mathbf{C}_{p.}) \mathbf{A}^T, \quad (10)$$

$$\mathbf{X}_{..n} = \begin{pmatrix} x_{1,1,n} & \cdots & x_{1,P,n} \\ \vdots & \ddots & \vdots \\ x_{K,1,n} & \cdots & x_{K,P,n} \end{pmatrix} = \mathbf{A} \text{diag}(\mathbf{S}_{n.}) \mathbf{C}^T. \quad (11)$$

So, using (1), the three vertically unfolded matrix representations for \mathcal{X} are respectively given by:

$$\mathbf{X}_1 = \begin{pmatrix} \mathbf{X}_{1..} \\ \vdots \\ \mathbf{X}_{K..} \end{pmatrix} = (\mathbf{A} \odot \mathbf{C}) \mathbf{S}^T \in \mathbb{C}^{KP \times N},$$

$$\mathbf{X}_2 = \begin{pmatrix} \mathbf{X}_{..1} \\ \vdots \\ \mathbf{X}_{..P} \end{pmatrix} = (\mathbf{C} \odot \mathbf{S}) \mathbf{A}^T \in \mathbb{C}^{PN \times K},$$

$$\mathbf{X}_3 = \begin{pmatrix} \mathbf{X}_{..1} \\ \vdots \\ \mathbf{X}_{..N} \end{pmatrix} = (\mathbf{S} \odot \mathbf{A}) \mathbf{C}^T \in \mathbb{C}^{NK \times P}.$$

2.2. PARAFAC fitting using ALS

Given the tensor \mathcal{X} , various algorithms can be used for estimating the factor matrices \mathbf{A} , \mathbf{C} , and \mathbf{S} (see [10] for a comparison of these methods). The alternating least squares (ALS) algorithm is the most used one. It acts by alternately minimizing the cost functions in the LS sense

$$\mathcal{J}_1 = \left\| \mathbf{X}_1 - (\mathbf{A} \odot \mathbf{C}) \mathbf{S}^T \right\|_F^2, \quad (12)$$

$$\mathcal{J}_2 = \left\| \mathbf{X}_2 - (\mathbf{C} \odot \mathbf{S}) \mathbf{A}^T \right\|_F^2, \quad (13)$$

$$\mathcal{J}_3 = \left\| \mathbf{X}_3 - (\mathbf{S} \odot \mathbf{A}) \mathbf{C}^T \right\|_F^2. \quad (14)$$

given initial approximations of two factor matrices, $\mathbf{A}^{(0)}$ and $\mathbf{C}^{(0)}$ for instance.

2.3. Problem statement

As stated above, the k th sensor in cluster B receives data that can be cast in the matrix $\mathbf{X}_{k..}$. In order, to retrieve the informative symbol matrix \mathbf{S} , a bilinear decomposition is involved. Such a decomposition is generally non unique. Uniqueness of PARAFAC (a trilinear decomposition) can be exploited by sending the matrices $\mathbf{X}_{k..}$ to a central node where collected data can be cast into a tensor \mathcal{X} . The central node performs the PARAFAC decomposition of \mathcal{X} and then send the estimated factor matrices, or at least \mathbf{S} to the sensors. Due to the existence of a central node, such a scheme is particularly vulnerable. Resorting to distributed estimation is then well suited. One can imagine that sensors exchange their received data samples with their neighbors. As a consequence, after such an information exchange, from its own data matrix and those received from its neighbors, each sensor can form a tensor, which is in fact a sub-tensor of \mathcal{X} . Unfortunately, we cannot ensure that all sub-tensors inherit the uniqueness property of the global tensor. Therefore, in such a scheme, some sensors can obtain undesirable estimates, i.e. estimates that cannot be linked to the actual factor matrices in a unique way. The purpose of the following section is to derive a distributed estimation scheme, preserving the PARAFAC uniqueness property.

3. DISTRIBUTED ALS ALGORITHM

Recently, a great effort has been devoted to the derivation of distributed estimation algorithms [7, 11, 12]. Most of them make use of the consensus algorithm. In this framework, from the centralized ALS, we derive a distributed ALS algorithm (D-ALS).

3.1. Consensus based estimation of the symbol matrix \mathbf{S}

The symbol matrix \mathbf{S} can be estimated by minimizing the cost function \mathcal{J}_1 , which can be rewritten as:

$$\mathcal{J}_1 = \left\| \mathbf{X}_1 - \mathbf{Y}\mathbf{S}^T \right\|_F^2,$$

with $\mathbf{Y} = \mathbf{A} \odot \mathbf{C}$. From the definition of the Khatri-Rao product (1), we get:

$$\mathbf{Y} = \begin{pmatrix} \mathbf{Y}_1 \\ \vdots \\ \mathbf{Y}_K \end{pmatrix} = \begin{pmatrix} \mathbf{C} \mathbf{diag}(\mathbf{A}_1) \\ \vdots \\ \mathbf{C} \mathbf{diag}(\mathbf{A}_K) \end{pmatrix}.$$

By minimizing \mathcal{J}_1 , given \mathbf{Y} , we get

$$\hat{\mathbf{S}}^T = \left(\mathbf{Y}^H \mathbf{Y} \right)^{-1} \mathbf{Y}^H \mathbf{X}_1.$$

This solution can also be written as:

$$\hat{\mathbf{S}}^T = \left(\frac{1}{K} \sum_{k=1}^K \mathbf{Y}_k^H \mathbf{Y}_k \right)^{-1} \left(\frac{1}{K} \sum_{k=1}^K \mathbf{Y}_k^H \mathbf{X}_{k..} \right). \quad (15)$$

Note that the computation of (15) results on averaging local estimates $\mathbf{\Gamma}_k(0) = \mathbf{Y}_k^H \mathbf{Y}_k$ and $\mathbf{\Theta}_k(0) = \mathbf{Y}_k^H \mathbf{X}_{k..}$. Such averaging can be achieved using the consensus algorithm. We have to run two average consensus in parallel so that

$$\begin{aligned} \mathbf{\Gamma}_k(t) &\rightarrow \frac{1}{K} \sum_{k=1}^K \mathbf{\Gamma}_k(0) = \frac{1}{K} \sum_{k=1}^K \mathbf{Y}_k^H \mathbf{Y}_k, \\ \mathbf{\Theta}_k(t) &\rightarrow \frac{1}{K} \sum_{k=1}^K \mathbf{\Theta}_k(0) = \frac{1}{K} \sum_{k=1}^K \mathbf{Y}_k^H \mathbf{X}_{k..} \end{aligned}$$

Therefore, the local estimate of \mathbf{S}^T , defined as $\hat{\mathbf{S}}_k^T = \mathbf{\Gamma}_k^{-1}(t) \mathbf{\Theta}_k(t)$ converges towards $\hat{\mathbf{S}}^T$. One can note that for the calculation of $\mathbf{\Gamma}_k(0)$ and $\mathbf{\Theta}_k(0)$ all the sensors should have the same approximation of the loading matrix \mathbf{C} , a non restrictive condition.

3.2. Consensus based estimation of the code matrix \mathbf{C}

The code matrix \mathbf{C} can be estimated by minimizing \mathcal{J}_3 , which can be rewritten as $\mathcal{J}_3 = \left\| \mathbf{X}_3 - \tilde{\mathbf{Z}}\mathbf{C}^T \right\|_F^2$, with $\tilde{\mathbf{Z}} = (\mathbf{S} \odot \mathbf{A})$. By using the property (4) of the Frobenius norm, we also have $\mathcal{J}_3 = \left\| \mathbf{\Pi}\mathbf{X}_3 - \mathbf{Z}\mathbf{C}^T \right\|_F^2$ where

$$\mathbf{Z} = \mathbf{A} \odot \mathbf{S} = \mathbf{\Pi}\tilde{\mathbf{Z}} = \begin{pmatrix} \mathbf{Z}_1 \\ \mathbf{Z}_2 \\ \vdots \\ \mathbf{Z}_K \end{pmatrix},$$

with $\mathbf{Z}_k = \mathbf{S} \mathbf{diag}(\mathbf{A}_k)$. Therefore, the minimization of \mathcal{J}_3 yields

$$\hat{\mathbf{C}}^T = \left(\mathbf{Z}^H \mathbf{Z} \right)^{-1} \mathbf{Z}^H \mathbf{\Pi}\mathbf{X}_3.$$

This solution can also be written as:

$$\hat{\mathbf{C}}^T = \left(\frac{1}{K} \sum_{k=1}^K \mathbf{Z}_k^H \mathbf{Z}_k \right)^{-1} \left(\frac{1}{K} \sum_{k=1}^K \mathbf{Z}_k^H \mathbf{X}_{k..} \right). \quad (16)$$

As previously, the computation of (16) results on averaging local estimates $\mathbf{\Lambda}_k(0) = \mathbf{Z}_k^H \mathbf{Z}_k$ and $\mathbf{\Psi}_k(0) = \mathbf{Z}_k^H \mathbf{X}_{k..}^T$. The averaging process by means of the consensus algorithm yields:

$$\begin{aligned} \mathbf{\Lambda}_k(t) &\rightarrow \frac{1}{K} \sum_{k=1}^K \mathbf{\Lambda}_k(0) = \frac{1}{K} \sum_{k=1}^K \mathbf{Z}_k^H \mathbf{Z}_k, \\ \mathbf{\Psi}_k(t) &\rightarrow \frac{1}{K} \sum_{k=1}^K \mathbf{\Psi}_k(0) = \frac{1}{K} \sum_{k=1}^K \mathbf{Z}_k^H \mathbf{X}_{k..}^T. \end{aligned}$$

Therefore, the local estimate of \mathbf{C}^T , defined as $\hat{\mathbf{C}}_k^T = \mathbf{\Lambda}_k^{-1}(t) \mathbf{\Psi}_k(t)$ converges towards $\hat{\mathbf{C}}^T$.

3.3. Estimation of the channel matrix \mathbf{A}

The channel parameters are intrinsically local. Therefore there is no need to share these parameters between different sensors. In a centralized scheme, the channel matrix is obtained by minimizing \mathcal{J}_2 . Knowing that the three cost functions are equivalent, we derive the estimation of the channel parameter by minimizing \mathcal{J}_1 . We can note that by using (5) and (6), \mathcal{J}_1 can also be written as

$$\begin{aligned} \mathcal{J}_1 &= \left\| \mathbf{X}_1^T - \mathbf{S}(\mathbf{A} \odot \mathbf{C})^T \right\|_F^2 \\ &= \left\| \left(\mathbf{X}_{1..}^T - \mathbf{S} \mathbf{diag}(\mathbf{A}_1) \mathbf{C}^T \dots \right. \right. \\ &\quad \left. \dots \mathbf{X}_{K..}^T - \mathbf{S} \mathbf{diag}(\mathbf{A}_K) \mathbf{C}^T \right\|_F^2 \\ &= \sum_{k=1}^K \left\| \mathbf{X}_{k..}^T - \mathbf{S} \mathbf{diag}(\mathbf{A}_k) \mathbf{C}^T \right\|_F^2 \\ &= \sum_{k=1}^K \left\| \mathbf{vec}(\mathbf{X}_{k..}^T) - (\mathbf{C} \odot \mathbf{S}) \mathbf{A}_k^T \right\|_2^2. \quad (17) \end{aligned}$$

As a consequence, by the code and the symbol matrices by their local estimates, the local channel parameters can be estimated as follows:

$$\hat{\mathbf{A}}_k^T = (\hat{\mathbf{C}}_k \odot \hat{\mathbf{S}}_k)^\dagger \mathbf{vec}(\mathbf{X}_{k..}^T). \quad (18)$$

4. AVERAGE CONSENSUS ALGORITHM

Let $\mathcal{G} = \{\mathcal{K}, \mathcal{E}\}$ be an undirected connected graph representing the communication graph between the collaborating sensors. $\mathcal{K} = \{1, \dots, K\}$ and \mathcal{E} denote respectively the node set and the edge set, where each edge $\{i, j\} \in \mathcal{E}$ is an unordered pair of distinct nodes. Let $\mathbf{R}_k(0)$ be a matrix assigned to node k at time $t = 0$. The distributed average consensus problem consists in computing the average $(1/K) \sum_{k=1}^K \mathbf{R}_k(0)$ at every node, via local communication and computation on the graph. So, node k carries out its update, at each step, based on its local state and communication with its neighbors $\mathcal{K}_i = \{j | \{i, j\} \in \mathcal{E}\}$.

There are several simple methods for distributed average consensus. For example, each node can store a table of all initial node values known at that time. At each step each pair of neighbors exchange tables of initial values and update their tables. In this flooding algorithm, all nodes know all initial values in a number of steps equal to the diameter of the graph, at which point each can compute the average [7]. In widely used average consensus algorithms, each

node updates itself by adding a weighted sum of differences between neighboring node values and its own. In matrix form, we get:

$$\mathbf{R}_k(t+1) = \mathbf{R}_k(t) + \sum_{j \in \mathcal{K}_i} w_{k,j} (\mathbf{R}_j(t) - \mathbf{R}_k(t)), \quad (19)$$

where $w_{k,j}$ is a weight associated with the edge $\{k, j\}$. In the sequel, we assume that the weights are symmetric. Asymptotic convergence is achieved by choosing for example uniform weights

$$w_{k,j} = 1/d_k \quad j \neq k, \{k, j\} \in \mathcal{E}, \quad (20)$$

where d_i is the degree of node i .

4.1. Distributed ALS algorithm using average consensus

The D-ALS algorithm is constituted by interlacing local ALS steps with consensus iterations. By considering perfect exchange during consensus iterations, it is summarized below:

1. For $k = 1, \dots, K$, initialize $\hat{\mathbf{C}}_k(0) = \mathbf{\Omega} \in \mathbb{C}^{P \times Q}$, with $\mathbf{\Omega}$ chosen in a pre-definite set of possible code matrices, and $\hat{\mathbf{A}}_k(0) \in \mathbb{C}^{1 \times Q}$ with random values. Set the D-ALS iteration $i = 0$, and select a number T of consensus iterations.
2. For $k = 1, \dots, K$, compute $\mathbf{Y}_k(i) = \hat{\mathbf{C}}_k(i) \text{diag}(\hat{\mathbf{A}}_k(i))$, $\mathbf{\Gamma}_k(i, 0) = \mathbf{Y}_k^H(i) \mathbf{Y}_k(i)$, and $\mathbf{\Theta}_k(i, 0) = \mathbf{Y}_k^H(i) \mathbf{X}_{k\dots}$.
3. Run the consensus algorithm for $\mathbf{\Gamma}_k$ and $\mathbf{\Theta}_k$

(a) For $t = 0, 1, \dots, T-1$,

$$\mathbf{\Gamma}_k(i, t+1) = \mathbf{\Gamma}_k(i, t) + \sum_{j \in \mathcal{K}_k} w_{k,j} (\mathbf{\Gamma}_j(i, t) - \mathbf{\Gamma}_k(i, t))$$

$$\mathbf{\Theta}_k(i, t+1) = \mathbf{\Theta}_k(i, t) + \sum_{j \in \mathcal{K}_k} w_{k,j} (\mathbf{\Theta}_j(i, t) - \mathbf{\Theta}_k(i, t))$$

(b) Set $\mathbf{\Gamma}_k(i) = \mathbf{\Gamma}_k(i, T)$ and $\mathbf{\Omega}_k(i) = \mathbf{\Omega}_k(i, T)$.

4. Increment i
5. Compute the local estimates of the symbol matrix \mathbf{S}

$$\hat{\mathbf{S}}_k(i) = \mathbf{\Gamma}_k^{-1}(i-1) \mathbf{\Omega}_k(i-1)$$

6. For $k = 1, \dots, K$, compute $\mathbf{Z}_k(i) = \hat{\mathbf{S}}_k(i) \text{diag}(\hat{\mathbf{A}}_k(i-1))$, $\mathbf{\Lambda}_k(i, 0) = \mathbf{Z}_k^H(i) \mathbf{Z}_k(i)$, and $\mathbf{\Psi}_k(i, 0) = \mathbf{Z}_k^H(i) \mathbf{X}_{k\dots}^T$.
7. Run the consensus algorithm for $\mathbf{\Lambda}_k$ and $\mathbf{\Psi}_k$

(a) For $t = 0, 1, \dots, T-1$,

$$\mathbf{\Lambda}_k(i, t+1) = \mathbf{\Lambda}_k(i, t) + \sum_{j \in \mathcal{K}_k} w_{k,j} (\mathbf{\Lambda}_j(i, t) - \mathbf{\Lambda}_k(i, t))$$

$$\mathbf{\Psi}_k(i, t+1) = \mathbf{\Psi}_k(i, t) + \sum_{j \in \mathcal{K}_k} w_{k,j} (\mathbf{\Psi}_j(i, t) - \mathbf{\Psi}_k(i, t))$$

(b) Set $\mathbf{\Lambda}_k(i) = \mathbf{\Lambda}_k(i, T)$ and $\mathbf{\Psi}_k(i) = \mathbf{\Psi}_k(i, T)$.

8. Compute the local estimates of the code matrix \mathbf{C}

$$\hat{\mathbf{C}}_k(i) = \mathbf{\Lambda}_k^{-1}(i) \mathbf{\Psi}_k(i)$$

9. Compute the local estimates of the channel parameters

$$\hat{\mathbf{A}}_k^T(i) = \left(\hat{\mathbf{C}}_k(i) \odot \hat{\mathbf{S}}_k(i) \right)^\dagger \text{vec}(\mathbf{X}_{k\dots}^T)$$

10. Return to step 2 until a convergence criterion is reached.

5. SIMULATION RESULTS

In this section, we present some results obtained by simulating a network with $Q = 3, 4$ and $K = 9$. The informative symbols were randomly generated from a QPSK alphabet. The code sequences were orthogonal binary sequences taking values from $\{-1, 1\}$. We considered three scenarios for the connection topology between the sensors in cluster B: a ring (all the sensors have a connectivity degree equals to 3), a grid (the connectivity degree are: 4, 6, 4, 6, 9, 6, 4, 6, 4), and a modified grid (3, 6, 4, 6, 7, 6, 4, 6, 4). We varied the number of consensus iterations between 1 and 3. We denoted by D-ALS(i, j), the D-ALS corresponding to the i th topology of connection with j consensus iterations. For the average consensus iterations, the weights were computed using the uniform scheme [13]. The results below are averaged values over 100 independent Monte-Carlo runs. The performance is evaluated according to the NMSE (Normalized Mean Square Error), given by: $(1/K) \sum_{k=1}^K \frac{\|\mathbf{X}_{k\dots} - \hat{\mathbf{C}}_k \text{diag}(\hat{\mathbf{A}}_k) \hat{\mathbf{S}}_k^T\|_F^2}{\|\mathbf{X}_{k\dots}\|_F^2}$.

We considered 1000 iterations of ALS. For both ALS and D-ALS, the convergence rate were greater than 90%. In Fig. 1, we depict the histogram corresponding to the number of iterations required for the algorithm convergence. We can note that the behavior of the ALS and D-ALS approaches are similar except for the case where the topology of connection in cluster B corresponds to a ring (We used three consensus iterations).

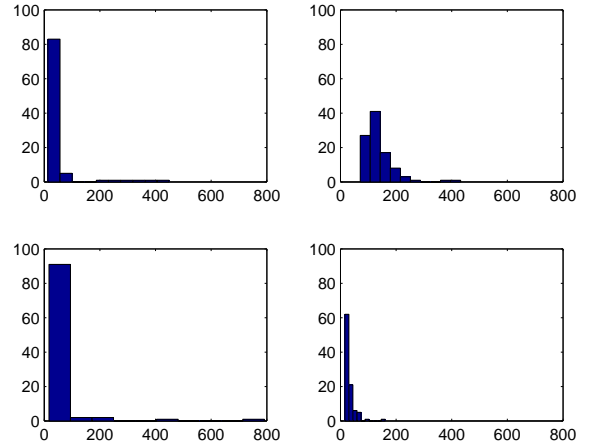


Fig. 1. Histogram of the number of iterations for convergence in the ALS (top left) and D-ALS cases: ring (top right), grid (bottom left), modified grid (bottom right).

In Figures 2 and 3, the NMSE (mean and median values) is plotted as a function of the number of iterations. It can be seen that the connection topology impacts the convergence of the D-ALS algorithm. Connection topologies with greater connectivity degree have convergence properties (speed and final value) similar to those obtained with ALS. In Fig. 4, we note that even for a single consensus iteration the D-ALS algorithm converges towards the same value than ALS. However, the convergence speed is lower. It can be accelerated by increasing the number of consensus iterations.

6. CONCLUSION

We have derived a distributed version of the ALS algorithm generally used for fitting PARAFAC or other tensor models. Resorting to a distributed scheme is of particularly importance when the receiver nodes are not equipped with an antenna array, as usually in wireless sensor networks. The proposed algorithm interleaves ALS iterations with average consensus ones. Our first evaluations of the proposed algorithm, show that for graphs with a strong connectivity a single consensus iteration can be sufficient for convergence. Future works include evaluation in noisy environments, optimality of consensus weighting matrix, asynchrony, and a deeper convergence analysis.

7. REFERENCES

- [1] R. Olfati-Saber, A. Fax, and R. Murray, "Consensus and cooperation in networked multi-agent systems," *Proc. of the IEEE*, vol. 95, no. 1, pp. 215–233, Jan. 2007.
- [2] N.D. Sidiropoulos, G.B. Giannakis, and R. Bro, "Blind PARAFAC receivers for DS-CDMA systems," *IEEE Trans. Signal Process.*, vol. 48, no. 3, pp. 810–823, March 2000.
- [3] A.L.F. de Almeida, G. Favier, and J.C.M. Mota, "PARAFAC-based unified tensor modeling for wireless communication systems with application to blind multiuser equalization," *Signal Proc.*, vol. 87, no. 2, pp. 337–351, Feb. 2007.
- [4] D. Nion and L. De Lathauwer, "An enhanced line search scheme for complex-valued tensor decompositions. Application in DS-CDMA," *Signal Proc.*, vol. 88, no. 3, pp. 749–755, March 2008.
- [5] A.Y. Kibangou and G. Favier, "Blind equalization of nonlinear channels using tensor decompositions with code/space/time diversities," *Signal Proc.*, vol. 89, no. 2, pp. 133–143, Feb. 2009.
- [6] R.A. Harshman, "Foundation of the PARAFAC procedure: models and conditions for an "explanatory" multimodal factor analysis," *UCLA work. papers phon.*, vol. 16, pp. 1–84, 1970.
- [7] L. Xiao, S. Boyd, and S.-J. Kim, "Distributed average consensus with least-mean-square deviation," *J. Parallel Distrib. Comput.*, vol. 67, pp. 33–46, 2007.
- [8] J.D. Carroll and J.J. Chang, "Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition," *Psychometrika*, vol. 35, pp. 283–319, 1970.
- [9] J.B. Kruskal, "Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics," *Lin. Algebra Appl.*, vol. 18, pp. 95–138, 1977.
- [10] G. Tomasi and R. Bro, "A comparison of algorithms for fitting the PARAFAC model," *Comp. Stat. Data Anal.*, vol. 50, no. 7, pp. 1700–1734, 2006.
- [11] G. Mateos, I.D. Schizas, and G.B. Giannakis, "Consensus-based distributed least-mean square algorithm using wireless ad hoc networks," in *Proc. Allerton Conf.*, UIUC, Illinois, USA, 2007, pp. 568–574.
- [12] S. Bolognani, S. Del Favero, L. Schenato, and D. Varagnolo, "Distributed sensor calibration and least-square parameter identification in WSNs using consensus algorithms," in *Proc. Allerton Conf.*, UIUC, Illinois, USA, 2008, pp. 1191–1198.
- [13] V.D. Blondel, J.M. Hendrickx, A. Olshevsky, and J.N. Tsitsiklis, "Convergence in multiagent coordination, consensus, and flocking," in *Proc. IEEE CDC/ECC*, Seville, Spain, Dec. 2005, pp. 2996–3000.

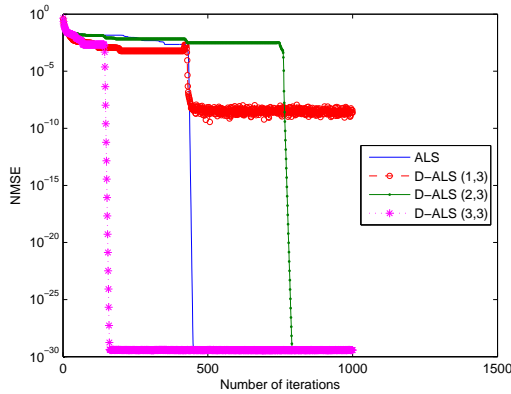


Fig. 2. Mean NMSE for $Q = 3$ and three consensus iterations.

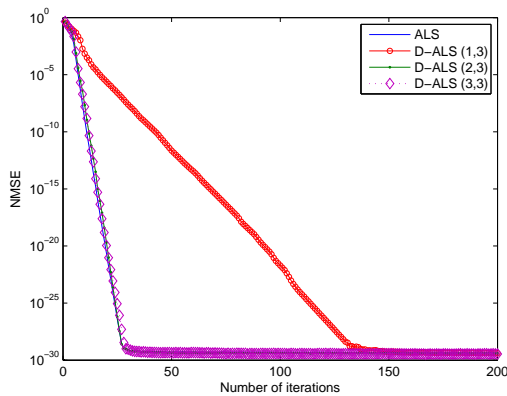


Fig. 3. Median NMSE for $Q = 3$ and three consensus iterations.

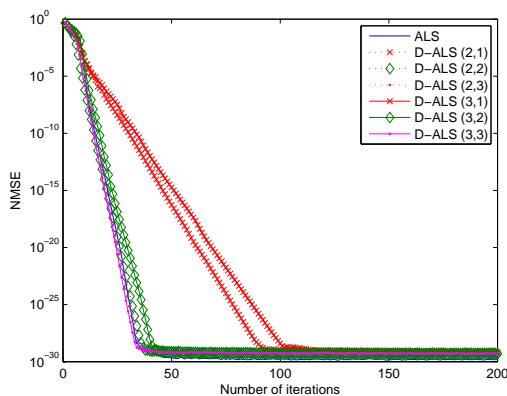


Fig. 4. Median NMSE for $Q = 4$ and different number of consensus iterations.