

Distributed Estimation over Unknown Fading Channels

Alain Y. Kibangou *

* *GIPSA-Lab, System Control Department, University Joseph Fourier, CNRS, 961 rue de la Houille Blanche, B.P. 46- 38402 Saint Martin d'Hères, France (e-mail: alain.kibangou@ujf-grenoble.fr)*

Abstract: In this paper, we consider the data detection problem involved in distributed estimation over unknown non-orthogonal fading channel. In general, when studying the distributed estimation problem, the impairments induced by communication channels are restricted to additive noise, quantization or packet loss. In addition, communication protocols are often of TDMA or FDMA type. Herein, by modulating the local data with doubly spread waveforms, we show that although each node receives a mixture of data transmitted by its neighbors, these data exhibit a trilinear structure, which can be used for separating the neighbors contributions. We state identifiability conditions and study the embedding of the data detection steps in a distributed estimation problem.

Keywords: Distributed estimation; Data detection; Fading channels; Tensor models; PARAFAC decomposition.

1. INTRODUCTION

A wireless sensor network (WSN) is constituted with nodes spatially distributed for collecting information of interest. The collected information can be aggregated at a fusion center or shared between neighbors using wireless transmissions. Each sensor in the network makes an observation of the quantity of interest, generates a local signal (either analog or digital), and then sends it to the fusion center or to other nodes in its neighborhood, where the received signals are combined to produce a final estimate of the observed quantity. Analog or digital transmission methods can be used. For the analog approach, the local observation is transmitted via analog modulation with an adequate amplification. Such a scheme is named Amplify-and-forward, Cui et al. (2007). In digital transmission, observations are quantized, digitized into bits, possibly compressed and/or encoded, and transmitted via digital modulation.

An important application of WSNs concerns distributed estimation of an unknown data using discrete-time samples collected across sensors. Such an estimation can be performed at a fusion center. However high transmission power, proportional to the covered geographic area, is required at each sensor to transmit its local information to the fusion center. In addition, such a scheme is not robust in case of fusion center failure, Schizas et al. (2008). On the contrary, WSNs whereby each sensor communicates only with its neighbors do not encounter these drawbacks and the estimation task can be performed in a totally distributed way.

Distributed estimation using WSNs is based on successive refinements of local estimates maintained at individual sensors. Each iteration of the algorithm comprises a communication step where the sensors interchange information

with their neighbors, and an update step where each sensor uses this information to refine its local estimate, Schizas et al. (2008), generally using the notion of average consensus.

Consensus averaging schemes are challenged by the presence of non ideal sensor links. Much of the literature devoted to non ideal links has focused on finite-rate transmissions of quantized sensor observations and additive noise. In addition, in most of works, the used network is the one where each sensor processes its individual measurements and transmits the result over an orthogonal multiple-access channel (MAC) to its neighbors. In such channels, collisions and interferences between nodes are avoided so that the main impairment of the communication channel concerns additive and quantization noise. Orthogonal MAC can be obtained using TDMA (Time Division Multiple Access), FDMA (Frequency Division Multiple Access) or CDMA (Code Division Multiple Access) protocols. In the first one, the time is divided into slots allocated to each node. Such a scheme induces a latency that can be crucial for some tasks (in control applications for example). In the second one, the bandwidth is divided into sub-band allocated to each user. The bandwidth being limited, scalability is a crucial question in this case. For the third scheme, each node is assigned a signaling waveform (or code) generally assumed to be orthogonal, equi-correlated and perfectly correlated with a perfect synchronization in sensor transmissions, Wimalajeewa and Jayaweera (2007). In the literature, other MAC protocols have been proposed; that is the case of Type based MAC, Mergen and Tong (2006), and likelihood-based MAC, Marano et al. (2007), both devoted to fusion center based WSNs.

In fact, in addition to noise, wireless transmissions are also subject to fading. In most works on distributed estimation, fading is in general ignored or assumed to be known. In a

recent work, a distributed estimation scheme including a channel estimation using pilot signals has been suggested, Senol and Tepedelenlioglu (2008). The derivations were made for parallel channels in a fusion center based WSN. It is necessary to point out that the number of parallel channels is limited by the bandwidth and the number of nodes in the network.

In this paper, we consider an adhoc WSN with transmissions via non-parallel channels subject to noise and fading. By introducing a particular waveform design, we show that the data samples at each sensor node constitute a three dimension array, or equivalently a third-order tensor. Therefore, at each sensor node, after data reception, the first step is to separate the contributions of its neighbors by means of a tensor decomposition.

Notations: Vectors are written as boldface lower-case letters ($\mathbf{a}, \mathbf{b}, \dots$), matrices as boldface capitals ($\mathbf{A}, \mathbf{B}, \dots$) and tensors as blackboard letters ($\mathbb{A}, \mathbb{B}, \dots$). \mathbf{I}_N denotes the $N \times N$ identity matrix. \mathbf{A}^T stands for the transpose of the matrix \mathbf{A} . $\text{diag}(\cdot)$ is the operator that forms a diagonal matrix from its vector argument. The symbol \circ denotes the vector outer product. The Kronecker product is denoted by \otimes , whereas \odot denotes the Khatri-Rao product defined for two arbitrary matrices \mathbf{A} and \mathbf{B} , having the same number of columns N , as $\mathbf{A} \odot \mathbf{B} = (\mathbf{A}_{\cdot 1} \otimes \mathbf{B}_{\cdot 1} \cdots \mathbf{A}_{\cdot N} \otimes \mathbf{B}_{\cdot N})$ where $\mathbf{A}_{\cdot i}$ (resp. $\mathbf{B}_{\cdot i}$) stands for the i th column (resp. row) of \mathbf{A} . For arbitrary matrices $\mathbf{A} \in \mathfrak{R}^{I \times K}$ and $\mathbf{B} \in \mathfrak{R}^{J \times K}$ we have the following property:

$$\mathbf{A} \odot \mathbf{B} = \begin{pmatrix} \mathbf{B} \text{diag}(\mathbf{A}_{1\cdot}) \\ \vdots \\ \mathbf{B} \text{diag}(\mathbf{A}_{I\cdot}) \end{pmatrix} \quad (1)$$

2. ESTIMATION PROBLEM

Consider the sensing model

$$\mathbf{x}^{(k)} = \boldsymbol{\theta} + \mathbf{n}^{(k)}, \quad k = 1, \dots, K \quad (2)$$

where $\boldsymbol{\theta} \in \mathfrak{R}^N$ is a real-valued unknown vector of parameters, $\mathbf{n}^{(k)}$ denotes the observation noise vector assumed to be jointly independent, and $\mathbf{x}^{(k)}$ is the measurement at the k th sensor. Without any assumption on the observation noise, we make use of the least squares estimator that corresponds to the mean of the observations. It is now well known that such an estimator can be carried out in a distributed way using average consensus.

Let $\mathcal{G} = \{\mathcal{K}, \mathcal{E}\}$ be an undirected connected graph representing the communication graph between the 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 $\hat{\boldsymbol{\theta}}_k(0)$ be a vector assigned to node k at time $t = 0$, i.e. $\hat{\boldsymbol{\theta}}_k(0) = \mathbf{x}^{(k)}$. The distributed average consensus problem consists in computing the average $(1/K) \sum_{k=1}^K \hat{\boldsymbol{\theta}}_k(0)$ 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, Xiao et al. (2007). 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:

$$\hat{\boldsymbol{\theta}}_k(t+1) = \hat{\boldsymbol{\theta}}_k(t) + \sum_{j \in \mathcal{K}_i} w_{k,j} (\hat{\boldsymbol{\theta}}_j(t) - \hat{\boldsymbol{\theta}}_k(t)), \quad (3)$$

where $w_{k,j}$ is a weight associated with the edge $\{k, j\}$ and \mathcal{K}_i is the subset of \mathcal{K} corresponding to the nodes in the neighborhood of the i th node. Asymptotic convergence of the average consensus algorithm is achieved by choosing the weights according to uniform, Laplacian, maximum degree, or Metropolis schemes.

The consensus algorithm (3) is efficient for ideal links. For noisy links, some authors proposed the use of a time-varying step-size for reducing noise accumulation, Mosquera et al. (2008); Huang and Manton (2009). For unknown and non parallel fading channels, it is necessary to first derive a data detection process between two consecutive consensus steps. Recall that in such a case, a given node receives a mixture of signals sent by its neighbors. Despite the properties of the data detector, its performance is impacted by the additive noise. Therefore, average consensus algorithms with data detection steps should be aware of data detection errors induced by the additive noise.

In additive noise, instead of using average consensus based methods, one can resort to data aggregation methods where noise accumulation is avoided. After a finite number of communication the overall information is available at each node or at a given number of nodes, heads of clusters for example. For large networks, such a scheme can induce an important latency except if efficient multiple access schemes are used. In the sequel, we derive the data detection approach for the suggested multiple access scheme.

3. DATA DETECTION

Let us consider K nodes transmitting their data, at the same time and within the same frequency band, towards their neighbors through quasi-static flat fading channels. Each node makes use of a single antenna. The k th sensor node has to transmit a sequence $\{s_{j,k}\}_{j=1, \dots, J}$ after modulation with two spreading waveforms. The modulation scheme can be viewed as a doubly spreading one, Wong and Lok (2000), i.e. the baseband signal transmitted by the k th sensor node is given by:

$$x_k(t) = \sum_{j=1}^J s_{j,k} f_k(t - jT_s), \quad (4)$$

T_s being the symbol period and $f_k(\cdot)$ the modulating waveform given by:

$$f_k(t) = \sum_{q=1}^Q b_{q,k} e_k(t - qT_f), \quad e_k(t) = \sum_{i=1}^I c_{i,k} g_k(t - iT_c)$$

$g_k(\cdot)$ being a pulse-shape filter. In the sequel, the spreading sequences $\{b_{q,k}\}$ and $\{c_{i,k}\}$, and the pulse-shape filter are

assumed to be strictly local and unknown to the other nodes of the network. Note that the informative sequence to be transmitted contains the N entries of the current local estimate $\boldsymbol{\theta}^{(k)}$ and some additional entries to be specified latter.

In the noiseless case, the baseband signal $y_l(t)$ received by the l th sensor node is given by:

$$y_l(t) = \sum_{k \in \mathcal{K}_l} \beta_{l,k} x_k(t - \tau_{l,k}), \quad (5)$$

where $\beta_{l,k}$ stands for the fading factor associated with the link between the l th and the k th sensors while $\tau_{l,k}$ denotes the associated delay that holds propagation delay and asynchronism, Nion and De Lathauwer (2008). By sampling the received signal at time instant $t = jT_s + qT_f + iT_c$, we get:

$$y_{j,q,i}^{(l)} = \sum_{k \in \mathcal{K}_l} s_{j,k} b_{q,k} h_{i,k}^{(l)} \quad (6)$$

where

$$h_{i,k}^{(l)} = \beta_{l,k} c_{i,k} g_k(t - jT_s - qT_f - iT_c - \tau_{l,k})|_{t=jT_s+qT_f+iT_c}.$$

The data in (6) can be viewed as a third-order multiway array or tensor admitting the so-called PARAFAC model, Harshman (1970).

Using vector outer product, the third-order tensor $\mathbb{Y}^{(l)}$, with $y_{j,q,i}^{(l)}$, $j = 1, \dots, J$, $i = 1, \dots, I$, $q = 1, \dots, Q$, as entries, can be written as follows:

$$\mathbb{Y}^{(l)} = \sum_{k \in \mathcal{K}_l} \mathbf{S}_{\cdot k}^{(l)} \circ \mathbf{B}_{\cdot k}^{(l)} \circ \mathbf{H}_{\cdot k}^{(l)}, \quad (7)$$

meaning that the tensor is completely characterized by the three loading, or factor, matrices $\mathbf{S}^{(l)} \in \mathbb{R}^{J \times K_l}$, $\mathbf{B}^{(l)} \in \mathbb{R}^{Q \times K_l}$, and $\mathbf{H}^{(l)} \in \mathbb{R}^{I \times K_l}$. PARAFAC model allows writing a given tensor as a sum of rank-one tensors. It can then be viewed as an analogue of the singular values decomposition without orthogonality constraints.

The data detection problem consists in retrieving the factor matrices, $\mathbf{S}^{(l)}$ in particular, given $\mathbb{Y}^{(l)}$. That is an inverse problem. For solving such a problem, uniqueness of the factor matrices is crucial.

3.1 Uniqueness conditions

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 $\{\mathbf{S}^{(l)}, \mathbf{B}^{(l)}, \mathbf{H}^{(l)}\}$ and $\{\tilde{\mathbf{S}}^{(l)}, \tilde{\mathbf{B}}^{(l)}, \tilde{\mathbf{H}}^{(l)}\}$ giving rise to the same tensor $\mathbb{Y}^{(l)}$ are linked by the following relations $\tilde{\mathbf{S}}^{(l)} = \mathbf{S}^{(l)} \mathbf{\Pi} \mathbf{\Delta}_1$, $\tilde{\mathbf{B}}^{(l)} = \mathbf{B}^{(l)} \mathbf{\Pi} \mathbf{\Delta}_2$, $\tilde{\mathbf{H}}^{(l)} = \mathbf{H}^{(l)} \mathbf{\Pi} \mathbf{\Delta}_3$, with $\mathbf{\Delta}_1 \mathbf{\Delta}_2 \mathbf{\Delta}_3 = \mathbf{I}_{K_l}$, where $\mathbf{\Pi}$ is a $K_l \times K_l$ permutation matrix whereas $\mathbf{\Delta}_1, \mathbf{\Delta}_2$, and $\mathbf{\Delta}_3$ are $K_l \times K_l$ diagonal matrices, K_l being the number of collaborating nodes in the neighborhood of the l th sensor. A sufficient condition for such an uniqueness, called Kruskal's condition, states that the PARAFAC decomposition (6) is essentially unique if, Kruskal (1977),

$$k_{\mathbf{S}^{(l)}} + k_{\mathbf{B}^{(l)}} + k_{\mathbf{H}^{(l)}} \geq 2(K_l + 1), \quad (8)$$

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 $k_{\mathbf{A}} \leq \text{rank}(\mathbf{A})$. The distinction between these two notions is important. $k_{\mathbf{A}} = r$ requires that every r columns are linearly independent, whereas $\text{rank} = r$ simply requires that there exists at least r linearly independent columns, Sidiropoulos et al. (2000).

We can note that node-wise independent fading and independent design of the pulse-shape filters imply that $\mathbf{H}^{(l)}$ is full rank and full k-rank with high probability. That is also the case for the $\mathbf{B}^{(l)}$ owing to independence of the spreading sequence $\{b_{q,k}\}$, not restricted to belong to a finite alphabet. We can therefore rewrite the Kruskal condition as follows:

$$k_{\mathbf{S}^{(l)}} + \min(Q, K_l) + \min(I, K_l) \geq 2(K_l + 1). \quad (9)$$

By setting, $Q \geq K_l$ and $I \geq K_l$, we get:

$$k_{\mathbf{S}^{(l)}} \geq 2, \quad (10)$$

meaning that the columns of $\mathbf{S}^{(l)}$ must be pairwise independent. If $\mathbf{S}^{(l)}$ only contains the estimate of the neighbors of the l th node then $\mathbf{S}^{(l)}$ could be equal to $\text{diag}(\boldsymbol{\alpha}) \mathbf{1} \mathbf{1}^T$, meaning that $k_{\mathbf{S}^{(l)}} = 1$. To prevent such a situation each node should transmit an augmented version of its local estimate by inserting some local data possibly known to its neighbors. Therefore, we set:

$$\mathbf{S}^{(l)} = \begin{pmatrix} \mathbf{z}_1 & \cdots & \mathbf{z}_{K_l} \\ \hat{\boldsymbol{\theta}}_1(0) & \cdots & \hat{\boldsymbol{\theta}}_{K_l}(0) \end{pmatrix} \quad (11)$$

with linearly independent vectors \mathbf{z}_q . For all the nodes of the network, the length of the spreading sequences should be chosen such that

$$Q \geq \max(K_l), \quad I \geq \max(K_l). \quad (12)$$

With these conditions, from the received data, each node can retrieve the informations sent by its neighbors up to columns scaling and permutation. The most penalizing ambiguity being the scaling one. Indeed, associating a given sequence to the actual sender is not necessary since we aim to compute an average value by summing the available data. The scaling ambiguity can be removed by setting the first entry to an arbitrary known value. If we assume that the nodes have an a priori knowledge of the coding sequence of their neighbors both permutation and scaling ambiguity can easily be removed.

3.2 Data detection method

Given the tensor $\mathbb{Y}^{(l)}$, various algorithms can be used for estimating the factor matrices $\mathbf{H}^{(l)}$, $\mathbf{B}^{(l)}$, and $\mathbf{S}^{(l)}$. The alternating least squares (ALS) algorithm is the most used one. It acts as follows: each time, update a subset of parameters using least squares conditioned on previously obtained estimates of the remaining parameters; proceed to update another subset of parameters; repeat until convergence. A distributed version of the ALS algorithm has been derived recently, Kibangou and de Almeida (2010).

Given noisy observations $\tilde{\mathbb{Y}}^{(l)}$, we aim to estimate the three factor matrices $\mathbf{H}^{(l)}$, $\mathbf{B}^{(l)}$, and $\mathbf{S}^{(l)}$. Let us first define different slices of the tensor obtained by fixing one index of the multiway array. We get:

$$\mathbf{Y}_{j..l} = \begin{pmatrix} y_{j,1,1}^{(l)} & \cdots & y_{j,1,I}^{(l)} \\ \vdots & \ddots & \vdots \\ y_{j,Q,1}^{(l)} & \cdots & y_{j,Q,I}^{(l)} \end{pmatrix} = \mathbf{B}^{(l)} \text{diag}(\mathbf{S}_j^{(l)}) \mathbf{H}^{(l)T} \in \mathbb{R}^{Q \times I}$$

$$\mathbf{Y}_{.q..l} = \begin{pmatrix} y_{1,q,1}^{(l)} & \cdots & y_{J,q,1}^{(l)} \\ \vdots & \ddots & \vdots \\ y_{1,q,I}^{(l)} & \cdots & y_{J,q,I}^{(l)} \end{pmatrix} = \mathbf{H}^{(l)} \text{diag}(\mathbf{B}_q^{(l)}) \mathbf{S}^{(l)T} \in \mathbb{R}^{I \times J}$$

$$\mathbf{Y}_{..i,l} = \begin{pmatrix} y_{1,1,i}^{(l)} & \cdots & y_{1,Q,i}^{(l)} \\ \vdots & \ddots & \vdots \\ y_{J,1,i}^{(l)} & \cdots & y_{J,Q,i}^{(l)} \end{pmatrix} = \mathbf{S}^{(l)} \text{diag}(\mathbf{H}_i^{(l)}) \mathbf{B}^{(l)T} \in \mathbb{R}^{J \times Q}$$

By concatenating slices of the same mode, we get the following three unfolded matrices:

For a third-order tensor $\mathbb{Y}^{(l)}$, there are three unfolded matrices: $\mathbf{Y}_{1,l} \in \mathbb{R}^{JQ \times I}$, $\mathbf{Y}_{2,l} \in \mathbb{R}^{QI \times J}$, and $\mathbf{Y}_{3,l} \in \mathbb{R}^{IJ \times Q}$ defined as:

$$\mathbf{Y}_{1,l} = \begin{pmatrix} \mathbf{Y}_{1..,l} \\ \vdots \\ \mathbf{Y}_{J..,l} \end{pmatrix} = (\mathbf{S}^{(l)} \odot \mathbf{B}^{(l)}) \mathbf{H}^{(l)T} \in \mathbb{R}^{JQ \times I}, \quad (13)$$

$$\mathbf{Y}_{2,l} = \begin{pmatrix} \mathbf{Y}_{.1..,l} \\ \vdots \\ \mathbf{Y}_{.Q..,l} \end{pmatrix} = (\mathbf{B}^{(l)} \odot \mathbf{H}^{(l)}) \mathbf{S}^{(l)T} \in \mathbb{R}^{QI \times J}, \quad (14)$$

$$\mathbf{Y}_{3,l} = \begin{pmatrix} \mathbf{Y}_{..1,l} \\ \vdots \\ \mathbf{Y}_{..I,l} \end{pmatrix} = (\mathbf{H}^{(l)} \odot \mathbf{S}^{(l)}) \mathbf{B}^{(l)T} \in \mathbb{R}^{IJ \times Q}. \quad (15)$$

When the noise is modeled as temporally and spatially white Gaussian, the maximum likelihood estimation and the least squares fitting result on:

$$\min_{\mathbf{H}^{(l)}, \mathbf{B}^{(l)}, \mathbf{S}^{(l)}} \left\| \tilde{\mathbf{Y}}_{1,l} - (\mathbf{S}^{(l)} \odot \mathbf{B}^{(l)}) \mathbf{H}^{(l)T} \right\|_F^2$$

$\tilde{\mathbf{Y}}_{n,l}$, $n = 1, 2, 3$, being the noisy version of $\mathbf{Y}_{1,l}$ and $\|\cdot\|_F$ denoting the Frobenius norm. It follows that the conditional least squares update of $\mathbf{H}^{(l)}$ is

$$\mathbf{H}^{(l)} = \left(\tilde{\mathbf{Y}}_{1,l} (\mathbf{S}^{(l)} \odot \mathbf{B}^{(l)})^\dagger \right)^T$$

where \dagger stands for the matrix pseudo-inverse.

Similarly, the conditional least squares updates of the two other matrices are given by

$$\mathbf{S}^{(l)} = \left(\tilde{\mathbf{Y}}_{2,l} (\mathbf{B}^{(l)} \odot \mathbf{H}^{(l)})^\dagger \right)^T$$

$$\mathbf{B}^{(l)} = \left(\tilde{\mathbf{Y}}_{3,l} (\mathbf{H}^{(l)} \odot \mathbf{S}^{(l)})^\dagger \right)^T.$$

As stated by Sidiropoulos et al. (2000), the conditional update of any given matrix may either improve or maintain but cannot worsen the current fit. Global monotone convergence to (at least) a local minimum follows directly from this observation. The convergence speed can be accelerated if the node knows the spreading sequence of its neighbors or by using extended line search methods, Rajih et al. (2008).

After convergence, the node can recover the data contained in $\mathbf{S}^{(l)}$ for improving its own estimate. Note that a scaling correction should be achieved before recovering the useful data.

4. DISTRIBUTED ESTIMATION SCHEMES

4.1 Average consensus in noiseless links

As stated previously, estimating the average value over the network can be carried out by means of the average consensus algorithm. In the framework proposed herein, a data detection step is included between two average consensus step. We illustrate the behavior of the proposed method by considering first a small network of 11 nodes described by the graph of Fig. 1. Then we will consider a larger network. For both networks, the data vector to be estimated is $\boldsymbol{\theta} = (2, 4, -1)^T$. The variance of the local estimates is set to 0.001.

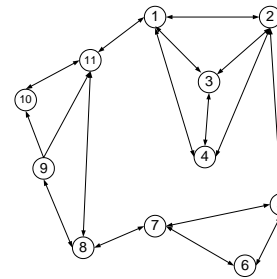


Fig. 1. Connection topology of the simulated network

Figure 2 depicts the average consensus based estimation over an unknown channel in the noiseless case, for the first network. The Weights were computed as $\mathbf{W} = \mathbf{I} - \epsilon \mathbf{L}$, where \mathbf{L} denotes the Laplacian matrix and $\epsilon = 0.2$. The

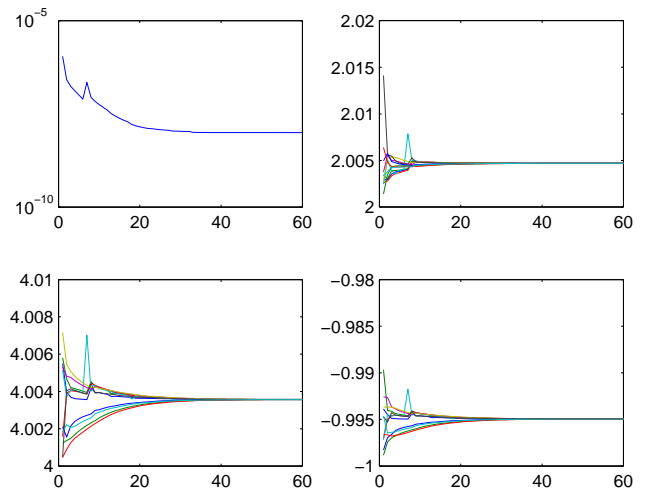


Fig. 2. Average consensus over unknown fading channel in the noiseless case: Normalized mean square error (top left), estimated average values (top right, bottom)

stop criterion for the ALS algorithm used for the data detection method was the number of iterations set to 1000. As depicted in Fig. 2, the data detection method

does not induce significant detection errors. However, we can note, for one node, a brutal variation due to a data detection error. Despite such an error, all the nodes converge to the same value very close to the actual mean. In fact, the convergence property of the consensus algorithm remains valid owing to a very low probability of detection error. Note that for this network the factor matrices were randomly initialized at each data detection step.

Now, we consider a network with 100 nodes distributed in a 10×10 grid. The communication between two nodes was possible when the distance between them was less than unity. For the first consensus iteration, the data detection method was initialized with random matrices, from a Gaussian distribution. Then, the estimated matrices were used for initializing the data detection of the next step. For the random initialization part, 20 independent initializations were run and then the best one was selected in the tensor reconstruction error sense. The ALS algorithm were stopped after 2000 iterations. In Fig. 3 and Fig. 4 we can note that the behavior of the average consensus algorithm is not disturbed by the data detection method. However, a deeper performance analysis should be carried out in further works.

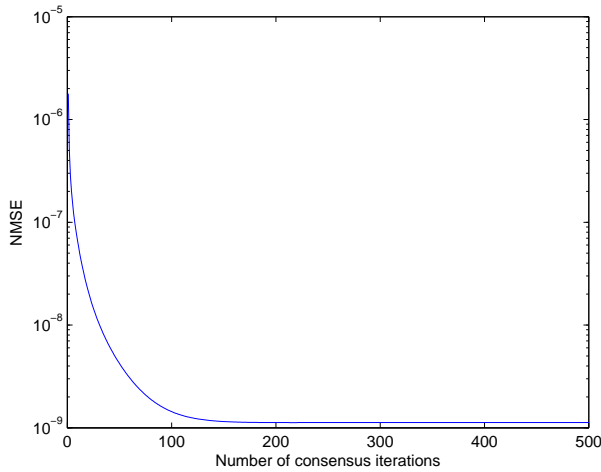


Fig. 3. Normalized Mean Square error of a consensus based distributed estimation (case of a 100 nodes network)

4.2 Distributed estimation in cluster-based sensor networks

In this subsection, we consider the estimation problem in a cluster-based sensor network (see Fig. 5). We assume that the network is cast in clusters. Inside a given cluster, each node is connected to the head of the cluster. In addition, the head of the clusters are all connected. We assume that the communication protocol is divided in three slots in time. A sub-band of frequencies is allocated to each cluster. Inside a given cluster the nodes can communicate simultaneously owing to the MAC scheme proposed in this paper. The head of cluster compute the mean of the cluster and then sends it to the other heads of cluster. The heads of clusters compute the mean of the overall network. And at the last step, each head communicates the average value to the other nodes of the cluster. The communication protocol is described in Fig. 6.

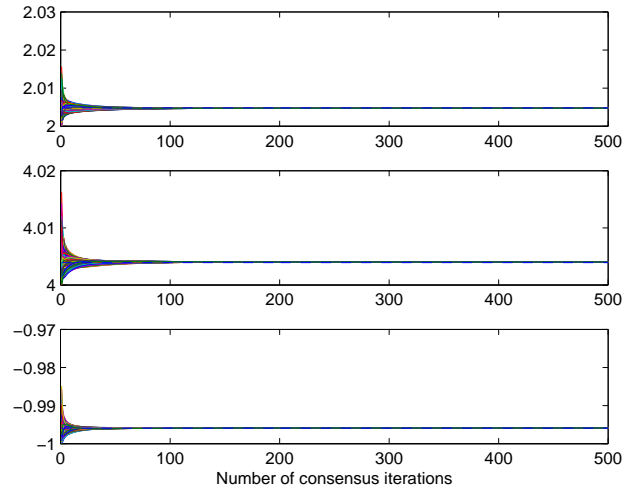


Fig. 4. Distributed parameter estimation using the average consensus over unknown fading channel (case of a 100 nodes network)

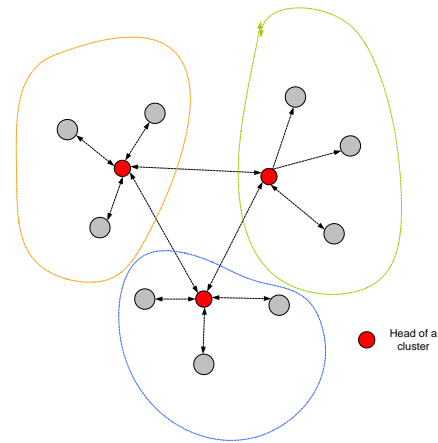


Fig. 5. Cluster based sensor network

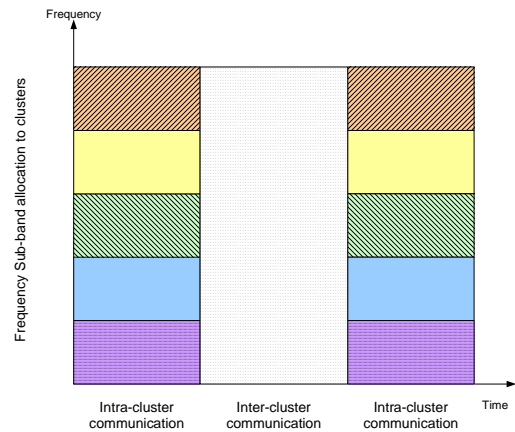


Fig. 6. Time-frequency bandwidth allocation

In table 1, we give some simulation results obtained for a network with 30 nodes and 5 clusters; each cluster having the same number of nodes. As in the previous case, for a noiseless channel, we get the actual average value, in mean. The results presented here being averaged values over 50 independent runs. The additive noise was a Gaussian one. Random initialization was used for data detection and the ALS algorithm was stopped after 1000 iterations.

Table 1. Estimation in a cluster based sensor network

Actual average	Estimated average value
2.003	2.003
4.005	4.004
-2.996	-2.996

Fig. 7 depicts the NMSE according to the SNR value. We can note a severe degradation of the NMSE for low and medium SNR values. The initialization scheme and the stopping criterion of the ALS algorithm are certainly the cause of such degradation. Performance can certainly be improved by resorting to other initialization strategies or to other PARAFAC fitting methods.

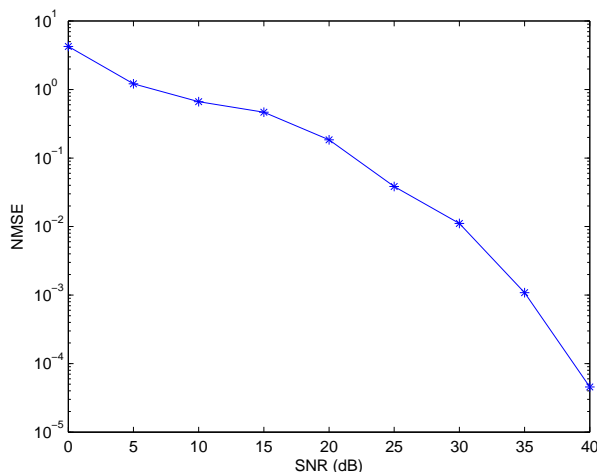


Fig. 7. NMSE for a cluster-based sensor network.

5. CONCLUSION

We have presented a new scheme for distributed estimation over unknown fading channel. Non orthogonal multiple access fading channels have been considered. By using doubly spread waveforms for modulating the local data, the nodes can communicate simultaneously. Although the sensor node receives a mixture of the data sent by its neighbors, these data exhibit a trilinear structure that can be used for separating the data. Such a separation can be carried out using the so-called PARAFAC decomposition. Based on the Kruskal's condition, we have derived sufficient conditions ensuring identifiability or uniqueness of the detected data. Therefore, the proposed data detection can be embedded in distributed estimation methods such as average consensus. However, the detection error induced by the channel noise can degrade the overall estimation process. A further analysis of the performance of the proposed data detection is still under investigation.

ACKNOWLEDGEMENT

This work was supported by the EU project FeedNetBack.

REFERENCES

- S. Cui, J.-J. Xiao, A. Goldsmith, Z.-Q. Luo, and V. Poor. Estimation diversity and energy efficiency in distributed sensing. *IEEE Trans. on Signal Proc.*, 55(9):4683–4695, September 2007.
- R.A. Harshman. Foundation of the PARAFAC procedure: models and conditions for an "explanatory" multimodal factor analysis. *UCLA working papers in phonetics*, 16: 1–84, 1970.
- M. Huang and J. Manton. Coordination and consensus of networked agents with noisy measurements: stochastic algorithms and asymptotic behavior. *SIAM J. Control Optim.*, 48(1):134–161, 2009.
- A.Y. Kibangou and A.L.F. de Almeida. Distributed PARAFAC based DS-CDMA blind receiver for wireless sensor networks. In *Proc. of IEEE Int. Workshop on Signal Processing Advances for Wireless Comm. (SPAWC)*, Marrakech, Morocco, June 20-23 2010.
- J.B. Kruskal. Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics. *Linear Algebra Applicat.*, 18: 95–138, 1977.
- A. Marano, V. Matta, L. Tong, and P. Willett. Bandwidth scaling for efficient inference over a power-limited MAC. In *Proc. of ICASSP*, Honolulu, HI, USA, 2007.
- G. Mergen and L. Tong. Type based estimation over multiaccess channels. *IEEE Trans. on Signal Proc.*, 54(2):613–626, February 2006.
- C. Mosquera, R. Lopez-Valcarce, and S.K. Jayaweera. Distributed estimation with noisy exchanges. In *Proc. of SPAWC*, pages 236–240, Recife, Brazil, July 2008.
- D. Nion and L. De Lathauwer. An enhanced line search scheme for complex-valued tensor decompositions. Application in DS-CDMA. *Signal Processing*, 88(3):749–755, March 2008.
- M. Rajih, P. Comon, and R. Harshman. Enhanced line search : a novel method to accelerate PARAFAC. 30(3): 1148–1171, September 2008.
- I. Schizas, A. Ribeiro, and G. Giannakis. Consensus in ad hoc WSNs with noisy links-part I: Distributed estimation of deterministic signals. *IEEE Trans. on Signal Proc.*, 56(1):350–364, January 2008.
- H. Senol and C. Tepedelenlioglu. Performance of distributed estimation over unknown parallel fading channels. *IEEE Trans. on Signal Proc.*, 56(12):6057–6068, December 2008.
- N.D. Sidiropoulos, G.B. Giannakis, and R. Bro. Blind PARAFAC receivers for DS-CDMA systems. *IEEE Trans. on Signal Processing*, 48(3):810–823, March 2000.
- T. Wimalajeewa and S.K. Jayaweera. Power efficient distributed estimation in a bandlimited wireless sensor network. In *Proc. of 41st Asilomar Conf. on Signals, Systems and Computers*, Pacific Grove, CA, USA, 2007.
- T.F. Wong and T.M. Lok. Doubly spread DS-CDMA for efficient blind interference cancellation. *IEE Proc. Commun.*, 147(5):299–304, 2000.
- L. Xiao, S. Boyd, and S.-J. Kim. Distributed average consensus with least-mean-square deviation. *J. Parallel Distrib. Comput.*, 67:33–46, 2007.