DISTRIBUTED KHATRI-RAO SPACE-TIME CODING AND DECODING FOR COOPERATIVE NETWORKS

Alain Y. Kibangou[†] *and Andre L.F. de Almeida*[‡]

[†] GIPSA-Lab, University Joseph Fourier, CNRS
961 rue de la Houille blanche,B.P. 46, F- 38402 Grenoble Cedex, France
phone: +(33) 4 76 82 64 51, e-mail: alain.kibangou@ujf-grenoble.fr

[‡] Department of Teleinformatics Engineering, Federal University of Ceará
CP 6005, 60455-760, Fortaleza, Brazil
phone: +(55) 85 3262- 4283, email: andre@gtel.ufc.br

ABSTRACT

Khatri-Rao space-time (KRST) coding is an efficient scheme proposed in the last decade for multi-antenna communication systems. The decoding process results on a tensor decomposition with interesting blind decodability properties. In this paper, we consider its extension to cooperative networks where each node has a single antenna. For this purpose, we propose a distributed KRST coding and decoding technique. The proposed distributed decoding scheme is based on average consensus embedded in an alternating least squares (ALS) algorithm. Unlike standard consensus algorithms where consensus is reached asymptotically, we derive closed form solutions allowing to reach the consensus in a finite number of iterations upper-bounded by the number of collaborating nodes. The performance of the proposed method is evaluated by means of simulations.

1. INTRODUCTION

In wireless communication systems, spatial diversity plays a key role in combating signal fading arising from multipath propagation. As long as the transmitter is equipped with multiple antennas, it is well known that spatial diversity can be exploited further at the transmitter by means of space-time coding [1]. In contrast to conventional (single-user) space-time coding/decoding, when dealing with cooperative wireless networks, spatial diversity must resort to distributed space-time coding/decoding, where a collection of distributed antennas belonging to multiple terminals work in a coordinated way to encode/decode the transmitted information [2].

In the last decade, new signal processing techniques based on tensor modeling have been developed for blind channel estimation and information recovery [3, 4, 5, 6]. In [7], a Khatri-Rao space-time (KRST) coding technique based on tensor modeling has been exploited to design space-time codes with variable multiplexing-diversity tradeoff and built-in blind detection property. Therein, the authors consider the use of an antenna array physically mounted at both the transmitter and/or the receiver. By combining tensor modeling with the idea of cooperative signal processing, the authors have shown in [8] that blind information recovery can be carried out in a distributed way. The work derived a distributed alternating least squares algorithm based on average consensus, for blind channel and symbol estimation in a direct-sequence code division multiple access (DS-CDMA) wireless sensor network. In a consensus problem, a group of network nodes try to reach agreement on a given quantity of interest that depends on their local values [9]. By using linear iterations, it is now well known that consensus can be asymptotically reached given some conditions on the graph modeling the interactions between nodes.

Alain Y. Kibangou is partially supported by the EU Project FeedNetBack. André L.F. de Almeida is partially supported by CNPq/Brazil (Grant no. 303238/2010-0)

In this paper, we consider the problem of space-time coding and decoding in a wireless cooperative network and propose a distributed KRST coding technique combined with a finite-time average consensus algorithm for distributed blind KRST decoding and channel estimation.

Notations: Vectors are written as boldface lower-case letters ($\mathbf{a}, \mathbf{b}, \cdots$) and matrices as boldface capitals ($\mathbf{A}, \mathbf{B}, \cdots$). $\mathbf{A}_{i.}$ and $\mathbf{A}_{.j}$ denote respectively the ith row and the jth column of the $I \times J$ matrix $\mathbf{A}$. $\mathbf{A}^T$, $\mathbf{A}^H$, and $\mathbf{A}^\dagger$ stand respectively for the transpose, the complex conjugate, and the Moore-Penrose pseudoinverse of $\mathbf{A}$. $\mathbf{J}_K = \mathbf{1}\mathbf{1}^T$ and $\mathbf{I}_K$ denote respectively $K \times K$ all ones matrix and an identity matrix. $diag(.)$ is the operator that forms a diagonal matrix from its vector argument whereas $vec(.)$ 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:

$$\mathbf{X} \odot \mathbf{Y} = \begin{pmatrix} \mathbf{Y} diag(\mathbf{X}_{1.}) \\ \mathbf{Y} diag(\mathbf{X}_{2.}) \\ \vdots \\ \mathbf{Y} diag(\mathbf{X}_{I.}) \end{pmatrix} \in \mathbb{C}^{IJ \times R}. \quad (1)$$

2. COOPERATIVE TRANSMISSION

Let us consider a wireless network with a set of transmitting terminals denoted $\mathscr{S} = \{1, 2, \cdots, N\}$. A given transmitting source terminal $s \in \mathscr{S}$ has information to transmit to a single destination terminal denoted $d(s) \notin \mathscr{S}$, potentially using terminals $\mathscr{S} - \{s\}$ as relays. We consider an amplify-and-forward based transmission protocol that consists in two phases [10]. In the first phase, the source transmits a coded information to the relays using orthogonal channels. In the second phase, both the source and the relays simultaneously transmit to the destination. In such a way, the source terminal and the relays constitute a virtual transmitting antenna array.

We consider block transmissions where the data stream is first parsed into vectors. Let $\mathbf{s}_t = \begin{pmatrix} s_{1,t} & \cdots & s_{M,t} \end{pmatrix}^T \in \mathbb{C}^{M \times 1}$, $t = 1, \cdots, T$, be the data vectors to be transmitted at time instant t. First, the source terminal encodes the data as follows:

$$\bar{s}_{n,t} = \mathbf{A}_{n.} \mathbf{s}_t$$

where the coding matrix $\mathbf{A} = \begin{pmatrix} \mathbf{A}_{1.}^T & \cdots & \mathbf{A}_{N.}^T \end{pmatrix}^T \in \mathbb{C}^{N \times M}$ is unknown by the relays but known to the destination node. Then the encoded data $\bar{s}_{n,t}$ is transmitted to the nth relay, $n = 2, 3, \cdots, N$, $n = 1$ corresponding to the source terminal itself.

The nth relay node, independently from the others, amplifies, with repetition, the received data and transmits the data vector

$$\bar{\mathbf{c}}_{n,t} = \bar{s}_{n,t} \mathbf{C}_{.n} \in \mathbb{C}^{L \times 1}.$$

Without lack of generality, we assume, for each time block t, the first transmissions are not amplified. Therefore the first row of $\mathbf{C}$ is an all ones vector.

In matrix form, the virtual antenna array formed by the N relay nodes transmits the data matrix

$$\bar{\mathbf{C}}_t = \begin{pmatrix} \bar{\mathbf{c}}_{1,t}^T \\ \vdots \\ \bar{\mathbf{c}}_{N,t}^T \end{pmatrix} = diag(\mathbf{A}\mathbf{s}_t)\mathbf{C}^T. \quad (2)$$

Such a scheme corresponds to a Khatri-Rao Space-Time (KRST) encoding through a virtual antenna array. KRST was introduced in [7] for multi-antenna communication systems. It exploits potentialities of tensors and multilinear algebra.

3. COOPERATIVE AND DISTRIBUTED DECODING

As for the transmission front-end, the destination node belongs to a set of nodes $\mathscr{K} = \{1,\cdots,K\}$, $\mathscr{K} \bigcap \mathscr{S} = \emptyset$, that collaborate. The interactions between these nodes are modeled by means of an undirected connected graph $\mathscr{G} = \{\mathscr{K},\mathscr{E}\}$ where $\mathscr{E}$ denotes the edge set, each edge $\{i,j\} \in \mathscr{E}$ being an unordered pair of distinct nodes.

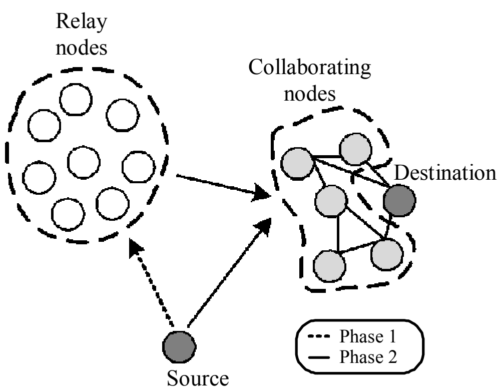

Figure 1: Cooperative communication system.

Let us assume that the communication channel is flat-fading and constant during block-time of length LT. We denote by $\mathbf{H} \in \mathbb{C}^{K\times N}$ the channel matrix between the N transmitting nodes and the K receiving ones. We can then write the received data matrix $\mathbf{Y}_{..t} \in \mathbb{C}^{K\times L}$ as:

$$\mathbf{Y}_{..t} = \mathbf{H}diag(\mathbf{A}\mathbf{s}_t)\mathbf{C}^T = \mathbf{H}diag(\mathbf{B}_{t.})\mathbf{C}^T, \quad t = 1,\cdots,T. \quad (3)$$

with $\mathbf{B}^T = \mathbf{AS}$, $\mathbf{S} = \begin{pmatrix} \mathbf{s}_1 & \cdots & \mathbf{s}_T \end{pmatrix}$. The matrices $\mathbf{Y}_{..t}$, $t = 1,\cdots,T$ can be viewed as slices of a third-order tensor $\mathbb{Y} \in \mathbb{C}^{K\times L\times T}$ along the third mode. Equivalently, we can define the slices along the first and the second modes as follows:

$$\mathbf{Y}_{k..} = \mathbf{C}diag(\mathbf{H}_{k.})\mathbf{B}^T, \quad k = 1,\cdots,K. \quad (4)$$

$$\mathbf{Y}_{.l.} = \mathbf{B}diag(\mathbf{C}_{l.})\mathbf{H}^T, \quad l = 1,\cdots,L. \quad (5)$$

The aim of the blind decoding process is to recover the $M\times T$ data matrix $\mathbf{S}$ solely from the received data tensor $\mathbb{Y}$. In fact, from (3) we can deduce that $\mathbb{Y}$ admits a CP model, also known as PARAFAC model [11], whose factor matrices are $\mathbf{H}$, $\mathbf{B}$, and $\mathbf{C}$. These factor matrices are essentially unique if [12, 3]

$$k_{\mathbf{H}} + k_{\mathbf{B}} + k_{\mathbf{C}} \geq 2(N+1), \quad (6)$$

where $k_{\mathbf{H}}$ denotes the Kruskal-rank, also called k-rank, of $\mathbf{H}$, i.e. the greatest integer $k_{\mathbf{H}}$ such that any set of $k_{\mathbf{H}}$ columns of $\mathbf{H}$ is independent. Recall that essential uniqueness means that each factor matrix can be determined up to column scaling and permutation.

Assuming that the communication system fulfilled the above condition, we derive in the sequel the associated decoding algorithm. In fact, the decoding process has two main phases. It begins by fitting the CP model, i.e. estimating the factor matrices $\mathbf{H}$, $\mathbf{C}$, and $\mathbf{B}$, and then, knowing $\mathbf{A}$, the information matrix $\mathbf{S}$ is deduced from the encoded information matrix $\mathbf{B}$.

3.1 Cooperative decoding using the Distributed Alternating Least Squares algorithm

Each node, numbered k, has at its disposal a matrix $\mathbf{Y}_{k..}$ containing linear combinations of the entries of the information matrix $\mathbf{B}$. Since $\mathbf{C}$ and the channel parameters are unknown, retrieving $\mathbf{B}$ needs to carry out a bilinear decomposition, which is non unique in general. By viewing $\mathbf{Y}_{k..}$ as a slice of a third-order tensor admitting a CP decomposition, the information matrix can be uniquely retrieved if all the slices are available at the destination node. Fitting the CP model is generally carried out using the Alternating Least Squares (ALS) algorithm. It acts by alternately minimizing the following cost functions in the Least Squares (LS) sense

$$J_K = \left\|\mathbf{Y}_K - (\mathbf{H}\odot\mathbf{C})\mathbf{B}^T\right\|_F^2 \quad (7)$$

$$J_T = \left\|\mathbf{Y}_T - (\mathbf{B}\odot\mathbf{H})\mathbf{C}^T\right\|_F^2 \quad (8)$$

$$J_L = \left\|\mathbf{Y}_L - (\mathbf{C}\odot\mathbf{B})\mathbf{H}^T\right\|_F^2. \quad (9)$$

The matrices $\mathbf{Y}_K \in \mathbb{C}^{KL\times T}$, $\mathbf{Y}_L \in \mathbb{C}^{LT\times K}$, and $\mathbf{Y}_T \in C^{TK\times L}$, called unfolded matrices, are obtained by concatenating slices of the same type, i.e. $\mathbf{Y}_K = \begin{pmatrix} \mathbf{Y}_{1..}^T & \cdots & \mathbf{Y}_{K..}^T \end{pmatrix}^T$ for example.

In a cooperative framework, one can take advantage of uniqueness of CP by sending the matrices $\mathbf{Y}_{k..}$ to the destination node where collected data can be cast into a tensor $\mathscr{Y}$. However, such a scheme requires additional storage ressources for the destination node. Recently, the authors have proposed a distributed version of the ALS algorithm (D-ALS) [8]. With D-ALS, there is no need to resort to additional storage capacities and all the collaborating nodes compute exactly the same CP factor matrices. However, the full decoding process can only be achieved by the destination node, the coding matrix $\mathbf{A}$ being unknown to the other nodes.

The main idea in D-ALS is to locally estimate the rows of $\mathbf{H}$ at each node, while distributively estimate $\mathbf{B}$ and $\mathbf{C}$ in the network, by computing respectively the LS solutions of (7) and (8):

$$\hat{\mathbf{B}}^T = \mathbf{\Theta}^{-1}\mathbf{\Gamma}, \quad (10)$$

$$\hat{\mathbf{C}}^T = \mathbf{\Phi}^{-1}\mathbf{\Psi}, \quad (11)$$

where $\mathbf{\Theta} = \frac{1}{K}\mathbf{X}^H\mathbf{X}$, $\mathbf{\Gamma} = \frac{1}{K}\mathbf{X}^H\mathbf{Y}_K$, $\mathbf{X} = \mathbf{H}\odot\mathbf{C}$, $\mathbf{\Phi} = \frac{1}{K}\mathbf{Z}^H\mathbf{Z}$, $\mathbf{\Psi} = \frac{1}{K}\mathbf{Z}^H\mathbf{\Pi}\mathbf{Y}_T$, $\mathbf{Z} = \mathbf{H}\odot\mathbf{B}$. Note that $\mathbf{\Pi}$ is a permutation matrix such that $\mathbf{Z} = \mathbf{\Pi}(\mathbf{B}\odot\mathbf{H})$.

Given $\mathbf{C}$ and $\mathbf{B}$, minimizing J_K is equivalent to minimize

$$\sum_{k=1}^{K}\left\|vec(\mathbf{Y}_{k..}^T) - (\mathbf{C}\odot\mathbf{B})\mathbf{H}_{k.}^T\right\|_2^2$$

(see [8] for detailed derivations). As a consequence, the estimation of the local channel parameters is given by:

$$\mathbf{H}_{k.}^T = (\mathbf{C}\odot\mathbf{B})^\dagger vec(\mathbf{Y}_{k..}^T). \quad (12)$$

$\mathbf{C}$ and $\mathbf{B}$ being global parameters, their estimation needs more collaboration between the nodes. From the definition of the Khatri-Rao product, we can write:

$$\mathbf{X} = \begin{pmatrix} \mathbf{X}_1 \\ \vdots \\ \mathbf{X}_K \end{pmatrix} = \begin{pmatrix} \mathbf{C}diag(\mathbf{H}_{1.}) \\ \vdots \\ \mathbf{C}diag(\mathbf{H}_{K.}) \end{pmatrix}$$

Hence, the matrices $\mathbf{\Theta}$ and $\mathbf{\Gamma}$ can be obtained through averaging of some local quantities:

$$\mathbf{\Theta} = \frac{1}{K}\sum_{k=1}^{K}\mathbf{X}_k^H\mathbf{X}_k, \quad \mathbf{\Gamma} = \frac{1}{K}\sum_{k=1}^{K}\mathbf{X}_k^H\mathbf{Y}_{k..}$$

Similarly, we can write:

$$\mathbf{Z} = \mathbf{H}\odot\mathbf{B} = \begin{pmatrix}\mathbf{Z}_1\\ \mathbf{Z}_2\\ \vdots\\ \mathbf{Z}_K\end{pmatrix} = \begin{pmatrix}\mathbf{B}diag(\mathbf{H}_{1.})\\ \vdots\\ \mathbf{B}diag(\mathbf{H}_{K.})\end{pmatrix}.$$

We get:

$$\mathbf{\Phi} = \frac{1}{K}\sum_{k=1}^{K}\mathbf{Z}_k^H\mathbf{Z}_k, \quad \mathbf{\Psi} = \frac{1}{K}\sum_{k=1}^{K}\mathbf{Z}_k^H\mathbf{Y}_{k..}^T.$$

The computation of such averaged values in a network can be achieved by using a consensus approach.

3.2 Average consensus in a network

In a consensus problem, a group of network nodes try to reach agreement on a given quantity of interest that depends on their states [9]. Consensus ensures that the action taken by any node in the network is consistent with that of its neighbors. For example, suppose that we need to compute the matrix $\mathbf{\Omega}$ as the average of the local matrices $\mathbf{R}_k$, i.e. $\mathbf{\Omega} = \frac{1}{K}\sum_{k=1}^{K}\mathbf{R}_k$. Average consensus can be reached using a linear iteration scheme where each node repeatedly updates its value as a weighted linear combination of its own value and those of its neighbors:

$$\mathbf{\Omega}_k(t+1) = w_{kk}\mathbf{\Omega}_k(t) + \sum_{q\in\mathscr{N}_k} w_{kq}\mathbf{\Omega}_q(t), \quad \mathbf{\Omega}_k(0) = \mathbf{R}_k.$$

$\mathscr{N}_k$ being the neighborhood of the kth node, i.e. nodes communicating directly with k.

Several algorithms have been proposed in the literature based on such a scheme. However, in the majority of the proposed algorithms the weights w_{kq} are chosen so that all the nodes asymptotically converge to the same value, i.e. $\lim_{t\to\infty}\mathbf{\Omega}_k(t) = \mathbf{\Omega} = \frac{1}{K}\sum_{k=1}^{K}\mathbf{R}_k$, that is achieved if the weight matrix $\mathbf{W}$, with w_{kq} as entries, is such that $\lim_{t\to\infty}\mathbf{W}^t = \frac{1}{K}\mathbf{J}_K$.

It is now well known that, in the noiseless case[1], consensus is achieved if and only if the weight matrix has a simple eigenvalue at 1 and all other eigenvalues have magnitude strictly less than 1, the left and right eigenvectors of $\mathbf{W}$ associated with the eigenvalue 1 being $\frac{1}{K}\mathbf{1}$ and $\mathbf{1}$ [13]. The above conditions guarantee asymptotic convergence.

In D-ALS, consensus are steps between ALS iterations. Therefore, the consensus algorithm should be stopped after a finite number of iterations as done in [8]. Obviously, the number of consensus iterations used in D-ALS impacts strongly the convergence of the overall algorithm. In fact, asymptotic convergence is not suitable in such a case. Even though, speed convergence of consensus algorithm have been explored [13, 14] in order to derive fast consensus algorithms. Errors due to running consensus algorithms in finite-time are in general not quantifiable. Therefore, it is interesting to address the question of exact consensus in finite-time.

A number of authors have studied finite-time consensus in the framework of discrete-time systems. For example, in [15], it has been shown that each node can calculate the consensus value as a linear combination of its own past values over at most D time-steps, D being the degree of the minimal polynomial of the associated weight matrix. In [16], based on properties of de Bruijn's graph and block Kronecker product, it has been shown that the average consensus problem can be reached in finite time if the number of nodes is an exact power of the maximum in-degree of the graph. In this paper, for time-invariant topologies, we show that the finite-time average consensus problem can be solved as a matrix factorization problem with joint diagonalizable matrices.

[1]We restrict our study to perfect data exchanges in the network of receiving nodes.

3.3 Finite time average consensus

Our goal is to find a set of matrices $\{\mathbf{W}_i\}_{i=1,\cdots,D}$, with (i,j)th entries equal to zero if $j\notin\mathscr{N}_i$, so that

$$\prod_{i=1}^{D}\mathbf{W}_i = \frac{1}{K}\mathbf{J}_K. \quad (13)$$

Finding this set of matrices is equivalent to solve a multivariate polynomial system of equations. In general, studying the existence of solutions to such a system of equations can be untractable. However, by assuming that the matrices $\mathbf{W}_i$ are jointly diagonalizable; i.e. it exists an orthogonal matrix $\mathbf{U}$ that diagonalizes the matrices $\mathbf{W}_i$, we get:

$$\mathbf{W}_i = \mathbf{U}\mathbf{E}_i\mathbf{U}^T, \quad \mathbf{U}^T\mathbf{U} = \mathbf{I}_K$$

$\mathbf{E}_i$ being a diagonal matrix. We can therefore rewrite (13) as:

$$\mathbf{U}\left(\prod_{i=1}^{D}\mathbf{E}_i\right)\mathbf{U}^T = \frac{1}{K}\mathbf{J}_K. \quad (14)$$

In the sequel, we define a set of matrices $\mathbf{W}_i$ jointly diagonalizable so that solutions of equation (14) be tractable. Assuming that $N_{max} = \max\{N_1,\cdots,N_K\}$ is known, N_i being the cardinal of the neighborhood of i, we define the matrix $\tilde{\mathbf{A}} = N_{max}\mathbf{I} - \mathbf{L}$, where $\mathbf{L}$ denotes the Laplacian matrix of the graph:

$$\mathbf{L} = [l_{ij}], \quad l_{ij} = \begin{cases} N_i, & \text{if} \quad i=j\\ -1 & \text{if} \quad j\in\mathscr{N}_i\\ 0 & \text{elsewhere}\end{cases}$$

The Laplacian matrix is symmetric for undirected graphs. Its eigenvalues, $\lambda_1\le\lambda_2\le\cdots\le\lambda_N$, contain very significant information about the topology of the graph $\mathscr{G}$. In particular, we have $\lambda_1 = 0$ with $\mathbf{1}$ as eigenvector. In addition, $\lambda_2 > 0$, for a connected graph, is called *algebraic connectivity* and plays a major role in the speed at which information can diffuse in the network [9].

From the properties of the Laplacian matrix we can deduce that:

1. $\tilde{\mathbf{A}}$ is symmetric;
2. Its eigenvalues are $N_{max}-\lambda_i$, where λ_i are the eigenvalues of the Laplacian matrix. In particular N_{max} is a simple eigenvalue.
3. The eigenvectors of the Laplacian matrix are also eigenvectors of $\tilde{\mathbf{A}}$. In particular $\mathbf{1}$ is the eigenvector associated with the eigenvalue N_{max}.

Denoting ε_i the eigenvalues of $\tilde{\mathbf{A}}$, it is straightforward to show that the set of matrices $\mathbf{W}_i = \alpha_i\mathbf{I} + \beta\tilde{\mathbf{A}}$ are jointly diagnalizable and:

$$\mathbf{U}^T\mathbf{W}_i\mathbf{U} = \mathbf{E}_i = diag(\alpha_i+\beta\varepsilon_1,\cdots,\alpha_i+\beta\varepsilon_N)$$

where $\mathbf{U}$ contains the eigenvectors of the Laplacian matrix. Therefore

$$\prod_{i=1}^{D}\mathbf{E}_i = diag(\prod_{i=1}^{D}(\alpha_i+\beta\varepsilon_1),\cdots,\prod_{i=1}^{D}(\alpha_i+\beta\varepsilon_N)) \quad (15)$$

One can note that $\mathbf{U}$ being orthogonal, its first column is equal to $\frac{1}{\sqrt{K}}\mathbf{1}$. We can then write:

$$\frac{1}{K}\mathbf{J}_K = \mathbf{U}diag(1 \quad 0\cdots0)\mathbf{U}^T \quad (16)$$

Since our goal is to solve (14), from (15) and (16) we get:

$$\begin{cases} \prod_{i=1}^{D}(\alpha_i+\beta\varepsilon_1)=1 \\ \prod_{i=1}^{D}(\alpha_i+\beta\varepsilon_k)=0, \quad k=2,\cdots,K \end{cases} \tag{17}$$

A solution to this problem is stated in the following theorem:

Theorem 1 *Given a connected undirected graph associated with the Laplacian matrix $\mathbf{L}$, the set of matrices $\mathbf{W}_k=(\alpha_k+N_{max}\beta)\mathbf{I}-\beta\mathbf{L}, k=1,\cdots,D$, $\beta\neq 0$ allows reaching the average consensus in D steps if:*

1. *$D+1$ is the number of distinct eigenvalues of the Laplacian matrix;*
2. *the parameters β and α_i are respectively given by $\beta=\left(\prod_{i=2}^{D+1}\lambda_i\right)^{-1/D}$ and $\alpha_k=\beta(\lambda_{k+1}-N_{max})$, $k=1,\cdots,D$, with λ_i, $i=2,\cdots,D+1$ the nonzero distinct eigenvalues of $\mathbf{L}$.*

Proof: *The system of equations (17) to be solved can be rewritten as follows:*

$$\prod_{k=1}^{D}(\alpha_k+N_{max}\beta) \quad = \quad 1 \tag{18}$$

$$\prod_{k=1}^{D}(\alpha_k+N_{max}\beta-\beta\lambda_i) \quad = \quad 0, \quad i=2,\cdots,N \tag{19}$$

One can note that the second equation above can be redundant for non simple eigenvalues λ_i. Let $D+1$ be the number of distinct eigenvalues. Therefore, we get D distinct equations $\prod_{k=1}^{D}(\alpha_k+N_{max}\beta-\beta\lambda_i)=0$, $i=2,\cdots,D+1$. A solution parameterized by β is given by: $\alpha_k=\beta(\lambda_{k+1}-N_{max})$, $k=1,\cdots,D$. Replacing these expressions in (18), we get: $\beta^D\prod_{k=1}^{D}\lambda_{k+1}=1$. Hence the solutions given above.■

3.4 D-ALS algorithm using finite time average consensus

In summary, the distributed decoding process is as follows:

1. $\forall k$ initialize $\mathbf{H}_{k.}(0)$ and $\mathbf{C}_k(0)=\hat{\mathbf{C}}$ with random values and compute $w_{kk}^j=\alpha_j+(N_{max}-N_k)\beta$, $j=1,\cdots,D$, using Theorem 1.
2. For $i=1,2,\cdots,$
   (a) At each node compute $\mathbf{\Gamma}_k(0)=\mathbf{Q}_k^H(i-1)\mathbf{Y}_{k..}$ and $\mathbf{\Theta}_k(0)=\mathbf{Q}_k^H(i-1)\mathbf{Q}_k(i-1)$, where $\mathbf{Q}_k(i-1)=\mathbf{C}_k(i-1)diag(\mathbf{H}_{k.}(i-1))$.
   (b) Finite time average consensus: for $j=1,\cdots,D$,

$$\mathbf{\Gamma}_k(j) \quad = \quad w_{kk}^j\mathbf{\Gamma}_k(j-1)+\beta\sum_{q\in\mathcal{N}_k}\mathbf{\Gamma}_q(j-1)$$

$$\mathbf{\Theta}_k(j) \quad = \quad w_{kk}^j\mathbf{\Theta}_k(j-1)+\beta\sum_{q\in\mathcal{N}_k}\mathbf{\Theta}_q(j-1)$$

   (c) Estimation of the encoded information matrix: $\mathbf{B}_k^T(i)=\mathbf{\Theta}_k^{-1}(D)\mathbf{\Gamma}_k(D)$
   (d) At each node compute $\mathbf{\Psi}_k(0)=\mathbf{P}_k^H(i)\mathbf{Y}_{k..}^T$ and $\mathbf{\Phi}_k(0)=\mathbf{P}_k^H(i)\mathbf{P}_k(i)$, with $\mathbf{P}_k(i)=\mathbf{B}_k(i)diag(\mathbf{H}_{k.}(i-1))$.
   (e) Finite time average consensus: for $j=1,\cdots,D$:

$$\mathbf{\Psi}_k(j) \quad = \quad w_{kk}^j\mathbf{\Psi}_k(j-1)+\beta\sum_{q\in\mathcal{N}_k}\mathbf{\Psi}_q(j-1)$$

$$\mathbf{\Phi}_k(j) \quad = \quad w_{kk}^j\mathbf{\Phi}_k(j-1)+\beta\sum_{q\in\mathcal{N}_k}\mathbf{\Phi}_q(j-1)$$

   (f) Estimation of the code matrix: $\mathbf{C}_k^T(i)=\mathbf{\Phi}_k^{-1}(D)\mathbf{\Psi}_k(D)$.
   (g) Estimation of local channel parameters as $\mathbf{H}_{k.}^T(i)=(\mathbf{C}_k(i)\odot\mathbf{B}_k(i))^\dagger vec(\mathbf{Y}_{k..}^T)$.
   (h) After convergence $\hat{\mathbf{B}}=\mathbf{B}_k(i), \forall k$, then the destination node decodes the information matrix: $\hat{\mathbf{S}}=(\mathbf{A}^H\mathbf{A})^{-1}\mathbf{A}^H\hat{\mathbf{B}}^T$.

4. SIMULATION RESULTS

In this section, we present some results corresponding a transmission/reception scenario where the source node collaborates with three other nodes, while the destination node collaborates with seven nodes. The virtual multi-antenna system has four transmitting antennas (nodes) and eight receiving ones. Thus, the parameters of the overall tensor model were $K=8$, $L=4$, $T=250$, and $N=4$. The considered graph topology at the receiver front-end was a circle. The informative symbols were randomly generated from a BPSK alphabet. The columns of the matrix $\mathbf{C}$ were randomly generated.

We compared the D-ALS algorithm with finite time consensus, named DALS/FTC, with the version derived in [8] and also with the centralized scheme where the destination node receive the tensor slices of the nodes in the network before computing the CP decomposition. For D-ALS, the weight were computed using the uniform scheme [17]. According to Theorem 1, the weights of the finite-time average consensus were computed using the 4 distinct nonzero eigenvalues of the Laplacian matrix $\{0.5858; 2; 3.4142; 4\}$. Therefore finite-time average consensus is achieved in 4 steps.

The results below are averaged values over 100 independent Monte-Carlo runs. At each Monte Carlo run, the columns of $\mathbf{C}$ (amplification factors) were randomly drawn from a uniform distribution $\mathscr{U}[1;5]$.

The performance is evaluated according to the normalized mean square error (NMSE) at the destination node, i.e.:

$$\text{NMSE}=\frac{\left\|\mathbf{Y}_{K..}-\hat{\mathbf{C}}_K diag(\hat{\mathbf{H}}_{K.})\hat{\mathbf{B}}_K^T\right\|_F^2}{\|\mathbf{Y}_{K..}\|_F^2}$$

and according to the bit error rate (BER).

As expected, the results obtained with the centralized scheme are strictly equal to those obtained with the DALS/FTC algorithm. Therefore we just give the comparison results between D-ALS and DALS/FTC. Figures 1 and 2 depict the NMSE in the noiseless case and for an additive white Gaussian noise (SNR=36 dB). Recall that we consider the communication inside a cluster of transmitting or receiving nodes are perfect. The additive noise only concerns inter-cluster communications. We can note that for both noiseless and noisy cases, for the same number of consensus iterations, the DALS/FTC gives better results than D-ALS. This is an expected result since D-ALS makes use of approximate values compared to exact values for DALS/FTC. The performance of D-ALS are improved when increasing the number of consensus iterations. Figure 3 depicts the decoding performance in terms of BER. We also note the same behavior as above.

5. CONCLUSION

In this paper, we have extended the Khatri-Rao Space time coding method proposed in [7] to cooperative networks. For cooperating nodes having a single antenna, these nodes constitute a virtual antenna array at both transmitting and receiving front-end. At each node, the received data can be viewed as slices of a third-order tensor. Therefore, retrieving the informative data is achieved by means of a CP tensor decomposition using an Alternating Least Squares (ALS) algorithm for example. When all the slices cannot be gathered at the same node, for storage resources limitations for example, a distributed ALS method can be used as in [8], which is an average consensus based method. Instead of using a standard consensus method where convergence is achieved asymptotically, we have

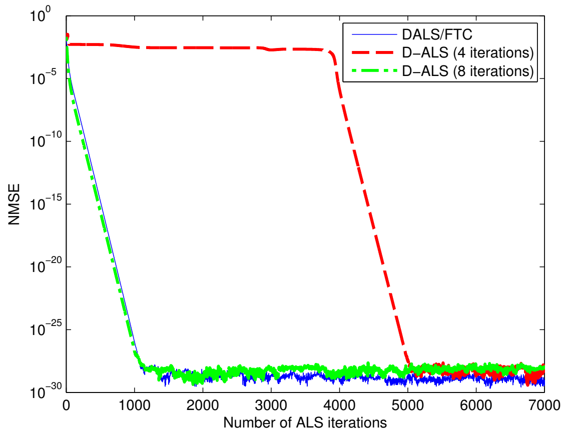

Figure 2: Comparison of D-ALS based on asymptotic and finite-time average consensus in terms of NMSE: Noiseless case.

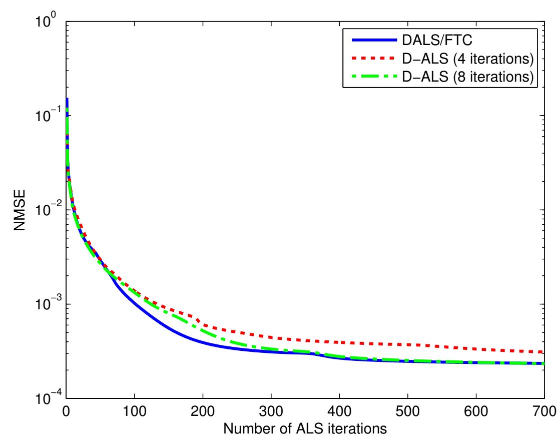

Figure 3: Comparison of D-ALS based on asymptotic and finite-time average consensus in terms of NMSE: Noisy case (SNR=36 dB)

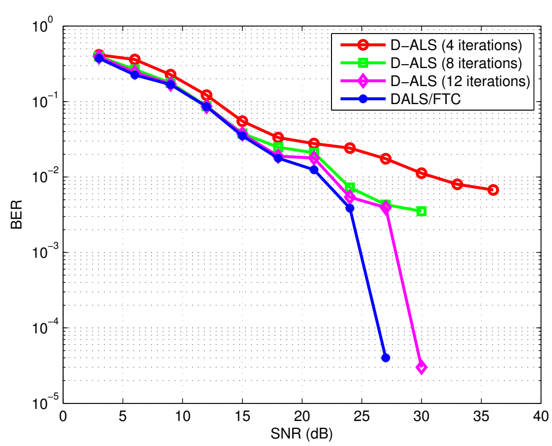

Figure 4: Comparison of D-ALS based on asymptotic and finite-time average consensus in terms of BER.

proposed a finite time average consensus approach that relies on the knowledge of the graph topology. Future works include the evaluation of the impact of imperfect data exchange or asynchronism. A deeper convergence analysis should also be addressed.

REFERENCES

[1] V. Tarokh, N. Seshadri, and A.R. Calderbank. Space-time codes for high data rate wireless communication: Performance criterion and code construction. *IEEE Trans. on Inform. Theory*, 44:744–765, March 1998.

[2] A. Sendonaris, E. Erkip, and B. Aazhang. User cooperation diversity. part i. system description. *IEEE Trans. on Communications*, 51(11):1927–1938, 2003.

[3] 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.

[4] 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 Processing*, 87(2):337–351, February 2007.

[5] 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.

[6] A.Y. Kibangou and G. Favier. Blind equalization of nonlinear channels using tensor decompositions with code/space/time diversities. *Signal Processing*, 89(2):133–143, February 2009.

[7] N. Sidiropoulos and R. Budampati. Khatri-Rao space-time codes. *IEEE Trans. on Signal Processing*, 50(10):2396–2407, Oct. 2002.

[8] A.Y. Kibangou and A.L.F. de Almeida. Distributed PARAFAC based DS-CDMA blind receiver for wireless sensor networks. In *Proc. of the IEEE Workshop SPAWC*, Marrakech, Morocco, June 20-23 2010.

[9] R. Olfati-Saber, A. Fax, and R. Murray. Consensus and cooperation in networked multi-agent systems. *Proc. of the IEEE*, 95(1):215–233, January 2007.

[10] J.N. Laneman and G.W. Wornell. Distributed space-time coded protocols for exploiting cooperative diversity in wireless networks. *IEEE Trans. on Information Theory*, 49(10):2415–2425, October 2003.

[11] 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.

[12] 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.

[13] L. Xiao and S. Boyd. Fast linear iterations for distributed averaging. *Systems Control Lett.*, 53:65–78, 2004.

[14] E. Kokiopoulou and P. Frossard. Accelerating distributed consensus using extrapolation. *IEEE Signal Proc. Letters*, 14(10):665–668, October 2007.

[15] S. Sundaram and C. Hadjicostis. Finite-time distributed consensus in graphs with time-invariant topologies. In *Proc. of American Control Conference (ACC)*, New York City, USA, July 11-13 2007.

[16] J.-C. Delvenne, R. Carli, and S. Zampieri. Optimal strategies in the average consensus problem. *Systems and Control Letters*, 58:759–765, 2009.

[17] V.D. Blondel, J.M. Hendrickx, A. Olshevsky, and J.N. Tsitsiklis. Convergence in multiagent coordination, consensus, and flocking. In *Proc. of the joint 44th IEEE Conf. on Decision and Control (CDC) and European Control Conf (ECC)*, pages 2996–3000, Seville, Spain, December 2005.