Distributed Adaptive Consensus Based Blind Channel Estimation in Ad Hoc Wireless Sensor Networks with Two Signal Sources and Virtual Reference Sensor

Zohre Foroushi†, Vahid TabatabaVakili ††

Department of Electrical Engineering Iran University of Science and Technology Tehran, Iran

Summary
Blind channel estimation has been noticed in many applications. The important challenge in this field is convolutive mixtures in a reverberant environment. On the other hand, in WSN, distributed techniques are preferred because centralized techniques require a large communication bandwidth and power and it is not scalable. In this paper the goal is to obtain the same estimation performance in a centralized algorithm. For this purpose, a MIMO system with 2 input signals will be converted into 2 SIMO systems and their spatial interference will be separated. For incoherence between the obtained equations, a tree structure is chosen and one virtual node is added to this network. The equations are merged and a linear equation is obtained from them. Then, the local channel estimations are coupled with those of their neighbors by means of consensus-constraints. The topology of network is unknown and nodes only have access to data of their direct neighbors. Simulation results show that the proposed distributed algorithm is flexible and robust to sensor failures and achieves the same performance as a centralized algorithm.

Key words:
Consensus based, blind channel estimation, wireless sensor network, distributed.

1. Introduction

Blind channel estimation techniques are used in signal processing and communication for example dereverberation, blind source separation, speech enhancement, wireless communication and so on. In almost all of these applications, a prior knowledge of the source signal is inaccessible or very expensive, so blind methods are necessary.

Distributed solutions in the adaptive wireless networks [1, 2, 3, 4], which consist of a collection of sensors with adaptation and learning abilities are studied in these papers. Simple computing and adaptive implementation are useful for blind channel estimation in real time applications. In [5] a channel identification algorithm is introduced which is based on the least square smoothing (LSS) algorithm. Least mean square (LMS) algorithm is presented in [6], which is based on the adaptive eigenvalue decomposition and multi-channel Newton (MCN) algorithm. However, when blind channel estimation algorithms are used in WSN, the properties of these networks must be considered.

In an ad hoc wireless sensor network [7], usually all observation are gathered in a fusion center for estimation of common parameter or signal. However, this requires a large communication bandwidth and power and it is not scalable. In order to solve this problems, distributed algorithms [8,9,10,11,12] are used, where each node perform local processing and estimate the common object by using data received from neighbors. The distributed algorithms provide robust and scalable solutions for large networks. An important aspect in WSN is the efficient usage of the available bandwidth in the wireless links between nodes. Furthermore, since the nodes of the WSN are generally battery powered, it is important to use a scalable distributed algorithm, where each node contributes to the processing, rather than a centralized algorithm gathering all signals in one central place.

In ad hoc wireless sensor networks, for blind dereverberation, each node accesses to a part of the whole signal and shares data with other nodes through wireless links for channel estimation. The goal is to implement a distributed estimator that is close to an optimal one which has access to all observations of all nodes in the network.

Distributed estimation in a linear least square (LLS) has been studied in [13-19]. This algorithm is applied for solving Uw=d equation, which is a linear regression problem, where d is an M-dimensional data vector and U is an $M \times P$ data matrix with $M \geq P$. The purpose of this algorithm is minimizing the squared error between left-hand and right-hand of the above equation by finding proper w vector. However, in many practical problems, matrix U is also noisy. In these situations, in [20], it is shown that the LLS algorithm is biased. In [21] total least square (TLS) estimation is generalized from LLS estimation, where both U and d are assumed to be noisy.

In this paper, we approach the problem by formulating a new function similar to Uw=d for the blind channel estimation. This equation could be solved by TLS algorithm in the central node using calculate eigenvector corresponding to the smallest eigenvalue of the whole U matrix. In this paper, we proposed consensus based distributed that each node instead of sending received data to the central node, transmit eigenvector corresponding to

Manuscript received May 5, 2016
Manuscript revised May 20, 2016

the smallest eigenvalue of local U matrix to the neighbor nodes.

The important challenge in this application is convolutive mixtures in a reverberant acoustic environment. In this paper, we convert a 2×N MIMO system to 2 SIMO systems. Then by converting the problem to a linear equation and TLS estimation, the problem can be solved iteratively.

The rest of this paper is organized as follows, in sections 2, we first introduce channel model and conversion of 2×N MIMO system to two SIMO systems. The proposed distributed consensus based method corresponding to channel model and convergence of this algorithm are introduced in section 3 and 4. Section 5 summarizes the experimental results. Finally, section 6 presents the concluding.

2. Channel Model

2.1 MIMO System and Conversion of 2×N MIMO System to two SIMO Systems

For a 2×N FIR system as presented in Fig. 1, we have 2 independent speech sources and N sensors. The ith observation $\mathbf{x}_{\mathrm{i}}(\mathrm{k})$ is the summation of linear convolution between the source signal $\mathbf{s}_{\mathrm{m}}(\mathrm{k})$ and the corresponding channel response $\mathbf{h}_{\mathrm{nm}}$, exposed to additive $\mathbf{b}_{\mathrm{n}}(\mathrm{k})$;

$$\boldsymbol{x}_n(k) = \sum_{m=1}^{2} \boldsymbol{h}_{nm}^T\, \boldsymbol{s}_m(k, L) + \boldsymbol{b}_n(k),\;\; k = 1,2,\dots,K, n = 1,2,\dots,N\;, m = 1,2 \qquad (1)$$

$(.)^T$ denotes the transpose of a matrix.

Where,

$$\boldsymbol{h}_{nm} = [h_{nm,0}, h_{nm,1}, \dots, h_{nm,L-1}]^T$$

$$\boldsymbol{s}_m(k, L) = [s_m(k)\; s_m(k-1)\; \dots .\, s_m(\mathrm{k}-\mathrm{L}+1)]^T$$

In the above equations, L is the length of the longest channel impulse response and $\mathbf{b}_{\mathrm{n}}(\mathrm{k})$ is a zero mean additive white Gaussian noise.

By applying z transform to (1), the model signal is expressed as:

$$\boldsymbol{X}_{\mathrm{n}}(\mathrm{z}) = \sum_{m=1}^{2} \mathbf{H}_{nm}(\mathrm{z})\, \mathbf{s}_m(\mathrm{z}) + \boldsymbol{B}_{\mathrm{n}}(\mathrm{z}), \mathrm{n} = 1, \dots, \mathrm{N} \quad (2)$$

First, Where in the above equation, $\mathbf{H}_{\mathrm{nm}}(\mathrm{z})$ is z transform of $\mathbf{h}_{\mathrm{nm}}$.

In blind channel estimation, priori knowledge about source signals or channel coefficients is not assumed. Here, we are looking for converting the MIMO system to several SIMO systems. In the 2×N MIMO system that is shown in Fig. 1, we have

$$\boldsymbol{X}_{\mathrm{i}}(\mathrm{z}) = \mathbf{H}_{i1}(\mathrm{z})\mathbf{s}_1(\mathrm{z}) + \mathbf{H}_{i2}(\mathrm{z})\mathbf{s}_2(\mathrm{z}) + \boldsymbol{B}_{\mathrm{i}}(\mathrm{z})$$
$$\boldsymbol{X}_{\mathrm{j}}(\mathrm{z}) = \mathbf{H}_{\mathrm{j}1}(\mathrm{z})\mathbf{s}_1(\mathrm{z}) + \mathbf{H}_{\mathrm{j}2}(\mathrm{z})\mathbf{s}_2(\mathrm{z}) + \boldsymbol{B}_{\mathrm{j}}(\mathrm{z}) \qquad (3)$$

It is observed that we can remove the interference caused by $\mathbf{s}_1(\mathrm{z})$ or $\mathbf{s}_2(\mathrm{z})$ in node 1 and 2 as follows:

$$\boldsymbol{X}_1(\mathrm{z})\mathbf{H}_{22}(\mathrm{z}) - \boldsymbol{X}_2(\mathrm{z})\mathbf{H}_{12}(\mathrm{z}) = [\mathbf{H}_{11}(\mathrm{z})\mathbf{H}_{22}(z) - \mathbf{H}_{21}(\mathrm{z})\; \mathbf{H}_{12}(\mathrm{z})]\mathbf{s}_1(\mathrm{z}) + [\mathbf{H}_{22}(\mathrm{z})B_1(\mathrm{z}) - \mathbf{H}_{12}(\mathrm{z})B_2(\mathrm{z})] \quad (4)$$

$$\boldsymbol{X}_1(\mathrm{z})\mathbf{H}_{21}(\mathrm{z}) - \boldsymbol{X}_2(\mathrm{z})\mathbf{H}_{11}(\mathrm{z}) = [\mathbf{H}_{12}(\mathrm{z})\mathbf{H}_{21}(\mathrm{z}) - \mathbf{H}_{22}(\mathrm{z})\; \mathbf{H}_{11}(\mathrm{z})]\mathbf{s}_2(\mathrm{z}) + [\mathbf{H}_{21}(\mathrm{z})B_1(\mathrm{z}) - \mathbf{H}_{11}(\mathrm{z})B_2(\mathrm{z})] \quad (5)$$

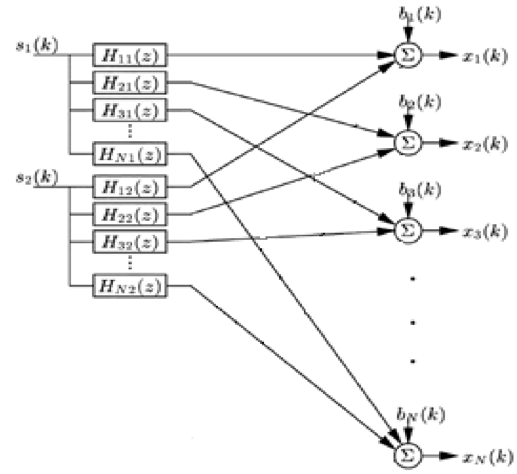

Fig. 1. Illustration of a MIMO system with M signal sources and N sensor nodes

Therefore, by selecting two sensors, we can obtain 2 SIMO systems.

In time domain we have:

$$\mathbf{x}_{\mathrm{i}}(\mathrm{n}) = \mathbf{H}_{\mathrm{i}1}.\,\mathbf{s}_1(\mathrm{n}) + \mathbf{H}_{\mathrm{i}2}.\,\mathbf{s}_2(\mathrm{n}) + \mathbf{b}_{\mathrm{i}}(\mathrm{n}) \quad (6)$$

$$\mathbf{x}_{\mathrm{i}}(\mathrm{n}) = [x_i(n)\;\; x_i(n-1) \dots\;\; x_i(n-L+1)]^T \qquad (7)$$

$$\mathbf{H}_{\mathrm{ij}} = \begin{bmatrix} \mathrm{h}_{\mathrm{ij},0} & \mathrm{h}_{\mathrm{ij},1} & \dots & \mathrm{h}_{\mathrm{ij},\mathrm{L}-1} & 0 & \dots & 0 \\ 0 & \mathrm{h}_{\mathrm{ij},0} & \dots & \mathrm{h}_{\mathrm{ij},\mathrm{L}-2} & \mathrm{h}_{\mathrm{ij},\mathrm{L}-1} & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & \mathrm{h}_{\mathrm{ij},0} & \mathrm{h}_{\mathrm{ij},1} & \dots & \mathrm{h}_{\mathrm{ij},\mathrm{L}-1} \end{bmatrix} \quad (8)$$

$$\mathbf{b}_{\mathrm{i}}(\mathrm{n}) = [\mathrm{b}_{\mathrm{i}}(\mathrm{n})\;\; \mathrm{b}_{\mathrm{i}}(\mathrm{n}-1) \dots\;\; \mathrm{b}_{\mathrm{i}}(\mathrm{n}-\mathrm{L}+1)]^{\mathrm{T}} \qquad (9)$$

$$\mathbf{s}_{\mathrm{j}}(\mathrm{n}) = [\mathrm{s}_{\mathrm{j}}(\mathrm{n})\;\; \mathrm{s}_{\mathrm{j}}(\mathrm{n}-1). \dots \mathrm{s}_{\mathrm{j}}(\mathrm{n}-2\mathrm{L}+2)\;]^{\mathrm{T}} \qquad (10)$$

In the above equations, the input signals $\mathbf{s}_1(\mathrm{n})$ and $\mathbf{s}_2(\mathrm{n})$ are 2L dimensional vectors and $\mathbf{H}_{\mathrm{i}1}$ and $\mathbf{H}_{\mathrm{i}2}$ are $\mathrm{L}\times 2\mathrm{L}$ dimensional matrixes. Now, assume that we trace to remove the interference caused by $\mathbf{s}_1(\mathrm{n})$ and $\mathbf{s}_2(\mathrm{n})$, respectively.

$$\mathbf{x}_{\mathrm{i}}(\mathrm{n}) = \mathbf{H}_{\mathrm{i}1}.\,\mathbf{s}_1(\mathrm{n}) + \mathbf{H}_{\mathrm{i}2}.\,\mathbf{s}_2(\mathrm{n}) + \mathbf{b}_{\mathrm{i}}(\mathrm{n}) \qquad (11)$$

$$\mathbf{x}_{\mathrm{j}}(\mathrm{n}) = \mathbf{H}_{\mathrm{j}1}.\,\mathbf{s}_1(\mathrm{n}) + \mathbf{H}_{j2}.\,\mathbf{s}_2(\mathrm{n}) + \mathbf{b}_{\mathrm{j}}(\mathrm{n}) \qquad (12)$$

By multiplying (11) and (12) in $\mathbf{h}_{\mathrm{j}2}$ and $\mathbf{h}_{\mathrm{i}2}$, respectively and subtract them from each other,

We have,

$$\mathbf{y}_{\mathrm{i},\mathbf{s}_1}(\mathrm{n}) = \mathbf{x}_{\mathrm{i}}(\mathrm{n})\mathbf{h}_{j2}(n) - \mathbf{x}_{\mathrm{j}}(\mathrm{n})\mathbf{h}_{i2}(n)$$
$$= [\mathbf{h}_{\mathrm{j}2}(\mathrm{n})\mathbf{H}_{\mathrm{i}1} - \mathbf{h}_{\mathrm{i}2}(\mathrm{n})\mathbf{H}_{\mathrm{j}1}]\mathbf{s}_1(\mathrm{n}) + [\mathbf{h}_{j2}\mathbf{b}_{\mathrm{i}}(\mathrm{n}) - \mathbf{h}_{i2}(n)\mathbf{b}_{\mathrm{j}}(\mathrm{n})] \qquad (13)$$

And

$$\mathbf{y}_{\mathrm{i},\mathbf{s}_2}(\mathrm{n}) = \mathbf{x}_{\mathrm{i}}(\mathrm{n})\mathbf{h}_{j1}(n) - \mathbf{x}_{\mathrm{j}}(\mathrm{n})\mathbf{h}_{i1}(n)$$

$= [\mathbf{h}_{j1}(n)\mathbf{H}_{i2} - \mathbf{h}_{i1}(n)\mathbf{H}_{j2}]\mathbf{s}_2(n) + [\mathbf{h}_{j1}\mathbf{b}_i(n) - \mathbf{h}_{i1}(n)\mathbf{b}_j(n)]$
(14)
Where

$$\mathbf{h}_{i2} = [h_{i2,0} \;\; h_{i2,1} \dots h_{i2,L-1}]$$
$$\mathbf{h}_{j2} = [h_{j2,0} \;\; h_{j2,1} \dots h_{j2,L-1}]$$

In (13) and (14), $\mathbf{y}_{i,s_1}(n)$ and $\mathbf{y}_{i,s_2}(n)$ are one dimensional vectors, while the $\mathbf{s}_1(n)$ and $\mathbf{s}_2(n)$ are 2L dimensional vectors, respectively. In this manner for each node we have 2 SIMO systems and 2 vector equations.

2.2 Channel model in SIMO system

In a SIMO system, According to [22] as presented in Fig. 2 by ignoring noise, we have:

$$x_i * h_j = s * h_i * h_j = x_j * h_i \;, \; i,j=1,2,\dots,N, \; i \neq j \qquad (15)$$
$$\mathbf{x}_i^T(n)\mathbf{h}_j = \mathbf{x}_j^T(n)\mathbf{h}_i \quad i,j = 1,2,\dots,N, i \neq j \qquad (16)$$

Multiplying (16) by $x_i(n)$ and taking expectation yields, we have

$$\mathbf{R}_{x_i x_i}\mathbf{h}_j = \mathbf{R}_{x_i x_j}\,\mathbf{h}_i \;, \quad i,j = 1,2,\dots,N, i \neq j \qquad (17)$$

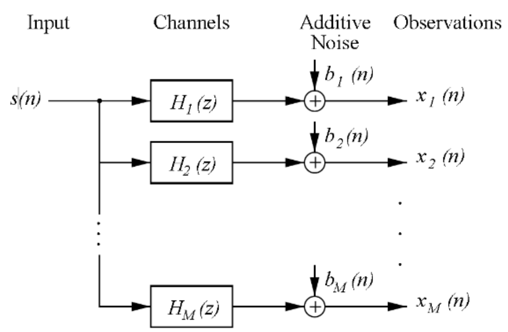

Fig. 2. Illustration of the relationship between the input s(n) and the observation $x_i(n)$ in a single-input multi channel FIR system

(17) specifies N(N-1) equations. By summation of N-1 cross relations associated with each particular channel, we have

$$\sum_{i=1,i\neq j}^{N} \mathbf{R}_{x_i x_i}\,\mathbf{h}_j = \sum_{i=1,i\neq j}^{N} \mathbf{R}_{x_i x_j}\,\mathbf{h}_i \quad j = 1,2,\dots,N \qquad (18)$$

Therefore, we have

$$\sum_{i=1,i\neq j}^{N} \mathbf{R}_{x_i x_i}\,\mathbf{h}_j = \mathbf{R}_{x_1 x_j}\,\mathbf{h}_1 + \dots + \mathbf{R}_{x_N x_j}\,\mathbf{h}_M \quad j = \dots, N, i \neq j \qquad (19)$$

In a matrix form, above equation is written as:

$$\mathbf{U}\mathbf{h} = 0 \qquad (20)$$

Where

$$\mathbf{U} = \begin{bmatrix} \sum_{i\neq 1}\mathbf{R}_{x_i x_i} & -\mathbf{R}_{x_2 x_1} & \dots & -\mathbf{R}_{x_M x_1} \\ -\mathbf{R}_{x_1 x_2} & \sum_{i\neq 2}\mathbf{R}_{x_i x_i} & \dots & -\mathbf{R}_{x_M x_2} \\ \vdots & \vdots & \ddots & \vdots \\ -\mathbf{R}_{x_1 x_M} & -\mathbf{R}_{x_2 x_M} & \dots & \sum_{i\neq M}\mathbf{R}_{x_i x_i} \end{bmatrix} \qquad (21)$$

And

$$\mathbf{h} = [\mathbf{h}_1^T \;\; \mathbf{h}_2^T \;\; \dots \;\; \mathbf{h}_M^T]^T \qquad (22)$$

In actual conditions, when observation noise is present, the right hand side of (20) is not zero and an error vector is produced.

$$\mathbf{E} = \mathbf{U}\mathbf{h} \qquad (23)$$

$\mathbf{E}$ and $\mathbf{U}$ matrices are exposed to white noise. A popular solution for this equation is total least square algorithm. In this algorithm, to solve (23) the new matrixes have been defined as follows:

$$\mathbf{U}_+ = [\mathbf{U}\;\mathbf{E}] \qquad (24)$$
$$\mathbf{R} = \mathbf{U}_+^T\mathbf{U}_+ \qquad (25)$$

Then, the eigenvector of R corresponding to the smallest eigenvalue is calculated. The relationship between the h vector, answer of (23), and the eigenvector of R corresponding to the smallest eigenvalue, x, is written as:

$$\mathbf{x} = \begin{bmatrix} \mathbf{h}^* \\ -1 \end{bmatrix} \qquad (26)$$

The above algorithm is centralized since in order to calculate the R matrix, each node transmits the received signal to the central node and calculation of eigenvector of R has been done in this node.

In this paper, we have introduced the consensus based distributed algorithm for solving (23). In the proposed algorithm, each node calculates eigenvector corresponding to the smallest eigenvalue, based on its received signal and this eigenvector is broadcasted to the neighbor nodes. Then, the local channel estimations are coupled with those of their neighbors by means of consensus-constraints.

3. Proposed Blind Channel Estimation Algorithm

After The Consider an ad-hoc WSN with a random and connected topology where set of nodes are denoted as $K = \{1, 2, \dots, N\}$. The set of neighbor nodes of node k, i.e. N_k are the nodes which can share data with node k. In this scenario, the received signal by node i is shown below.

$$x_i(n) = [x_i(n) \;\; x_i(n-1) \dots \;\; x_i(n-L+1)]^T \qquad (27)$$

In this section, by using the equations in previous parts which converts the MIMO system to two SIMO systems for each of the nodes as shown in Fig. 3, we have:

$\mathbf{y}_{i,s_1}(n) = \mathbf{x}_i(n)\mathbf{h}_{j2}(n) - \mathbf{x}_j(n)\mathbf{h}_{i2}(n) = [\mathbf{h}_{j2}(n)\mathbf{H}_{i1} - \mathbf{h}_{i2}(n)\mathbf{H}_{j1}]\mathbf{s}_1(n) + [\mathbf{h}_{j2}\mathbf{b}_i(n) - \mathbf{h}_{i2}(n)\mathbf{b}_j(n)]$ (28) And

$\mathbf{y}_{i,s_2}(n) = \mathbf{x}_i(n)\mathbf{h}_{j1}(n) - \mathbf{x}_j(n)\mathbf{h}_{i1}(n) = [\mathbf{h}_{j1}(n)\mathbf{H}_{i2} - \mathbf{h}_{i1}(n)\mathbf{H}_{j2}]\mathbf{s}_2(n) + [\mathbf{h}_{j1}\mathbf{b}_i(n) - \mathbf{h}_{i1}(n)\mathbf{b}_j(n)]$ (29)

By using (21) for each signal source, we have two equations in the form of $\mathbf{U}_{s_i}\mathbf{h}_{s_i}=0$ where $\mathbf{U}_{s_i}$ is a square matrix and its dimensions are equal to the number of sensors and $\mathbf{h}_{s_i}$ is a matrix of N×2L-1 dimension. Now, by writing first L columns of matrix $\mathbf{h}_{s_i}$ under the previous one, matrix $\mathbf{h}_{s_i}$ is converted to an NL dimensional vector.

We must convert matrix $\mathbf{U}_{\mathbf{s}_\mathbf{i}}$ to the NL dimension square matrix as follows.

$$\mathbf{U}_{\mathrm{new},\boldsymbol{s}_\mathbf{i}} = \begin{bmatrix} \mathbf{U}_{\mathbf{s}_\mathbf{i}} & 0 & \dots & 0 \\ 0 & \mathbf{U}_{\mathbf{s}_\mathbf{i}} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mathbf{U}_{\mathbf{s}_\mathbf{i}} \end{bmatrix}, \quad i = 1,2 \tag{30}$$

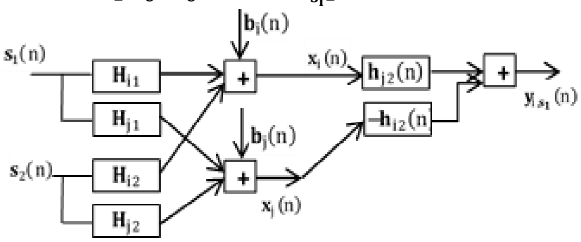

Fig. 3. Converting the MIMO system to two SIMO systems

Now, we have 2 equations similar to (23), that $\mathbf{U}_{\mathrm{new},\boldsymbol{s}_1}$ and $\mathbf{U}_{\mathrm{new},\boldsymbol{s}_2}$ are NL dimensional square matrixes and $\mathbf{h}_{\boldsymbol{new},\boldsymbol{s}_1}$ and $\mathbf{h}_{\boldsymbol{new},\boldsymbol{s}_2}$ are NL dimensional vectors.

Where,

$$\begin{bmatrix} \mathbf{U}_{\boldsymbol{s}_1} & 0 & \dots & 0 \\ 0 & \mathbf{U}_{\boldsymbol{s}_1} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mathbf{U}_{\boldsymbol{s}_1} \end{bmatrix} \mathbf{h}_{\boldsymbol{new},\boldsymbol{s}_1} = \mathbf{0} \tag{31}$$

$$\begin{bmatrix} \mathbf{U}_{\boldsymbol{s}_2} & 0 & \dots & 0 \\ 0 & \mathbf{U}_{\boldsymbol{s}_2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mathbf{U}_{\boldsymbol{s}_2} \end{bmatrix} \mathbf{h}_{\boldsymbol{new},\boldsymbol{s}_2} = \mathbf{0} \tag{32}$$

By combination of two $\mathbf{U}_{\mathrm{new},\boldsymbol{s}_1}$ and $\mathbf{U}_{\mathrm{new},\boldsymbol{s}_2}$ matrixes as follows, we can obtain a square matrix $\mathbf{U}_{\mathrm{new}}$ where its dimensions are 2NL.

$$\begin{bmatrix} \mathbf{U}_{\mathrm{new},\boldsymbol{s}_1} & 0 \\ 0 & \mathbf{U}_{\mathrm{new},\boldsymbol{s}_2} \end{bmatrix} \begin{bmatrix} \mathbf{h}_{\boldsymbol{new},\boldsymbol{s}_1} \\ \mathbf{h}_{\boldsymbol{new},\boldsymbol{s}_2} \end{bmatrix} = \mathbf{0} \tag{33}$$

Where

$$\mathbf{U}_{\mathrm{new}} = \begin{bmatrix} \mathbf{U}_{\mathrm{new},\boldsymbol{s}_1} & 0 \\ 0 & \mathbf{U}_{\mathrm{new},\boldsymbol{s}_2} \end{bmatrix}$$

$$\mathbf{h}_{\boldsymbol{new}} = \begin{bmatrix} \mathbf{h}_{\boldsymbol{new},\boldsymbol{s}_1} \\ \mathbf{h}_{\boldsymbol{new},\boldsymbol{s}_2} \end{bmatrix}$$

However, if there is even one loop in this network, the equations in (33) are self-dependent. To avoid dependencies between them, a tree structure is selected from this network and nodes only transmit data to the neighbor nodes which are associated to this tree structure. By this solution, the number of equations is decreased to (N-1) $\times$ L that is less than the number of unknown parameters. For solving this problem, we used a virtual node which is located in the center of one of the signal source. Other filter coefficients are calculated based on this virtual node. In this virtual node, the channel coefficients between the virtual node and the signal source which is located in the same location is one and other coefficients are zero.

For solving equation Uh=E which U and h are noisy data as was mentioned, we propose the distributed consensus based TLS algorithm, but in the proposed algorithm U is unknown and we must calculate it iteratively. Results are compared to actual values and the coefficients which are obtained by solving (33) with TLS algorithm. In order to use TLS algorithm, all nodes in this network should transmit their data to the central node.

In this scenario, by allocation of primary values to $\mathbf{h}_{\mathrm{i}1}(\mathrm{n})$ and $\mathbf{h}_{\mathrm{i}2}(\mathrm{n})$, we use the random values, $\mathbf{y}_{\mathrm{i},\mathrm{s}_1}(\mathrm{n})$ and $\mathbf{y}_{\mathrm{i},\mathrm{s}_2}(\mathrm{n})$ are calculated with (28) and (29) in each node. In each node and transmitted to the neighbors. The primary entries of $\mathbf{U}_{\mathbf{s}_\mathbf{i}}$ in each node, for example node 1, are as follows:

$$\mathrm{U}_{\mathrm{s}_1} = \begin{bmatrix} \mathrm{R}_{\mathrm{y}_{2,\mathrm{s}_1}\mathrm{y}_{2,\mathrm{s}_1}} + \mathrm{R}_{\mathrm{y}_{\mathrm{N},\mathrm{s}_1}\mathrm{y}_{\mathrm{N},\mathrm{s}_1}} & -\mathrm{R}_{\mathrm{y}_{2,\mathrm{s}_1}\mathrm{y}_{1,\mathrm{s}_1}} & \dots & -\mathrm{R}_{\mathrm{y}_{\mathrm{N},\mathrm{s}_1}\mathrm{y}_{1,\mathrm{s}_1}} \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix} \tag{34}$$

$$\mathrm{U}_{\mathrm{s}_2} = \begin{bmatrix} \mathrm{R}_{\mathrm{y}_{2,\mathrm{s}_2}\mathrm{y}_{2,\mathrm{s}_2}} + \mathrm{R}_{\mathrm{y}_{\mathrm{N},\mathrm{s}_2}\mathrm{y}_{\mathrm{N},\mathrm{s}_2}} & -\mathrm{R}_{\mathrm{y}_{2,\mathrm{s}_2}\mathrm{y}_{1,\mathrm{s}_2}} & \dots & -\mathrm{R}_{\mathrm{y}_{\mathrm{N},\mathrm{s}_2}\mathrm{y}_{1,\mathrm{s}_2}} \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix} \tag{35}$$

In (34) and (35), we assumed that node 1 is neighbored to node 2 and N. However, when the network is very large, the benefits of the proposed algorithm are more significant. In this algorithm, each node broadcasts the eigenvector corresponding to the smallest eigenvalue to the neighbor nodes. The relationship between eigenvalues and eigenvectors of $\mathbf{R}_{\boldsymbol{k}} = \mathbf{U}_{+}^{T}\mathbf{U}_{+}$ is as follows:

$$\mathbf{R}_{\boldsymbol{k}} \times \boldsymbol{V} = \boldsymbol{V} \times \boldsymbol{D} \tag{36}$$

Where the columns of V are the eigenvectors of U and D is a diagonal matrix with U's eigenvalues on the main diagonal. So, we have:

$$\mathbf{R}_{\boldsymbol{k}} = \boldsymbol{V} \times \boldsymbol{D} \times \boldsymbol{V}^{-1} \tag{37}$$

We can observe that $\mathbf{R}_{\boldsymbol{k}} \cong \boldsymbol{V} \times \boldsymbol{V}^{-1}$, but we don't have access to all of eigenvectors. We have received the eigenvectors corresponding to the smallest eigenvalues from the neighbor nodes. Thus, we must update $\mathbf{R}_{\boldsymbol{k}}$ by using that eigenvectors.

$$\overline{\mathbf{R}}_{\mathbf{k}}^{(\mathbf{i})} = \mathbf{R}_{\mathbf{k}} + \mathbf{Q}_{\mathbf{k}}^{(\mathbf{i})} \tag{38}$$

Where,

$$\mathbf{Q}_{\mathrm{k}}^{(\mathrm{i}+1)} = \mathbf{Q}_{\mathrm{k}}^{(\mathrm{i})} + \mu_{\mathrm{i}} \left(|\mathrm{N}_{\mathrm{k}}| \mathbf{x}_{k}^{(i)} \mathbf{x}_{k}^{(i)^{T}} \sum_{\mathrm{m}\in\mathrm{N}_{\mathrm{k}}} \mathbf{x}_{m}^{(i)} \mathbf{x}_{m}^{(i)^{T}} \right) \tag{39}$$

$\mathbf{X}_{\boldsymbol{k}}$ is the eigenvector corresponding to smallest eigenvalue in node k and μ_{i} is the positive step-size. The main idea in (39) is to optimize and update $\mathbf{Q}_{\boldsymbol{k}}$ until a consensus is reached among all neighbor nodes. If node q is neighbor of

node k, the constraint that $\mathbf{X}_k = \mathbf{X}_m$ is denoted as consensus constraint [23].
The aim is to compute $\mathbf{h}$ in a distributed manner.
The solution of this problem according to (26) is:

$$\hat{\mathbf{h}} = -\frac{1}{\mathbf{n}_N^T \mathbf{x}^*}[\mathbf{I}_{N+1} \,|\mathbf{0}_{N+1}]\mathbf{x}^* \quad (40)$$

Where, $\mathbf{n}_N$ is an N-dimensional vector which its last entry is 1 and all others are zero, and it is used to normalize the last entry of X to one. $\mathbf{0}_{N+1}$ is an N+1 dimensional vector with all zero entries and $\mathbf{I}_{N+1}$ denotes the identity matrix. Assume an $N \times N$ matrix $\mathbf{R}_k = \mathbf{U}_{k+}^T \mathbf{U}_{k+}$ with $\mathbf{U}_{k+} = [\mathbf{U}_k|\mathbf{d}_k]$ and $\mathbf{x}^*$ be the eigenvector related to the smallest eigenvalue of $\mathbf{R}$. The consensus based distributed algorithm for dereverberation is described in Table 1.
According to the proposed algorithm, in each iteration by selecting appropriate μ_i, $\mathbf{Q}_k^{(i)}$ is calculated and $\hat{\mathbf{h}}_{new}$ is derived from it.

Table 1: proposed algorithm for dereverberation

proposed algorithm for dereverberation

1) $\forall k \in K$: Initialize $\mathbf{Q}_k^{(0)} = 0_{D\times D}$, with $D=(N+1)L+1$
2) We defined a virtual node which is located in the center of one of the signal source. For example node number one. In this virtual node, the channel coefficients between the virtual node and the source number 1 are one and other coefficients are zero.
3) Allocate primary random value to $\mathbf{h}_{i1}(n)$ and $\mathbf{h}_{i2}(n)$.
4) Each node receives signal only from neighbor nodes and according to the tree structures, $\mathbf{y}_{i,s_1}(n)$ and $\mathbf{y}_{i,s_2}(n)$ are calculated in each node. Received signal in virtual node is equals to $\mathbf{s}_1(n)$ and calculated according to $\mathbf{y}_{i,s_1}(n)$ and the values of $\mathbf{h}_{i1}(n)$ and $\mathbf{h}_{i2}(n)$.
5) Initialize $\mathbf{U}_{s_1}$ and $\mathbf{U}_{s_2}$ according to (34) and (35), where, the entries in k*th* row could be opposite to zero and the other entries are zero.
6) According to $\mathbf{U}_{s_1}$ and $\mathbf{U}_{s_2}$, $\mathbf{U}_{new}$ is calculated in each node.
7) Each node $k \in K$ computes the $\overline{\mathbf{R}}_k^{(0)}$ according to (25)
8) $i \leftarrow 0$
9) Each node $k \in K$ computes the eigenvector $\mathbf{x}_k^{(i)}$ corresponding to the smallest eigenvalue of $\overline{\mathbf{R}}_k^{(i)}$ defined by $\overline{\mathbf{R}}_k^{(i)} = \mathbf{R}_k + \mathbf{Q}_k^{(i)}$
   Where $\mathbf{x}_k^{(i)}$ is scaled such that $\|\mathbf{x}_k^{(i)}\|=1$.
10) Each node $k \in K$ transmit $\mathbf{x}_k^{(i)}$ to the nodes in N_k
11) Each node $k \in K$ updates
   $\mathbf{Q}_k^{(i+1)} = \mathbf{Q}_k^{(i)} + \mu_i \left(|N_k|\mathbf{x}_k^{(i)}\mathbf{x}_k^{(i)^T} - \sum_{q\in N_k} \mathbf{x}_m^{(i)}\mathbf{x}_m^{(i)^T}\right)$
   $\mu_i > 0$.
12) Compute the local TLS solution $\hat{\mathbf{h}} = -\frac{1}{n_N^T x^*}[I_{N+1} \,|0_{N+1}]x^*$
13) $i \leftarrow i+1$
14) Return to step 8 until i=threshold
15) From $\hat{\mathbf{h}}_{new}$ the channel coefficients are computed.
16) Return to step 4 until the difference between new channel coefficients and the old one smaller than desired threshold.

4. Convergence of the Proposed Algorithm

Where Convergence of the network is accessed when $Q_k^{(i+1)} \cong Q_k^{(i)}$. Thus, in this situation we have:

$$|N_k| x_k^{(i)} x_k^{(i)^T} - \sum_{q\epsilon N_k} x_q^{(i)} x_q^{(i)^T} < \varepsilon \quad (41)$$

$$\sum_{k\in J}\sum_{q\epsilon N_k} ||x_k^{(i)} x_k^{(i)^T} - x_q^{(i)} x_q^{(i)^T}||^2 < \varepsilon \quad (42)$$

$$\left\|x_k^{(i)} x_k^{(i)^T} - \mathbf{x}_q^{(i)}\mathbf{x}_q^{(i)^T}\right\|^2 = tr\left(\left(x_k^{(i)} x_k^{(i)^T} - \mathbf{x}_q^{(i)}\mathbf{x}_q^{(i)^T}\right)^2\right)$$
$$= tr\left((x_k^{(i)} x_k^{(i)^T})^2\right) + tr\left((x_k^{(i)} x_k^{(i)^T})^2\right) - 2tr\left(x_k^{(i)^T} x_k^{(i)} \times x_k^{(i)} x_k^{(i)^T}\right) = 2-2(x_k^{(i)^T} x_k^{(i)})^2 \quad (43)$$

From the (23), maximum bound for the stepsize is:

$$0 < \mu < \frac{\lambda_{min}(\sum_{k\epsilon J}\boldsymbol{R}_k) - \sum_{k\epsilon J}\lambda_{min}(\overline{R}_k^{(i)})}{|L| - \sum_{k\epsilon J}\sum_{q\epsilon N_k}(x_k^{(i)^T} x_k^{(i)})^2} \quad (44)$$

In (44), $\lambda_{min}(\mathbf{Q})$ denotes the smallest eigenvalue of Q and $|\mathcal{L}|$ is the number of links. The numerator of (43) is equal to the difference between smallest eigenvalue of current value and its optimal one, which approaches to zero if current value gets closer to the optimal value. The dominator of (43) is the summation of squared consensus error over all links of the network. It should be mentioned that dominator depends on the number of links $|\mathcal{L}|$. i.e., networks that have large number of links require smaller μ. Not only, does not this necessarily result in slower convergence, but also it will be shown that in a strongly connected network, information diffuses much faster over the network. It is observed that when $|J|$ increases, $\lambda_{min}(\sum_{k\epsilon J}\mathbf{R}_k)$ is increased, which means the numerator depends on the number of nodes.

5. Simulation results

In this section, we prove the convergence of consensus based distributed algorithm estimation to the desired channel impulse response of an identifiable multi-channel system by simulation results. To demonstrate the general behavior, we averaged the results over multiple Monte-Carlo (MC) runs. In each MC run, the following process is used to generate the network and the sensor observation:

Construct an $NL \times 2$ -dimensional vector h where entries are drawn from a zero-mean normal distribution with unit variance.
Create a connected network with N nodes
For each node k:
Construct a 2L-dimensional vector input data, s, where entries are calculated from a zero mean normal distribution with unit variance.
Compute $\mathbf{H}_i$.
Compute the received signal in each node from $\mathbf{x}_i(n) = \mathbf{H}_{i1}.\mathbf{s_1}(n) + \mathbf{H}_{i2}.\mathbf{s_2}(n) + \mathbf{b}_i(n)$
Assign initial value to $\mathbf{h}_{j2}(n)$ and $\mathbf{h}_{i2}(n)$, based on the selected tree structure and compute $\mathbf{y}_i(n) = \mathbf{x}_i(n)h_{j2}(n) - \mathbf{x}_j(n)h_{i2}(n)$.
Construct an 2(N+1)L×2(N+1)L data matrix U_k
Compute $E_k=U_k h=0$.
Add zero-mean white Gaussian noise with a standard deviation of 0.5 to the entries in U_k and E_k.

In each experiment, we chose L=2 and N=10, so D=23 and we run this algorithm for 400 iterations.

To evaluate the convergence of the proposed algorithm, the error between the true solution and the local estimation, averaged over M nodes in the network is used.

To stimulate the use of proposed algorithm, results of this technique is compared to exact value one.
Figure 4 shows exact entries of 8-dimensional vector h, for 4 nodes, and the proposed algorithm after 400 iterations, averaged over 100 MC runs, where the step size is fixed to μ=0.1.
This figure illustrates that the proposed distributed algorithm is very close to the exact values.

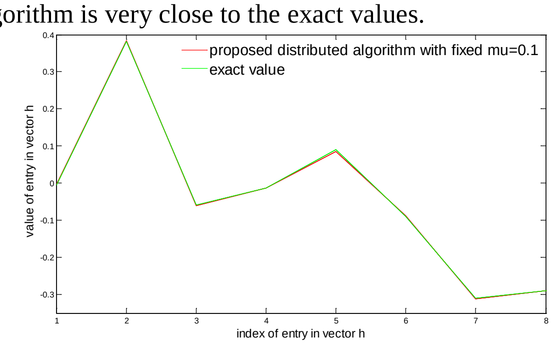

Fig. 4. Comparison between blind channel estimation by proposed distributed algorithm with the exact value averaged over 100 Monte-Carlo runs

5.1 Influence of step size μ

To demonstrate convergence properties of the proposed distributed algorithm, we used different values for fixed step size, and compared the results. Results are shown in Figure 5. This figure shows that $\mu = 0.01$ gives better answer. Practically, the convergence of the proposed distributed algorithm strongly depends on the step size. For the step size larger than 0.5, convergence becomes a vague.

5.2 Effect of connectivity in network

In this section, the influence of the connectivity of the network is shown in the network with N=10 nodes. Results are shown for three networks: a network with a ring topology in which every node has two neighbors, a fully connected network in which every node has (N-1) neighbors and a network in between them in which every node neighbors with N/2 nodes. In this experiment, we used a fixed step size which depends on the number of links, i.e.,$\mu = 10/|\mathcal{L}|$. Results are shown in Figure 6. It is observed that convergence speed increases by addition of the number of links, even though a smaller μ is used. Figure 6 shows that increasing the convergence speed affects less for larger $|\mathcal{L}|$.

5.3 Influence of network size

In this part, network size is increased by addition of nodes. The average links per node is fixed to 3. We simulated 50 MC runs for a network with 4,10 and 15 nodes and with the step size fixed to $\mu = 0.01$. Also, length of the channel impulse response of each node is 3. Results are shown in Figure 7. It is observed that size of the network has no serious effect on the convergence speed.

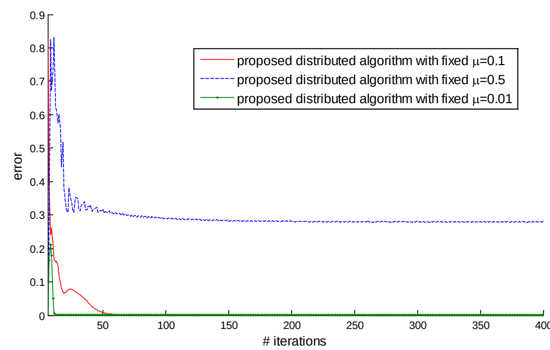

Fig. 5 Convergence properties of the proposed distributed algorithm for different step sizes averaged over 100 Monte-Carlo runs.

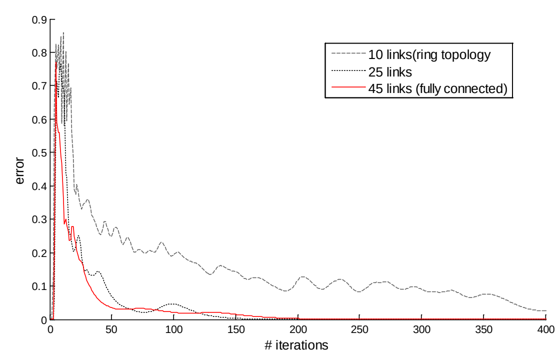

Fig. 6. Convergence properties of the proposed distributed algorithm for different levels of connectivity averaged over 200 Monte-Carlo runs.

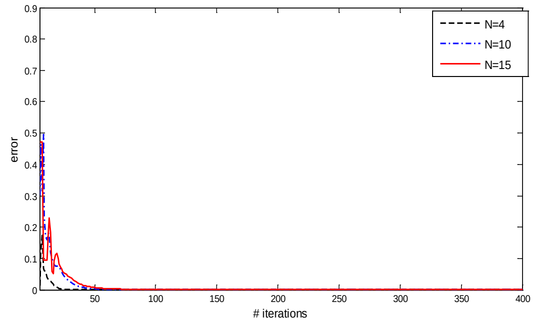

Fig. 7. Convergence properties of the proposed distributed algorithm for different size of the network averaged over 50 Monte-Carlo runs.

5.4 Random graphs

In this section, we show robustness of the proposed distributed algorithm against random link failures. In each iteration, the probability of failure of each link equals to p%, where $p \in \{0, 50, 90\}$. We simulated 100 MC runs for a fixed step size $\mu = 0.01$. From Figure 8, it is observed that the proposed distributed algorithm is robust against random link failure.

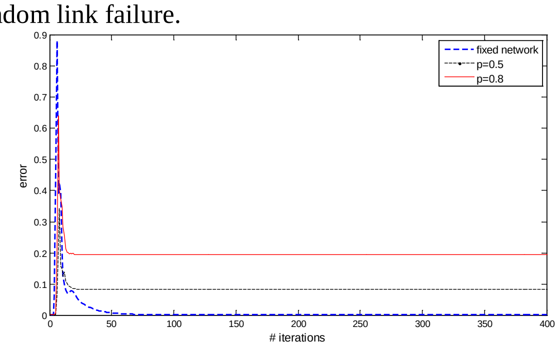

Fig. 8 Convergence properties of the proposed distributed algorithm for random network graph with different link failure probabilities, averaged over 100 Monte-Carlo runs.

6. Conclusions

The distributed blind channel estimation problem in ad-hoc wireless sensor networks is studied in this paper. The important challenge in this application is convolutive mixtures in a reverberant acoustic environment. In this paper, a MIMO system that has 2 input signals is converted into 2 SIMO systems and their spatial interference is separated. A tree structure is chosen from fully connected network and one virtual node is added to this network at signal source 1. The equations of 2 SIMO systems are merged and a linear equation is obtained from them. This equation is solved by consensus based distributed algorithm and channel coefficients are obtained iteratively. The proposed algorithm is totally distributed and flexible, i.e., the network topology is unknown, and nodes only share data with their direct neighbors through local broadcasts. Simulation results show that the proposed distributed algorithm is flexible and robust to sensor failures. Because of these features there is no single point of failure. Provided MC simulation results illustrate the effectiveness of the algorithm.

References

[1] A. H. Sayed. "Adaptive networks". Proceedings of the IEEE, 102(4):460–497, April 2014.

[2] A. H. Sayed. " Diffusion adaptation over networks". In R. Chellapa and S. Theodoridis, editors, E-Reference Signal Processing, vol. 3, pages 323–454. Academic Press, 2014.

[3] A. H. Sayed, S.-Y. Tu, J. Chen, X. Zhao, and Z. Towfic. "Diffusion strategies for adaptation and learning over networks", IEEE Signal Processing Magazine, 30(3):155–171, May 2013.

[4] M.Thejovathi, Dynamic Load Balancing Algorithms for Distributed Networks, IJCSNS International Journal of Computer Science and Network Security, vol. 14 No.2, February 2014

[5] Q. Zhao, L. Tong: "Adaptive blind channel estimation by least squares smoothing," IEEE Trans. Signal Process. 47 (11) (November 1999) 3000–3012.

[6] A. H. Sayed. Adaptive Filters. Wiley, NJ, 2008.

[7] D. Estrin, L. Girod, G. Pottie, and M. Srivastava, "Instrumenting the world with wireless sensor networks," in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP'01), vol. 4, pp. 2033–2036, 2001.

[8] P. Di Lorenzo and A. H. Sayed., "Sparse distributed learning based on diffusion adaptation," IEEE Trans. Signal Process., 61(6):1419–1433,March 2013.

[9] G. Mateos, I. D. Schizas, and G. B. Giannakis, "Distributed recursive least-squares for consensus-based in-network adaptive estimation," IEEE Trans. Signal Process., vol. 57, no. 11, pp. 4583–4588, 2009.

[10] E. Dall'Anese and G. Giannakis, "Distributed cognitive spectrum sensing via group sparse total least-squares," IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), dec. 2011, pp. 341 –344.

[11] A. Bertrand and M. Moonen, "Distributed adaptive node-specific signal estimation in fully connected sensor networks – part I: sequential node updating," IEEE Transactions on Signal Processing, vol. 58, no. 10, pp. 5277–5291, 2010

[12] C. Silva, R. Costa, A. Pires, "A Cluster-based Approach to provide Energy-Efficient in WSN," IJCSNS International Journal of Computer Science and Network Security, VOL.13 No.1, January 2013.

[13] A. Bertrand and M. Moonen, "Distributed adaptive node-specific signal estimation in fully connected sensor networks—Part II: Simultaneous and asynchronous node updating," IEEE Trans. Signal Process., vol. 58, pp. 5292–5306, Oct. 2010.

[14] C. G. Lopes and A. H. Sayed, "Incremental adaptive strategies over distributed networks," IEEE Trans. Signal Process., vol. 55, pp. 4064–4077, Aug. 2007.

[15] C. G. Lopes and A. H. Sayed, “Diffusion least-mean squares over adaptive networks: Formulation and performance analysis,” IEEE Trans. Signal Process., vol. 56, pp. 3122–3136, July 2008.
[16] S. Kar and J. M. F. Moura, “Convergence rate analysis of distributed gossip (linear parameter) estimation: Fundamental limits and tradeoffs,” IEEE Journal on Selected Topics in Signal Processing, 5(4):674–690, Aug. 2011.
[17] G. Mateos, I. Schizas, and G. Giannakis, “Closed-form MSE performance of the distributed LMS algorithm,” 5th IEEE Signal Process. Education Workshop (DSP/SPE), 2009, pp. 66–71, 4–7.
[18] G. Mateos, I. D. Schizas, and G. B. Giannakis, “Performance analysis of the consensus-based distributed LMS algorithm,” EURASIP J. Adv. Signal Processing, vol. 2009, p. 19, 2009, Article ID 981030.
[19] G. Mateos, I. D. Schizas, and G. B. Giannakis, “Distributed recursive least-squares for consensus-based in-network adaptive estimation,” IEEE Trans. Signal Process., vol. 57, no. 11, pp. 4583–4588, 2009.
[20] C. E. Davila, “An efficient recursive total least squares algorithm for FIR adaptive filtering,” IEEE Trans. Signal Process., vol. 42, pp. 267–280, 1994.
[21] I. Markovsky and S. Van Huffel, “Overview of total least-squares methods,” Signal Processing, vol. 87, no. 10, pp. 2283–2302, 2007.
[22] Y. Huang, J. Benesty, G.W. Elko, “Adaptive eigenvalue decomposition algorithm for real-time acoustic source localization system”, IEEE International Conference Acoustics, Speech, Signal Processing, Vol. 2, 1999, pp. 937–940.
[23] A. H. Sayed,: “Adaptation, Learning, and Optimization over Networks,” Foundations and Trends in machine learning, vol. 7, no. 4;5, pp. 311-801, 2014.

Zohre Foroushi received the BS and the MS degrees from Iran University of Science and Technology, Tehran, Iran, in 2007 and 2010. Since 2010, she has been working towards the Ph.D degree under the supervision of Prof. V. TabatabaVakili at the Iran University of Science and Technology (IUST).

Vahid Tabata Vakili received the BS degree from Sharif University of Technology, Tehran, Iran, In 1970, the MS degree from the University of Manchester, Manchester, U.K., in 1973, and the Ph.D. degree from the University of Bradford, Bradford, U.K., in 1977, all in Electrical Engineering. In 1985, he joined the Department of Electrical Engineering, Iran University of Science and Technology, Tehran. In 1997, he was promoted as Professor