International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:3, No:11, 2009

Distributed Estimation Using an Improved Incremental Distributed LMS Algorithm

Amir Rastegarnia, Mohammad Ali Tinati, and Azam Khalili

***Abstract*—**In this paper we consider the problem of distributed adaptive estimation in wireless sensor networks for two different observation noise conditions. In the first case, we assume that there are some sensors with high observation noise variance (*noisy sensors*) in the network. In the second case, different variance for observation noise is assumed among the sensors which is more close to real scenario. In both cases, an initial estimate of each sensor's observation noise is obtained. For the first case, we show that when there are such sensors in the network, the performance of conventional distributed adaptive estimation algorithms such as incremental distributed least mean square (IDLMS) algorithm drastically decreases. In addition, detecting and ignoring these sensors leads to a better performance in a sense of estimation. In the next step, we propose a simple algorithm to detect theses noisy sensors and modify the IDLMS algorithm to deal with noisy sensors. For the second case, we propose a new algorithm in which the step-size parameter is adjusted for each sensor according to its observation noise variance. As the simulation results show, the proposed methods outperforms the IDLMS algorithm in the same condition.

***Keywords*—**Distributes estimation, sensor networks, adaptive filter, IDLMS.

I. Introduction

SENSOR networks are emerging as a key technology for a variety of applications [1]. In all of the supposed applications, each node in the network could collect noisy observations related to a certain parameter or phenomenon of interest. It is necessary to exploit spatial dimension alongside the temporal dimension in order to enhance the robustness of the processing tasks [2]. In general, there are two different strategies to process the collected data from the sensors including, I) centralized processing and II) distributed processing. In a centralized solution, the collected noisy observations are sent to a central location for processing known as fusion center (FC). The central processor would then perform the definite process on the data and broadcast the result back to the network. On the other hand, in distributed processing the estimation task is divided between sensors and FC.

In some distributed estimation schemes, the FC is perfectly removed. In these schemes each sensor interacts with its neighbors in a certain manner, as dictated by the network topology, in order to obtain a suitable estimate of the desired parameter [3-7]. In comparison, centralized solution requires a powerful processor in the central node, in addition to extensive amount of communication between the nodes. Recently, distributed adaptive estimation algorithms are proposed to enhance the performance of distributed estimation algorithms over sensor networks [3-7]. In these algorithms, a network of nodes are equipped such that to function as an adaptive entity. In [3,4] one distributed adaptive algorithm using incremental optimization technique is developed. The resulting algorithm, known as IDLMS, is distributed, cooperative, and able to respond in real time to changes in the environment. In IDLMS algorithm, each node is allowed to communicate with its immediate neighbor in order to exploit the spatial dimension while limiting the communications burden at the same time. When more communication resources are available the diffusion implementation is possible that each node communicates with all its neighbors as dictated by the network topology. The algorithms given in [6] and [7] are based on diffusion implementation. In general, diffusion based algorithms have better performance but more complexity than incremental-based one [7].

In the existing distributed adaptive estimation algorithms, either equal observation noise is assumed for all the nodes or same algorithm is used for a different observation noise condition. In practice, equal observation noise assumption fails due to the many physical problems. On the other hand, as it is shown in this paper, when same strategy is used for different conservation noise condition, the performance of the distributed adaptive estimation algorithms drastically decreases.

To address the mentioned problems, in this paper we consider the problem of distributed adaptive estimation in two different cases: I) when there are some noisy sensors in the network, and II) a more realistic scenario, when each sensor's observation noise is different from the other. For each case, we propose new algorithms that are based on the IDLMS. In the proposed algorithms first an initial estimate of each sensor's observation noise is obtained. In the next step using the obtained initial estimate of the unknown parameter, the variance of the observation for each sensor is estimated. Finally according to the estimated observation noise and the IDLMS algorithm is modified. For the first case, the noisy sensors are ignored from the estimation process and for the second case; the step-size parameter is adjusted for each sensor according to its estimated observation noise variance.

II. Estimation Problem And The Adaptive Distributed Solution

A. Notation

A list of the symbols used through the paper, for ease of reference, are shown in Table I.

Amir Rastegarnia is with the Faculty of Electrical and Computer Engineering, University of Tabriz, Tabriz, Iran, email: a_rastegar@ieee.org
Azam Khalili is with the Faculty of Electrical and Computer Engineering, University of Tabriz, Tabriz, Iran, email: a-khalili@tabrizu.a.cir
Mohammad Ali Tinati is Professor and Chairman of Electrical Engineering with the University of the of Tabriz, Tabriz, Iran, email: tinati@tabrizu.ac.ir

TABLE I
LIST OF THE MAIN USED SYMBOLS

Symbol	Description
w_i	Weight vector estimate at iteration i
$e(i)$	Output estimation at iteration i
$d(i)$	Value of a *scalar* variable d at iteration i
u_i	Value of a *vector* variable u at iteration i

B. Problem Statement

Here the goal is to estimate the unknown vector w^o from multiple spatially independent but possibly time-correlated measurements collected at N nodes in a network [3] (See Fig. 1).

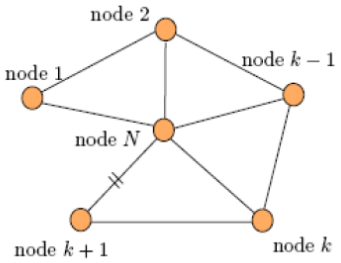

Fig. 1. A distributed network with N active nodes accessing space-time data.

Each node k has access to time-realizations $\{d_k(i), u_{k,i}\}$ of zero-mean spatial data $\{d_k, u_k\}$ where each d_k is a scalar measurement and each u_k is a $1\times M$ row regression vector. We assume that the unknown vector w^o relates to the $\{d_k(i), u_{k,i}\}$ by

$$d_k(i) = u_{k,i}w^o + v_k(i), \tag{1}$$

where $v_k(i)$ is some white noise sequence (also known as *observation noise*) with variance $\sigma^2_{v,k}$, and is independent of $\{d_k(i), u_{k,i}\}$. By collecting regression and measurement data into global matrices results:

$$U \triangleq col\{u_1, u_2, ..., u_N\} \tag{2}$$

$$d \triangleq col\{d_1, d_2, ..., d_N\} \tag{3}$$

where the notation $\mathrm{col}\{\cdot\}$ denotes a column vector (or matrix) with the specified entries stacked on top of each other. The objective is to estimate the vector that solves

$$\min_w J(w) \; where \; J(w) = E\|d - Uw\|^2 \tag{4}$$

The optimal solution satisfies the normal equations [8]

$$R_{du} = R_u w^o \tag{5}$$

where

$$R_{du} = E\{U^* d\} = \sum_{k=1}^{N} R_{du,k}, \tag{6}$$

$$R_u = E\{U^* U\} = \sum_{k=1}^{N} R_{u,k}. \tag{7}$$

where in (6), the symbol * denotes the Hermitian transform. Note that in order to use (5) to compute w^o each node must have access to the global statistical information $\{R_u, R_{du}\}$ which in turn requires more communications between nodes and computational resources [3,4]. To obtain a distributed algorithm for computing w^o we begin by steepest descent method as follows.

C. Incremental Distributed LMS Solution

The standard gradient-descent implementation to solve the normal equation (5) is as

$$w_i = w_{i-1} + \mu\left[\nabla J\left(w_{i-1}\right)\right]^*, \tag{8}$$

where μ is a suitably chosen step-size parameter, w_i is the estimate of desired parameter (i.e. w^o) in ith iteration and $\nabla J(\cdot)$ denotes the gradient vector of $J(w)$ evaluated at w_{i-1}. If μ is sufficiently small then $w_i \to w^o$ as $i \to \infty$ [3-6]. In order to obtain a distributed version of (8), first the cost function $J(w)$ is decomposed as

$$J(w) = \sum_{k=1}^{N} J_k(w), \tag{9}$$

where

$$J_k \triangleq E\left\{|d_k - U_k w|^2\right\}. \tag{10}$$

Using (9) and (10) the standard gradient-descent implementation of (8) can be rewritten as [3-6]:

$$w_i = w_{i-1} - \mu\left[\sum_{k=1}^{N} \nabla J_k\left(w_{i-1}\right)\right]^* \tag{11}$$

By defining the $\psi_k^{(i)}$ as the local estimate of the w^o at node k and time i, then w_i can be evaluated as [3]

$$\psi_k^{(i)} = \psi_{k-1}^{(i)} - \mu\left[\nabla J_k\left(w_{i-1}\right)\right]^*, \; k = 1, 2, \ldots, N \tag{12}$$

This scheme still requires all node to share global information w_{i-1}. The fully distributed solution can be achieved by using the local estimate $\psi_k^{(i)}$ at each node k instead of w_{i-1}, i.e.

$$\psi_k^{(i)} = \psi_{k-1}^{(i)} - \mu\left[\nabla J_k\left(\psi_k^{(i)}\right)\right]^*, \; k = 1, 2, \ldots, N \tag{13}$$

Now, we need to determine the gradient of J and replace it in (13). To do this, the following approximations are used

$$R_{du,k} \approx d_k(i)\, u_{k,i}^* \tag{14}$$

$$R_{u,k} \approx u_{k,i}^* u_{k,i} \tag{15}$$

The resulting IDLMS algorithm is as follows [5]

$$\begin{cases} \psi_0^{(i)} \leftarrow w_{i-1} \\ \psi_k^{(i)} = \underset{k=1,2,\ldots,N}{\psi_{k-1}^{(i)} - \mu\, u_{k,i}^*}\left[d_k(i) - u_{k,i}\psi_{k-1}^{(i)}\right] \\ w_i \leftarrow \psi_N^{(i)} \end{cases} \tag{16}$$

III. Proposed Algorithm

A. Motivation

As mentioned in the introduction section, in the existing distributed adaptive estimation algorithms such as IDLMS either equal observation noise is assumed for all the nodes in the network or same algorithm is used for a different observation noise condition. Now, we consider a situation where there are some noisy sensors in the network. Using these noisy sensors to update the $\psi_k^{(i)}$ in the IDLMS algorithm (see (16)) will cause a severe decrease in the algorithm's performance. To show this we consider a network with $N = 40$ nodes and assume Gaussian regressors with $R_{u,k} = I$. We further assume that there are $N_s = 10$ noisy sensors in the network and background white noise for these sensors has a variance of $\sigma_v^2 = 10^{-1}$ whereas for other sensors this quantity is equal to $\sigma_v^2 = 10^{-3}$. The step-size parameter is chosen $\mu = 0.01$.

In Fig. 2 the mean square error (MSE) performance of IDLMS algorithm for two different cases where there exist a number of noisy sensors in the network and when these noisy sensors are ignored is compared. As it is clear from Fig. 2, considering the noisy sensors in the IDLMS algorithm makes a severe decrease in the performance of IDLMS. On the other hand if these sensors are perfectly known and ignored in IDLMS update process, the better result can be achieved.

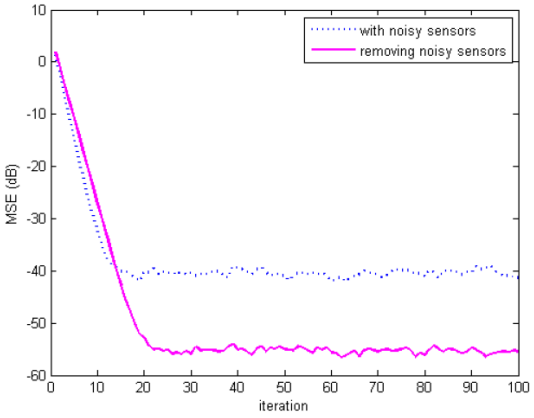

Fig. 2. Performance of the IDLMS algorithm in two different cases.

The main feature of noisy sensors that distinguishes them from other sensors is their observation noise variances. So to recognize the noisy sensors it is necessary to obtain an initial estimate of the observation noise at each node in the network. The following section describes our proposed method to enhance the IDLMS algorithm.

B. Case I: There are Some Noisy Sensors in the Network, Proposed Algorithm

The proposed method to enhance the poor performance of IDLMS algorithm consisting of following steps

1) Obtaining an initial estimate of w^o.
2) Estimating the observation noise in each node.
3) Detecting the noisy sensors and using modified IDLMS algorithm.

To obtain an initial estimate of the observation noise at each node, we consider again the equation (1). Using (16) and repeating it L_s times (where L_s is a suitably chosen integer), it is possible to have an primary estimate of w^o. It must be noted that the initial estimate of w^o is used just to obtain an estimate of observation noise at each sensor and this estimate is not the final estimate of w^o. We denote the estimate of w^o in the L_sth iteration and in Nth node by $\psi_k^{(Ls)}$, i.e.

$$\psi_N^{(L_s)} = \psi_k^{(i)}|_{k=N,i=L_s} \tag{17}$$

Using (1) and (17) the observation noise at each sensor can be estimated as

$$n_k(i) = d_k(i) - u_{k,i}\psi_N^{(L_s)},\ i = 1, 2, ..., L_s \tag{18}$$

In the nest step we define the following vector for each sensor

$$B_k(i) = n_k(i),\ i = 1, 2, ..., L_s \tag{19}$$

To use the fact that the observation noise for noisy sensors are bigger than the other sensors we define the following parameters for each sensor

$$\underset{k=1,2,...,N}{M_k} = \left(\frac{1}{L_s}\right)\sum_{i=1}^{L_s} B_k(i). \tag{20}$$

$$\tilde{\sigma}_k = \sum_{i=1}^{L_s}\left(B_k(i) - M_k\right)^2 \tag{21}$$

Now by using the vectors σ_k and a suitable threshold it is possible to recognize the noisy sensors in the network. The main drawback of this method for noisy sensors detection is that for any specific network a new threshold must be chosen. To address this problem, we do as follows: First we define the following parameters

$$V_k = \frac{1}{N}\sum_{k=1}^{N}\tilde{\sigma}_k. \tag{22}$$

$$T_k = \tilde{\sigma}_k - V \tag{23}$$

In Fig. 3 the $\tilde{\sigma}_k$ and T_k are plotted.

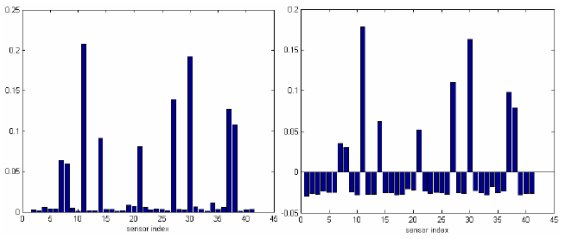

Fig. 3. The $\{\tilde{\sigma}_k\}_{k=1}^N$ (left) and $\{T_k\}_{k=1}^N$ (right).

As it is clear from Fig. 3, noisy sensors in general have $T_k > 0$ so finding the sensors with this feature is equivalent to (with high probability) finding the noisy sensors. As the noisy sensors are detected, for the rest of w^o estimation process, (i.e. $i = L_s + 1, L_s + 2, ...$) the update equation (16) is modified as follows

- For noisy sensors

$$\psi_k^{(i)} = \psi_{k-1}^{(i)}, \quad i > L_s,\ k = 1, 2, ..., N \tag{24}$$

- For other sensors

$$\underset{k=1,2,...,N}{\psi_k^{(i)} = \psi_{k-1}^{(i)}} - \mu\, u_{k,i}^* \left[d_k(i) - u_{k,i}\psi_{k-1}^{(i)} \right] \tag{25}$$

As $i \to \infty$ all nodes will contain the appropriate estimate of w^o i.e.

$$\lim_{i\to\infty} \psi_k^{(i)} \to w^o,\ k = 1, 2, ..., N \tag{26}$$

C. Case II: Different Observation Noise Variance, Proposed Algorithm

In this case we consider again the $\tilde{\sigma}_k$ as defend in (21). It must be noticed that as $\tilde{\sigma}_k$ increases, the reliability of d_k decreases, so there is an inverse relation between $\tilde{\sigma}_k$ and sensor's reliability. Keeping this fact in mind, we define a new parameter as

$$s_k = \frac{1}{\tilde{\sigma}_k},\ k = 1, 2, ..., N \tag{27}$$

In the next step, we define the following parameter

$$\lambda_k = \max\{s_k\},\ k = 1, 2, ..., N, \tag{28}$$

Finally, we define the step-size of the our incremental LMS algorithm as

$$\mu_k = \mu_{\text{glob}} \lambda_k^{-1} s_k \tag{29}$$

where μ_{glob} is the global step-size parameter which is constant for all sensors. Finally, the IDLMS algorithm is modified as follows

$$\underset{k=1,2,...,N}{\psi_k^{(i)} = \psi_{k-1}^{(i)}} - \mu_k\, u_{k,i}^* \left[d_k(i) - u_{k,i}\psi_{k-1}^{(i)} \right] \tag{30}$$

Here again as $i \to \infty$ all nodes will contain the appropriate estimate of w^o.

IV. Simulation Results

A. Simulation Results for Case I

To evaluate the performance of the proposed algorithm in this section the result of simulation results are presented. We consider a network with $N = 40$ nodes and Gaussian regressors with $R_{u,k} = I$. We further assume that there are $N_s = 10$ noisy sensors in the network and background white noise for these sensors has a variance of $\sigma_v^2 = 10^{-1}$ and other sensors $\sigma_v^2 = 10^{-3}$. The curves are obtained by averaging over 200 experiments with $\mu = 0.01$. In the proposed algorithm for enhancement the performance of IDLMS, it is necessary to obtain a suitable primary estimate of w^o which is strongly depends on the value of L_s. Fig. 4 shows the performance of proposed algorithm for $L_s = 2$. It is obvious from Fig. 4 that by choosing an improper value for L_s, both IDLMS and proposed algorithm have same performance. When the L_s is chosen properly, the proposed algorithm outperforms than IDLMS. This is shown in Fig. 5 in which we set $L_s = 10$. As Fig. 5 shows the proposed algorithm is capable of detecting noisy sensors and in addition by using it, the performance

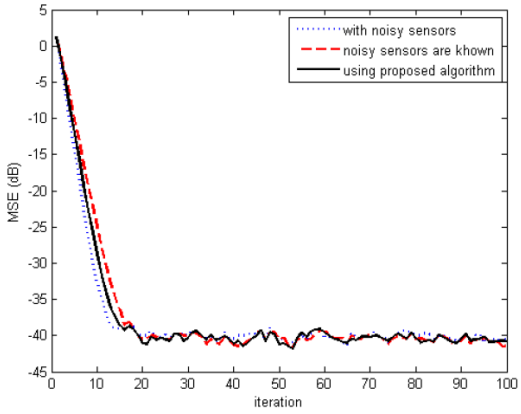

Fig. 4. The effect of choosing an improper L_s, ($L_s = 2$).

of IDLMS increases drastically. It must be noticed that by increasing $L_s > 10$ (in this example) no better estimate of $\psi_k^{(L_s)}$ can be achieved and as a result no better estimate of observation noise variances can obtained. However, using the obtained estimate of noise variances in the proposed algorithm helps to find a better estimate of interested parameter w^o, (see Fig. 5).

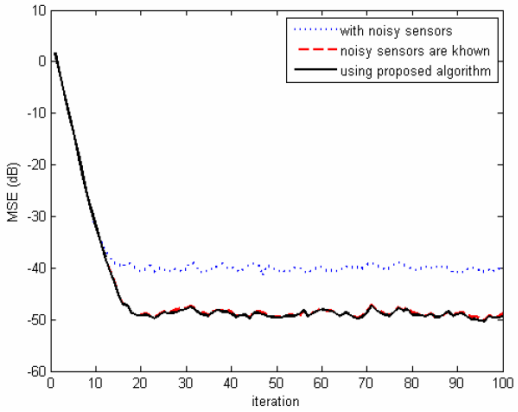

Fig. 5. The MSE performance for different algorithms.

B. Simulation Results for Case II

In this case we consider the same assumptions as in case I except that that this time observation noise for sensors are between $\sigma_v^2 = 10^{-3}$ and $\sigma_v^2 = 10^{-1}$. We choose $\mu_{\text{glob}} = 0.01$. In Fig. 6 the performance of the proposed algorithm for $L_s = 20$ in comparison with the conventional IDLMS algorithm is depicted. To compare the performance of different algorithms we use the MSE criteria again. As it is clear from Fig. 6, the proposed algorithm has better performance than IDLMS in a sense of steady state estimation error. The step size for different sensors (i.e. vector μ_k) is plotted in Fig. 7. In the proposed algorithm, it is necessary to choose a suitable value for parameter L_s since it determines how $\psi_k^{(L_s)}$ is close to

w^o. As our simulation results show, $L_s = L/10$ (where L is the total of number of iterations) is a good choice. In the Fig. 8, the performance of the IDLMS and proposed algorithm for different (and proper) L_s values is plotted. As Fig. 8 shows, regardless of the value of L_s, the proposed algorithm has better performance that the IDLMS algorithm in the same condition.

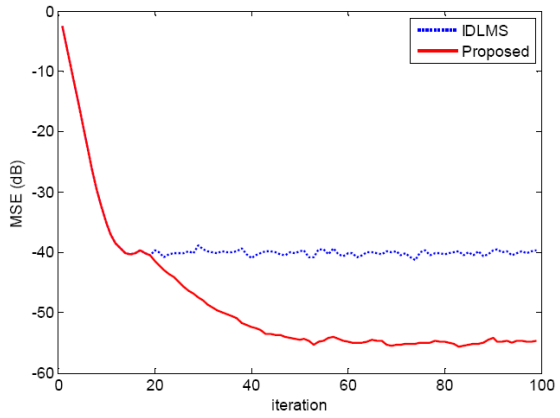

Fig. 6. The performance of the proposed algorithm ($L_s = 20$).

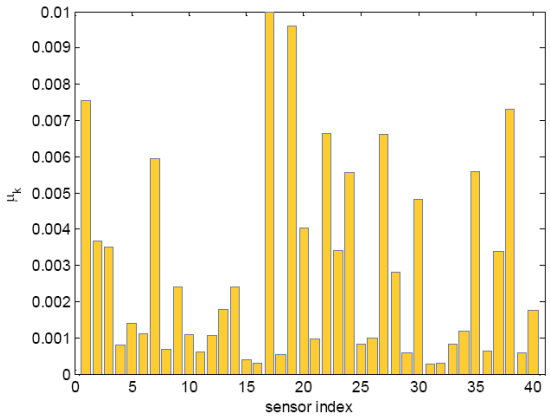

Fig. 7. The step-size for different sensors.

V. Conclusions

In this paper the issue of distributed adaptive estimation over sensor networks in two different cases was considered. In the first case, we assumed that there exists some high observation sensors in the network, and in the second case, different observation noise were regarded among the sensors. In both cases, first an estimate of each sensor's observation noise were obtained and then, according to the extracted information, the IDLMS algorithm was modified. As our simulation results shows, the proposed algorithm in both cases outperform the IDLMS algorithm given in the literature.

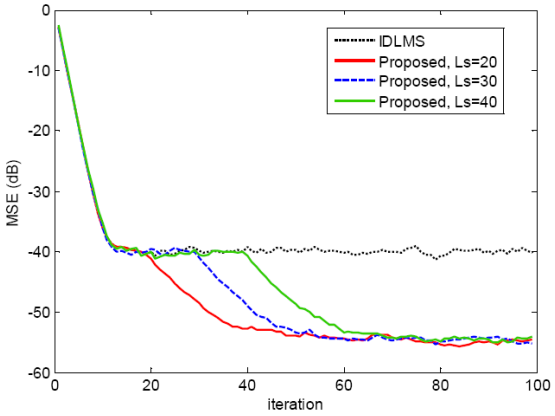

Fig. 8. The performance of the IDLMS and proposed algorithm for different values of L_s.

Acknowledgment

The Authors would like to thank Iran telecommunication research center (ITRC) for their financial support of this research.

References

[1] D. Estrin, G. Pottie and M. Srivastava, Intrumenting the world with wireless sensor networks, *Proc. IEEE ICASSP*, pp. 2033-2036, May 2001.
[2] D. Bertsekas, A new class of incremental gradient methods for least squares problems, *SIAM J. Optim.*, vol.7, no. 4, pp. 913-926, Nov.1997.
[3] [3] Lopes, C. G. and Sayed, A. H., Distributed adaptive incremental strategies: formulation and perform, *Proc. ICASSP 06*, 3: 584-587, 2006.
[4] Lopes, C. G. and Sayed, A. H., Incremental adaptive strategies over distributed networks, *IEEE trans. on signal processing*, vol. 55, pp. 4064-4077, 2007.
[5] Rabbat M. G. and Nowak, R. D., Quantized incremental algorithms for distributed optimization, *IEEE Journal on Sel. Areas in Comm.*, vol. 23, pp. 798-808, 2005.
[6] Lopes, C. G. and Sayed, A. H., Diffusion least- mean squares over adaptive networks, *Proc. ICASSP 07*, vol. 3, pp. 917-920, 2007.
[7] Sayed, A. H. and Lopes, C. G., Distributed recursive least-squares strategies over adaptive networks, *Proc. Asilomar Conference on Signals, Systems and Computers*, pp. 233-237, 2006.
[8] A. H. Sayed, *Fundamentals of Adaptive Filtering*, Wiley, NJ, 2003.