Unsupervised Algorithms For Distributed Estimation Over Adaptive Networks

M. O. Bin Saeed, *Member, IEEE,* A. Zerguine, *Senior Member, IEEE,* S. A. Zummo, *Senior Member, IEEE,* and A. H. Sayed, *Fellow, IEEE*

***Abstract*—This work shows how to develop distributed versions of block blind estimation techniques that have been proposed before for batch processing. Using diffusion adaptation techniques, data are accumulated at the nodes to form estimates of the auto-correlation matrices and to carry out local SVD and/or Cholesky decomposition steps. Local estimates at neighborhoods are then aggregated to provide online streaming estimates of the parameters of interest. Simulation results illustrate the performance of the algorithms.**

Index Terms – Blind estimation, diffusion strategy, SVD, Cholesky factorization, auto-correlation matrix.

I. Introduction

This work studies the problem of blind distributed estimation over an ad-hoc network. We consider a set of N sensor nodes spread over a geographic area as shown in Fig. 1. Sensor measurements are taken at each node at every time instant. The objective of the network is to estimate an unknown parameter vector using the node measurements. Several algorithms have been devised in the literature for distributed estimation using diffusion strategies or consensus strategies— see, e.g., [1]-[7]

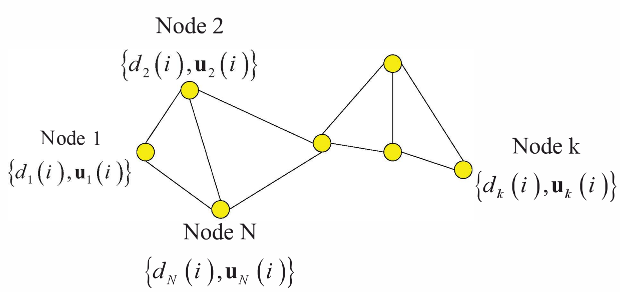

Fig. 1. A network of N nodes collecting measurements $\{d_k(i), \mathbf{u}_k(i)\}$.

In these previous works on distributed estimation, it is generally assumed that the regression data, $\mathbf{u}_k(i)$, are available at the sensors. If this information is not available, then the estimation problem needs to be solved in a *blind* manner. There have been considerable contributions in the literature to the problem of blind estimation, but mainly in the context of stand-alone filters. Among the many useful works on blind estimation and equalization, we may mention [8]-[12].

In this paper we extend the diffusion strategy of [1], [2] to the case of distributed blind estimation over networks. Diffusion strategies have several useful properties compared to other distributed techniques in terms of their convergence rate, mean-square performance, robustness, stability and scalability [3].

Notation. We use boldface letters for vectors/matrices and normal font for scalar quantities. Matrices are defined by capital letters and vectors by small letters. The notation $(.)^T$ stands for transposition for vectors and matrices and $E[\cdot]$ denotes expectation.

II. Problem Statement

Each node k has access to a time realization of a scalar process $d_k(i)$, which is assumed to satisfy the linear regression model:

$$d_k(i) = \mathbf{u}_k^T(i)\mathbf{w}^o + v_k(i), \tag{1}$$

where $\mathbf{u}_k(i)$ is the $M \times 1$ unavailable regression vector, $v_k(i)$ is a spatially independent zero-mean additive white noise with variance $\sigma_{v,k}^2$ and i denotes the time index. The objective is to estimate the unknown $M \times 1$ vector $\mathbf{w}^o$ through local collaboration among the nodes by using only the sensed data $d_k(i)$. The estimate of the unknown vector by node k at time i is denoted by $\mathbf{w}_k(i)$. Assuming that each node cooperates only with its neighbors, each node k has access to estimates $\mathbf{w}_l(i)$ from its neighborhood $l \in \mathcal{N}_k$, where $\mathcal{N}_k$ denotes the set of neighbors of node k.

III. Batch Blind Algorithms

The analysis that follows benefits from the contributions of [10], [11] in the context of blind channel estimation for stand-alone nodes. The work in [10] exploits the data second-order-statistics to estimate the unknown parameter vector, while reference [11] simplifies the construction of [10] at the cost of performance degradation. In the sequel, we show how the diffusion framework for adaptation and estimation over networks from [1]-[3] can be extended to handle similar blind estimation formulations.

M. O. Bin Saeed, A. Zerguine and S. A. Zummo are with the Department of Electrical Engineering, King Fahd University of Petroleum & Minerals, Dhahran, 31261, KSA (e-mail: {mobs,azzedine,zummo}@kfupm.edu.sa).

A. H. Sayed is with the Department of Electrical Engineering, University of California, Los Angeles, CA 90095 USA (e-mail: sayed@ee.ucla.edu).

A. Singular Value Decomposition-Based Solution

The batch processing version of the singular value decomposition (SVD) method involves the following steps. Select an integer $L \geq M$ and let $P = 2M - 1$. Introduce the following $P \times 1$ vector with $P - M$ trailing zeros

$$\mathbf{u}_k(i) = [u_{k,0}(i), ..., u_{k,M-1}(i), 0, ..., 0]^T, \tag{2}$$

Introduce further the following convolution matrix of size $P \times P$ and defined in terms of the entries of the unknown parameter vector $w^o = [w_o, \ldots, w_{M-1}]^T$:

$$\mathbf{W} = \begin{bmatrix} w_0 & \cdots & w_{M-1} & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & w_0 & \cdots & w_{M-1} \end{bmatrix}^T. \tag{3}$$

A total of P data points are collected at node k at each time i. Collecting these measurements into a vector $\mathbf{d}_k(i)$, we find that it satisfies:

$$\mathbf{d}_k(i) = \mathbf{W}\mathbf{u}_k(i) + \mathbf{v}_k(i), \tag{4}$$

where $\mathbf{v}_k(i) = [v_{k,0}(i), \cdots, v_{k,P-1}(i)]^T$ is the noise vector. The output blocks $\{\mathbf{d}_k(i)\}$ are collected together over a period of size N to form the matrix:

$$\mathbf{D}_{k,N} = (\mathbf{d}_k(0)\ \mathbf{d}_k(1)\ \cdots\ \mathbf{d}_k(N-1)), \tag{5}$$

where it is assumed that sufficient data are collected for $\mathbf{D}_{k,N}$ to be full-rank. The SVD of the auto-correlation of $\mathbf{D}_{k,N}$ gives a set of null eigenvectors. Specifically, if we introduce the eigen-decomposition:

$$\mathbf{D}_{k,N}\mathbf{D}_{k,N}^T = \begin{pmatrix} \bar{\mathbf{S}}_k & \tilde{\mathbf{S}}_k \end{pmatrix} \begin{pmatrix} \mathbf{\Sigma}_{k,M\times M} & \mathbf{0}_{M\times(M-1)} \\ \mathbf{0}_{(M-1)\times M} & \mathbf{0}_{(M-1)\times(M-1)} \end{pmatrix} \begin{pmatrix} \bar{\mathbf{S}}_k^T \\ \tilde{\mathbf{S}}_k^T \end{pmatrix}. \tag{6}$$

Then, the columns of the $P \times (M-1)$ matrix $\tilde{\mathbf{S}}_k$ form a basis for the null space of $\mathbf{D}_{k,N}$. This implies that for the case where there is no added noise

$$\tilde{\mathbf{s}}_{\hat{k}}^T \mathbf{W} = \mathbf{0}, \tag{7}$$

where $\hat{k} = 1, ..., M-1$ and $\tilde{\mathbf{s}}_{\hat{k}}$ is the $\hat{k}_{th}$ column of $\tilde{\mathbf{S}}_k$. Since $\mathbf{W}$ is a convolution matrix, equation (7) can be rewritten as:

$$\mathbf{w}^{oT}\mathcal{S}_k := \mathbf{w}^{oT}[\mathcal{S}_{k,1}...\mathcal{S}_{k,M-1}] = \mathbf{0}^T, \tag{8}$$

where $\mathcal{S}_{k,\hat{k}}$ is an M × M Hankel matrix given by

$$\mathcal{S}_{k,\hat{k}} = \begin{bmatrix} \tilde{\mathbf{s}}_{\hat{k}}(0) & \tilde{\mathbf{s}}_{\hat{k}}(1) & \cdots & \tilde{\mathbf{s}}_{\hat{k}}(M-1) \\ \vdots & \vdots & \vdots & \vdots \\ \tilde{\mathbf{s}}_{\hat{k}}(M-1) & \tilde{\mathbf{s}}_{\hat{k}}(M) & \cdots & \tilde{\mathbf{s}}_{\hat{k}}(P-1) \end{bmatrix}. \tag{9}$$

It is important to note that in the presence of noise (8) becomes approximately equal to zero. The desired parameter estimate can be obtained by solving (8) up to a constant factor.

B. Cholesky Factorization Based Solution

The work in [11] describes a method that replaces the SVD operation with a Cholesky factorization. Evaluating the auto-correlation of $\mathbf{d}_k(i)$ in equation (4) and assuming the regression data arises from a zero-mean white noise process with variance $\sigma^2_{\mathbf{u}_k}$, we get

$$\mathbf{R}_{\mathbf{d}_k} = \sigma^2_{\mathbf{u}_k}\mathbf{W}\mathbf{W}^T + \sigma^2_{v_k}\mathbf{I}. \tag{10}$$

Now if the second-order statistics of both the input regressor data as well as the additive noise are known beforehand then we can recover the matrix:

$$\begin{aligned} \mathbf{R}_{\mathbf{w}^o} &\triangleq \mathbf{W}\mathbf{W}^T \\ &= \left(\mathbf{R}_{\mathbf{d}_k} - \sigma^2_{v_k}\mathbf{I}\right)/\sigma^2_{\mathbf{u}_k}. \end{aligned} \tag{11}$$

As such, the Cholesky decomposition of $\mathbf{R}_{\mathbf{w}^\bullet}$ can be related to the desired channel vector. However, the information about the input regressor data is not always known. Therefore, $\mathbf{R}_{\mathbf{w}^\bullet}$ needs to be approximated. The algorithm in [11] uses the Cholesky factor of this estimated matrix to provide a least-squares estimate for the unknown parameter vector.

The method is summarized as follows. Using K blocks of data we compute the ensemble average:

$$\hat{\mathbf{R}}_{\mathbf{d}_k} = \frac{1}{K}\sum_{i=1}^{K}\mathbf{d}_k(i)\mathbf{d}_k^T(i), \tag{12}$$

and estimate $\mathbf{R}_{\mathbf{w}^\bullet}$ at node k as:

$$\hat{\mathbf{R}}_{\mathbf{w}_k} = \hat{\mathbf{R}}_{\mathbf{d}_k} - \hat{\sigma}^2_{v_k}\mathbf{I}, \tag{13}$$

where $\hat{\sigma}^2_{v_k}$ is the noise variance estimated by averaging the smallest eigenvalues of $\hat{\mathbf{R}}_{\mathbf{d}_k}$ [11]. We now determine the upper triangular Cholesky factor of $\hat{\mathbf{R}}_{\mathbf{w}_k}$ and define the vector:

$$\hat{\mathbf{g}}_k = \mathrm{vec}\left\{\mathrm{chol}\left\{\hat{\mathbf{R}}_{\mathbf{w}_k}\right\}\right\}, \tag{14}$$

where $\mathrm{chol}\{.\}$ is the Cholesky factorization operation. It is shown in [11] that the actual vector $\mathbf{g}_k$ and the parameter vector $\mathbf{w}^o$ are related through $\mathbf{g}_k = \mathbf{Q}\mathbf{w}^o$, where $\mathbf{Q}$ is an $M^2 \times M$ selection matrix given by $\mathbf{Q} = \left[\mathbf{J}_1^T\mathbf{J}_2^T...\mathbf{J}_M^T\right]^T$, and the $M \times M$ matrices $\mathbf{J}_q$ are defined as

$$[\mathbf{J}_q]_{r,t} = \begin{cases} 1, & \text{if } r + t = q - 1 \\ 0, & \text{otherwise.} \end{cases} \tag{15}$$

The least-squares estimate of the unknown parameter vector is then constructed as:

$$\hat{\mathbf{w}}_k = \left(\mathbf{Q}^T\mathbf{Q}\right)^{-1}\mathbf{Q}^T\hat{\mathbf{g}}_k. \tag{16}$$

IV. Blind Diffusion Algorithms

The two aforementioned methods require several blocks of data to be stored before estimation can be performed. Although the least-squares approximation gives a good estimate, it is nevertheless a centralized solution. We now show how to develop fully decentralized and recursive solutions.

We start with the SVD-based solution. Rather than collect blocks of data at each node for every time instant, we use the available $\mathbf{d}_k(i)$ to update the estimate of the auto-correlation matrix iteratively as follows:

$$\hat{\mathbf{R}}_{\mathbf{d}_k}(i) = \hat{\mathbf{R}}_{\mathbf{d}_k}(i-1) + \mathbf{d}_k(i)\mathbf{d}_k^T(i). \quad (17)$$

As more data blocks are processed, the rank of $\hat{\mathbf{R}}_{\mathbf{d}_k}(i)$ becomes gradually full. Next, computing the SVD of $\hat{\mathbf{R}}_{\mathbf{d}_k}(i)$, we obtain the $\mathrm{P} \times (\mathrm{M}-1)$ null space matrix $\tilde{\mathbf{S}}_k$. From $\tilde{\mathbf{S}}_k$, we then form the M Hankel matrices of size $\mathrm{M} \times \mathrm{M}$ each, which are then concatenated to form the matrix $\mathcal{S}_k(i)$ from which the estimate $\bar{\mathbf{w}}_k(i)$ is finally derived. This sequential derivation process is summarized below

$$\mathrm{SVD}\left\{\hat{\mathbf{R}}_{\mathbf{d}_k}(i)\right\} \Rightarrow \mathbf{S}_k(i) \Rightarrow \tilde{\mathbf{S}}_k(i) \Rightarrow \mathcal{S}_k(i) \Rightarrow \bar{\mathbf{w}}_k(i). \quad (18)$$

The update for the estimate of the unknown parameter vector is then given by

$$\hat{\mathbf{w}}_k(i) = \delta_k \hat{\mathbf{w}}_k(i-1) + (1-\delta_k)\,\bar{\mathbf{w}}_k(i), \quad (19)$$

where δ_k is a forgetting factor. It can be seen from (18) that the recursive algorithm does not become computationally less complex. However, it does require less memory compared to the original algorithm of [10] and the result improves with an increase in the number of data blocks processed.

With regards to the Cholesky-based construction, expression (12) motivates the following recursive construction as time progresses:

$$\hat{\mathbf{R}}_{\mathbf{w}_k}(i) = \frac{1}{i}\left[\mathbf{d}_k(i)\mathbf{d}_k^T(i) - \hat{\sigma}_{v_k}^2\mathbf{I}\right] + \frac{i-1}{i}\hat{\mathbf{R}}_{\mathbf{w}_k}(i-1). \quad (20)$$

Let

$$\hat{\mathbf{g}}_k(i) = \mathrm{vec}\left\{\mathrm{chol}\left[\hat{\mathbf{R}}_{\mathbf{w}_k}(i)\right]\right\}, \quad (21)$$

and

$$\mathbf{Q}_A = \left(\mathbf{Q}^T\mathbf{Q}\right)^{-1}\mathbf{Q}^T, \quad (22)$$

which can be evaluated offline. Then

$$\bar{\mathbf{w}}_k(i) = \mathbf{Q}_A\hat{\mathbf{g}}_k(i). \quad (23)$$

We further apply a smoothing step to get the final estimate:

$$\hat{\mathbf{w}}_k(i) = \lambda_k(i)\hat{\mathbf{w}}_k(i-1) + (1-\lambda_k(i))\,\bar{\mathbf{w}}_k(i), \quad (24)$$

where $\lambda_k(i) = 1 - \frac{1}{i}$ is a variable forgetting factor.

The works in [1]-[3] proposed diffusion strategies that improve the estimation performance of the network through data sharing. Motivated by these schemes (and especially the adapt-then-combine (ATC) version), we list in the tables below, the distributed versions of the blind solutions described before. In these implementations, each node in the network cooperates with its neighbors to compute its local estimate of the parameter vector. Moreover, the term $\hat{\mathbf{h}}_k(i)$ denotes an intermediate weight estimate by node k at time i.

V. Simulations and Results

We simulate the algorithms for a network with $N = 20$ nodes, shown in Fig. 2. The two blind distributed algorithms of Tables I and II are used to identify an unknown vector of length $M = 4$. The block size is taken as $K = 8$. Figure 3 shows the performance of the proposed algorithms with diffusion (DBC, DBS) and no cooperation (NBC, NBS) for a signal-to-noise ratio (SNR) of 20 dB. As can be seen from this figure, a 5 dB improvement is brought about by the algorithms with diffusion as compared to no cooperation. The performance of the SVD-based algorithm is better than that of the Cholesky-based algorithm. It should be noted that the Cholesky-based algorithm is less complex than the SVD-based algorithm.

SVD-based Diffusion Strategy

Step 1. At each node k and time i, form
$\hat{\mathbf{R}}_{\mathbf{d},k}(i) = \mathbf{d}_k(i)\mathbf{d}_k^T(i) + \hat{\mathbf{R}}_{\mathbf{d},k}(i-1)$

Step 2. Get $\tilde{\mathbf{S}}_k(i)$ from the null space of the SVD of $\hat{\mathbf{R}}_{\mathbf{d},k}(i)$.

Step 3. Form Hankel matrices of size $\mathrm{M} \times \mathrm{M}$ from $\tilde{\mathbf{S}}_k(i)$.

Step 4. Form $\mathcal{S}_k(i)$ by concatenating the Hankel matrices.

Step 5. The null eigenvector from the SVD of $\mathcal{S}_k(i)$ is the estimate $\bar{\mathbf{w}}_k(i)$.

Step 6. Compute the intermediate update $\hat{\mathbf{h}}_k(i)$.
$\hat{\mathbf{h}}_k(i) = \delta_k\hat{\mathbf{w}}_k(i-1) + (1-\delta_k)\,\bar{\mathbf{w}}_k(i)$

Step 7. Combine estimates from neighbors of node k to get $\hat{\mathbf{w}}_k(i)$.
$\hat{\mathbf{w}}_k(i) = \sum_{l \in N_k} c_{lk}\hat{\mathbf{h}}_l(i),$
where the nonnegative coefficients satisfy:
$\sum_{l\in\mathcal{N}_k} c_{lk} = 1, \quad c_{lk} = 0 \text{ if } l \notin \mathcal{N}_k$

TABLE I
Algorithm for SVD-based Diffusion Strategy

Cholesky-based Diffusion Strategy

Step 1. Let $\lambda_k(i) = 1 - \frac{1}{i}$.

Step 2. At each node k and time i, compute
$\hat{\mathbf{R}}_{\mathbf{w},k}(i) = (1-\lambda_k(i))\left(\mathbf{d}_k(i)\mathbf{d}_k^T(i) - \hat{\sigma}_{v,k}^2\mathbf{I}\right) + \lambda_k(i)\hat{\mathbf{R}}_{\mathbf{w},k}(i-1)$

Step 3. Get the Cholesky factor of $\hat{\mathbf{R}}_{\mathbf{w},k}(i)$ and apply the *vec* operator to get $\hat{\mathbf{g}}_k(i)$.

Step 4. The intermediate update is given by
$\hat{\mathbf{h}}_k(i) = (1-\lambda_k(i))\mathbf{Q}_A\hat{\mathbf{g}}_k(i) + \lambda_k(i)\hat{\mathbf{w}}_k(i-1)$.

Step 5. Combine the estimates from the neighbors of node k:
$\hat{\mathbf{w}}_k(i) = \sum_{l \in N_k} c_{lk}\hat{\mathbf{h}}_l(i)$

TABLE II
Algorithm for Cholesky-based Diffusion Strategy

The robustness of the proposed diffusion algorithms is tested under a scenario where five nodes with the maximum number of neighbors switch off during the estimation process after 750 data blocks. The network readjusts accordingly and the results in Fig. 4 show that both diffusion algorithms still perform close to the case where all nodes function perfectly. In this figure, the word "off" indicates the curves obtained for the case when the five nodes stopped working.

VI. Conclusion

This work develops diffusion algorithms for blind estimation over networks; the algorithms are based on the SVD and Cholesky factorization of the auto-correlation matrix and involve cooperation among neighboring nodes to enhance performance.

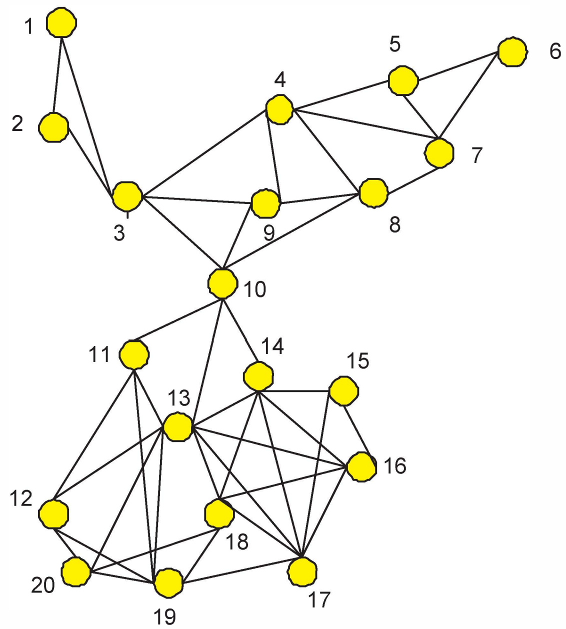

Fig. 2. Network with $N = 20$ nodes used in the simulations.

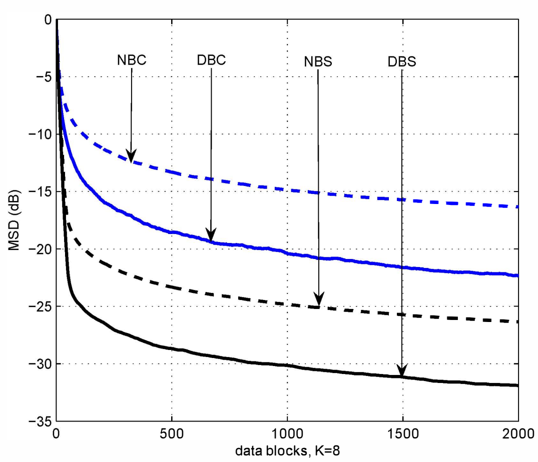

Fig. 3. Comparison of the Mean Square Deviation at SNR=20 dB.

ACKNOWLEDGEMENT

The work of M. O. Bin Saeed, A. Zerguine and S. A. Zummo was funded by King Fahd University of Petroleum & Minerals (KFUPM) under research grant SB101024. The work of A. H. Sayed was supported in part by NSF grants CCF-0942936 and CCF-1011918.

REFERENCES

[1] C.G. Lopes and A.H. Sayed, "Diffusion least-mean squares over adaptive networks: formulation and performance analysis," *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 3122-3136, July 2008.

[2] F. Cattivelli and A. H. Sayed, "Diffusion LMS strategies for distributed estimation," *IEEE Trans. on Signal Process.*, vol. 58, no. 3, pp. 1035-1048, March 2010.

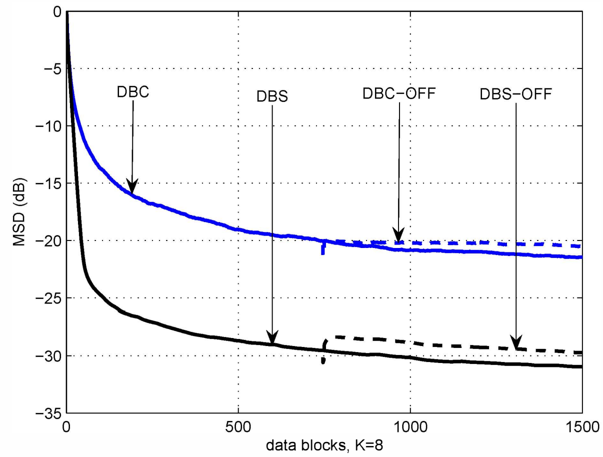

Fig. 4. Robustness of algorithms in case of nodes malfunction.

[3] A. H. Sayed, "Diffusion adaptation over networks," to appear in *E-Reference Signal Processing*, R. Chellapa and S. Theodoridis, editors, Elsevier, 2013. Also available online as arXiv:1205.4220v1, May 2012.

[4] S. Sardellitti, M. Giona, and S. Barbarossa, "Fast distributed average consensus algorithms based on advection-diffusion processes," *IEEE Trans. on Signal Process.*, vol. 58, no. 2, pp. 826-842, Feb. 2010.

[5] L. Li and J. A. Chambers, "Distributed adaptive estimation based on the APA algorithm over diffusion networks with changing topology," *Proc. IEEE Statist. Signal Process. Workshop*, pp. 757-760, Cardiff, Wales, Sep. 2009.

[6] N. Takahashi and I. Yamada, "Link probability control for probabilistic diffusion least-mean squares over resource constrained networks," *Proc. IEEE ICASSP*, pp. 3518-3521, Dallas, TX, Mar. 2010.

[7] M.O. Bin Saeed, A. Zerguine, and S.A. Zummo, "Noise constrained diffusion least mean squares over adaptive networks," in *Proc. PIMRC*, pp. 288–292, Istanbul, Turkey, Sep. 2010.

[8] Y. Sato, "A method of self-recovering equalization for multilevel amplitude-modulation," *IEEE Trans. Commun.*, vol. COM-23, no. 6, pp. 679-682, Jun. 1975.

[9] L. Tong and S. Perreau, "Multichannel blind identification: from subspace to maximum likelihood methods," *Proc. IEEE*, vol. 86, no. 10, pp. 1951-1968, Oct 1998.

[10] A. Scaglione, G.B. Giannakis, and S. Barbarossa, "Redundant filterbank precoders and equalizers part II: blind channel estimation, synchronization, and direct equalization," *IEEE Tran. Signal Proc.*, vol. 47, no. 7, pp. 2007-2022, July 1999.

[11] J. Choi and C.-C. Lim, "A cholesky factorization based approach for blind FIR channel identification," *IEEE Tran. Signal Proc.*, vol. 56, no. 4, pp. 1730-1735, Apr. 2008.

[12] B. Su and P.P. Vaidyanathan, "A generalized algorithm for blind channel identification with linear redundant precoders," *EURASIP J. Adv. Signal Proc.*, vol. 2007, pp. 1-13, 2007, Article ID 25672.

[13] A.H. Sayed, *Fundamentals of Adaptive Filtering*. New York: Wiley, 2003.