2010 IEEE 21st International Symposium on Personal Indoor and Mobile Radio Communications

Noise Constrained Diffusion Least Mean Squares Over Adaptive Networks

Muhammad Omer Bin Saeed, Azzedine Zerguine and Salam A. Zummo
Department of Electrical Engineering
King Fahd University of Petroleum and Minerals
Dhahran, 31261
Saudi Arabia
Email: {mobs,azzedine,zummo}@kfupm.edu.sa

***Abstract*—This paper presents the design of a new diffusion algorithm over adaptive networks. The algorithm assumes knowledge of variance of additive noise. The design is based on the Noise-Constrained Least-Mean Squares (LMS) Algorithm and the new algorithm becomes a type of variable step-size algorithm for which the step-size variation rule results directly from the constraint. The design of the Noise-Constrained Diffusion LMS algorithm has been included. Simulation results show that the new algorithm outperforms the existing Diffusion LMS algorithm as well as its Incremental counterpart.**

***Index Terms*—Adaptive filters, adaptive networks, diffusion, noise constrained algorithms.**

I. Introduction

There has been considerable work done for the problem of distributed estimation over adaptive networks in recent times [1]-[5]. Unlike ordinary algorithms, the nodes in the network cooperate with the closest neighbors in order to estimate some parameters of interest. The spatial and temporal diversity of the network is utilized in order to improve the performance significantly.

Authors have suggested several modes of cooperation, each having its own merits and demerits. In [1], the authors suggest an incremental approach in which each update is simply transferred to the next node which then uses its own data set to improve upon the update and pass it on to the next node. This method is not reliable as node failure will result in a broken link and the whole network would break down. The authors in [2] suggest a diffusion algorithm as a solution to this problem. Nodes share their estimates with the closest neighbors. Each node then combines all estimates using some combiner methodology. The node then performs adaptation on this combined estimate and the new estimate is then diffused into the network. A simpler version of this algorithm is the probabilistic diffusion algorithm [3]. Several combination rules have been used by authors such as Metropolis [4] and relative degree [5]. The authors in [6] used a constrained based approach to come up with their own methodology for combining the estimates from the neighbors and improving the estimate. Adaptive combining has also been tried in order to improve performance [2], [7].

In this paper, we look at a new constrained approach to the problem. We assume knowledge of the additive noise variance [9] and use the Robbins-Munro algorithm [10] to come up with a new LMS algorithm. Extensive simulation results are carried out to assess the performance of the proposed algorithm and as expected improved results are obtained using this technique.

The paper is organized as follows. Section II briefly describes the diffusion LMS algorithm and the adaptive network setup. The new algorithm is then derived in Section III. Section IV gives the mean transient analysis of the new algorithm. Simulation results and comparisons are given in Section V and conclusions are presented in Section VI.

II. Adaptive Networks and Diffusion LMS Solutions

The notation used in this paper is as follows. Boldface letters are used for vectors and matrices while normal letters denote scalars. Matrices are denoted by capital letters and vectors by small letters. The notation $(.)^T$ is used for transpose and $(.)^*$ for conjugate transpose. For scalars, $(.)^*$ denotes complex conjugation. $\mathrm{E}[.]$ denotes the expectation operator.

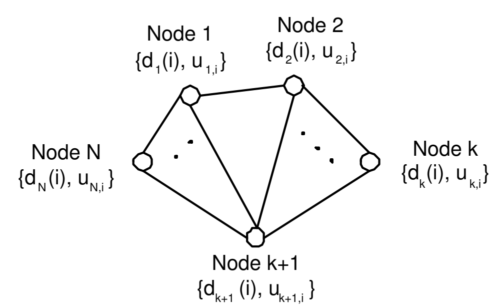

Fig. 1. Adaptive network of *N* nodes

As shown in Fig. 1, consider a network of N nodes arranged in a predefined topology with each node having N_k neighboring nodes, including itself. At time i each node has access to an input regressor row vector $\mathbf{u}_{k,i}$ of length *M* and an output to an unknown system, $d_k(i)$ that are related according to

$$d_k\left(i\right) = \mathbf{u}_{k,i}\mathbf{w}^o + v_k\left(i\right) \tag{1}$$

where $\mathbf{w}^o$ is an unknown column vector of length M defining the parameters of the unknown system and $v_k(i)$ is additive noise. Diffusion LMS algorithms use the data to find an estimate for the unknown vector. There are two different strategies for this estimation. The first strategy collects estimates from its neighbors from the previous iteration and combines them using some convex combiner method. Then the combined estimate is used to calculate a new estimate using the available data $\{d_k(i), \mathbf{u}_{k,i}\}$

$$\begin{cases} \phi_{k,i-1} = \sum\limits_{l \in N_k} c_{lk} \psi_{l,i-1}, \\ \psi_{k,i} = \phi_{k,i-1} + \mu_k u_{k,i}^* \left(d_k\left(i\right) - \mathbf{u}_{k,i} \phi_{k,i-1} \right), \end{cases} \tag{2}$$

where $\phi_{k,i-1}$ is the combined estimate and $\psi_{k,i}$ is the estimate for node k at iteration i. This is known as *Combine-then-Adapt* (CTA) algorithm [2].

Another variation of the algorithm is achieved by reversing the order of the equations. The new algorithm is known as *Adapt-then-Combine* (ATC) algorithm [7], [8].

$$\begin{cases} \phi_{k,i-1} = \psi_{k,i-1} + \mu_k u_{k,i}^* \left(d_k\left(i\right) - \mathbf{u}_{k,i} \psi_{k,i-1} \right), \\ \psi_{k,i} = \sum\limits_{l \in N_k} c_{lk} \phi_{l,i}. \end{cases} \tag{3}$$

Figure 2 shows the *ATC* version of the algorithm.

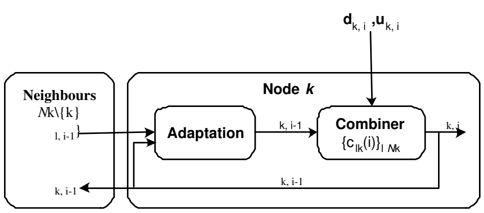

Fig. 2. Adaptive network of *N* nodes

The new algorithm will be tested against both these schemes and it will be shown that performance improves without much increase in complexity.

III. Noise-Constrained LMS Algorithm

A. Problem Formulation

The objective for a diffusion algorithm is to minimize the cost function $J(\mathbf{w})$ with respect to $\mathbf{w}$ where $J(\mathbf{w})$ is given by

$$\min_{\mathbf{w}} J(\mathbf{w}) = \sum_{k=1}^{N} \mathrm{E}\left[\left| d_k - \mathbf{u}_k \mathbf{w} \right|^2 \right]. \tag{4}$$

This is a global cost function. The local cost function at each node would look like [8]

$$J_k\left(\mathbf{w}\right) = \mathrm{E}\left[\left| d_k - \mathbf{u}_k \mathbf{w} \right|^2 \right]. \tag{5}$$

Completing the squares and noting that $\mathrm{E}\left[\mathbf{u}_k^* \mathbf{u}_k\right] = \mathbf{R}_{\mathbf{u},k}$, we can rewrite (5) as

$$J_k\left(\mathbf{w}\right) = \left\| \mathbf{w} - \mathbf{w}_k \right\|^2_{\mathbf{R}_{\mathbf{u},k}} + \mathrm{mmse}. \tag{6}$$

Now the global cost function becomes

$$J\left(\mathbf{w}\right) = \mathrm{E}\left[\left| d_k - \mathbf{u}_k \mathbf{w} \right|^2 \right] + \sum_{l \neq k}^{N} \left\| \mathbf{w} - \mathbf{w}_l \right\|^2_{\mathbf{R}_{\mathbf{u},l}}. \tag{7}$$

This model assumes that each node has access to the entire network. In a practical setup, however, a node has access only to its close neighbors. So the cost function has to be approximated with only data from neighbors being shared at each node. As a result the weighting matrix for the 2nd term does not remain $\mathbf{R}_{\mathbf{u},l}$ but instead has to be replaced by a constant weighting factor b_{lk}. The value of $\mathbf{w}_l$ is also replaced by its intermediate estimate from node l, ψ_l. Eventually, the cost function looks like

$$J_k\left(\mathbf{w}\right) = \mathrm{E}\left[\left| d_k - \mathbf{u}_k \mathbf{w} \right|^2 \right] + \sum_{l \in N_k / \{k\}} b_{lk} \left\| \mathbf{w} - \psi_l \right\|^2. \tag{8}$$

Defining $J_k^1\left(\mathbf{w}\right) = \mathrm{E}\left[\left| d_k - \mathbf{u}_k \mathbf{w} \right|^2 \right]$, we get

$$\min_{\mathbf{w}} J_k\left(\mathbf{w}\right) = J_k^1\left(\mathbf{w}\right) + \sum_{l \in N_k / \{k\}} b_{lk} \left\| \mathbf{w} - \psi_l \right\|^2. \tag{9}$$

B. Algorithm Design

Assuming we have knowledge of the additive noise variance $\sigma^2_{v,k}$ we can modify the cost function as follows. The 2nd term in the above equation goes to zero asymptotically. This means the problem boils down to minimizing $J_k^1(\mathbf{w})$ with respect to $\mathbf{w}$ based on the constraint $J_k^1(\mathbf{w}) = \sigma^2_{v,k}$. The Lagrangian for this problem is [11]

$$\begin{aligned} \min_{\mathbf{w}_k} J_k'\left(\mathbf{w}_k\right) &= J_k^1\left(\mathbf{w}_k\right) + \sum_{l \in N_k} b_{lk} \left\| \mathbf{w}_k - \psi_l \right\|^2 \\ &+ \gamma \lambda \left(J_k^1\left(\mathbf{w}_k\right) - \sigma^2_{v,k} \right) - \gamma \lambda_k^2, \end{aligned} \tag{10}$$

where the last term is added as a correction term to avoid any spurious behavior. Using the Robbins-Munro algorithm [10], the adaptive solution now becomes

$$\mathbf{w}_{k,i} = \mathbf{w}_{k,i-1} - \mu_k \frac{\partial J_k'\left(\mathbf{w}_k\right)}{\partial \mathbf{w}_k}, \tag{11}$$

$$\lambda_{k,i} = \lambda_{k,i-1} + \beta \frac{\partial J_k'\left(\mathbf{w}_k\right)}{\partial \lambda_{k,i-1}}. \tag{12}$$

1) Steepest Descent Solution: Solving the first partial derivative gives

$$\begin{aligned} \frac{\partial J_k\left(\mathbf{w}_k\right)}{\partial \mathbf{w}_k} &= \left(1 + \gamma \lambda_k\right) \left(\mathbf{R}_{\mathbf{u},k} \mathbf{w}_k - \mathbf{R}_{d\mathbf{u},k} \right) \\ &+ \sum_{l \in N_k / \{k\}} b_{lk} \left(\mathbf{w}_k - \psi_l \right). \end{aligned} \tag{13}$$

Similarly, solving the 2nd partial derivative gives

$$\frac{\partial J_k'\left(\mathbf{w}_k\right)}{\partial \lambda_k} = \gamma \left(\mathrm{E}\left[\left| d_k - \mathbf{u}_k \mathbf{w} \right|^2 \right] - \sigma^2_{v,k} \right) - 2\gamma\lambda_k. \tag{14}$$

which leads to

$$\lambda_{k,i} = \lambda_{k,i-1} + \beta\gamma\left(\mathrm{E}\left[\left|d_k - \mathbf{u}_k\mathbf{w}\right|^2\right] - \sigma_{v,k}^2\right) - 2\beta\gamma\lambda_{k,i-1}. \tag{15}$$

Replacing $\beta\gamma$ by $\beta/2$ and incorporating the partial derivatives into the algorithm, we get the steepest descent solution

$$\begin{aligned} \mathbf{w}_{k,i} &= \mathbf{w}_{k,i-1} + \mu_k\left(1+\gamma\lambda_{k,i-1}\right)\left(\mathbf{R}_{\mathbf{u},k}\mathbf{w}_k - \mathbf{R}_{d\mathbf{u},k}\right), \\ &+ \upsilon_k \sum_{l\in N_k/\{k\}} b_{lk}\left(\psi_{l,i-1} - \mathbf{w}_{k,i-1}\right) \quad (16) \\ \lambda_{k,i} &= \left(1-\beta\right)\lambda_{k,i-1} \\ &+ \frac{\beta}{2}\left(\mathrm{E}\left[\left|d_k - \mathbf{u}_k\mathbf{w}\right|^2\right] - \sigma_{v,k}^2\right). \quad (17) \end{aligned}$$

The first equation can be broken into a two-step process, namely

$$\begin{aligned} \psi_{k,i} &= \mathbf{w}_{k,i-1} + \mu_k\left(1+\gamma\lambda_{k,i-1}\right) \\ &\quad .\left(\mathbf{R}_{\mathbf{u},k}\mathbf{w}_k - \mathbf{R}_{d\mathbf{u},k}\right), \quad (18) \\ \mathbf{w}_{k,i} &= \psi_{k,i} + \upsilon_k \sum_{l\in N_k/\{k\}} b_{lk}\left(\psi_{l,i-1} - \psi_{k,i-1}\right) \\ &= \psi_{k,i}\left(1 - \upsilon_k + b_{kk}\upsilon_k\right) + \upsilon_k \sum_{l\in N_k/\{k\}} b_{lk}\psi_{l,i-1} \\ &= \sum_{l\in N_k} c_{lk}\psi_{l,i-1}, \quad (19) \end{aligned}$$

where

$$c_{lk} = \begin{cases} 1 - \upsilon_k + \upsilon_k b_{kk}, & l = k \\ \upsilon_k b_{lk}, & l \neq k \end{cases}$$

Combining (17), (18) with (16) gives us the steepest descent solution to the noise-constrained problem.

2) Noise-Constrained Diffusion LMS: To get an adaptive solution we simply replace $\mathbf{R}_{\mathbf{u},k}$, $\mathbf{R}_{d\mathbf{u},k}$ and $\mathrm{E}\left[\left|d_k - \mathbf{u}_k\mathbf{w}\right|^2\right]$ by their instantaneous values. Noting that $e_k(i) = d_k(i) - \mathbf{u}_{k,i}\mathbf{w}_{k,i-1}$, we get

$$\begin{aligned} \psi_{k,i} &= \mathbf{w}_{k,i-1} + \mu_k\left(1+\gamma\lambda_{k,i-1}\right)\mathbf{u}_{k,i}^* e_k(i), \quad (20) \\ \mathbf{w}_{k,i} &= \sum_{l\in N_k} c_{lk}\psi_{l,i}, \quad (21) \\ \lambda_{k,i} &= \left(1-\beta\right)\lambda_{k,i-1} + \frac{\beta}{2}\left(e_{k,i}^2 - \sigma_{v,k}^2\right), \quad (22) \end{aligned}$$

where

$$c_{lk} = \begin{cases} 1 - \upsilon_k + \upsilon_k b_{kk}, & l = k \\ \upsilon_k b_{lk}, & l \neq k \end{cases}$$

So equations (19)-(21) form the Noise-Constrained Diffusion LMS (NCDLMS) algorithm using the *ATC* approach. In case of *CTA* the error is given by $e_k(i) = d_k(i) - \mathbf{u}_{k,i}\psi_{k,i-1}$ and the algorithm would become

$$\begin{aligned} \psi_{k,i-1} &= \sum_{l\in N_k} c_{lk}\mathbf{w}_{l,i-1} \quad (23) \\ \mathbf{w}_{k,i} &= \psi_{k,i-1} + \mu_k\left(1+\gamma\lambda_{k,i-1}\right)\mathbf{u}_{k,i}^* e_k(i) \quad (24) \\ \lambda_{k,i} &= \left(1-\beta\right)\lambda_{k,i-1} + \frac{\beta}{2}\left(e_{k,i}^2 - \sigma_{v,k}^2\right) \quad (25) \end{aligned}$$

IV. Mean Transient Analysis

Here we show briefly the analysis for the *ATC* scheme. Let

$$\begin{aligned} \mathbf{w}^{(o)} &\triangleq \mathbf{1}_N \otimes \mathbf{w}^o, \qquad \mathbf{d}_i \triangleq \mathrm{col}\left\{\mathrm{d}_1\left(\mathrm{i}\right), ..., \mathrm{d}_\mathrm{N}\left(\mathrm{i}\right)\right\}, \\ \mathbf{U}_i &\triangleq \mathrm{diag}\left\{\mathbf{u}_{1,\mathrm{i}}, ...\mathbf{u}_{\mathrm{N},\mathrm{i}}\right\}, \qquad \mathbf{v}_\mathrm{i} \triangleq \mathrm{col}\left\{\mathrm{v}_1\left(\mathrm{i}\right), ...\mathrm{v}_\mathrm{N}\left(\mathrm{i}\right)\right\}, \end{aligned}$$

where $\otimes$ is the Kronecker product and the $N \times 1$ vector $\mathbf{d}_i$ is given by $\mathbf{d}_i = \mathbf{U}_i\mathbf{w}^{(o)} + \mathbf{v}_i$. Defining $\mathbf{D} = \mathrm{diag}\left\{\mu_1\mathbf{I}_\mathrm{M}, ..., \mu_\mathrm{N}\mathbf{I}_\mathrm{M}\right\}$ and $\mathbf{L}_{i-1} = \mathrm{diag}\left\{(1+\gamma\lambda_{1,\mathrm{i}-1})\mathbf{I}_\mathrm{M}, ..., (1+\gamma\lambda_{\mathrm{N},\mathrm{i}-1})\mathbf{I}_\mathrm{M}\right\}$. Let the combiner weight-matrix be defined as $\mathbf{C} \triangleq [\mathbf{c}_1, ..., \mathbf{c}_N]$, then we have $\mathbf{G} \triangleq \mathbf{C}\otimes\mathbf{I}_M$. Now defining the weight-error vector as $\tilde{\mathbf{w}}_i \triangleq \mathbf{w}^o - \mathbf{w}_i$ and noting the fact that $\mathbf{w}^i \triangleq col\left\{\mathbf{w}_i, ..., \mathbf{w}_N\right\}$, we have

$$\tilde{\mathbf{w}}^i \triangleq \mathbf{w}^{(o)} - \mathbf{w}^i. \tag{26}$$

Rearranging the algorithm slightly we get

$$\begin{aligned} \tilde{\mathbf{w}}^i &= \mathbf{w}^{(o)} - \mathbf{w}^i \\ &= \mathbf{G}\left(\mathbf{w}^{(o)} - \psi^i\right) \\ &= \mathbf{G}\left[\mathbf{w}^{(o)} - \mathbf{w}^{i-1} - \mathbf{D}\mathbf{L}_{i-1}\mathbf{U}_i^*\left(\mathbf{d}_i - \mathbf{U}_i\mathbf{w}^{i-1}\right)\right] \\ &= \mathbf{G}\left[\tilde{\mathbf{w}}^{i-1} - \mathbf{D}\mathbf{L}_{i-1}\mathbf{U}_i^*\left(\mathbf{U}_i\tilde{\mathbf{w}}^{i-1} + \mathbf{v}_i\right)\right]. \quad (27) \end{aligned}$$

Taking expectation and solving gives

$$\mathrm{E}\left[\tilde{\mathbf{w}}^i\right] = \mathbf{G}\left(\mathbf{I}_{NM} - \mathbf{D}\mathrm{E}\left[\mathbf{L}_{i-1}\right]\mathbf{R}_U\right)\mathrm{E}\left[\tilde{\mathbf{w}}^{i-1}\right]. \tag{28}$$

Let $\mathrm{E}\left[\mathbf{B}_{i-1}\right] = \left(\mathbf{I}_{NM} - \mathbf{D}\mathrm{E}\left[\mathbf{L}_{i-1}\right]\mathbf{R}_U\right)$. We can see then that the stability of the system depends on $\bar{\lambda}\left|\mathbf{G}\mathrm{E}\left[\mathbf{B}_{i-1}\right]\right| < 1$ where $\bar{\lambda}$ defines the eigenvalues for $\left|\mathbf{G}\mathrm{E}\left[\mathbf{B}_{i-1}\right]\right|$. The 2-norm of a matrix is defined as the largest singular value of the matrix. Using the 2-norm we can express the product of $\mathbf{G}$ and $\mathrm{E}\left[\mathbf{B}_{\mathrm{i}-1}\right]$ as

$$\left\|\mathbf{G}\mathrm{E}\left[\mathbf{B}_{i-1}\right]\right\|_2 \leq \left\|\mathbf{G}\right\|_2 . \left\|\mathrm{E}\left[\mathbf{B}_{i-1}\right]\right\|_2 . \tag{29}$$

Since we already have $\mathbf{G} = \mathbf{C}\otimes\mathbf{I}_M$ and we know that $\mathbf{B}_{i-1}$ and $\mathbf{R}_U$ are Hermitian and block diagonal, the above equation reduces to

$$\left|\bar{\lambda}_{\max}\left(\mathbf{G}\mathrm{E}\left[\mathbf{B}_{i-1}\right]\right)\right| \leq \left\|\mathbf{C}\right\|_2 . \left|\bar{\lambda}_{\max}\left(\mathrm{E}\left[\mathbf{B}_{i-1}\right]\right)\right|. \tag{30}$$

A combiner rule is picked such that $\left\|\mathbf{C}\right\|_2 \leq 1$ so that the cooperative scheme is providing robustness over the non-cooperative scheme. For a symmetric matrix $\mathbf{C}$, we have $\left\|\mathbf{C}\right\|_2 = 1$ so that we get

$$\left|\bar{\lambda}_{\max}\left(\mathbf{G}\mathrm{E}\left[\mathbf{B}_{i-1}\right]\right)\right| \leq \left|\bar{\lambda}_{\max}\left(\mathrm{E}\left[\mathbf{B}_{i-1}\right]\right)\right|. \tag{31}$$

This means that the cooperative system will always be stable as long as the non-cooperative system is stable. For the non-cooperative system to be stable, the step-size should be such that

$$\left|\bar{\lambda}_{\max}\left(\mathrm{E}\left[\mathbf{B}_{i-1}\right]\right)\right| \leq 1.$$

Expanding and solving gives us the step-size range for node k as

$$0 < \mu_k \leq \frac{2}{\bar{\lambda}_{\max}\left(\mathrm{E}\left[1+\gamma\lambda_{k,\infty}\right].\mathbf{R}_{\mathbf{u},k}\right)}, \tag{32}$$

where $\lambda_{k,\infty}$ is the steady-state value for the Lagrangian and should ideally reduce to zero resulting in the familiar expression for the step-size

$$0 < \mu_k \leq \frac{2}{\bar{\lambda}_{\max}\left(\mathbf{R}_{\mathbf{u},k}\right)}. \tag{33}$$

V. Simulation Results

Simulation results are shown to illustrate the performance of the NCDLMS algorithm. Comparisons are shown with Diffusion LMS (DLMS) algorithm. Simulations have been done for a network containing $N = 15$ nodes connected as shown in Fig. 3. Fig. 4 shows the characteristics of the signals and noise variances at each node of the network. The unknown vector is of length $M = 5$ and is taken to be $\mathbf{w}^o = \mathbf{1}/\sqrt{5}$ where $\mathbf{1}$ is a column vector of length M containing all 1s. The input regressor vectors are zero-mean independent Gaussian random variables with variance for each node as per the SNR shown in Fig. 4 (top). The additive noise is also zero-mean Gaussian with variance as plotted in Fig. 4 (bottom). The NCDLMS algorithm is compared with DLMS for both the *CTA* as well as the *ATC* version. Also plotted is the case where no cooperation takes place between nodes, that is, simple LMS adaptation without combining. Step-sizes for the DLMS and no cooperation case are all set to $\mu_k = 0.01$. For the NCDLMS algorithm, the initial step-size is set at $\mu_k = 0.025$ whereas $\beta = 0.01$ and $\gamma = 141$. For all three diffusion cases, the Metropolis rule [4] is used for combining.

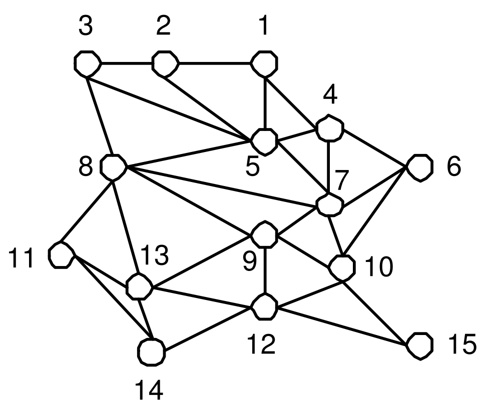

Fig. 3. Adaptive network of $N = 15$ nodes

Fig. 5 shows the Mean Square Deviation (MSD) and Excess Mean Square Error (EMSE) for all algorithms. Since the convergence rate is approximately the same, it can be seen that the difference is in the error floor. The *ATC* scheme slightly outperforms the *CTA* scheme, however, NCDLMS algorithm provides the best performance. Fig. 6 depicts how the step-size $\mu_k(1+\gamma\lambda_{k,i}$ varies with time for node 13. As expected, λ_k goes to zero for all nodes asymptotically resulting in a constant step-size. A similar trend is followed for the step-size at each node. Fig. 7 shows the steady-state MSD and EMSE values for each node and as can be seen from this figure great improvement in performance is obtained through the use of the proposed algorithm.

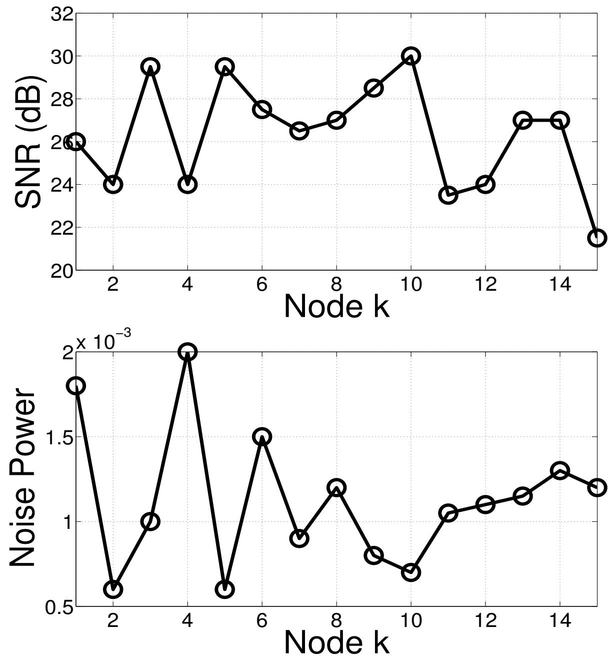

Fig. 4. SNR (top) and Noise-variance $\sigma^2_{v,k}$ (bottom) for $N = 15$ nodes

VI. Conclusion

In this paper we have derived a new algorithm for adaptive networks based on the constraint that the additive noise variance is known at all nodes. The design of the new algorithm is shown followed by its mean transient analysis. Simulation results are then included that show clearly the improvement achieved by the new NCDLMS algorithm over the 2 versions of the DLMS algorithm. Results show improvement over the entire network as well as at individual node level. Furthermore, this improvement is achieved at the cost of a small increase in complexity, making the NCDLMS algorithm a better alternative over the DLMS algorithms.

Acknowledgement

This research work is funded by King Fahd University of Petroleum & Minerals (KFUPM) under Research Grant (FT100012).

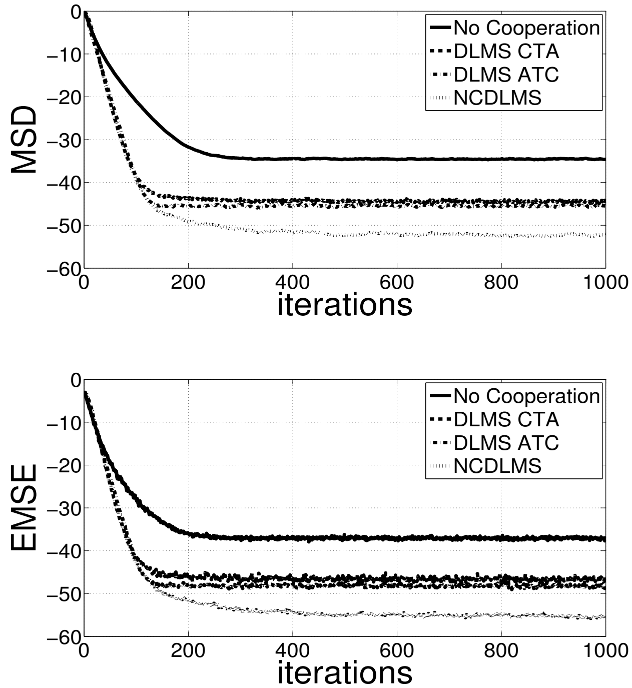

Fig. 5. Average MSD (top) and EMSE (bottom) for all 4 algorithms

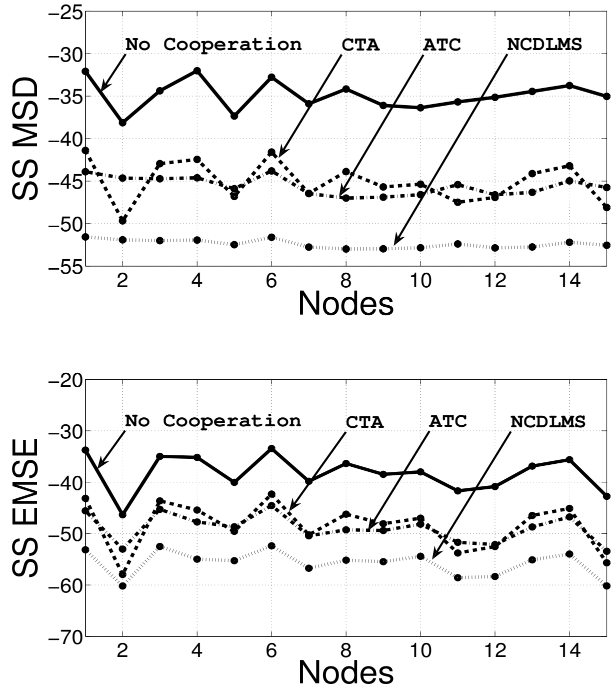

Fig. 7. Steady-state MSD (top) and EMSE (bottom) for all nodes

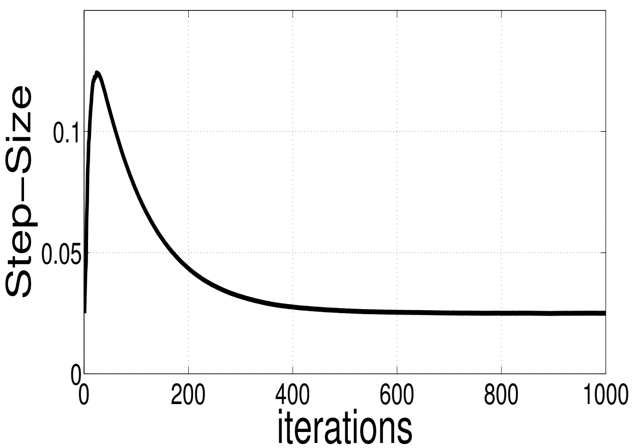

Fig. 6. Step-size evolution for Node 13

REFERENCES

[1] C.G. Lopes and A.H. Sayed, "Incremental adaptive strategies over distributed networks," *IEEE Trans. Signal Process.*, vol. 55, no. 8, pp. 4064-4077, Aug. 2007.

[2] 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.

[3] C.G. Lopes and A.H. Sayed, "Diffusion adaptive networks with changing topologies," in *Proc. IEEE ICASSP*, Las Vegas, NV, Apr. 2008, pp. 3285-3288.

[4] L.Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems and Control Letters*, vol. 53, no. 1, pp. 65-78, Sept. 2004.

[5] F.S. Cattivelli, C.G. Lopes and A.H. Sayed, "Diffusion recursive least-squares for distributed estimation over adaptive networks," *IEEE Trans. Signal Process.*, vol. 56, no. 5, pp. 1865-1877, May 2008.

[6] I.D. Schizas, A. Ribeiro, G.B. Giannakis, "Distributed LMS for consensus-based in-network adaptive processing," *IEEE Trans. Signal Process.*, vol. 57, no. 6, pp. 2365-2382, Jun. 2009.

[7] N. Takahashi, I. Yamada and A.H. Sayed, "Diffusion least-mean squares with adaptive combiners," in *Proc. ICASSP*, Taipei, Taiwan, April 2009, pp. 2845-2848.

[8] 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.

[9] Y. Wei, S.B. Gelfand and J.V. Krogmeier, "Noise-constrained least-mean squares algorithm," *IEEE Trans. Signal Process.*, vol. 49, no. 9, pp. 1961-1970, Sept. 2001.

[10] M. Dufflo, *Random Iterative Models*, Berlin, Germany: Springer-Verlag, 1997.

[11] D.P. Bertsekas, *Constrained Optimization and Lagrange Multipliers Methods*, Academic Press, New York, USA, 1982.