DISTRIBUTED ADAPTIVE INCREMENTAL STRATEGIES: FORMULATION AND PERFORMANCE ANALYSIS

Cassio G. Lopes and Ali H. Sayed

Department of Electrical Engineering
University of California Los Angeles, CA, 90095.
Email: {cassio, sayed@ee.ucla.edu}

ABSTRACT

An adaptive distributed estimation strategy is developed based on incremental gradient techniques. The proposed scheme addresses the problem of distributed linear estimation in a cooperative fashion, resulting in a distributed algorithm that can respond in real time to changes in the environment. Each node is allowed to communicate only with its immediate neighbor in order to exploit the spatial dimension while at the same time reducing the communications burden. A spatial-temporal energy conservation argument is used to evaluate the steady-state mean-square-error performance of the individual nodes across the adaptive distributed network. Computer simulations illustrate the results.

1. INTRODUCTION

Distributed networks linking PCs, laptops, cell phones, sensors and actuators will form the backbone of future data, communication, and control networks. Applications will range from sensor networks to precision agriculture, environment monitoring, disaster relief management, smart spaces, target localization, as well as medical applications [1, 2, 3]. In all these cases, the distribution of the nodes in the field yields spatial diversity, which should be exploited alongside the temporal dimension in order to enhance the robustness of the processing tasks and improve the probability of signal and event detection. Collaborative signal processing has been advocated as a way to achieve the efficient fusion of information. Regardless of the cooperative technique adopted, it is an accepted fact nowadays that distributed processing will need to be both *adaptive* and *cooperative*. This is because not only the environmental conditions vary with time and space, but the network topology may vary.

Motivated by incremental gradient ideas [4, 5], this paper develops an adaptive distributed processing algorithm. It is a fully distributed and cooperative scheme that can respond in real time to changes in the environment. Moreover, the scheme balances the tradeoff between cooperation and communications by sharing the computational burden among the nodes while decreasing the amount of communications that would be necessary in comparison to centralized solutions. We illustrate the adaptive procedure by using an LMS-type update, which eliminates the need to embed powerful processors at the nodes. More sophisticated adaptation rules could be used as well. We also develop a spatial-temporal energy conservation argument to evaluate the steady-state mean-square error performance of the individual nodes across the adaptive distributed network, and illustrate the results via computer simulations.

It may be noted that most distributed techniques in the literature tend to be iterative in nature as opposed to adaptive. In an adaptive distributed implementation, the individual nodes can process the data in real-time and the entire network can be viewed as an adaptive entity in its own right. The state of the entire network would change with each measurement and these changes would get propagated throughout the nodes with time.

This material was based on work supported in part by the National Science Foundation under awards CCR-0208573 and ECS-0401188. The work of the first author was also supported by a fellowship from CAPES, Brazil, under award 1168/01-0.

2. THE ESTIMATION PROBLEM AND THE ADAPTIVE SOLUTION

We are interested in estimating an unknown vector w^o from multiple *spatially independent* but possibly *time-correlated* measurements collected at N nodes in a network (see Fig. 1). Each node k has access to realizations of zero-mean data $\{\boldsymbol{d}_k, \boldsymbol{u}_k\}$, $k = 1, \ldots, N$, where $\boldsymbol{d}_k$ is a scalar and $\boldsymbol{u}_k$ is $1 \times M$. We collect the regression and measurement data into global matrices:

$$\boldsymbol{U} \triangleq \mathrm{col}\{\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_N\} \qquad (N \times M) \qquad (1)$$

$$\boldsymbol{d} \triangleq \mathrm{col}\{\boldsymbol{d}_1, \boldsymbol{d}_2, \ldots, \boldsymbol{d}_N\} \qquad (N \times 1) \qquad (2)$$

and pose the *minimum mean-square* error estimation problem:

$$\min_w J(w), \quad \text{where} \quad J(w) = E\|\boldsymbol{d} - \boldsymbol{U}w\|^2 \qquad (3)$$

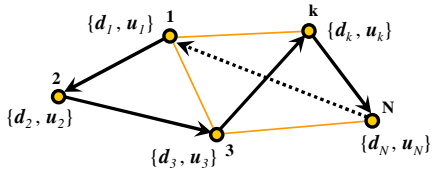

Fig. 1. A distributed network with N nodes and the incremental algorithm path.

The optimal solution w^o satisfies the *normal equations* [6]:

$$R_{du} = R_u w^o \qquad (4)$$

where

$$R_u = E\,\boldsymbol{U}^*\boldsymbol{U} = \sum_{k=1}^{N} R_{u,k} \quad , \quad R_{du} = E\,\boldsymbol{U}^*\boldsymbol{d} = \sum_{k=1}^{N} R_{du,k} \qquad (5)$$

Computing w^o from (4) would require *every node* to have access to the global statistical information $\{R_u, R_{du}\}$, thus draining communications and computational resources. We seek a *distributed* solution that allows cooperation among nodes through limited local communications, while equipping the nodes with adaptive mechanisms to respond to time-variations in the underlying signal statistics.

We start from the standard gradient-descent implementation

$$w_i = w_{i-1} - \mu\ [\nabla J(w_{i-1})]^* \qquad (6)$$

for solving the normal equations (4), where $\mu > 0$ is a suitably chosen step-size parameter, w_i is an estimate for w^o at iteration i, and $\nabla J(\cdot)$ denotes the gradient vector of $J(w)$ evaluated at w_{i-1}. For μ sufficiently small we will have $w_i \to w^o$ as $i \to \infty$. This iterative solution could be applied at every node k or centrally at some central node. A distributed version can be motivated as follows.

The cost function $J(w)$ can be decomposed as:

$$J(w) = \sum_{k=1}^{N} J_k(w), \text{ where } \quad J_k(w) \triangleq E|\boldsymbol{d}_k - \boldsymbol{u}_k w|^2 \qquad (7)$$

which allows us to rewrite (6) as

$$w_i = w_{i-1} - \mu \left[\sum_{k=1}^{N} \nabla J_k(w_{i-1})\right]^* \qquad (8)$$

Now let $\psi_k^{(i)}$ be a *local estimate* of w^o at node k and time i and assign the initial condition $\psi_0^{(i)} \leftarrow w_{i-1}$. Then w_i can be evaluated by iterating $\psi_0^{(i)}$ through the nodes in the following manner:

$$\psi_k^{(i)} = \psi_{k-1}^{(i)} - \mu\,[\nabla J_k(w_{i-1})]^* \quad , \quad k = 1, \ldots, N \qquad (9)$$

At the end of the procedure (9), the last node will contain the global estimate w_i from (8), i.e., $w_i \leftarrow \psi_N^{(i)}$. This scheme still requires all nodes to share the global information w_{i-1}. A *fully* distributed solution can be achieved by resorting to incremental strategies, which would require each node in (9) to evaluate its partial gradient $\nabla J_k(\cdot)$ at its local estimate $\psi_{k-1}^{(i)}$, instead of w_{i-1}. This approach leads to the incremental algorithm:

$$\psi_k^{(i)} = \psi_{k-1}^{(i)} - \mu\,[\,\nabla J_k(\psi_{k-1}^{(i)})\,]^* \quad , \quad k = 1, \ldots, N \qquad (10)$$

This cooperative scheme requires each node k to communicate *only* with its immediate neighbor $k-1$ over a pre-defined path. Moreover, it is an established result in optimization theory that the incremental solution (10) can outperform the centralized solution (9) as illustrated in Fig. 2. The figure compares the excess mean square error (EMSE) of both algorithms for a network with $N = 20$ nodes and using Gaussian regressors with $R_{u,k} = I$. The background noise is white and Gaussian with $\sigma_v^2 = 10^{-3}$. The curves are obtained by averaging over 500 experiments with $\mu = 0.05$.

Now using instantaneous approximations $\hat{R}_{du,k} = d_k(i)u_{k,i}^*$ and $\hat{R}_{u,k} = u_{k,i}^* u_{k,i}$ in (10), and allowing for different step-sizes

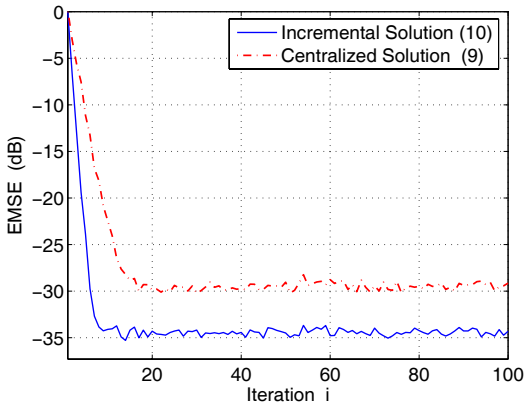

Fig. 2. Excess mean square error (EMSE) performance for both the distributed incremental solution (10) and the centralized solution (9) at node 1.

at different nodes, leads to a *distributed incremental* LMS algorithm, summarized below:

$$\begin{cases} \psi_0^{(i)} \leftarrow w_{i-1} \\ \psi_k^{(i)} = \psi_{k-1}^{(i)} + \mu_k u_{k,i}^* \left(d_k(i) - u_{k,i}\psi_{k-1}^{(i)}\right) \\ w_i \leftarrow \psi_N^{(i)} \end{cases} \qquad (11)$$

with $k = 1, \ldots, N$. In this algorithm, a weight estimate is circulated through a path defined over the network and updated by local adaptive filters using local data solely – see Fig. 3.

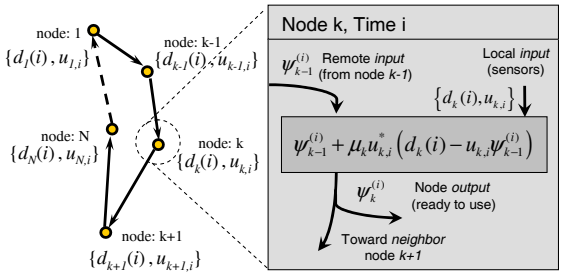

Fig. 3. The structure of the adaptive distributed algorithm.

3. ANALYSIS FRAMEWORK

Algorithm (11) exploits both the spatial and temporal dimensions of the data. In order to study its performance, we shall extend the energy conservation approach of [6] to treat the space-time case. Due to space constraints, only the main steps are presented.

3.1. Data Model and Assumptions

The subsequent analysis relies on the following data assumptions for the random variables $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$:

1. The unknown vector w^o relates $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$ as

$$\boldsymbol{d}_k(i) = \boldsymbol{u}_{k,i} w^o + \boldsymbol{v}_k(i) \qquad (12)$$

where $\boldsymbol{v}_k(i)$ is some white noise sequence with variance $\sigma_{v,k}^2$ and independent of $\{\boldsymbol{d}_l(j), \boldsymbol{u}_{l,j}\}$ for all l, j.

2. $\boldsymbol{u}_{k,i}$ is independent of $\boldsymbol{u}_{l,i}$ for $k \neq l$ (spatial independence).
3. For every k, the sequence $\{\boldsymbol{u}_{k,i}\}$ is independent over time (time independence).

3.2. Weighted Energy Conservation Relation

For the algorithm (11), we define the error signals:

$$\widetilde{\boldsymbol{\psi}}_{k-1}^{(i)} \triangleq w^o - \boldsymbol{\psi}_{k-1}^{(i)}, \qquad \widetilde{\boldsymbol{\psi}}_{k}^{(i)} \triangleq w^o - \boldsymbol{\psi}_{k}^{(i)} \tag{13}$$

$$\boldsymbol{e}_{a,k}(i) \triangleq \boldsymbol{u}_{k,i}\widetilde{\boldsymbol{\psi}}_{k-1}^{(i)}, \qquad \boldsymbol{e}_{p,k}(i) \triangleq \boldsymbol{u}_{k,i}\widetilde{\boldsymbol{\psi}}_{k}^{(i)} \tag{14}$$

$$\boldsymbol{e}_k(i) \triangleq \boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}\boldsymbol{\psi}_{k-1}^{(i)} \tag{15}$$

where (13) are the weight-error vectors, (14) defines the a priori and a posteriori local errors, and (15) is the output error. Note that

$$\boldsymbol{e}_k(i) = \boldsymbol{e}_{a,k}(i) + \boldsymbol{v}_k(i) \tag{16}$$

We are interested in evaluating in steady-state and for each node k, the mean-square deviation (MSD), the excess mean-square error (EMSE), and the mean-square error (MSE). These quantities are defined as:

$$\eta_k \triangleq E\|\widetilde{\boldsymbol{\psi}}_{k-1}^{(\infty)}\|^2 = E\|\widetilde{\boldsymbol{\psi}}_{k-1}^{(\infty)}\|_I^2 \quad \text{(MSD)} \tag{17}$$

$$\zeta_k \triangleq E|\boldsymbol{e}_{a,k}(\infty)|^2 = E\|\widetilde{\boldsymbol{\psi}}_{k-1}^{(\infty)}\|_{R_{u,k}}^2 \quad \text{(EMSE)} \tag{18}$$

$$\xi_k \triangleq E|\boldsymbol{e}_k(\infty)|^2 = \zeta_k + \sigma_{v,k}^2 \quad \text{(MSE)} \tag{19}$$

In (17) and (18), a *weighted norm* notation is introduced. For a vector x and a Hermitian positive-definite matrix Σ, $\|x\|_\Sigma^2 = x^*\Sigma\, x$. Introduce further the *weighted a priori* and *a posteriori local error* signals for node k:

$$\boldsymbol{e}_{a,k}^{\Sigma}(i) \triangleq \boldsymbol{u}_{k,i}\Sigma\widetilde{\boldsymbol{\psi}}_{k-1}^{(i)} \quad \text{and} \quad \boldsymbol{e}_{p,k}^{\Sigma}(i) \triangleq \boldsymbol{u}_{k,i}\Sigma\widetilde{\boldsymbol{\psi}}_{k}^{(i)} \tag{20}$$

for some Hermitian positive-definite matrix Σ (at our choice). Different choices of Σ enable us to evaluate different performance measures. Following [6, 7], a *space-time energy relation* that relates the *local* errors quantities

$$\left\{\widetilde{\boldsymbol{\psi}}_{k-1}^{(i)}, \widetilde{\boldsymbol{\psi}}_{k}^{(i)}, \boldsymbol{e}_{a,k}^{\Sigma}(i), \boldsymbol{e}_{p,k}^{\Sigma}(i)\right\} \tag{21}$$

can be found to be

$$\|\widetilde{\boldsymbol{\psi}}_{k}\|_\Sigma^2 + \frac{|\boldsymbol{e}_{a,k}^{\Sigma}|^2}{\|\boldsymbol{u}_k\|_\Sigma^2} = \|\widetilde{\boldsymbol{\psi}}_{k-1}\|_\Sigma^2 + \frac{|\boldsymbol{e}_{p,k}^{\Sigma}|^2}{\|\boldsymbol{u}_k\|_\Sigma^2} \tag{22}$$

Equation (22) is a *space-time* version of the weighted energy conservation relation used in [6] in the context of regular adaptive implementations. The time index i has been dropped for compactness of notation.

3.3. Variance Relation

Substituting (20) into (22), expanding and taking expectations – under the independence assumptions on the regression data – gives:

$$E\|\widetilde{\boldsymbol{\psi}}_{k}\|_\Sigma^2 = E\|\widetilde{\boldsymbol{\psi}}_{k-1}\|_{\Sigma'}^2 + \mu_k^2\sigma_{v,k}^2 E\|\boldsymbol{u}_k\|_\Sigma^2 \tag{23}$$

$$\Sigma' = \Sigma - \mu_k E\left(\boldsymbol{u}_k^*\boldsymbol{u}_k\Sigma + \Sigma\,\boldsymbol{u}_k^*\boldsymbol{u}_k\right) + \mu_k^2 E\|\boldsymbol{u}_k\|_\Sigma^2\,\boldsymbol{u}_k^*\boldsymbol{u}_k \tag{24}$$

Recursion (23) is a *variance relation* that can be used to infer the *steady-state* performance at every node k. Note that Σ' is solely regressor-dependent and, therefore, decoupled from the weight-error vector. For simplicity, in this work we assume that the regressors $\{\boldsymbol{u}_k\}$ arise from a source with circular Gaussian distribution with covariance matrix $R_{u,k}$. Define the transformed quantities

$$\overline{\boldsymbol{\psi}}_k = U_k^*\widetilde{\boldsymbol{\psi}}_k,\ \ \overline{\boldsymbol{\psi}}_{k-1} = U_k^*\widetilde{\boldsymbol{\psi}}_{k-1},\ \ \overline{\boldsymbol{u}}_k = \boldsymbol{u}_k U_k,\ \ \overline{\Sigma} = U_k^*\Sigma U_k$$

through the eigen-decomposition $R_{u,k} = U_k\Lambda_k U_k^*$, where U_k is unitary and Λ_k is a diagonal matrix with the eigenvalues of $R_{u,k}$. Invoking known results for Gaussian signals [6], we can show that (23) and (24) become:

$$E\|\overline{\boldsymbol{\psi}}_k\|_{\overline{\Sigma}}^2 = E\|\overline{\boldsymbol{\psi}}_{k-1}\|_{\overline{\Sigma}'}^2 + \mu_k^2\sigma_{v,k}^2\,\mathrm{Tr}\left(\Lambda_k\overline{\Sigma}\right) \tag{25}$$

$$\overline{\Sigma}' = \left(I - 2\mu_k\Lambda_k + \gamma\mu_k^2\Lambda_k^2\right)\overline{\Sigma} + \mu_k^2\Lambda_k\overline{\Sigma}\Lambda_k \tag{26}$$

where $\gamma = 1$ for *complex signals* and $\gamma = 2$ for *real signals*.

3.4. Diagonal Notation

Note from (26) that choosing $\overline{\Sigma}$ to be diagonal, $\overline{\Sigma}'$ will be diagonal as well, suggesting a more compact notation. Collect the diagonal matrices $\overline{\Sigma}$ and $\overline{\Sigma}'$ into column vectors[1]:

$$\overline{\sigma} \triangleq \mathrm{diag}\{\overline{\Sigma}\}\,, \quad \overline{\sigma}' \triangleq \mathrm{diag}\{\overline{\Sigma}'\}\,, \quad \lambda_k \triangleq \mathrm{diag}\{\Lambda_k\}$$

We can then rewrite (26) in terms of $\{\overline{\sigma}, \lambda_k\}$ as:

$$\overline{\sigma}' = \left(I - 2\mu_k\Lambda_k + \gamma\mu_k^2\Lambda_k^2\right)\overline{\sigma} + \mu_k^2(\lambda_k^T\overline{\sigma})\lambda_k \triangleq \overline{F}_k\overline{\sigma}$$

where $\overline{F}_k = I - 2\mu_k\Lambda_k + \gamma\mu_k^2\Lambda_k^2 + \mu_k^2\lambda_k\lambda_k^T$. Moreover, expression (25) becomes

$$E\|\overline{\boldsymbol{\psi}}_k\|_{\overline{\sigma}}^2 = E\|\overline{\boldsymbol{\psi}}_{k-1}\|_{\overline{F}_k\overline{\sigma}}^2 + \mu_k^2\sigma_{v,k}^2(\lambda_k^T\overline{\sigma}) \tag{27}$$

where the $\mathrm{diag}\{\}$ operator has been dropped from the weights for compactness of notation.

3.5. Steady-State Behavior

Unlike the standard case [6], here the weight error vectors converge to a spatial error profile, stabilizing at individual error energy levels, i.e.:

$$E\|\widetilde{\boldsymbol{\psi}}_k^{(i)}\|^2 \to c_k\,, \quad as\ \ i \to \infty$$

with a value c_k that is possibly different for each node k. Let then $\boldsymbol{p}_k = \boldsymbol{\psi}_k^{(\infty)}$ and $\overline{\boldsymbol{p}}_k = \overline{\boldsymbol{\psi}}_k^{(\infty)}$ be the weight-error vector in steady-state and its transformed version, respectively. Also define the *row* vector $g_k = \mu_k^2\sigma_{v,k}^2\lambda_k^T$. Then (27) gives as $i \to \infty$

$$E\|\overline{\boldsymbol{p}}_k\|_{\overline{\sigma}_k}^2 = E\|\overline{\boldsymbol{p}}_{k-1}\|_{\overline{F}_k\overline{\sigma}_k}^2 + g_k\overline{\sigma}_k\,, \quad k = 1, \ldots, N \tag{28}$$

where we are choosing a different weighting $\overline{\sigma}_k$ for each node k. Choosing $\overline{\sigma}_{k-2} = \overline{F}_{k-1}\overline{\sigma}_{k-1}$ in (28) leads to:

$$E\|\overline{\boldsymbol{p}}_{k-2}\|_{\overline{F}_{k-1}\overline{\sigma}_{k-1}}^2 = E\|\overline{\boldsymbol{p}}_{k-3}\|_{\overline{F}_{k-2}\overline{F}_{k-1}\overline{\sigma}_{k-1}}^2 + g_{k-2}\overline{F}_{k-1}\overline{\sigma}_{k-1} \tag{29}$$

[1] We use the notation $\Lambda = \mathrm{diag}\{\lambda\}$ to denote a diagonal matrix formed from the entries of the vector λ, while $\lambda = \mathrm{diag}\{\Lambda\}$ denotes a vector retrieved from the main diagonal of Λ.

Now, substituting (29) into (28) gives:

$$E\|\overline{\boldsymbol{p}}_{k-1}\|^2_{\overline{\sigma}_{k-1}} = E\|\overline{\boldsymbol{p}}_{k-3}\|^2_{\overline{F}_{k-2}\overline{F}_{k-1}\overline{\sigma}_{k-1}} + g_{k-2}\overline{F}_{k-1}\overline{\sigma}_{k-1} + g_{k-1}\overline{\sigma}_{k-1} \quad (30)$$

The procedure can be repeated until all N equations in (28) are used. It can then be verified that (30) can be written as

$$E\|\overline{\boldsymbol{p}}_{k-1}\|^2_{(I-\Pi_{k,1})\overline{\sigma}_{k-1}} = a_k\overline{\sigma}_{k-1} \quad (31)$$

where

$$\Pi_{k,l} \triangleq \overline{F}_{k+l-1}\overline{F}_{k+l}\cdots\overline{F}_N\overline{F}_1\cdots\overline{F}_{k-1}\ , \quad l = 1,\ldots,N$$

$$a_k \triangleq g_k\Pi_{k,2} + g_{k+1}\Pi_{k,3} + \cdots + g_{k-2}\Pi_{k,N} + g_{k-1}$$

With different choices of the weighting vector $\overline{\sigma}_{k-1}$ in (31) we are now able to calculate the MSD, EMSE and MSE for each node. For the MSD, we select $\overline{\sigma}_{\eta,k-1} = (I-\Pi_{k,1})^{-1}\,q$ where $q = \mathrm{col}\{1,1,\cdots,1\}$. Then

$$\eta_k = E\|\overline{p}_{k-1}\|^2_q = a_k\,(I-\Pi_{k,1})^{-1}\,q \quad (32)$$

Likewise, to determine the EMSE for node k we choose $\overline{\sigma}_{\zeta,k-1} = (I-\Pi_{k,1})^{-1}\,\lambda_k$, so that

$$\zeta_k = E\|\overline{p}_{k-1}\|^2_{\lambda_k} = a_k\,(I-\Pi_{k,1})^{-1}\,\lambda_k \quad (33)$$

The MSE follows from (19) and the result above.

4. SIMULATIONS

We consider a network with $N = 16$ nodes where each local filter has $M = 10$ taps. The system evolves for 50000 iterations and the results are averaged over 100 independent experiments. The steady-state values are obtained by averaging the last 5000 time samples. Each node accesses time-correlated spatially independent Gaussian regressors $\boldsymbol{u}_{k,i}$ with correlation functions $r_k(i) = \sigma^2_{u,k}\cdot(\alpha_k)^{|i|}$, $i = 0,\ldots,M-1$, with $\{\alpha_k\}$ and $\{\sigma^2_{u,k}\}$ randomly chosen in $[0,1)$ and depicted in Fig. 4. The background noise $\boldsymbol{v}_k(i)$ has variance $\sigma^2_{v,k} = 10^{-3}$ across the network. The MSE curves show a good match between theory and practice.

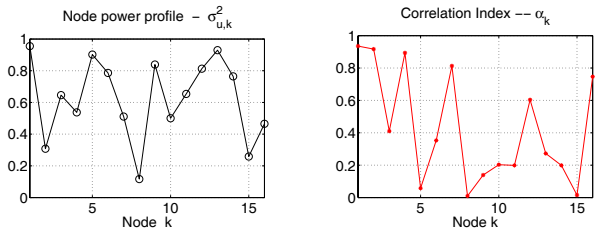

Fig. 4. Regressor (left) and noise (right) profile per node.

5. CONCLUSIONS

The inherent cooperative strategy of the proposed scheme not only improves performance, but it also decreases the amount of communications needed to implement cooperation among the nodes. Energy conservation arguments have been used to study the steady-state performance of the individual nodes in the Gaussian case.

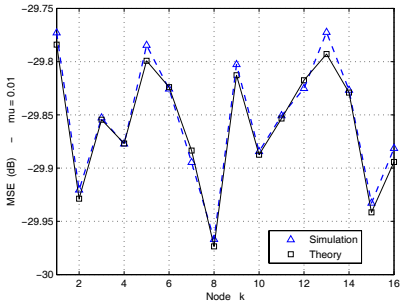

Fig. 5. MSE versus node.

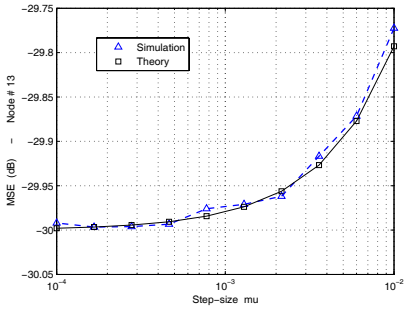

Fig. 6. MSE versus μ for node 13.

More general data distribution, and also more sophisticated cooperative schemes with each node cooperating with a subset of nearby nodes, are useful extensions and will be studied in future work.

6. REFERENCES

[1] D. Estrin, G. Pottie and M. Srivastava. "Intrumenting the world with wireless sensor networks," *Proc. IEEE ICASSP*, pp. 2033-2036, Salt Lake City, UT, May 2001.

[2] D. Li, K. D. Wong, Y. H. Hu and A. M. Sayeed. "Detection, classification, and tracking of targets," *IEEE Signal Processing Magazine*, vol. 19, pp. 17-29, March 2002.

[3] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, pp. 102-114, August 2002.

[4] D. Bertsekas. "A new class of incremental gradient methods for least squares," Problems *SIAM J. Optim.*, vol.7, no. 4, pp. 913-926, November 1997.

[5] M. G. Rabbat and R. D. Nowak. "Quantized incremental algorithms for distributed optimization," *IEEE Journal on Selected Areas in Communications*, vol.23, No.4, pp. 798-808, April 2005.

[6] A. H. Sayed. *Fundamentals of Adaptive Filtering*, Wiley, NJ, 2003.

[7] T. Y. Al-Naffouri and A. H. Sayed. "Transient analysis of data-normalized adaptive filters," *IEEE Transactions on Signal Processing*, vol. 51, no. 3, pp. 639–652, March 2003.