DISTRIBUTED PROCESSING OVER ADAPTIVE NETWORKS

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

Distributed adaptive algorithms are proposed based on incremental and diffusion strategies. Adaptation rules that are suitable for ring topologies and general topologies are described. Both distributed LMS and RLS implementations are considered in order to endow a network of nodes with learning abilities; thus resulting in a network that is an adaptive entity in its own right.

1. INTRODUCTION

Distributed and sensor networks are emerging as a formidable technology for a variety of applications, ranging from precision agriculture, to environment surveillance, to target localization. However, the advantages advocated by the use of distributed and cooperative processing [1] demand adaptive processing capabilities at the distributed nodes in order to be able to cope with time variations in the environment and the network. In addition, the adaptive processors should enable the network to respond to such variations in real-time and to adjust its performance accordingly. Inspired by incremental strategies [2], we propose distributed processing strategies over what we refer to as adaptive networks (e.g., [3]). The proposed strategies require the adaptive nodes to share information only locally and to exploit both spatial and temporal information in a cooperative fashion. Different cooperation policies will lead to different distributed algorithms.

Each node k across an N-node network is assumed to have access to time realizations $\{d_k(i), u_{k,i}\}$, of zero-mean random data $\{\boldsymbol{d}_k, \boldsymbol{u}_k\}$, with $d_k(i)$ a scalar measurement and $u_{k,i}$ a regression row vector; both at time i – see Figure 1. The nodes should use the data to estimate some unknown common vector w^o. Rather than expect each node to function independently of the other nodes, the nodes will instead collaborate with each other in an adaptive manner in order to achieve three objectives: (1) improve global performance with reduced communication; (2) allow the nodes to converge to the desired estimate without the need to share global information; (3) endow the network with learning abilities.

Recently the authors proposed a scheme [3] that implements a distributed incremental gradient algorithm in which an initial vector estimate is updated along a collaboration cycle over the network. Each local filter updates the estimate received from the previous neighbor $k-1$ with its local data and passes the estimate to the next node $k+1$, operating over a collaboration cycle - see Figure 3 further ahead. This approach requires limited communications and increases the network autonomy [3]. The network may also learn at an enhanced pace as compared to a standard gradient-descent solution.

This material was based on work supported in part by the National Science Foundation under awards ECS-0401188 and ECS-0601266. The work of Mr. Lopes was also supported by a fellowship from CAPES, Brazil, under award 1168/01-0.

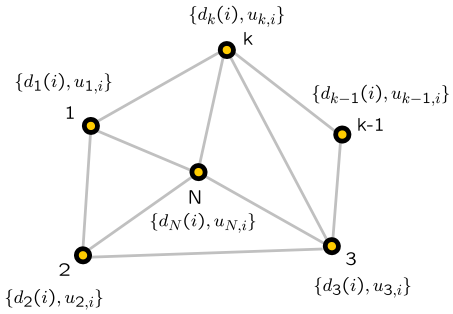

Fig. 1. A distributed network with N nodes.

In this paper, we propose a least-squares framework that equips the nodes with a RLS-type adaptation rule, while keeping the same cooperation strategy, yielding to a distributed and recursive least-squares solution (dRLS). The resulting algorithm conveys an exact global least-squares estimate, for the unknown vector w^o, to every node in the network. This scheme further allows an alternative dRLS implementation with decreased communication requirements, saving energy compared with its exact counterpart. It also has the striking feature that in steady-state, both algorithms present similar performance in the mean-square error sense.

When more communication and energy resources are available, the topology constraints implied by the aforementioned algorithms can be removed by resorting to a diffusion cooperative scheme, where the adaptive processor at node k updates its estimate using all available estimates from the neighbors, as well as local data and its own past estimate – see Figure 6.

2. DISTRIBUTED ESTIMATION

We are interested in estimating an unknown vector w^o from 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:

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

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

and pose the minimum mean-square error estimation problem:

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

The optimal solution w^o satisfies the *normal equations* [4]:

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

where

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

Computing w^o from (4) requires *every node* to have access to the global statistical information $\{R_u, R_{du}\}$, thus draining communications and computational resources. In [3] we proposed a *distributed* solution (incremental LMS) 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.

3. INCREMENTAL LMS ADAPTATION

In this section, we review the distributed incremental LMS algorithm [3], which is a starting point for the later developments. The algorithm is obtained as follows. 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:

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

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:

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

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:

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

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 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 $\mu = 0.05$.

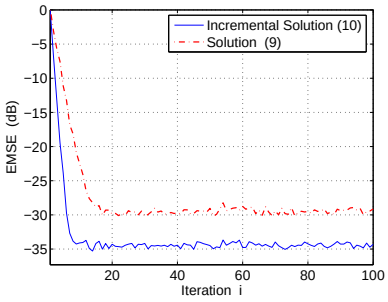

Fig. 2. Excess mean square error (EMSE) performance for both the distributed incremental solution (10) and the centralized solution (9) at node 1.

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 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 – see Fig. 3.

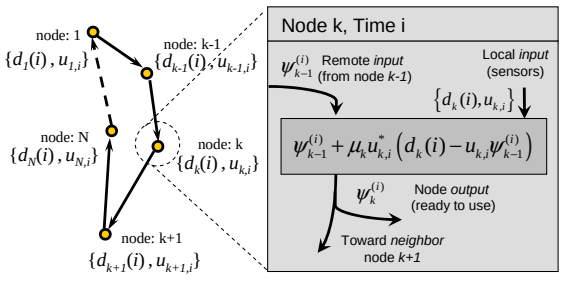

Fig. 3. The structure of incremental LMS.

4. EXACT DISTRIBUTED RLS ADAPTATION

We formulate in this section a least-squares solution for estimating the unknown parameter vector w^o. At each time instant i, the network has access to the following space-time data:

$$y_i = \begin{bmatrix} d_1(i) \\ d_2(i) \\ \vdots \\ d_N(i) \end{bmatrix} \quad \text{and} \quad H_i = \begin{bmatrix} u_{1,i} \\ u_{2,i} \\ \vdots \\ u_{N,i} \end{bmatrix}. \tag{12}$$

We can then seek an estimate for w^o by solving a regularized least-squares problem of the form [4]:

$$\min_w \left[w^* \Pi w + \left\| \mathcal{Y}_i - \mathcal{H}_i w \right\|^2 \right] \tag{13}$$

where $\Pi > 0$ is a regularization matrix and $\mathcal{Y}_i$ and $\mathcal{H}_i$ collect all the data that are available up to time i:

$$\mathcal{Y}_i = \begin{bmatrix} \frac{y_0}{y_1} \\ \hline \vdots \\ \hline y_i \end{bmatrix} \quad \text{and} \quad \mathcal{H}_i = \begin{bmatrix} \frac{H_0}{H_1} \\ \hline \vdots \\ \hline H_i \end{bmatrix}. \tag{14}$$

One could also incorporate weighting into (13) to account for spatial relevance, temporal relevance, and node relevance. Here we continue without weighting in order to convey the main idea. We are thus interested in deriving a distributed implementation of the least-squares solution. Some related work has been recently proposed where a global least-squares solution is achieved only approximately at each node, and the algorithm demands large communication and energy resources [5]. We proceed instead as follows. Assume that w_{i-1} is the solution to the following least-squares (LS) problem using the data that are available up to time $i-1$:

$$\min_w \left[w^* \Pi w + \left\| \mathcal{Y}_{i-1} - \mathcal{H}_{i-1} w \right\|^2 \right]. \tag{15}$$

We are interested in updating w_{i-1} to w_i by accounting for the incoming data blocks y_i and H_i at time i. An algorithm that updates w_{i-1} incrementally is given by:

$$\begin{array}{l} \psi_0^{(i)} \leftarrow w_{i-1}; \quad P_{0,i} \leftarrow P_{i-1} \\ \text{for } k = 1 : N \\ \quad e_k(i) = d_k(i) - u_{k,i} \psi_{k-1}^{(i)} \\ \quad \psi_k^{(i)} = \psi_{k-1}^{(i)} + \frac{P_{k-1,i}}{1 + u_{k,i} P_{k-1,i} u_{k,i}^*} u_{k,i}^* e_k(i) \\ \quad P_{k,i} = P_{k-1,i} - \frac{P_{k-1,i} u_{k,i}^* u_{k,i} P_{k-1,i}}{1 + u_{k,i} P_{k-1,i} u_{k,i}^*} \\ \text{end} \\ w_i \leftarrow \psi_N^{(i)}; \quad P_i \leftarrow P_{N,i}. \end{array} \tag{16}$$

Similarly to the incremental LMS in Section 3, algorithm (16) induces a cycle across the network, along which the estimate w_{i-1} is spatially updated by sequentially visiting every node once. At each node k, the estimate $\psi_k^{(i)}$ at time i is the LS solution that is based on the data blocks $\mathcal{Y}_{i-1}$ and $\mathcal{H}_{i-1}$ and the data collected along the path up to that node, namely

$$\min_\psi \left[\psi^* \Pi \psi + \left\| \mathcal{Y}_i^k - \mathcal{H}_i^k \psi \right\|^2 \right] \quad \Longrightarrow \quad \psi_k^{(i)} \tag{17}$$

where now

$$\mathcal{Y}_i^k = \begin{bmatrix} \frac{\mathcal{Y}_{i-1}}{d_1(i)} \\ d_2(i) \\ \vdots \\ d_k(i) \end{bmatrix} \quad \text{and} \quad \mathcal{H}_i^k = \begin{bmatrix} \frac{\mathcal{H}_{i-1}}{u_{1,i}} \\ u_{2,i} \\ \vdots \\ u_{k,i} \end{bmatrix}. \tag{18}$$

At the end of the cycle, $\psi_N^{(i)}$ will contain the desired solution w_i. If we start from $i = 0$ with $w_{-1} = 0$ and $P_{-1} = \Pi^{-1}$ and repeatedly apply (16), then $\psi_N^{(i)}$ will be the solution to (13). The distributed RLS (dRLS) algorithm (16) can be motivated as follows. Note first that the solution for (17) is given by [4]:

$$\psi_k^{(i)} = P_{k,i} \mathcal{H}_i^{k*} \mathcal{Y}_i^k \tag{19}$$

with

$$P_{k,i} = \left(\Pi + \mathcal{H}_i^{k*} \mathcal{H}_i^k \right)^{-1}. \tag{20}$$

Partition $\mathcal{Y}_i^k$ and $\mathcal{H}_i^k$ as follows:

$$\mathcal{Y}_i^k = \left[\frac{\mathcal{Y}_i^{k-1}}{d_k(i)} \right] \quad \text{and} \quad \mathcal{H}_i^k = \left[\frac{\mathcal{H}_i^{k-1}}{u_{k,i}} \right] \tag{21}$$

Then a spatial recursion for $P_{k,i}$ can be found by substituting (21) into (20):

$$\begin{aligned} P_{k,i}^{-1} &= \Pi + \mathcal{H}_i^{k*} \mathcal{H}_i^k \\ &= \Pi + \mathcal{H}_i^{k-1*} \mathcal{H}_i^{k-1} + u_{k,i}^* u_{k,i} \\ &= P_{k-1,i}^{-1} + u_{k,i}^* u_{k,i} \end{aligned} \tag{22}$$

which, by applying the matrix inversion lemma, leads to

$$P_{k,i} = P_{k-1,i} - \frac{P_{k-1,i} u_{k,i}^* u_{k,i} P_{k-1,i}}{1 + u_{k,i} P_{k-1,i} u_{k,i}^*}. \tag{23}$$

Substituting (21) and (23) into (19) we get:

$$\begin{aligned} \psi_k^{(i)} &= P_{k,i} \Big(\mathcal{H}_i^{k-1*} \mathcal{Y}_i^{k-1} + u_{k,i}^* d_k(i) \Big) \\ &= \underbrace{P_{k-1,i} \mathcal{H}_i^{k-1*} \mathcal{Y}_i^{k-1}}_{= \psi_{k-1}^{(i)}} \\ &\quad + P_{k-1,i} u_{k,i}^* \Big(1 - \frac{u_{k,i} P_{k-1,i} u_{k,i}^*}{1 + u_{k,i} P_{k-1,i} u_{k,i}^*} \Big) d_k(i) \\ &\quad - \frac{P_{k-1,i} u_{k,i}^* u_{k,i}}{1 + u_{k,i} P_{k-1,i} u_{k,i}^*} \underbrace{P_{k-1,i} \mathcal{H}_i^{k-1*} \mathcal{Y}_i^{k-1}}_{= \psi_{k-1}^{(i)}} \end{aligned}$$

which leads to

$$\psi_k^{(i)} = \psi_{k-1}^{(i)} + \frac{P_{k-1,i}}{1 + u_{k,i} P_{k-1,i} u_{k,i}^*} u_{k,i}^* e_k(i) \tag{24}$$

with

$$e_k(i) = d_k(i) - u_{k,i} \psi_{k-1}^{(i)}. \tag{25}$$

Grouping (23), (24), and (25) leads to (16). The algorithm structure is relatively simple and it can be understood as a standard least-squares solution unwrapped along the collaboration cycle. However, the nodes are exposed to data with distinct spatial and noise profiles. This variation reflects itself in the performance of the algorithm, which will be studied elsewhere.

5. LOW-COMMUNICATION DISTRIBUTED RLS ADAPTATION

The algorithm proposed in the previous section implements exact RLS distributively, whereby the nodes share information about the local weight estimates $\{\psi_k^{(i)}\}$ and the matrices $\{P_{k,i}\}$. A less complex solution that only shares information about the weight estimates can be obtained by requiring the matrices $\{P_{k,i}\}$ to evolve locally. This strategy leads to:

$$
\begin{aligned}
&\psi_0^{(i)} \leftarrow w_{i-1} \\
&\text{for } k = 1 : N \\
&\quad e_k(i) = d_k(i) - u_{k,i}\psi_{k-1}^{(i)} \\
&\quad \psi_k^{(i)} = \psi_{k-1}^{(i)} + \frac{P_{k,i-1}}{1+u_{k,i}P_{k,i-1}u_{k,i}^*}u_{k,i}^* e_k(i) \\
&\quad P_{k,i} = \left[P_{k,i-1} - \frac{P_{k,i-1}u_{k,i}^* u_{k,i} P_{k,i-1}}{1+u_{k,i}P_{k,i-1}u_{k,i}^*}\right] \\
&\text{end} \\
&w_i \leftarrow \psi_N^{(i)}.
\end{aligned}
\tag{26}
$$

To illustrate the operation of both algorithms dRLS and its low-communication counterpart (LC-dRLS), we consider a network with $N = 15$ nodes where each local filter has $M = 10$ taps. The system evolves for 30000 iterations and the results are averaged over 100 independent experiments. The steady-state mean-square error values are obtained by averaging the last 500 time samples. Each node accesses time-correlated spatially independent Gaussian regressors $\boldsymbol{u}_{k,i}$ with correlation functions $r_k(i) = \sigma_{u,k}^2 \cdot (\alpha_k)^{|i|}$, $i = 0, \ldots, M-1$, with $\{\alpha_k\}$ and $\{\sigma_{u,k}^2\}$ randomly chosen in $[0, 1)$ and depicted in Fig. 4. The background noise $\boldsymbol{v}_k(i)$ has variance $\sigma_{v,k}^2 = 10^{-3}$ across the network.

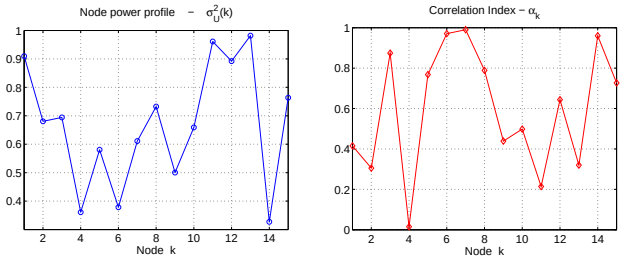

Fig. 4. Network statistical profile.

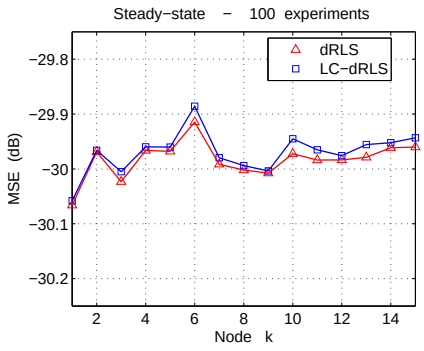

Fig. 5. MSE performance for both algorithms dRLS and LC-dRLS across the network.

Note in Fig. 5 that both distributed RLS algorithms have similar performance in the mean-square error sense, suggesting that the low-communication implementation can be a quite competitive suboptimal implementation.

6. DIFFUSION LMS ADAPTATION

When more communication and energy resources are available, we may take advantage of the network connectivity and devise more sophisticated peer-to-peer cooperation rules. We explore a simple diffusion protocol (see Fig. 6). Each individual node k consults its peer nodes from the neighborhood[1] $\mathcal{N}_k(i-1)$ and combines their past estimates $\{\psi_\ell^{(i-1)}; \ell \in \mathcal{N}_k(i-1)\}$ with its own past estimate $\psi_k^{(i-1)}$. The node generates an aggregated estimate $\phi_k^{(i-1)}$ and feeds it in its local adaptive filter. This strategy can be expressed as follows:

$$
\begin{aligned}
\phi_k^{(i-1)} &= f_k\left(\psi_\ell^{(i-1)}; \ell \in \mathcal{N}_k(i-1)\right) \\
\psi_k^{(i)} &= \phi_k^{(i-1)} + \mu u_{k,i}^*\left(d_k(i) - u_{k,i}\phi_k^{(i-1)}\right)
\end{aligned}
\tag{27}
$$

for some combiner $f_k(\cdot)$.

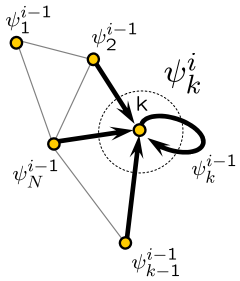

Fig. 6. A network with diffusion cooperation strategy.

In this work we explore a simple combining rule in which the aggregated estimate is generated by averaging local and neighbors' previous estimates, i.e.,

$$
\begin{aligned}
\phi_k^{(i-1)} &= \sum_{\ell \in \mathcal{N}_k} a(k,\ell)\, \psi_\ell^{(i-1)} \\
\psi_k^{(i)} &= \phi_k^{(i-1)} + \mu u_{k,i}^*\left(d_k(i) - u_{k,i}\phi_k^{(i-1)}\right)
\end{aligned}
\tag{28}
$$

where $a(k,\ell) = 1/\deg(k)$, with $\deg(k)$ denoting the degree of node k (number of incident links at this node, including itself). This scheme exploits network connectivity leading to more robust algorithms. Furthermore, since more information is aggregated in the local adaptive filter updates, individual nodes can attain better learning behavior when compared to the non-cooperative case, provided that the combiners f_k are well designed. To illustrate this fact, we run a simulation example with a network of $N = 10$ adaptive filters with $M = 10$ taps each. The topology is presented in Fig. 7, while the network statistical profile is presented in Fig. 8.

[1]The neighborhood of a node k is the set of nodes directly connected to it, including itself.

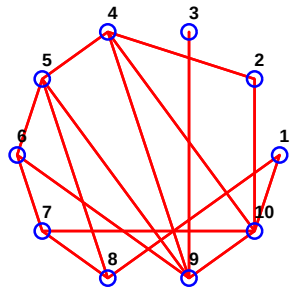

Fig. 7. Network topology (self-loops omitted) for the diffusion LMS example.

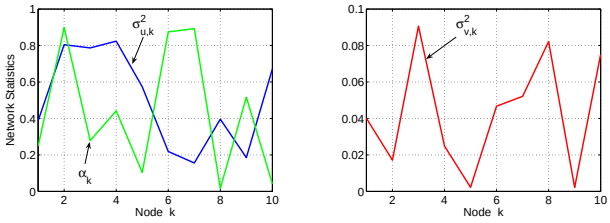

Fig. 8. Network statistical profile: regressors power and correlation indices (left) and noise power (right).

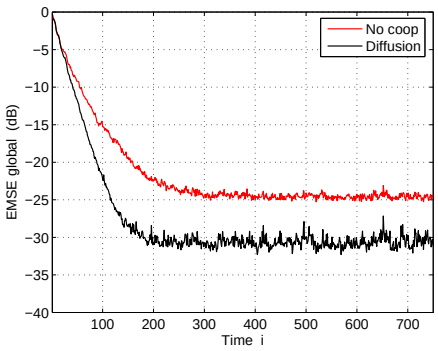

Fig. 9. Diffusion LMS - Transient global EMSE.

We compare the diffusion LMS with the non-cooperative case, in which the adaptive filters evolve independently accessing local data and consulting their own past estimates only. We use the global EMSE average, defined as

$$\zeta_g \triangleq \frac{1}{N}\sum_{k=1}^{N}\zeta_k \tag{29}$$

as a figure of merit, where ζ_k is the local EMSE at node k. Figure 9 shows the global learning behavior and Figure 9 presents the network individual EMSE profile in steady-state. Despite some losses at a few nodes, as for this case in node 9, on average, the entire system benefits from cooperation.

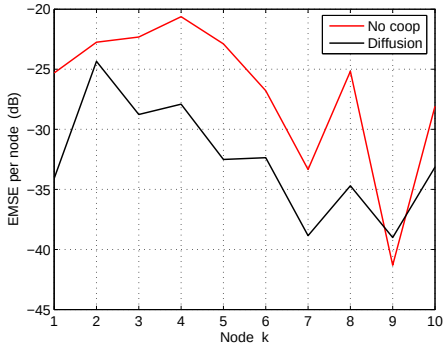

Fig. 10. Steady-state EMSE per node for diffusion LMS.

7. CONCLUDING REMARKS

We have described adaptive schemes to perform distributed estimation in a cooperative fashion. When communication and energy resources are scarce, the incremental LMS scheme may be used. When more powerful processors are available at the nodes, distributed RLS implementations with limited cooperation can be employed. For general topologies, and with more energy and communication resources, one can resort to diffusion LMS strategies. The diffusion techniques can also be extended to recursive least-squares formulations, which we will examine elsewhere.

8. 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. 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] C. Lopes and A. H. Sayed, "Distributed adaptive incremental strategies: formulation and performance analysis," *Proc. ICASSP*, Toulouse, France, May 2006.

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

[5] L. Xiao, S. Boyd and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," *Fourth IPSN*, Los Angeles, UCLA, Apr. 2005, pp. 63-70.