A DIFFUSION RLS SCHEME FOR DISTRIBUTED ESTIMATION OVER ADAPTIVE NETWORKS

Federico S. Cattivelli *Cassio G. Lopes* *Ali H. Sayed*

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

ABSTRACT

We consider the problem of distributed estimation in adaptive networks where a collection of nodes are required to estimate in a collaborative manner some parameter of interest from their measurements. The centralized solution to the problem uses a fusion center, thus requiring a large amount of energy for communication. We propose a diffusion recursive least-squares algorithm where nodes need to communicate only with their closest neighbors. The algorithm has no topology constraints, and requires no transmission or inversion of matrices, therefore saving in communications and complexity. We show that the algorithm is stable and analyze its performance comparing it to the centralized global solution.

1. INTRODUCTION

We consider the problem of distributed estimation where a collection of nodes are required to estimate in a collaborative manner some parameter of interest from their measurements. In the centralized solution to the problem, measurements are transmitted to a central fusion center for processing, and the resulting estimate is communicated back to the nodes. This approach enables the calculation of the global solution, but has the disadvantage of requiring a large amount of energy and communication [1]. An alternative approach is the distributed solution, in which the nodes communicate only with their closest neighbors and processing is done locally at every node, thereby saving communications and network resources.

A distributed estimation approach should have the following desirable features:

- Estimation performance: The nodes in the network should obtain estimates that are close to the global solution.
- Energy awareness and complexity: The solution should minimize communications and local processing at the nodes.
- Ad-hoc deployment: The system should be able to cope with possibly dynamic configurations of the network.

Distributed estimation algorithms have been proposed to address these issues to some extent. In [2, 3, 4], a distributed incremental RLS solution was proposed for obtaining the exact global least-squares estimate. The algorithm requires the definition of a path through the network, which may not be practical for large networks or dynamic configurations. In [5], an isotropic diffusion algorithm was proposed; it requires every node to transmit and invert a matrix at every iteration, which is prohibitive for large matrices or for low complexity sensor nodes.

We propose a distributed diffusion algorithm based on RLS that has performance close to the global solution and outperforms earlier solutions; it also does not require transmission or inversion of matrices, therefore saving in communications and computational complexity. The algorithm has no topological constraints, and constitutes a fully adaptive, recursive and distributed solution that is also asymptotically unbiased and stable under the modeling assumptions of Section 4.

This material was based on work supported in part by the National Science Foundation under awards ECS-0401188 and ECS-0601266.

2. THE ESTIMATION PROBLEM

2.1. Global Least-Squares Problem

Consider a set of N nodes spatially distributed over some region. Let $\mathcal{N}_k$ denote the *closed* neighborhood of node k (i.e., the set of all neighbors of k including itself). The objective is to collectively estimate an unknown deterministic column vector of length M, denoted by w^o, using least-squares estimation. At *every* time instant i, node k obtains a measurement $d_k(i)$ that is related to the unknown vector by

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

where $u_{k,i}$ is a row vector of length M (the regressor of node k at time i), and $v_k(i)$ is a zero-mean, spatially uncorrelated Gaussian white noise process with variance $\sigma^2_{v_k}$. At time i, we collect the measurements and noise samples of all N nodes into vectors y_i and v_i of length N, and the regressors into an N by M matrix H_i, as follows:

$$\begin{array}{rcll} y_i & = & \text{col}\{d_1(i), ..., d_N(i)\} & (N \times 1) \\ H_i & = & \text{col}\{u_{1,i}, ..., u_{N,i}\} & (N \times M) \\ v_i & = & \text{col}\{v_1(i), ..., v_N(i)\} & (N \times 1) \end{array}$$

Let v_i^* denote the conjugate transpose of vector v_i. The covariance matrix of the noise vector is $R_v = Ev_iv_i^* = \text{diag}\{\sigma^2_{v_1}, ..., \sigma^2_{v_N}\}$. We collect the regressors, measurements and covariance matrices from time 0 up to time i as follows:

$$\begin{array}{rcl} \mathcal{Y}_i & = & \text{col}\{y_i, ..., y_0\} \\ \mathcal{H}_i & = & \text{col}\{H_i, ..., H_0\} \\ \mathcal{V}_i & = & \text{col}\{v_i, ..., v_0\} \end{array}$$

and let $\mathcal{R}_{v,i} = \text{E}\mathcal{V}_i\mathcal{V}_i^*$.

The objective is to estimate w^o by solving the following weighted, regularized, least-squares problem:

$$\min_w ||w - \bar{w}||^2_{\Pi_i} + ||\mathcal{Y}_i - \mathcal{H}_i w||^2_{\mathcal{W}_i} \tag{2}$$

The global solution, w_i, is given by [6]

$$w_i = \bar{w} + [\Pi_i + \mathcal{H}_i^* \mathcal{W}_i \mathcal{H}_i]^{-1} \mathcal{H}_i^* \mathcal{W}_i (\mathcal{Y}_i - \mathcal{H}_i \bar{w}) \quad (3)$$

where $\Pi_i > 0$ is a regularization matrix and $\mathcal{W}_i \geq 0$ is a weighting matrix. Both Π_i and $\mathcal{W}_i$ are Hermitian.

An exponentially weighted problem can be formulated by choosing $\mathcal{W}_i = \mathcal{R}_{v,i}^{-1} \Lambda_i$ and $\Pi_i = \lambda^{i+1} \Pi$, with $0 < \lambda \leq 1$, $\Pi > 0$ and

$$\Lambda_i \triangleq \text{diag}\{I_N, \lambda I_N, ..., \lambda^i I_N\}$$

Usually, $\Pi = \delta^{-1} I_M$ where $\delta > 0$ is large. For the choice $\bar{w} = 0$, the estimation problem becomes:

$$w_i = \arg\min_w \left\{ \lambda^{i+1} ||w||_\Pi^2 + \sum_{j=0}^{i} \lambda^{i-j} \sum_{l=1}^{N} \frac{|d_l(j) - u_{l,j} w|^2}{\sigma_{v_l}^2} \right\} \quad (4)$$

We refer to this problem as the *global least-squares problem*, since at time i, the solution takes into account all measurements from all nodes up to time i. This solution may be computed by using a centralized approach, or the distributed incremental RLS algorithm of [3, 4, 7].

2.2. Clustered Least-Squares Problem

We now develop distributed estimation schemes where nodes have access to limited data, namely the data from the neighboring nodes. When node k can only share measurements and regressors with its neighbors, it can locally solve the following least-squares problem:

$$\psi_{k,i} = \arg\min_w \left\{ \lambda^{i+1} ||w||_\Pi^2 + \sum_{j=0}^{i} \lambda^{i-j} \sum_{l=1}^{N} \frac{C_{l,k} |d_l(j) - u_{l,j} w|^2}{\sigma_{v_l}^2} \right\} \quad (5)$$

for some choice of coefficients $C_{l,k}$ such that $C_{l,k} = 0$ if $l \notin \mathcal{N}_k$. We also choose these coefficients such that $\sum_{k=1}^{N} C_{l,k} = 1$, $l = 1 \ldots N$, and $\sum_{l=1}^{N} C_{l,k} = 1$, $k = 1 \ldots N$, for reasons that will become clearer later in this paper. Note that the nodes in the network will generally have access to different data, so their estimates will be solutions to different least-squares problems. Naturally, we want these estimates to be close to the global least-squares solution of (4).

The coefficients $C_{l,k}$ can be incorporated into the weighting matrix of (2) by forming an $N \times N$ matrix C such that its $\{l, k\}$ element is $C_{l,k}$ and replacing $\mathcal{W}_i$ by:

$$\mathcal{W}_{k,i} = \mathcal{R}_{v,i}^{-1} \Lambda_i \text{diag}\{C_k, C_k, ..., C_k\} \quad (6)$$

with $C_k \triangleq \text{diag}\{C e_k\}$, and e_k being an $N \times 1$ vector with a unity entry in position k and zeros elsewhere.

Now, for every node k, we express its local estimate $\psi_{k,i}$ as a perturbation of w^o, say for some error vectors $z_{k,i}$.

$$\begin{bmatrix} \psi_{1,i} \\ \psi_{2,i} \\ \vdots \\ \psi_{N,i} \end{bmatrix} = \begin{bmatrix} I_M \\ I_M \\ \vdots \\ I_M \end{bmatrix} w^o + \underbrace{\begin{bmatrix} z_{1,i} \\ z_{2,i} \\ \vdots \\ z_{N,i} \end{bmatrix}}_{z_i}$$

Note that this model holds exactly for any least-squares problem of the form (2) when there is no regularization ($\Pi_i = 0$), since in this case $\psi_{k,i}$ is given by

$$\psi_{k,i} = [\mathcal{H}_i^* \mathcal{W}_{k,i} \mathcal{H}_i]^{-1} \mathcal{H}_i^* \mathcal{W}_{k,i} \mathcal{Y}_i$$

and $\mathcal{Y}_i = \mathcal{H}_i w^o + \mathcal{V}_i$ by definition. Then it holds that

$$z_{k,i} = [\mathcal{H}_i^* \mathcal{W}_{k,i} \mathcal{H}_i]^{-1} \mathcal{H}_i^* \mathcal{W}_{k,i} \mathcal{V}_i$$

and $z_{k,i}$ is zero-mean and Gaussian as well.

Given the local estimates $\psi_{k,i}$ we may now pose the following least-squares problem:

$$\min_w \left\| \begin{bmatrix} \psi_{1,i} \\ \psi_{2,i} \\ \vdots \\ \psi_{N,i} \end{bmatrix} - \begin{bmatrix} I_M \\ I_M \\ \vdots \\ I_M \end{bmatrix} w \right\|_W^2 \quad (7)$$

If we assume that the individual solutions $\psi_{k,i}$ are good approximations for the optimal solution w^o, then we would expect the solution of problem (7) to be closer to the optimal. For instance, for the choice of weighting matrix

$$W = \begin{bmatrix} \mathcal{H}_i^* \mathcal{W}_{1,i} \mathcal{H}_i & & \\ & \ddots & \\ & & \mathcal{H}_i^* \mathcal{W}_{N,i} \mathcal{H}_i \end{bmatrix}$$

we have that the solution of (7) is

$$\left[\mathcal{H}_i^* \left(\sum_{k=1}^{N} \mathcal{W}_{k,i} \right) \mathcal{H}_i \right]^{-1} \mathcal{H}_i^* \left(\sum_{k=1}^{N} \mathcal{W}_{k,i} \right) \mathcal{Y}_i \quad (8)$$

For the choice of coefficients $C_{l,k}$ such that $C\mathbb{1} = \mathbb{1}$, it follows that $\sum_k C_k = I_N$. Thus Equation (8) becomes

$$\left[\mathcal{H}_i^* \mathcal{R}_{v,i}^{-1} \Lambda_i \mathcal{H}_i \right]^{-1} \mathcal{H}_i^* \mathcal{R}_{v,i}^{-1} \Lambda_i \mathcal{Y}_i$$

which is the solution of (4) (i.e., the global solution) when there is no regularization.

This result leads to an estimation process in two steps. First, every node solves a local least-squares problem using local data as in (5), and second, the nodes communicate to solve problem (7). An inconvenience of this approach is that in order to obtain the global solution, the second step requires knowledge of the local estimate of every other node, and also knowledge of the matrix W, which depends on all regressors in the network.

However, we can modify this method to fit a diffusion estimation scheme whereby nodes only share estimates with their neighbors. To do so, consider the case where we choose the weighting matrix W in (7) as a node-dependent diagonal matrix of the form $W_k = \text{diag}\{A_{1,k} I, ..., A_{N,k} I\}$, where $A_{l,k} = 0$ if $l \notin \mathcal{N}_k$, and $\sum_{l=1}^{N} A_{l,k} = 1$, $k = 1, \ldots, N$. Then, the solution to the least-squares problem (7) takes the form:

$$w_{k,i} = \sum_{l=1}^{N} A_{l,k} \psi_{l,i} \quad (9)$$

This means that the second step of the aforementioned method can be simplified by a weighted average of local estimates.

This sequence of two least-squares problems represents an attempt to solve the global least-squares problem in a distributed manner. However, the method is not adaptive, since we first need to calculate local estimates, and then average the results as in (9). We can make the method adaptive if we perform both steps for every measurement. That is, first aggregate the new data into the previous estimate, and then combine estimates with neighbors. This procedure, referred to as diffusion RLS, has good convergence properties as is shown in Section 4.

3. THE DIFFUSION RLS ALGORITHM

We therefore propose a diffusion RLS algorithm to collectively estimate w^o from individual measurements in two steps:

1. At time i, the nodes communicate their measurements $d_k(i)$ and regressors $u_{k,i}$ with their closest neighbors, and use this data to update their local estimates using RLS iterations (via incremental update). The resulting pre-estimates are named $\psi_{k,i}$ from (5).

2. The nodes communicate their local pre-estimates with their closest neighbors and perform a weighted average as in (9) to obtain the final estimate $w_{k,i}$ (via spatial update).

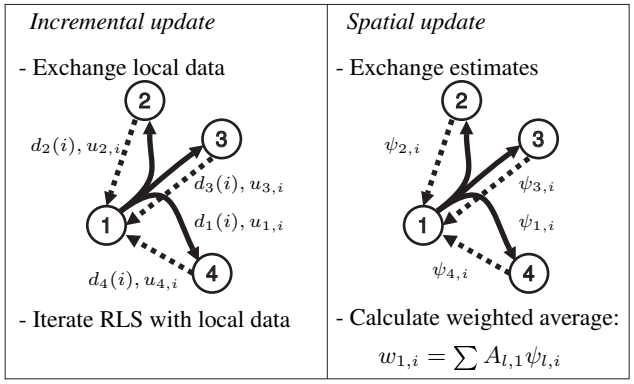

Fig. 1. Diffusion RLS algorithm at node 1. Incremental update: all neighboring nodes exchange measurements and regressors and compute $\psi_{k,i}$. Spatial update: neighboring nodes exchange $\psi_{k,i}$ and perform a weighted average to obtain $w_{k,i}$

The algorithm, shown schematically in Figure 1, is described by Equations (10) below. Start by selecting $N \times N$ matrices A and C with non-negative entries such that $A_{l,k} = C_{l,k} = 0$ if $l \notin \mathcal{N}_k$, $\mathbb{1}^* A = \mathbb{1}^*$, $\mathbb{1}^* C = \mathbb{1}^*$ and $C\mathbb{1} = \mathbb{1}$. The algorithm is given by the following equations

Diffusion RLS Algorithm

Start with $w_{k,-1} = 0$ and $P_{k,-1} = \Pi^{-1}$ for each node k.

For every time instant *i*, repeat

Incremental update: for every node k, repeat

$$\begin{aligned}
&\psi_{k,i} = w_{k,i-1} \\
&P_{k,i} = \lambda^{-1} P_{k,i-1} \\
&\text{for all } l \in \mathcal{N}_k \\
&\quad \psi_{k,i} \leftarrow \psi_{k,i} + \tfrac{C_{l,k} P_{k,i} u_{l,i}^* [d_l(i) - u_{l,i}\psi_{k,i}]}{\sigma_{v_l}^2 + C_{l,k} u_{l,i} P_{k,i} u_{l,i}^*} \\
&\quad P_{k,i} \leftarrow P_{k,i} - \tfrac{C_{l,k} P_{k,i} u_{l,i}^* u_{l,i} P_{k,i}}{\sigma_{v_l}^2 + C_{l,k} u_{l,i} P_{k,i} u_{l,i}^*} \\
&\text{end}
\end{aligned} \tag{10}$$

Spatial update: for every node k, repeat

$w_{k,i} = \sum_{l \in \mathcal{N}_k} A_{l,k} \psi_{l,i}$

Note that the algorithm requires no matrix inversion, and nodes only need to know the estimates $\psi_{k,i}$, measurements $d_k(i)$ and regressors $u_{k,i}$ of their neighbors. Thus, every node needs to communicate a total of $2M + 1$ scalars to neighboring nodes, compared with $(M^2 + 3M)/2$ scalars if the matrices $P_{k,i}$ were to be shared instead.

4. ANALYSIS

In this section we show that the algorithm (10) is asymptotically unbiased in the mean and that it converges in the mean-square error sense under some simplifying assumptions. We also provide an expression for the steady-state mean-square deviation (MSD) and compare to the global least-squares solution.

4.1. Data model

We will make the following assumption on the regressors.

Assumption 1 *The regressors $u_{k,i}$ are zero-mean, and independent in time. Moreover, the covariance matrix $R_{u_k} \triangleq E u_{k,i}^* u_{k,i}$ is invariant over time.*

From (10) we know that

$$\begin{aligned}
P_{k,i}^{-1} &= \lambda P_{k,i-1}^{-1} + \sum_{l=1}^{N} \frac{C_{l,k}}{\sigma_{v_l}^2} u_{l,i}^* u_{l,i} \\
&= \lambda^{i+1}\Pi + \sum_{j=0}^{i} \lambda^{i-j} \sum_{l=1}^{N} \frac{C_{l,k}}{\sigma_{v_l}^2} u_{l,j}^* u_{l,j}
\end{aligned} \tag{11}$$

We are interested in the steady-state behavior of the matrix $P_{k,i}$. As $i \to \infty$, and for $0 < \lambda < 1$, the steady state mean value of $P_{k,i}^{-1}$ is given by

$$\lim_{i \to \infty} \mathrm{E} P_{k,i}^{-1} = \frac{1}{1-\lambda} \sum_{l=1}^{N} \frac{C_{l,k}}{\sigma_{v_l}^2} R_{u_l} \triangleq P_k^{-1} \tag{12}$$

In order to make the performance analysis tractable, we introduce the following ergodicity assumption.

Assumption 2 *$\exists i_0$ such that for all $i > i_0$, $P_{k,i}^{-1}$ can be replaced by $E P_{k,i}^{-1} = P_k^{-1}$*

Assumption 2 states that the time average of a sequence of random variables can be replaced by its expected value. This is a common assumption in the analysis of the performance of RLS (see for example [6]), and yields good results in practice as is shown by simulation in Section 5.

Furthermore, to prove mean-square convergence we will replace the random matrix $P_{k,i}$ by P_k. We will also need the following assumption.

Assumption 3 *The covariance matrix $R_{u_k} = E u_{k,i}^* u_{k,i}$ and the noise variance $\sigma_{v_k}^2$ are the same at every node, i.e., $R_{u_1} = ... = R_{u_N} \triangleq R_u$ and $\sigma_{v_1}^2 = ... = \sigma_{v_N}^2 \triangleq \sigma_v^2$.*

4.2. Mean performance

Combining Equations (1), (10) and (11), we have the following relation for the local estimate:

$$\begin{aligned}
P_{l,i}^{-1}\psi_{l,i} &= \lambda P_{l,i-1}^{-1} w_{l,i-1} + \sum_{m=1}^{N} \frac{C_{m,l}}{\sigma_{v_m}^2} u_{m,i}^* d_m(i) \\
&= \lambda P_{l,i-1}^{-1} w_{l,i-1} + \sum_{m=1}^{N} \frac{C_{m,l}}{\sigma_{v_m}^2} u_{m,i}^* [u_{m,i} w^o + v_m(i)] \\
&= \lambda P_{l,i-1}^{-1} \tilde{w}_{l,i-1} + P_{l,i}^{-1} w^o + \sum_{m=1}^{N} \frac{C_{m,l}}{\sigma_{v_m}^2} u_{m,i}^* v_m(i)
\end{aligned}$$

It then follows that the error vector $\tilde{w}_{k,i} \triangleq w_{k,i} - w^o$ is given by

$$\begin{aligned}\tilde{w}_{k,i} &= \sum_{l=1}^N A_{l,k}[\psi_{l,i} - w^o] \\ &= \sum_{l=1}^N A_{l,k} P_{l,i} \left[\lambda P_{l,i-1}^{-1} \tilde{w}_{l,i-1} + \sum_{m=1}^N \frac{C_{m,l}}{\sigma_{v_m}^2} u_{m,i}^* v_m(i)\right]\end{aligned} \tag{13}$$

Taking expectations of both sides yields the following result:

$$\mathrm{E}\{\tilde{w}_{k,i}\} = \lambda \mathrm{E}\left\{\sum_{l=1}^N A_{l,k} P_{l,i} P_{l,i-1}^{-1} \tilde{w}_{l,i-1}\right\}$$

Using assumption 2 and noting that $P_{k,i}^{-1}$ becomes independent of i for $i > i_0$, and grouping the vectors $\tilde{w}_{k,i}$ into a matrix $\tilde{W}_i = \mathrm{row}\{\tilde{w}_{1,i}, ..., \tilde{w}_{N,i}\}$, we have for large enough i:

$$\begin{aligned}\mathrm{E}\{\tilde{W}_i\} &= \lambda \mathrm{E}\{\tilde{W}_{i-1}\} A \\ &= \lambda^{i-i_0} \mathrm{E}\{\tilde{W}_{i_0}\} A^{i-i_0}\end{aligned}$$

Assuming that all elements of $\mathrm{E}\{\tilde{W}_{i_0}\}$ are bounded in absolute value by some constant $0 \leq a < \infty$, and since all elements of A^i are between zero and one, we have that every element of $\mathrm{E}\{\tilde{W}_i\}$ must be bounded in absolute value by $\lambda^{i-i_0} N a$. Thus, for $0 < \lambda < 1$, every element of $\mathrm{E}\{\tilde{W}_i\}$ converges to zero as $i \to \infty$, and the estimator is asymptotically unbiased.

4.3. Mean-square performance

We now show that the algorithm converges in the mean-square sense, i.e., $\lim_{i\to\infty} E||\tilde{w}_{k,i}||^2 < \infty$. To show mean-square convergence we use assumptions 1, 2 and 3.

From assumption 3, P_k becomes independent of the node k, and we can define:

$$P^{-1} \triangleq P_k^{-1} = \frac{1}{1-\lambda} \frac{R_u}{\sigma_{v_l}^2}$$

From equation (13) and assumption 2 we have:

$$\begin{aligned}\tilde{w}_{k,i} &= \lambda \sum_{l=1}^N A_{l,k} \tilde{w}_{l,i-1} + P \sum_{l=1}^N A_{l,k} \sum_{m=1}^N \frac{C_{m,l}}{\sigma_{v_m}^2} u_{m,i}^* v_m(i) \\ &= \lambda \tilde{W}_{i-1} A e_k + P H_i^* R_v^{-1} \mathrm{diag}(v_i) C A e_k\end{aligned} \tag{14}$$

Again grouping the error vectors $\tilde{w}_{k,i}$ into a matrix $\tilde{W}_i$ we have:

$$\begin{aligned}\tilde{W}_i &= \lambda \tilde{W}_{i-1} A + P H_i^* R_v^{-1} \mathrm{diag}(v_i) C A \\ &= \tilde{W}_{i_0} (\lambda A)^{i-i_0} + P \sum_{j=i_0+1}^i H_j^* R_v^{-1} \mathrm{diag}(v_j) C A (\lambda A)^{i-j}\end{aligned}$$

For node k we have:

$$\tilde{w}_{k,i} = \tilde{W}_{i_0} (\lambda A)^{i-i_0} e_k + P \sum_{j=i_0+1}^i H_j^* \mathrm{diag}[C A (\lambda A)^{i-j} e_k] R_v^{-1} v_j$$

Now define $B \triangleq A^{i-i_0}$, and the mean-square deviation is

$$\begin{aligned}\mathrm{E}||\tilde{w}_{k,i}||^2 = &\ \lambda^{2(i-i_0)} \mathrm{E}\left|\left|\sum_{l=1}^N B_{l,k} \tilde{w}_{l,i_0}\right|\right|^2 + \\ &\ \mathrm{E}\mathrm{Tr}\left[\frac{P^2}{\sigma_v^2} \sum_{j=i_0+1}^i H_j^* \mathrm{diag}[C A (\lambda A)^{i-j} e_k]^2 H_j\right] \\ = &\ \lambda^{2(i-i_0)} \mathrm{E}\left|\left|\sum_{l=1}^N B_{l,k} \tilde{w}_{l,i_0}\right|\right|^2 + \\ &\ (1-\lambda)^2 \sigma_v^2 \mathrm{Tr}(R_u^{-1}) \sum_{j=i_0+1}^i ||C A (\lambda A)^{i-j} e_k||^2\end{aligned}$$

Since the matrix B has positive entries between 0 and 1, the first term of equation (15) can be ignored for large i, and we are left with the right term in steady state. Thus, the MSD of the diffusion RLS algorithm is:

$$\boxed{\mathrm{MSD}_k^{\mathrm{diff}} \triangleq \mathrm{E}||\tilde{w}_{k,\infty}||^2 = (1-\lambda)^2 \sigma_v^2 \mathrm{Tr}(R_u^{-1}) S_{k,k}} \tag{15}$$

where $S_{k,k}$ is the (k,k) element of the matrix S defined by

$$S = \lim_{i\to\infty} \sum_{j=0}^i \lambda^{2j} (A^*)^{j+1} C^* C A^{j+1} \tag{16}$$

This result can be simplified for the case $A = A^* = C$. Again using the fact that powers of A have positive elements between 0 and 1, we can conclude that the steady state MSD is given by:

$$\mathrm{MSD}_k^{\mathrm{diff}} = (1-\lambda)^2 \sigma_v^2 \mathrm{Tr}(R_u^{-1}) e_k^* A^4 [I - \lambda^2 A^2]^{-1} e_k \tag{17}$$

The steady state MSD of the global solution can be found by choosing $A = (1/N)\mathbb{1}\mathbb{1}^*$ in equation (17) and noting that A is idempotent, so that $A[I - \lambda^2 A]^{-1} = A/(1-\lambda^2)$ to give:

$$\boxed{\mathrm{MSD}^{\mathrm{global}} = \frac{1-\lambda}{1+\lambda} \frac{\sigma_v^2}{N} \mathrm{Tr}(R_u^{-1})} \tag{18}$$

This is a known result for the MSD of RLS with λ close to 1 [6].

We summarize the results of this section in the following Lemma

Lemma 1 *Under assumptions 1, 2 and 3, the diffusion RLS algorithm of (10) is asymptotically unbiased and its mean-square deviation is given by (15).*

5. SIMULATIONS

We now show simulations of the performance of the diffusion RLS algorithm of (10), and compare it to other algorithms such as the distributed RLS (dRLS) of [2] and the space-time diffusion (STD) of [5].

The measurements were generated according to equation (1), and the regressors $u_{k,i}$ were chosen Gaussian iid. The network had a total of $N = 20$ nodes, the size of the unknown vector was $M = 5$, and the noise was Gaussian with variance 0.2 for every node. To simulate the STD of [5], Metropolis weights were used, defined as:

$$C_{l,k} = \begin{cases} 1/\max\{\deg_l, \deg_k\} & l \in \mathcal{N}_k, l \neq k \\ 1 - \sum_{l\neq k} C_{l,k} & l = k \\ 0 & \text{otherwise} \end{cases} \tag{19}$$

where $\deg_k$ represents the degree of node k (i.e., the cardinality of its closed neighborhood). For diffusion RLS, Metropolis weights were used for C, and A was chosen such that $A_{l,k}$ is proportional to the degree of node l and normalizing such that $\mathbb{1}^* A = \mathbb{1}^*$. Since STD does not account for the forgetting factor λ, we set $\lambda = 1$ for diffusion RLS in these simulations.

Figure 2 shows the performance of the different algorithms in terms of the MSD as a function of the number of measurements per node, averaged over 100 experiments. The curve labeled "Local" corresponds to the local least-squares estimates where nodes can

only access measurements and regressors of their closest neighbors. Also shown are the curves for diffusion RLS (diffRLS) of equation (10), STD of [5] and dRLS of [2]. We can see that diffusion RLS improves the performance considerably compared to the local estimate, and also has advantages over STD.

Figure 3 shows the MSD of different algorithms as a function of the number of scalars communicated per node. We also observe that diffusion RLS has important improvements over STD. That is, in order to obtain a fixed MSD, diffusion RLS requires less communications (measured in terms of scalars transmitted) than STD, and the performance in this sense is close to that of dRLS.

In Figure 4 we compare the theoretical results derived in Section 4.3 for the steady state MSD of diffusion RLS given by Equation (15) and the global solution given by Equation (18), using a value of $\lambda = 0.9$. We can see that in both cases, the simulated steady state MSD agrees well with the theoretical results.

The algorithms were also compared using different data models, where the time independence and identically distributed assumptions were dropped. Figure 5 shows the performance of the algorithms when the noise variances were different for every node (between 0.1 and 0.2), and the regressors were chosen to have shift structure, the elements of $u_{k,i}$ forming a Markov process with different coefficients for every node. Again a value of $\lambda = 1$ was used. We did not observe a significant degradation of performance by using different data models.

6. CONCLUSIONS

We have addressed the problem of distributed estimation in sensor networks. We have proposed a diffusion RLS algorithm that obtains good performance without having to transmit or invert matrices, has no topology constraints, and is robust to link failure. We have shown mean and mean-squared convergence under ergodicity assumptions and derived expressions for the steady state mean-squared deviation that agree with the simulation results. We also simulated the filter under some general conditions, always observing good performance. Thus, diffusion RLS is a promising alternative in adaptive networks where good performance, low complexity and low communications are required.

7. REFERENCES

[1] D. Estrin, L. Girod, G. Pottie, and M. Srivastava, "Instrumenting the world with wireless sensor networks," in *Proc. ICASSP*, Salt Lake City, UT, May 2001, pp. 2033–2036 vol.4.

[2] C. G. Lopes and A. H. Sayed, "Distributed processing over adaptive networks," in *Proc. of the Adaptive Sensor Array Processing (ASAP) Workshop*, Cambridge, MA, June 2006.

[3] A. H. Sayed and C. G. Lopes, "Distributed recursive least-squares strategies over adaptive networks," in *Proc. 40th Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, October 2006.

[4] A. H. Sayed and C. G. Lopes, "Adaptive processing over distributed networks," to appear in *IEICE Transactions*, 2007.

[5] L. Xiao, S. Boyd, and S. Lall, "A space-time diffusion scheme for peer-to-peer least-squares estimation," in *Proc IPSN*, Nashville, TN, April 2006, pp. 168–176.

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

[7] C. G. Lopes and A. H. Sayed, "Incremental adaptive strategies over distributed networks," to appear in *IEEE Transactions on Signal Processing*, 2007.

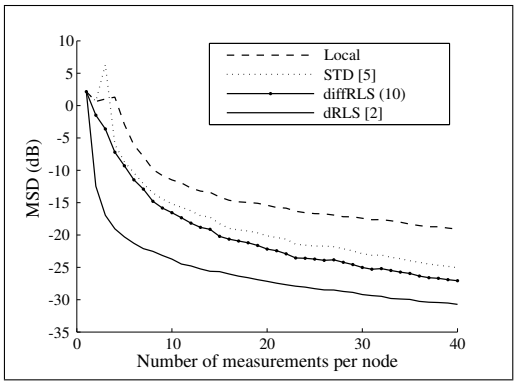

Fig. 2. MSD curves of different algorithms, $\lambda = 1$.

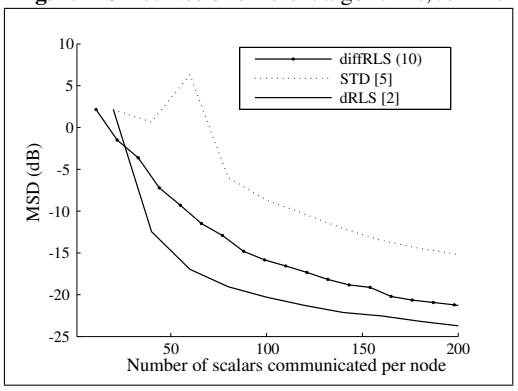

Fig. 3. MSD curves of different algorithms as a function of number of scalars communicated per node, $\lambda = 1$.

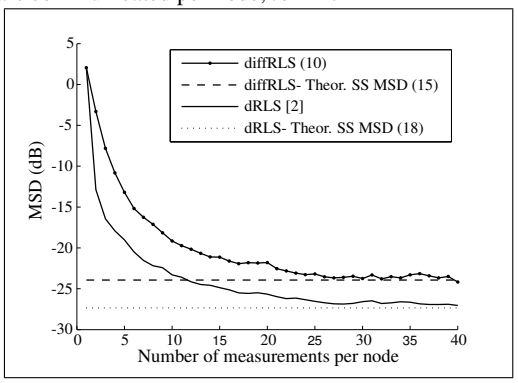

Fig. 4. Theoretical and simulated MSD curves, $\lambda = 0.9$.

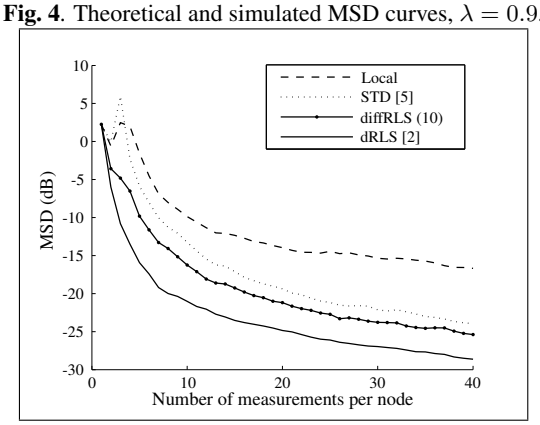

Fig. 5. MSD curves of different algorithms, $\lambda = 1$.