# DISTRIBUTED RECURSIVE LEAST-SQUARES STRATEGIES OVER ADAPTIVE NETWORKS

*Ali H. Sayed and Cassio G. Lopes*

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

## ABSTRACT

A distributed least-squares estimation strategy is developed by appealing to collaboration techniques that exploit the space-time structure of the data, achieving an exact recursive solution that is fully distributed. Each node is allowed to communicate with its immediate neighbor in order to exploit the spatial dimension, while it evolves locally to account for the time dimension as well. In applications where communication and energy resources are scarce, an approximate RLS scheme that is also fully distributed is proposed in order to decrease the communication burden necessary to implement distributed collaborative solution. The performance of the resulting algorithm tends to its exact counterpart in the mean-square sense as the forgetting factor $\lambda$ tends to unity. A spatial-temporal energy conservation argument is used to evaluate the steady-state performance of the individual nodes across the adaptive distributed network for the low communications RLS implementation. Computer simulations illustrate the results.

## 1. INTRODUCTION

In recent work, distributed adaptive algorithms have been proposed to address the issue of estimation over distributed networks [1], [2]. The algorithms recognize the importance of collaborative and adaptive processing, and particularly the use of simple processing and cooperative strategies in order to attend to applications with limited resources [3], [4]. For example, the use of LMS-type algorithms eliminates the need to embed powerful processors at the nodes in such applications [1]. However, available processors continuously decrease in cost and increase in computational capability. In order to equip the network with more sophisticated adaptation rules, we derive in this paper an exact and distributed RLS implementation that delivers to every node the global least-squares solution considering all data collected by the network. 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].

In the scheme proposed in this paper, every node will resort to a collaboration strategy that requires limited local node interaction, thus decreasing the communication requirement to solve the problem distributively. We also modify the proposed scheme to decrease further the communication burden, while keeping the performance relatively close to that of the exact solution in the mean-square sense. Using a spatial-temporal energy conservation argument [1, 6], the steady-state performance of the individual nodes across the adaptive distributed network is evaluated for the low-communications scheme. The results are illustrated via computer simulations.

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.

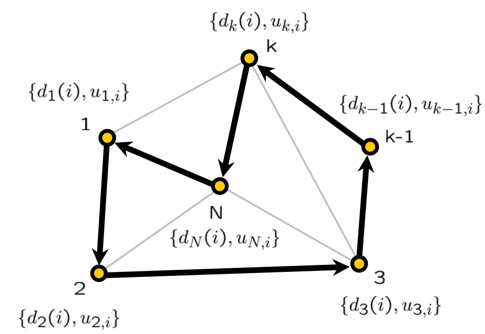


**Fig. 1**. A distributed network with $N$ nodes and the collaboration path.

## 2. PROBLEM FORMULATION

We are interested in estimating an unknown vector $w^o$ from multiple measurements collected at $N$ nodes in a network (see Fig. 1). Each node $k$ has access to regressor and measurement data $u_{k,i}$ and $d_k(i)$, $k = 1, \ldots, N$, where $d_k(i)$ is a scalar and $u_{k,i}$ is $1 \times M$. At each time instant $i$, the network has access to space-time data

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

Here $y_i$ and $H_i$ are snapshot matrices revealing the network data status at time $i$. We collect all the data available up to time $i$ into global matrices $\mathcal{Y}_i$ and $\mathcal{H}_i$

$$\mathcal{Y}_i = \begin{bmatrix} y_0 \\ \hline y_1 \\ \hline \vdots \\ \hline y_i \end{bmatrix} \quad \text{and} \quad \mathcal{H}_i = \begin{bmatrix} H_0 \\ \hline H_1 \\ \hline \vdots \\ \hline H_i \end{bmatrix}. \tag{2}$$

and pose the problem of seeking an estimate for $w^o$ that takes into account time and space-node relevance by solving a regularized weighted least-squares (LS) problem of the form:

$$\min_w \left[\lambda^{i+1} w^* \Pi w + \left\|\mathcal{Y}_i - \mathcal{H}_i w\right\|^2_{\mathcal{W}_i}\right] \tag{3}$$

where a *weighted norm* notation is employed. For a vector $x$ and a Hermitian matrix $A > 0$, $\|x\|_A^2 = x^* A x$. The weighting matrix in (3) is chosen as

$$\mathcal{W}_i \triangleq \operatorname{diag}\{\lambda^i D, \lambda^{i-1} D, \cdots, \lambda D, D\} \tag{4}$$

with a spatial weighting factor $D = \operatorname{diag}\{\gamma_1, \gamma_2, \cdots, \gamma_N\}$, $\gamma_i \geq 0$ and (time) forgetting factor $0 \ll \lambda \leq 1$. Moreover, $\Pi > 0$. The solution of problem (3) is given by [6]:

$$w_i = P_i \mathcal{H}_i^* \mathcal{W}_i \mathcal{Y}_i \tag{5}$$

where

$$P_i = \left(\lambda^{i+1}\Pi + \mathcal{H}_i^* \mathcal{W}_i \mathcal{H}_i\right)^{-1} \tag{6}$$

## 3. EXACT DISTRIBUTED LEAST-SQUARES IMPLEMENTATION

We are interested in a distributed recursion to update $w_{i-1}$ to $w_i$. We proceed by first deriving a distributed recursion for $P_i$. Relation (6) can be written as

$$\begin{aligned} P_i^{-1} &= \lambda^{i+1}\Pi + \mathcal{H}_i^* \mathcal{W}_i \mathcal{H}_i \\ &= \lambda\left(\lambda^i \Pi + \mathcal{H}_{i-1}^* \mathcal{W}_{i-1} \mathcal{H}_{i-1}\right) + H_i^* D H_i \\ &= \lambda P_{i-1}^{-1} + H_i^* D H_i \end{aligned} \tag{7}$$

which can be rewritten as a sequence of rank-1 updates:

$$\begin{aligned} P_{0,i} &\leftarrow \lambda^{-1} P_{i-1} \\ P_{1,i} &= \left(P_{0,i}^{-1} + \gamma_1 u_{1,i}^* u_{1,i}\right)^{-1} \\ P_{2,i} &= \left(P_{1,i}^{-1} + \gamma_2 u_{2,i}^* u_{2,i}\right)^{-1} \\ &\vdots \\ P_{N,i} &= \left(P_{N-1,i}^{-1} + \gamma_N u_{N,i}^* u_{N,i}\right)^{-1} \\ P_i &\leftarrow P_{N,i} \end{aligned}$$

By using the matrix inversion lemma for node $k$, a distributed recursion for $P_i$ is obtained:

$$\left\{ \begin{array}{l} P_{0,i} \leftarrow \lambda^{-1} P_{i-1} \\ \text{for } k = 1:N \\ \quad P_{k,i} = P_{k-1,i} - \frac{P_{k-1,i} u_{k,i}^* u_{k,i} P_{k-1,i}}{\gamma_k^{-1} + u_{k,i} P_{k-1,i} u_{k,i}^*} \\ \text{end} \\ P_i \leftarrow P_{N,i} \,. \end{array} \right. \tag{8}$$

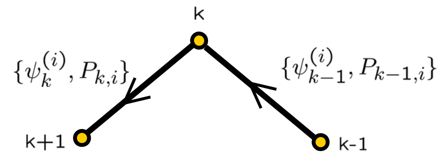


**Fig. 2**. The cooperation strategy of the exact distributed RLS algorithm (dRLS).

Now assume an incremental path is defined across the network cycling from node 1, to node 2, and so forth, until node $N$. Define the intermediate global matrices $\mathcal{Y}_{i-1}^k$ and $\mathcal{H}_{i-1}^k$ that collect the data blocks $\{\mathcal{Y}_{i-1}, \mathcal{H}_{i-1}\}$ in addition to the data collected along the network at time $i$ up to node $k$:

$$\mathcal{Y}_i^k = \begin{bmatrix} \mathcal{Y}_{i-1} \\ \hline d_1(i) \\ d_2(i) \\ \vdots \\ d_k(i) \end{bmatrix} \quad \text{and} \quad \mathcal{H}_i^k = \begin{bmatrix} \mathcal{H}_{i-1} \\ \hline u_{1,i} \\ u_{2,i} \\ \vdots \\ u_{k,i} \end{bmatrix} \tag{9}$$

Let $\psi_k^{(i)}$ be the solution to the following LS problem:

$$\min_\psi \left[\lambda^{i+1} \psi^* \Pi \psi + \left\|\mathcal{Y}_i^k - \mathcal{H}_i^k \psi\right\|^2_{\mathcal{W}_i^k}\right] \quad \Rightarrow \quad \psi_k^{(i)} \tag{10}$$

where

$$\mathcal{W}_i^k = \begin{bmatrix} \lambda \mathcal{W}_{i-1} & \mathbf{0} \\ \mathbf{0} & D_k \end{bmatrix}, \quad D_k \triangleq \operatorname{diag}\{\gamma_1, \cdots, \gamma_k\}. \tag{11}$$

Note that

$$\mathcal{W}_i^k = \begin{bmatrix} \mathcal{W}_i^{k-1} & \mathbf{0} \\ \mathbf{0} & \gamma_k \end{bmatrix} \tag{12}$$

Therefore, by using (9) and (12), a recursion to update $\psi_k^{(i)}$ in a distributed fashion over the network can be found as follows:

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

which leads to

$$\begin{aligned} \psi_k^{(i)} &= \psi_{k-1}^{(i)} \\ &+ \frac{P_{k-1,i}}{\gamma_k^{-1} + u_{k,i} P_{k-1,i} u_{k,i}^*} u_{k,i}^* \left(d_k(i) - u_{k,i} \psi_{k-1}^{(i)}\right). \end{aligned} \tag{13}$$

Grouping recursions (8) and (13) leads to an incrementally distributed and *exact* RLS (dRLS) solution to problem (3). Figure 2 illustrates the operation of algorithm, where each node shares with its successor node in the ring the quantities $\{\psi_k^{(i)}, P_{k,i}\}$:

$$\begin{cases} \psi_0^{(i)} \leftarrow w_{i-1}; \quad P_{0,i} \leftarrow \lambda^{-1} P_{i-1} \\ \text{for } k = 1 : N \\ \quad e_k(i) = d_k(i) - u_{k,i}\psi_{k-1}^{(i)} \\ \quad \psi_k^{(i)} = \psi_{k-1}^{(i)} + \frac{P_{k-1,i}}{\gamma_k^{-1} + u_{k,i}P_{k-1,i}u_{k,i}^*} u_{k,i}^* e_k(i) \\ \quad P_{k,i} = P_{k-1,i} - \frac{P_{k-1,i}u_{k,i}^* u_{k,i} P_{k-1,i}}{\gamma_k^{-1} + u_{k,i}P_{k-1,i}u_{k,i}^*} \\ \text{end} \\ w_i \leftarrow \psi_N^{(i)}; \quad P_i \leftarrow P_{N,i} \ . \end{cases} \tag{14}$$

## 4. LOW-COMMUNICATION DISTRIBUTED RLS ADAPTATION

The algorithm proposed in the previous section implements exact RLS distributively and it requires $O(M^2)$ transmission complexity. However, this cost can be prohibitive in some applications [3], which motivates the pursuit of an alternative implementation that requires less communications while keeping the performance close to the exact implementation. One proposition is to allow collaboration for the estimates while keeping the matrices $P_{k,i}$ evolving locally and independent from the neighbor nodes. This would lead to the following approximate algorithm:

$$\begin{cases} \psi_0^{(i)} \leftarrow w_{i-1}; \\ \text{for } k = 1 : N \\ \quad e_k(i) = d_k(i) - u_{k,i}\psi_{k-1}^{(i)} \\ \quad \psi_k^{(i)} = \psi_{k-1}^{(i)} + \frac{\lambda^{-1}P_{k,i-1}}{\gamma_k^{-1} + \lambda^{-1}u_{k,i}P_{k,i-1}u_{k,i}^*} u_{k,i}^* e_k(i) \\ \quad P_{k,i} = \lambda^{-1}\left[P_{k,i-1} - \frac{\lambda^{-1}P_{k,i-1}u_{k,i}^* u_{k,i} P_{k,i-1}}{\gamma_k^{-1} + \lambda^{-1}u_{k,i}P_{k,i-1}u_{k,i}^*}\right] \\ \text{end} \\ w_i \leftarrow \psi_N^{(i)}. \end{cases} \tag{15}$$

Algorithm (15) requires transmission complexity $O(M)$ only, as in [1]. Figure 3 describes the algorithm's collaboration strategy, in which estimates are shared along the path and matrices $P_{k,i}$ evolve locally. Therefore, the time forgetting factor $\lambda$ is assigned locally at the nodes, as matrices $P_{k,i}$ evolve independently. Obviously, decreasing the required information among the nodes no longer leads to an exact RLS implementation, and not surprisingly leads to some performance degradation.

## 5. MEAN-SQUARE PERFORMANCE ANALYSIS

We now study the mean-square performance of the low communications distributed RLS (LC-dRLS) algorithm (15). In order to proceed, we extend the space-time energy conservation approach of [1, 6] to treat the case. Due to space constraints, only the main steps are presented.

We assume that the data along the network are zero-mean and with known second-order moments. More specifically, we regard the data captured at node $k$ and time $i$ as the random variables $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$.

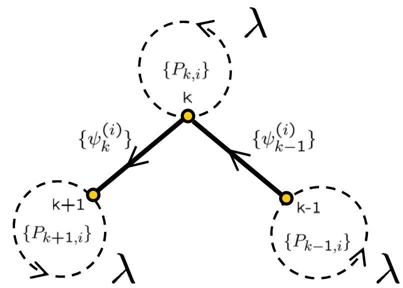


**Fig. 3**. The cooperation strategy of the low communications distributed RLS algorithm (LC-dRLS).

### 5.1. Data Model and Assumptions

The challenge implied by the spatial dimension requires us to rely on some data assumptions for the random variables $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$. The assumptions lead to a good match between analysis and simulations, as shown in the next sections. The data assumptions adopted are the following:

1. The unknown vector $w^o$ relates the data $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$ as
$$\boldsymbol{d}_k(i) = \boldsymbol{u}_{k,i}w^o + \boldsymbol{v}_k(i) \tag{16}$$
where $\boldsymbol{v}_k(i)$ is some temporally and spatially 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).

### 5.2. Weighted Energy Conservation Relation

For the algorithm (15), we define the error signals:

$$\widetilde{\boldsymbol{\psi}}_{k-1}^{(i)} = w^o - \boldsymbol{\psi}_{k-1}^{(i)}, \qquad \widetilde{\boldsymbol{\psi}}_{k}^{(i)} = w^o - \boldsymbol{\psi}_{k}^{(i)} \tag{17}$$

$$\boldsymbol{e}_{a,k}(i) = \boldsymbol{u}_{k,i}\widetilde{\boldsymbol{\psi}}_{k-1}^{(i)}, \qquad \boldsymbol{e}_{p,k}(i) = \boldsymbol{u}_{k,i}\widetilde{\boldsymbol{\psi}}_{k}^{(i)} \tag{18}$$

$$\boldsymbol{e}_k(i) = \boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}\boldsymbol{\psi}_{k-1}^{(i)} \tag{19}$$

where (17) are the weight-error vectors, (18) defines the a priori and a posteriori local errors, and (19) is the output error. Note that

$$\boldsymbol{e}_k(i) = \boldsymbol{e}_{a,k}(i) + \boldsymbol{v}_k(i) \tag{20}$$

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

$$\eta_k = E\|\widetilde{\boldsymbol{\psi}}_{k-1}^{(\infty)}\|^2 = E\|\widetilde{\boldsymbol{\psi}}_{k-1}^{(\infty)}\|_I^2 \quad \text{(MSD)} \tag{21}$$

$$\zeta_k = E|\boldsymbol{e}_{a,k}(\infty)|^2 = E\|\widetilde{\boldsymbol{\psi}}_{k-1}^{(\infty)}\|_{R_{u,k}}^2 \quad \text{(EMSE)} \tag{22}$$

$$\xi_k = E|\boldsymbol{e}_k(\infty)|^2 = \zeta_k + \sigma_{v,k}^2 \quad \text{(MSE)} \tag{23}$$

where a weighted norm notation is employed in (21) and (22). Furthermore, we introduce the *weighted a priori* and *a posteriori local error* signals for node $k$:

$$e_{a,k}^{\Gamma\Sigma}(i) = u_{k,i}\Gamma\Sigma\widetilde{\psi}_{k-1}^{(i)} \quad \text{and} \quad e_{p,k}^{\Gamma\Sigma}(i) = u_{k,i}\Gamma\Sigma\widetilde{\psi}_{k}^{(i)} \tag{24}$$

for Hermitian positive-definite matrices $\Sigma$ (at our choice) and $\Gamma = \Gamma[u_{l,i}; l \in N_k(i)]$ (data-dependent), where $N_k$ is the set of neighbor nodes of node $k$, including itself. In this work we analyze the partial information implementation, in which by inspecting (15) one chooses $\Gamma$ to be

$$\Gamma = \frac{\lambda^{-1}P_{k,i-1}}{\gamma_k^{-1} + \lambda^{-1}u_{k,i}P_{k,i-1}u_{k,i}^*} \,. \tag{25}$$

A *space-time energy relation* that relates the *local* error quantities

$$\left\{\widetilde{\psi}_{k-1}^{(i)}, \widetilde{\psi}_{k}^{(i)}, e_{a,k}^{\Gamma\Sigma}(i), e_{p,k}^{\Gamma\Sigma}(i)\right\} \tag{26}$$

can be found to be

$$\boxed{\|\widetilde{\psi}_k\|_\Sigma^2 + \frac{|e_{a,k}^{\Gamma\Sigma}|^2}{\|u_k\|_{\Gamma\Sigma\Gamma}^2} = \|\widetilde{\psi}_{k-1}\|_\Sigma^2 + \frac{|e_{p,k}^{\Gamma\Sigma}|^2}{\|u_k\|_{\Gamma\Sigma\Gamma}^2}} \tag{27}$$

Equation (27) is a *space-time* version of the weighted energy conservation relation in [6] in the context of matrix-valued data-normalized filters. The time index $i$ has been dropped for compactness.

### 5.3. 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.,

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

with a value $\varepsilon_k$ that is possibly different for each node $k$. Moreover, due to cooperation, the nodes are interconnected. This fact makes the analysis more challenging. For simplicity, in this work we assume that the $\{u_{k,i}\}$ arise from circular Gaussian distributions with covariance matrices $R_{u,k}$. Now, introduce the eigendecomposition $R_{u,k} = U_k\Lambda_k U_k^*$, where $U_k$ is unitary and $\Lambda_k$ is a diagonal matrix with the eigenvalues of $R_{u,k}$, and let $p_k = \widetilde{\psi}_k^{(\infty)}$ and $\overline{p}_k = U_k\widetilde{\psi}_k^{(\infty)}$ be the *a posteriori*[1] weight-error vector in steady-state and its transformed version, respectively. Furthermore, define $p_{k-1} = \widetilde{\psi}_{k-1}^{(\infty)}$ as the *a priori* weight-error vector in steady-state and let $\overline{p}_{k-1} = U_k\widetilde{\psi}_{k-1}^{(\infty)}$ be its transformed version. In steady-state, relations (27) lead to a set of $N$ coupled variance relations [6], which can be decoupled and solved for each individual node by introducing and exploiting local weighting matrices $\Sigma_k$ and $\overline{\Sigma}_k = U_k\Sigma_k U_k^*$. It can be shown that the solution to that set of equations is given by

$$\boxed{E\|\overline{p}_{k-1}\|^2_{(I-\Pi_{k,1})\overline{\sigma}_{k-1}} = a_k\overline{\sigma}_{k-1}} \tag{28}$$

[1]Here *a priori* and *a posteriori* have spatial connotations.

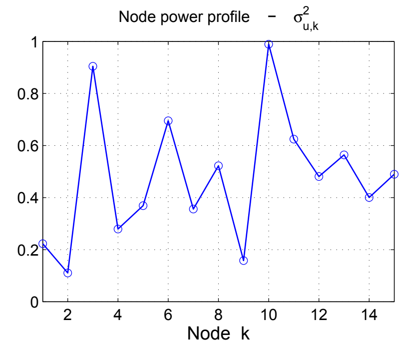


**Fig. 4**. Regressor statistics profile per node - power

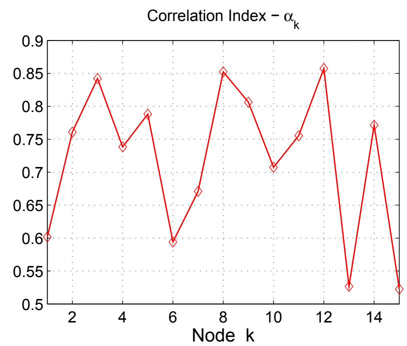


**Fig. 5**. Regressor statistics profile per node - correlation

for each node $k$ in the network, in terms of a compact weighting vector $\overline{\sigma}_k$ which we are free to choose [2] [1, 6]. In (28), $\Pi_{k,1}$ is a transition matrix along the collaboration ring for node $k$

$$\Pi_{k,\ell} \triangleq \overline{F}_{k+\ell-1}\overline{F}_{k+\ell}\cdots\overline{F}_N\overline{F}_1\cdots\overline{F}_{k-1} \,, \quad \ell = 1,\ldots,N \tag{29}$$

in terms of matrices $\overline{F}_k$ that capture the local regressor statistics

$$\overline{F}_k = (1 - 2\beta_k + \delta\beta_k^2)I + \beta_k^2 b_k c_k^T \tag{30}$$

with $\delta = 1$ for complex signals and $\delta = 2$ for real signals, $b_k = \text{diag}\{\Lambda_k\}$, $c_k = \text{diag}\{\Lambda_k^{-1}\}$, and $\beta_k$ given by

$$\beta_k = \begin{cases} \frac{1-\lambda}{\gamma_k^{-1}} \,, & \text{for } \lambda \to 1 \\ \frac{1-\lambda}{\gamma_k^{-1}\lambda + (1-\lambda)M} \,, & \text{for smaller } \lambda \,. \end{cases}$$

The row vector $a_k$ is given by

$$a_k \triangleq g_k\Pi_{k,2} + g_{k+1}\Pi_{k,3} + \cdots + g_{k-2}\Pi_{k,N} + g_{k-1} \tag{31}$$

with $g_k = \sigma_{v,k}^2\beta_k^2 c_k^T$ and $\sigma_{v,k}^2$ is the background noise power.

Now we resort to the weighting vectors $\overline{\sigma}_{k-1}$ in (28) to calculate the MSD, EMSE and MSE for each node. For the MSD, we

[2]Note that $\overline{\Sigma}_k = U_k\Sigma_k U_k^*$ is diagonal if we choose $\Sigma_k$ properly. Thus, we employ a compact notation in terms of the diagonal entries of $\overline{\Sigma}_k$, collected in the weighting vector $\overline{\sigma}_k$, i.e., $\overline{\Sigma}_k = \text{diag}\{\overline{\sigma}_k\} \to \overline{\sigma}_k$– see [1].

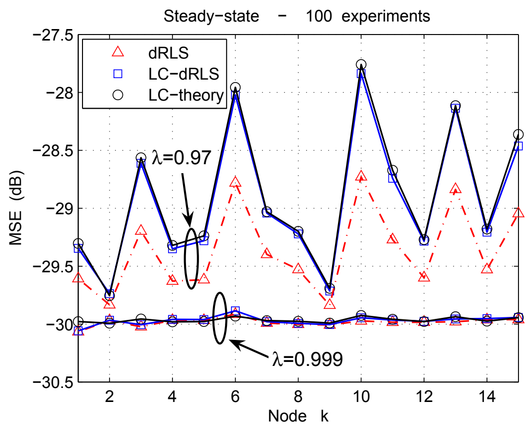


**Fig. 6**. MSE versus Node $k$ for $\lambda = 0.97$ and $\lambda = 0.999$.

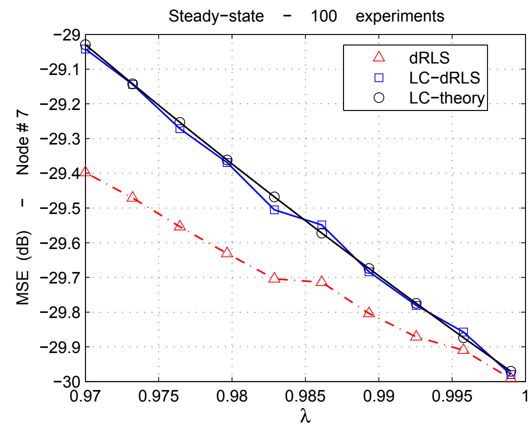


**Fig. 7**. MSE versus $\lambda$ for node 7.

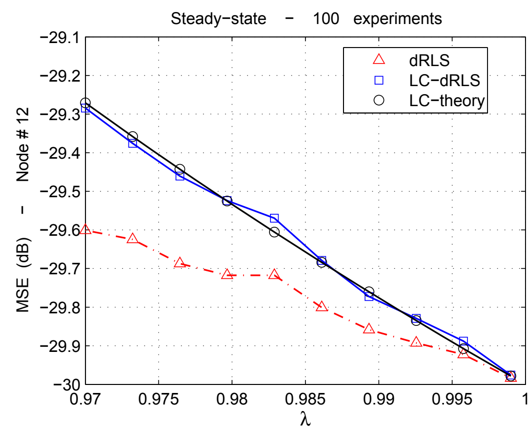


**Fig. 8**. MSE versus $\lambda$ for node 12.

select $\overline{\sigma}_{\eta,k-1} = (I - \Pi_{k,1})^{-1} q$, where $q = \mathrm{col}\{1, 1, \cdots, 1\}$. Then

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

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

$$\zeta_k = E\|\overline{\boldsymbol{p}}_{k-1}\|_{b_k}^2 = a_k \left(I - \Pi_{k,1}\right)^{-1} b_k \tag{33}$$

The MSE follows from (23) and the result above. Summarizing the results, the network mean-square performance in steady-state for every node $k$ is given by

$$\boxed{\begin{array}{lcll} \eta_k & = & a_k \left(I - \Pi_{k,1}\right)^{-1} q & \text{(MSD)} \\ \zeta_k & = & a_k \left(I - \Pi_{k,1}\right)^{-1} b_k & \text{(EMSE)} \\ \xi_k & = & \zeta_k + \sigma_{v,k}^2 & \text{(MSE)} \end{array}}$$

with $\Pi_{k,1}$ and $a_k$ given in (29) and (31).

## 6. SIMULATIONS

We consider a network with $N = 15$ nodes where each local filter has $M = 10$ taps. The system evolves for 10000 iterations and the results are averaged over 100 independent experiments. The steady-state 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 Figs. 4 and 5. The background noise $\boldsymbol{v}_k(i)$ has variance $\sigma_{v,k}^2 = 10^{-3}$ across the network. Figures 6-8 illustrate the effect of decreasing $\lambda$ on the network performance. As the forgetting factor gets closer to unity, both algorithms (13) and (15) lead the network to similar performance.

## 7. CONCLUSIONS AND FUTURE WORK

We proposed a distributed RLS implementation that delivers at every node the exact least-squares solution at lowered communication costs, if compared to existing schemes in the literature. We also proposed a simplified distributed RLS implementation that reduces communication among the nodes implementing collaboration of estimates only; and it tends to the performance of the exact implementation in the mean-square sense as the forgetting factor $\lambda$ approaches unity.

## 8. REFERENCES


[1] C. Lopes and A. H. Sayed, "Distributed adaptive incremental strategies: formulation and performance analysis," *Proc. ICASSP*, pp. 584-587, vol. 3, Toulouse, France, May 2006.

[2] C. G. Lopes and A. H. Sayed, "Distributed processing over adaptive networks," *Proc. Adaptive Sensor Array Processing Workshop*, MIT Lincoln Lab., Lexington, MA, June 2006.

[3] 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.

[4] 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.

[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.

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