Incremental Adaptive Strategies Over Distributed Networks

Cassio G. Lopes, *Student Member, IEEE*, and Ali H. Sayed, *Fellow, IEEE*

***Abstract*—An adaptive distributed strategy is developed based on incremental techniques. The proposed scheme addresses the problem of linear estimation in a cooperative fashion, in which nodes equipped with local computing abilities derive local estimates and share them with their predefined neighbors. The resulting algorithm is distributed, cooperative, and able to respond in real time to changes in the environment. Each node is allowed to communicate with its immediate neighbor in order to exploit the spatial dimension while limiting the communications burden at the same time. A spatial–temporal energy conservation argument is used to evaluate the steady-state performance of the individual nodes across the entire network. Computer simulations illustrate the results.**

***Index Terms*—Adaptive networks, consensus, cooperation, diffusion algorithm, distributed processing, incremental algorithm.**

I. Introduction

DISTRIBUTED processing deals with the extraction of information from data collected at nodes that are distributed over a geographic area. For example, each node in a network of nodes could collect noisy observations related to a certain parameter or phenomenon of interest. The nodes would then interact with their neighbors in a certain manner, as dictated by the network topology, in order to arrive at an estimate of the parameter or phenomenon of interest. The objective is to arrive at an estimate that is as accurate as the one that would be obtained if each node had access to the information across the entire network. In comparison, in a traditional centralized solution, the nodes in the network would collect observations and send them to a central location for processing. The central processor would then perform the required estimation tasks and broadcast the result back to the individual nodes. This mode of operation requires a powerful central processor, in addition to extensive amounts of communication between the nodes and the processor. In the distributed solution, the nodes rely solely on their local data and on interactions with their immediate neighbors. The amount of processing and communications is significantly reduced [1]–[3].

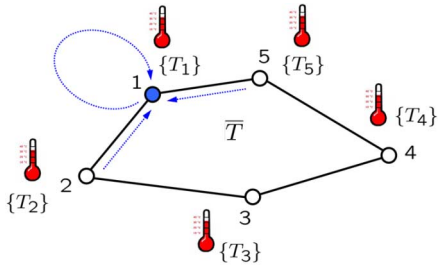


Fig. 1. Distributed network with N nodes accessing temperature data.

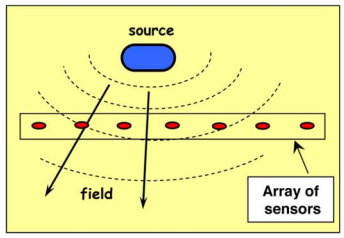


Fig. 2. Monitoring a diffusion phenomenon by a network of sensors deployed in the field.

A. Applications

Let us illustrate these ideas with an example. Consider a collection of N nodes spread over a geographic area, as shown in Fig. 1. Each node has access to a local temperature measurement T_i. The objective is to provide each node with information about the average temperature $\overline{T}$ across the network. In one distributed solution to this problem (known as a consensus implementation [4]–[6]), each node combines the measurements from its immediate neighbors (those that are connected to it). The result of the combination becomes this node's new measurement, i.e.,

$$x_1(i) \longleftarrow \alpha_1 x_1(i-1) + \alpha_2 x_2(i-1) + \alpha_5 x_5(i-1) \quad \text{(node 1)}$$

where $x_1(i)$ denotes the updated measurement of node 1 at iteration i, and the α's are appropriately chosen coefficients. Every other node in the network performs the same operation and the process is repeated. Under suitable conditions on the α's and network topology, all node measurements will converge asymptotically to the desired average temperature $\overline{T}$.

A more sophisticated application is to use measurements collected in time and space by a group of sensors in order to monitor the concentration of a chemical in the air or water (see Fig. 2).

Manuscript received February 27, 2006; revised November 28, 2006. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Jonathon Chambers. This material was based on work supported in part by the National Science Foundation under awards ECS-0401188 and ECS-0601266. The work of C. G. Lopes was also supported by a fellowship from CAPES, Brazil, under award 1168/01-0. A short version of this work appeared in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing* (ICASSP), Toulouse, France, May 2006, vol. 3, pp. 584–587.

The authors are with the Department of Electrical Engineering, University of California, Los Angeles, CA 90095 USA (e-mail: cassio@ee.ucla.edu; sayed@ee.ucla.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2007.896034

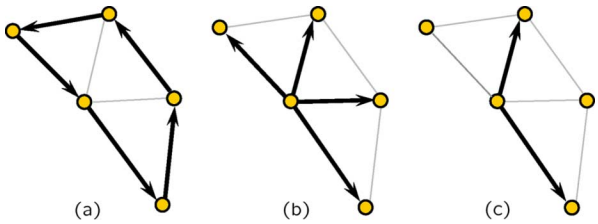


Fig. 3. Three modes of cooperation: (a) Incremental; (b) diffusion; and (c) probabilistic diffusion.

These measurements can then be used to estimate the parameters $\{\theta_1, \theta_2, \theta_3\}$ of the model that dictates the diffusion of the chemical in the environment according to some diffusion equation subject to boundary conditions, e.g.,

$$\frac{\partial c(x,t)}{\partial t} = \theta_1 \frac{\partial^2 c(x,t)}{\partial x^2} + \theta_2 \frac{\partial c(x,t)}{\partial x} + \theta_3 c(x,t) + u(x,t)$$

where $c(x,t)$ denotes the concentration at location x at time t [7]. Another application of distributed processing is monitoring a moving target in a region monitored by a collection of sensors [8]. The sensors would share their noisy measurements through local interactions in order to detect the presence of the target and track its trajectory.

Such 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], [8]–[10]. 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 [1].

B. Modes of Cooperation

Obviously, the effectiveness of any distributed implementation will depend on the modes of cooperation that are allowed among the nodes. Fig. 3 illustrates three such modes of cooperation.

In an incremental mode of cooperation, information flows in a sequential manner from one node to the adjacent node. This mode of operation requires a cyclic pattern of collaboration among the nodes, and it tends to require the least amount of communications and power [2], [11], [12]. In a diffusion implementation, on the other hand, each node communicates with all its neighbors as dictated by the network topology [13]–[15]. The amount of communication in this case is higher than in an incremental solution. Nevertheless, the nodes have access to more data from their neighbors. The communications in the diffusion implementation can be reduced by allowing each node to communicate only with a subset of its neighbors. The choice of which subset of neighbors to communicate with can be randomized according to some performance criterion. In this paper, we focus on the incremental mode of collaboration.

C. Consensus Strategy

The temperature example that we mentioned before is a special case of a more general strategy for distributed processing, known as consensus (e.g., [4]–[6], and [16]). Broadly, consensus implementations employ two time scales and they function as follows. Assume the network is interested in estimating a certain parameter. Each node collects observations over a period of time and reaches an individual decision about the parameter. During this time, there is limited interaction among the nodes; the nodes act more like individual agents. Following this initial stage, the nodes then combine their estimates through several consensus iterations; under suitable conditions, the estimates generally converge asymptotically to the desired (global) estimate of the parameter.

Let us consider another example of a consensus implementation, which will serve as further motivation for the contributions in this work. Consider again a collection of nodes. Each node has access to a data vector y_k and a data matrix H_k. The y_k are noisy and distorted measurements of some unknown vector w^o, as follows:

$$y_k = H_k w^o + v_k.$$

Each node can evaluate the least-squares estimate of w^o based on its own local data $\{y_k, H_k\}$. To do so, each node evaluates its local cross-correlation vector $\theta_k = H_k^* y_k$ and its autocorrelation matrix $R_k = H_k^* H_k$. Then, the local estimate of w^o can be found from $\hat{w}_k = R_k^{-1}\theta_k$. This operation requires that each node collects sufficient data into y_k and H_k. Once the local quantities $\{\theta_k, R_k\}$ have been evaluated at the individual nodes, one can apply consensus iterations at the nodes to determine $\hat{R}$ and $\hat{\theta}$, which are estimates of the overall (mean) quantities R and θ defined by [13], as follows:

$$R = \frac{1}{N}\sum_{k=1}^{N} R_k \quad \text{and} \quad \theta = \frac{1}{N}\sum_{k=1}^{N} \theta_k.$$

A global estimate of w^o is given by $\hat{w} = \hat{R}^{-1}\hat{\theta}$. For all practical purposes, a least-squares implementation in this manner is an offline or nonrecursive solution. For example, if a particular node collects one more entry in y_k and one more row in H_k, a difficulty that occurs is how to update the current solution $\hat{w}$ to account for new data without having to repeat prior processing and iterations afresh. In addition, the offline averaging limits the ability of consensus-based solutions to track fast-changing environments, especially in networks with limited communication resources.

D. Contributions

To address the aforementioned issues (need for adaptive implementations, real-time operation, and low computational and communications complexity), we propose a distributed least-mean-squares (LMS)-like algorithm that requires less complexity for both communications and computations and inherits the robustness of LMS implementations [17]. The proposed solution promptly responds to new data, as the information flows through the network. It does not require intermediate averaging as in consensus implementations; it

neither requires two separate time scales. The distributed adaptive solution is an extension of adaptive filters and can be implemented without requiring any direct knowledge of data statistics; in other words, it is model independent. While we focus in this paper on LMS-type updates for simplicity, the same ideas and techniques apply to other types of adaptation rules.

Our objective is therefore to develop distributed algorithms that enable a network of nodes to function as an adaptive entity in its own right. Thus, recall that a regular adaptive filter responds in real time to its data and to variations in the statistical properties of this data. We want to extend this ability to the network domain [12], [18], [19]. Specifically, the purpose of this paper is threefold:

1) to motivate a family of *incremental adaptive algorithms* for distributed estimation inspired by distributed optimization techniques [2], [20], [21];
2) to use the incremental algorithms to propose an adaptive network structure composed of an interconnected set of nodes that is able to respond to data in real time and to track variations in the statistical properties of the data as follows:
   a) each time a node receives a new piece of information, this information is readily used by the node to update its local estimate of the parameter of interest;
   b) the local estimates of the parameter are shared with the immediate neighbors of the node in a process that allows the information to flow to other nodes in the network;
3) to analyze the performance of the resulting interconnected network of nodes. This task is challenging since an adaptive network comprises a "system of systems" that processes data cooperatively in both time and space. Different nodes will converge to different mean-square-error (MSE) levels, reflecting the statistical diversity of the data and the different noise levels.

In summary, we propose an incremental adaptive algorithm over ring topologies and derive closed form expressions for its mean-square performance.

E. Notation and Paper Organization

In this paper, we need to distinguish not only between vectors and matrices, but also between random and nonrandom quantities (see, e.g., [17]). Thus, we adopt boldface letters for random quantities and normal font for nonrandom (deterministic) quantities. We also use capital letters for matrices and small letters for vectors. For example, $\boldsymbol{d}$ is a random observation quantity, and d is a realization or measurement for it, and R is a covariance matrix while w is a weight vector. The notation * is used to denote complex conjugation for scalars and complex-conjugate transposition for matrices.

The paper is organized as follows. In Section II, a distributed estimation problem is formulated and a framework for distributed adaptive processing is described, with the subsequent derivation of a distributed incremental LMS algorithm. In Section III, the performance of the temporal and spatial adaptive strategy is studied, providing closed-form expressions for the mean-square behavior of the distributed algorithm. The theoretical results are compared with simulations in Section IV. Section V points out future extensions that are currently being developed.

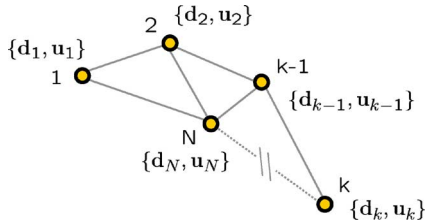


Fig. 4. Distributed network with N active nodes accessing space–time data.

II. Estimation Problem and the Adaptive Distributed Solution

There have been extensive works in the literature on incremental methods for solving distributed optimization problems (e.g., [2], [11], [20], [22], and [23]). It is known that whenever a cost function can be decoupled into a sum of individual cost functions, a distributed algorithm can be developed for minimizing the cost function through an incremental procedure. We explain the procedure as follows in the context of MSE estimation.

Consider a network with N nodes (see Fig. 4). Each node k has access to time realizations $\{d_k(i), u_{k,i}\}$ of zero-mean spatial data $\{\boldsymbol{d}_k, \boldsymbol{u}_k\}$, $k = 1, \ldots, N$, where each $\boldsymbol{d}_k$ is a scalar measurement and each $\boldsymbol{u}_k$ is a $1 \times M$ row regression vector. We collect the regression and measurement data into two global matrices, as follows:

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

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

These quantities collect the data across all N nodes. The objective is to estimate the $M \times 1$ vector w that solves

$$\min_{w} \quad J(w) \tag{3}$$

where the cost function $J(w)$ denotes the MSE, as follows:

$$J(w) = E\|\boldsymbol{d} - \boldsymbol{U}w\|^2 \tag{4}$$

and E is the expectation operator. The optimal solution w^o of (3) satisfies the orthogonality condition [17]

$$E\boldsymbol{U}^*(\boldsymbol{d} - \boldsymbol{U}w^o) = 0 \tag{5}$$

so that w^o is the solution to the *normal equations*

$$R_{du} = R_u w^o \tag{6}$$

which are defined in terms of the correlation and cross-correlation quantities

$$R_u = E\boldsymbol{U}^*\boldsymbol{U} \quad (M \times M), \qquad R_{du} = E\boldsymbol{U}^*\boldsymbol{d} \quad (M \times 1). \tag{7}$$

If the optimal solution w^o were to be computed from (6), then *every node* in the network would need to have access to the global statistical information $\{R_u, R_{du}\}$. Alternatively, the solution w^o could be computed centrally and the result broadcast

to all nodes. Either way, these approaches drain considerable communications and computational resources and they do not endow the network with the necessary adaptivity to cope with possible changes in the statistical properties of the data.

We shall instead develop and study a *distributed* solution that allows cooperation among the nodes through limited local communications, while at the same time equipping the network with an adaptive mechanism [12]. Specifically, in this paper we focus on the incremental mode of cooperation, where the estimation task is distributed among the nodes and each node is allowed to cooperate only with one of its direct neighbors at a time. The single-neighbor case is already challenging in its own right, and the analysis will bring forth several interesting observations. Extensions to other modes of cooperation are possible [15], [18], [19].

A. Steepest-Descent Solution

To arrive at the adaptive distributed solution we first review the steepest-descent solution and its incremental implementation. To begin with, we note from (4) and (7) that the cost function $J(w)$ can be decomposed as

$$J(w) = \sum_{k=1}^{N} J_k(w) \tag{8}$$

where each $J_k(w)$ is given by

$$\begin{aligned} J_k(w) &\triangleq E|\boldsymbol{d}_k - \boldsymbol{u}_k w|^2 && (9)\\ &= \sigma_{d,k}^2 - R_{ud,k}w - w^* R_{du,k} + w^* R_{u,k} w && (10)\end{aligned}$$

and the second-order moment quantities are defined by

$$\sigma_{d,k}^2 = E|\boldsymbol{d}_k|^2, \quad R_{u,k} = E\boldsymbol{u}_k^*\boldsymbol{u}_k, \quad \text{and} \quad R_{du,k} = E\boldsymbol{d}_k\boldsymbol{u}_k^*. \tag{11}$$

In other words, $J(w)$ can be expressed as the sum of N individual cost functions $J_k(w)$, one for each node k. Thus, the traditional iterative steepest-descent solution for determining w^o can be expressed in the form

$$\begin{aligned} w_i &= w_{i-1} - \mu\,[\nabla J(w_{i-1})]^*, \quad w_{-1} = \text{ initial condition}\\ &= w_{i-1} - \mu \sum_{k=1}^{N} [\nabla J_k(w_{i-1})]^*\\ &= w_{i-1} + \mu \sum_{k=1}^{N} (R_{du,k} - R_{u,k} w_{i-1}) \end{aligned} \tag{12}$$

where $\mu > 0$ is a suitably chosen positive step-size parameter, w_i is an estimate for w^o at iteration i, and $\nabla J(w_{i-1})$ denotes the gradient vector of $J(w)$ with respect to w evaluated at w_{i-1}. For μ sufficiently small, we will have $w_i \to w^o$ as $i \to \infty$ for any initial condition. An equivalent implementation can be motivated as follows.

Let us define a *cycle* visiting every node over the network topology only once such that each node has access only to its immediate neighbor node in this cycle [2], [11], [21]. Let $\psi_k^{(i)}$ denote a *local estimate* of w^o at node k at time i. Thus, assume that node k has access to $\psi_{k-1}^{(i)}$, which is an estimate of w^o at its immediate neighbor node $k-1$ in the defined cycle (see Fig. 5). If at each time instant i we start with the initial condition $\psi_0^{(i)} = w_{i-1}$ at node 1 (i.e., with the current global estimate w_{i-1} for w^o), and iterate cyclicly across the nodes then, at the end of the procedure, the local estimate at node N will coincide with w_i from (12), i.e., $\psi_N^{(i)} = w_i$. In other words, the following implementation is equivalent to (12):

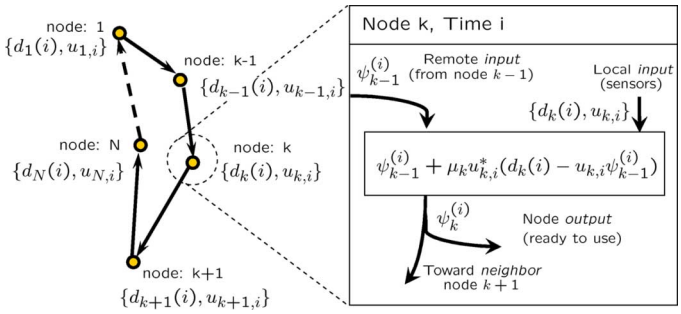


Fig. 5. Data processing in the proposed adaptive distributed structure.

$$\begin{cases} \psi_0^{(i)} = w_{i-1} \\ \psi_k^{(i)} = \psi_{k-1}^{(i)} - \mu_k\,[\nabla J_k(w_{i-1})]^*, \quad k = 1, \ldots, N \\ w_i = \psi_N^{(i)}. \end{cases} \tag{13}$$

Observe that in this steepest-descent implementation, the iteration for $\psi_k^{(i)}$ is over the spatial index k.

B. Incremental Steepest-Descent Solution

Although recursion (13) is cooperative in nature, with each node k using information from its immediate neighbor (represented by $\psi_{k-1}^{(i)}$), this implementation still requires the nodes to have access to the global information w_{i-1} in order to evaluate $\nabla J_k(w_{i-1})$. This fact undermines our stated objective of a fully distributed solution.

In order to resolve this difficulty, we call upon the concept of incremental gradient algorithms [11], [20], [21]. If each node evaluates the required partial gradient $\nabla J_k(\cdot)$ at the local estimate $\psi_{k-1}^{(i)}$ received from node $k-1$, as opposed to w_{i-1}, then an incremental version of algorithm (13) would result, namely

$$\begin{cases} \psi_0^{(i)} = w_{i-1} \\ \psi_k^{(i)} = \psi_{k-1}^{(i)} - \mu_k \left[\nabla J_k\left(\psi_{k-1}^{(i)}\right)\right]^*, \quad k = 1, \ldots, N \\ w_i = \psi_N^{(i)}. \end{cases} \tag{14}$$

This cooperative scheme relies only on locally available information, leading to a truly distributed solution. The scheme requires each node to communicate *only* with its immediate neighbor, thus saving on communication and energy resources [2], [11].

C. Incremental Adaptive Solution

The incremental solution (14) relies on knowledge of the second-order moments $R_{du,k}$ and $R_{u,k}$, which are needed to evaluate the local gradients ∇J_k. An adaptive implementation of (14) can be obtained by replacing the second-order moments

$\{R_{du,k}, R_{u,k}\}$ by instantaneous approximations, say of the LMS type, as follows[1]:

$$R_{du,k} \approx d_k(i)u_{k,i}^* \quad R_{u,k} \approx u_{k,i}^* u_{k,i} \tag{15}$$

by using data realizations $\{d_k(i), u_{k,i}\}$ at time i. The approximations (15) lead to an adaptive distributed incremental algorithm, or simply a *distributed incremental LMS* algorithm of the following form:

For each time $i \geq 0$, repeat:

$$k = 1, \ldots, n$$

$$\begin{cases} \psi_0^{(i)} = w_{i-1} \\ \psi_k^{(i)} = \psi_{k-1}^{(i)} + \mu_k u_{k,i}^* \left(d_k(i) - u_{k,i}\psi_{k-1}^{(i)} \right), \quad k = 1, \ldots, n \\ w_i = \psi_n^{(i)} \end{cases} \tag{16}$$

The operation of algorithm (16) is illustrated in Fig. 5. At each time instant i, each node uses local data realizations $\{d_k(i), u_{k,i}\}$ and the weight estimate $\psi_{k-1}^{(i)}$ received from its adjacent node to perform the following three tasks:

1) evaluate a local error quantity: $e_k(i) = d_k(i) - u_{k,i}\psi_{k-1}^{(i)}$;
2) update its weight estimate: $\psi_k^{(i)} = \psi_{k-1}^{(i)} + \mu_k u_{k,i}^* e_k(i)$;
3) pass the updated weight estimate $\psi_k^{(i)}$ to its neighbor node $k+1$.

This distributed incremental adaptive implementation generally has better steady-state performance and convergence rate than a nondistributed implementation that is based on using w_{i-1} in place of $\psi_{k-1}^{(i)}$ in the expression for $e_k(i)$ (see Appendix A), i.e.,

$$\psi_k^{(i)} = \psi_{k-1}^{(i)} + \mu_k u_{k,i}^* \left(d_k(i) - u_{k,i} w_{i-1} \right), \quad k = 1, \ldots, N. \tag{17}$$

This is a reflection of a result in optimization theory that the incremental strategy (14) can outperform the steepest-descent technique (13), [20], [21]. Intuitively, this is because the incremental solution incorporates local information on-the-fly into the operation of the algorithm, i.e., it exploits the spatial diversity more fully. While the steepest-descent solution (13) has w_{i-1} fixed throughout all $k = 1, 2, \ldots, N$ spatial updates, the incremental solution (14) uses instead the successive updates $\{\psi_0^{(i)}, \psi_1^{(i)}, \ldots, \psi_{N-1}^{(i)}\}$. More detailed comparisons between both implementations are provided in Appendix A.

In order to illustrate these observations, we run a simulation comparing the excess mean-square error (EMSE) of the incremental adaptive algorithm (16) and the stochastic implementation (17) at node 1, i.e., $E|\boldsymbol{u}_{k,i}(\boldsymbol{\psi}_1^{(i)} - w^o)|^2$—see Figs. 6 and 7. The network has $N = 20$ nodes pursuing the same unknown vector $w^o = \text{col}\{1, 1, \cdots, 1\}/\sqrt{M}$, with $M = 10$, and relying on independent Gaussian regressors with $R_{u,k} = I$. The background noise is white and Gaussian with $\sigma_v^2 = 10^{-3}$, and we assume the data are related via $d_k(i) = u_{k,i}w^o + v_k(i)$ for each node. The curves are obtained by averaging over 500 experiments. Fig. 6 shows the transient EMSE performance for both

[1]Other approximations are possible and they lead to alternative adaptation rules [17].

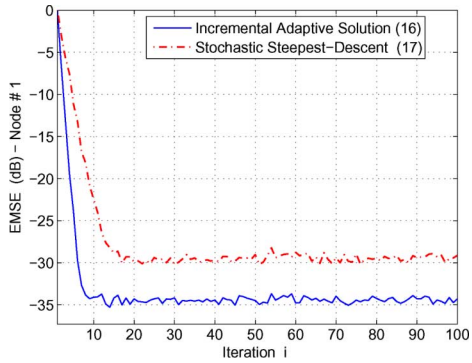


Fig. 6. Transient EMSE performance at node 1 for both incremental adaptive solution (16) and stochastic steepest-descent solution (17).

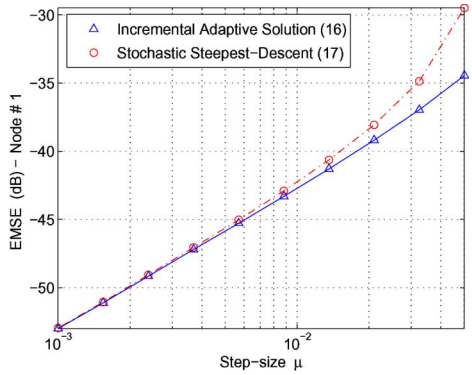


Fig. 7. Steady-state EMSE performance at node 1 as $\mu \to 0$ for both the incremental adaptive solution (16) and the stochastic steepest-descent solution (17).

algorithms, whereas Fig. 7 plots the steady-state values for decreasing step sizes, obtained by averaging the last 1000 samples after convergence.

In terms of complexity, the incremental solution (16) requires $O(M)$ computations per node and also $O(M)$ scalar transmissions per node. As such, the algorithm is intrinsically simple and is especially suitable for networks with low-energy resources.

III. Performance Analysis

An important question now is, how well does the adaptive incremental solution (16) perform? That is, how close does each $\psi_k^{(i)}$ (local estimate at node k) get to the desired solution w^o as time evolves? Studying the performance of such an interconnected network of nodes is challenging (more so than studying the performance of a single LMS filter) for the following reasons:

1) each node k is influenced by local data with local statistics $\{R_{du,k}, R_{u,k}\}$ (spatial information);
2) each node k is influenced by its neighbors through the incremental mode of cooperation (spatial interaction);
3) each node is subject to local noise with variance $\sigma_{v,k}^2$ (spatial noise profile).

In the next section, we provide a framework for studying the performance of such network by examining the flow of energy through the network both in time and space. For instance, we shall derive expressions that measure for each node k the steady-state values $E\|\boldsymbol{\psi}_k^{(i)} - w^o\|^2$ and $E|\boldsymbol{e}_k(i)|^2$ as $i \to \infty$. It will be shown that despite the quite simple cooperation strategy adopted, in steady-state each individual node is affected by the

whole network, with some emphasis given to local statistics. Furthermore, as the step size is decreased asymptotically, both quantities [mean-square deviation (MSD) and EMSE] approach zero for every node in the network, which also drives the MSE for every node asymptotically to the background noise level $\sigma_{v,k}^2$.

In order to pursue the performance analysis, we shall rely on the energy conservation approach of [17]. This energy-based approach needs to be extended to account for the space dimension because the distributed adaptive algorithm (16) involves both a time variable i and a space variable k. Moreover, we need to deal with the energy flow across interconnected filters and since each node in the network can now stabilize at an individual MSE value, some of the simplifications that were possible in the single node case [17] cannot be applied here. For example, due to cooperation, the performance of every node ends up depending on the whole network, an effect which we shall capture by a set of coupled equations. In order to evaluate individual node performance, weighting will be used to decouple the equations and to evaluate the quantities of interest in steady state.

The main result of this section is Theorem 1, further ahead, which provides closed-form expressions for the performance of each node in the network in terms of the so-called MSE, MSD, and EMSE measures defined next.

A. Data Model and Assumptions

To carry out the performance analysis, we first need to assume a model for the data as is commonly done in the literature of adaptive algorithms. As indicated earlier, we denote random variables by boldface letters. Thus, $\{d_k(i), u_{k,i}\}$ are realizations of the random quantities $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$. The subsequent analysis assumes the following data model for $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$:

A1) the desired unknown vector w^o relates $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$ as

$$\boldsymbol{d}_k(i) = \boldsymbol{u}_{k,i} w^o + \boldsymbol{v}_k(i) \tag{18}$$

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;

A2) $\boldsymbol{u}_{k,i}$ is independent of $\boldsymbol{u}_{l,i}$ for $k \neq l$ (spatial independence);

A3) $\boldsymbol{u}_{k,i}$ is independent of $\boldsymbol{u}_{k,j}$ for $i \neq j$ (time independence).

Linear models of the form (18) arise in several applications [7], [13], [17], [24], [25]. The model (18) assumes that the network is attempting to estimate an unknown vector w^o. This is often referred to as the stationary model. It captures the space dimension by assigning different signals $\{u_{k,i}, d_k(i)\}$ at different nodes k. The time dimension is accounted for by sequentially observing the temporal evolution of such signals. The signals $\{u_{k,i}, d_k(i)\}$ can be regarded simply as observations and measurements, collected locally by the nodes or, for example, as excitation/response pairs. The subsequent analysis can be extended to handle nonstationary models where w^o also varies with time ([17], [26]). For space limitations, we study here only the stationary case. Nevertheless, the distributed adaptive solution (16) also applies to nonstationary scenarios. Moreover, as is common in the study of traditional adaptive filters, we are assuming that the regressors are spatially and temporally independent to simplify the analysis. However, the distributed adaptive solution applies regardless of the assumptions, although the analysis would be far more demanding.

B. Weighted Energy Conservation Relation

To proceed, we define the following local error signals at each node k:

$$\widetilde{\boldsymbol{\psi}}_k^{(i)} \triangleq w^o - \boldsymbol{\psi}_k^{(i)} \ \text{(weight error vector at time } i) \tag{19}$$

$$\boldsymbol{e}_{a,k}(i) \triangleq \boldsymbol{u}_{k,i} \widetilde{\boldsymbol{\psi}}_{k-1}^{(i)} \ (a\ priori\ \text{error}) \tag{20}$$

$$\boldsymbol{e}_{p,k}(i) \triangleq \boldsymbol{u}_{k,i} \widetilde{\boldsymbol{\psi}}_{k}^{(i)} \ (a\ posteriori\ \text{error}) \tag{21}$$

$$\boldsymbol{e}_k(i) \triangleq \boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i} \boldsymbol{\psi}_{k-1}^{(i)} \ \text{(output error)}. \tag{22}$$

It should be noted that we are using traditional terminology (e.g., as in [17]), except that now the terms "*a priori*" and "*a posteriori*" have a *spatial* connotation as opposed to temporal connotation. The vector $\widetilde{\boldsymbol{\psi}}_k^{(i)}$ measures the difference between the weight estimate at node k and the desired solution w^o. The signal $\boldsymbol{e}_k(i)$ measures the estimation error in approximating $\boldsymbol{d}_k(i)$ by using information available locally, i.e., $\boldsymbol{u}_{k,i}\boldsymbol{\psi}_{k-1}^{(i)}$. If $\boldsymbol{\psi}_k^{(i)}$ were to converge to w^o, then by (18) we would expect $\boldsymbol{e}_k(i) \to \boldsymbol{v}_k(i)$, so that the variance of $\boldsymbol{e}_k(i)$ would tend to $\sigma_{v,k}^2$. Thus, by evaluating how close $E|\boldsymbol{e}_k(i)|^2$ gets to $\sigma_{v,k}^2$, we can estimate the performance of node k. Note that the error $\boldsymbol{e}_k(i)$ can be related to the *a priori* error $\boldsymbol{e}_{a,k}(i)$ by using the data model (18) as

$$\begin{aligned}\boldsymbol{e}_k(i) &= \boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}\boldsymbol{\psi}_{k-1}^{(i)} = \boldsymbol{u}_{k,i} w^o + \boldsymbol{v}_k(i) - \boldsymbol{u}_{k,i}\boldsymbol{\psi}_{k-1}^{(i)} \\ &= \boldsymbol{e}_{a,k}(i) + \boldsymbol{v}_k(i).\end{aligned} \tag{23}$$

Hence, $E|\boldsymbol{e}_k(i)|^2 = E|\boldsymbol{e}_{a,k}(i)|^2 + \sigma_{v,k}^2$, so that evaluating $E|\boldsymbol{e}_{a,k}(i)|^2$ is useful for evaluating $E|\boldsymbol{e}_k(i)|^2$.

We are interested in evaluating the MSD, the MSE, and the EMSE in steady state for every node k. These quantities are defined as follows:

$$\eta_k \triangleq E\|\widetilde{\boldsymbol{\psi}}_{k-1}^{(\infty)}\|^2 \quad \text{(MSD)} \tag{24}$$

$$\zeta_k \triangleq E\,|\boldsymbol{e}_{a,k}(\infty)|^2 \quad \text{(EMSE)} \tag{25}$$

$$\xi_k \triangleq E\,|\boldsymbol{e}_k(\infty)|^2 = \zeta_k + \sigma_{v,k}^2 \quad \text{(MSE)}. \tag{26}$$

In order to arrive at expressions for these quantities, we shall find it useful to resort to weighted norms as we now explain. Introduce the *weighted norm* notation $\|x\|_\Sigma^2 \triangleq x^*\Sigma x$ for a vector x and a Hermitian positive definite matrix $\Sigma > 0$. Then, under the assumed data conditions we have that

$$\eta_k = E\|\widetilde{\boldsymbol{\psi}}_{k-1}^{(\infty)}\|_I^2, \quad \zeta_k = E\|\widetilde{\boldsymbol{\psi}}_{k-1}^{(\infty)}\|_{R_{u,k}}^2. \tag{27}$$

In other words, we need to evaluate the means of two weighted norms of $\widetilde{\boldsymbol{\psi}}_{k-1}^{(\infty)}$ in (27). To do so, we shall first establish that there is a fundamental spatio–temporal energy balance relating the error variables (19)–(21).

We start by defining weighted *a priori* and *a posteriori* local error signals for each node k as follows:

$$\boxed{\boldsymbol{e}_{a,k}^{\Sigma}(i) \triangleq \boldsymbol{u}_{k,i}\Sigma\widetilde{\boldsymbol{\psi}}_{k-1}^{(i)} \quad \text{and} \quad \boldsymbol{e}_{p,k}^{\Sigma}(i) \triangleq \boldsymbol{u}_{k,i}\Sigma\widetilde{\boldsymbol{\psi}}_{k}^{(i)}} \tag{28}$$

for some Hermitian positive-definite matrix Σ that we are free to choose. As we shall see later, different choices for Σ allow us to evaluate and examine different performance measures [17], [27]. We now seek an energy relation that compares the norms of the following error quantities:

$$\left\{\widetilde{\boldsymbol{\psi}}_{k-1}^{(i)}, \widetilde{\boldsymbol{\psi}}_{k}^{(i)}, \boldsymbol{e}_{a,k}^{\Sigma}(i), \boldsymbol{e}_{p,k}^{\Sigma}(i)\right\}. \tag{29}$$

Using algorithm (16) and subtracting w^o from both sides gives

$$\widetilde{\boldsymbol{\psi}}_{k}^{(i)} = \widetilde{\boldsymbol{\psi}}_{k-1}^{(i)} - \mu_k \boldsymbol{u}_{k,i}^{*}\boldsymbol{e}_k(i). \tag{30}$$

Multiplying the previous equation from the left by $\boldsymbol{u}_{k,i}\Sigma$ gives

$$\boldsymbol{u}_{k,i}\Sigma\widetilde{\boldsymbol{\psi}}_{k}^{(i)} = \boldsymbol{u}_{k,i}\Sigma\widetilde{\boldsymbol{\psi}}_{k-1}^{(i)} - \mu_k\|\boldsymbol{u}_{k,i}\|_{\Sigma}^{2}\boldsymbol{e}_k(i) \tag{31}$$

so that from the definitions (28)

$$\boldsymbol{e}_{p,k}^{\Sigma}(i) = \boldsymbol{e}_{a,k}^{\Sigma}(i) - \mu_k\|\boldsymbol{u}_{k,i}\|_{\Sigma}^{2}\boldsymbol{e}_k \tag{32}$$

and, subsequently

$$\boldsymbol{e}_k(i) = \frac{1}{\mu_k}\frac{\left(\boldsymbol{e}_{a,k}^{\Sigma}(i) - \boldsymbol{e}_{p,k}^{\Sigma}(i)\right)}{\|\boldsymbol{u}_{k,i}\|_{\Sigma}^{2}}. \tag{33}$$

Substituting (33) into (30) and rearranging terms, we get

$$\widetilde{\boldsymbol{\psi}}_{k}^{(i)} + \frac{\boldsymbol{u}_{k,i}^{*}\boldsymbol{e}_{a,k}^{\Sigma}(i)}{\|\boldsymbol{u}_{k,i}\|_{\Sigma}^{2}} = \widetilde{\boldsymbol{\psi}}_{k-1}^{(i)} + \frac{\boldsymbol{u}_{k,i}^{*}\boldsymbol{e}_{p,k}^{\Sigma}(i)}{\|\boldsymbol{u}_{k,i}\|_{\Sigma}^{2}}. \tag{34}$$

Equating the weighted norm of both sides of (34), we find that the cross terms cancel out, and we end up with only energy terms, i.e.,

$$\boxed{\|\widetilde{\boldsymbol{\psi}}_{k}^{(i)}\|_{\Sigma}^{2} + \frac{\left|\boldsymbol{e}_{a,k}^{\Sigma}(i)\right|^2}{\|\boldsymbol{u}_{k,i}\|_{\Sigma}^{2}} = \|\widetilde{\boldsymbol{\psi}}_{k-1}^{(i)}\|_{\Sigma}^{2} + \frac{\left|\boldsymbol{e}_{p,k}^{\Sigma}(i)\right|^2}{\|\boldsymbol{u}_{k,i}\|_{\Sigma}^{2}}}. \tag{35}$$

Equation (35) is a *space–time* version of the weighted energy conservation relation developed in [17] in the context of regular adaptive implementations. This is an exact relation that shows how the energies of several error variables are related to each other in space and time. No approximations are used to derive (35).

C. Variance Relation

We now proceed to show how the energy conservation relation can be used to evaluate the performance of each individual node. First, we drop the time index i for compactness of notation. Then, we transform the relation (35) into a recursion of $\|\widetilde{\boldsymbol{\psi}}_{k}^{(i)}\|_{\Sigma}^{2}$ by substituting (32) into (35) and rearranging terms, as follows:

$$\|\widetilde{\boldsymbol{\psi}}_{k}\|_{\Sigma}^{2} = \|\widetilde{\boldsymbol{\psi}}_{k-1}\|_{\Sigma}^{2} - \mu_k\boldsymbol{e}_{a,k}^{\Sigma *}\boldsymbol{e}_k - \mu_k\boldsymbol{e}_k^{*}\boldsymbol{e}_{a,k}^{\Sigma} + \mu_k^2\|\boldsymbol{u}_k\|_{\Sigma}^{2}\cdot|\boldsymbol{e}_k|^2. \tag{36}$$

Using (23) and taking expectations of both sides leads to

$$\begin{aligned}E\|\widetilde{\boldsymbol{\psi}}_{k}\|_{\Sigma}^{2} = {} & E\|\widetilde{\boldsymbol{\psi}}_{k-1}\|_{\Sigma}^{2} - \mu_k E\boldsymbol{e}_{a,k}^{\Sigma *}\boldsymbol{e}_{a,k} - \mu_k E\boldsymbol{e}_{a,k}^{\Sigma}\boldsymbol{e}_{a,k}^{*} \\ & + \mu_k^2\sigma_{v,k}^{2}E\|\boldsymbol{u}_k\|_{\Sigma}^{2} + \mu_k^2 E\|\boldsymbol{u}_k\|_{\Sigma}^{2}\cdot|\boldsymbol{e}_{a,k}|^2.\end{aligned} \tag{37}$$

Using again the weighted error definitions (28), we can expand (37) in terms of weighted error vectors and the regressor data as follows:

$$\begin{aligned}E\|\widetilde{\boldsymbol{\psi}}_{k}\|_{\Sigma}^{2} = {} & E\|\widetilde{\boldsymbol{\psi}}_{k-1}\|_{\Sigma}^{2} - \mu_k E\widetilde{\boldsymbol{\psi}}_{k-1}^{*}\Sigma\boldsymbol{u}_k^{*}\boldsymbol{u}_k\widetilde{\boldsymbol{\psi}}_{k-1} \\ & - \mu_k E\widetilde{\boldsymbol{\psi}}_{k-1}^{*}\boldsymbol{u}_k^{*}\boldsymbol{u}_k\Sigma\widetilde{\boldsymbol{\psi}}_{k-1} \\ & + \mu_k^2 E\widetilde{\boldsymbol{\psi}}_{k-1}^{*}\boldsymbol{u}_k^{*}\boldsymbol{u}_k\Sigma\boldsymbol{u}_k^{*}\boldsymbol{u}_k\widetilde{\boldsymbol{\psi}}_{k-1} \\ & + \mu_k^2\sigma_{v,k}^{2}E\|\boldsymbol{u}_k\|_{\Sigma}^{2}.\end{aligned} \tag{38}$$

Now, given that $\|x\|_A^2 + \|x\|_B^2 = \|x\|_{A+B}^2$, the previous equation can be rewritten more compactly as

$$E\|\widetilde{\boldsymbol{\psi}}_{k}\|_{\Sigma}^{2} = E\left(\|\widetilde{\boldsymbol{\psi}}_{k-1}\|_{\boldsymbol{\Sigma}'}^{2}\right) + \mu_k^2\sigma_{v,k}^{2}E\|\boldsymbol{u}_k\|_{\Sigma}^{2} \tag{39}$$

in terms of the stochastic weighting matrix

$$\boldsymbol{\Sigma}' = \Sigma - \mu_k\left(\boldsymbol{u}_k^{*}\boldsymbol{u}_k\Sigma + \Sigma\boldsymbol{u}_k^{*}\boldsymbol{u}_k\right) + \mu_k^2\|\boldsymbol{u}_k\|_{\Sigma}^{2}\boldsymbol{u}_k^{*}\boldsymbol{u}_k. \tag{40}$$

Invoking the independence of the regression data $\{\boldsymbol{u}_k\}$ allows us to write

$$E\left(\|\widetilde{\boldsymbol{\psi}}_{k-1}\|_{\boldsymbol{\Sigma}'}^{2}\right) = E\|\widetilde{\boldsymbol{\psi}}_{k-1}\|_{E\boldsymbol{\Sigma}'}^{2} \tag{41}$$

so that (39) and (40) become

$$\boxed{E\|\widetilde{\boldsymbol{\psi}}_{k}\|_{\Sigma}^{2} = E\|\widetilde{\boldsymbol{\psi}}_{k-1}\|_{\Sigma'}^{2} + \mu_k^2\sigma_{v,k}^{2}E\|\boldsymbol{u}_k\|_{\Sigma}^{2}} \tag{42}$$

where $\Sigma' = E\boldsymbol{\Sigma}'$ is given by

$$\boxed{\Sigma' = \Sigma - \mu_k E\left(\boldsymbol{u}_k^{*}\boldsymbol{u}_k\Sigma + \Sigma\boldsymbol{u}_k^{*}\boldsymbol{u}_k\right) + \mu_k^2 E\|\boldsymbol{u}_k\|_{\Sigma}^{2}\boldsymbol{u}_k^{*}\boldsymbol{u}_k} \tag{43}$$

and Σ' is now a *deterministic* matrix.

D. Gaussian Data

Recursion (42) is a spatial *variance relation* that will allow us to evaluate the *steady-state* performance of every node k. Note that Σ' in (43) is solely regressor dependent, and, therefore, its value is decoupled from (42). As a consequence, the study of the network behavior relies solely on the evaluation of the three data moments, as follows:

$$E\boldsymbol{u}_k^{*}\boldsymbol{u}_k = R_{u,k},\ \ E\|\boldsymbol{u}_k\|_{\Sigma}^{2} = \mathrm{Tr}(R_{u,k}\Sigma),\ \ \text{and}\ \ E\|\boldsymbol{u}_k\|_{\Sigma}^{2}\boldsymbol{u}_k^{*}\boldsymbol{u}_k. \tag{44}$$

The evaluation of the third moment can be involved for non-Gaussian data [17]. In this paper, we assume Gaussian data

for simplicity. Thus, assume that the $\{\boldsymbol{u}_k\}$ arise from a circular Gaussian distribution and introduce the eigendecomposition $R_{u,k} = U_k \Lambda_k U_k^*$, where U_k is unitary and Λ_k is a diagonal matrix with the eigenvalues of $R_{u,k}$. Introduce further the transformed quantities

$$\begin{aligned} &\overline{\boldsymbol{\psi}}_k \triangleq U_k^* \widetilde{\boldsymbol{\psi}}_k,\ \ \overline{\boldsymbol{\psi}}_{k-1} \triangleq U_k^* \widetilde{\boldsymbol{\psi}}_{k-1},\ \ \overline{\boldsymbol{u}}_k \triangleq \boldsymbol{u}_k U_k, \\ &\overline{\Sigma} \triangleq U_k^* \Sigma U_k,\ \ \overline{\Sigma}' \triangleq U_k^* \Sigma' U_k \end{aligned}.$$

Since U_k is unitary, we have that $\|\widetilde{\boldsymbol{\psi}}_{k-1}\|_{\Sigma}^2 = \|\overline{\boldsymbol{\psi}}_{k-1}\|_{\overline{\Sigma}}^2$ and $\|\boldsymbol{u}_k\|_{\Sigma}^2 = \|\overline{\boldsymbol{u}}_k\|_{\overline{\Sigma}}^2$, so that (42) and (43) can be rewritten in the equivalent forms

$$E\|\overline{\boldsymbol{\psi}}_k\|_{\overline{\Sigma}}^2 = E\|\overline{\boldsymbol{\psi}}_{k-1}\|_{\overline{\Sigma}'}^2 + \mu_k^2 \sigma_{v,k}^2 E\|\overline{\boldsymbol{u}}_k\|_{\overline{\Sigma}}^2 \tag{45}$$

$$\begin{aligned} \overline{\Sigma}' = &\overline{\Sigma} - \mu_k E\left(\overline{\boldsymbol{u}}_k^* \overline{\boldsymbol{u}}_k \overline{\Sigma} + \overline{\Sigma} \overline{\boldsymbol{u}}_k^* \overline{\boldsymbol{u}}_k\right) \\ &+ \mu_k^2 E\|\overline{\boldsymbol{u}}_k\|_{\overline{\Sigma}}^2 \overline{\boldsymbol{u}}_k^* \overline{\boldsymbol{u}}_k. \end{aligned} \tag{46}$$

The moments we need to evaluate are now $E\|\overline{\boldsymbol{u}}_k\|_{\overline{\Sigma}}^2$, $E\overline{\boldsymbol{u}}_k^* \overline{\boldsymbol{u}}_k$, and $E\|\overline{\boldsymbol{u}}_k\|_{\overline{\Sigma}}^2 \overline{\boldsymbol{u}}_k^* \overline{\boldsymbol{u}}_k$. The first two moments are straightforward since

$$E\|\overline{\boldsymbol{u}}_k\|_{\overline{\Sigma}}^2 = \mathrm{Tr}(\Lambda_k \overline{\Sigma}) \quad \text{and} \quad E\overline{\boldsymbol{u}}_k^* \overline{\boldsymbol{u}}_k = \Lambda_k. \tag{47}$$

The third moment is given for Gaussian regressors by [17]

$$E\|\overline{\boldsymbol{u}}_k\|_{\overline{\Sigma}}^2 \overline{\boldsymbol{u}}_k^* \overline{\boldsymbol{u}}_k = \Lambda_k \mathrm{Tr}(\overline{\Sigma}\Lambda_k) + \gamma \Lambda_k \overline{\Sigma} \Lambda_k \tag{48}$$

where $\gamma = 1$ for circular *complex data* and $\gamma = 2$ for *real data*. Substituting (47) and (48) into the variance relation (45), (46) leads to

$$\boxed{E\|\overline{\boldsymbol{\psi}}_k\|_{\overline{\Sigma}}^2 = E\|\overline{\boldsymbol{\psi}}_{k-1}\|_{\overline{\Sigma}'}^2 + \mu_k^2 \sigma_{v,k}^2 \mathrm{Tr}(\Lambda_k \overline{\Sigma})} \tag{49}$$

$$\boxed{\overline{\Sigma}' = \overline{\Sigma} - \mu_k\left(\Lambda_k \overline{\Sigma} + \overline{\Sigma}\Lambda_k\right) + \mu_k^2\left(\Lambda_k \mathrm{Tr}\left(\overline{\Sigma}\Lambda_k\right) + \gamma \Lambda_k \overline{\Sigma} \Lambda_k\right)}. \tag{50}$$

E. Diagonalization

Since Σ is at our choice, we choose it such that both $\overline{\Sigma}$ and $\overline{\Sigma}'$ will become diagonal in (50). This suggests a more compact notation in terms of the diagonal entries of $\overline{\Sigma}$ and $\overline{\Sigma}'$. Thus, introduce the $M \times 1$ column vectors

$$\overline{\sigma} \triangleq \mathrm{diag}\{\overline{\Sigma}\}, \quad \overline{\sigma}' \triangleq \mathrm{diag}\{\overline{\Sigma}'\}, \quad \lambda_k \triangleq \mathrm{diag}\{\Lambda_k\} \tag{51}$$

where the $\mathrm{diag}\{\}$ notation will be used in two ways: $\Lambda = \mathrm{diag}\{\lambda\}$ is a diagonal matrix whose entries are those of the vector λ, and $\lambda = \mathrm{diag}\{\Lambda\}$ is a vector containing the main diagonal of Λ.

Using the diagonal notation, expression (50) can be rewritten in terms of $\{\overline{\sigma}, \lambda_k\}$ as

$$\begin{aligned} \overline{\sigma}' &= \left(I - 2\mu_k \Lambda_k + \gamma \mu_k^2 \Lambda_k^2\right) \overline{\sigma} + \mu_k^2 \left(\lambda_k^T \overline{\sigma}\right) \lambda_k \\ &= \overline{F}_k \overline{\sigma} \end{aligned} \tag{52}$$

where the coefficient matrix $\overline{F}_k$ is defined by

$$\boxed{\overline{F}_k \triangleq I - 2\mu_k \Lambda_k + \gamma \mu_k^2 \Lambda_k^2 + \mu_k^2 \lambda_k \lambda_k^T}. \tag{53}$$

Moreover, expression (49) becomes

$$E\|\overline{\boldsymbol{\psi}}_k\|_{\mathrm{diag}\{\overline{\sigma}\}}^2 = E\|\overline{\boldsymbol{\psi}}_{k-1}\|_{\mathrm{diag}\{\overline{F}_k \overline{\sigma}\}}^2 + \mu_k^2 \sigma_{v,k}^2 \left(\lambda_k^T \overline{\sigma}\right). \tag{54}$$

For the sake of compactness, the $\mathrm{diag}\{\}$ notation will be dropped from the subscripts in (54), keeping only the corresponding vectors

$$\boxed{E\|\overline{\boldsymbol{\psi}}_k^{(i)}\|_{\overline{\sigma}_k}^2 = E\|\overline{\boldsymbol{\psi}}_{k-1}^{(i)}\|_{\overline{F}_k \overline{\sigma}_k}^2 + \mu_k^2 \sigma_{v,k}^2 \left(\lambda_k^T \overline{\sigma}_k\right)} \tag{55}$$

where we are restoring the time index i for clarity, and we are replacing $\{\overline{\sigma}, \overline{\sigma}'\}$ by $\{\overline{\sigma}_k, \overline{\sigma}_k'\}$ in order to indicate that the weighting matrix can be node dependent.

F. Steady-State Behavior

Let $\overline{\boldsymbol{p}}_k \triangleq \overline{\boldsymbol{\psi}}_k^{(\infty)}$ and $g_k \triangleq \mu_k^2 \sigma_{v,k}^2 \lambda_k^T$ (a row vector). Then, for $i \to \infty$ (i.e., in steady state), the variance relation (55) gives

$$E\|\overline{\boldsymbol{p}}_k\|_{\overline{\sigma}_k}^2 = E\|\overline{\boldsymbol{p}}_{k-1}\|_{\overline{F}_k \overline{\sigma}_k}^2 + g_k \overline{\sigma}_k, \quad k = 1, 2, \ldots, N. \tag{56}$$

We want to use this expression to evaluate the performance measures, as follows:

$$\eta_k = E\|\overline{\boldsymbol{p}}_{k-1}\|_q^2, \quad q \triangleq \mathrm{diag}\{I\} \quad \text{(MSD)} \tag{57}$$

$$\zeta_k = E\|\overline{\boldsymbol{p}}_{k-1}\|_{\lambda_k}^2, \quad \lambda_k = \mathrm{diag}\{\Lambda_k\} \quad \text{(EMSE)} \tag{58}$$

$$\xi_k = \zeta_k + \sigma_{v,k}^2 \quad \text{(MSE)} \tag{59}$$

with weighting vectors q and λ_k. Observe, however, that (56) is a coupled equation: it involves both $\overline{\boldsymbol{p}}_k$ and $\overline{\boldsymbol{p}}_{k-1}$, i.e., information from two spatial locations. The ring topology together with the weighting matrices can be exploited to resolve this difficulty.

Thus, note that by iterating (56) we get a set of N coupled equalities

$$\begin{aligned} E\|\overline{\boldsymbol{p}}_1\|_{\overline{\sigma}_1}^2 &= E\|\overline{\boldsymbol{p}}_N\|_{\overline{F}_1 \overline{\sigma}_1}^2 + g_1 \overline{\sigma}_1 \\ E\|\overline{\boldsymbol{p}}_2\|_{\overline{\sigma}_2}^2 &= E\|\overline{\boldsymbol{p}}_1\|_{\overline{F}_2 \overline{\sigma}_2}^2 + g_2 \overline{\sigma}_2 \\ &\vdots \\ E\|\overline{\boldsymbol{p}}_{k-2}\|_{\overline{\sigma}_{k-2}}^2 &= E\|\overline{\boldsymbol{p}}_{k-3}\|_{\overline{F}_{k-2} \overline{\sigma}_{k-2}}^2 + g_{k-2} \overline{\sigma}_{k-2} \\ E\|\overline{\boldsymbol{p}}_{k-1}\|_{\overline{\sigma}_{k-1}}^2 &= E\|\overline{\boldsymbol{p}}_{k-2}\|_{\overline{F}_{k-1} \overline{\sigma}_{k-1}}^2 + g_{k-1} \overline{\sigma}_{k-1} \\ &\vdots \\ E\|\overline{\boldsymbol{p}}_N\|_{\overline{\sigma}_N}^2 &= E\|\overline{\boldsymbol{p}}_{N-1}\|_{\overline{F}_N \overline{\sigma}_N}^2 + g_N \overline{\sigma}_N. \end{aligned} \tag{60, 61}$$

These equations can be solved for $\{\eta_k, \zeta_k\}$ via a suitable choice of the free parameters $\{\overline{\sigma}_k\}$ and proper manipulation of the equations. Thus, note that (61) expresses $E\|\overline{\boldsymbol{p}}_{k-1}\|_{\overline{\sigma}_{k-1}}^2$ in terms of $E\|\overline{\boldsymbol{p}}_{k-2}\|_{\overline{F}_{k-1} \overline{\sigma}_{k-1}}^2$. Then, choosing $\overline{\sigma}_{k-2} = \overline{F}_{k-1} \overline{\sigma}_{k-1}$ in

(60) gives

$$E\|\overline{\boldsymbol{p}}_{k-2}\|^2_{\overline{F}_{k-1}\overline{\sigma}_{k-1}}=E\|\overline{\boldsymbol{p}}_{k-3}\|^2_{\overline{F}_{k-2}\overline{F}_{k-1}\overline{\sigma}_{k-1}}+g_{k-2}\overline{F}_{k-1}\overline{\sigma}_{k-1} \tag{62}$$

so that

$$E\|\overline{\boldsymbol{p}}_{k-1}\|^2_{\overline{\sigma}_{k-1}} = E\|\overline{\boldsymbol{p}}_{k-3}\|^2_{\overline{F}_{k-2}\overline{F}_{k-1}\overline{\sigma}_{k-1}} + g_{k-2}\overline{F}_{k-1}\overline{\sigma}_{k-1} + g_{k-1}\overline{\sigma}_{k-1}. \tag{63}$$

Iterating in this manner, we end up with an equality involving only $\overline{\boldsymbol{p}}_{k-1}$, namely

$$\begin{aligned} E\|\overline{\boldsymbol{p}}_{k-1}\|^2_{\overline{\sigma}_{k-1}} =& E\|\overline{\boldsymbol{p}}_{k-1}\|^2_{\overline{F}_k\cdots\overline{F}_N\overline{F}_1\cdots\overline{F}_{k-1}\overline{\sigma}_{k-1}} \\ &+ g_k\overline{F}_{k+1}\cdots\overline{F}_N\overline{F}_1\cdots\overline{F}_{k-1}\overline{\sigma}_{k-1} \\ &+ g_{k+1}\overline{F}_{k+2}\cdots\overline{F}_N\overline{F}_1\cdots\overline{F}_{k-1}\overline{\sigma}_{k-1} \\ &+ \cdots + g_{k-2}\overline{F}_{k-1}\overline{\sigma}_{k-1} + g_{k-1}\overline{\sigma}_{k-1}. \end{aligned} \tag{64}$$

It is convenient to define, for each node k, a set of N matrices in terms of products of $\overline{F}$ matrices

$$\Pi_{k,\ell} \triangleq \overline{F}_{k+\ell-1}\overline{F}_{k+\ell}\cdots\overline{F}_N\overline{F}_1\cdots\overline{F}_{k-1}, \quad \ell = 1,2,\ldots,N \tag{65}$$

where the subscripts are all $mod\ N$. The matrix $\Pi_{k,\ell}$ can be interpreted as the transition matrix that is necessary for the weighting vector $\overline{\sigma}_{k-1}$ to reach node k cyclicly through nodes $k-1,k-2,\ldots,N,N-1,\ldots,k$. With this definition, we can rewrite (64) as

$$\boxed{E\|\overline{\boldsymbol{p}}_{k-1}\|^2_{(I-\Pi_{k,1})\overline{\sigma}_{k-1}} = a_k\overline{\sigma}_{k-1}} \tag{66}$$

where the *row* vector a_k is defined as

$$a_k \triangleq g_k\Pi_{k,2} + g_{k+1}\Pi_{k,3}\cdots + g_{k-2}\Pi_{k,N} + g_{k-1}. \tag{67}$$

Likewise, the vector a_k has the interpretation of the combined effect of transformed noise and local data statistics reaching node k from other nodes over the ring topology.

Recall that we wish to evaluate $\eta_k = E\|\overline{\boldsymbol{p}}_{k-1}\|^2_q$ and $\zeta_k = E\|\overline{\boldsymbol{p}}_{k-1}\|^2_{\lambda_k}$. Thus, different choices of the weighting vector $\overline{\sigma}_{k-1}$ in (66) should enable us to calculate the quantities that describe the steady-state behavior of each node. Selecting the weighting vector $\overline{\sigma}_{k-1}$ as the solution of the linear equation $(I-\Pi_{k,1})\overline{\sigma}_{\eta,k-1} = q$, we arrive at an expression for the desired MSD:

$$\boxed{\eta_k = a_k(I-\Pi_{k,1})^{-1}q \quad \text{(MSD)}}. \tag{68}$$

Likewise, for the EMSE, we choose the weighting vector $\overline{\sigma}_{k-1}$ as the solution of $(I-\Pi_{k,1})\overline{\sigma}_{\zeta,k-1} = \lambda_k$ so that

$$\boxed{\zeta_k = a_k(I-\Pi_{k,1})^{-1}\lambda_k \quad \text{(EMSE)}}. \tag{69}$$

We summarize the results in the following theorem. The statement quantifies the performance of a spatio–temporal distributed adaptive solution.

Theorem 1: Consider a distributed adaptive scheme of the form (16) with space–time data $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$ satisfying the model (18) and the assumptions A1)–A3) in Section III-A. Assume further that the regressors $\boldsymbol{u}_{k,i}$ are circularly Gaussian. The steady-state performance of each node in the mean-square sense is given by

$$\begin{aligned} \eta_k &= a_k(I-\Pi_{k,1})^{-1}q \quad \text{(MSD)} \\ \zeta_k &= a_k(I-\Pi_{k,1})^{-1}\lambda_k \quad \text{(EMSE)} \\ \xi_k &= \zeta_k + \sigma^2_{v,k} \quad \text{(MSE)} \end{aligned}$$

where $\Pi_{k,1}$ and a_k are defined by (65) and (67), respectively, the vectors q and λ_k are defined by (51) and (57), and $g_k = \mu_k^2\sigma^2_{v,k}\lambda_k^T$.

Analyzing these results, we find that the presence of the matrix $\Pi_{k,1}$ and the vector a_k indicate that every node individually experiences the influence of the entire network, with some emphasis given to the local statistics, as represented by λ_k and by $\sigma^2_{v,k}$.

For small step sizes, $\overline{F}_k \approx I - 2\mu_k\Lambda_k$, i.e., $\overline{F}_k$ becomes a diagonal matrix. As a result, matrix $\Pi_{k,\ell} = \Pi = \overline{F}_1\overline{F}_2\cdots\overline{F}_N$ will be diagonal as well and can be approximated by its dominant terms, i.e.,

$$\begin{aligned} \Pi &= (I-2\mu_1\Lambda_1)(I-2\mu_2\Lambda_2)\cdots(I-2\mu_N\Lambda_N) \\ &\approx I - (2\mu_1\Lambda_1 + 2\mu_2\Lambda_2 + \cdots + 2\mu_N\Lambda_N) \end{aligned}$$

so that

$$I - \Pi \approx 2\mu_1\Lambda_1 + 2\mu_2\Lambda_2 + \cdots + 2\mu_N\Lambda_N$$

and $a_k \approx \Sigma_{k=1}^N g_k$. From (68), we get

$$\eta_k \approx \left(\mu_1^2\sigma^2_{v,1}\lambda_1^T + \cdots \mu_N^2\sigma^2_{v,N}\lambda_N^T\right)(2\mu_1\Lambda_1 + \cdots + 2\mu_N\Lambda_N)^{-1}q$$

as $\mu_l \to 0$, which reveals an interesting behavior. Despite the quite simple cooperation mode (i.e., incremental), for small step sizes, there is an equalization effect on the MSD throughout the network, suggesting that the intermediate averaging procedures in consensus implementations, as mentioned in the introduction, can be avoided. This result has been confirmed in the simulations, even though the step size there was relatively large (see Fig. 10). Note that making $\mu_k \to 0$ drives the MSD to zero for every node in the network. In the same vein, the EMSE for small step sizes is given by

$$\begin{aligned} \zeta_k \approx \left(\mu_1^2\sigma^2_{v,1}\lambda_1^T + \cdots \mu_N^2\sigma^2_{v,N}\lambda_N^T\right) \\ \times(2\mu_1\Lambda_1 + \cdots + 2\mu_N\Lambda_N)^{-1}\lambda_k \end{aligned}$$

which also goes asymptotically to zero as $\mu_l \to 0$, causing the MSE to achieve the background noise level $\sigma^2_{v,k}$ everywhere.

IV. Simulations

In this section, we provide computer simulations comparing the theoretical performance to simulation results. Although the analysis relied on the independence assumptions, all simulations were carried out using regressors with shift structure to cope

with realistic scenarios. Therefore, the regressors are filled up as

$$u_{k,i} = \mathrm{col}\{u_k(i), u_k(i-1), \ldots, u_k(i-M+1)\}. \quad (70)$$

In order to generate the performance curves, 100 independent experiments were performed and averaged. The steady-state curves are generated by running the network learning process for 50 000 iterations. The quantities of interest, namely, MSD, EMSE, and MSE, are then obtained by averaging the last 5000 samples of the corresponding learning curves. The measurement data $\{d_k(i)\}$ are generated according to the model (18) with the (unknown) $M \times 1$ vector set as $w^o = \mathrm{col}\{1, 1, \ldots, 1\}/\sqrt{M}$.

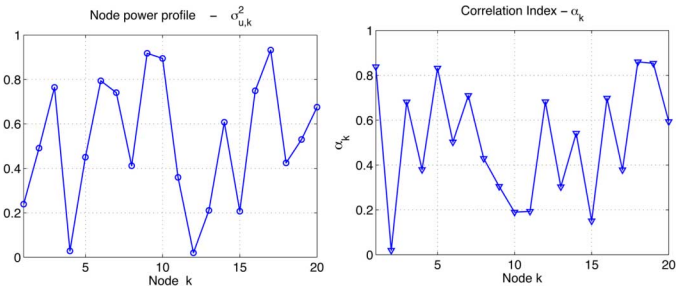
Two kinds of curves are presented. One kind depicts the steady-state quantities as a function of the node k for a particular choice of the step size μ. These curves can be used to guide the design of the network. For instance, they tell the designer how to adjust the step-size at a certain node to compensate for a signal power increase in nearby nodes. Or even, how the filters are affected by a noise power increase at some nodes in the network. A second kind of curve depicts the behavior of the steady-state quantities as a function of the step size for a particular node. These curves evaluate the quality of the theoretical model [17]. Usually, large deviations between theory and simulation are expected for bigger step sizes: that is, when the simplifying assumptions adopted in the analysis are no longer reasonable; therefore, curves like those in Figs. 13–15 have a strong theoretical appeal.

In this example, the network consists of $N = 20$ nodes, with each regressor of size (1×10) collecting data by observing a time-correlated sequence $\{u_k(i)\}$, generated as

$$u_k(i) = \alpha_k \cdot u_k(i-1) + \beta_k \cdot z_k(i), \quad i > -\infty. \quad (71)$$

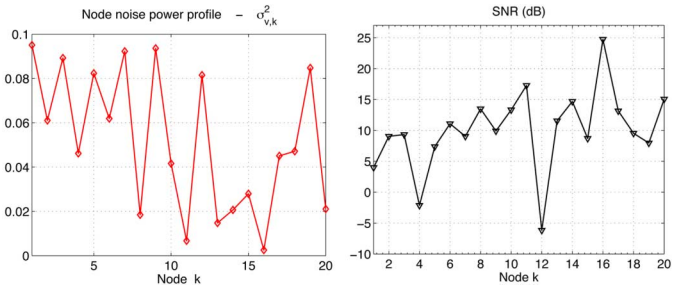
Here, $\alpha_k \in [0, 1)$ is the correlation index and $z_k(i)$ is a spatially independent white Gaussian process with unit variance and $\beta_k = \sqrt{\sigma_{u,k}^2 \cdot (1 - \alpha_k^2)}$. It is straightforward to show that the resulting regressors have Toeplitz covariance matrices $R_{u,k}$, with correlation sequence $r_k(i) = \sigma_{u,k}^2 (\alpha_k)^{|i|}$, $i = 0, \ldots, M-1$. The regressor power profile $\{\sigma_{u,k}^2\} \in (0, 1]$, the correlation indexes $\{\alpha_k\} \in [0, 1)$ and the Gaussian noise variances $\{\sigma_{v,k}^2\} \in (0, 0.1]$ were chosen at random and are depicted in Fig. 8(a) and (b) and 9(a), respectively. The corresponding signal-to-noise ratio (SNR) is plotted in Fig. 9(b).

One can see in Figs. 10–15 a good match between theory and simulations, even for larger step sizes. Note also that despite the diverse statistical profile, the MSD in Fig. 10 is roughly even over the network, with a deviation of ± 0.25 dB at -20.5 dB. On the other hand, the EMSE and the MSE are more sensitive to local statistics, as depicted in Figs. 11 and 12. In Fig. 12, one also sees that the MSE roughly reflects the noise power, which indicates the good performance of the adaptive network: in steady state if the adaptive node is performing well, $\psi_k^{(\infty)}$ is a good estimate for w^o. Therefore, from (23), we would have that the residual error $e_k(\infty)$ should be close to the background noise.



(a) (b)

Fig. 8. (a) Regressor power profile. (b) Correlation index per node.



(a) (b)

Fig. 9. (a) Noise power profile. (b) Signal-to-noise ratio profile.

It would be desirable to drive the whole network to an equalized performance, as in Fig. 10. A good step-size design, together with the cooperative scheme that has been proposed, may take advantage of the spatial diversity provided by the adaptive network. By properly tuning the step size at each node, a good level of performance equalization could be achieved throughout the network. For instance, the network could be driven to the performance of nodes 11 and 16 in Fig. 12; nodes presenting poor performance, or high noise level, can be assigned with small step sizes, such that, in the limit case, they would become simply relay nodes.

V. Remarks and Future Work

Several efforts have been pursued in the literature to develop distributed estimation schemes based on consensus strategies. In [13], a distributed approximate least-squares solution for a limited number of measurements was proposed, using one measurement per node; the resulting solution is not adaptive. The original formulation was extended for the multiple measurements case in [14], and the solution lies on two time scales. Both algorithms achieve asymptotically the least-squares solution. They involve matrix products and matrix inversions at every node and every iteration leading to $O(M^3)$ computational complexity per node and $O(M^2)$ communication complexity per node.

In the context of Kalman filtering, some of the proposed strategies for distributed estimation are not fully distributed and rely on hierarchical architectures or central processing nodes [28], [29]. By relying on consensus strategies for distributed processing, some approximate distributed Kalman implementations are also available [16], [30]. However, they suffer from

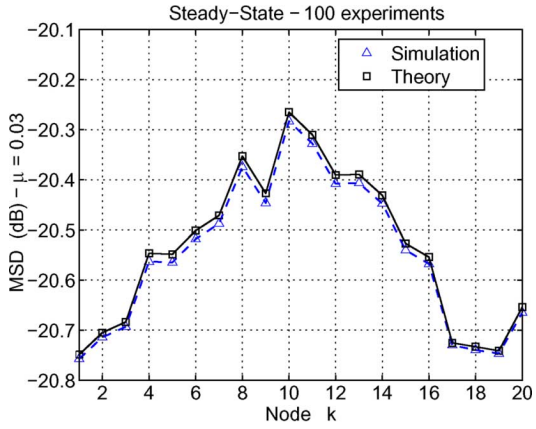


Fig. 10. MSD versus node—$\mu = 0.03$.

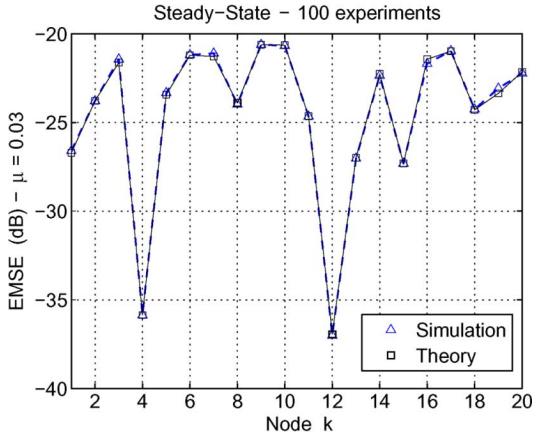


Fig. 11. EMSE versus node—$\mu = 0.03$.

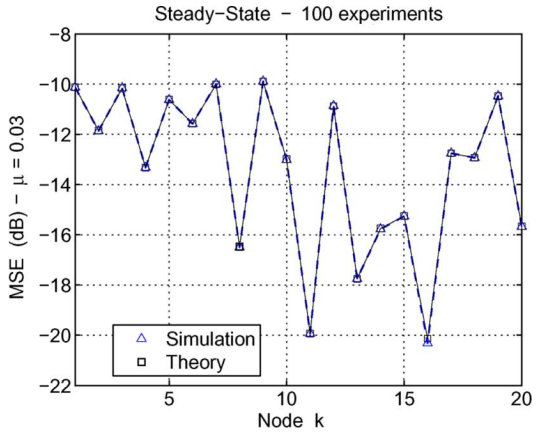


Fig. 12. MSE versus node—$\mu = 0.03$.

limited response to fast changing processes when the communication network has limited resources. Moreover, in Kalman filter implementations, one needs to model the process under observation, e.g., as first-order Markov processes, as well as use data statistics. Subsequently, the model has to be shared among the nodes in order to take advantage of the *a priori* knowledge.

In a more general vein, particle filters are more flexible in terms of process modeling. They account for nonlinear models and non-Gaussian data through a set of filter candidates (or particles) that try to approximate the process under observation. When properly designed, they may achieve accurate descriptions of the process of interest. However, as in Kalman filtering,

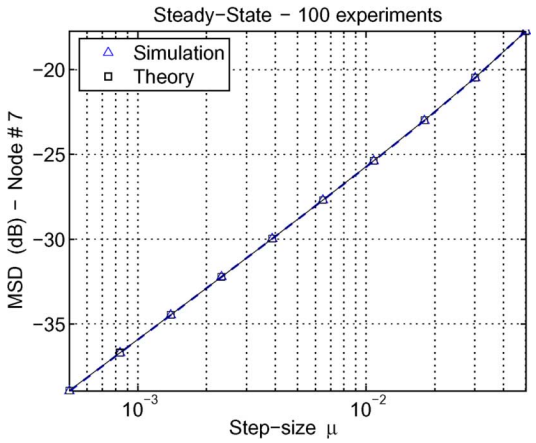


Fig. 13. MSD versus μ—node 7.

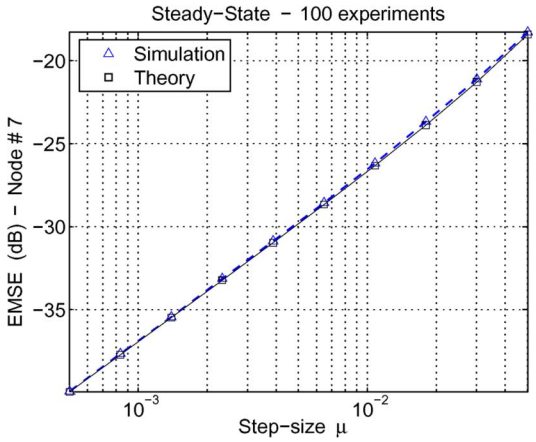


Fig. 14. EMSE versus μ—node 7.

particle filters require *a priori* knowledge about the process, represented by probability distributions that are assumed known. They are also computationally complex and challenging to implement in a distributed manner. In [31], distributed particle filters that rely on ring, or "chain," topologies (trees are also possible) are proposed under certain conditions on the probability distributions of the process. The resulting algorithms are demanding in terms of computational complexity and communication resources, thus limiting the ability of the network to respond efficiently to fast changing environments.

The results developed in this paper lead to good performance allied with low communication and computational requirements. The proposed distributed scheme requires $O(M)$ operations per iteration per node for both communications and computations. The scheme inherits the robustness of LMS implementations and promptly respond to new data. The solution does not require two separate time scales. More sophisticated cooperative modes (rather than the incremental mode) can also be pursued, e.g., a diffusion mode of the form [15], [18]:

$$\phi_k^{(i-1)} = f_{k,i-1}\left(\psi_\ell^{(i-1)}; \ell \in \mathcal{N}_k(i-1)\right)$$
$$\psi_k^{(i)} = \phi_k^{(i-1)} + \mu u_{k,i}^* \left(d_k(i) - u_{k,i}\phi_k^{(i-1)}\right)$$

for a local combiner $f_{k,i-1}(\cdot)$ that consults nodes from the neighborhood $\mathcal{N}_k(i-1)$. This additional level of complexity leads to more challenging and more interesting distributed

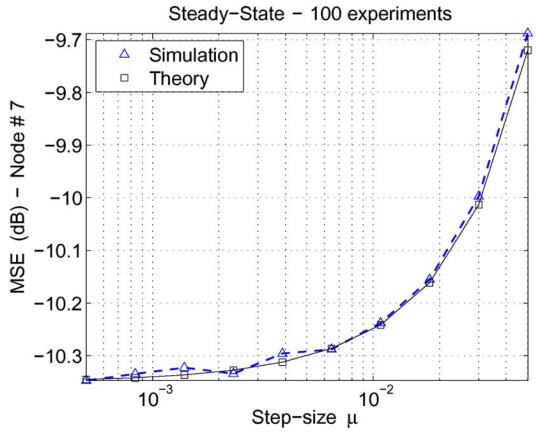


Fig. 15. MSE versus μ—node 7.

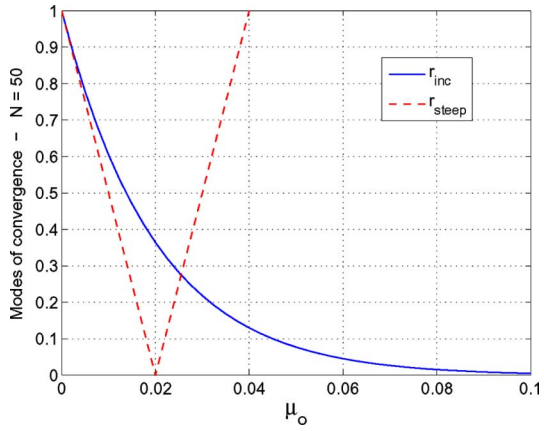


Fig. 16. Modes of convergence for algorithms (13) and (14).

adaptive structures that can still be examined and studied by the tools and framework developed in this paper, albeit with more effort. This work can also be extended to operate over the collaboration protocols proposed in [32] and [33] and to other adaptive algorithms at the node level. We will extend the current discussion to these more general scenarios in future works.

Appendix A
Comparing Steepest-Descent and Its Incremental Version

We want to compare algorithms (14) and (13) to provide insights about their stochastic counterparts. For simplicity, assume that the statistical profile throughout the network is roughly similar, i.e., $R_{u,k} = R_u > 0$, and $R_{du,k} = R_{du}$, such that (14) becomes

$$\psi_k^{(i)} = \psi_{k-1}^{(i)} + \mu\left(R_{du} - R_u\psi_{k-1}^{(i)}\right). \tag{72}$$

Subtracting both sides from w^o and using (19) yields $\widetilde{\psi}_k^{(i)} = (I - \mu R_u)\widetilde{\psi}_{k-1}^{(i)}$, or, equivalently

$$\widetilde{\psi}_N^{(i)} = (I - \mu R_u)^N \widetilde{\psi}_0^{(i)} \tag{73}$$

so that

$$\widetilde{w}_i = (I - \mu R_u)^N \widetilde{w}_{i-1} \quad \text{(Incremental)} \tag{74}$$

where $\widetilde{w}_i \triangleq w^o - w_i$ is the global weight error vector. Now, employing the eigendecomposition $R_u = U\Lambda U^*$ and defining the transformed vector $\overline{w}_i \triangleq U^*\widetilde{w}_i$, we get

$$\overline{w}_i = (I - \mu\Lambda)^N \overline{w}_{i-1}. \tag{75}$$

Similarly, for the steepest-descent algorithm (13), we have

$$\psi_k^{(i)} = \psi_{k-1}^{(i)} + \mu(R_{du} - R_u w_{i-1}) \tag{76}$$

so that $\widetilde{\psi}_k^{(i)} = \widetilde{\psi}_{k-1}^{(i)} - \mu R_u\widetilde{w}_{i-1}$ and

$$\widetilde{w}_i = (I - \mu N R_u)\widetilde{w}_{i-1} \quad \text{(Steepest-Descent)}. \tag{77}$$

Writing (77) in terms of the transformed vector, we arrive at

$$\overline{w}_i = (I - \mu N\Lambda)\overline{w}_{i-1}. \tag{78}$$

From (75) and (78), we find that the M modes of convergence of the algorithms are given by

$$\begin{aligned} r_{\text{inc},\ell}(\mu) &= |1 - \mu\lambda_\ell|^N \\ r_{\text{steep},\ell}(\mu) &= |1 - \mu\lambda_\ell N| \end{aligned} \tag{79}$$

for $\ell = 1, \ldots, M$. Fig. 16 shows both modes of convergence for the case $R_u = I$, where $\mu_o = \mu\lambda$ is the normalized step size. Note that for larger step sizes, the incremental algorithm has a faster convergence rate than the steepest-descent solution. The incremental algorithm has a fast convergence rate even for small step sizes, which supports why its corresponding incremental adaptive version has better convergence rate together with a smaller MSE in steady state. Furthermore, the stability range for the incremental algorithm is wider, leading to more robust implementations.

As a matter of fact, for diminishing step sizes, the incremental solution (14) and the steepest-descent solution (13) tend to the same behavior also with diverse statistical profiles over the network. To observe this fact, we first note from (9) that the k^{th} partial gradient is given by

$$[\nabla J_k(w)]^* = -R_{du,k} + R_{u,k}w. \tag{80}$$

Inspecting (80), we note that the following equality holds for a scalar μ and any two column vectors x and y:

$$[\nabla J_k(x - \mu y)]^* = [\nabla J_k(x)]^* - \mu[\nabla J_k(y)]^* - \mu R_{du,k} \tag{81}$$

where ∇J_k is computed relative to w. Now, we iterate the incremental solution (14) starting with $\psi_0^{(i)} = w_{i-1}$, as follows:

$$\begin{aligned} \psi_1^{(i)} &= w_{i-1} - \mu\left[\nabla J_1\left(\psi_0^{(i)}\right)\right]^* \\ \psi_2^{(i)} &= w_{i-1} - \mu\left[\nabla J_1\left(\psi_0^{(i)}\right)\right]^* - \mu\left[\nabla J_2\left(\psi_1^{(i)}\right)\right]^* \\ &\vdots \\ \psi_{k-1}^{(i)} &= w_{i-1} - \mu\sum_{\ell=1}^{k-1}\left[\nabla J_\ell\left(\psi_{\ell-1}^{(i)}\right)\right]^*. \end{aligned} \tag{82}$$

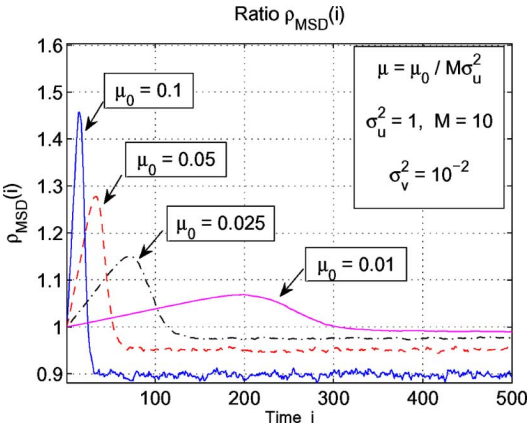


Fig. 17. Plot of ratio $\rho_{\rm MSD}(i)$ at node 1 for algorithms (16) and (17).

Substituting (82) into the argument of ∇J_k in (14) gives

$$\psi_k^{(i)} = \psi_{k-1}^{(i)} - \mu\left[\nabla J_k\left(w_{i-1} - \mu\sum_{\ell=1}^{k-1}\left[\nabla J_\ell\left(\psi_{\ell-1}^{(i)}\right)\right]^*\right)\right]^*. \tag{83}$$

Using relation (81) with the choices $x = w_{i-1}$ and $y = \sum_{\ell=1}^{k-1}[\nabla J_\ell(\psi_{\ell-1}^{(i)})]^*$ leads to

$$\psi_k^{(i)} = \underbrace{\psi_{k-1}^{(i)} - \mu\left([\nabla J_k(w_{i-1})]^*\right)}_{\text{steepest-descent as in (13)}} + \underbrace{\mu^2\left(\left[\nabla J_k\left(\sum_{\ell=1}^{k-1}\left[\nabla J_\ell\left(\psi_{\ell-1}^{(i)}\right)\right]^*\right)\right]^* - R_{du,k}\right)}_{\text{extra term due to incremental procedure}}. \tag{84}$$

Therefore, the incremental algorithm can be written as a sum of the steepest-descent update plus extra terms. As $\mu \to 0$, the μ term dominates the μ^2 term, and the incremental algorithm (14) and the standard gradient (13) tend to the same behavior. The ratio

$$\rho_{\rm MSD}(i) \triangleq \frac{E\|\widetilde{\boldsymbol{\psi}}_{\rm inc}^{(i)}\|^2}{E\|\widetilde{\boldsymbol{\psi}}_{steep}^{(i)}\|^2} \tag{85}$$

is plotted in Fig. 17 for the incremental adaptive solution (16) and the stochastic steepest-descent (17), both at node 1. The same settings as the example presented in Figs. 6 and 7 are used, and 200 experiments are run, with $\sigma_v^2 = 10^{-2}$ throughout the network. As $\mu \to 0$, both algorithms exhibit similar behavior, as suggested by (84).

ACKNOWLEDGMENT

The authors would like to acknowledge the contribution of the graduate student F. Cattivelli to some of the arguments in Appendix A.

REFERENCES

[1] D. Estrin, G. Pottie, and M. Srivastava, "Intrumenting the world with wireless sensor setworks," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, Salt Lake City, UT, May 2001, pp. 2033–2036.

[2] M. G. Rabbat and R. D. Nowak, "Quantized incremental algorithms for distributed optimization," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 4, pp. 798–808, Apr. 2005.

[3] M. Wax and T. Kailath, "Decentralized processing in sensor arrays," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-33, no. 4, pp. 1123–1129, Oct. 1985.

[4] J. Tsitsiklis and M. Athans, "Convergence and asymptotic agreement in distributed decision problems," *IEEE Trans. Autom. Control*, vol. AC-29, no. 1, pp. 42–50, Jan. 1984.

[5] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1520–1533, Sep. 2004.

[6] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Syst. Control Lett.*, vol. 53, no. 1, pp. 65–78, Sep. 2004.

[7] L. A. Rossi, B. Krishnamachari, and C.-C. J. Kuo, "Distributed parameter estimation for monitoring diffusion phenomena using physical models," in *Proc. 1st IEEE Conf. Sensor and Ad Hoc Communications and Networks*, Santa Clara, CA, Oct. 2004, pp. 460–469.

[8] D. Li, K. D. Wong, Y. H. Hu, and A. M. Sayeed, "Detection, classification, and tracking of targets," *IEEE Signal Process. Mag.*, vol. 19, no. 2, pp. 17–29, Mar. 2002.

[9] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Commun. Mag.*, vol. 40, no. 8, pp. 102–114, Aug. 2002.

[10] D. Culler, D. Estrin, and M. Srivastava, "Overview of sensor networks," *Computer*, vol. 37, no. 8, pp. 41–49, Aug. 2004.

[11] M. G. Rabbat and R. D. Nowak, "Decentralized source localization and tracking," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, Montreal, QC, Canada, May 2004, vol. III, pp. 921–924.

[12] C. G. Lopes and A. H. Sayed, "Distributed adaptive incremental strategies: Formulation and performance analysis," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, Toulouse, France, May 2006, vol. 3, pp. 584–587.

[13] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *Proc. 4th Int. Symp. Information Processing in Sensor Networks*, Los Angeles, CA, Apr. 2005, pp. 63–70.

[14] L. Xiao, S. Boyd, and S. Lall, "A space-time diffusion scheme for peer-to-peer least-squares estimation," in *Proc. 5th Int. Symp. Information Processing in Sensor Networks*, Nashville, TN, Apr. 2006.

[15] C. G. Lopes and A. H. Sayed, "Diffusion least-mean squares over adaptive networks," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, Honolulu, HI, Apr. 2007, pp. 917–920.

[16] D. P. Spanos, R. Olfati-Saber, and R. M. Murray, "Approximate distributed Kalman filtering in sensor networks with quantifiable performance," in *Proc. 4th Int. Symp. Information Processing in Sensor Networks*, Los Angeles, CA, Apr. 2005, pp. 133–139.

[17] A. H. Sayed, *Fundamentals of Adaptive Filtering*. New York: Wiley, 2003.

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

[19] A. H. Sayed and C. G. Lopes, "Distributed recursive least-squares strategies over adaptive networks," in *Proc. Asilomar Conf. Signals, Systems, Computers*, Monterey, CA, Oct. 2006, pp. 233–237.

[20] 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.

[21] A. Nedic and D. Bertsekas, "Incremental subgradient methods for nondifferentiable optimization," *SIAM J. Optim.*, vol. 12, no. 1, pp. 109–138, 2001.

[22] J. Tsitsiklis, D. P. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Trans. Autom. Control*, vol. AC-31, no. 9, pp. 650–655, Sep. 1986.

[23] B. T. Poljak and Y. Z. Tsypkin, "Pseudogradient adaptation and training algorithms," *Autom. Remote Control*, vol. 12, pp. 83–94, 1973.

[24] J. Haupt and R. Nowak, "Signal reconstruction from randomized projections with applications to wireless sensing," in *Proc. 13th Workshop on Statistical Signal Processing*, Bordeaux, France, Jul. 2005, pp. 1182–1187.
[25] H. Benali, M. Pelegrini, and F. Kruggel, "Spatio-temporal covariance model for medical images sequences: Application to functional MRI data," in *Information Processing in Medical Imaging*, ser. Lecture Notes in Computer Science. Berlin: Springer, 2001, vol. 2082, pp. 197–203.
[26] N. R. Yousef and A. H. Sayed, "A unified approach to the steady-state and tracking analyses of adaptive filters," *IEEE Trans. Signal Process.*, vol. 49, no. 2, pp. 314–324, Feb. 2001.
[27] T. Y. A. Naffouri and A. H. Sayed, "Transient analysis of data-normalized adaptive filters," *IEEE Trans. Signal Process.*, vol. 51, no. 3, pp. 639–652, Mar. 2003.
[28] S. C. Felter, "An overview of decentralized Kalman filter techniques," in *Proc. IEEE Southern Tier Technical Conf.*, Binghamton, NY, Apr. 1990, pp. 79–87.
[29] L. Hong, "Adaptive distributed filtering in multicoordinated systems," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 27, no. 4, pp. 715–724, Jul. 1991.
[30] D. Spanos, R. Olfati-Saber, and R. Murray, "Distributed sensor fusion using dynamic consensus," presented at the 16th IFAC World Congr., Prague, Czech Republic, Jul. 2005.
[31] M. Coates, "Distributed particle filters for sensor networks," in *Proc. 3rd Int. Symp. Information Processing in Sensor Networks*, Apr. 2004, pp. 99–107.
[32] C. E. Guestrin, P. Bodik, R. Thibaux, M. A. Paskin, and S. Madden, "Distributed regression: An efficient framework for modeling sensor network data," in *Proc. 3rd Int. Symp. Information Processing in Sensor Networks*, Berkeley, CA, Apr. 2004, pp. 1–10.
[33] M. A. Paskin, C. E. Guestrin, and J. McFadden, "A robust architecture for inference in sensor networks," in *Proc. 4th Int. Symp. Information Processing in Sensor Networks*, Los Angeles, CA, Apr. 2005, pp. 55–62.

Cassio G. Lopes (S'06) received the B.S. degree and the M.S. degree in electrical engineering from the Federal University of Santa Catarina, Brazil, in 1997 and 1999, respectively, and the M.S. degree in electrical engineering from the University of California, Los Angeles (UCLA) in 2004, where he is currently pursuing the Ph.D. degree in electrical engineering at the UCLA Adaptive Systems Laboratory, in distributed adaptive estimation.

From 1999 to 2002, he worked with hardware design for biomedical signals acquisition and transmission over the Internet. He also taught undergraduate-level courses in computer science and electrical engineering. Since 2005, he has been working at the Adaptive Systems Lab/UCLA as a Researcher Staff to develop efficient frequency estimation algorithms for direct-to-earth Mars communications, in cooperation with the Jet Propulsion Lab/NASA. His current research interests are theory and methods for adaptive and statistical signal processing, distributed adaptive estimation, as well as frequency estimation algorithms for space communications.

Ali H. Sayed (S'90–M'92–SM'99–F'01) received the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA, in 1992.

He is Professor and Chairman of Electrical Engineering with the University of California, Los Angeles (UCLA). He is also the Principal Investigator of the UCLA Adaptive Systems Laboratory (http://www.ee.ucla.edu/asl). He has more than 280 journal and conference publications. He is the author of *Fundamentals of Adaptive Filtering* (New York: Wiley, 2003), coauthor of the research monograph *Indefinite Quadratic Estimation and Control* (Philadelphia, PA: SIAM, 1999) and of the graduate-level textbook *Linear Estimation* (Englewood Cliffs, NJ: Prentice-Hall, 2000). He is also coeditor of the volume *Fast Reliable Algorithms for Matrices with Structure* (Philadelphia, PA: SIAM, 1999). He has contributed several articles to engineering and mathematical encyclopedias and handbooks and has served on the program committees of several international meetings. His research interests span several areas, including adaptive and statistical signal processing, filtering and estimation theories, signal processing for communications, interplays between signal processing and control methodologies, system theory, and fast algorithms for large-scale problems.

Dr. Sayed received the 1996 IEEE Donald G. Fink Award, 2002 Best Paper Award from the IEEE Signal Processing Society, 2003 Kuwait Prize in Basic Science, 2005 Frederick E. Terman Award, and is coauthor of two Best Student Paper awards at international meetings (1999, 2001). He is also a member of the Technical Committees on Signal Processing Theory and Methods (SPTM) and on Signal Processing for Communications (SPCOM), both of the IEEE Signal Processing Society. He is a member of the editorial board of the IEEE *Signal Processing Magazine*. He has also served twice as an Associate Editor of the IEEE TRANSACTIONS ON SIGNAL PROCESSING, and as Editor-in-Chief of the same journal during 2003-2005. He is serving as Editor-in-Chief of the *EURASIP Journal on Advances in Signal Processing* and as General Chairman of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2008. He sits on the Board of Governors of the IEEE Signal Processing Society.