Hindawi Publishing Corporation
EURASIP Journal on Advances in Signal Processing
Volume 2009, Article ID 981030, 19 pages
doi:10.1155/2009/981030

Research Article

Performance Analysis of the Consensus-Based Distributed LMS Algorithm

Gonzalo Mateos, Ioannis D. Schizas, and Georgios B. Giannakis

Department of Electrical and Computer Engineering, University of Minnesota, 200 Union Street SE, Minneapolis, MN 55455, USA

Correspondence should be addressed to Georgios B. Giannakis, georgios@umn.edu

Received 15 May 2009; Accepted 8 October 2009

Recommended by Husheng Li

Low-cost estimation of stationary signals and reduced-complexity tracking of nonstationary processes are well motivated tasks than can be accomplished using ad hoc wireless sensor networks (WSNs). To this end, a fully *distributed* least mean-square (D-LMS) algorithm is developed in this paper, in which sensors exchange messages with single-hop neighbors to consent on the network-wide estimates adaptively. The novel approach does not require a Hamiltonian cycle or a special bridge subset of sensors, while communications among sensors are allowed to be noisy. A mean-square error (MSE) performance analysis of D-LMS is conducted in the presence of a time-varying parameter vector, which adheres to a first-order autoregressive model. For sensor observations that are related to the parameter vector of interest via a linear Gaussian model and after adopting simplifying independence assumptions, exact closed-form expressions are derived for the global and sensor-level MSE evolution as well as its steady-state (s.s.) values. Mean and MSE-sense stability of D-LMS are also established. Interestingly, extensive numerical tests demonstrate that for small step-sizes the results accurately extend to the pragmatic setting whereby sensors acquire temporally correlated, not necessarily Gaussian data.

1. Introduction

The advent of wireless sensor networks (WSNs) has created renewed interest in the field of distributed computing, calling for collaborative solutions that enable low-cost estimation of stationary signals as well as reduced-complexity tracking of nonstationary processes. Different from WSN topologies that include a fusion center (FC), *ad hoc* ones are devoid of hierarchies and rely on *in-network* processing to effect agreement among sensors on the estimate of interest. A great body of literature has been amassed in recent years, building-up the field of consensus-based distributed signal processing; the reader is referred to the tutorial in [1] for general results and a vast list of related works. Formidable challenges arise as emergent WSN-based estimation applications demand promptly available, yet accurate local estimates under increasingly restrictive and unpredictable operational constraints. Specifically, often times sensors need to perform estimation in a constantly changing environment without having available a (statistical) model for the underlying processes of interest. This has motivated the development of *distributed adaptive* estimation schemes, generalizing the notion of adaptive filtering to a setup involving networked sensing/processing devices [2, Section I.B].

The first such approach introduced a sequential scheme, whereby information circulation through a topological cycle in conjunction with least mean-square- (LMS-) type adaptive filtering per sensor allows the network to account for time variations in the signal statistics [3]. For more general estimators, a similar stochastic incremental gradient descent algorithm was developed in [4], which subsumes [3] as a special case. While appealing for small-size WSNs, such schemes inherently require a Hamiltonian cycle through which signal estimates are continuously refined. In the eventuality of a sensor failure, determination of a new cycle is an NP-hard problem [5], thus challenging the applicability of incremental schemes in medium- to large-size WSNs. Time-critical applications may encounter additional challenges, since the delay for a local estimate update may be significant as the network; hence, the cycle size scales.

Without topological constraints and by fully exploiting the available links in the network, the so-termed (combine-then-adapt) diffusion LMS [6] offers an improved alternative at the price of increasing communication cost. Performance gains result from interchanging the order of the aforementioned steps, that is, adapt-then-combine [7], leading to the diffusion LMS variant originally proposed in [8]. An alternative to reduce steady-state (s.s.) estimation errors involves diffusing raw sensor observations and regression vectors per neighborhood [7]. This facilitates the flow of new data across the WSN but can degrade performance in the presence of communication noise and essentially doubles the communication cost. Tailored to applications in which fast convergence is at a premium and increased computational burden per sensor can be afforded, distributed recursive least squares (RLS) counterparts can be found in [2, 9, 10].

The present paper develops a fully distributed (D-) LMS-type algorithm, which performs consensus-based, in-network, adaptive estimation for linear regression applications. It is applicable to general ad hoc WSNs, can account for additive communication noise, and does not require circulation through a Hamiltonian cycle. Different from [11] and the model-based distributed Kalman trackers in [12–14], D-LMS can be applied to a wide class of distributed estimation tasks as it requires no knowledge of the sensor data model. The algorithm is derived in Section 3, by minimizing a separable reformulation of the convex LMS cost using distributed optimization techniques; namely the alternating-direction method of multipliers (AD-MoM) [15, page 253]. Relative to the D-LMS variant in [16], the present reformulation of the LMS cost circumvents the requirement of the special type of sensors comprising the so-called bridge sensor subset; see also [13, 17]. As a byproduct, this approach results in a fully distributed algorithm whereby all sensors perform identical tasks, without introducing hierarchies that may require intricate recovery protocols to cope with sensor failures. Utilization of a constant step-size endows D-LMS with tracking capabilities, without hurting its resilience to communication noise. This is desirable in a constantly changing environment, within which WSNs are envisioned to operate. Interestingly, it is shown in Section 3.2 that whenever the use of powerful channel codes renders inter-sensor links virtually noise-free, the D-LMS algorithm can be modified to achieve an identical communication overhead, at improved convergence rates with respect to (w.r.t.) [6] as illustrated via extensive numerical simulations.

A main contribution of the present paper pertains to a detailed mean-square error (MSE) *tracking performance analysis* for D-LMS (Section 5). Evaluating the performance of (centralized) adaptive filters in nonstationary environments is a challenging problem in its own right; prior art is surveyed in, for example, [18], [19, page 120], [20, page 357] and the extensive list of references therein. To the best of our knowledge, this paper conducts a tracking analysis for the first time in the context of distributed adaptive algorithms used by WSNs. This setting introduces unique challenges in the analysis such as space-time sensor data and multiple sources of additive noise, a consequence of imperfect sensors and communication links. The approach pursued here capitalizes on an equivalent representation of the local recursions comprising D-LMS, as a global dynamical system described by a difference-equation derived in Section 4.1. The covariance matrix of the resulting state is then shown to encompass all the information needed to evaluate the relevant global and sensor-level performance metrics (Section 4.2). Alternative analysis techniques include the energy-conservation approach in [21], [20, page 287] and stochastic averaging [19, page 229]. For performance analysis of distributed adaptive algorithms seeking *time-invariant* parameters, the former has been applied in [3, 6, 7], while the latter can be found in [16].

For a *time-varying* parameter fluctuating as a first-order autoregressive [AR(1)] process, and sensor observations that are linearly related to it, the simplifying independence Gaussian setting assumptions [19, page 110], [20, page 448] are key enablers towards deriving exact closed-form expressions for the MSE evolution and its s.s. value (Section 5.2). Mean and MSE stability are also established, revealing sufficient conditions under which s.s. is attained. The AR(1) model subsumes a time-invariant parameter as a special case, and performance results for the stationary case are readily obtained as a byproduct. Hence, the estimation/tracking capabilities of D-LMS in the presence of: (i) time-invariant; and (ii) time-varying parameters can be contrasted in a unified fashion. Of particular interest in these two scenarios are the corresponding s.s. MSE versus step-size characteristics, which reveal fundamental insights and differences similar to those observed in the classic LMS algorithm (Section 5.3). All in all, the importance of the aforementioned results is threefold: (i) an exact tracking MSE characterization is provided for D-LMS; (ii) for the stationary case and ideal inter-sensor links, similar results for the diffusion LMS algorithm [6] lay a common ground for fair comparisons; and (iii) for small step-sizes the conclusions extend to temporally correlated (non-)Gaussian data. Numerical tests corroborating the theoretical findings of this paper are presented in Section 6, while concluding remarks are given in Section 7.

Notation 1. Operators $\otimes$, $\circ$, $(\cdot)^T$, $(\cdot)^\dagger$, $\lambda_{\max}(\cdot)$, $\operatorname{tr}(\cdot)$, $\operatorname{diag}(\cdot)$, $b\operatorname{diag}(\cdot)$, $E[\cdot]$, $\operatorname{vec}[\cdot]$ will denote Kronecker product, Hadamard product, transposition, matrix pseudo-inverse, spectral radius, matrix trace, diagonal matrix, block diagonal matrix, expectation, and matrix vectorization, respectively. Functions $\max(\cdot,\cdot)$ and $\min(\cdot,\cdot)$ respectively denote the maximum and minimum value of their scalar arguments. For both vector and matrices, $\|\cdot\|$ will stand for the 2-norm and $|\cdot|$ for the cardinality of a set. Positive definite matrices will be denoted by $\mathbf{M} \succ \mathbf{0}$. The $n \times n$ identity matrix will be represented by $\mathbf{I}_n$, while $\mathbf{1}_n$ will denote the $n \times 1$ vector of all ones and $\mathbf{1}_{n\times m} := \mathbf{1}_n\mathbf{1}_m^T$. Similar notation will be adopted for vectors (matrices) of all zeros. For matrix $\mathbf{M} \in \mathbb{R}^{m\times n}$, $\operatorname{range}(\mathbf{M}) := \{\mathbf{y} \in \mathbb{R}^m : \mathbf{y} = \mathbf{M}\mathbf{x}$ for some $\mathbf{x} \in \mathbb{R}^n\}$ and $\operatorname{nullspace}(\mathbf{M}) := \{\mathbf{x} \in \mathbb{R}^n : \mathbf{M}\mathbf{x} = \mathbf{0}_m\}$. The ith vector in the canonical basis for $\mathbb{R}^n$ will be denoted by $\mathbf{b}_{n,i}$, $i = 1, \ldots, n$.

2. Network Model and Estimation Problem Statement

Consider an ad hoc WSN with sensors $\{1,\ldots,J\} := \mathcal{J}$. To effect energy-aware communications, sensor j communicates only with its (nearby) single-hop neighbors in $\mathcal{N}_j \subseteq \mathcal{J}$. Under the natural assumption of symmetric inter-sensor links, the WSN is modeled as an undirected graph $G(\mathcal{V},\mathcal{E})$. The vertices in $\mathcal{V}$ correspond to the sensors and hence are biunivocally mapped to the elements of $\mathcal{J}$, while the edges in $\mathcal{E}$ represent the available wireless links. Global connectivity information is captured by the symmetric adjacency matrix $\mathbf{E} \in \mathbb{R}^{J\times J}$, where $[\mathbf{E}]_{ij} = 1$ if $i \in \mathcal{N}_j$ and $[\mathbf{E}]_{ij} = 0$ otherwise. By convention $[\mathbf{E}]_{jj} = 1$ for all $j \in \mathcal{J}$ so that G is not simple. The graph Laplacian $\mathbf{L} \in \mathbb{R}^{J\times J}$ will be useful in the sequel, where $\mathbf{L} := \mathbf{D} - \mathbf{E}$, and $\mathbf{D} := \operatorname{diag}(|\mathcal{N}_1|,\ldots,|\mathcal{N}_J|)$. The adjustment of the sensors' transmission power as well as the initial WSN deployment are assumed to render G connected. This is a minimum requirement ensuring that sensors' data can percolate across the whole WSN. Different from [3, 4, 6], the present network model accounts explicitly for non-ideal sensor-to-sensor links, through a zero-mean additive noise vector $\boldsymbol{\eta}_j^i(t)$ with covariance matrix $\mathbf{R}_{\eta_{j,i}} := E[\boldsymbol{\eta}_j^i(t)\boldsymbol{\eta}_j^i(t)^T]$ corrupting signals received at sensor j from sensor i at discrete-time instant t. The noise vectors $\{\boldsymbol{\eta}_j^i(t)\}_{j\in\mathcal{J}}^{i\in\mathcal{N}_j}$ are assumed temporally and spatially uncorrelated.

The WSN is deployed to estimate a signal vector $\mathbf{s}_0(t) \in \mathbb{R}^{p\times 1}$ in a collaborative fashion subject to single-hop communication constraints, by resorting to the linear LMS criterion; see, for example, [20, page 171]. Per time instant $t = 0, 1, 2, \ldots$, each sensor has available a regression vector $\mathbf{h}_j(t) \in \mathbb{R}^{p\times 1}$ and a scalar observation $x_j(t)$, both assumed zero-mean without loss of generality. The network-wide snapshot of data at time instant t can be compactly represented in the global vector $\mathbf{x}(t) := [x_1(t)\cdots x_J(t)]^T \in \mathbb{R}^{p\times 1}$ and regression matrix $\mathbf{H}(t) := [\mathbf{h}_1(t)\cdots\mathbf{h}_J(t)]^T \in \mathbb{R}^{J\times p}$. A similar data setting was considered in [6]. The global LMS estimator of interest can be written as [3, 6, 16]

$$\begin{aligned}\hat{\mathbf{s}}(t) &= \arg\min_{\mathbf{s}} E\left[\|\mathbf{x}(t)-\mathbf{H}(t)\mathbf{s}\|^2\right]\\ &= \arg\min_{\mathbf{s}} \sum_{j=1}^{J} E\left[\left(x_j(t)-\mathbf{h}_j^T(t)\mathbf{s}\right)^2\right].\end{aligned} \tag{1}$$

Suitable reformulation may be needed in order to acquire $\{\mathbf{h}_j(t)\}_{j\in\mathcal{J}}$ based on the available information across sensors. There are no general guidelines to this end, which is dictated by the specific estimation/tracking problem at hand. For example, in target tracking applications where sensors rely on power or range measurements, the nonlinear data models must be linearized before obtaining regressors as a function of sensor observations; see, for example, [22, page 137]. Another possibility is to obtain the regression vectors from the physics of the problem, using standard kinematic models; see, for example, [22, Chapter 6]. A distributed power spectrum estimation problem was described in [16, Remark 1], where regressors are formed by stacking the last p local sensor observations.

Remark 1 (Motivation for adaptive processing). The gradient of the differentiable cost in (1) depends on the (cross-) covariances $\mathbf{R}_H(t) := E[\mathbf{H}^T(t)\mathbf{H}(t)]$ and $\mathbf{r}_{Hx}(t) := E[\mathbf{H}^T(t)\mathbf{x}(t)]$. Any attempt to iteratively obtain $\hat{\mathbf{s}}(t)$ via gradient-based optimization algorithms will require knowledge of $\mathbf{R}_H(t)$ and $\mathbf{r}_{Hx}(t)$. In many WSN applications however, this information may be either unavailable or time-varying, and thus impossible to acquire continuously. Tuned with the reduced-complexity requirements of WSNs, the arguably simplest approach involves approximating the expectations coarsely via instantaneous realizations of the sensor data as in the classical LMS, that is, $\mathbf{R}_H(t) \approx \mathbf{H}^T(t)\mathbf{H}(t)$ and $\mathbf{r}_{Hx}(t) \approx \mathbf{H}^T(t)\mathbf{x}(t)$.

This exemplification of the adaptive step in turn leads to centralized stochastic (noisy) gradient iterations, which one expects to perform well after sufficient data are acquired and the unknown statistics are learnt. Still, the challenge is to enable such learning mechanisms even when data is not centrally available, that is, when entries of $\mathbf{x}(t)$ and rows of $\mathbf{H}(t)$ are scattered across the WSN. In this context, the present paper aims to develop a fully distributed LMS-type algorithm, which relies on in-network, adaptive processing of the available information across the WSN.

3. The D-LMS Algorithm

In this section we introduce the D-LMS algorithm, first going through the algorithmic construction steps and salient features of its operation. The approach followed includes three main building blocks: (i) recast (1) into an equivalent separable form which facilitates distributed implementation; (ii) split the optimization problem into simpler subtasks executed locally at each sensor; and (iii) invoke a stochastic approximation iteration to obtain an adaptive LMS-type algorithm that can both handle the unavailability of (cross-) covariance information, and also remain robust to signal variations. Important differences w.r.t. the related approach in [16] are encountered in steps (i) and (ii); see also Remark 2 for a summary of the merits of the present contribution relative to [16].

To distribute the cost in (1), replace the global variable $\mathbf{s}$ which couples the per-sensor summands with auxiliary local variables $\mathbf{s} := \{\mathbf{s}_j\}_{j=1}^{J}$ representing candidate estimates of $\mathbf{s}$ per sensor. These local estimates are utilized to reformulate (1) as the following convex *constrained* minimization problem:

$$\begin{aligned}\left\{\hat{\mathbf{s}}_j(t)\right\}_{j=1}^{J} &= \arg\min_{\mathbf{s}} \sum_{i=1}^{J} E\left[\left(x_i(t)-\mathbf{h}_i^T(t)\mathbf{s}_i\right)^2\right],\\ &\text{s.t. } \mathbf{s}_j = \mathbf{s}_{j'}, \quad j \in \mathcal{J},\ j' \in \mathcal{N}_j.\end{aligned} \tag{2}$$

The set of equality constraints in (2) involves variables of neighboring sensors only, forcing an agreement across each sensor's neighborhood. If the WSN graph G is connected, these constraints impose *network-wide* consensus a fortiori, that is, $\mathbf{s}_j = \mathbf{s}_{j'}$ for all $j, j' \in \mathcal{J}$. As an immediate consequence, one finds that the optimization problems (1) and (2) are equivalent in the sense that their optimal solutions coincide; that is, $\hat{\mathbf{s}}_j(t) = \hat{\mathbf{s}}(t)$, for all $j \in \mathcal{J}$.

3.1. Algorithm Construction. In order to tackle (2) in a distributed fashion, we will resort to the AD-MoM [15, page 253]. Towards this end, consider the auxiliary variables $\mathbf{z} := \{\mathbf{z}_j^{j'}\}_{j \in \mathcal{J}}^{j' \in \mathcal{N}_j}$ and replace the constraints in (2) with the equivalent ones

$$\mathbf{s}_j = \mathbf{z}_j^{j'}, \quad \mathbf{s}_{j'} = \mathbf{z}_j^{j'}, \quad j \in \mathcal{J}, \quad j' \in \mathcal{N}_j, \quad j \neq j'. \tag{3}$$

The sole purpose of the variables $\mathbf{z}$ is to facilitate application of the AD-MoM, and they will be eventually eliminated due to their inherent redundancy. Next, associate Lagrange multipliers $[\mathbf{v}, \mathbf{u}] := \{\mathbf{v}_j^{j'}, \mathbf{u}_j^{j'}\}_{j \in \mathcal{J}}^{j' \in \mathcal{N}_j}$ with the constraints in (3), and form the quadratically augmented Lagrangian function

$$\begin{aligned}
&\mathcal{L}_a[\mathbf{s}, \mathbf{z}, \mathbf{v}, \mathbf{u}] \\
&\quad = \sum_{j=1}^{J} E\left[\left(x_j(t+1) - \mathbf{h}_j^T(t+1)\mathbf{s}_j\right)^2\right] \\
&\qquad + \sum_{j=1}^{J} \sum_{j' \in \mathcal{N}_j} \left[\left(\mathbf{v}_j^{j'}\right)^T \left(\mathbf{s}_j - \mathbf{z}_j^{j'}\right) + \left(\mathbf{u}_j^{j'}\right)^T \left(\mathbf{s}_j - \mathbf{z}_{j'}^{j}\right)\right] \\
&\qquad + \sum_{j=1}^{J} \sum_{j' \in \mathcal{N}_j} \frac{c}{2}\left[\left\|\mathbf{s}_j - \mathbf{z}_j^{j'}\right\|^2 + \left\|\mathbf{s}_j - \mathbf{z}_{j'}^{j}\right\|^2\right],
\end{aligned} \tag{4}$$

where $c > 0$ is a penalty coefficient. Sensor j will locally store and update a total of $3|\mathcal{N}_j| + 1$ vectors in $\mathbb{R}^{p \times 1}$, namely $\mathbf{s}_j$ and $\{\mathbf{z}_j^{j'}, \mathbf{v}_j^{j'}, \mathbf{u}_j^{j'}\}_{j' \in \mathcal{N}_j}$. We reiterate however, that in the process of deriving the local updating recursions many of these variables will turn out to be redundant.

The AD-MoM entails an iterative process comprising three steps per time instant $t = 0, 1, 2, \ldots$.

[S1] *Multiplier updates:*

$$\mathbf{v}_j^{j'}(t) = \mathbf{v}_j^{j'}(t-1) + c\left(\mathbf{s}_j(t) - \mathbf{z}_j^{j'}(t)\right), \quad j \in \mathcal{J}, j' \in \mathcal{N}_j, \tag{5}$$

$$\mathbf{u}_j^{j'}(t) = \mathbf{u}_j^{j'}(t-1) + c\left(\mathbf{s}_j(t) - \mathbf{z}_{j'}^{j}(t)\right), \quad j \in \mathcal{J}, j' \in \mathcal{N}_j. \tag{6}$$

[S2] *Local estimate updates:*

$$\mathbf{s}(t+1) = \arg\min_{\mathbf{s}} \mathcal{L}_a[\mathbf{s}, \mathbf{z}(t), \mathbf{v}(t), \mathbf{u}(t)]. \tag{7}$$

[S3] *Auxiliary variable updates:*

$$\mathbf{z}(t+1) = \arg\min_{\mathbf{z}} \mathcal{L}_a[\mathbf{s}(t+1), \mathbf{z}, \mathbf{v}(t), \mathbf{u}(t)]. \tag{8}$$

The multiplier recursions in [S1] correspond to gradient ascent iterations seeking the optimal dual prices, and are customary in various methods of multipliers [15, Chapter 3]. On the other hand, [S2] and [S3] represent block coordinate descent steps which respectively update $\mathbf{s}$ and $\mathbf{z}$ in a cyclic fashion. At each step while minimizing the augmented Lagrangian, the variables not being updated are treated as fixed constants and substituted with their most up to date values. Interestingly, it is shown in Appendix A that [*S*1]–[*S*3] boil down to the following simple set of subtasks carried out locally at each sensor $j \in \mathcal{J}$,

$$\mathbf{v}_j^{j'}(t) = \mathbf{v}_j^{j'}(t-1) + \frac{c}{2}\left(\mathbf{s}_j(t) - \mathbf{s}_{j'}(t)\right), \quad j' \in \mathcal{N}_j, \tag{9}$$

$$\begin{aligned}
\mathbf{s}_j(t+1) = \arg\min_{\mathbf{s}_j} \Bigg[& E\left[\left(x_j(t+1) - \mathbf{h}_j^T(t+1)\mathbf{s}_j\right)^2\right] \\
& + \sum_{j' \in \mathcal{N}_j} \left[\mathbf{v}_j^{j'}(t) - \mathbf{v}_{j'}^{j}(t)\right]^T \mathbf{s}_j \\
& + c \sum_{j' \in \mathcal{N}_j} \left\|\mathbf{s}_j - \frac{1}{2}\left(\mathbf{s}_j(t) + \mathbf{s}_{j'}(t)\right)\right\|^2 \Bigg].
\end{aligned} \tag{10}$$

As promised, the set of auxiliary variables $[\mathbf{z}, \mathbf{u}]$ have been eliminated; and each sensor, say the jth, has to store and update only $(|\mathcal{N}_j| + 1)p$ scalars. To carry out the unconstrained minimization in (10), observe that the cost is convex and differentiable. Thus, the first-order necessary condition is also sufficient for optimality. Computing the gradient with respect to $\mathbf{s}_j$ and setting the result equal to zero, yields

$$\begin{aligned}
E\Bigg[& -2\mathbf{h}_j(t+1)\left(x_j(t+1) - \mathbf{h}_j^T(t+1)\mathbf{s}_j\right) \\
& + \sum_{j' \in \mathcal{N}_j} \left\{\mathbf{v}_j^{j'}(t) - \mathbf{v}_{j'}^{j}(t) + 2c\left(\mathbf{s}_j - \frac{1}{2}\left(\mathbf{s}_j(t) + \mathbf{s}_{j'}(t)\right)\right)\right\}\Bigg] = \mathbf{0}_p.
\end{aligned} \tag{11}$$

It is now apparent that $\mathbf{s}_j(t+1)$ can be obtained as the root of an equation of the form $\mathbf{f}(\mathbf{s}_j) := E[\boldsymbol{\varphi}(\mathbf{s}_j, x_j(t+1), \mathbf{h}_j(t+1))] = \mathbf{0}_p$, where $\boldsymbol{\varphi}$ stands for the function inside the expectation in (11) and corresponds to the stochastic gradient of the cost in (10). In lieu of local (cross-) covariance information, namely $\mathbf{r}_{h_j x_j} := E[\mathbf{h}_j(t+1)x_j(t+1)]$ and $\mathbf{R}_{h_j} := E[\mathbf{h}_j(t+1)\mathbf{h}_j^T(t+1)]$, the root of $\mathbf{f}(\mathbf{s}_j) = \mathbf{0}_p$ is not computable in closed form since $\mathbf{f}$ is unknown. Hence, motivated by stochastic approximation techniques (such as the celebrated Robbins-Monro algorithm; see e.g., [23, Chapter 1]) which iteratively find the root of an unknown function $\mathbf{f}(\mathbf{s}_j)$ given a time-seriesof noisy observations $\{\boldsymbol{\varphi}(\mathbf{s}_j(t), x_j(t+1), \mathbf{h}_j(t+1))\}_{t=0}^{\infty}$, the proposed recursion for every $j \in \mathcal{J}$ is

$$\mathbf{s}_j(t+1) = \mathbf{s}_j(t) + \mu_j \Bigg[2\mathbf{h}_j(t+1)e_j(t+1) - \sum_{j'\in\mathcal{N}_j}\left(\mathbf{v}_j^{j'}(t) - \mathbf{v}_{j'}^{j}(t)\right) - c\sum_{j'\in\mathcal{N}_j}\left(\mathbf{s}_j(t)-\mathbf{s}_{j'}(t)\right)\Bigg], \tag{12}$$

where μ_j is a constant step-size and $e_j(t+1) := x_j(t+1) - \mathbf{h}_j^T(t+1)\mathbf{s}_j(t)$ is the local *a priori* error. Recursions (9) and (12) are tabulated as Algorithm 1, and constitute the D-LMS algorithm that can be arbitrarily initialized. To capture the effects of receiving noise corrupting the variables exchanged among neighboring sensors, the D-LMS recursions are modified to yield

$$\mathbf{v}_j^{j'}(t) = \mathbf{v}_j^{j'}(t-1) + \frac{c}{2}\left(\mathbf{s}_j(t) - \left(\mathbf{s}_{j'}(t) + \boldsymbol{\eta}_j^{j'}(t)\right)\right), \quad j' \in \mathcal{N}_j, \tag{13}$$

$$\mathbf{s}_j(t+1) = \mathbf{s}_j(t) + \mu_j \Bigg[2\mathbf{h}_j(t+1)e_j(t+1) - \sum_{j'\in\mathcal{N}_j}\left(\mathbf{v}_j^{j'}(t) - \left(\mathbf{v}_{j'}^{j}(t) + \bar{\boldsymbol{\eta}}_j^{j'}(t)\right)\right) - c\sum_{j'\in\mathcal{N}_j}\left(\mathbf{s}_j(t) - \left(\mathbf{s}_{j'}(t) + \boldsymbol{\eta}_j^{j'}(t)\right)\right)\Bigg], \tag{14}$$

where $\boldsymbol{\eta}_j^{j'}(t)$ and $\bar{\boldsymbol{\eta}}_j^{j'}(t)$ denote the additive communication noise present in the reception of $\mathbf{s}_{j'}(t)$ and $\mathbf{v}_{j'}^{j}(t)$ at sensor j, respectively. In detail, during time instant $t+1$ sensor j receives the local estimates $\{\mathbf{s}_{j'}(t) + \boldsymbol{\eta}_j^{j'}(t)\}_{j'\in\mathcal{N}_j}$ and plugs them into (13) to evaluate $\mathbf{v}_j^{j'}(t)$ for $j' \in \mathcal{N}_j$. Each one of the updated local Lagrange multipliers $\{\mathbf{v}_j^{j'}(t)\}_{j'\in\mathcal{N}_j}$ is subsequently transmitted to the corresponding neighbor $j' \in \mathcal{N}_j$. Then, upon reception of $\{\mathbf{v}_{j'}^{j}(t) + \bar{\boldsymbol{\eta}}_j^{j'}(t)\}_{j'\in\mathcal{N}_j}$, the multipliers are jointly used along with $\{\mathbf{s}_{j'}(t) + \boldsymbol{\eta}_j^{j'}(t)\}_{j'\in\mathcal{N}_j}$ and the newly acquired local data $\{x_j(t+1), \mathbf{h}_j(t+1)\}$ to obtain $\mathbf{s}_j(t+1)$ via (14). The $(t+1)$-st iteration is concluded after sensor j broadcasts $\mathbf{s}_j(t+1)$ to its neighbors.

The use of a constant step-size μ_j endows D-LMS with tracking capabilities. This is desirable in a constantly changing environment, within which WSNs are envisioned to operate. Some related consensus-based estimation approaches compromise adaptability, by introducing a diminishing step-size to suppress the error-propagation effects of communication noise; see, for example, [24, 25]. D-LMS is shown to be robust against communication noise in Section 5, a property directly inherited from the AD-MoM; see also [17] for related claims in single-shot non-adaptive distributed estimation.

```
Arbitrarily initialize $\{\mathbf{s}_j(0)\}_{j=1}^{J}$ and $\{\mathbf{v}_j^{j'}(-1)\}_{j\in\mathcal{J}}^{j'\in\mathcal{N}_j}$.
for $t = 0, 1, \ldots$ do
    All $j \in \mathcal{J}$: transmit $\mathbf{s}_j(t)$ to neighbors in $\mathcal{N}_j$.
    All $j \in \mathcal{J}$: update $\{\mathbf{v}_j^{j'}(t)\}_{j'\in\mathcal{N}_j}$ using (13).
    All $j \in \mathcal{J}$: transmit $\mathbf{v}_j^{j'}(t)$ to each $j' \in \mathcal{N}_j$.
    All $j \in \mathcal{J}$: update $\mathbf{s}_j(t+1)$ using (14).
end for
```

Algorithm 1: D-LMS.

Remark 2 (Comparison with [16]). A similar consensus-based LMS algorithm was put forth in [16]. To enable task parallelization via AD-MoM while ensuring that estimates agree across the whole WSN, the approach in [16] reformulates (1) by relying on the so called bridge sensor subset. Not only setting-up—but also readjusting the bridge sensor set, for example, when sensors inevitably fail in battery-limited WSN deployments—requires additional coordination among sensors with an associated communication overhead. Compared to [16], the approach followed here does not require such a bridge sensor set, and in this sense, it offers a fully distributed, robust, and resource efficient LMS-type algorithm for use in ad hoc WSNs. The contributions in this paper are relevant to the D-LMS variant in [16] too, as the performance analysis in the forthcoming sections carries over with minor adjustments; see also [26].

3.2. D-LMS Algorithm with Ideal Links. Consider an ideal scenario whereby sensors are able to communicate via error-free links. Such an operational setup may arise, for example, whenever the use of powerful channel codes renders inter-sensor links virtually noise-free. Next, we show that under such assumptions, D-LMS can be simplified to yield a set of local recursions which are equivalent to (9) and (12), while incurring notably lower communication and reduced computational complexities w.r.t. to the original version of the algorithm.

Specifically, note first that if the Lagrange multipliers $\mathbf{v}$ are initialized such that $\mathbf{v}_j^{j'}(-1) = -\mathbf{v}_{j'}^{j}(-1)$ with $j \in \mathcal{J}$ and $j' \in \mathcal{N}_j$, then in the absence of communication noise one finds from (9) that $\mathbf{v}_j^{j'}(t) = -\mathbf{v}_{j'}^{j}(t)$ for all $t \geq 0$. By relying on this identity and starting from (12), we arrive at a simplified recursion to update the local estimates $\mathbf{s}_j(t+1)$ for all $j \in \mathcal{J}$

$$\mathbf{s}_j(t+1) = \mathbf{s}_j(t) + \mu_j \Bigg[2\mathbf{h}_j(t+1)e_j(t+1) - 2\sum_{j'\in\mathcal{N}_j}\mathbf{v}_j^{j'}(t) - c\sum_{j'\in\mathcal{N}_j}\left(\mathbf{s}_j(t) - \mathbf{s}_{j'}(t)\right)\Bigg]. \tag{15}$$

The specific initialization requirement for the multipliers is not restrictive, as it can be readily satisfied by selecting $\mathbf{v}_j^{j'}(-1) = \mathbf{0}_p$ for $j \in \mathcal{J}$ and $j\prime \in \mathcal{N}_j$ without the need of extra coordination among sensors.

The second summand inside the square brackets in (15) incorporates only the local multipliers $\{\mathbf{v}_j^{j'}(t)\}_{j\in\mathcal{N}_j}$ stored at sensor j. Hence, multipliers need not be communicated to the neighboring sensors at all. What is more, multipliers enter (15) only through their local sum across $j' \in \mathcal{N}_j$, so that there is no need to keep track of them separately. This motivates introducing the new set of local variables $\mathbf{p}_j(t) := 2\sum_{j'\in\mathcal{N}_j} \mathbf{v}_j^{j'}(t)$ for $j = 1,\ldots,J$, which have to be updated in conjunction with $\mathbf{s}_j(t)$. The updating rule for $\mathbf{p}_j(t)$ follows immediately from (9), and the final recursions per sensor $j \in \mathcal{J}$ are

$$\mathbf{p}_j(t) = \mathbf{p}_j(t-1) + c \sum_{j'\in\mathcal{N}_j} \left(\mathbf{s}_j(t) - \mathbf{s}_{j'}(t)\right) \tag{16}$$

$$\begin{aligned}\mathbf{s}_j(t+1) = \mathbf{s}_j(t) + \mu_j \Bigg[& 2\mathbf{h}_j(t+1)e_j(t+1) - \mathbf{p}_j(t) \\ & -c \sum_{j'\in\mathcal{N}_j} \left(\mathbf{s}_j(t) - \mathbf{s}_{j'}(t)\right)\Bigg].\end{aligned} \tag{17}$$

Interestingly, (16)-(17) require each sensor to store and update only $2p$ scalars, regardless of the WSN topology and corresponding neighborhood sizes. While diffusion LMS [6] needs half the number of scalar recursions, in D-LMS (cf. (13)-(14)) sensor j has to store and update $(|\mathcal{N}_j| + 1)p$ scalars. With regards to communication cost incurred by the D-LMS variant in (16)-(17), on a per-iteration basis, sensor j receives $|\mathcal{N}_j|p$ scalars due to $\{\mathbf{s}_{j'}(t)\}_{j'\in\mathcal{N}_j}$ and transmits p scalars corresponding to $\mathbf{s}_j(t+1)$; exactly as in diffusion LMS. Recall that D-LMS in (13)-(14) incurs an additional cost of $2|\mathcal{N}_j|p$ communicated scalars due to the reception and transmission of $\{\mathbf{v}_{j'}^{j}(t)\}_{j'\in\mathcal{N}_j}$ and $\{\mathbf{v}_j^{j'}(t)\}_{j'\in\mathcal{N}_j}$, respectively. Recursions (16)-(17) (also tabulated as Algorithm 2) are equivalent to D-LMS whenever $\boldsymbol{\eta}_j^{j'}(t) = \bar{\boldsymbol{\eta}}_j^{j'}(t) = \mathbf{0}_p$ in (13)-(14). Hence, both will achieve identical convergence rates and estimation performance, making Algorithm 2 the most attractive alternative when noise is not an issue as corroborated via the numerical tests in Section 6. However, there is a price paid for the reduced amount of communications and computational savings as clarified on the ensuing remark.

Remark 3 (Communication noise resilience). The D-LMS variant in (16)-(17) is only applicable when communication links are ideal. Being equivalent to D-LMS under this assumption, one might still be tempted to replace $\{\mathbf{s}_{j'}(t)\}_{j'\in\mathcal{N}_j}$ with $\{\mathbf{s}_{j'}(t) + \boldsymbol{\eta}_j^{j'}(t)\}_{j'\in\mathcal{N}_j}$ in recursions (16)-(17) to capture the effects of noise corrupting the exchanged local estimates. As it turns out, in the process of running (16)-(17) noise will accumulate resulting in local estimates whose variance grows unbounded as $t \to \infty$. The reduced communication overhead is thus counterbalanced by the lack of resilience in the presence of communication noise. As a byproduct, this highlights the key role played by the Lagrange multiplier exchanges in rendering D-LMS—and generally all MoM-based distributed algorithms—robust to communication noise; see also [17] for further details.

4. Analysis Preliminaries

Our approach to performance analysis relies on a compact *error-form* representation of D-LMS as a linear time-varying stochastic difference equation. As discussed in this section, the resulting estimation error covariance matrix encompasses all the information needed to evaluate the relevant performance metrics; namely MSE, excess mean-square error (EMSE) and mean-square deviation (MSD). The aforementioned figures of merit ultimately assess the performance of D-LMS, both on a per-sensor basis and collectively by considering the WSN as a whole.

4.1. Error-Form D-LMS. In this subsection, we start from the D-LMS recursions in (13)-(14) and characterize the evolution of the local estimation errors $\{\mathbf{y}_{1,j}(t) := \mathbf{s}_j(t) - \mathbf{s}_0(t)\}_{j=1}^J$ and multiplier-based quantities $\{\mathbf{y}_{2,j}(t) := \sum_{j'\in\mathcal{N}_j}(\mathbf{v}_j^{j'}(t-1) - \mathbf{v}_{j'}^{j}(t-1))\}_{j=1}^J$. It turns out that a convenient global state capturing the spatio-temporal dynamics of D-LMS can be defined as $\mathbf{y}(t) := [\mathbf{y}_1^T(t)\ \mathbf{y}_2^T(t)]^T = [\mathbf{y}_{1,1}^T(t)\cdots\mathbf{y}_{1,J}^T(t)\ \mathbf{y}_{2,1}^T(t)\cdots\mathbf{y}_{2,J}^T(t)]^T$. While the need of the local errors within $\mathbf{y}_1(t)$ is apparent, augmentation with the seemingly unnecessary $\mathbf{y}_2(t)$ will prove useful to obtain a simple, first-order difference equation for $\mathbf{y}(t)$. Otherwise, a first-order recursion for $\mathbf{y}_1(t)$ is impossible. In order to proceed, we shall require for all $j \in \mathcal{J}$ that

(a1) *Sensor observations obey $x_j(t) = \mathbf{h}_j^T(t)\mathbf{s}_0(t-1) + \epsilon_j(t)$, where the zero-mean white noise $\{\epsilon_j(t)\}$ has variance $\sigma_{\epsilon_j}^2$.*

Linear models are commonly used throughout the adaptive signal processing literature to facilitate stability and performance analysis, for example, [20], [19, Chapters 5,9], [3, 6, 16]. Observation noise variances can differ across sensors, accounting for faulty sensing devices presumably leading to larger values of $\sigma_{\epsilon_j}^2$.

Remark 4 (Sensor data assumptions). An attractive feature of D-LMS is that it can be applied to a wide class of distributed linear regression problems. Indeed, D-LMS does not require prior knowledge of a data model to describe the sensor observations, as the underlying process statistics are learnt "on-the-fly". In this sense, D-LMS differs from the distributed Kalman filtering approaches in [12–14], which are only applicable when state and observation models are available locally at each sensor. When it comes to stability and performance evaluation however, a meaningful "ground-truth" model should be adopted to carry out the analysis and enable fair comparison among competing alternatives. It is true that assumption (a1) delimitates the scope of the forthcoming analysis, though by no means restrains D-LMS from being applied in broader settings.

To concisely capture the effects of both observation and communication noise on the estimation errors across the WSN, define the $Jp \times 1$ noise vectors $\boldsymbol{\epsilon}(t) := 2\mu[\mathbf{h}_1^T(t)\epsilon_1(t)\cdots\mathbf{h}_J^T(t)\epsilon_J(t)]^T$ and $\bar{\boldsymbol{\eta}}(t) :=$

$[\overline{\boldsymbol{\eta}}_1^T(t) \cdots \overline{\boldsymbol{\eta}}_J^T(t)]^T$, where vectors $\{\overline{\boldsymbol{\eta}}_j(t)\}_{j\in\mathcal{J}}$ are given by

$$\overline{\boldsymbol{\eta}}_j(t) := \sum_{j'\in\mathcal{N}_j} \overline{\boldsymbol{\eta}}_j^{j'}(t) \tag{18}$$

and $\mathbf{R}_{\overline{\eta}} := E[\overline{\boldsymbol{\eta}}(t)\overline{\boldsymbol{\eta}}^T(t)]$. The vector in (18) amounts to the accumulated communication noise at sensor j, due to the reception of all required multipliers at time t, namely $\{\mathbf{v}_{j'}^j(t)\}_{j'\in\mathcal{N}_j}$. Next, introduce the $p(\sum_{j=1}^J |\mathcal{N}_j|) \times 1$ vector

$$\boldsymbol{\eta}(t) := \left[\left\{\left(\boldsymbol{\eta}_{j'}^1(t)\right)^T\right\}_{j'\in\mathcal{N}_1} \cdots \left\{\left(\boldsymbol{\eta}_{j'}^J(t)\right)^T\right\}_{j'\in\mathcal{N}_J}\right]^T \tag{19}$$

which comprises the receiver noises corrupting transmissions of local estimates across the whole network at time instant t, and define $\mathbf{R}_\eta := E[\boldsymbol{\eta}(t)\boldsymbol{\eta}^T(t)]$. Finally, let $\mathbf{L}_c := c\mathbf{L} \otimes \mathbf{I}_p \in \mathbb{R}^{Jp\times Jp}$ be a matrix capturing the WSN topology through the (scaled) graph Laplacian $\mathbf{L}$, and arrange the instantaneous outer products of regression vectors in $\mathbf{R}_h(t) := b\text{diag}(\mathbf{h}_1(t)\mathbf{h}_1^T(t), \ldots, \mathbf{h}_J(t)\mathbf{h}_J^T(t)) \in \mathbb{R}^{Jp\times Jp}$. Based on these definitions and assuming for simplicity in exposition that $\mu_j = \mu$ for all $j \in \mathcal{J}$, the following instrumental lemma is established in Appendix B.

Lemma 5. *Under (a1) and for $t \geq 0$, the global state $\mathbf{y}(t)$ evolves according to*

$$\begin{aligned}\mathbf{y}(t+1) = {}& b\text{diag}\left(\mathbf{I}_{Jp}, \mathbf{L}_c\right)\mathbf{z}(t+1) + \begin{bmatrix}\mu\mathbf{I}_{Jp}\\ \mathbf{0}_{Jp\times Jp}\end{bmatrix}\overline{\boldsymbol{\eta}}(t)\\ &+ \begin{bmatrix}\mu\left(3\mathbf{P}_\alpha - \mathbf{P}_\beta\right)\\ \mathbf{P}_\beta - \mathbf{P}_\alpha\end{bmatrix}\boldsymbol{\eta}(t),\end{aligned} \tag{20}$$

where the inner state $\mathbf{z}(t) := [\mathbf{z}_1^T(t)\ \ \mathbf{z}_2^T(t)]^T$ is arbitrarily initialized and updated according to

$$\begin{aligned}\mathbf{z}(t+1) = {}& \boldsymbol{\Phi}(t+1,\mu)\mathbf{z}(t) + \begin{bmatrix}\boldsymbol{\epsilon}(t+1)\\ \mathbf{0}_{Jp}\end{bmatrix}\\ &- \begin{bmatrix}\mathbf{1}_J \otimes (\mathbf{s}_0(t+1) - \mathbf{s}_0(t))\\ \mathbf{0}_{Jp}\end{bmatrix}\\ &+ \boldsymbol{\Phi}(t+1,\mu)\begin{bmatrix}\mu\mathbf{I}_{Jp}\\ \mathbf{0}_{Jp\times Jp}\end{bmatrix}\overline{\boldsymbol{\eta}}(t-1)\\ &+ \boldsymbol{\Phi}(t+1,\mu)\begin{bmatrix}\mu\left(3\mathbf{P}_\alpha - \mathbf{P}_\beta\right)\\ \mathbf{C}\end{bmatrix}\boldsymbol{\eta}(t-1)\end{aligned} \tag{21}$$

and the $2Jp \times 2Jp$ transition matrix $\boldsymbol{\Phi}(t,\mu)$ consists of the blocks $[\boldsymbol{\Phi}(t,\mu)]_{11} = \mathbf{I}_{Jp} - 2\mu(\mathbf{R}_h(t) + \mathbf{L}_c)$, $[\boldsymbol{\Phi}(t,\mu)]_{12} = -\mu\mathbf{L}_c$ and $[\boldsymbol{\Phi}(t,\mu)]_{21} = [\boldsymbol{\Phi}(t,\mu)]_{22} = \mathbf{L}_c\mathbf{L}_c^\dagger$. The matrix $\mathbf{C}$ is chosen such that $\mathbf{L}_c\mathbf{C} = \mathbf{P}_\alpha - \mathbf{P}_\beta$, where the structure of the time-invariant matrices $\mathbf{P}_\alpha$ and $\mathbf{P}_\beta$ is given in Appendix B.

The desired state $\mathbf{y}(t)$ is obtained as a rank-deficient linear transformation of the inner state $\mathbf{z}(t)$, plus a stochastic offset due to the effects of communication noise. A linear, time-varying, first-order difference equation describes the dynamics of $\mathbf{z}(t)$, and hence of $\mathbf{y}(t)$, via the algebraic transformation in (20). The time-varying nature of (20)-(21) is inherited from $\boldsymbol{\Phi}(t,\mu)$ that depends on the regression vectors within $\mathbf{R}_h(t)$. Four stochastic inputs are clearly discernible from (21): (i) communication noise $\boldsymbol{\eta}(t-1)$ affecting the transmission of local estimates; (ii) communication noise $\overline{\boldsymbol{\eta}}(t-1)$ contaminating the Lagrange multipliers; (iii) observation noise within $\boldsymbol{\epsilon}(t+1)$; and (iv) a forcing term due to the true "parameter speed" $\mathbf{s}_0(t+1) - \mathbf{s}_0(t)$.

```
Arbitrarily initialize $\{\mathbf{s}_j(0)\}_{j=1}^J$ and set $\{\mathbf{p}_j(-1) = \mathbf{0}_p\}_{j\in\mathcal{J}}$.
For $t = 0, 1, \ldots$ do
    All $j \in \mathcal{J}$: transmit $\mathbf{s}_j(t)$ to neighbors in $\mathcal{N}_j$.
    All $j \in \mathcal{J}$: update $\mathbf{p}_j(t)$ using (16).
    All $j \in \mathcal{J}$: update $\mathbf{s}_j(t+1)$ using (17).
end for
```

Algorithm 2: D-LMS with Ideal Links.

4.2. Performance Metrics. When it comes to performance evaluation of adaptive algorithms, it is customary to consider as figures of merit the so-called MSE, excess mean-square error (EMSE) and mean-square deviation (MSD) [19, 20]. In the present setup for distributed adaptive estimation, it is pertinent to address both global (network-wide) and local (per-sensor) performance [6]. After recalling the definitions of the local a priori error $e_j(t) := x_j(t) - \mathbf{h}_j^T(t)\mathbf{s}_j(t-1)$ and local estimation error $\mathbf{y}_{1,j}(t) := \mathbf{s}_j(t) - \mathbf{s}_0(t)$, the per-sensor performance metrics are defined as

$$\begin{aligned}\text{MSE}_j(t) &:= E\left[e_j^2(t)\right],\\ \text{EMSE}_j(t) &:= E\left[\left(\mathbf{h}_j^T(t)\mathbf{y}_{1,j}(t-1)\right)^2\right],\\ \text{MSD}_j(t) &:= E\left[\left\|\mathbf{y}_{1,j}(t)\right\|^2\right],\end{aligned} \tag{22}$$

whereas their global counterparts are defined as the respective averages across sensors, for example, $\text{MSE}(t) := J^{-1}\sum_{j=1}^J E[e_j(t)^2]$ and so on. Assume for all $j \in \mathcal{J}$ that

(a2) Vectors $\{\mathbf{h}_j(t)\}$ *are spatio-temporally white with covariance matrix $\mathbf{R}_{h_j} \succ \mathbf{0}_{p\times p}$; and*

(a3) Vectors $\{\mathbf{h}_j(t)\}$, $\{\epsilon_j(t)\}$, $\{\boldsymbol{\eta}_j^{j'}(t)\}_{j'\in\mathcal{N}_j}$ *and* $\{\overline{\boldsymbol{\eta}}_j^{j'}(t)\}_{j'\in\mathcal{N}_j}$ *are independent.*

Assumptions (a1)–(a3) comprise the widely adopted *independence setting*, which is instrumental in rendering the subsequent performance analysis tractable; see for example, [19, page 110], [20, page 448]. Clearly, (a2) can be violated in, for example, FIR filtering of signals (regressors) with a shift structure as in the distributed power spectrum estimation problem described in [16, Remark 1]. Nonetheless, for small step-sizes the upshot of the analysis extends to correlated data as will be demonstrated via computer simulations in Section 6.

Next, we show that it suffices to evaluate the state covariance matrix $\mathbf{R}_y(t) := E[\mathbf{y}(t)\mathbf{y}^T(t)]$ in order to assess the aforementioned performance metrics. To this end, note that by virtue of (a1) it is possible to write $e_j(t) = -\mathbf{h}_j^T(t)\mathbf{y}_{1,j}(t-1) + \epsilon_j(t)$. Because $\mathbf{y}_{1,j}(t-1)$ is independent of the zero-mean $\{\mathbf{h}_j(t), \epsilon_j(t)\}$ under (a1)–(a3), from the previous relationship between the a priori and estimation errors one finds that $\text{MSE}_j(t) = \text{EMSE}_j(t) + \sigma_{\epsilon_j}^2$. Hence, it suffices to focus on the evaluation of $\text{EMSE}_j(t)$, through which $\text{MSE}_j(t)$ can also be determined. If $\mathbf{R}_{y_{1,j}}(t) := E[\mathbf{y}_{1,j}(t)\mathbf{y}_{1,j}^T(t)]$ denotes the jth local error covariance matrix, then $\text{MSD}_j(t) = \text{tr}(\mathbf{R}_{y_{1,j}}(t))$; and under (a1)–(a3), a simple manipulation yields

$$\begin{aligned}\text{EMSE}_j(t) &= E\left[\text{tr}\left(\left(\mathbf{h}_j^T(t)\mathbf{y}_{1,j}(t-1)\right)^2\right)\right]\\ &= \text{tr}\left(E\left[\mathbf{h}_j(t)\mathbf{h}_j^T(t)\mathbf{y}_{1,j}(t-1)\mathbf{y}_{1,j}^T(t-1)\right]\right)\\ &= \text{tr}\left(E\left[\mathbf{h}_j(t)\mathbf{h}_j^T(t)\right]E\left[\mathbf{y}_{1,j}(t-1)\mathbf{y}_{1,j}^T(t-1)\right]\right)\\ &= \text{tr}\left(\mathbf{R}_{h_j}\mathbf{R}_{y_{1,j}}(t-1)\right).\end{aligned} \tag{23}$$

Observe that the global error covariance matrix corresponds to the $Jp \times Jp$ upper left submatrix $[\mathbf{R}_y(t)]_{11}$ of $\mathbf{R}_y(t)$. Further, its jth $p \times p$ diagonal submatrix ($j = 1, \ldots, J$) denoted by $[\mathbf{R}_y(t)]_{11,j}$ is $\mathbf{R}_{y_{1,j}}(t)$. It follows that with $\mathbf{R}_h := E[\mathbf{R}_h(t)] = b\text{diag}(\mathbf{R}_{h_1}, \ldots, \mathbf{R}_{h_J})$, the global performance metrics are given by $\text{MSD}(t) = J^{-1}\text{tr}([\mathbf{R}_y(t)]_{11})$ and $\text{EMSE}(t) = J^{-1}\text{tr}(\mathbf{R}_h[\mathbf{R}_y(t-1)]_{11})$. In a nutshell, deriving a closed-form expression for $\mathbf{R}_y(t)$ enables the evaluation of all performance metrics of interest.

5. Performance Analysis of D-LMS Tracking

In this section, a performance analysis is conducted for the D-LMS tracking algorithm. For that purpose, we first introduce a commonly adopted model for the time-varying parameter $\mathbf{s}_0(t)$, and resort to a simplifying Gaussian assumption on the regression vectors to complement (a1)–(a3). Main results include deriving an exact closed-form recursion for the global error covariance matrix $[\mathbf{R}_y(t)]_{11}$, and establishing the existence of step-sizes ensuring stability of D-LMS both in the mean and MSE-sense. Expressions for the s.s. local and global figures of merit are also provided. To conclude, the s.s. EMSE is viewed as a function of the step-size, and is compared for time-invariant and time-varying parameters. While in the former case the trend is monotonically increasing, when tracking slowly time-varying processes there exists a non-vanishing optimal step-size minimizing the limiting error.

To characterize fluctuations of the time-varying parameter $\mathbf{s}_0(t)$, consider for all $j \in \mathcal{J}$:

(a4) *First-order autoregressive [AR(1)] model, that is,*

$$\begin{aligned}\mathbf{s}_0(t) &= \mathbf{s}_0 + \check{\mathbf{s}}(t),\\ \check{\mathbf{s}}(t) &= \boldsymbol{\Theta}\check{\mathbf{s}}(t-1) + \boldsymbol{\zeta}(t)\end{aligned} \tag{24}$$

where $\boldsymbol{\Theta} \in \mathbb{R}^{p\times p}$ has eigenvalues with modulus within $[0, 1)$, the driving noise $\{\boldsymbol{\zeta}(t)\}$ is zero-mean, white with covariance matrix $\mathbf{R}_\zeta \succ \mathbf{0}_{p\times p}$; and $E[\check{\mathbf{s}}(-1)] = \mathbf{0}_p$.

Under (a4), the perturbation due to the parameter velocity in (21) becomes

$$\begin{aligned}&-\begin{bmatrix}\mathbf{1}_J \otimes (\mathbf{s}_0(t+1) - \mathbf{s}_0(t))\\ \mathbf{0}_{Jp}\end{bmatrix}\\ &\quad= \begin{bmatrix}\mathbf{1}_J \otimes \left(\mathbf{I}_p - \boldsymbol{\Theta}\right)\check{\mathbf{s}}(t)\\ \mathbf{0}_{Jp}\end{bmatrix} - \begin{bmatrix}\mathbf{1}_J \otimes \boldsymbol{\zeta}(t+1)\\ \mathbf{0}_{Jp}\end{bmatrix},\end{aligned} \tag{25}$$

and its expectation vanishes for all $t \geq 0$. Further, (a3) is augmented and replaced by

(a5) Vectors $\{\mathbf{h}_j(t)\}$, $\{\epsilon_j(t)\}$, $\{\boldsymbol{\zeta}(t)\}$, $\check{\mathbf{s}}(-1)$, $\{\boldsymbol{\eta}_j^{j'}(t)\}_{j'\in\mathcal{N}_j}$ *and $\{\bar{\boldsymbol{\eta}}_j^{j'}(t)\}_{j'\in\mathcal{N}_j}$ are independent.*

The model in (a4) provides a simple description of a time-varying parameter, and has been widely adopted to evaluate the performance of (centralized) adaptive filters [19, page 121], [20, page 360]. The true parameter $\mathbf{s}_0(t)$ has been split into a "DC level" $\mathbf{s}_0$ which is superimposed to the "AC component" $\check{\mathbf{s}}(t)$, with fluctuations adhering to a stable vector AR(1) process. Two other models of interest are obtained by simple modifications to (a4):

(a4.1) *Time-invariant parameter, that is, $\mathbf{s}_0(t) = \mathbf{s}_0$ is subsumed by (a4), by selecting $\boldsymbol{\Theta} = \mathbf{R}_\zeta = \mathbf{0}_{p\times p}$.*

(a4.2) *Random-walk model, that is, $\mathbf{s}_0(t) = \mathbf{s}_0(t-1) + \boldsymbol{\zeta}(t)$, where $\{\boldsymbol{\zeta}(t)\}$ is zero-mean, white, with covariance matrix $\mathbf{R}_\zeta \succ \mathbf{0}$; and $E[\mathbf{s}_0(-1)] = \mathbf{0}_p$. Such a model is obtained by letting $\mathbf{s}_0 = \mathbf{0}_p$ and $\boldsymbol{\Theta} = \mathbf{I}_p$ in (a4).*

A random-walk is the simplest stochastic model to describe variations of $\mathbf{s}_0(t)$, and has been also considered for performance analysis of trackers; see, for example, [19, page 121] and [20, page 359]. It could be arguably thought as not meaningful due to its increasing variance, thus violating the sensor's limited dynamical range which requires $E[x_j^2(t)] < \infty$, for all t. To circumvent this problem, the forthcoming analysis generalizes [26] by considering the (asymptotically) stationary case in (a4). When $\boldsymbol{\Theta}$ is a stable matrix, that is, $\lambda_{\max}(\boldsymbol{\Theta}) < 1$, the s.s. covariance matrix of $\mathbf{s}_0(t)$ has finite entries, and obeys the Lyapunov equation $\mathbf{R}_{\mathbf{s}_0}(\infty) = \boldsymbol{\Theta}\mathbf{R}_{\mathbf{s}_0}(\infty)\boldsymbol{\Theta}^T + \mathbf{R}_\zeta$. In any case, the model is simple but well justified as the resulting analysis sheds sufficient light on the key aspects of D-LMS when it comes to tracking.

5.1. Mean Stability. From Lemma 5 it is straightforward to establish that local estimates obtained via D-LMS are asymptotically unbiased, implying that consensus in the mean-sense is achieved on $\mathbf{s}_0$.

Proposition 6. *Under (a1)-(a2) and (a4)-(a5), the D-LMS algorithm achieves consensus in the mean, that is,*

$$\lim_{t\to\infty} E\left[\mathbf{y}_{1,j}(t)\right] = \mathbf{0}_p, \quad \forall j \in \mathcal{J} \tag{26}$$

provided the step-size is chosen such that $\mu \in (0, \mu_u)$ with

$$\mu_u := \min\left(\frac{2}{\lambda_{\max}(\mathbf{R}_h + \mathbf{L}_c)}, \frac{2}{\lambda_{\max}(2\mathbf{R}_h + (3/2)\mathbf{L}_c)}\right). \tag{27}$$

Proof. Based on the independence setting assumptions (a1)-(a2), (a5) and since the data is zero-mean, one obtains after taking expectations on (20) and (21) that $E[\mathbf{y}(t)] = b\text{diag}(\mathbf{I}_{Jp}, \mathbf{L}_c)E[\mathbf{z}(t)]$ and $E[\mathbf{z}(t)] = \mathbf{\Phi}(\mu)E[\mathbf{z}(t-1)]$, where $\mathbf{\Phi}(\mu) := E[\mathbf{\Phi}(t,\mu)]$. Assumption (a4) was also invoked to render the expectation of (25) null. The following lemma specifies the step-size values under which $\mathbf{\Phi}(\mu)$ is a stable matrix.

Lemma 7. *If $\mu > 0$ is chosen smaller than* (27)*, then $\mathbf{\Phi}(\mu)$ is a stable matrix, that is, $\lambda_{\max}(\mathbf{\Phi}(\mu)) < 1$.*

Proof. Following steps similar to those in [17, Appendix H], it is possible to express the eigenvalues of $\mathbf{\Phi}(\mu)$ as the roots of a second-order polynomial to determine bounds on μ that ensure $\lambda_{\max}(\mathbf{\Phi}(\mu)) < 1$. Further, for sufficiently small μ the eigenvalues with largest modulus correspond to a complex conjugate pair, while the spectral radius scales as $\lambda_{\max}(\mathbf{\Phi}(\mu)) \sim 1 - \mu\kappa$, where $\kappa > 0$ is a finite constant. □

From Lemma 7 and the theory of linear time-invariant dynamical systems, $E[\mathbf{z}(t)]$ is exponentially convergent to zero for $\mu \in (0, \mu_u)$. Noting that $E[\mathbf{y}_1(t)] = E[\mathbf{z}_1(t)]$ (cf. (20)), the result follows. □

Interestingly, μ_u resembles the first-order (mean) stability bound for the centralized LMS algorithm, namely $2/\lambda_{\max}(\mathbf{R}_h)$ [19, page 111]. The main difference here is that this bound is also affected by the topology of the WSN, via the graph Laplacian matrix within $\mathbf{L}_c$.

5.2. MSE Stability and Performance Evaluation. Turning to MSE stability and performance analysis, observe from the upper $Jp \times 1$ block of $\mathbf{y}(t+1)$ in (20) that $\mathbf{y}_1(t+1) = \mathbf{z}_1(t+1) + \mu[\overline{\boldsymbol{\eta}}(t) + (3\mathbf{P}_\alpha - \mathbf{P}_\beta)\boldsymbol{\eta}(t)]$. Under (a2) and (a5), $\mathbf{z}_1(t+1)$ is independent of the zero-mean $\{\overline{\boldsymbol{\eta}}(t), \boldsymbol{\eta}(t)\}$; hence,

$$\begin{aligned}\left[\mathbf{R}_y(t)\right]_{11} &= [\mathbf{R}_z(t)]_{11} + \mu^2 \\ &\quad\times\left[\mathbf{R}_{\overline{\eta}} + \left(3\mathbf{P}_\alpha - \mathbf{P}_\beta\right)\mathbf{R}_{\eta}\left(3\mathbf{P}_\alpha - \mathbf{P}_\beta\right)^T\right]\end{aligned} \tag{28}$$

based on which we obtain $\mathbf{R}_z(t) := E[\mathbf{z}(t)\mathbf{z}^T(t)]$. From the coupling between $\mathbf{z}(t)$ and $\check{\mathbf{s}}(t)$ entering through (25), it is convenient to consider the augmented state $\check{\mathbf{z}}(t) := [\mathbf{z}^T(t) \quad \mathbf{1}_J^T \otimes \check{\mathbf{s}}^T(t)]^T$ in order to perform covariance calculations [19, page 124]. From (21), (a4) and (25), one finds that $\check{\mathbf{z}}(t)$ can be recursively updated as

$$\begin{aligned}\check{\mathbf{z}}(t+1) &= \begin{bmatrix} \mathbf{\Phi}(t+1,\mu) & \left[\left(\mathbf{I}_J \otimes \left(\mathbf{I}_p - \mathbf{\Theta}\right)\right)^T \quad \mathbf{0}_{Jp\times Jp}\right]^T \\ \mathbf{0}_{Jp\times 2Jp} & \mathbf{I}_J \otimes \mathbf{\Theta} \end{bmatrix}\check{\mathbf{z}}(t) \\ &\quad + \begin{bmatrix} \boldsymbol{\epsilon}(t+1) \\ \mathbf{0}_{2Jp} \end{bmatrix} + \begin{bmatrix} -\mathbf{1}_J \otimes \boldsymbol{\zeta}(t+1) \\ \mathbf{0}_{Jp} \\ \mathbf{1}_J \otimes \boldsymbol{\zeta}(t+1) \end{bmatrix} \\ &\quad + \begin{bmatrix} \mathbf{\Phi}(t+1,\mu) \\ \mathbf{0}_{Jp\times 2Jp} \end{bmatrix}\left(\overline{\boldsymbol{\eta}}_\mu(t-1) + \boldsymbol{\eta}_\mu(t-1)\right)\end{aligned} \tag{29}$$

$$:= \mathbf{\Omega}(t+1,\mu)\check{\mathbf{z}}(t) + \boldsymbol{\nu}(t+1), \tag{30}$$

where for notational convenience $\mathbf{\Omega}(t+1,\mu)$ denotes the new transition matrix and $\boldsymbol{\nu}(t+1)$ encapsulates all three forcing terms. Note that in writing (29) we have introduced

$$\overline{\boldsymbol{\eta}}_\mu(t) := \begin{bmatrix} \mu\mathbf{I}_{Jp} \\ \mathbf{0}_{Jp\times Jp} \end{bmatrix}\overline{\boldsymbol{\eta}}(t), \qquad \boldsymbol{\eta}_\mu(t) := \begin{bmatrix} \mu\left(3\mathbf{P}_\alpha - \mathbf{P}_\beta\right) \\ \mathbf{C} \end{bmatrix}\boldsymbol{\eta}(t) \tag{31}$$

while the structure of the respective covariance matrices $\mathbf{R}_{\overline{\eta}_\mu} := E[\overline{\boldsymbol{\eta}}_\mu(t)\overline{\boldsymbol{\eta}}_\mu^T(t)]$ and $\mathbf{R}_{\eta_\mu} := E[\boldsymbol{\eta}_\mu(t)\boldsymbol{\eta}_\mu^T(t)]$ is given in Appendix C.

By definition of the augmented state $\check{\mathbf{z}}(t)$, the desired covariance matrix $\mathbf{R}_z(t)$ clearly corresponds to the $2Jp \times 2Jp$ upper left submatrix of $\mathbf{R}_{\check{z}}(t) := E[\check{\mathbf{z}}(t)\check{\mathbf{z}}^T(t)]$. Towards obtaining a closed-form expression for $\mathbf{R}_{\check{z}}(t)$, observe that for all $j \in \mathcal{J}$ there exist $p \times p$ unitary matrices $\mathbf{U}_j$ that are arranged in $\mathbf{U} := b\text{diag}(\mathbf{U}_1, \ldots, \mathbf{U}_J)$ such that $\mathbf{U}_j\mathbf{R}_{h_j}\mathbf{U}_j^T = \mathbf{\Lambda}_j = \text{diag}(\lambda_1^j, \ldots, \lambda_p^j)$, and also $\mathbf{U}\mathbf{R}_h\mathbf{U}^T = \mathbf{\Lambda} = b\text{diag}(\mathbf{\Lambda}_1, \ldots, \mathbf{\Lambda}_J)$. For the subsequent arguments, it will prove useful to introduce the (invertible) change of variables $\tilde{\mathbf{z}}(t) := \tilde{\mathbf{U}}\check{\mathbf{z}}(t)$ with $\tilde{\mathbf{U}} := b\text{diag}(\mathbf{U}, \mathbf{I}_{2Jp})$. To proceed, specialize (a2) by assuming that:

(a6) Vectors $\{\mathbf{h}_j(t)\}$ *are spatio-temporally white Gaussian with covariance matrix* $\mathbf{R}_{h_j} \succ \mathbf{0}_p$.

The Gaussianity assumption is instrumental in obtaining closed-form expressions for the regressors' fourth-order moments, which arise in the evaluation of $\mathbf{R}_{\tilde{z}}(t+1)$ as shown next.

Proposition 8. *Under (a4)–(a6) and for $t \geq 0$, the covariance matrix of $\tilde{\mathbf{z}}(t)$ obeys the first-order matrix recursion given by*

$$\begin{aligned}\mathbf{R}_{\tilde{z}}(t+1) = {} & \mathcal{M}\left(\tilde{\mathbf{\Omega}}(\mu), \mathbf{R}_{\tilde{z}}(t)\right) \\ & + \mathcal{M}\Big(\tilde{\mathbf{\Phi}}(\mu), b\text{diag}\left(\mathbf{U}, \mathbf{I}_{Jp}\right) \\ & \times \left(\mathbf{R}_{\bar{\eta}_\mu} + \mathbf{R}_{\eta_\mu}\right) b\text{diag}\left(\mathbf{U}^T, \mathbf{I}_{Jp}\right)\Big) \\ & + 4\mu^2 b\text{diag}\left(\sigma_{\epsilon_1}^2 \mathbf{\Lambda}_1, \ldots, \sigma_{\epsilon_J}^2 \mathbf{\Lambda}_J, \mathbf{0}_{2Jp\times 2Jp}\right) \\ & + \tilde{\mathbf{U}}\left(\left([-1\ 0\ 1]^T[-1\ 0\ 1]\right) \otimes \left(\mathbf{1}_{J\times J} \otimes \mathbf{R}_\zeta\right)\right)\tilde{\mathbf{U}}^T\end{aligned} \tag{32}$$

with $\tilde{\mathbf{\Omega}}(\mu) := \tilde{\mathbf{U}}E[\mathbf{\Omega}(t,\mu)]\tilde{\mathbf{U}}^T$ *and* $\tilde{\mathbf{\Phi}}(\mu) := \tilde{\mathbf{U}}E[[\mathbf{\Phi}(t+1,\mu)^T\ \mathbf{0}_{2Jp\times Jp}]^T]b\text{diag}(\mathbf{U}^T, \mathbf{I}_{Jp}))$*, while*

$$\begin{aligned}&\mathcal{M}(\mathbf{S}, \mathbf{T}) \\ &:= \mathbf{STS}^T + 4\mu^2 b\text{diag}\left(\left(\mathbf{I}_J \otimes \mathbf{1}_{p\times p}\right) \circ \mathbf{\Lambda}[\mathbf{T}]_{11}\mathbf{\Lambda}, \mathbf{0}_{2Jp\times 2Jp}\right) \\ &+ 4\mu^2 b\text{diag}\left(tr\left(\mathbf{\Lambda}_1[\mathbf{T}]_{11,1}\right)\mathbf{\Lambda}_1, \ldots, tr\left(\mathbf{\Lambda}_J[\mathbf{T}]_{11,J}\right)\mathbf{\Lambda}_J, \mathbf{0}_{2Jp\times 2Jp}\right).\end{aligned} \tag{33}$$

Proof. In the transformed space, (30) becomes $\tilde{\mathbf{z}}(t) = \tilde{\mathbf{\Omega}}(t,\mu)\tilde{\mathbf{z}}(t-1) + \tilde{\boldsymbol{\nu}}(t)$, where $\tilde{\mathbf{\Omega}}(t,\mu) := \tilde{\mathbf{U}}\mathbf{\Omega}(t,\mu)\tilde{\mathbf{U}}^T$ and $\tilde{\boldsymbol{\nu}}(t) := \tilde{\mathbf{U}}\boldsymbol{\nu}(t)$. Using (a4)–(a6), it follows that $E[\tilde{\mathbf{\Omega}}(t,\mu)\tilde{\mathbf{z}}(t-1)\tilde{\boldsymbol{\nu}}^T(t)] = \mathbf{0}_{3Jp\times 3Jp}$. Therefore, $\mathbf{R}_{\tilde{z}}(t) = E[\tilde{\mathbf{\Omega}}(t,\mu)\mathbf{R}_{\tilde{z}}(t-1)\tilde{\mathbf{\Omega}}^T(t,\mu)] + E[\tilde{\boldsymbol{\nu}}(t)\tilde{\boldsymbol{\nu}}^T(t)]$ and we start by showing that the first expectation is $\mathcal{M}(\tilde{\mathbf{\Omega}}(\mu), \mathbf{R}_{\tilde{z}}(t-1))$. Split $\tilde{\mathbf{\Omega}}(t,\mu) = \bar{\mathbf{\Omega}}(\mu) - 2\mu b\text{diag}(\mathbf{R}_{\tilde{h}}(t), \mathbf{0}_{2Jp\times 2Jp})$ into its deterministic and random components, and drop for simplicity the $t-1$ argument in $\mathbf{R}_{\tilde{z}}(t-1)$ to obtain

$$\begin{aligned}&E\left[\tilde{\mathbf{\Omega}}(t,\mu)\mathbf{R}_{\tilde{z}}\tilde{\mathbf{\Omega}}^T(t,\mu)\right] \\ &= \bar{\mathbf{\Omega}}(\mu)\mathbf{R}_{\tilde{z}}\bar{\mathbf{\Omega}}^T(\mu) \\ &\quad + 4\mu^2 E\left[b\text{diag}\left(\mathbf{R}_{\tilde{h}}(t), \mathbf{0}_{2Jp\times 2Jp}\right)\mathbf{R}_{\tilde{z}}b\text{diag}\left(\mathbf{R}_{\tilde{h}}(t), \mathbf{0}_{2Jp\times 2Jp}\right)\right] \\ &\quad - 2\mu\Big[\bar{\mathbf{\Omega}}(\mu)\mathbf{R}_{\tilde{z}}b\text{diag}\left(\mathbf{\Lambda}, \mathbf{0}_{2Jp\times 2Jp}\right) \\ &\qquad + \left(\bar{\mathbf{\Omega}}(\mu)\mathbf{R}_{\tilde{z}}b\text{diag}\left(\mathbf{\Lambda}, \mathbf{0}_{2Jp\times 2Jp}\right)\right)^T\Big].\end{aligned} \tag{34}$$

The second summand in the rhs of (34) has the structure $4\mu^2 b\text{diag}(\mathbf{A}, \mathbf{0}_{2Jp\times 2Jp})$, where $\mathbf{A} \in \mathbb{R}^{Jp\times Jp}$ can be partitioned into $p \times p$ blocks

$$[\mathbf{A}]_{i,j} = \begin{cases} E\left[\tilde{\mathbf{h}}_j(t)\tilde{\mathbf{h}}_j^T(t)[\mathbf{R}_{\tilde{z}}]_{11,j}\tilde{\mathbf{h}}_j(t)\tilde{\mathbf{h}}_j^T(t)\right] & \\ \quad = 2\mathbf{\Lambda}_j[\mathbf{R}_{\tilde{z}}]_{11,j}\mathbf{\Lambda}_j + \text{tr}\left(\mathbf{\Lambda}_j[\mathbf{R}_{\tilde{z}}]_{11,j}\right)\mathbf{\Lambda}_j, & i = j \\ E\left[\tilde{\mathbf{h}}_i(t)\tilde{\mathbf{h}}_i^T(t)[\mathbf{R}_{\tilde{z}}]_{11,i,j}\tilde{\mathbf{h}}_j(t)\tilde{\mathbf{h}}_j^T(t)\right] & \\ \quad = \mathbf{R}_{h_i}[\mathbf{R}_{\tilde{z}}]_{11,i,j}\mathbf{R}_{h_j}, & i \neq j \end{cases} \tag{35}$$

for $i, j = 1, \ldots, J$. To evaluate the regressor's fourth-order moments in the diagonal blocks of $\mathbf{A}$, we have relied on the Gaussianity of $\tilde{\mathbf{h}}_j(t)$ for all $j \in \mathcal{J}$, which follows from (a6). The expectations in the nondiagonal blocks follow immediately as regressors are also assumed spatially uncorrelated. Substituting in (34) and regrouping terms one obtains

$$\begin{aligned}&\tilde{\mathbf{\Omega}}(\mu)\mathbf{R}_{\tilde{z}}\tilde{\mathbf{\Omega}}^T(\mu) \\ &\quad = \bar{\mathbf{\Omega}}(\mu)\mathbf{R}_{\tilde{z}}\bar{\mathbf{\Omega}}^T(\mu) \\ &\qquad + 4\mu^2 b\text{diag}\left(\mathbf{A} - b\text{diag}\left([\mathbf{A}]_{1,1}, \ldots, [\mathbf{A}]_{J,J}\right), \mathbf{0}_{2Jp\times 2Jp}\right) \\ &\qquad + 4\mu^2 b\text{diag}\left(\mathbf{\Lambda}_1[\mathbf{R}_{\tilde{z}}]_{11,1}\mathbf{\Lambda}_1, \ldots, \mathbf{\Lambda}_J[\mathbf{R}_{\tilde{z}}]_{11,J}\mathbf{\Lambda}_J, \mathbf{0}_{2Jp\times Jp}\right) \\ &\qquad - 2\mu\Big[\bar{\mathbf{\Omega}}(\mu)\mathbf{R}_{\tilde{z}}b\text{diag}\left(\mathbf{\Lambda}, \mathbf{0}_{2Jp\times 2Jp}\right) \\ &\qquad\qquad + \left(\bar{\mathbf{\Omega}}(\mu)\mathbf{R}_{\tilde{z}}b\text{diag}\left(\mathbf{\Lambda}, \mathbf{0}_{2Jp\times 2Jp}\right)\right)^T\Big]\end{aligned} \tag{36}$$

which finally yields

$$\begin{aligned}&E\left[\tilde{\mathbf{\Omega}}(t,\mu)\mathbf{R}_{\tilde{z}}\tilde{\mathbf{\Omega}}^T(t,\mu)\right] \\ &= \tilde{\mathbf{\Omega}}(\mu)\mathbf{R}_{\tilde{z}}\tilde{\mathbf{\Omega}}^T(\mu) \\ &\quad + 4\mu^2 b\text{diag}\left(\mathbf{\Lambda}_1[\mathbf{R}_{\tilde{z}}]_{11,1}\mathbf{\Lambda}_1, \ldots, \mathbf{\Lambda}_J[\mathbf{R}_{\tilde{z}}]_{11,J}\mathbf{\Lambda}_J, \mathbf{0}_{2Jp\times Jp}\right) \\ &\quad + 4\mu^2 b\text{diag}\left(\text{tr}\left(\mathbf{\Lambda}_1[\mathbf{R}_{\tilde{z}}]_{11,1}\right)\mathbf{\Lambda}_1, \ldots, \text{tr}\left(\mathbf{\Lambda}_J[\mathbf{R}_{\tilde{z}}]_{11,J}\right)\mathbf{\Lambda}_J, \mathbf{0}_{2Jp\times Jp}\right).\end{aligned} \tag{37}$$

Simple manipulations on the second term in the rhs of (37) lead to the desired result (cf. (33)). Back to the remaining covariance $\mathbf{R}_{\tilde{\nu}} := E[\tilde{\boldsymbol{\nu}}(t)\tilde{\boldsymbol{\nu}}^T(t)]$, because the three noise terms within $\tilde{\boldsymbol{\nu}}(t)$ (cf. (29)) are pairwise independent and zero-mean, we have that

$$\begin{aligned}\mathbf{R}_{\tilde{\nu}} = {} & E\Big[\tilde{\mathbf{U}}\left[\mathbf{\Phi}(t+1,\mu)^T\ \mathbf{0}_{2Jp\times Jp}\right]^T\left(\mathbf{R}_{\bar{\eta}_\mu} + \mathbf{R}_{\eta_\mu}\right) \\ & \times\left(\tilde{\mathbf{U}}\left[\mathbf{\Phi}(t+1,\mu)^T\ \mathbf{0}_{2Jp\times Jp}\right]^T\right)^T\Big] \\ & + 4\mu^2 b\text{diag}\left(\sigma_{\epsilon_1}^2\mathbf{\Lambda}_1, \ldots, \sigma_{\epsilon_J}^2\mathbf{\Lambda}_J, \mathbf{0}_{2Jp\times 2Jp}\right) \\ & + \tilde{\mathbf{U}}\left(\left([-1\ 0\ 1]^T[-1\ 0\ 1]\right) \otimes \left(\mathbf{1}_{J\times J} \otimes \mathbf{R}_\zeta\right)\right)\tilde{\mathbf{U}}^T,\end{aligned} \tag{38}$$

where the last two terms follow after using (a4)–(a6), and correspond to the covariance matrices of the second and third vectors in the rhs of (29). The structure of $\mathbf{R}_{\bar{\eta}_\mu}$ and $\mathbf{R}_{\eta_\mu}$ is provided in Appendix C. The first expectation in (38) can be treated similarly as $E[\tilde{\mathbf{\Omega}}(t,\mu)\mathbf{R}_{\tilde{z}}(t-1)\tilde{\mathbf{\Omega}}^T(t,\mu)]$ to yield the second summand in the rhs of (32). □

The covariance recursion in Proposition 8 (indirectly) characterizes the exact tracking MSE evolution of the D-LMS algorithm, under the white Gaussian setting assumptions and the vector AR(1) model for $\mathbf{s}_0(t)$. With the appropriate simplifications indicated in (a4.1) and (a4.2), (32) enables performance evaluation when the parameter vector of interest, $\mathbf{s}_0(t)$, is either time-invariant or adheres to a random-walk model. For example, under (a4.1) the last matrix in the rhs of (32) vanishes because $\mathbf{R}_{\zeta} = \mathbf{0}_{p\times p}$ while the inner structure of $\widetilde{\mathbf{\Omega}}(\mu)$ should be adapted to $\mathbf{\Theta} = \mathbf{0}_{p\times p}$.

Starting from Proposition 8, the recipe towards obtaining the performance metrics described in Section 4.2 is the following. Given (32) and upon inverting the change of variables to yield $\mathbf{R}_{\check{z}}(t) = \widetilde{\mathbf{U}}^T\mathbf{R}_{\tilde{z}}(t)\widetilde{\mathbf{U}}$, one can readily extract $[\mathbf{R}_z(t)]_{11}$ as the upper-left $Jp\times Jp$ submatrix of $\mathbf{R}_{\check{z}}(t)$. Closed-form evaluation of the MSE(t), EMSE(t) and MSD(t) for all $t \geq 0$ and every sensor $j \in \mathcal{J}$ is now possible by using (28) to obtain $[\mathbf{R}_y(t)]_{11}$, and then resorting to the formulae in Section 4.2.

The next step is to reformulate (32) into a first-order vector recursion which is better suited for stability analysis. Specifically, (32) can be vectorized to obtain $\text{vec}[\mathbf{R}_{\tilde{z}}(t+1)] = \text{vec}[\mathcal{M}(\widetilde{\mathbf{\Omega}}(\mu), \mathbf{R}_{\tilde{z}}(t))] + \text{vec}[\mathbf{R}_{\tilde{\boldsymbol{\nu}}}]$. As asserted in the following lemma, further simplification is possible by relying on properties of the matrix vectorization operator [27]. It is shown in Appendix D the following.

Lemma 9. *Under (a4)–(a6) and for $t \geq 0$, the vectorized covariance matrix of $\tilde{\mathbf{z}}(t)$ obeys the first-order vector recursion given by*

$$\text{vec}[\mathbf{R}_{\tilde{z}}(t+1)] = \widetilde{\mathbf{\Psi}}(\mu)\,\text{vec}[\mathbf{R}_{\tilde{z}}(t)] + \text{vec}[\mathbf{R}_{\tilde{\boldsymbol{\nu}}}]. \tag{39}$$

The $(3Jp)^2 \times (3Jp)^2$ transition matrix $\widetilde{\mathbf{\Psi}}(\mu)$ is

$$\begin{aligned}\widetilde{\mathbf{\Psi}}(\mu) := {}& \widetilde{\mathbf{\Omega}}(\mu) \otimes \widetilde{\mathbf{\Omega}}(\mu) + 4\mu^2 \\ & \times \Bigg[\Big(b\text{diag}\big(\mathbf{\Lambda}, \mathbf{0}_{2Jp\times 2Jp}\big) \otimes b\text{diag}\big(\mathbf{\Lambda}, \mathbf{0}_{2Jp\times 2Jp}\big)\Big) \\ & \times \text{diag}\Big(\text{vec}\big[\mathbf{I}_{3J} \otimes \mathbf{1}_{p\times p}\big]\Big) + \sum_{j=1}^{J}\mathbf{q}_j\mathbf{q}_j^T \Bigg]\end{aligned} \tag{40}$$

where $\mathbf{q}_j := \text{vec}[\text{diag}(\mathbf{b}_{3J,j}) \otimes \mathbf{\Lambda}_j]$ for all $j \in \mathcal{J}$.

An immediate consequence of Lemma 9 is that the D-LMS algorithm is MSE stable if $\lambda_{\max}(\widetilde{\mathbf{\Psi}}(\mu)) < 1$. Although deriving explicit bounds on μ for stability appears intractable, the following proposition provides an important existence result.

Proposition 10. *Under (a1), (a4)–(a6) the D-LMS algorithm is MSE stable, that is, $\lim_{t\to\infty}[\mathbf{R}_y(t)]_{11}$ has bounded entries, provided that $\mu > 0$ is chosen sufficiently small.*

Proof. The eigenvalues of $\widetilde{\mathbf{\Omega}}(\mu) \otimes \widetilde{\mathbf{\Omega}}(\mu)$ are the pairwise products of those of $\widetilde{\mathbf{\Omega}}(\mu)$. From (40) it is possible to upper-bound $\lambda_{\max}(\widetilde{\mathbf{\Psi}}(\mu)) \leq \lambda_{\max}(\widetilde{\mathbf{\Omega}}(\mu))^2 + \kappa_1\mu^2$, with κ_1 a finite positive constant. Given the block upper-triangular structure of $E[\mathbf{\Omega}(t,\mu)]$ (cf. (29)) which has the same eigenvalues as $\widetilde{\mathbf{\Omega}}(\mu)$, for $\kappa_2 \in (0,\infty)$ one obtains that $\lambda_{\max}(\widetilde{\mathbf{\Omega}}(\mu)) = \max(\lambda_{\max}(\mathbf{\Phi}(\mu)),\lambda_{\max}(\mathbf{\Theta})) \sim \max(1-\mu\kappa_2, \lambda_{\max}(\mathbf{\Theta}))$, where the scaling of $\lambda_{\max}(\mathbf{\Phi}(\mu))$ follows from the proof of Lemma 7. By virtue of (a4), $\lambda_{\max}(\mathbf{\Theta}) \in [0,1)$ and is independent of μ. Hence, $\lambda_{\max}(\widetilde{\mathbf{\Omega}}(\mu)) \sim 1-\mu\kappa_2$ for μ small enough so that $\lambda_{\max}(\widetilde{\mathbf{\Psi}}(\mu)) \leq 1-\mu[2\kappa_2-\mu(\kappa_2^2+\kappa_1)]$, which can be made smaller than one for $\mu > 0$ sufficiently small. This readily implies that $\lim_{t\to\infty}\mathbf{R}_{\tilde{z}}(t)$ has bounded entries, and can be established also for $\lim_{t\to\infty}[\mathbf{R}_y(t)]_{11}$ via the process described after Proposition 8. □

While the proof for Proposition 10 is still valid for a time-invariant parameter vector, the argument clearly breaks down for the random-walk model because $\lambda_{\max}(\mathbf{\Theta}) = \lambda_{\max}(\mathbf{I}_p) = 1$. In this case, $\mathbf{R}_{\mathbf{s}_0}(t)$ grows unbounded; thus, one would expect that the same happens to the inner state $\mathbf{z}(t)$. However, note that the coupling between $\mathbf{z}(t)$ and $\mathbf{s}_0(t)$ arising in (25) disappears under (a4.2). For this reason, it is possible to reproduce all previous results by working just with $\mathbf{z}(t)$ (instead of $\check{\mathbf{z}}(t)$) to finally conclude that Proposition 10 holds true for the random-walk model also [26].

Next, we consider an alternative notion of stochastic stability that can be inferred from Proposition 10. Specifically, it is possible to show that under the white Gaussian setting assumptions, the error norm $\|\mathbf{y}_1(t)\|$ remains most of the time in a finite interval, that is, errors are weakly stochastic bounded (WSB) [18], [19, page 110]. This WSB stability guarantees that for any $\theta > 0$ there exists a $\delta > 0$ such that $\Pr[\|\mathbf{y}_1(t)\| < \delta] = 1-\theta$ uniformly in t. It is a weak notion of stability, providing an alternative for the analysis of adaptive filters when the presence of, for example, time-correlated data, renders variance calculations impossible; see also [16, 18]. Nevertheless, it is an important practical notion as it ensures—on a per-realization basis—that there is no probability mass allowing estimation errors escape to infinity. Similar to Proposition 10, this property holds for the D-LMS algorithm in the presence of communication noise.

Proposition 11. *Under (a1), (a4)–(a6) and if the step-size $\mu > 0$ is chosen sufficiently small, then the D-LMS algorithm yields estimation errors which are WSB; that is,*

$$\lim_{\delta\to\infty}\sup_{t\geq 0}\Pr[||\mathbf{y}_1(t)|| \geq \delta] = 0. \tag{41}$$

Proof. Chebyshev's inequality implies that

$$\Pr[||\mathbf{y}_1(t)|| \geq \delta] \leq \frac{E\left[||\mathbf{y}_1(t)||^2\right]}{\delta^2} = \frac{\text{tr}\left(\left[\mathbf{R}_y(t)\right]_{11}\right)}{\delta^2}. \tag{42}$$

From Proposition 10, $\lim_{t\to\infty}[\mathbf{R}_y(t)]_{11}$ has bounded entries, implying that $\sup_{t\geq 0}\text{tr}([\mathbf{R}_y(t)]_{11}) < \infty$. Taking the limit as $\delta \to \infty$, while relying on the bound in (42) yields the desired result. □

5.3. MSE Performance in Steady-State. Under the stability conditions in Proposition 10, the s.s. covariance matrix $\mathbf{R}_{\tilde{z}}(\infty) := \lim_{t\to\infty}\mathbf{R}_{\tilde{z}}(t)$ has bounded entries. Lemma 9 enables the evaluation of $\mathrm{vec}[\mathbf{R}_{\tilde{z}}(\infty)]$ as a fixed point of (39); thus,

$$\mathrm{vec}[\mathbf{R}_{\tilde{z}}(\infty)] = \left(\mathbf{I}_{(3Jp)^2} - \widetilde{\mathbf{\Psi}}(\mu)\right)^{-1} \mathrm{vec}[\mathbf{R}_{\tilde{\boldsymbol{\nu}}}]. \tag{43}$$

Note that if D-LMS is MSE stable, that is, $\widetilde{\mathbf{\Psi}}(\mu)$ is a stable matrix, matrix $(\mathbf{I}_{(3Jp)^2} - \widetilde{\mathbf{\Psi}}(\mu))^{-1}$ is guaranteed to exist thanks to Gershgorin's circle theorem. Exactly as before, all relevant local and global figures of merit in s.s. can be evaluated provided $[\mathbf{R}_y(\infty)]_{11}$ is available (cf. Section 4.2). Just reshape (43) to obtain $\mathbf{R}_{\tilde{z}}(\infty)$, undo the change of variables to extract $\mathbf{R}_z(\infty)$ from $\mathbf{R}_{\check{z}}(\infty)$, and finally use (28).

While MSE stability ensures, for example, a bounded EMSE(∞), satisfactory tracking of $\mathbf{s}_0(t)$ ultimately requires the error to be small. This will depend on μ and the speed of parameter variation roughly dictated by $\mathrm{tr}(\mathbf{1}_{J\times J} \otimes \mathbf{R}_\zeta) = J\mathrm{tr}(\mathbf{R}_\zeta)$. For simplicity in exposition, consider in the sequel that communication links are ideal so that $\mathbf{R}_{\bar{\boldsymbol{\eta}}_\mu} = \mathbf{R}_{\boldsymbol{\eta}_\mu} = \mathbf{0}_{2Jp\times 2Jp}$ in (32). Interestingly, whenever $J\mathrm{tr}(\mathbf{R}_\zeta)$ is comparable to $4\mu^2 \sum_{j=1}^J \sigma_{\epsilon_j}^2 \mathrm{tr}(\mathbf{\Lambda}_j)$, there exists an optimal $\mu^\star$ minimizing EMSE(∞); see also the numerical examples in Section 6. Because the latter term is $\mathcal{O}(\mu^2)$, $\mathrm{tr}(\mathbf{R}_\zeta)$ should also be small to ensure that EMSE(∞) has an acceptable level. This further implies that D-LMS can track satisfactorily slowly time-varying processes. Inevitable communication-induced delays will affect the D-LMS algorithm, and may further limit the tracking capabilities of the proposed scheme. However, delay analysis falls beyond the scope of the present paper.

The existence of a $\mu^\star$ should not be surprising, given the known results for the centralized LMS algorithm, [20, page 367], [19, page 123]. Excessive adaptation leads to the same MSE inflation as in the absence of parameter variation, while if μ is too small the tracking ability may be lost and once again an MSE penalty is expected. To gain some insight into this tradeoff for the D-LMS algorithm, recall from Section 4.2 and (43) that

$$\begin{aligned}
\mathrm{EMSE}(\infty) &= \frac{1}{J}\sum_{j=1}^{J}\mathrm{tr}\left(\mathbf{R}_{h_j}\left[\mathbf{R}_y(\infty)\right]_{11,j}\right) \\
&= \frac{1}{J}\sum_{j=1}^{J}\mathrm{tr}\left(\mathbf{\Lambda}_j[\mathbf{R}_{\tilde{z}}(\infty)]_{11,j}\right) \\
&= \frac{1}{J}\sum_{j=1}^{J}\mathbf{q}_j^T \mathrm{vec}[\mathbf{R}_{\tilde{z}}(\infty)] \\
&= \frac{1}{J}\sum_{j=1}^{J}\mathbf{q}_j^T\left(\mathbf{I}_{(3Jp)^2} - \widetilde{\mathbf{\Psi}}(\mu)\right)^{-1} \mathrm{vec}[\mathbf{R}_{\tilde{\boldsymbol{\nu}}}],
\end{aligned} \tag{44}$$

where in obtaining the third equality we used that $\mathrm{tr}(\mathbf{R}^T\mathbf{S}) = \mathrm{vec}[\mathbf{R}]^T \mathrm{vec}[\mathbf{S}]$, and the $\{\mathbf{q}_j\}_{j=1}^J$ were defined as in Lemma 9. Now, in the absence of communication noise (cf. (32))

$$\begin{aligned}
\mathbf{R}_{\tilde{\boldsymbol{\nu}}} &= 4\mu^2 b\mathrm{diag}\left(\sigma_{\epsilon_1}^2\mathbf{\Lambda}_1,\ldots,\sigma_{\epsilon_J}^2\mathbf{\Lambda}_J,\mathbf{0}_{2Jp\times 2Jp}\right) \\
&\quad + \widetilde{\mathbf{U}}\left(\left([-1\ 0\ 1]^T[-1\ 0\ 1]\right)\otimes(\mathbf{1}_{J\times J}\otimes\mathbf{R}_\zeta)\right)\widetilde{\mathbf{U}}^T
\end{aligned} \tag{45}$$

so that the term due to observation noise is $\mathcal{O}(\mu^2)$, and the second summand due to parameter nonstationarities is $\mathcal{O}(1)$. Roughly, $(\mathbf{I}_{(3Jp)^2} - \widetilde{\mathbf{\Psi}}(\mu))^{-1} = \mathcal{O}(\mu^{-1})$ and one finds from (44) that EMSE(∞) $= \mathcal{O}(\mu^{-1})$ for small μ, whereas EMSE(∞) $= \mathcal{O}(\mu)$ for moderate- to large values of the step-size approaching the stability bound. This advocates the existence of an optimal step-size $\mu^\star$ minimizing the s.s. EMSE. Unfortunately, deriving an explicit formula for $\mu^\star$ is a formidable task. If needed however, 1−D minimization can be carried out numerically using, for example, Newton's method, as the derivatives of the EMSE(∞) cost in (44) are readily computable in closed form.

If $J\mathrm{tr}(\mathbf{R}_\zeta) \gg 4\mu^2 \sum_{j=1}^J \sigma_{\epsilon_j}^2 \mathrm{tr}(\mathbf{\Lambda}_j)$, then $\mathbf{R}_{\tilde{\boldsymbol{\nu}}} \approx \widetilde{\mathbf{U}}(([-1\ 0\ 1]^T[-1\ 0\ 1]) \otimes (\mathbf{1}_{J\times J} \otimes \mathbf{R}_\zeta))\widetilde{\mathbf{U}}^T$ in the whole range of stable step-sizes so that EMSE(∞) $= \mathcal{O}(\mu^{-1})$, and will not attain a minimum. To achieve the best tracking performance in this scenario, the step-size should be chosen as large as possible while ensuring stability. The other extreme $J\mathrm{tr}(\mathbf{R}_\zeta) \ll 4\mu^2 \sum_{j=1}^J \sigma_{\epsilon_j}^2 \mathrm{tr}(\mathbf{\Lambda}_j)$ corresponds to a small degree of nonstationarity, which in the limit $\mathbf{R}_\zeta \to \mathbf{0}_{p\times p}$ leads to the time-invariant parameter model in (a4.1). Then, $\mathbf{R}_{\tilde{\boldsymbol{\nu}}} \approx 4\mu^2 b\mathrm{diag}(\sigma_{\epsilon_1}^2\mathbf{\Lambda}_1,\ldots,\sigma_{\epsilon_J}^2\mathbf{\Lambda}_J,\mathbf{0}_{2Jp\times 2Jp})$, and as expected EMSE($\infty$) $= \mathcal{O}(\mu)$. The s.s. error can be reduced as much as needed by choosing μ sufficiently small, but this comes at the price of reduced convergence rates.

6. Numerical Tests

Here we corroborate the analytical results of Section 5 through numerical experiments. Substantiating the comments in Remark 4, the usefulness of the analysis is corroborated as the results extend accurately beyond the white Gaussian data setting, allowing for correlated data provided the step-size is small enough. For $J = 20$ sensors, a connected ad hoc WSN is generated as a realization of the random geometric graph model on the unity square, with communication range $r = 0.3$. Hence, sensors are deployed uniformly at random over $[0,1]^2$ and an edge joining two sensors is included in $\mathcal{E}$ whenever their Euclidean distance does not exceed r; see Figure 1. To model noisy links, additive white Gaussian noise (AWGN) with variance $\sigma_\eta^2 = 10^{-2}$ is added at the receiving end.

With $p = 4$, observations obey a linear model (cf. (a1)) with sensing WGN of spatial variance profile $\sigma_{\epsilon_j}^2 = 10^{-1}\alpha_j$, where $\alpha_j \sim \mathcal{U}[0,1]$ (uniform distribution) and i.i.d.. The regression vectors $\mathbf{h}_j(t) = [h_j(t)\cdots h_j(t-p+1)]^T$ have a shift structure and entries which evolve according to $h_j(t) = (1-\rho)\beta_j h_j(t-1) + \sqrt{\rho}\omega_j(t)$ for all $j \in \mathcal{J}$. We choose $\rho = 5\times 10^{-1}$, the $\beta_j \sim \mathcal{U}[0,1]$ i.i.d. in space, and the driving white noise $\omega_j(t) \sim \mathcal{U}[-\sqrt{3}\sigma_{\omega_j}, \sqrt{3}\sigma_{\omega_j}]$ has a spatial variance profile given by $\sigma_{\omega_j}^2 = 2\gamma_j$ with $\gamma_j \sim \mathcal{U}[0,1]$ and i.i.d.. The local regressor's covariance matrices $\mathbf{R}_{h_j}$ have symmetric Toeplitz structure, whereby the elements on the ith diagonal are $[\mathbf{R}_{h_j}]_{i+l,l} = [(1-\rho)\beta_j]^i\rho\sigma_{\omega_j}^2/(1-[(1-\rho)\beta_j]^2)$ for $i = 0,1,\ldots,p-1$ and $1 \leq i+l \leq p$ ($i = 0$ corresponds to the main diagonal). Observe that the data is temporally-correlated and non-Gaussian, implying that (a6) does not hold here. Two

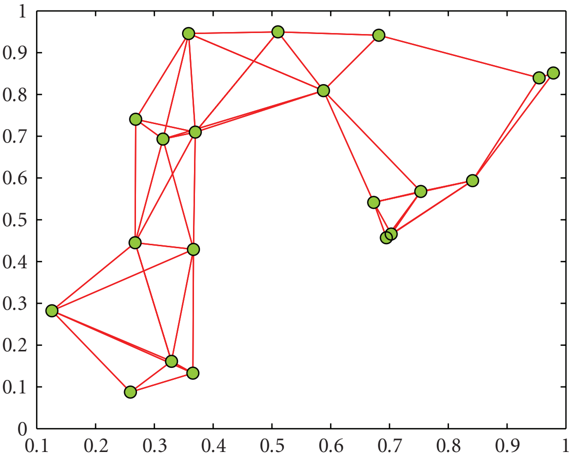
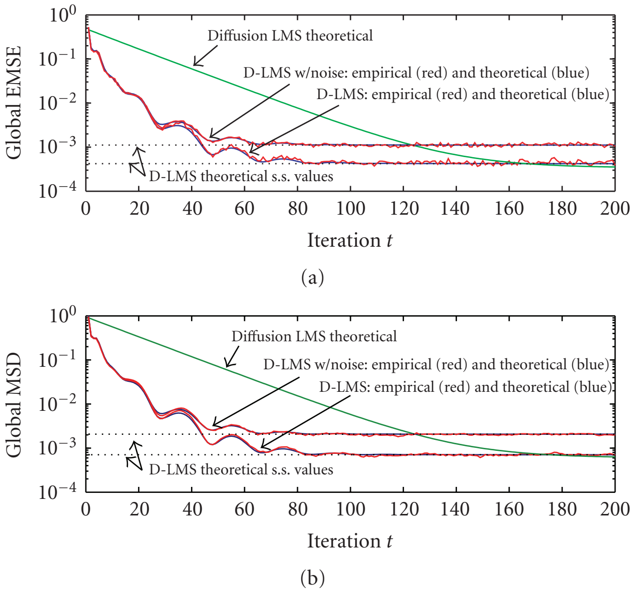


FIGURE 1: An ad hoc WSN with $J = 20$ sensors.



(a)

(b)

FIGURE 2: Global performance evaluation for a time-invariant parameter.

test cases will be considered with regards to the nature of $\mathbf{s}_0(t)$:

TC1: Large-amplitude slowly time-varying parameters adhering to (a4) with $\mathbf{s}_0 = \mathbf{0}_p$ and $\mathbf{\Theta} = (1 - 10^{-4})\,\text{diag}(\theta_1, \ldots, \theta_p)$ with $\theta_i \sim \mathcal{U}[0, 1]$ for $i = 1, \ldots, p$. The driving noise is normally distributed with $\mathbf{R}_\zeta = 10^{-4}\mathbf{I}_p$.

TC2: Time-invariant parameters adhering to (a4.1), with $\mathbf{s}_0 = \mathbf{1}_p$.

For all experimental performance curves obtained by running the algorithms, the ensemble averages are approximated via sample averaging 500 runs of the experiment.

First, under TC2 and $\mu = 5 \times 10^{-2}$, $c = 1$ for D-LMS, Figure 2 depicts the network performance through the evolution of EMSE(t) and MSD(t) figures of merit. Both noisy and ideal links are considered, while for the latter case the D-LMS variant in Section 3.2 has been used. Even though the simulated data does not adhere to (a6), the empirical curves closely follow the theoretical trajectories evaluated via Proposition 8 (and the formulae in Section 4.2). The s.s. limiting values found in Section 5.3 are also extremely accurate. As intuitively expected and analytically corroborated via the noise-related additive terms in (28) and (32), the performance penalty due to non-ideal links is also apparent. Theoretical error trajectory curves for the diffusion LMS [6, equations (73)-(74)] with Metropolis combining weights are also included. While in this case diffusion LMS has a slight edge on s.s. performance, note that it comes at the price of a much slower convergence rate. Similar overall conclusions can be drawn from the plots in Figure 3, that gauge local performance of two randomly selected representative sensors. Even though the noise levels of both sensors are dissimilar ($\sigma^2_{\epsilon_3} = 7.2 \times 10^{-2}$ and $\sigma^2_{\epsilon_{12}} = 2.3 \times 10^{-2}$), effective percolation of information across the WSN renders the s.s performance of both sensors very simliar. The curves for D-LMS with noisy links have been removed in the interest of clarity.

Turning our attention to the tracking performance of the D-LMS algorithm, Figures 2 and 3 are reproduced under TC1 as Figures 4 and 5. Once more, it is appealing how well the theoretical findings in Section 5.2 agree with the true behavior for all $t \geq 0$. Curves for diffusion LMS are not included as time-varying parameters have not been considered in [6]. To conclude, Figure 6 corroborates the discussion in Section 5.3, by showing the theoretically assessed dependence of the s.s. global quantities EMSE(∞) and MSD(∞) on μ, under both TC1 and TC2. While the trend is similar for moderate- to large step-sizes, for small μ the MSE penalty in the tracking setup due to lack of adaptation becomes dominant, and is increasingly severe as $\mu \to 0$. The existence of $\mu^\star \approx 5 \times 10^{-2}$ is also highlighted by Figure 6. From another perspective, Figure 7 illustrates how the adaptation level affects the resulting per-sensor estimates when tracking time-varying parameters with D-LMS. Under TC1 and for $\mu = 5 \times 10^{-4}$ (slow adaptation; see also Figure 6) and $\mu = 5 \times 10^{-2}$ (near optimal adaptation), we depict the third entry of the parameter vector $[\mathbf{s}_0(t)]_3$ and the respective estimates from the randomly chosen sixth sensor. Under optimal adaptation the sensor estimate closely follows the true variations, while—as expected—for the smaller step-size D-LMS fails to provide an accurate estimate.

7. Concluding Remarks

We developed a distributed LMS-type adaptive estimation/tracking algorithm for use in ad hoc WSNs, in which sensors communicate only with their single-hop neighbors via noisy wireless links. The crux of our approach is to reformulate the convex global LMS estimator as a separable constrained optimization problem, which is well-suited for distributed implementation. Capitalizing on this favorable

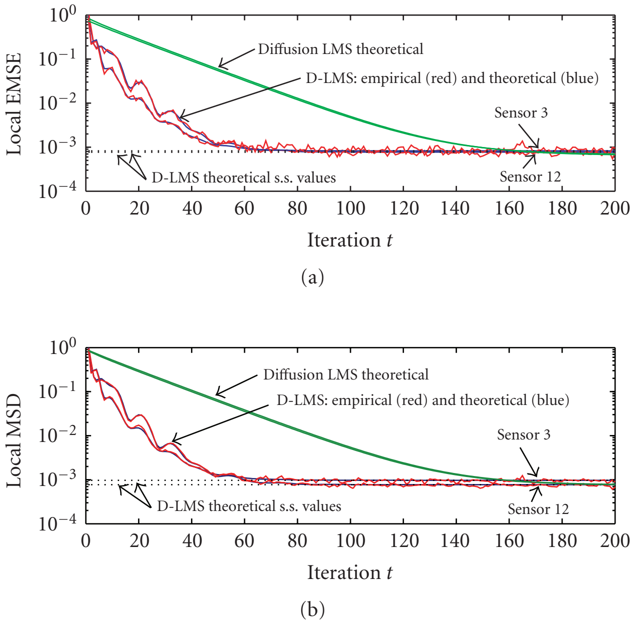


FIGURE 3: Local performance evaluation for a time-invariant parameter: empirical and theoretical curves for sensors 3 and 12 under ideal links.

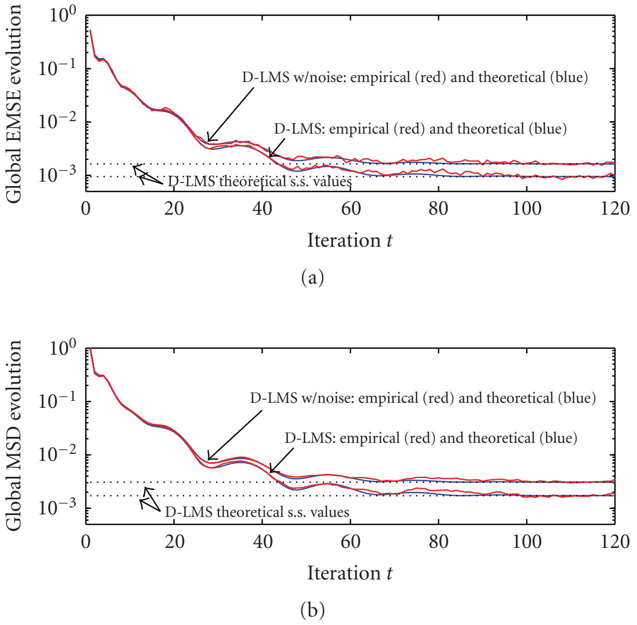


FIGURE 4: Global performance evaluation for a time-varying parameter.

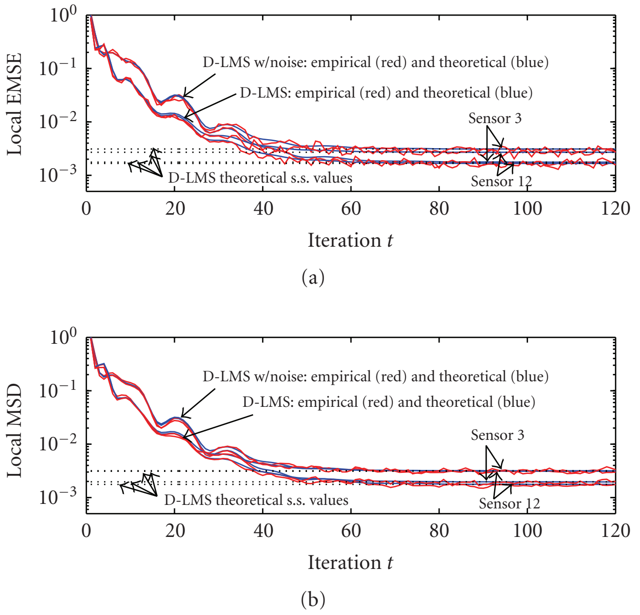


FIGURE 5: Local performance evaluation for a time-varying parameter: empirical and theoretical curves for sensors 3 and 12, both with and without communication noise.

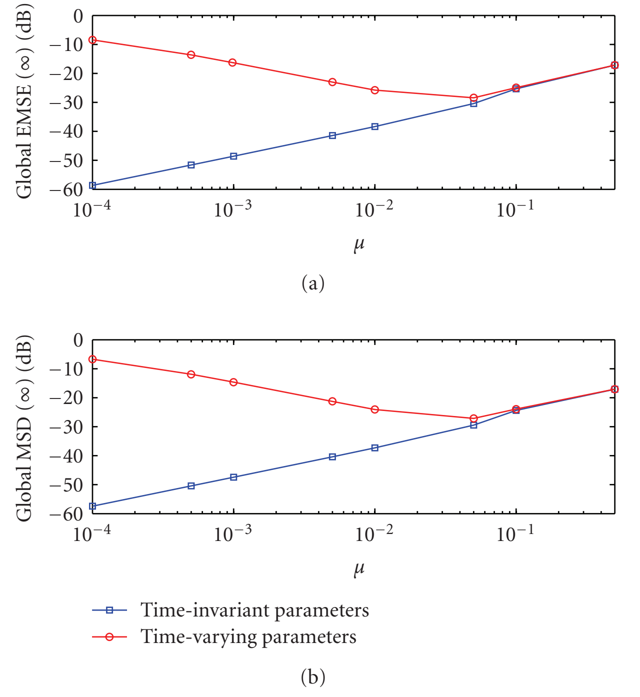


FIGURE 6: Global s.s. EMSE and MSD versus step-size dependencies.

structure through the AD-MoM, we arrived after using a stochastic approximation iteration to simple adaptive recursions executed locally per sensor. Sensors percolate their updated local estimate and Lagrange multipliers in the neighborhood, a means to efficiently and robustly disseminate newly acquired sensor data across the WSN. The novel scheme does not require a Hamiltonian cycle or a subset of bridge sensors, and can tackle linear regression problems in which a statistical data model is not available. When communication noise is not an issue, a cost-effective variant

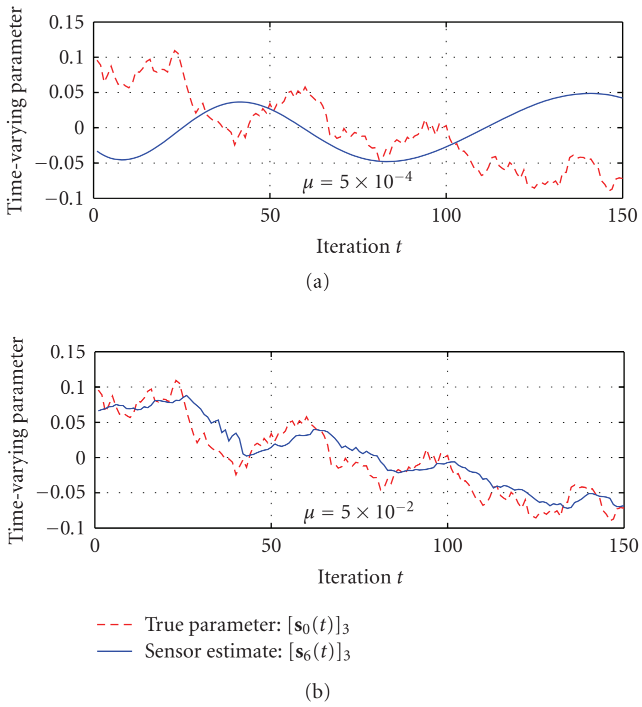


Figure 7: Tracking with D-LMS: slow and optimal adaptation levels.

of D-LMS can be used which circumvents communicating Lagrange multipliers yet incurs no performance penalty.

A detailed MSE tracking performance analysis was conducted for D-LMS, when the parameter fluctuations adhere to a stable first-order AR model. By deriving an exact recursion for the global error covariance matrix under the white Gaussian setting assumptions, the network-wide and per-sensor performance metrics became available for any t, and in particular as $t \to \infty$. D-LMS was shown stable in the mean and MSE-sense in the presence of additive receiver noise, provided $\mu > 0$ is sufficiently small. As a corollary, the resulting local estimation errors satisfy the WSB property and hence remain within a finite interval with overwhelming probability. The tracking analysis led to the conclusion that—different from the time-invariant case whereby one should decrease μ to reduce the s.s. error—for a slowly time-varying parameter there exists an optimal $\mu^{\star}$. While a vanishing step-size renders D-LMS incapable of adapting to the underlying variations, a large one amplifies both observation and communication noise. Numerical simulations demonstrated that the analytical findings of this paper carry over to more pragmatic setups, including temporally correlated (non-) Gaussian sensor data.

Appendices

A. Proof of (9)-(10)

Starting with [S3], observe that the per sensor decomposable structure of (2) is also present in the the augmented Lagrangian. Thus, (8) decouples into $\sum_{j=1}^{J} |\mathcal{N}_j|$ quadratic sub-problems

$$\begin{aligned}\mathbf{z}_j^{j'}(t+1) &= \arg\min_{\mathbf{z}_j^{j'}} \Big[-\left[\mathbf{v}_j^{j'}(t) + \mathbf{u}_{j'}^{j}(t)\right]^T \mathbf{z}_j^{j'} \\ &\quad + \frac{c}{2}\left[\left\|\mathbf{s}_j(t+1) - \mathbf{z}_j^{j'}\right\|^2 + \left\|\mathbf{s}_{j'}(t+1) - \mathbf{z}_j^{j'}\right\|^2\right]\Big] \end{aligned} \tag{A.1}$$

which admit the closed-form solutions

$$\begin{aligned}\mathbf{z}_j^{j'}(t+1) &= \frac{1}{2c}\left[\mathbf{v}_j^{j'}(t) + \mathbf{u}_{j'}^{j}(t)\right]^T \\ &\quad + \frac{1}{2}\left[\mathbf{s}_j(t+1) + \mathbf{s}_{j'}(t+1)\right], \quad j \in \mathcal{J},\ j' \in \mathcal{N}_j. \end{aligned} \tag{A.2}$$

Using (A.2) to eliminate $\mathbf{z}_j^{j'}(t)$ and $\mathbf{z}_{j'}^{j}(t)$ from (5) and (6) respectively, a simple induction argument establishes that if the initial Lagrange multipliers obey $\mathbf{v}_j^{j'}(-1) = -\mathbf{u}_{j'}^{j}(-1)$, then $\mathbf{v}_j^{j'}(t) = -\mathbf{u}_{j'}^{j}(t)$ for all $t \geq 0$ where $j \in \mathcal{J}$ and $j' \in \mathcal{N}_j$. The set $\mathbf{u}$ of multipliers has been shown redundant, and (A.2) readily simplifies to

$$\mathbf{z}_j^{j'}(t+1) = \frac{1}{2}\left[\mathbf{s}_j(t+1) + \mathbf{s}_{j'}(t+1)\right], \quad j \in \mathcal{J},\ j' \in \mathcal{N}_j. \tag{A.3}$$

The symmetry in (A.3) implies that $\mathbf{z}_j^{j'}(t) = \mathbf{z}_{j'}^{j}(t)$ for all $t \geq 0$. Upon substituting (A.3) in (5), the validity of (9) follows readily.

Next, observe that the optimization (7) in [S2] can be split into J sub-problems

$$\begin{aligned}\mathbf{s}_j(t+1) &= \arg\min_{\mathbf{s}_j} \Bigg[E\left[\left(x_j(t+1) - \mathbf{h}_j^T(t+1)\mathbf{s}_j\right)^2\right] \\ &\quad + \sum_{j' \in \mathcal{N}_j} \left[\mathbf{v}_j^{j'}(t) + \mathbf{u}_j^{j'}(t)\right]^T \mathbf{s}_j \\ &\quad + \frac{c}{2}\sum_{j' \in \mathcal{N}_j}\left[\left\|\mathbf{s}_j - \mathbf{z}_j^{j'}(t)\right\|^2 + \left\|\mathbf{s}_j - \mathbf{z}_{j'}^{j}(t)\right\|^2\right]\Bigg]. \end{aligned} \tag{A.4}$$

To arrive at (10), use the identities: (i) $\mathbf{u}_j^{j'}(t) = -\mathbf{v}_{j'}^{j}(t)$ to eliminate $\mathbf{u}_j^{j'}(t)$ from (A.4); and (ii) $\mathbf{z}_j^{j'}(t) = \mathbf{z}_{j'}^{j}(t)$ to recognize that the two quadratic terms in the last summand of (A.4) are identical, while $\mathbf{z}_j^{j'}(t)$ can be eliminated using (A.3).

B. Proof of Lemma 5

Introduce first the $Jp \times 1$ communication noise supervectors $\boldsymbol{\eta}_\alpha(t) := [(\boldsymbol{\eta}_1^\alpha(t))^T \cdots (\boldsymbol{\eta}_J^\alpha(t))^T]^T$ and $\boldsymbol{\eta}_\beta(t) := [(\boldsymbol{\eta}_1^\beta(t))^T \cdots (\boldsymbol{\eta}_J^\beta(t))^T]^T$, where for $j \in \mathcal{J}$

$$\boldsymbol{\eta}_j^\alpha(t) := \frac{c}{2} \sum_{j' \in \mathcal{N}_j} \boldsymbol{\eta}_j^{j'}(t), \qquad \boldsymbol{\eta}_j^\beta(t) := \frac{c}{2} \sum_{j' \in \mathcal{N}_j} \boldsymbol{\eta}_{j'}^{j}(t). \tag{B.1}$$

In order to relate these vectors with $\boldsymbol{\eta}(t)$ in (19), we introduce two $Jp \times (\sum_{j=1}^J |\mathcal{N}_j|)p$ matrices $\mathbf{P}_\alpha := [\mathbf{p}_1 \cdots \mathbf{p}_J]^T$ and $\mathbf{P}_\beta := [\mathbf{p}_1' \cdots \mathbf{p}_J']^T$. The $(\sum_{j=1}^J |\mathcal{N}_j|)p \times p$ submatrices $\mathbf{p}_j$, $\mathbf{p}_j'$ are given by $\mathbf{p}_j' := [(\mathbf{p}_{j,1}')^T \cdots (\mathbf{p}_{j,J}')^T]^T$ and $\mathbf{p}_{j,r}' := [(\mathbf{p}_{j,1}')^T \cdots (\mathbf{p}_{j,J}')^T]^T$, with $\mathbf{p}_{j,r}$, $\mathbf{p}_{j',r}$ defined for $r = 1, \ldots, J$ as

$$\mathbf{p}_{j,r}^T := \begin{cases} \dfrac{c}{2}\mathbf{b}_{|\mathcal{N}_r|,r(j)}^T \otimes \mathbf{I}_p & \text{if } j \in \mathcal{N}_r, \\ \mathbf{0}_{p \times |\mathcal{N}_r|p} & \text{if } j \notin \mathcal{N}_r, \end{cases}$$
$$\left(\mathbf{p}_{j,r}'\right)^T := \begin{cases} \dfrac{c}{2}\mathbf{1}_{1 \times |\mathcal{N}_r|} \otimes \mathbf{I}_p & \text{if } r = j, \\ \mathbf{0}_{p \times |\mathcal{N}_r|p} & \text{if } r \neq j. \end{cases} \tag{B.2}$$

Note that $r(j) \in \{1, \ldots, |\mathcal{N}_r|\}$ denotes the order in which $\boldsymbol{\eta}_j^r(t)$ appears in $\{\boldsymbol{\eta}_{j'}^r(t)\}_{j' \in \mathcal{N}_r}$ (cf. (19)). It is straightforward to verify that $\boldsymbol{\eta}_\alpha(t) = \mathbf{P}_\alpha \boldsymbol{\eta}(t)$ and $\boldsymbol{\eta}_\beta(t) = \mathbf{P}_\beta \boldsymbol{\eta}(t)$.

The proof entails two steps, the first one being summarized in the following lemma.

Lemma 12. *Under (a1) and for $t \geq 0$, the global state $\mathbf{y}(t)$ evolves according to*

$$\begin{aligned} \mathbf{y}(t+1) = {} & \boldsymbol{\Upsilon}(t+1,\mu)\mathbf{y}(t) + \begin{bmatrix} \boldsymbol{\epsilon}(t+1) \\ \mathbf{0} \end{bmatrix} \\ & - \begin{bmatrix} \mathbf{1}_J \otimes (\mathbf{s}_0(t+1) - \mathbf{s}_0(t)) \\ \mathbf{0} \end{bmatrix} \\ & + \begin{bmatrix} \mu \mathbf{I}_{Jp} \\ \mathbf{0} \end{bmatrix} \overline{\boldsymbol{\eta}}(t) + \begin{bmatrix} \mu 3 \mathbf{I}_{Jp} \\ -\mathbf{I}_{Jp} \end{bmatrix} \boldsymbol{\eta}_\alpha(t) - \begin{bmatrix} \mu \mathbf{I}_{Jp} \\ -\mathbf{I}_{Jp} \end{bmatrix} \boldsymbol{\eta}_\beta(t), \end{aligned} \tag{B.3}$$

where the $2Jp \times 2Jp$ transition matrix $\boldsymbol{\Upsilon}(t,\mu)$ consists of the $Jp \times Jp$ blocks $[\boldsymbol{\Upsilon}(t,\mu)]_{11} = \mathbf{I}_{Jp} - 2\mu(\mathbf{R}_h(t) + \mathbf{L}_c)$, $[\boldsymbol{\Upsilon}(t,\mu)]_{12} = -\mu \mathbf{I}_{Jp}$, $[\boldsymbol{\Upsilon}(t,\mu)]_{21} = \mathbf{L}_c$ and $[\boldsymbol{\Upsilon}(t,\mu)]_{22} = \mathbf{I}_{Jp}$. The initial condition $\mathbf{y}(0)$ should be selected as $\mathbf{y}(0) = \text{bdiag}(\mathbf{I}_{Jp}, \mathbf{L}_c)\mathbf{y}'(0)$, where $\mathbf{y}'(0)$ is any vector in $\mathbb{R}^{2Jp}$.

Proof. After summing $\mathbf{v}_j^{j'}(t) - \mathbf{v}_{j'}^{j}(t)$ over $j' \in \mathcal{N}_j$, it follows from (13) that for all $j \in \mathcal{J}$

$$\begin{aligned} \mathbf{y}_{2,j}(t+1) := \sum_{j' \in \mathcal{N}_j} \left(\mathbf{v}_j^{j'}(t) - \mathbf{v}_{j'}^{j}(t) \right) = {} & \mathbf{y}_{2,j}(t) \\ & + c \sum_{j' \in \mathcal{N}_j} \left(\mathbf{s}_j(t) - \mathbf{s}_{j'}(t) \right) \\ & - \frac{c}{2} \sum_{j' \in \mathcal{N}_j} \left(\boldsymbol{\eta}_j^{j'}(t) - \boldsymbol{\eta}_{j'}^{j}(t) \right) \end{aligned} \tag{B.4}$$

$$\begin{aligned} = {} & \mathbf{y}_{2,j}(t) + c \sum_{j' \in \mathcal{N}_j} \left(\mathbf{y}_{1,j}(t) - \mathbf{y}_{1,j'}(t) \right) \\ & - \boldsymbol{\eta}_j^\alpha(t) + \boldsymbol{\eta}_j^\beta(t), \end{aligned} \tag{B.5}$$

where the last equality was obtained after adding and subtracting $c|\mathcal{N}_j|\mathbf{s}_0(t)$ from the rhs of (B.4), and relying on the definitions in (B.1). Next, starting from (14) and upon: (i) using (a1) to eliminate $e_j(t+1) = -\mathbf{h}_j^T(t+1)\mathbf{y}_{1,j}(t) + \epsilon_j(t+1)$ from (14); (ii) subtracting $\mathbf{s}_0(t+1) + \mathbf{s}_0(t)$ from both sides of (14); (iii) replacing the sums of noise vectors with the quantities defined in (18) and (B.1); and (iv) recognizing $\mathbf{y}_{2,j}(t+1)$ in the rhs of (14) and substituting it with (B.5), one arrives at

$$\begin{aligned} \mathbf{y}_{1,j}(t+1) = {} & \mathbf{y}_{1,j}(t) + \mu \Bigg[-2\mathbf{h}_j(t+1)\mathbf{h}_j^T(t+1)\mathbf{y}_{1,j}(t) \\ & -\mathbf{y}_{2,j}(t) - 2c \sum_{j' \in \mathcal{N}_j} \left(\mathbf{y}_{1,j}(t) - \mathbf{y}_{1,j'}(t) \right) \Bigg] \\ & + 3\mu \boldsymbol{\eta}_j^\alpha(t) - \mu \boldsymbol{\eta}_j^\beta(t) + \mu \overline{\boldsymbol{\eta}}_j(t) \\ & + 2\mu \mathbf{h}_j(t+1)\epsilon_j(t+1) - (\mathbf{s}_0(t+1) - \mathbf{s}_0(t)). \end{aligned} \tag{B.6}$$

Again, the term $2c\sum_{j' \in \mathcal{N}_j}(\mathbf{y}_{1,j}(t) - \mathbf{y}_{1,j'}(t))$ inside the square brackets is obtained in error-form after adding and subtracting $2c|\mathcal{N}_j|\mathbf{s}_0(t)$.

What remains to be shown is that after stacking the recursions (B.6) and (B.5) for $j = 1, \ldots, J$ to form the one for $\mathbf{y}(t+1)$, we can obtain the compact representation in (B.3). Consider first the forcing terms in (B.3). Stacking the channel noise terms from (B.6) and (B.5), readily yields the last three terms in (B.3). Likewise, independently stacking the terms $2\mu \mathbf{h}_j(t+1)\epsilon_j(t+1)$ for $j = 1, \ldots, J$ and $\mathbf{s}_0(t+1) - \mathbf{s}_0(t)$ yields the second and third terms in the rhs of (B.3), which are due to the observation noise and parameter velocity, respectively. These terms are not present in (B.5), which explains the zero vectors at the lower part of the second and third terms in (B.3).

To specify the structure of the transition matrix $\boldsymbol{\Upsilon}(t,\mu)$, note that the first term on the rhs of (B.5) explains why $[\boldsymbol{\Upsilon}(t,\mu)]_{22} = \mathbf{I}_{Jp}$. Similarly, the second term inside the square brackets in (B.6) explains why $[\boldsymbol{\Upsilon}(t,\mu)]_{12} = -\mu \mathbf{I}_{Jp}$. Next, it follows readily that upon stacking the terms $c\sum_{j' \in \mathcal{N}_j}(\mathbf{y}_{1,j}(t) - \mathbf{y}_{1,j'}(t))$, which correspond to a scaled Laplacian-based combination of $p \times 1$ vectors, one obtains $c(\mathbf{L} \otimes \mathbf{I}_p)\mathbf{y}_1(t) = \mathbf{L}_c\mathbf{y}_1(t)$. This justifies why $[\boldsymbol{\Upsilon}(t,\mu)]_{21} = \mathbf{L}_c$. Using similar arguments and recalling that $\mathbf{R}_h(t) := b\text{diag}(\mathbf{h}_1(t)\mathbf{h}_1^T(t),\ldots,\mathbf{h}_J(t)\mathbf{h}_J^T(t))$, we establish that $[\boldsymbol{\Upsilon}(t,\mu)]_{11} = \mathbf{I}_{Jp} - 2\mu(\mathbf{R}_h(t) + \mathbf{L}_c)$.

Although the vectors $\{\mathbf{y}_{1,j}(0)\}_{j=1}^J$ are decoupled so that $\mathbf{y}_1(0)$ can be chosen arbitrarily, this is not the case for $\{\mathbf{y}_{2,j}(0)\}_{j=1}^J$ which are coupled and satisfy

$$\begin{aligned}\sum_{j=1}^{J}\mathbf{y}_{2,j}(t) &= \sum_{j=1}^{J}\sum_{j'\in\mathcal{N}_j}\left(\mathbf{v}_j^{j'}(t-1) - \mathbf{v}_{j'}^{j}(t-1)\right)\\ &= \mathbf{0}_p, \quad \forall t \geq 0.\end{aligned} \tag{B.7}$$

The coupling across $\{\mathbf{y}_{2,j}(t)\}_{j=1}^J$ dictates that $\mathbf{y}_2(0)$ should be chosen in compliance with (B.7), so that the system (B.3) is equivalent to D-LMS for all $t \geq 0$. Let $\mathbf{y}_2(0) = \mathbf{L}_c\mathbf{y}_2'(0)$, where $\mathbf{y}_2'(0)$ is any vector in $\mathbb{R}^{Jp}$. Then, $\mathbf{y}_2(0)$ satisfies the conservation law (B.7) as (recall that $\mathbf{1}_J$ = nullspace($\mathbf{L}$))

$$\begin{aligned}\sum_{j=1}^{J}\mathbf{y}_{2,j}(0) &= \left(\mathbf{1}_J^T \otimes \mathbf{I}_p\right)\mathbf{y}_2(0)\\ &= \left(\mathbf{1}_J^T \otimes \mathbf{I}_p\right)c\left(\mathbf{L}\otimes\mathbf{I}_p\right)\mathbf{y}_2'(0)\\ &= c\left(\left(\mathbf{1}_J^T\mathbf{L}\right)\otimes\mathbf{I}_p\right)\mathbf{y}_2'(0)\\ &= \mathbf{0}_p.\end{aligned} \tag{B.8}$$

In conclusion, for arbitrary $\mathbf{y}'(0) \in \mathbb{R}^{2Jp}$ the recursion (B.3) should be initialized as $\mathbf{y}(0) = b\text{diag}(\mathbf{I}_{Jp},\mathbf{L}_c)\mathbf{y}'(0)$, and the proof of Lemma 12 is completed. □

The second step of the proof involves establishing the equivalence between the dynamical systems in (B.3) and (20) for all $t \geq 0$, when the inner state is arbitrarily initialized as $\mathbf{z}(0) = \mathbf{y}'(0)$. We will argue by induction. For $t = 0$, it follows from (21) that $\mathbf{z}(1) = \boldsymbol{\Phi}(1,\mu)\mathbf{y}'(0) + [\boldsymbol{\epsilon}^T(1)\ \mathbf{0}^T]^T - [\mathbf{1}_J^T \otimes (\mathbf{s}_0(1) - \mathbf{s}_0(0))^T\ \mathbf{0}^T]^T$. Upon substituting $\mathbf{z}(1)$ into (20), we find

$$\begin{aligned}\mathbf{y}(1) = b\text{diag}&\left(\mathbf{I}_{Jp},\mathbf{L}_c\right)\boldsymbol{\Phi}(1,\mu)\mathbf{y}'(0)\\ &+ \begin{bmatrix}\boldsymbol{\epsilon}(1)\\ \mathbf{0}\end{bmatrix} - \begin{bmatrix}\mathbf{1}_J\otimes(\mathbf{s}_0(1)-\mathbf{s}_0(0))\\ \mathbf{0}\end{bmatrix}\\ &+ \begin{bmatrix}\mu\mathbf{I}_{Jp}\\ \mathbf{0}\end{bmatrix}\overline{\boldsymbol{\eta}}(0) + \begin{bmatrix}\mu\left(3\mathbf{P}_\alpha - \mathbf{P}_\beta\right)\\ \mathbf{P}_\beta - \mathbf{P}_\alpha\end{bmatrix}\boldsymbol{\eta}(0).\end{aligned} \tag{B.9}$$

Note that: (i) $b\text{diag}(\mathbf{I}_{Jp},\mathbf{L}_c)\boldsymbol{\Phi}(t,\mu) = \boldsymbol{\Upsilon}(t,\mu)b\text{diag}(\mathbf{I}_{Jp},\mathbf{L}_c)$ for all $t \geq 1$; (ii) $\mathbf{y}(0) = b\text{diag}(\mathbf{I}_{Jp},\mathbf{L}_c)\mathbf{y}'(0)$ for the system in Lemma 12; and (iii) $\boldsymbol{\eta}_\alpha(t) = \mathbf{P}_\alpha\boldsymbol{\eta}(t)$, while $\boldsymbol{\eta}_\beta(t) = \mathbf{P}_\beta\boldsymbol{\eta}(t)$. Thus, the rhs of (B.9) is equal to the rhs of (B.3) for $t = 0$.

Suppose next that (20) and (21) hold true for $\mathbf{y}(t)$ and $\mathbf{z}(t)$. The same will be shown for $\mathbf{y}(t+1)$ and $\mathbf{z}(t+1)$. To this end, replace $\mathbf{y}(t)$ with the rhs of (20) evaluated at time instant t, into (B.3) to obtain

$$\begin{aligned}&\mathbf{y}(t+1)\\ &\quad = \boldsymbol{\Upsilon}(t+1,\mu)b\text{diag}\left(\mathbf{I}_{Jp},\mathbf{L}_c\right)\mathbf{z}(t)\\ &\qquad + \boldsymbol{\Upsilon}(t+1,\mu)\begin{bmatrix}\mu\mathbf{I}_{Jp}\\ \mathbf{0}\end{bmatrix}\overline{\boldsymbol{\eta}}(t-1)\\ &\qquad + \boldsymbol{\Upsilon}(t+1,\mu)\begin{bmatrix}\mu\left(3\mathbf{P}_\alpha - \mathbf{P}_\beta\right)\\ \mathbf{P}_\beta - \mathbf{P}_\alpha\end{bmatrix}\boldsymbol{\eta}(t-1)\\ &\qquad + \begin{bmatrix}\boldsymbol{\epsilon}(t+1)\\ \mathbf{0}\end{bmatrix} - \begin{bmatrix}\mathbf{1}_J\otimes(\mathbf{s}_0(t+1)-\mathbf{s}_0(t))\\ \mathbf{0}\end{bmatrix}\\ &\qquad + \begin{bmatrix}\mu\mathbf{I}_{Jp}\\ \mathbf{0}\end{bmatrix}\overline{\boldsymbol{\eta}}(t) + \begin{bmatrix}\mu3\mathbf{I}_{Jp}\\ -\mathbf{I}_{Jp}\end{bmatrix}\boldsymbol{\eta}_\alpha(t) - \begin{bmatrix}\mu\mathbf{I}_{Jp}\\ -\mathbf{I}_{Jp}\end{bmatrix}\boldsymbol{\eta}_\beta(t)\\ &\quad = b\text{diag}\left(\mathbf{I}_{Jp},\mathbf{L}_c\right)\\ &\qquad \times\left(\boldsymbol{\Phi}(t+1,\mu)\mathbf{z}(t) + \begin{bmatrix}\boldsymbol{\epsilon}(t+1)\\ \mathbf{0}\end{bmatrix}\right.\\ &\qquad\qquad - \begin{bmatrix}\mathbf{1}_J\otimes(\mathbf{s}_0(t+1)-\mathbf{s}_0(t))\\ \mathbf{0}\end{bmatrix}\\ &\qquad\qquad + \boldsymbol{\Phi}(t+1,\mu)\begin{bmatrix}\mu\mathbf{I}_{Jp}\\ \mathbf{0}\end{bmatrix}\overline{\boldsymbol{\eta}}(t-1)\\ &\qquad\qquad \left. + \boldsymbol{\Phi}(t+1,\mu)\begin{bmatrix}\mu\left(3\mathbf{P}_\alpha - \mathbf{P}_\beta\right)\\ \mathbf{C}\end{bmatrix}\boldsymbol{\eta}(t-1)\right)\\ &\qquad + \begin{bmatrix}\mu\mathbf{I}_{Jp}\\ \mathbf{0}\end{bmatrix}\overline{\boldsymbol{\eta}}(t) + \begin{bmatrix}\mu\left(3\mathbf{P}_\alpha - \mathbf{P}_\beta\right)\\ \mathbf{P}_\beta - \mathbf{P}_\alpha\end{bmatrix}\boldsymbol{\eta}(t).\end{aligned} \tag{B.10}$$

In obtaining the last equality in (B.10), we used: (i) $b\text{diag}(\mathbf{I}_{Jp},\mathbf{L}_c)\boldsymbol{\Phi}(t,\mu) = \boldsymbol{\Upsilon}(t,\mu)b\text{diag}(\mathbf{I}_{Jp},\mathbf{L}_c)$; (ii) the relationship between $\boldsymbol{\eta}_\alpha(t), \boldsymbol{\eta}_\beta(t)$ and $\boldsymbol{\eta}(t)$; and (iii) the existence of a matrix $\mathbf{C}$ such that $\mathbf{L}_c\mathbf{C} = \mathbf{P}_\beta - \mathbf{P}_\alpha$. This made possible to extract the common factor $b\text{diag}(\mathbf{I}_{Jp},\mathbf{L}_c)$ and deduce from (B.10) that $\mathbf{y}(t+1)$ is given by (20), while $\mathbf{z}(t+1)$ is provided by (21).

In order to complete the proof, we must show the existence of matrix $\mathbf{C}$. To this end, via a simple evaluation one can check that nullspace($\mathbf{L}_c$) ⊆ nullspace($\mathbf{P}_\beta^T - \mathbf{P}_\alpha^T$), and since $\mathbf{L}_c$ is symmetric, we have nullspace($\mathbf{L}_c$) ⊥ range($\mathbf{L}_c$). As nullspace($\mathbf{P}_\beta^T - \mathbf{P}_\alpha^T$) ⊥ range($\mathbf{P}_\beta - \mathbf{P}_\alpha$), it follows that range($\mathbf{P}_\beta - \mathbf{P}_\alpha$) ⊆ range($\mathbf{L}_c$), which further implies that we can find $\mathbf{C}$ such that $\mathbf{L}_c\mathbf{C} = \mathbf{P}_\beta - \mathbf{P}_\alpha$.

C. Structure of $\mathbf{R}_{\overline{\eta}_\mu}$ and $\mathbf{R}_{\eta_\mu}$

From (31) we have

$$\begin{aligned} \mathbf{R}_{\overline{\eta}_\mu} &= \begin{bmatrix} \mu\mathbf{I}_{Jp} \\ \mathbf{0}_{Jp\times Jp} \end{bmatrix} \mathbf{R}_{\overline{\eta}} \begin{bmatrix} \mu\mathbf{I}_{Jp} \\ \mathbf{0}_{Jp\times Jp} \end{bmatrix}^T, \\ \mathbf{R}_{\eta_\mu} &= \begin{bmatrix} \mu\left(3\mathbf{P}_\alpha - \mathbf{P}_\beta\right) \\ \mathbf{C} \end{bmatrix} \mathbf{R}_{\eta} \begin{bmatrix} \mu(3\mathbf{P}_\alpha - \mathbf{P}_\beta) \\ \mathbf{C} \end{bmatrix}^T \end{aligned} \tag{C.1}$$

so that it suffices to focus on the structure of $\mathbf{R}_{\overline{\eta}}$ and $\mathbf{R}_{\eta}$. From the definition in (18) and recalling that communication noise vectors are assumed uncorrelated in space, it follows that

$$\mathbf{R}_{\overline{\eta}} = b\text{diag}\left(\sum_{j'\in\mathcal{N}_1\setminus\{1\}} \mathbf{R}_{\eta_{1,j'}}, \dots, \sum_{j'\in\mathcal{N}_J\setminus\{J\}} \mathbf{R}_{\eta_{J,j'}} \right). \tag{C.2}$$

In the same way it follows from (19) that $\mathbf{R}_{\eta}$ is a block diagonal matrix with a total of $\sum_{j=1}^{J} |\mathcal{N}_j|$ diagonal blocks of size $p \times p$, namely

$$\mathbf{R}_{\eta} = b\text{diag}\left(\left\{\mathbf{R}_{\eta_{j',1}}\right\}_{j'\in\mathcal{N}_1}, \dots, \left\{\mathbf{R}_{\eta_{j',J}}\right\}_{j'\in\mathcal{N}_J} \right). \tag{C.3}$$

Note also that the blocks $\mathbf{R}_{\eta_{j,j}} = \mathbf{0}_{p\times p}$ for all $j \in \mathcal{J}$.

D. Proof of Lemma 9

We separately treat each of the three summands in $\mathcal{M}(\widetilde{\mathbf{\Omega}}(\mu), \mathbf{R}_{\tilde{z}}(t))$ (cf. (33)) and finally add the results. The algebraic property $\text{vec}[\mathbf{RST}] = (\mathbf{T}^T \otimes \mathbf{R})\,\text{vec}[\mathbf{S}]$, allows writing the first term as $(\widetilde{\mathbf{\Omega}}(\mu) \otimes \widetilde{\mathbf{\Omega}}(\mu))\,\text{vec}[\mathbf{R}_{\tilde{z}}(t)]$. The second term can be rewritten as $4\mu^2 b\text{diag}(\mathbf{\Lambda}, \mathbf{0}_{2Jp\times 2Jp})[(\mathbf{I}_{3J} \otimes \mathbf{1}_{p\times p}) \circ \mathbf{R}_{\tilde{z}}(t)]b\text{diag}(\mathbf{\Lambda}, \mathbf{0}_{2Jp\times 2Jp})$, which is vectorized upon using the property for $\text{vec}[\mathbf{RST}]$ followed by $\text{vec}[\mathbf{R} \circ \mathbf{S}] = \text{diag}(\text{vec}[\mathbf{R}])\,\text{vec}[\mathbf{S}]$, to yield

$$\begin{aligned} &4\mu^2 \left(b\text{diag}\left(\mathbf{\Lambda}, \mathbf{0}_{2Jp\times 2Jp}\right) \otimes b\text{diag}\left(\mathbf{\Lambda}, \mathbf{0}_{2Jp\times 2Jp}\right) \right) \\ &\qquad \times \text{diag}\left(\text{vec}\left[\mathbf{I}_{3J} \otimes \mathbf{1}_{p\times p}\right]\right) \text{vec}[\mathbf{R}_{\tilde{z}}(t)]. \end{aligned} \tag{D.1}$$

The third term is a diagonal matrix, which can be decomposed as (up to a constant factor $4\mu^2$)

$$\begin{aligned} &b\text{diag}\left(\text{tr}\left(\mathbf{\Lambda}_1 [\mathbf{R}_{\tilde{z}}(t)]_{11,1}\right)\mathbf{\Lambda}_1, \dots, \text{tr}\left(\mathbf{\Lambda}_J [\mathbf{R}_{\tilde{z}}(t)]_{11,J}\right)\mathbf{\Lambda}_J, \mathbf{0}_{2Jp\times 2Jp}\right) \\ &\quad = \sum_{j=1}^{J} \text{tr}\left(\mathbf{\Lambda}_j [\mathbf{R}_{\tilde{z}}(t)]_{11,j}\right) \left[\text{diag}\left(\mathbf{b}_{3J,j}\right) \otimes \mathbf{\Lambda}_j\right] \\ &\quad = \sum_{j=1}^{J} \text{tr}\left(\left[\text{diag}\left(\mathbf{b}_{3J,j}\right) \otimes \mathbf{\Lambda}_j\right] \mathbf{R}_{\tilde{z}}(t)\right) \left[\text{diag}\left(\mathbf{b}_{3J,j}\right) \otimes \mathbf{\Lambda}_j\right] \\ &\quad = \sum_{j=1}^{J} \left(\text{vec}\left[\text{diag}\left(\mathbf{b}_{3J,j}\right) \otimes \mathbf{\Lambda}_j\right]^T \text{vec}[\mathbf{R}_{\tilde{z}}(t)] \right) \\ &\qquad \times \left[\text{diag}\left(\mathbf{b}_{3J,j}\right) \otimes \mathbf{\Lambda}_j\right], \end{aligned} \tag{D.2}$$

where the last equality follows from the identity $\text{tr}(\mathbf{R}^T\mathbf{S}) = \text{vec}[\mathbf{R}]^T\,\text{vec}[\mathbf{S}]$. Upon scaling and vectorizing (D.2), while letting $\mathbf{q}_j := \text{vec}[\text{diag}(\mathbf{b}_{3J,j}) \otimes \mathbf{\Lambda}_j]$, one obtains $4\mu^2 \sum_{j=1}^{J} \mathbf{q}_j \mathbf{q}_j^T\,\text{vec}[\mathbf{R}_{\tilde{z}}(t)]$. The result follows readily after summing the three vectorized terms and taking $\text{vec}[\mathbf{R}_{\tilde{z}}(t)]$ as common factor.

Acknowledgments

The work in this paper was supported by the USDoD ARO Grant no. W911NF-05-1-0283 and also through collaborative participation in the C&N Consortium sponsored by the US ARL under the CTA Program, Cooperative Agreement DAAD19-01-2-0011. The US Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies of the Army Research Laboratory or the U.S. Government. Part of the paper appeared in the 13th *Digital Signal Proc. Workshop*, Marco Island, Florida, January 4–7, 2009.

References

[1] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[2] F. S. Cattivelli, C. G. Lopes, and A. H. Sayed, "Diffusion recursive least-squares for distributed estimation over adaptive networks," *IEEE Transactions on Signal Processing*, vol. 56, no. 5, pp. 1865–1877, 2008.

[3] C. G. Lopes and A. H. Sayed, "Incremental adaptive strategies over distributed networks," *IEEE Transactions on Signal Processing*, vol. 55, no. 8, pp. 4064–4077, 2007.

[4] S. S. Ram, A. Nedic, and V. V. Veeravalli, "Stochastic incremental gradient descent for estimation in sensor networks," in *Proceedings of the 41st Asilomar Conference on Signals, Systems and Computers*, pp. 582–586, Pacific Grove, Calif, USA, 2007.

[5] C. H. Papadimitriou, *Computational Complexity*, Addison-Wesley, 1993.

[6] C. G. Lopes and A. H. Sayed, "Diffusion least-mean squares over adaptive networks: formulation and performance analysis," *IEEE Transactions on Signal Processing*, vol. 56, no. 7, pp. 3122–3136, 2008.

[7] F. S. Cattivelli and A. H. Sayed, "Diffusion LMS algorithms with information exchange," in *Proceedings of the 42nd Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, Calif, USA, November 2008.

[8] S. S. Stankovic, M. S. Stankovic, and D. S. Stipanovic, "Decentralized parameter estimation by consensus based stochastic approximation," in *Proceedings of the 46th Conference on Decision and Control*, pp. 1535–1540, New Orleans, La, USA, December 2007.

[9] A. H. Sayed and C. G. Lopes, "Distributed recursive least-squares over adaptive networks," in *Proceedings of the 40th Asilomar Conference on Signals, Systems and Computers*, pp. 233–237, Pacific Grove, Calif, USA, October-November 2006.

[10] G. Mateos, I. D. Schizas, and G. B. Giannakis, "Distributed recursive least-squares for consensus- based in-network adaptive estimation," *IEEE Transactions on Signal Processing*, vol. 57, no. 11, pp. 4583–4588, 2009.

[11] L. Xiao, S. Boyd, and S. Lall, "A space-time diffusion scheme for peer-to-peer least-squares estimation," in *Proceedings of the International Conference on Information Processing in Sensor Networks*, pp. 168–176, Nashville, Tenn, USA, 2006.

[12] F. S. Cattivelli, C. G. Lopes, and A. H. Sayed, "Diffusion strategies for distributed Kalman filtering: formulation and performance analysis," in *Proceedings of the IAPR Workshop on Cognitive Information Processing*, Santorini, Greece, June 2008.

[13] I. D. Schizas, G. B. Giannakis, S. I. Roumeliotis, and A. Ribeiro, "Consensus in ad hoc WSNs with noisy links—part II: distributed estimation and smoothing of random signals," *IEEE Transactions on Signal Processing*, vol. 56, no. 4, pp. 1650–1666, 2008.

[14] R. Olfati-Saber, "Distributed Kalman filtering for sensor networks," in *Proceedings of the 46th Conference on Decision and Control*, pp. 5492–5498, New Orleans, La, USA, December 2007.

[15] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, Athena-Scientific, 2nd edition, 1999.

[16] I. D. Schizas, G. Mateos, and G. B. Giannakis, "Distributed LMS for consensus-based in-network adaptive processing," *IEEE Transactions on Signal Processing*, vol. 57, no. 6, pp. 2365–2382, 2009.

[17] I. D. Schizas, A. Ribeiro, and G. B. Giannakis, "Consensus in ad hoc WSNs with noisy links—part I: distributed estimation of deterministic signals," *IEEE Transactions on Signal Processing*, vol. 56, no. 1, pp. 350–364, 2008.

[18] V. Solo, "The stability of LMS," *IEEE Transactions on Signal Processing*, vol. 45, no. 12, pp. 3017–3026, 1997.

[19] V. Solo and X. Kong, *Adaptive Signal Processing Algorithms: Stability and Performance*, Prentice-Hall, Upper Saddle River, NJ, USA, 1995.

[20] A. H. Sayed, *Fundamentals of Adaptive Filtering*, John Wiley & Sons, New York, NY, USA, 2003.

[21] N. R. Yousef and A. H. Sayed, "A unified approach to the steady-state and tracking analyses of adaptive filters," *IEEE Transactions on Signal Processing*, vol. 49, no. 2, pp. 314–324, 2001.

[22] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*, Wiley-Interscience, New York, NY, USA, 2001.

[23] H. J. Kushner and G. G. Yin, *Stochastic Approximation and Recursive Algorithms and Applications*, Springer, Berlin, Germany, 2nd edition, 2003.

[24] Y. Hatano, A. K. Das, and M. Mesbahi, "Agreement in presence of noise: pseudogradients on random geometric networks," in *Proceedings of the 44th Conference on Decision and Control*, pp. 6382–6387, Seville, Spain, December 2005.

[25] S. Kar and J. M. F. Moura, "Distributed consensus algorithms in sensor networks with imperfect communication: link failures and channel noise," *IEEE Transactions on Signal Processing*, vol. 57, no. 1, pp. 355–369, 2009.

[26] G. Mateos, I. D. Schizas, and G. B. Giannakis, "Closed-from MSE performance of the distributed LMS algorithm," in *Proceedings of the 13th Digital Signal Processing Workshop*, Marco Island, Fla, USA, January 2009.

[27] T. Minka, "Old and new matrix algebra useful for statistics," December 2000, http://research.microsoft.com/en-us/um/people/minka/papers/matrix/.