Diffusion LMS Strategies in Sensor Networks With Noisy Input Data

Reza Abdolee, *Student Member, IEEE*, and Benoit Champagne, *Senior Member, IEEE*

Abstract—We investigate the performance of distributed least-mean square (LMS) algorithms for parameter estimation over sensor networks where the regression data of each node are corrupted by white measurement noise. Under this condition, we show that the estimates produced by distributed LMS algorithms will be biased if the regression noise is excluded from consideration. We propose a bias-elimination technique and develop a novel class of diffusion LMS algorithms that can mitigate the effect of regression noise and obtain an unbiased estimate of the unknown parameter vector over the network. In our development, we first assume that the variances of the regression noises are known *a priori*. Later, we relax this assumption by estimating these variances in real time. We analyze the stability and convergence of the proposed algorithms and derive closed-form expressions to characterize their mean-square error performance in transient and steady-state regimes. We further provide computer experiment results that illustrate the efficiency of the proposed algorithms and support the analytical findings.

Index Terms—Bias-compensated least-mean square (LMS), diffusion adaptation, distributed parameter estimation, network optimization.

I. INTRODUCTION

ONE OF the critical issues encountered in distributed parameter estimation over sensor networks is the distortion of the collected regression data by noise, which occurs when the local copy of the underlying system input signal at each node is corrupted by various sources of impairments such as measurement or quantization noise. This problem has been extensively investigated for the case of single-node processing devices [2]–[17]. These studies have shown that if the deleterious effect of the input noise is not taken into account, the parameter estimates so obtained will be inaccurate and biased. Various practical solutions have been suggested to mitigate the effect of the input measurement noise or to remove the bias from the resulting estimates [5]–[17]. These solutions, however, may no longer lead to optimal results in sensor networks with decentralized processing structure where the data measurement and parameter estimation are performed at multiple processing nodes in parallel and with cooperation.

For networking applications, a distributed total-least-squares (DTLS) algorithm has been proposed that is developed using semidefinite relaxation and convex semidefinite programming [18]. This algorithm mitigates the effect of white input noise by running a local TLS algorithm at each sensor node and exchanging the locally estimated parameters between the nodes for further refinement. The DTLS algorithm computes the eigendecomposition of an augmented covariance matrix at every iteration for all nodes in the network, and is therefore mainly suitable for applications involving nodes with powerful processing abilities. In a follow-up paper, the same authors proposed a low-complexity DTLS algorithm [19] that uses an inverse power iteration technique to reduce the computational complexity of the DTLS while demanding lower communication power.

In recent years, several classes of distributed *adaptive* algorithms for parameter estimation over networks have been proposed, including incremental [20]–[23], consensus [24]–[27], [27]–[30], and diffusion algorithms [31]–[42]. Incremental techniques require the definition of a cyclic path over the nodes, which is generally an NP-hard problem; these techniques are also sensitive to link failures. Consensus techniques require doubly-stochastic combination policies and, when used in the context of adaptation with constant step-sizes, can lead to unstable behavior even if all individual nodes can solve the inference task in a stable manner [38]. In this work, we focus on diffusion strategies because they have been shown to be more robust and to lead to a stable behavior regardless of the underlying topology, even when some of the underlying nodes are unstable [38].

A bias-compensated diffusion-based recursive least-squares (RLS) algorithm has been developed in [43] that can obtain unbiased estimates of the unknown system parameters over sensor networks, where the regression data are distorted by colored noise. While this algorithm offers fast convergence speed, its high computational complexity and numerical instability may be a hindrance in some applications. In contrast, the diffusion least-mean square (LMS) algorithms are characterized by low complexity and numerical stability. Motivated by these features, in this paper, we investigate the performance of standard diffusion LMS algorithms [31]–[33] over sensor networks where the input regression data are corrupted by additive white noise. To overcome the limitations of these algorithms, as exposed by our analysis under this scenario, we then propose an alternative problem formulation that leads to a novel class of diffusion LMS algorithms, which we call bias-compensated diffusion strategies.

More specifically, we first show that in the presence of noisy input data, the parameter estimates produced by standard diffusion LMS algorithms are biased. We then reformulate this

estimation problem in terms of an alternative cost function and develop bias-compensated diffusion LMS strategies that can produce unbiased estimates of the system parameters. The development of these algorithms relies on a bias-elimination strategy that assumes prior knowledge about the regression noise variances over the network. The analysis results show that if the step-sizes are within a given range, the algorithms will be stable in the mean and mean-square sense and the estimated parameters will converge to their true values. Finally, we relax the known variance assumption by incorporating a recursive approach into the algorithm to estimate the variances in real time.

In summary, the contributions of this article are the following: 1) performance evaluation of standard diffusion LMS algorithms in networks with noisy input regression data; 2) development of a novel class of diffusion LMS strategies that are robust under this condition; 3) presentation of a recursive estimation approach to obtain the regression noise variances without using the second-order statistics of the data; 4) derivation of conditions under which the proposed algorithms are stable in the mean and mean-square sense; 5) characterization of their mean-square deviation (MSD) and excess mean-square error (EMSE) in transient and steady-state regimes; and 6) validation of theoretical findings through numerical simulations of newly proposed algorithms for parameter estimation over sensor networks.

The remainder of the paper is organized as follows. In Section II, we formulate the problem and discuss the effects of input measurement noise on the performance of diffusion LMS over sensor networks. In Section III, we propose bias-compensated diffusion LMS algorithms along with a recursive estimation of the regression noise variance. In Section IV, we analyze the stability and convergence behavior of the developed algorithms and obtain conditions under which the algorithms are stable in the mean and mean-square sense. We present the computer experiment results in Section VI and conclude the paper in Section VII.

Notation: Matrices are represented by uppercase fonts, vectors by lowercase fonts. Boldface letters are reserved for random variables, and normal letters are used for deterministic variables. Superscripts $(\cdot)^T$ and $(\cdot)^*$, respectively, denote transposition and conjugate transposition. Symbols $\text{Tr}(\cdot)$ and $\rho(\cdot)$ denote the trace and spectral radius of their matrix argument. The operator $\mathbb{E}[\cdot]$ stands for statistical expectation, and $\lambda_k(\cdot)$ denotes the kth eigenvalue of its matrix argument. The Kronecker product is denoted by $\otimes$, and the *block Kronecker products* [44] are denoted by $\otimes_b$. The operator $\text{diag}\{\cdot\}$ converts its argument list into a (block) diagonal matrix. The operator $\text{col}\{\cdot\}$ performs a vertical stacking of its arguments, while $\text{vec}(\cdot)$ is the standard vectorization for matrices. The symbol $\text{bvec}(\cdot)$ is the block vectorization operator that transforms a block-partitioned matrix into a column vector [44].

II. PROBLEM STATEMENT

Consider a collection of N sensor nodes distributed over a geographical area and used to monitor a physical phenomenon characterized by some unknown parameter vector $w^o \in \mathbb{C}^{M \times 1}$. As illustrated in Fig. 1, at discrete-time $i \in \mathbb{N}$, each node $k \in \{1, 2, \ldots, N\}$ collects noisy samples of the system input and
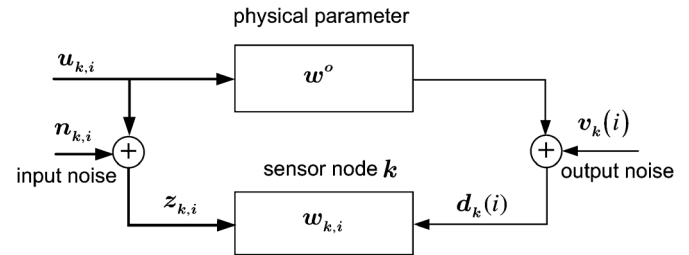

Fig. 1.   Measurement model for node k.

output denoted by $\boldsymbol{z}_{k,i} \in \mathbb{C}^{1 \times M}$ and $\boldsymbol{d}_k(i) \in \mathbb{C}$, respectively. These measurement samples can be expressed as

$$\boldsymbol{z}_{k,i} = \boldsymbol{u}_{k,i} + \boldsymbol{n}_{k,i} \tag{1}$$

$$\boldsymbol{d}_k(i) = \boldsymbol{u}_{k,i}w^o + \boldsymbol{v}_k(i) \tag{2}$$

where $\boldsymbol{u}_{k,i} \in \mathbb{C}^{1 \times M}$, $\boldsymbol{n}_{k,i} \in \mathbb{C}^{1 \times M}$, and $\boldsymbol{v}_k(i) \in \mathbb{C}$, respectively, denote the regression data vector, the input measurement noise vector, and the output measurement noise.[1]

Assumption 1: The random variables in data model (1)–(2) satisfy the following conditions.

a) The regression data vectors are independent and identically distributed (i.i.d.) over time and independent over space, with zero-mean and covariance matrix $R_{u,k} = \mathbb{E}[\boldsymbol{u}_{k,i}^*\boldsymbol{u}_{k,i}] > 0$.

b) The regression noise vectors $\boldsymbol{n}_{k,i}$ are Gaussian, i.i.d. over time, and independent over space, with zero-mean and covariance matrix $R_{n,k} = \mathbb{E}[\boldsymbol{n}_{k,i}^*\boldsymbol{n}_{k,i}] = \sigma_{n,k}^2 I$.

c) The output noise samples $\boldsymbol{v}_k(i)$ are i.i.d. over time and independent over space, with zero-mean and variance $\sigma_{v,k}^2$.

d) The random variables $\boldsymbol{u}_{k,i}$, $\boldsymbol{n}_{\ell,j}$ and $\boldsymbol{v}_p(m)$ are independent for all k, ℓ, p, i, j, and m.

The linear model (1)–(2) differs from those used in previous works on distributed estimation, such as [23], [31], and [33]. In these references, it is assumed that the actual regression vector $\boldsymbol{u}_{k,i}$ is available at each node k. There are many practical situations, however, where the nodes only have access to noisy measurements of the regression data. We use relation (1) to model such disturbance in the regressors, and to investigate the effect of the noise process $\boldsymbol{n}_{k,i}$ on the distributed estimation of w^o. To better understand the effect of this noise, we first examine the behavior of a centralized estimation solution under this condition and then explain how the resulting effect carries over to distributed approaches.

In centralized estimation, nodes transmit their measurement data $\{\boldsymbol{z}_{k,i}, \boldsymbol{d}_k(i)\}_{k=1}^N$ to a central processing unit. In the absence of measurement noise, i.e., $\boldsymbol{n}_{k,i} = 0$, the central processor can estimate the unknown parameter vector w^o by, e.g., minimizing the following mean-square error (MSE) function [45]:

$$J_u(w) = \sum_{k=1}^N \mathbb{E} \left| \boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}w \right|^2 . \tag{3}$$

Let us introduce $r_{du,k} \triangleq \mathbb{E}[\boldsymbol{d}_k(i)\boldsymbol{u}_{k,i}^*]$ and denote the sums of covariance matrices and cross-covariance vectors over the set of nodes by

$$R_u = \sum_{k=1}^N R_{u,k} \quad r_{du} = \sum_{k=1}^N r_{du,k}. \tag{4}$$

---

[1]We use parentheses to refer to the time indices of scalar variables, such as $\boldsymbol{d}_k(i)$, and subscripts to refer to time indices of vector variables, such as $\boldsymbol{z}_{k,i}$.

It can be verified that under Assumption 1, the solution of (3) is

$$w^o = R_u^{-1} r_{du}. \tag{5}$$

Now consider the recovery of the unknown parameter vector w^o for the noisy regression system described by (1) and (2). Since the regression noise $\boldsymbol{n}_{k,i}$ is independent of $\boldsymbol{u}_{k,i}$ and $\boldsymbol{d}_k(i)$, we have

$$R_{z,k-} \triangleq \mathbb{E}\left[\boldsymbol{z}_{k,i}^* \boldsymbol{z}_{k,i}\right] = R_{u,k} + \sigma_{n,k}^2 I \tag{6}$$

$$r_{dz,k} \triangleq \mathbb{E}\left[\boldsymbol{d}_k(i) \boldsymbol{z}_{k,i}^*\right] = r_{du,k}. \tag{7}$$

Considering these relations and now minimizing the global MSE function

$$J_z(w) = \sum_{k=1}^N \mathbb{E}\left|\boldsymbol{d}_k(i) - \boldsymbol{z}_{k,i} w\right|^2 \tag{8}$$

with $\boldsymbol{u}_{k,i}$ in (3) replaced by $\boldsymbol{z}_{k,i}$ in (8), we arrive at the biased solution

$$w^b = \left(R_u + \sigma_n^2 I\right)^{-1} r_{du} \tag{9}$$

where

$$\sigma_n^2 = \sum_{k=1}^N \sigma_{n,k}^2. \tag{10}$$

Let us define the bias implicit in solution (9) as $b = w^o - w^b$. To evaluate b, we may use the following identity, which holds for square matrices X_1 and X_2 provided that X_1 and $X_1 + X_2$ are both invertible [46]

$$(X_1 + X_2)^{-1} = X_1^{-1} - \left(I + X_1^{-1} X_2\right)^{-1} X_1^{-1} X_2 X_1^{-1}. \tag{11}$$

Here, R_u and $\left(R_u + \sigma_n^2 I\right)$ are invertible, and therefore we obtain

$$\left(R_u + \sigma_n^2 I\right)^{-1} = R_u^{-1} - \sigma_n^2 \left(I + \sigma_n^2 R_u^{-1}\right)^{-1} R_u^{-2}. \tag{12}$$

Considering this expression and relation (9), the bias resulting from the minimum MSE estimation at the fusion center can be expressed as

$$b = \sigma_n^2 \left(I + \sigma_n^2 R_u^{-1}\right)^{-1} R_u^{-1} w^o. \tag{13}$$

In the absence of regressor noise, it has been shown in previous studies that the parameter estimates obtained from standard diffusion LMS strategies approach the minimizer of the network global MSE function [33]. This also holds in noisy regression applications for diffusion LMS developed based on the global cost (8), meaning that the estimates generated by standard diffusion LMS algorithms will eventually approach (9). As shown by (13), this solution is biased and deviates from the optimal estimate by b. This issue will become more explicit in our convergence analysis in Section IV-A.

In the sequel, we explain how by forming a suitable objective function, the bias can be compensated in both centralized and distributed LMS implementations.

III. BIAS-COMPENSATED LMS ALGORITHMS

In our development, we initially assume that the regression noise variances, $\{\sigma_{n,k}^2\}_{k=1}^N$, are known *a priori*. We later remove this assumption by estimating these variances in real time. In networks with centralized signal processing structure, one way to obtain the unbiased optimal solution (5) is to search for a global cost function whose gradient vector is identical to that of cost (3). It is straightforward to verify that the following global cost function satisfies this requirement:

$$J(w) = \left(\sum_{k=1}^N \mathbb{E}\left|\boldsymbol{d}_k(i) - \boldsymbol{z}_{k,i} w\right|^2\right) - \left(\sum_{k=1}^N \sigma_{n,k}^2 \|w\|^2\right). \tag{14}$$

Remark 1: In bias-compensation techniques for single-node adaptive algorithms, including [12], [13], and [16], the authors first apply a least squares (LS) or minimum MSE procedure to obtain an estimate of the unknown parameter vector. The resulting estimate consists of the desired solution along with an additive bias term. The bias, which is normally expressed in terms of the second-order statistics of the regression data and the input and output measurement noises, is removed from the solution by subtraction. In the proposed technique in this paper, we start by considering bias removal one step earlier, meaning that we design a convex objective function such that its unique stationary point leads to an unbiased estimate. In this respect, our approach is mostly inspired from the derivation of the modified LMS and RLS algorithms in [8] and [15]. However, these algorithms still assume the knowledge of the ratio of input-to-output noise variances in their update equations.

The derivation of distributed algorithms will be made easier if we can decouple the network global cost function and write it as sum of local cost functions that are formed using the local data. The global cost (14) already has such a desired form. For this to become more explicit, we express (14) as

$$J(w) = \sum_{k=1}^N J_k(w) \tag{15}$$

where $J_k(w)$ is the cost function associated with node k and is given in terms of local data $\boldsymbol{d}_k(i)\,\boldsymbol{z}_{k,i}$, i.e.,

$$J_k(w) = \mathbb{E}\left|\boldsymbol{d}_k(i) - \boldsymbol{z}_{k,i} w\right|^2 - \sigma_{n,k}^2 \|w\|^2. \tag{16}$$

Remark 2: Under Assumption 1, the Hessian matrix of (16) is positive definite, i.e., $\nabla_w^2 J_k(w) > 0$, hence, $J(w)$ is strongly convex [47].

We first comment on the centralized LMS algorithm that solves (14), and then elaborate on how to develop the unbiased distributed counterparts.

A. Bias-Compensated Centralized LMS Algorithm

To minimize (15) iteratively, a centralized steepest descent algorithm [45] can be implemented as

$$\boldsymbol{w}_i = \boldsymbol{w}_{i-1} - \mu \left[\sum_{k=1}^N \nabla J_k(\boldsymbol{w}_{i-1})\right]^*. \tag{17}$$

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

4   IEEE/ACM TRANSACTIONS ON NETWORKING

where $\mu > 0$ is the step-size, and $\nabla J_k(w)$ is a row vector representing the gradient of J_k with respect to the vector w. Computing the gradient vectors from (16) leads to

$$\boldsymbol{w}_i = \boldsymbol{w}_{i-1} + \mu \sum_{k=1}^{N} \left(r_{dz,k} - R_{z,k} \boldsymbol{w}_{i-1} + \sigma_{n,k}^2 \boldsymbol{w}_{i-1} \right). \quad (18)$$

In practice, the moments $R_{z,k}$ and $r_{dz,k}$ are usually unavailable. We, therefore, replace these moments by their instantaneous approximations $\boldsymbol{z}_{k,i}^* \boldsymbol{z}_{k,i}$ and $\boldsymbol{z}_{k,i}^* \boldsymbol{d}_k(i)$, respectively, and obtain the bias-compensated centralized LMS algorithm

$$\boldsymbol{w}_i = \boldsymbol{w}_{i-1} + \mu \sum_{k=1}^{N} \left(\boldsymbol{z}_{k,i}^* \left[\boldsymbol{d}_k(i) - \boldsymbol{z}_{k,i} \boldsymbol{w}_{i-1} \right] + \sigma_{n,k}^2 \boldsymbol{w}_{i-1} \right). \quad (19)$$

In Section V, we propose an adaptive scheme to estimate the variances of the regression noise required in the above centralized LMS algorithm as well as in its distributed counterpart derived below.

B. Bias-Compensated Diffusion LMS Strategies

There exist different distributed optimization techniques that can be applied on (14) to find w^o [31], [33], [48]. We concentrate on diffusion strategies [31], [33] because they endow the network with real-time adaptation and learning abilities.

In particular, diffusion optimization strategies lead to distributed algorithms that can estimate the parameter vector w^o and track its changes over time [31], [33], [37], [49]. Here, we briefly explain how diffusion LMS algorithms can be developed for parameter estimation in systems with noisy regression data. The main step in the development of these algorithms is to reformulate the global cost (14) and represent it as a group of optimization problems of the form

$$\min_w \left\{ \sum_{\ell \in \mathcal{N}_k} c_{\ell,k} \left(\mathbb{E} \left| \boldsymbol{d}_\ell(i) - \boldsymbol{z}_{\ell,i} w \right|^2 - \sigma_{n,\ell}^2 \|w\|^2 \right) \right.$$
$$\left. + \sum_{\ell \in \mathcal{N}_k \setminus \{k\}} b_{\ell,k} \|w - w^o\|^2 \right\} \quad (20)$$

where $\mathcal{N}_k$ is the set of nodes with which node k shares information, including node k itself. The nonnegative scalars $\{c_{\ell,k}\}$ are the entries of a right-stochastic matrix $C \in \mathbb{R}^{N \times N}$ that satisfy

$$c_{\ell,k} = 0 \text{ if } \ell \notin \mathcal{N}_k, \text{ and } \sum_{k=1}^{N} c_{\ell,k} = 1. \quad (21)$$

The scalars $\{b_{\ell,k}\}$ are scaling coefficients that will end up being incorporated into the combination coefficients $\{a_{\ell,k}\}$ that appear in the final statement (23) of the algorithm below. The first term in the objective function (20) is the modified mean-squared function incorporating the noise variances of neighboring nodes $\ell \in \mathcal{N}_k$. This part of the objective is based on the same strategy as in the above centralized objective function for bias removal. The second term in (20) is in fact a constraint that forces the estimate of the node k to be aligned with the true parameter vector w^o. Since w^o is not known initially, it will be alternatively substituted by an appropriate vector during the optimization process. One can use the cost function (20) and follow similar arguments to those used in [33], [37], and [49] to arrive at the bias-compensated adapt-then-combine (ATC) LMS strategy (Algorithm 1). Due to space limitations, these steps are omitted.

Algorithm 1: ATC Bias-Compensated Diffusion LMS

$$\boldsymbol{\psi}_{k,i} = \boldsymbol{w}_{k,i-1} - \mu_k \sum_{\ell \in \mathcal{N}_k} c_{\ell,k} \left[\widehat{\nabla J_\ell}(\boldsymbol{w}_{k,i-1}) \right]^* \quad (22)$$

$$\boldsymbol{w}_{k,i} = \sum_{\ell \in \mathcal{N}_k} a_{\ell,k} \boldsymbol{\psi}_{\ell,i} \quad (23)$$

Algorithm 2: CTA Bias-Compensated Diffusion LMS

$$\boldsymbol{\psi}_{k,i-1} = \sum_{\ell \in \mathcal{N}_k} a_{\ell,k} \boldsymbol{w}_{\ell,i-1} \quad (27)$$

$$\boldsymbol{w}_{k,i} = \boldsymbol{\psi}_{k,i-1} - \mu_k \sum_{\ell \in \mathcal{N}_k} c_{\ell,k} \left[\widehat{\nabla J_\ell}(\boldsymbol{\psi}_{k,i-1}) \right]^* \quad (28)$$

In this algorithm, $\mu_k > 0$ is the step-size at node k, the vectors $\boldsymbol{\psi}_k$ and $\boldsymbol{w}_{k,i}$ are the intermediate estimates of w^o at node k, and the stochastic gradient vector is computed as

$$\left[\widehat{\nabla J_\ell}(\boldsymbol{w}_{k,i-1}) \right]^* = - \left[\boldsymbol{z}_{\ell,i}^* \left(\boldsymbol{d}_\ell(i) - \boldsymbol{z}_{\ell,i} \boldsymbol{w}_{k,i-1} \right) + \sigma_{n,\ell}^2 \boldsymbol{w}_{k,i-1} \right] \quad (24)$$

which is an instantaneous approximation to the gradient of (16). Moreover, the nonnegative coefficients $a_{\ell,k}$ are the elements of a left-stochastic matrix $A \in \mathbb{R}^{N \times N}$ satisfying

$$a_{\ell,k} = 0 \text{ if } \ell \notin \mathcal{N}_k, \text{ and } \sum_{\ell \in \mathcal{N}_k} a_{\ell,k} = 1. \quad (25)$$

To run the algorithm, we only need to select the coefficients $\{c_{\ell,k}, a_{\ell,k}\}$, which can be computed based on any combination rules that satisfy (21) and (25). One choice to compute the entries of matrix A is

$$a_{\ell,k} = \frac{\sigma_{n,\ell}^{-2}}{\sum_{\ell \in \mathcal{N}_k} \sigma_{n,\ell}^{-2}} \quad \text{and} \quad a_{k,k} = 1 - \sum_{\ell \in \mathcal{N}_k \setminus k} a_{\ell,k}. \quad (26)$$

This rule implies that the entry $a_{\ell,k}$ is inversely proportional to the regressor noise variance of node ℓ. Other left-stochastic choices for A are possible, including those that take into account both the noise variances and the degree of connectivity of the nodes [39].

By reversing the order of the adaptation and combination steps in Algorithm 1, we can obtain the following combine-then-adapt (CTA) diffusion strategy (Algorithm 2). As we will show in the analysis, the proposed ATC and CTA bias-compensated diffusion-LMS, in average, will converge to the unbiased solution (5) even when the regression data are corrupted by noise. In comparison, the estimate of the previous diffusion LMS strategies such as one proposed in [33] will be biased under such condition.

Remark 3: In the proposed ATC algorithm, each node k receives $\{\boldsymbol{u}_{\ell,i}, \boldsymbol{d}_\ell(i), \sigma_{n,\ell}^2\}$ from its neighbors in the adaptation step, and $\boldsymbol{\psi}_{\ell,i}$ in the combination step, where $\ell \in \mathcal{N}_k$. In total, it will receive $(2M + 2)|\mathcal{N}_k|$ scalar data from its neighbors. To reduce the communication overhead of the network, one solution is to choose $C = I$. Doing so, we can reduce the amount

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

ABDOLEE AND CHAMPAGNE: DIFFUSION LMS STRATEGIES IN SENSOR NETWORKS WITH NOISY INPUT DATA 5

of exchanged data at each node k to $M|\mathcal{N}_k|$ while maintaining almost similar performance results, as evidenced in Section VI. Note that the amount of information exchange in this case will be equal to that of the standard ATC diffusion LMS in [33]. This conclusion is also valid for the proposed CTA Algorithm 2.

IV. PERFORMANCE ANALYSIS

In this section, we analyze the convergence and stability of the proposed ATC and CTA bias-compensated diffusion LMS algorithms by viewing them as special cases of a more general diffusion algorithm of the form

$$\boldsymbol{\phi}_{k,i-1} = \sum_{\ell \in \mathcal{N}_k} a_{\ell,k}^{(1)} \boldsymbol{w}_{\ell,i-1} \tag{29}$$

$$\boldsymbol{\psi}_{k,i} = \boldsymbol{\phi}_{k,i-1} - \mu_k \sum_{\ell \in \mathcal{N}_k} c_{\ell,k} \left[\widehat{\nabla_\phi J_\ell}(\boldsymbol{\phi}_{k,i-1}) \right]^* \tag{30}$$

$$\boldsymbol{w}_{k,i} = \sum_{\ell \in \mathcal{N}_k} a_{\ell,k}^{(2)} \boldsymbol{\psi}_{\ell,i} \tag{31}$$

where $\{a_{\ell,k}^{(1)}\}$ and $\{a_{\ell,k}^{(2)}\}$ are nonnegative real coefficients corresponding to the (ℓ,k)th entries of left-stochastic matrices A_1 and A_2, respectively, which have the same properties as A. Different choices for A_1 and A_2 correspond to different operation modes. For instance, $A_1 = I$ and $A_2 = A$ correspond to ATC, whereas $A_1 = A$ and $A_2 = I$ generate CTA. For mathematical tractability, in our analysis, we assume that the variances of the regression noises, i.e., $\sigma_{n,k}^2$, over the network are known *a priori*.

We define the local weight-error vectors as $\tilde{\boldsymbol{w}}_{k,i} = w^o - \boldsymbol{w}_{k,i}$, $\tilde{\boldsymbol{\psi}}_{k,i} = w^o - \boldsymbol{\psi}_{k,i}$, and $\tilde{\boldsymbol{\phi}}_{k,i} = w^o - \boldsymbol{\phi}_{k,i}$ and form the global weight-error vectors by stacking the local error vectors, i.e.,

$$\tilde{\boldsymbol{\phi}}_i = \text{col}\{\tilde{\boldsymbol{\phi}}_{1,i}, \tilde{\boldsymbol{\phi}}_{2,i}, \dots, \tilde{\boldsymbol{\phi}}_{N,i}\} \tag{32}$$

$$\tilde{\boldsymbol{\psi}}_i = \text{col}\{\tilde{\boldsymbol{\psi}}_{1,i}, \tilde{\boldsymbol{\psi}}_{2,i}, \dots, \tilde{\boldsymbol{\psi}}_{N,i}\} \tag{33}$$

$$\tilde{\boldsymbol{w}}_i = \text{col}\{\tilde{\boldsymbol{w}}_{1,i}, \tilde{\boldsymbol{w}}_{2,i}, \dots, \tilde{\boldsymbol{w}}_{N,i}\}. \tag{34}$$

We also define the block variables

$$\boldsymbol{g}_i = \mathcal{C}^T \text{col}\left\{ \boldsymbol{z}_{\ell,i}^* \boldsymbol{v}_1(i), \dots, \boldsymbol{z}_{N,i}^* \boldsymbol{v}_N(i) \right\} \tag{35}$$

$$\boldsymbol{\mathcal{R}}_i = \text{diag}\left\{ \sum_{\ell \in \mathcal{N}_k} c_{\ell,k} \left(\boldsymbol{z}_{\ell,i}^* \boldsymbol{z}_{\ell,i} - \sigma_{n,\ell}^2 I \right), k = 1, \dots, N \right\} \tag{36}$$

$$\boldsymbol{\mathcal{P}}_i = \text{diag}\left\{ \sum_{\ell \in \mathcal{N}_k} c_{\ell,k} \left(\boldsymbol{z}_{\ell,i}^* \boldsymbol{n}_{\ell,i} - \sigma_{n,\ell}^2 I \right), k = 1, \dots, N \right\} \tag{37}$$

$$\mathcal{M} = \text{diag}\left\{ \mu_1 I_M, \dots, \mu_N I_M \right\} \tag{38}$$

and introduce the following extended combination matrices:

$$\mathcal{A}_1 = A_1 \otimes I_M \quad \mathcal{A}_2 = A_2 \otimes I_M \quad \mathcal{C} = C \otimes I_M. \tag{39}$$

Using these definitions and update (29)–(31), it can be verified that the following relations hold:

$$\tilde{\boldsymbol{\phi}}_{i-1} = \mathcal{A}_1^T \tilde{\boldsymbol{w}}_{i-1}$$
$$\tilde{\boldsymbol{\psi}}_i = \tilde{\boldsymbol{\phi}}_{i-1} - \mathcal{M} \left(\boldsymbol{g}_i - \boldsymbol{\mathcal{P}}_i \omega^o + \boldsymbol{\mathcal{R}}_i \tilde{\boldsymbol{\phi}}_{i-1} \right)$$
$$\tilde{\boldsymbol{w}}_i = \mathcal{A}_2^T \tilde{\boldsymbol{\psi}}_i \tag{40}$$

where $\omega^o = \mathbb{1} \otimes w^o$. From the set of equations given in (40), it is deduced that the network error vector $\tilde{\boldsymbol{w}}_i$ evolves with time according to the recursion

$$\tilde{\boldsymbol{w}}_i = \boldsymbol{\mathcal{B}}_i \tilde{\boldsymbol{w}}_{i-1} - \mathcal{A}_2^T \mathcal{M} \boldsymbol{g}_i + \mathcal{A}_2^T \mathcal{M} \boldsymbol{\mathcal{P}}_i \omega^o \tag{41}$$

where the time-varying matrix $\boldsymbol{\mathcal{B}}_i$ is defined as

$$\boldsymbol{\mathcal{B}}_i = \mathcal{A}_2^T (I - \mathcal{M} \boldsymbol{\mathcal{R}}_i) \mathcal{A}_1^T. \tag{42}$$

A. Mean Convergence and Stability

Tacking the expectation of both sides of (41) and considering Assumption 1, we arrive at

$$\mathbb{E}[\tilde{\boldsymbol{w}}_i] = \mathcal{B} \left(\mathbb{E}[\boldsymbol{w}_{i-1}] \right) \tag{43}$$

where in this relation

$$\mathcal{B} \triangleq \mathbb{E}[\boldsymbol{\mathcal{B}}_i] = \mathcal{A}_2^T (I - \mathcal{M}\mathcal{R}) \mathcal{A}_1^T \tag{44}$$

$$\mathcal{R} \triangleq \mathbb{E}[\boldsymbol{\mathcal{R}}_i] = \text{diag}\left\{ \sum_{\ell \in \mathcal{N}_k} c_{\ell,k} R_{u,\ell}, k = 1, \dots, N \right\}. \tag{45}$$

To obtain (43), we used the fact that $\mathbb{E}[\mathcal{A}_2^T \mathcal{M} \boldsymbol{g}_i] = 0$ because $\boldsymbol{v}_{k,i}$ is independent of $\boldsymbol{z}_{k,i}$ and $\mathbb{E}[\boldsymbol{v}_k(i)] = 0$. Moreover, we have $\mathbb{E}[\boldsymbol{\mathcal{P}}_i] = 0$ because $\mathbb{E}[\boldsymbol{z}_{\ell,i}^* \boldsymbol{n}_{\ell,i}] = \sigma_{n,\ell}^2 I$. According to (43), $\lim_{i \to \infty} \mathbb{E}\|\tilde{\boldsymbol{w}}_i\| \to 0$ if $\mathcal{B}$ is stable (i.e., when $\rho(\mathcal{B}) < 1$). In fact, because $\rho(\mathcal{A}_1) = \rho(\mathcal{A}_2) = 1$ and $\mathcal{R} > 0$ choosing the step-sizes according to

$$0 < \mu_k < \frac{2}{\rho \left(\sum_{\ell \in \mathcal{N}_k} c_{\ell,k} R_{u,\ell} \right)} \tag{46}$$

guarantees $\rho(\mathcal{B}) < 1$. We omit the proof, but a similar argument can be found in [49] and [35]. We summarize the mean-convergence results of the proposed bias-compensated diffusion LMS in the following.

Theorem 1: Consider an adaptive network that operates using diffusion Algorithms 1 or 2 with the space-time data (1) and (2). In this network, if we assume that the regressors noise variances are known or perfectly estimated, the mean error vector evolves with time according to (43). Furthermore, Algorithms 1 and 2 will be asymptotically unbiased and stable provided that the step-sizes satisfy (46).

Remark 4: In networks with noisy regression data (1), the estimates generated by the previous diffusion LMS strategies such as the ones proposed in [33] and [49] are biased, i.e., $\mathbb{E}[\tilde{\boldsymbol{w}}_i] \neq 0$ as $i \to \infty$. This can be readily shown if we remove $\sigma_{n,k}^2$ from (36) and (37). In this scenario, (43) will be stable if

$$0 < \mu_k < \frac{2}{\rho \left(\sum_{\ell \in \mathcal{N}_k} c_{\ell,k} \left(R_{u,\ell} + \sigma_{n,\ell}^2 I_M \right) \right)}. \tag{47}$$

Then, for sufficiently small step-sizes, satisfying (47), it can be verified that the estimate of the standard diffusion LMS deviates from the network optimal solution ω^o by

$$\lim_{i \to \infty} \mathbb{E}[\tilde{\boldsymbol{w}}_i] = (I_{NM} - \mathcal{B}')^{-1} \mathcal{A}_2^T \mathcal{M} \mathcal{P}' \omega^o \tag{48}$$

where

$$\mathcal{B}' \triangleq \mathcal{A}_2^T (I_{NM} - \mathcal{M}\mathcal{R}') \mathcal{A}_1^T \tag{49}$$

$$\mathcal{R}' \triangleq \text{diag}\left\{ \sum_{\ell \in \mathcal{N}_k} c_{\ell,k} \left(R_{u,\ell} + \sigma_{n,\ell}^2 I_M \right), k = 1, \dots, N \right\} \tag{50}$$

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

6

IEEE/ACM TRANSACTIONS ON NETWORKING

$$\mathcal{P}' \triangleq \text{diag} \left\{ \sum_{\ell \in \mathcal{N}_k} c_{\ell,k} \sigma_{n,\ell}^2 I_M, k = 1, \ldots, N \right\}. \qquad (51)$$

As it is clear from (48), the bias is created by the regression noise $\{\boldsymbol{n}_{k,i}\}$ only, whereas the noise $\{\boldsymbol{v}_k(i)\}$ has no effect on generating the bias.

B. Mean-Square Convergence and Stability

To study the mean-square performance of the proposed algorithms, we first follow the energy conservation arguments of [33] and [45] and determine a variance relation that is suitable in the current context. The relation can be obtained in the limit, as $i \to \infty$, by computing the expectation of the weighted squared norm of (41) under Assumption 1, i.e.,

$$\mathbb{E}\|\tilde{\boldsymbol{w}}_i\|_{\Sigma}^2 = \mathbb{E}\left(\|\tilde{\boldsymbol{w}}_{i-1}\|_{\Sigma'}^2\right) + \mathbb{E}\left[\boldsymbol{g}_i^* \mathcal{M}\mathcal{A}_2 \Sigma \mathcal{A}_2^T \mathcal{M}\boldsymbol{g}_i\right] \\ + \mathbb{E}\left[\omega^{o*}\boldsymbol{\mathcal{P}}_i^* \mathcal{M}\mathcal{A}_2 \Sigma \mathcal{A}_2^T \mathcal{M}\boldsymbol{\mathcal{P}}_i\omega^o\right] \qquad (52)$$

where $\|x\|_{\Sigma}^2 = x^* \Sigma x$ and $\Sigma \geq 0$ is a weighting matrix that we are free to choose. Note that (52) is obtained by eliminating the following terms:

$$\mathbb{E}\left[(\mathcal{A}_2^T \mathcal{M}\boldsymbol{g}_i)^* \Sigma \mathcal{A}_2^T (I - \mathcal{M}\boldsymbol{\mathcal{R}}_i)\mathcal{A}_1^T \tilde{\boldsymbol{w}}_{i-1}\right] = 0 \qquad (53)$$

$$\mathbb{E}\left[(\mathcal{A}_2^T \left(I - \mathcal{M}\boldsymbol{\mathcal{R}}_i\right)\mathcal{A}_1^T \tilde{\boldsymbol{w}}_{i-1})^* \Sigma \mathcal{A}_2^T \mathcal{M}\boldsymbol{g}_i\right] = 0 \qquad (54)$$

$$\mathbb{E}\left[(\mathcal{A}_2^T \mathcal{M}\boldsymbol{\mathcal{P}}_i w^o)^* \Sigma \mathcal{A}_2^T (I - \mathcal{M}\boldsymbol{\mathcal{R}}_i)\mathcal{A}_1^T \tilde{\boldsymbol{w}}_{i-1}\right] = 0 \qquad (55)$$

$$\mathbb{E}\left[(\mathcal{A}_2^T (I - \mathcal{M}\boldsymbol{\mathcal{R}}_i)\mathcal{A}_1^T \tilde{\boldsymbol{w}}_{i-1})^* \Sigma \mathcal{A}_2^T \mathcal{M}\boldsymbol{\mathcal{P}}_i w^o\right] = 0. \qquad (56)$$

These terms are zero first because $\tilde{\boldsymbol{w}}_{i-1}$ is independent of $\boldsymbol{g}_i$, $\boldsymbol{\mathcal{P}}_i$, and $\boldsymbol{\mathcal{R}}_i$ under Assumption 1 [50], and second, since the proposed algorithms are unbiased, $\mathbb{E}[\tilde{\boldsymbol{w}}_i]$ is zero for large i if the step-sizes are chosen as in (46).

In relation (52), we have

$$\boldsymbol{\Sigma}' = \boldsymbol{\mathcal{B}}_i^* \Sigma \boldsymbol{\mathcal{B}}_i. \qquad (57)$$

It follows from Assumption 1 that $\tilde{\boldsymbol{w}}_{i-1}$ and $\boldsymbol{\mathcal{R}}_i$ are independent of each other so that

$$\mathbb{E}\left(\|\tilde{\boldsymbol{w}}_{i-1}\|_{\Sigma'}^2\right) = \mathbb{E}\|\tilde{\boldsymbol{w}}_{i-1}\|_{\mathbb{E}[\Sigma']}^2. \qquad (58)$$

Substituting this expression into (52), we arrive at

$$\mathbb{E}\|\tilde{\boldsymbol{w}}_i\|_{\Sigma}^2 = \mathbb{E}\|\tilde{\boldsymbol{w}}_{i-1}\|_{\Sigma'}^2 + \text{Tr}[\Sigma \mathcal{A}_2^T \mathcal{M}\mathcal{G}\mathcal{M}\mathcal{A}_2] \\ + \text{Tr}\left[\Sigma \mathcal{A}_2^T \mathcal{M}\Pi\mathcal{M}\mathcal{A}_2\right] \qquad (59)$$

where

$$\Sigma' = \mathbb{E}\left[\boldsymbol{\mathcal{B}}_i^* \Sigma \boldsymbol{\mathcal{B}}_i\right]. \qquad (60)$$

In (59), $\mathcal{G} = \mathbb{E}[\boldsymbol{g}_i \boldsymbol{g}_i^*]$, which using (35) is given by (see Appendix A)

$$\mathcal{G} = \mathcal{C}^T \text{diag}\left\{\sigma_{v,1}^2 \left(R_{u,1} + \sigma_{n,1}^2 I\right),\right. \\ \left. \ldots, \sigma_{v,N}^2 \left(R_{u,N} + \sigma_{n,N}^2 I\right)\right\} \mathcal{C}. \qquad (61)$$

In relation (59), $\Pi = \mathbb{E}[\boldsymbol{\mathcal{P}}_i \omega^o \omega^{o*} \boldsymbol{\mathcal{P}}_i^*]$, and its (k,j)th block is computed as (see Appendix B)

$$\Pi_{k,j} = \sum_{\ell} c_{\ell,k} c_{\ell,j} \left\{\sigma_{n,\ell}^2 \|w^o\|^2 \left(R_{u,\ell} + \sigma_{n,\ell}^2 I\right)\right. \\ \left. + (\beta - 1)\sigma_{n,\ell}^4 w^o w^{o*}\right\} \qquad (62)$$

where $\beta = 2$ for real-valued data and $\beta = 1$ for complex-valued data. If we introduce $\sigma = \text{bvec}(\Sigma)$ and $\sigma' = \text{bvec}(\Sigma')$, then we can write $\sigma' = \mathcal{F}\sigma$ where

$$\mathcal{F} = \mathbb{E}\left[\boldsymbol{\mathcal{B}}_i^T \otimes_b \boldsymbol{\mathcal{B}}_i^*\right]. \qquad (63)$$

Considering these definitions, the variance relation in (59) can be rewritten more compactly as

$$\mathbb{E}\|\tilde{\boldsymbol{w}}_i\|_{\sigma}^2 = \mathbb{E}\|\tilde{\boldsymbol{w}}_{i-1}\|_{\mathcal{F}\sigma}^2 + \gamma^T \sigma \qquad (64)$$

where we are using the notation $\|x\|_{\sigma}^2$ as a short form for $\|x\|_{\Sigma}^2$, and where

$$\gamma = \text{bvec}\left(\mathcal{A}_2^T \mathcal{M}\mathcal{G}^T \mathcal{M}\mathcal{A}_2 + \mathcal{A}_2^T \mathcal{M}\Pi^T \mathcal{M}\mathcal{A}_2\right). \qquad (65)$$

To compute $\mathcal{F}$, we expand Σ' from (60) to get

$$\Sigma' = \mathcal{A}_1 \left(\mathcal{A}_2 \Sigma \mathcal{A}_2^T - \mathcal{R}\mathcal{M}\mathcal{A}_2 \Sigma \mathcal{A}_2^T - \mathcal{A}_2 \Sigma \mathcal{A}_2^T \mathcal{M}\mathcal{R}\right) \mathcal{A}_1^T \\ + \mathbb{E}\left[\mathcal{A}_1 \boldsymbol{\mathcal{R}}_i^* \mathcal{M}\mathcal{A}_2 \Sigma \mathcal{A}_2^T \mathcal{M}\boldsymbol{\mathcal{R}}_i \mathcal{A}_1^T\right]. \qquad (66)$$

The last term in (66) depends on $\mathcal{M}^2$ and can, therefore, be neglected for small step-sizes. As a result, we obtain

$$\mathcal{F} \approx (\mathcal{A}_1 \otimes_b \mathcal{A}_1)(I - I \otimes_b \mathcal{R}\mathcal{M} - \mathcal{R}^T \mathcal{M} \otimes_b I)(\mathcal{A}_2 \otimes_b \mathcal{A}_2). \qquad (67)$$

We can also derive a more compact expression to compute $\mathcal{F}$. To this end, we first note that the last term in (66) can be expressed as

$$\mathbb{E}\left[\mathcal{A}_1 \boldsymbol{\mathcal{R}}_i^* \mathcal{M}\mathcal{A}_2 \Sigma \mathcal{A}_2^T \mathcal{M}\boldsymbol{\mathcal{R}}_i \mathcal{A}_1^T\right] \\ = \mathbb{E}\left[\mathcal{A}_1 \mathcal{R}^* \mathcal{M}\mathcal{A}_2 \Sigma \mathcal{A}_2^T \mathcal{M}\mathcal{R}\mathcal{A}_1^T\right] + O(\mathcal{M}^2) \qquad (68)$$

Now by substituting (68) into (66) and ignoring the remaining terms that depend on $\mathcal{M}^2$, under the small step-size condition, we arrive at

$$\mathcal{F} \approx \mathcal{B}^T \otimes_b \mathcal{B}^*. \qquad (69)$$

We now proceed to show the stability of the algorithm in the mean-square error sense, as follows. Using (64), we can write

$$\lim_{i \to \infty} \mathbb{E}\|\tilde{\boldsymbol{w}}_i\|_{\sigma}^2 = \lim_{i \to \infty} \mathbb{E}\|\tilde{\boldsymbol{w}}_{-1}\|_{\mathcal{F}^{i+1}\sigma}^2 + \gamma^T \sum_{j=0}^{\infty} \mathcal{F}^j \sigma. \qquad (70)$$

As it is evident from this expression, the proposed algorithms will be stable in the mean-square sense if $\mathcal{F}$ is stable. From (69), we deduce that $\mathcal{F}$ will be stable if $\mathcal{B}$ is stable. According to our mean-convergence analysis, the stability of $\mathcal{B}$ is guaranteed if (46) holds. Therefore, the step-size condition (46) is sufficient to guarantee the stability of the algorithms both in the mean and mean-square sense.

C. Mean-Square Steady-State Performance

To obtain MSE steady-state expressions for the network, we let i go to infinity and use expression (64) to write

$$\lim_{i \to \infty} \mathbb{E}\|\tilde{\boldsymbol{w}}_i\|_{(I-\mathcal{F})\sigma}^2 = \gamma^T \sigma. \qquad (71)$$

By definition, the MSD and EMSE at each node k are respectively computed as

$$\eta_k = \lim_{i \to \infty} \mathbb{E}\|\tilde{\boldsymbol{w}}_{k,i}\|^2 \quad \zeta_k = \lim_{i \to \infty} \mathbb{E}\|\tilde{\boldsymbol{w}}_{k,i}\|_{R_{u,k}}^2. \qquad (72)$$

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

ABDOLEE AND CHAMPAGNE: DIFFUSION LMS STRATEGIES IN SENSOR NETWORKS WITH NOISY INPUT DATA 7

The MSD and EMSE of the nodes can be retrieved from the network error vector $\tilde{\boldsymbol{w}}_i$ by writing

$$\eta_k = \lim_{i \to \infty} \mathbb{E}\|\tilde{\boldsymbol{w}}_i\|^2_{\{diag(e_k) \otimes I\}} \qquad (73)$$

$$\zeta_k = \lim_{i \to \infty} \mathbb{E}\|\tilde{\boldsymbol{w}}_i\|^2_{\{diag(e_k) \otimes R_{u,k}\}} \qquad (74)$$

where e_k is a canonical basis vector in $\mathbb{R}^N$ with entry one at position k. From (71) and (73), we can obtain the MSD at node k, for $k \in \{1, 2, \ldots, N\}$, as

$$\eta_k = \gamma^T (I - \mathcal{F})^{-1} \text{bvec}\left(\text{diag}(e_k) \otimes I_M\right). \qquad (75)$$

In the same manner, we compute the EMSE at node k as

$$\zeta_k = \gamma^T (I - \mathcal{F})^{-1} \text{bvec}\left(\text{diag}(e_k) \otimes R_{u,k}\right). \qquad (76)$$

The network MSD and EMSE are defined as the average of MSD and EMSE values over the network, i.e.,

$$\eta = \frac{1}{N}\sum_{k=1}^{N} \eta_k, \quad \zeta = \frac{1}{N}\sum_{k=1}^{N} \zeta_k. \qquad (77)$$

D. Mean-Square Transient Behavior

We use (64) to obtain an expression for the mean-square behavior of the algorithm in transient-state. In this expression, if we substitute $\boldsymbol{w}_{k,-1} = 0, \forall k \in \{1, \ldots, N\}$, we obtain

$$\|\tilde{\boldsymbol{w}}_i\|^2_\sigma = \|w^o\|^2_{\mathcal{F}^{i+1}\sigma} + \gamma^T \sum_{j=0}^{i} \mathcal{F}^j \sigma. \qquad (78)$$

Writing this recursion for $i-1$ and subtracting it from (78) leads to

$$\|\tilde{\boldsymbol{w}}_i\|^2_\sigma = \|\tilde{\boldsymbol{w}}_{i-1}\|^2_\sigma + \|w^o\|^2_{\mathcal{F}^i(I-\mathcal{F})\sigma} + \gamma^T \mathcal{F}^i \sigma. \qquad (79)$$

By replacing σ with $\sigma_{\text{msd}_k} = \text{bvec}(\text{diag}\{e_k\} \otimes I_M)$ and $\sigma_{\text{emse}_k} = \text{bvec}(\text{diag}\{e_k\} \otimes R_{u,k})$ and using $\boldsymbol{w}_{k,-1} = 0$, we arrive at the following two recursions for the evolution of MSD and EMSE over time:

$$\eta_k(i) = \eta_k(i-1) - \|w^o\|_{\mathcal{F}^i(I-\mathcal{F})\sigma_{\text{msd}_k}} + \gamma^T \mathcal{F}^i \sigma_{\text{msd}_k} \qquad (80)$$

$$\zeta_k(i) = \zeta_k(i-1) - \|w^o\|_{\mathcal{F}^{i-1}(I-\mathcal{F})\sigma_{\text{emse}_k}} + \gamma^T \mathcal{F}^{i-1} \sigma_{\text{emse}_k}. \qquad (81)$$

The MSD and EMSE of the network can be computed either by averaging the nodes transient behavior, or by substituting

$$\sigma_{\text{msd}} = \frac{1}{N} \text{bvec}(I_{MN}) \qquad (82)$$

$$\sigma_{\text{emse}} = \frac{1}{N} \text{bvec}\left(\text{diag}\{R_{u,1}, \ldots, R_{u,N}\}\right) \qquad (83)$$

in recursion (79). We summarize the mean-square analysis results of the algorithms in the following.

Theorem 2: Consider an adaptive network operating under bias-compensated diffusion Algorithm 1 or 2 with the space-time data (1) and (2) that satisfy Assumption 1. In this network, if we assume that the regressors noise variances are known or

perfectly estimated and nodes initialized at zero, then the MSD and EMSE of each node k evolve with time according to (80) and (81) and the network MSD and EMSE follow recursions

$$\eta(i) = \eta(i-1) - \|w^o\|_{\mathcal{F}^i(I-\mathcal{F})\sigma_{\text{msd}}} + \gamma^T \mathcal{F}^i \sigma_{\text{msd}}$$

$$\zeta(i) = \zeta(i-1) - \|w^o\|_{\mathcal{F}^{i-1}(I-\mathcal{F})\sigma_{\text{emse}}} + \gamma^T \mathcal{F}^{i-1} \sigma_{\text{emse}}$$

where σ_{msd} and σ_{emse} are defined in (82) and (83) and $\mathcal{F}$ is given by (63). Moreover, if the step-sizes are chosen to satisfy (46), the network will be stable, converge in the mean and mean-square sense, and reach the steady-state MSD and EMSE characterized by (77).

V. REGRESSION NOISE VARIANCE ESTIMATION

In the proposed algorithms, each node still needs to have the regression noise variances, $\{\sigma^2_{n,\ell}\}_{k=1}^{\mathcal{N}_k}$, to evaluate the stochastic gradient vector, $\widehat{\nabla J_\ell}$. In practice, such information is rarely available and normally obtained through estimation. A review of previous works reveals that the regression noise variances can be either estimated offline [43] or in real time when the unknown parameter vector, w^o, is being estimated [51], [52]. For example, in the context of speech analysis, they can be estimated offline during silent periods in between words and sentences [43]. In some other applications, these variances are estimated during the operation of the algorithm using the second-order moments of the regression data and the system output signal [51], [52]. In what follows, we propose an adaptive recursive approach to estimate the regression noise variances without using the second-order moments of the data.

The variance of the regression noise at each node is classified as local information and, hence, it can be estimated from the node's local data. When the regression data at node k is not corrupted by measurement noise (i.e., $\boldsymbol{z}_{k,i} = \boldsymbol{u}_{k,i}$), and when the node operates independently of all other nodes to estimate w^o by minimizing $\mathbb{E}|\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}w|^2$, the minimum attainable MSE can be expressed as [45]

$$J_{\min} \triangleq \sigma^2_{d,k} - r^*_{du,k} R^{-1}_{u,k} r_{du,k}. \qquad (84)$$

Under noisy regression scenarios where node k operates independently to minimize the cost (16), the minimum achievable cost will still be (84). To verify this, we note from Remark 2 that since $J_k(w)$ is positive definite and, hence, strongly convex, its unique minimizer under Assumption 1 will be w^o. Therefore, substituting w^o into (16) will give its minimum, i.e.,

$$\min_w J_k(w) = \mathbb{E}|\boldsymbol{d}_k(i) - \boldsymbol{z}_{k,i}w^o|^2 - \sigma^2_{n,k}\|w^o\|^2$$

$$= \sigma^2_{d,k} - r^*_{du,k} R^{-1}_{u,k} r_{du,k} \qquad (85)$$

$$= J_{\min}. \qquad (86)$$

We use this result to estimate the regression noise variance $\sigma^2_{n,k}$ at each node k.

Now, let us introduce

$$\boldsymbol{e}_k(i) \triangleq \boldsymbol{d}_k(i) - \boldsymbol{z}_{k,i}\boldsymbol{w}_{k,i-1} \qquad (87)$$

where $\boldsymbol{w}_{k,i-1}$ is the weight estimate from ATC diffusion (which would be replaced by $\boldsymbol{\psi}_{k,i-1}$ for CTA diffusion). Considering

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

8   IEEE/ACM TRANSACTIONS ON NETWORKING

$J_k(\boldsymbol{w}_{k,i-1})$, for sufficiently small step-sizes and in the limit when the weight estimate is close enough to w^o, it holds that

$$\mathbb{E}\,|\boldsymbol{e}_k(i)|^2 - \sigma_{n,k}^2\,\|w^o\|^2 \approx J_{\min}. \qquad (88)$$

From (2) and (84), it can be verified that $J_{\min} = \sigma_{v,k}^2$, and hence from (88), we can write

$$\mathbb{E}\,|\boldsymbol{e}_k(i)|^2 \approx \sigma_{v,k}^2 + \sigma_{n,k}^2\,\|w^o\|^2. \qquad (89)$$

In this relation, $\sigma_{v,k}^2$ can be ignored if $\sigma_{n,k}^2\|w^o\|^2 \gg \sigma_{v,k}^2$. Under such circumstances, if we assume $\|w^o\|^2 \neq 0$, which is true for systems with at least one nonzero coefficient, then the variance of the regression noise can be obtained by

$$\sigma_{n,k}^2 \approx \frac{\mathbb{E}\,|\boldsymbol{e}_k(i)|^2}{\|w^o\|^2}. \qquad (90)$$

Since, in (90), $\mathbb{E}|\boldsymbol{e}_k(i)|^2$ and the unknown parameter, w^o, are initially unavailable, we can estimate $\sigma_{n,k}^2$ using the following relations as the latest estimates of these quantities become available, i.e.:

$$\boldsymbol{f}_k(i) = \alpha \boldsymbol{f}_k(i-1) + (1-\alpha)|\boldsymbol{e}_k(i)|^2 \qquad (91)$$

$$\boldsymbol{\sigma}_{n,k}^2(i) = \frac{\boldsymbol{f}_k(i)}{\|\boldsymbol{w}_{k,i}\|^2} \qquad (92)$$

where $0 \ll \alpha < 1$ is a smoothing factor with nominal values in the range of $[0.95, 0.99]$.

Assumption 2: The regression noise variance, $\sigma_{n,k}^2$, and the output measurement noise, $\sigma_{v,k}^2$, satisfy the following inequality:

$$\sigma_{n,k}^2\,\|w^o\|^2 \gg \sigma_{v,k}^2. \qquad (93)$$

Under this assumption, the regressor noise variance at each node k can be adaptively estimated via (91) and (92) using the data samples $\boldsymbol{e}_k(i)$ and $\boldsymbol{w}_{k,i-1}$ supplied from the bias-compensated LMS iterations.

VI. SIMULATION RESULTS

In this section, we present computer experiments to illustrate the efficiency of the proposed algorithms and to verify the theoretical findings. We evaluate the algorithm performance for known regressor noise variance and with adaptive noise variance estimation. We consider a connected network with $N = 20$ nodes that are positioned randomly on a unit square area with maximum communication distance of 0.4 unit length. The network topology is shown in Fig. 2. We choose $A_1 = I$, compute A_2 using the relative-variance rule (26), and choose the matrix C according to the metropolis criterion [31], [49]. In the plots, we use A_{rel} and C_{met} to refer to this particular choice of A_2 and C. The network data are generated according to model (1) and (2). The aim is to estimate the system parameter vector $w^o = [1,1]^T/\sqrt{2}$ over the network using the proposed bias-compensated diffusion algorithms. In all our experiments, the curves from the simulation results are drawn from the average of 500 independent runs.

We choose the step-sizes as $\mu_k = 0.05$ and set $\boldsymbol{w}_{k,-1} = [0,0]^T$, for all k. We adopt Gaussian distribution to generate $\boldsymbol{v}_k(i)$, $\boldsymbol{n}_{k,i}$, and $\boldsymbol{u}_{k,i}$. The covariance matrices of the regression data and the regression noise are of the form $R_{u,k} = \sigma_{u,k}^2 I_M$ and $\sigma_{n,k}^2 I_M$, respectively. The network signal and noise power profile are given in Table I.

1) Transient MSE Results With Perfect Noise Variance Estimation: In Fig. 3, we demonstrate the network transient behavior in terms of MSD and EMSE for the proposed diffusion
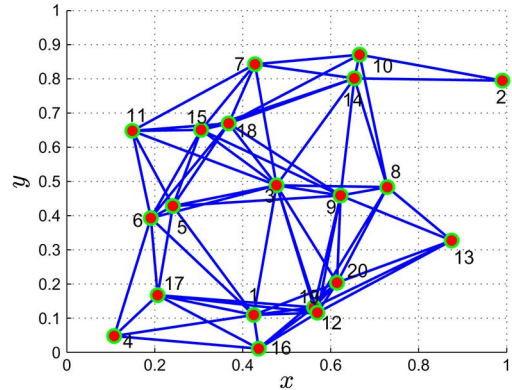

Fig. 2. Network topology used in the simulations.

TABLE I
NETWORK SIGNAL AND NOISE POWER PROFILE

Node k	Parameters		
	$\sigma_{v,k}^2$	$\mathrm{Tr}(R_{u,k})$	$\sigma_{n,k}^2$
1	0.0230	0.3000	0.0170
2	0.0020	0.7500	0.0970
3	0.0160	0.5250	0.0620
4	0.0040	0.4250	0.0570
5	0.0420	0.6000	0.0600
6	0.0400	0.6500	0.0730
7	0.0120	1.0000	0.0560
8	0.0120	0.7750	0.0860
9	0.0310	0.7250	0.0250
10	0.0280	0.6750	0.0490
11	0.0350	0.6500	0.0680
12	0.0500	0.6000	0.0760
13	0.0090	0.2750	0.0600
14	0.0340	0.3500	0.0150
15	0.0290	0.6250	0.0160
16	0.0280	0.9250	0.0490
17	0.0020	0.3250	0.0830
18	0.0080	0.8750	0.0370
19	0.0410	0.2500	0.0170
20	0.0460	0.8000	0.0160

LMS algorithm, standard diffusion LMS algorithm [33], and the noncooperative mode of the proposed algorithm. Note that $A_2 = I$ and $C = I$ correspond to the noncooperative network mode of the proposed algorithm, where each node runs a stand-alone bias-compensated LMS. As the results indicate, the performance of the cooperative network with C_{met} and A_{rel} exceeds that of the noncooperative case by 12 dB. We also observe that the proposed algorithm outperforms the standard diffusion LMS [33] by more than 12 dB. It is interesting to note that the noncooperative algorithm outperforms the standard diffusion LMS by about 1 dB.

We also present the EMSE and MSD of some randomly chosen nodes in Fig. 4. In particular, we plot the EMSE learning curves of nodes 4 and 18 and the MSD learning curves of nodes 5 and 15. We observe that the MSD curves of the chosen nodes are identical. Since the algorithm is unbiased, this implies that these nodes have reached agreement about the unknown network parameter, w^o. As we will show in the steady-state results, all nodes over the network almost reach agreement. We note that, in all scenarios, there is a good agreement between simulations and the analysis results.

2) Steady-State MSE Results With Perfect Noise Variance Estimation: The network steady-state MSD and EMSE are shown

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

ABDOLEE AND CHAMPAGNE: DIFFUSION LMS STRATEGIES IN SENSOR NETWORKS WITH NOISY INPUT DATA 9
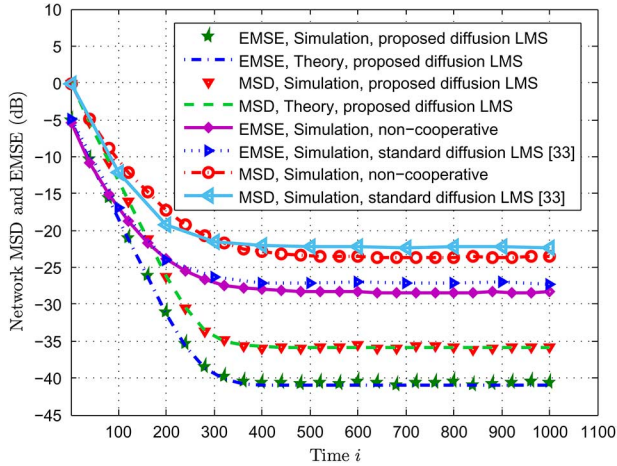

Fig. 3. Convergence behavior of the proposed bias-compensated diffusion LMS, standard diffusion LMS, and noncooperative LMS algorithms.

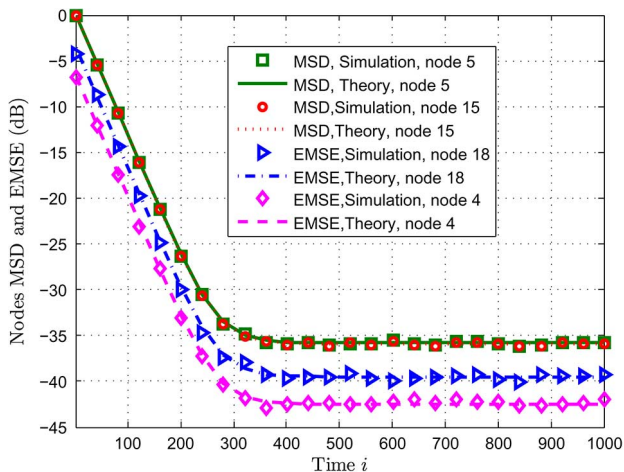Fig. 4. MSD learning curves of nodes 5 and 15 and EMSE learning curves of nodes 4 and 18.

Fig. 5. Network steady-state MSD for different combination matrices.
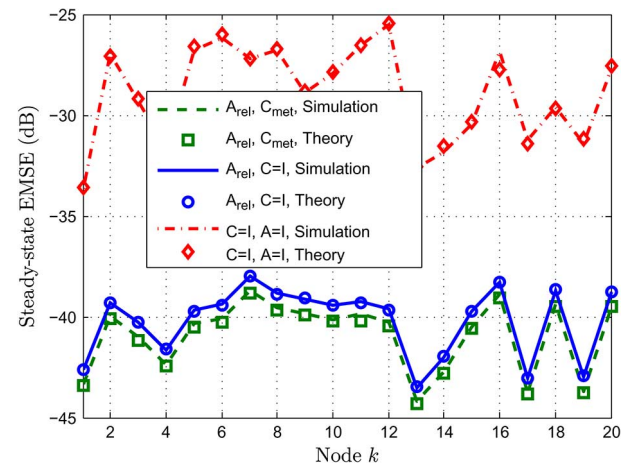
Fig. 6. Network steady-state EMSE for different combination matrices.

in Figs. 5 and 6. From these figures, we observe that there is a good agreement between simulations and analytical findings. In addition, we consider the case when nodes only exchange their intermediate estimates (i.e., when $C = I$). It is seen that the MSD performance of the algorithm with C_{met} is 1 dB superior than that with $C = I$. We also observe that the performance discrepancies between nodes in terms of MSD are less than 0.5 dB for cooperative scenarios, while in the noncooperative scenario it is more than 5 dB. This shows agreement in the network in spite of different noise and energy profiles at each node. Note that the fluctuations in EMSE over the network are due to differences in energy level in the nodes' input signals, but this does not preclude the cooperating nodes from reaching a consensus in the estimated parameters.

3) MSE Results of the Algorithm With Adaptive Noise Variance Estimation: We compare the transient and steady-state behavior of the bias-compensated diffusion LMS with known regressor noise variance and adaptive noise variance estimation. For this experiment, we consider the same network topology and noise profile as above. However, the unknown parameter vector to be estimated, in this case, is $w^o = 2\mathbb{1}_5 + 2j\mathbb{1}_5$, where $\mathbb{1}_M$
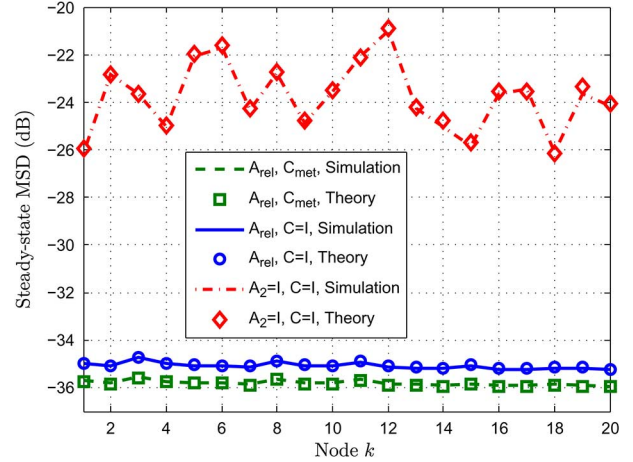
is an $M \times 1$ column vector with unit entries. The network energy profile is chosen as $\mathrm{Tr}(R_{u,k}) = 20\,\mathrm{Tr}(\sigma_{n,k}^2 I)$. Using these choices, Assumption 2 will be satisfied. We set $\alpha = 0.99$ and $\mu_k = 0.01$ for all k.

Figs. 7 and 8 show the steady-state EMSE and MSD of the network for these two cases. The steady-state values are obtained by averaging over the last 200 samples after initial convergence. We observe that the performance of the proposed bias-compensated LMS algorithm with adaptive noise variance estimation is almost identical to that of the ideal case with known noise variances.

Fig. 9 illustrates the tracking performance of the bias-compensated diffusion LMS algorithm for these two cases for a sudden change in the unknown parameter w^o and compares the results to that of the standard diffusion LMS algorithm given in [33]. The variation in the unknown parameter vector occurs at iteration $i = 550$ when w^o changes to $2w^o$. Similar conclusion as in Fig. 7 and 8 can be made for the proposed algorithms with known and estimated regression noise variances. We also observe that the proposed algorithms outperform the standard diffusion LMS [33] by nearly 10 dB in steady state.

Fig. 10 illustrates the results of regression noise variance estimation in the steady state. In this experiment, we observe that for $i \geq 350$, $\mathbb{E}[\sigma_{n,k}^2(i)] \to \sigma_{n,k}^2$. This indicates that the

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

10   IEEE/ACM TRANSACTIONS ON NETWORKING
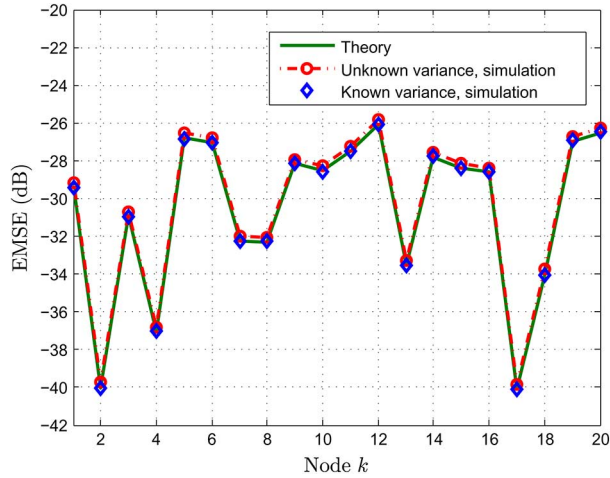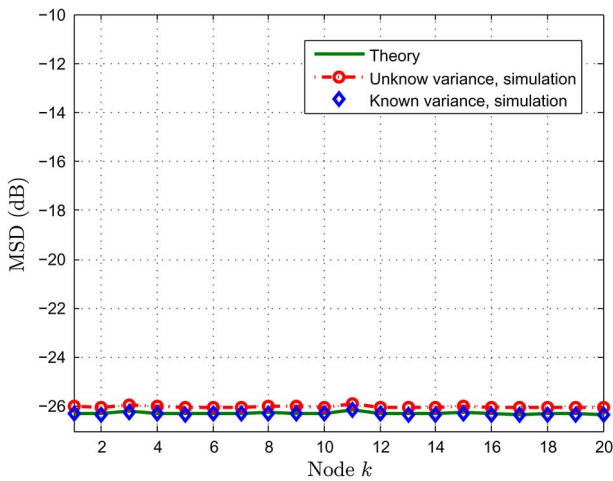

Fig. 7.   Steady-state network EMSE with known and estimated regressor noise variances.

Fig. 8.   Steady-state network MSD with known and estimated regressor noise variances.
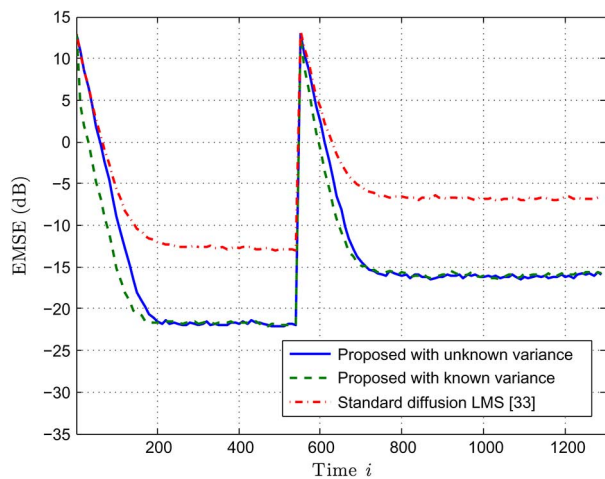

Fig. 9.   EMSE Tracking performance with known and estimated regressor noise variances.

proposed adaptive estimation strategy for computation of the nodes' regression noise variance over the network works well.
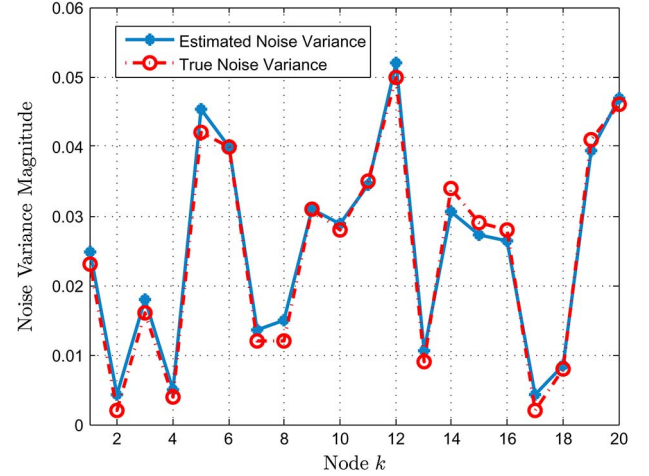

Fig. 10.   Estimated and true value of the regression noise variance, $\sigma_{n,k}^2$, over the network.

VII. CONCLUSION

We developed bias-compensated diffusion LMS strategies for parameter estimation over sensor networks where the regression data are corrupted with additive noise. The algorithms operate in a distributed manner and exchange data via single-hop communication to save energy and communication resources. The proposed algorithms estimate the regression noise variances and use them to remove the bias from the estimate. In the analysis, it has been shown that the proposed bias-compensated diffusion algorithms are unbiased and converge in the mean and mean-square error sense for sufficiently small step-sizes. We carried out computer experiments that confirmed the effectiveness of the algorithms and support the analytical findings.

APPENDIX A
COMPUTATION OF $\mathcal{G}$

This can be computed by substituting $\boldsymbol{g}(i)$ from (35) into $\mathcal{G} = \mathbb{E}[\boldsymbol{g}_i \boldsymbol{g}_i^*]$, and as a result

$$\mathcal{G} = \mathcal{C}^T \mathbb{E} \begin{bmatrix} \boldsymbol{z}_{1,i}^* \boldsymbol{v}_1(i) \\ \dots \\ \boldsymbol{z}_{N,i}^* \boldsymbol{v}_N(i) \end{bmatrix} [\boldsymbol{v}_1^*(i) \boldsymbol{z}_{1,i}, \dots, \boldsymbol{v}_N^*(i) \boldsymbol{z}_{N,i}] \, \mathcal{C}. \quad (94)$$

The (k, j)th block of the above matrix can be computed as

$$[\mathcal{G}]_{k,j} = \begin{cases} 0, & k \neq j \\ \sigma_{v,k}^2 \left(R_{u,k} + \sigma_{n,k}^2 I \right), & k = j \end{cases} \quad (95)$$

and (61) follows.

APPENDIX B
COMPUTATION OF Π

We rewrite Π as

$$\Pi = \mathbb{E}[\boldsymbol{\mathcal{P}}_i \Omega \boldsymbol{\mathcal{P}}_i^*] \quad (96)$$

where $\Omega = \omega^o \omega^{o*}$. The (k, j)th block of Π can be computed as

$$\Pi_{k,j} = \mathbb{E} \sum_\ell \sum_m c_{\ell,k} c_{m,j} \left(\boldsymbol{z}_{\ell,i}^* \boldsymbol{n}_{\ell,i} - \sigma_{n,\ell}^2 I \right)$$
$$\times \Omega_{kj} \left(\boldsymbol{n}_{m,i}^* \boldsymbol{z}_{m,i} - \sigma_{n,m}^2 I \right). \quad (97)$$

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

ABDOLEE AND CHAMPAGNE: DIFFUSION LMS STRATEGIES IN SENSOR NETWORKS WITH NOISY INPUT DATA 11

We use (1) to replace $\boldsymbol{z}_{\ell,i}$ and $\boldsymbol{z}_{m,i}$

$$\Pi_{k,j} = \mathbb{E} \sum_{\ell} \sum_{m} c_{\ell,k} c_{m,j} \left(\boldsymbol{u}_{\ell,i}^* \boldsymbol{n}_{\ell,i} + \boldsymbol{n}_{\ell,i}^* \boldsymbol{n}_{\ell,i} - \sigma_{n,\ell}^2 I \right) \Omega_{kj}$$
$$\times \left(\boldsymbol{n}_{m,i}^* \boldsymbol{u}_{m,i} + \boldsymbol{n}_{m,i}^* \boldsymbol{n}_{m,i} - \sigma_{n,m}^2 I \right). \quad (98)$$

This leads to

$$\Pi_{k,j} = \sum_{\ell} \sum_{m} c_{\ell,k} c_{m,j} \mathbb{E} \left[\boldsymbol{u}_{\ell,i}^* \boldsymbol{n}_{\ell,i} \Omega_{kj} \boldsymbol{n}_{m,i}^* \boldsymbol{u}_{m,i} \right] \quad (99)$$
$$+ \sum_{\ell} \sum_{m} c_{\ell,k} c_{m,j} \mathbb{E} \left[\boldsymbol{n}_{\ell,i}^* \boldsymbol{n}_{\ell,i} \Omega_{kj} \boldsymbol{n}_{m,i}^* \boldsymbol{n}_{m,i}^* \right]$$
$$- \sum_{\ell} \sum_{m} c_{\ell,k} c_{m,j} \mathbb{E} \left[\sigma_{n,\ell}^2 \Omega_{kj} \boldsymbol{n}_{m,i}^* \boldsymbol{n}_{m,i} \right]. \quad (100)$$

If we assume that the regression $\{\boldsymbol{u}_{k,i}\}$ and noise $\{\boldsymbol{n}_{k,i}\}$ are zero-mean circular Gaussian complex-valued vectors with uncorrelated entries, then

$$\mathbb{E} \left[\boldsymbol{u}_{\ell,i}^* \boldsymbol{n}_{\ell,i} \Omega_{kj} \boldsymbol{n}_{m,i}^* \boldsymbol{u}_{m,i} \right]$$
$$= \begin{cases} 0, & \ell \neq m \\ \sigma_{n,\ell}^2 \text{Tr}(\Omega_{kj}) R_{u,\ell}, & \ell = m \end{cases} \quad (101)$$

$$\mathbb{E} \left[\boldsymbol{n}_{\ell,i}^* \boldsymbol{n}_{\ell,i} \Omega_{kj} \boldsymbol{n}_{m,i}^* \boldsymbol{n}_{m,i} \right]$$
$$= \begin{cases} \sigma_{n,\ell}^2 \Omega_{kj} \sigma_{n,m}^2, & \ell \neq m \\ \beta \sigma_{n,\ell}^2 \Omega_{kj} \sigma_{n,m}^2 + \sigma_{n,\ell}^2 I \text{Tr} \left(\Omega_{kj} \sigma_{n,\ell}^2 I \right), & \ell = m \end{cases}$$
$$(102)$$

and

$$\mathbb{E} \left[\sigma_{n,\ell}^2 I \Omega_{kj} \boldsymbol{n}_{m,i}^* \boldsymbol{n}_{m,i} \right] = \sigma_{n,\ell}^2 \Omega_{kj} \sigma_{n,m}^2. \quad (103)$$

We note that

$$\Omega_{k,j} = w_k^o w_j^{o*} \quad (104)$$

where $w_j^o = w_k^o$, $\forall k, j \in \{1, 2, \ldots N\}$. Therefore

$$\Omega_{\ell k} = \Omega_{mn} \qquad \forall \ell, k, m, n \in \{1, 2, \ldots, N\} \quad (105)$$

and $\text{Tr}(\Omega_{kj}) = \|w^o\|^2$. As a result

$$\Pi_{k,j} = \sum_{\ell} c_{\ell,k} c_{\ell,j} \left\{ \sigma_{n,\ell}^2 \|w^o\|^2 \left(R_{u,\ell} + \sigma_{n,\ell}^2 I \right) \right.$$
$$\left. + (\beta - 1) \sigma_{n,\ell}^4 w^o w^{o*} \right\}. \quad (106)$$

ACKNOWLEDGMENT

REFERENCES

[1] R. Abdolee, B. Champagne, and A. H. Sayed, "A diffusion LMS strategy for parameter estimation in noisy regressor applications," in *Proc. 20th EUSIPCO*, Bucharest, Romania, Aug. 2012, pp. 749–753.

[2] G. Golub and C. Van Loan, "An analysis of the total least squares problem," *SIAM J. Numer. Anal.*, vol. 17, no. 6, pp. 883–893, 1980.

[3] S. Van Huffel and J. Vandewalle, *The Total Least Squares Problem: Computational Aspects and Analysis*. Philadelphia, PA, USA: SIAM, 1991.

[4] R. DeGroat and E. Dowling, "Data least squares problem and channel equalization," *IEEE Trans. Signal Process.*, vol. 41, no. 1, pp. 407–411, Jan. 1993.

[5] P. Regalia, "An unbiased equation error identifier and reduced-order approximations," *IEEE Trans. Signal Process.*, vol. 42, no. 6, pp. 1397–1412, Jun. 1994.

[6] K. C. Ho and Y. T. Chan, "Bias removal in equation-error adaptive IIR filters," *IEEE Trans. Signal Process.*, vol. 43, no. 1, pp. 51–62, Jan. 1995.

[7] H. Shin and W. Song, "Bias-free adaptive IIR filtering," *IEICE Trans. Fund. Elect., Commun. and Comput. Sci.*, vol. 84, pp. 1273–1279, May 2001.

[8] H. C. So, "Modified LMS algorithm for unbiased impulse response estimation in nonstationary noise," *Electron. Lett.*, vol. 35, no. 10, pp. 791–792, Oct. 1999.

[9] C. Davila, "An efficient recursive total least squares algorithm for FIR adaptive filtering," *IEEE Trans. Signal Process.*, vol. 42, no. 2, pp. 268–280, Feb. 1994.

[10] D. Feng, Z. Bao, and L. Jiao, "Total least mean squares algorithm," *IEEE Trans. Signal Process.*, vol. 46, no. 8, pp. 2122–2130, Aug. 1998.

[11] H. C. So, "LMS-based algorithm for unbiased FIR filtering with noisy measurements," *Electron. Lett.*, vol. 37, no. 23, pp. 1418–1420, 2001.

[12] S. Jo and S. Kim, "Consistent normalized least mean square filtering with noisy data matrix," *IEEE Trans. Signal Process.*, vol. 53, no. 6, pp. 2112–2123, Jun. 2005.

[13] S. Sagara and K. Wada, "On-line modified least-squares parameter estimation of linear discrete dynamic systems," *Int. J. Control*, vol. 25, no. 3, pp. 329–343, Mar. 1977.

[14] L. Jia, M. Ikenoue, C. Jin, and K. Wada, "On bias compensated least squares method for noisy input-output system identification," in *Proc. 40th IEEE Conf. Decision Control*, Orlando, FL, USA, Dec. 2001, pp. 3332–3337.

[15] D. Z. Feng, Z. Bao, and X. Zhang, "Modified RLS algorithm for unbiased estimation of FIR system with input and output noise," *Electron. Lett.*, vol. 36, no. 3, pp. 273–274, Mar. 2000.

[16] W. Zheng and C. Feng, "Unbiased parameter estimation of linear systems in the presence of input and output noise," *Int. J. Adaptive Control Signal Process.*, vol. 3, no. 3, pp. 231–251, Mar. 1989.

[17] B. Kang, J. Yoo, and P. Park, "Bias-compensated normalised LMS algorithm with noisy input," *Electron. Lett.*, vol. 49, no. 8, pp. 538–539, Apr. 2013.

[18] A. Bertrand and M. Moonen, "Consensus-based distributed total least squares estimation in *ad hoc* wireless sensor networks," *IEEE Trans. Signal Process.*, vol. 59, no. 5, pp. 2320–2330, May 2011.

[19] A. Bertrand and M. Moonen, "Low-complexity distributed total least squares estimation in *ad hoc* sensor networks," *IEEE Trans. Signal Process.*, vol. 60, no. 8, pp. 4321–4333, Aug. 2012.

[20] D. P. Bertsekas, "A new class of incremental gradient methods for least squares problems," *SIAM J. Optim.*, vol. 7, no. 4, pp. 913–926, Apr. 1997.

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

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

[23] C. G. Lopes and A. H. Sayed, "Incremental adaptive strategies over distributed networks," *IEEE Trans. Signal Process.*, vol. 55, no. 8, pp. 4064–4077, Aug. 2007.

[24] J. Tsitsiklis, D. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Trans. Autom. Control*, vol. AC-31, no. 9, pp. 803–812, Sep. 1986.

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

[26] P. Braca, S. Marano, and V. Matta, "Running consensus in wireless sensor networks," in *Proc. Int. Conf. Inf. Fusion*, Cologne, Germany, Jun. 2008, pp. 1–6.

[27] S. Sardellitti, M. Giona, and S. Barbarossa, "Fast distributed average consensus algorithms based on advection-diffusion processes," *IEEE Trans. Signal Process.*, vol. 58, no. 2, pp. 826–842, Feb. 2010.

[28] A. G. Dimakis, S. Kar, J. M. F. Moura, M. G. Rabbat, and A. Scaglione, "Gossip algorithms for distributed signal processing," *Proc. IEEE*, vol. 98, no. 11, pp. 1847–1864, Nov. 2010.

[29] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Trans. Autom. Control*, vol. 54, no. 1, pp. 48–61, Jan. 2009.

[30] S. Kar and J. M. Moura, "Convergence rate analysis of distributed gossip (linear parameter) estimation: Fundamental limits and trade-offs," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 4, pp. 674–690, Aug. 2011.

[31] C. G. Lopes and A. H. Sayed, "Diffusion least-mean squares over adaptive networks: Formulation and performance analysis," *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 3122–3136, Jul. 2008.

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

[33] F. S. Cattivelli and A. H. Sayed, "Diffusion LMS strategies for distributed estimation," *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1035–1048, Mar. 2010.

[34] P. Di Lorenzo and S. Barbarossa, "A bio-inspired swarming algorithm for decentralized access in cognitive radio," *IEEE Trans. Signal Process.*, vol. 59, no. 12, pp. 6160–6174, Dec. 2011.

[35] N. Takahashi, I. Yamada, and A. H. Sayed, "Diffusion least-mean squares with adaptive combiners: Formulation and performance analysis," *IEEE Trans. Signal Process.*, vol. 58, no. 9, pp. 4795–4810, Sep. 2010.

[36] S. Chouvardas, K. Slavakis, and S. Theodoridis, "Adaptive robust distributed learning in diffusion sensor networks," *IEEE Trans. Signal Process.*, vol. 59, no. 10, pp. 4692–4707, Oct. 2011.

[37] J. Chen and A. H. Sayed, "Diffusion adaptation strategies for distributed optimization and learning over networks," *IEEE Trans. Signal Process.*, vol. 60, no. 8, pp. 4289–4305, Aug. 2012.

[38] S. Y. Tu and A. H. Sayed, "Diffusion strategies outperform consensus strategies for distributed estimation over adaptive networks," *IEEE Trans. Signal Process.*, vol. 60, no. 12, pp. 6127–6234, Dec. 2012.

[39] X. Zhao and A. H. Sayed, "Performance limits for distributed estimation over LMS adaptive networks," *IEEE Trans. Signal Process.*, vol. 60, no. 10, pp. 5107–5124, Oct. 2012.

[40] A. H. Sayed, S. Y. Tu, J. Chen, X. Zhao, and Z. Towfic, "Diffusion strategies for adaptation and learning over networks," *IEEE Signal Process. Mag.*, vol. 30, no. 5, pp. 155–171, May 2013.

[41] P. Di Lorenzo and A. H. Sayed, "Sparse distributed learning based on diffusion adaptation," *IEEE Trans. Signal Process.*, vol. 61, no. 6, pp. 1419–1433, Mar. 2013.

[42] S.-Y. Tu and A. H. Sayed, "Distributed decision-making over adaptive networks," *IEEE Trans. Signal Process.*, vol. 62, no. 5, pp. 1054–1069, Mar. 2014.

[43] A. Bertrand, M. Moonen, and A. H. Sayed, "Diffusion bias-compensated RLS estimation over adaptive networks," *IEEE Trans. Signal Process.*, vol. 59, no. 11, pp. 5212–5224, Nov. 2011.

[44] R. Koning, H. Neudecker, and T. Wansbeek, "Block Kronecker products and the vecb operator," *Linear Algebra Its Appl.*, vol. 149, pp. 165–184, Apr. 1991.

[45] A. H. Sayed, *Adaptive Filters*. Hoboken, NJ, USA: Wiley, 2008.

[46] K. S. Miller, "On the inverse of the sum of matrices," *Math. Mag.*, vol. 54, no. 2, pp. 67–72, 1981.

[47] S. P. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[48] D. Bertsekas and J. Tsitsiklis, *Parallel and Distributed Computation*. Upper Saddle River, NJ, USA: Prentice-Hall, 1989.

[49] A. H. Sayed, "Diffusion adaptation over networks," in *E-Reference Signal Processing*, R. Chellapa and S. Theodoridis, Eds. Amsterdam, The Netherlands: Elsevier, 2013, manuscript arXiv:1205.4220v1, May 2012.

[50] X. Zhao, S. Y. Tu, and A. Sayed, "Diffusion adaptation over networks under imperfect information exchange and non-stationary data," *IEEE Trans. Signal Process.*, vol. 60, no. 7, pp. 3460–3475, Jul. 2012.

[51] W. X. Zheng, "A least-squares based algorithm for FIR filtering with noisy data," in *Proc. ISCAS*, Bangkok, Thailand, May 2003, vol. 4, pp. 444–447.

[52] L. Jia, R. Tao, Y. Wang, and K. Wada, "Forward/backward prediction solution for adaptive noisy FIR filtering," *Sci. China, Inf. Sci.*, vol. 52, no. 6, pp. 1007–1014, 2009.

Reza Abdolee (S'11) is currently pursuing the Ph.D. degree in electrical and computer engineering at McGill University, Montreal, QC, Canada.

In 2012, he was a Research Scholar with Bell Labs, Alcatel-Lucent, Stuttgart, Germany. In 2011, he was a visiting Ph.D. student with the Department of Electrical Engineering, University of California, Los Angeles (UCLA), CA, USA. From 2006 to 2008, he worked as a Staff Engineer with the Wireless Communication Center, University of Technology, Malaysia (UTM), Johor, Malaysia, where he implemented a switch-beam smart antenna system for wireless network applications. His research interests include communication theory, statistical signal processing, optimization, and hardware design and integration.

Mr. Abdolee was a recipient of several awards and scholarships, including the NSERC Postgraduate Scholarship, FQRNT Doctoral Research Scholarship, McGill Graduate Research Mobility Award, DAAD-RISE International Internship Scholarship (Germany), FQRNT International Internship Scholarship, McGill Graduate Funding and Travel Award, McGill International Doctoral Award, and ReSMiQ International Doctoral Scholarship.

Benoit Champagne (S'87–M'89–SM'03) received the B.Ing. degree in engineering physics from the Ecole Polytechnique de Montréal, Montréal, QC, Canada, in 1983, the M.Sc. degree in physics from the Université de Montréal, Montréal, QC, Canada, in 1985, and the Ph.D. degree in electrical engineering from the University of Toronto, Toronto, ON, Canada, in 1990.

From 1990 to 1999, he was an Assistant and then Associate Professor with INRS-Telecommunications, Université du Quebec, Montréal, QC, Canada. In 1999, he joined McGill University, Montréal, QC, Canada, where he is now a Full Professor with the Department of Electrical and Computer Engineering. He also served as Associate Chairman of Graduate Studies in the department from 2004 to 2007. His research focuses on the development and performance analysis of advanced algorithms for the processing of information bearing signals by digital means. His interests span many areas of statistical signal processing, including detection and estimation, sensor array processing, adaptive filtering, and applications thereof to broadband communications and speech processing, where he has published extensively. His research has been funded by the Natural Sciences and Engineering Research Council (NSERC) of Canada, the Fonds de Recherche sur la Nature et les Technologies from the Government of Quebec, Prompt Quebec, as well as some major industrial sponsors, including Nortel Networks, Bell Canada, InterDigital, and Microsemi.