Estimation of Space-Time Varying Parameters Using a Diffusion LMS Algorithm

Reza Abdolee, *Student Member, IEEE*, Benoit Champagne, *Senior Member, IEEE*, and Ali H. Sayed, *Fellow, IEEE*

***Abstract*—We study the problem of distributed adaptive estimation over networks where nodes cooperate to estimate physical parameters that can vary over both *space* and *time* domains. We use a set of basis functions to characterize the space-varying nature of the parameters and propose a diffusion least mean-squares (LMS) strategy to recover these parameters from successive time measurements. We analyze the stability and convergence of the proposed algorithm, and derive closed-form expressions to predict its learning behavior and steady-state performance in terms of mean-square error. We find that in the estimation of the space-varying parameters using distributed approaches, the covariance matrix of the regression data at each node becomes rank-deficient. Our analysis reveals that the proposed algorithm can overcome this difficulty to a large extent by benefiting from the network stochastic matrices that are used to combine exchanged information between nodes. We provide computer experiments to illustrate and support the theoretical findings.**

***Index Terms*—Diffusion adaptation, distributed processing, interpolation, parameter estimation, sensor networks, space-varying parameters.**

I. Introduction

In previous studies on diffusion algorithms for adaptation over networks, including least-mean-squares (LMS) or recursive least squares (RLS) types, the parameters being estimated are often assumed to be *space-invariant* [1]–[6]. In other words, all agents are assumed to sense and measure data that arise from an underlying physical model that is represented by fixed parameters over the spatial domain. Some studies considered particular applications of diffusion strategies to data that arise from potentially different models [7], [8]. However, the proposed techniques in these works are not immediately applicable to scenarios where the estimation parameters vary over space across the network. This situation is encountered in many applications, including the monitoring of fluid flow in underground porous media [9], the tracking of population dispersal in ecology [10], the localization of distributed sources in dynamic systems [11], and the modeling of diffusion phenomena in inhomogeneous media [12]. In these applications, the space-varying parameters being estimated usually result from discretization of the coefficients of an underlying partial differential equation through spatial sampling.

The estimation of spatially-varying parameters has been addressed in several previous studies, including [13]–[17]. In these works and other similar references on the topic, the solutions typically rely on the use of a central processing (fusion) unit and less attention is paid to distributed and in-network processing solutions. Distributed algorithms are useful in large networks when there is no powerful fusion center and when the energy and communication resources of individual nodes are limited. Many different classes of distributed algorithms for parameter estimation over networks have been proposed in the recent literature, including incremental method [18]–[22], consensus methods [23]–[34], and diffusion methods [2], [3], [6], [35]–[37]. Incremental techniques require to set-up a cyclic path between nodes over the network and are therefore sensitive to link failures. Consensus techniques require doubly-stochastic combination policies and can cause network instability in applications involving continuous adaptation and tracking [5]. In comparison, diffusion strategies demonstrate a stable behavior over networks regardless of the topology and endow networks with real-time adaptation and learning abilities [5], [6], [36].

Motivated by these considerations, in this paper, we develop a distributed LMS algorithm of the diffusion type to enable the estimation and tracking of parameters that may vary over both *space* and *time*. Our approach starts by introducing a linear regression model to characterize space-time varying phenomena over networks. This model is derived by discretizing a representative second-order partial differential equation (PDE), which can be useful in describing many dynamic systems with spatially-varying properties. We then introduce a set of basis functions, e.g., shifted Chebyshev polynomials, to represent the space-varying parameters of the underlying phenomena in terms of a finite set of space-invariant expansion coefficients. Building on this representation, we develop a diffusion LMS strategy that cooperatively estimates, interpolates, and tracks the model parameters over the network. We analyze the convergence and stability of the developed algorithm, and derive closed-form expressions to characterize the learning and convergence behavior of the nodes in mean-square-error sense. It turns out that in the context of space-time varying models, the covariance matrices of the regression data at the various nodes can become rank deficient. This property influences the learning behavior of the network and causes the estimates to become biased. We elaborate on how the judicious use of stochastic combination matrices can help alleviate this difficulty.

Manuscript received March 08, 2013; revised June 28, 2013 and October 14, 2013; accepted October 14, 2013. Date of publication November 07, 2013; date of current version December 26, 2013. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Hing Cheung So. The work of R. Abdolee and B. Champagne was supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada. The work of A. H. Sayed was supported in part by NSF grant CCF-1011918.

R. Abdolee and B. Champagne are with the Department of Electrical and Computer Engineering, McGill University, Montreal, QC H3A 2A7, Canada (e-mail: reza.abdolee@mail.mcgill.ca; benoit.champagne@mcgill.ca).

A. H. Sayed is with the Department of Electrical Engineering, University of California, Los Angeles, CA 90095 USA (e-mail: sayed@ee.ucla.edu).

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

Digital Object Identifier 10.1109/TSP.2013.2289888

The paper is organized as follows. In Section II, we introduce a space-varying linear regression model which is motivated from a physical phenomenon characterized by a PDE, and formulate an optimization problem to find the unknown parameters of the introduced model. In Section III, we derive a diffusion LMS algorithm that solves this problem in a distributed and adaptive manner. We analyze the performance of the algorithm in Section IV, and present the numerical results of computer simulations in Section V. The concluding remarks appear in Section VI.

Notation: Matrices are represented by upper-case and vectors by lower-case letters. Boldface fonts are reserved for random variables and normal fonts are used for deterministic quantities. Superscript $(\cdot)^T$ denotes transposition for real-valued vectors and matrices and $(\cdot)^*$ denotes conjugate transposition for complex-valued vectors and matrices. The symbol $\mathbb{E}[\cdot]$ is the expectation operator, $\mathrm{Tr}(\cdot)$ represents the trace of its matrix argument and $\mathrm{diag}\{\cdot\}$ extracts the diagonal entries of a matrix, or constructs a diagonal matrix from a vector. I_M represents the identity matrix of size $M \times M$ (subscript M is omitted when the size can be understood from the context). The $\mathrm{vec}(\cdot)$ operator vectorizes a matrix by stacking its columns on top of each other. A set of vectors are stacked into a column vector by $\mathrm{col}\{\cdot\}$.

II. Modeling and Problem Formulation

In this section, we introduce a linear regression model that can be used to describe dynamic systems with spatially varying properties. We derive the model from a representative second-order one-dimensional PDE that is used to characterize the evolution of the pressure distribution in inhomogeneous media and features a diffusion coefficient and an input source, both of which vary over space. Extension and generalization of the proposed approach, in modeling space-varying phenomena, to PDEs of higher order or defined over two-dimensional space are generally straightforward (see, e.g., Section V-C).

The PDE we consider is expressed as [12], [38]:

$$\frac{\partial f(x,t)}{\partial t} = \frac{\partial}{\partial x}\left(\theta(x)\frac{\partial f(x,t)}{\partial x}\right) + q(x,t) \tag{1}$$

where $(x,t) \in [0,L] \times [0,T]$ denote the space and time variables with upper limits $L \in \mathbb{R}^+$ and $T \in \mathbb{R}^+$, respectively, $f(x,t) : \mathbb{R}^2 \rightarrow \mathbb{R}$, represents the system distribution (e.g., pressure or temperature) under study, $\theta(x) : \mathbb{R} \rightarrow \mathbb{R}$ is the space-varying diffusion coefficient and $q(x,t) : \mathbb{R}^2 \rightarrow \mathbb{R}$ is the input distribution that includes sources and sinks. PDE (1) is assumed to satisfy the Dirichlet boundary conditions[1], $f(0,t) = f(L,t) = 0$ for all $t \in [0,T]$. The distribution of the system at $t = 0$ is given by $f(x,0) = y(x)$ for $x \in [0,L]$. It is convenient to rewrite (1) as:

$$\frac{\partial f(x,t)}{\partial t} = \theta(x)\frac{\partial^2 f(x,t)}{\partial x^2} + \frac{\partial \theta(x)}{\partial x}\frac{\partial f(x,t)}{\partial x} + q(x,t) \tag{2}$$

and employ the finite difference method (FDM) to discretize the PDE over the time and space domains [39]. For N and P given positive integers, let $\Delta x = L/(N+1)$ and $x_k = k\Delta x$ for $k \in \{0,1,2,\ldots,N+1\}$, and similarly, let $\Delta t = T/P$ and $t_i = i\Delta t$ for $i \in \{0,1,2,\ldots,P\}$. We further introduce the sampled values of the pressure distribution $f_k(i) \triangleq f(x_k,t_i)$, input $q_k(i) \triangleq q(x_k,t_i)$, and space-varying coefficient $\theta_k \triangleq \theta(x_k)$. It can be verified that applying FDM to (2) yields the following recursion:

$$f_k(i) = u_{k,i}h_k^o + \Delta t\, q_k(i-1), \quad k \in \{1,2,\ldots,N\} \tag{3}$$

where the vectors $h_k^o \in \mathbb{R}^{3\times 1}$ and $u_{k,i} \in \mathbb{R}^{1\times 3}$ are defined as

$$h_k^o \triangleq [h_{1,k}^o, h_{2,k}^o, h_{3,k}^o]^T \tag{4}$$

$$u_{k,i} \triangleq [f_{k-1}(i-1),\ f_k(i-1),\ f_{k+1}(i-1)] \tag{5}$$

the entries $h_{m,k}^o \in \mathbb{R}$ are:

$$h_{1,k}^o = \frac{\nu}{4}(\theta_{k-1} + 4\theta_k - \theta_{k+1}) \tag{6}$$

$$h_{2,k}^o = 1 - 2\nu\,\theta_k \tag{7}$$

$$h_{3,k}^o = \frac{\nu}{4}(-\theta_{k-1} + 4\theta_k + \theta_{k+1}) \tag{8}$$

and $\nu = \Delta t/\Delta x^2$. Note that relation (3) is defined for $k \in \{1,2,\ldots,N\}$, i.e., no data sampling is required to be taken at $x = \{0,L\}$ because $f_0(i)$ and $f_{N+1}(i)$ respectively correspond to the known boundary conditions $f(0,t)$ and $f(L,t)$. For monitoring purposes (e.g., estimation of $\theta(x)$), sensor nodes collect noisy measurement samples of $f(x,t)$ across the network. We denote these scalar measurement samples by

$$\boldsymbol{z}_k(i) = f_k(i) + \boldsymbol{n}_k(i) \tag{9}$$

where $\boldsymbol{n}_k(i) \in \mathbb{R}$ is random noise term. Substituting (3) into (9) leads to

$$\boldsymbol{d}_k(i) = u_{k,i}h_k^o + \boldsymbol{n}_k(i) \tag{10}$$

where

$$\boldsymbol{d}_k(i) \triangleq \boldsymbol{z}_k(i) - \Delta t\, q_k(i-1). \tag{11}$$

The space-dependent model (10) can be generalized to accommodate higher order PDE's, or to describe systems with more than one spatial dimension. In the generalized form, we assume that $u_{k,i}$ is random due to the possibility of sampling errors, and therefore represent it using boldface notation $\boldsymbol{u}_{k,i}$. We also let h_k^o and $\boldsymbol{u}_{k,i}$ be M-dimensional vectors. In addition, we denote the noise more generally by the symbol $\boldsymbol{v}_k(i)$ to account for different sources of errors, including the measurement noise shown in (9) and modeling errors. Considering this generalization, the space-varying regression model that we shall consider is of the form:

$$\boldsymbol{d}_k(i) = \boldsymbol{u}_{k,i}h_k^o + \boldsymbol{v}_k(i) \tag{12}$$

where $\boldsymbol{d}_k(i) \in \mathbb{R}, \boldsymbol{u}_{k,i} \in \mathbb{R}^{1\times M}, h_k^o \in \mathbb{R}^{M\times 1}$ and $\boldsymbol{v}_k(i) \in \mathbb{R}$. In this work, we study networks that monitor phenomena characterized by regression models of the form (12), where the objective is to estimate the space-varying parameter vectors h_k^o for $k \in \{1,2,\ldots,N\}$. In particular, we seek a distributed solution in the form of an adaptive algorithm with a diffusion mode of cooperation to enable the nodes to estimate and track these

[1] Generalization of the boundary conditions to nonzero values is possible as well.

parameters over both space and time. The available information for estimation of the $\{h_k^o\}$ are the measurement samples, $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$, collected at the N spatial position x_k, which we take to represent N nodes.

Several studies, e.g., [13]–[15], solved space-varying parameter estimation problems using *non-adaptive centralized* techniques. In centralized optimization, the space-varying parameters $\{h_k^o\}$ are found by minimizing the following global cost function over the variables $\{h_k\}$:

$$J(h_1, \ldots, h_N) \triangleq \sum_{k=1}^{N} J_k(h_k) \tag{13}$$

where

$$J_k(h_k) \triangleq \mathbb{E}|\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i} h_k|^2. \tag{14}$$

To find h_k^o using distributed mechanisms, however, preliminary steps are required to transform the global cost (13) into a suitable form convenient for decentralized optimization [2]. Observe from (6)–(8) that collaborative processing is beneficial in this case because the h_k^o of neighboring nodes are related to each other through the space-dependent function $\theta(x)$.

Remark 1: Note that if nodes individually estimate their own space-varying parameters by minimizing $J_k(h_k)$, then at each time instant, they will need to transmit their estimates to a fusion center for interpolation, in order to determine the value of the model parameters over regions of space where no measurements were collected. Using the proposed distributed algorithm in Section III-B, it will be possible to update the estimates and interpolate the results in a fully distributed manner. Cooperation also helps the nodes refine their estimates and perform more accurate interpolation. ■

III. Adaptive Distributed Optimization

In distributed optimization over networked systems, nodes achieve their common objective through collaboration. Such an objective may be defined as finding a global parameter vector that minimizes a given cost function that encompasses the entire set of nodes. For the problem stated in this study, the unknown parameters in (13) are node-dependent. However, as we explained in Section II, these space-varying parameters are related through a well-defined function, e.g., $\theta(x)$ over the spatial domain. In the continuous space domain, the entries of each h_k^o, i.e., $\{h_{1,k}^o, \ldots, h_{M,k}^o\}$ can be interpreted as samples of M unknown space-varying parameter functions $\{h_1^o(x), \ldots, h_M^o(x)\}$ at location $x = x_k$, as illustrated in Fig. 1.

We can now use the well-established theory of interpolation to find a set of linear expansion coefficients, common to all the nodes, in order to estimate space-varying parameters using distributed optimization. Specifically, we assume that the m-th unknown space-varying parameter function, $h_m^o(x)$ can be expressed as a unique linear combination of some N_b space basis functions, i.e.,

$$h_m^o(x) = W_{m,1} b_1(x) + W_{m,2} b_2(x) + \cdots + W_{m,N_b} b_{N_b}(x) \tag{15}$$

where $\{W_{m,n}\}$ are the unique expansion coefficients and $\{b_n(x)\}$ are the basis functions. In the application examples treated in Section V, we adopt shifted Chebyshev polynomials as basis functions, which are generated using the following expressions [40]

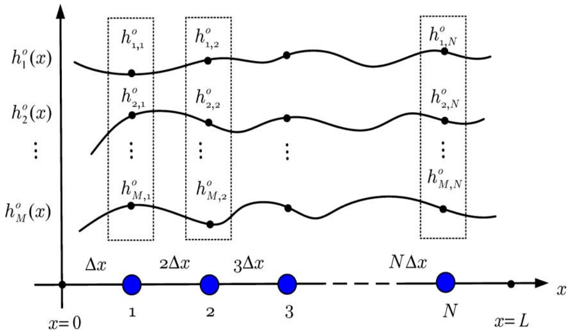


Fig. 1. An example of the space-varying parameter estimation problem over a one-dimensional network topology. The larger circles on the x-axis represent the node locations at $x = x_k$. These nodes collect samples $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$ to estimate the space-varying parameters $\{h_k^o\}$. For simplicity in defining the vectors b_k in (20), for this example, we assume that the node positions x_k are uniformly spaced, however, generalization to non-uniform spacing is straightforward.

$$b_1(x) = 1, \qquad b_2(x) = 2x - 1 \tag{16}$$

$$b_{n+1}(x) = 2(2x-1)b_n(x) - b_{n-1}(x), \quad 2 < n < N_b. \tag{17}$$

The choice of a suitable set of basis functions $\{b_n(x)\}_{n=1}^{N_b}$ is application-specific and guided by multiple considerations such as representation efficiency, low computational complexity, interpolation capability, and other desirable properties, such as orthogonality. Chebyshev basis functions yield good results in terms of the above criteria and helps avoid the Runge's phenomenon at the endpoints of the space interval [40].

The sampled version of the m-th space-varying parameter $h_m^o(x)$ in (15), at $x = x_k = k\Delta x$, can be written as:

$$h_{m,k}^o = W_m^T b_k \tag{18}$$

where

$$W_m \triangleq [W_{m,1}, W_{m,2}, \ldots, W_{m,N_b}]^T \tag{19}$$

$$b_k \triangleq [b_{1,k}, \ldots, b_{N_b,k}]^T \tag{20}$$

and each entry $b_{n,k}$ is obtained by sampling the corresponding basis function at the same location, i.e.,

$$b_{n,k} \triangleq b_n(x_k) = b_n(k\Delta x). \tag{21}$$

Collecting the sampled version of all M functions $h_m^o(x)$ for $m \in \{1, \ldots, M\}$ into a column vector as

$$h_k^o = [h_{1,k}^o, h_{2,k}^o, \ldots, h_{M,k}^o]^T \tag{22}$$

and using (18), we arrive at:

$$h_k^o = W^o b_k \tag{23}$$

where

$$W^o \triangleq \begin{bmatrix} W_{1,1}^o & W_{1,2}^o & \cdots & W_{1,N_b}^o \\ W_{2,1}^o & W_{2,2}^o & \cdots & W_{2,N_b}^o \\ \vdots & \vdots & \ldots & \vdots \\ W_{M,1}^o & W_{M,2}^o & \cdots & W_{M,N_b}^o \end{bmatrix}. \tag{24}$$

Remark 2: Several other interpolation techniques can be used to obtain the basis functions $b_n(x)$, such as the so-called kriging method [41]. The latter is a data-based weighting approach, rather than a distance-based interpolation. It is applicable in scenarios where the unknown random field to be interpolated, in our case h_k^o, is wide-sense stationary; accordingly, it requires knowledge about the means and covariances of the random field over space, as employed in [42]. If these covariances are not available, then the variogram models, describing the degree of spatial dependence of the random field, are used to generate substitutes for these covariances [43]. However, *a-priori* knowledge about the parameters of variogram models, including nugget, sill, and range, are required to obtain the spatial covariances. In this work, since neither the means and covariances nor the parameters of the variogram models of the random fields are available, we focus on interpolation techniques that rely on distance information rather than the statistics of the random field to be interpolated. ■

Returning to (23), it is convenient to rearrange W^o into an $MN_b \times 1$ column vector w^o by stacking up the columns of W^{oT}, i.e., $w^o = \text{vec}(W^{oT})$, and defining the block diagonal matrix $B_k \in \mathbb{R}^{M \times MN_b}$ as:

$$B_k \triangleq I_M \otimes b_k^T. \tag{25}$$

Then, relation (23) can be rewritten in terms of the unique parameter vector w^o as:

$$h_k^o = B_k w^o \tag{26}$$

so that substituting h_k^o from (26) into (12) yields:

$$\boldsymbol{d}_k(i) = \boldsymbol{u}_{k,i} B_k w^o + \boldsymbol{v}_k(i). \tag{27}$$

Subsequently, the global cost function (13) becomes:

$$J(w) = \sum_{k=1}^{N} \mathbb{E}|\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i} B_k w|^2. \tag{28}$$

In the following, we elaborate on how the parameter vector w^o and, hence, the $\{h_k^o\}$ can be estimated from the data $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$ using centralized and distributed adaptive optimization.

A. Centralized Adaptive Solution

We begin by stating the assumed statistical conditions on the data over the network.

Assumption 1: We assume that $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}, \boldsymbol{v}_k(i)\}$ in model (27) satisfy the following conditions:

1) $\boldsymbol{d}_k(i)$ and $\boldsymbol{u}_{k,i}$ are zero-mean, jointly wide-sense stationary random processes with second-order moments:

$$r_{du,k} = \mathbb{E}[\boldsymbol{d}_k(i) \boldsymbol{u}_{k,i}^T] \in \mathbb{R}^{M \times 1} \tag{29}$$

$$R_{u,k} = \mathbb{E}[\boldsymbol{u}_{k,i}^T \boldsymbol{u}_{k,i}] \in \mathbb{R}^{M \times M}. \tag{30}$$

2) The regression data $\{\boldsymbol{u}_{k,i}\}$ are i.i.d. over time, independent over space, and their covariance matrices, $R_{u,k}$, are positive definite for all k.
3) The noise processes $\{\boldsymbol{v}_k(i)\}$ are zero-mean, i.i.d. over time, and independent over space with variances $\{\sigma_{v,k}^2\}$.
4) The noise process $\boldsymbol{v}_k(i)$ is independent of the regression data $\boldsymbol{u}_{m,j}$ for all i, j and k, m.

■

The optimal parameter w^o that minimizes (28) can be found by setting the gradient vector of $J(w)$ to zero. This yields the following normal equations:

$$\Big(\sum_{k=1}^{N} \bar{R}_{u,k}\Big) w^o = \sum_{k=1}^{N} \bar{r}_{du,k} \tag{31}$$

where $\{\bar{R}_{u,k}, \bar{r}_{du,k}\}$ denote the second-order moments of $\boldsymbol{u}_{k,i} B_k$ and $\boldsymbol{d}_k(i)$:

$$\bar{R}_{u,k} \triangleq B_k^T R_{u,k} B_k, \quad \bar{r}_{du,k} \triangleq B_k^T r_{du,k}. \tag{32}$$

It is clear from (31) that when $\sum_{k=1}^{N} \bar{R}_{u,k} > 0$, then w^o can be determined uniquely. If, on the other hand, $\sum_{k=1}^{N} \bar{R}_{u,k}$ is singular, then we can use its pseudo-inverse to recover the minimum-norm solution of (31). Once the global solution is estimated, we can retrieve the space-varying parameter vectors h_k^o by substituting w^o into (26).

Alternatively the solution w^o of (31) can be sought iteratively by using the following steepest descent recursion:

$$\boldsymbol{w}_i^{(c)} = \boldsymbol{w}_{i-1}^{(c)} + \mu \sum_{k=1}^{N} \left(\bar{r}_{du,k} - \bar{R}_{u,k} \boldsymbol{w}_{i-1}^{(c)}\right) \tag{33}$$

where $\mu > 0$ is a step-size parameter and $\boldsymbol{w}_i^{(c)}$ is the estimate of w^o at the i-th iteration. Recursion (33) requires the centralized processor to have knowledge of the covariance matrices, $\bar{R}_{u,k}$, and cross covariance vectors, $\bar{r}_{du,k}$, across all nodes. In practice, these moments are unknown in advance, and we therefore use their instantaneous approximations in (33). This substitution leads to the centralized LMS strategy (34)–(35) for space-varying parameter estimation over networks.

Algorithm 1: Centralized LMS

$$\boldsymbol{w}_i^{(c)} = \boldsymbol{w}_{i-1}^{(c)} + \mu \sum_{k=1}^{N} B_k^T \boldsymbol{u}_{k,i}^T \left(\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i} B_k \boldsymbol{w}_{i-1}^{(c)}\right) \tag{34}$$

$$\boldsymbol{h}_{k,i} = B_k \boldsymbol{w}_i^{(c)}, \quad k \in \{1, 2, \ldots, N\}. \tag{35}$$

In this algorithm, at any given time instant i, each node transmits its data $\{\boldsymbol{u}_{k,i}, \boldsymbol{d}_k(i)\}$ to the central processing unit to update $\boldsymbol{w}_{i-1}^{(c)}$. Subsequently, the algorithm obtains an estimate for the space-varying parameters, $\boldsymbol{h}_{k,i}$, by using the updated estimate $\boldsymbol{w}_i^{(c)}$, and the basis function matrix at location k, (i.e., B_k). This latter step can also be used as an interpolation mechanism to estimate the space-varying parameters at locations other than the pre-determined locations $\{x_k\}$, by using the corresponding matrix $B(x)$ for some desired location x.

B. Adaptive Diffusion Strategy

There are different distributed optimization techniques that can be applied to (28) in order to estimate w^o and consequently obtain the optimal space-varying parameters h_k^o. Let $\mathcal{N}_k$ denote the index set of the neighbors of node k, i.e., the nodes with which node k can share information (including k itself). One possible optimization strategy is to decouple the global cost (28) and write it as a set of constrained optimization problems with local variables w_k, [44], i.e.,

$$\begin{aligned}&\min_{w_k} \sum_{\ell\in\mathcal{N}_k} c_{\ell,k}\mathbb{E}|\boldsymbol{d}_\ell(i) - \boldsymbol{u}_{\ell,i}B_\ell w_k|^2\\ &\text{subject to} \quad w_k = w\end{aligned} \tag{36}$$

where $c_{\ell,k}$ are nonnegative entries of a right-stochastic matrix $C \in \mathbb{R}^{N\times N}$ satisfying:

$$c_{\ell,k} = 0 \text{ if } \ell \notin \mathcal{N}_k \text{ and } C\mathbb{1} = \mathbb{1} \tag{37}$$

and $\mathbb{1}$ is the column vector with unit entries.

The optimization problem (36) can be solved using, for example, the alternating directions method of multipliers (ADMM) [44], [45]. In the algorithm derived using this method, the Lagrangian multipliers associated with the constraints need to be updated at every iteration during the optimization process. To this end, information about the network topology is required to establish a hierarchical communication structure between nodes. In addition, the constraints imposed by (36) require all agents to agree on an exact solution; this requirement degrades the learning and tracking abilities of the nodes over the network. When some nodes observe relevant data, it is advantageous for them to be able to respond quickly to the data without being critically constrained by perfect agreement at that stage. Doing so, would enable information to diffuse more rapidly across the network.

A technique that does not suffer from these difficulties and endows networks with adaptation and learning abilities in real-time is the diffusion strategy [2], [3], [6], [35], [36]. In this technique, minimizing the global cost (28) motivates solving the following unconstrained local optimization problem for $k \in \{1,\ldots,N\}$[2]:

$$\min_w \Big(\sum_{\ell\in\mathcal{N}_k} c_{\ell,k}\mathbb{E}|\boldsymbol{d}_\ell(i) - \boldsymbol{u}_{\ell,i}B_\ell w|^2 + \sum_{\ell\in\mathcal{N}_k\backslash\{k\}} p_{\ell,k}\|w-\psi_\ell\|^2\Big) \tag{38}$$

where ψ_ℓ is the available estimate of the global parameter at node ℓ, $\mathcal{N}_k\backslash\{k\}$ denotes set $\mathcal{N}_k$ excluding node k, and $\{p_{\ell,k}\}$ are nonnegative scaling parameters. Following the arguments in [2], [3], [6], the minimization of (38) leads to a general form of the diffusion strategy described by (39)–(42), which can be specialized to several simpler yet useful forms.

Algorithm 2: Diffusion LMS

$$\boldsymbol{\phi}_{k,i-1} = \sum_{\ell\in\mathcal{N}_k} a^{(1)}_{\ell,k}\boldsymbol{w}_{\ell,i-1} \tag{39}$$

$$\boldsymbol{\psi}_{k,i} = \boldsymbol{\phi}_{k,i-1} + \mu_k \sum_{\ell\in\mathcal{N}_k} c_{\ell,k}B_\ell^T\boldsymbol{u}_{\ell,i}^T\big(\boldsymbol{d}_\ell(i) - \boldsymbol{u}_{\ell,i}B_\ell\boldsymbol{\phi}_{k,i-1}\big) \tag{40}$$

$$\boldsymbol{w}_{k,i} = \sum_{\ell\in\mathcal{N}_k} a^{(2)}_{\ell,k}\boldsymbol{\psi}_{\ell,i} \tag{41}$$

$$\boldsymbol{h}_{k,i} = B_k\boldsymbol{w}_{k,i}. \tag{42}$$

In this algorithm, $\mu_k > 0$ is the step-size at node k, $\{\boldsymbol{w}_{k,i}, \boldsymbol{\psi}_{k,i}, \boldsymbol{\phi}_{k,i-1}\}$ are intermediate estimates of w^o, $\boldsymbol{h}_{k,i}$ is an intermediate estimate of h_k^o, and $\{a^{(1)}_{\ell,k}, a^{(2)}_{\ell,k}\}$ are nonnegative entries of left-stochastic matrices $A_1, A_2 \in \mathbb{R}^{N\times N}$ that satisfy:

$$a^{(1)}_{\ell,k} = a^{(2)}_{\ell,k} = 0 \quad \text{if } \ell \notin \mathcal{N}_k \tag{43}$$

$$A_1^T\mathbb{1} = \mathbb{1} \quad A_2^T\mathbb{1} = \mathbb{1}. \tag{44}$$

Each node k in the first combination step fuses $\{\boldsymbol{w}_{\ell,i-1}\}_{\ell\in\mathcal{N}_k}$ in a convex manner to generate $\boldsymbol{\phi}_{k,i-1}$. In the following step, named adaptation, each node k uses its own data and that of neighboring nodes, i.e., $\{\boldsymbol{u}_{\ell,i}, \boldsymbol{d}_\ell(i)\}_{\ell\in\mathcal{N}_k}$ to adaptively update $\boldsymbol{\phi}_{k,i-1}$ to an intermediate estimate $\boldsymbol{\psi}_{k,i}$. In the third step, which is also a combination, the intermediate estimates $\{\boldsymbol{\psi}_{\ell,i}\}_{\ell\in\mathcal{N}_k}$ are fused to further align the global parameter estimate at node k to that of its neighbors. Subsequently, the desired space-varying parameter $\boldsymbol{h}_{k,i}$ is obtained from $\boldsymbol{w}_{k,i}$. Note that each step in the algorithm runs concurrently over the network.

Remark 3: The main difference between Algorithm 2 and the previously developed diffusion LMS strategies in, e.g., [2], [6], [35] is in the transformed domain regression data $\boldsymbol{u}_{\ell,i}B_\ell$ in (40) which now have singular covariance matrices. Moreover, there is an additional interpolation step (42). ■

Remark 4: The proposed diffusion LMS algorithm estimates NM spatially dependent variables $\{h_k^o\}$ using N_bM global invariant coefficients in w^o. From the computational complexity and energy efficiency point of view, it seems this is advantageous when the number of nodes, N, is greater than the number of basis functions N_b. However, even if this is not the case, using the estimated N_bM global invariant coefficients, the algorithm not only can estimate the space-varying parameters at the locations of the N nodes, but can also estimate the space-varying parameters at locations where no measurements are available. Therefore, even when $N < N_b$, the algorithm is still useful as it can perform interpolation. ■

There are different choices for the combination matrices $\{A_1, A_2, C\}$. For example, the choice $A_1 = A_2 = C = I$ reduces the above diffusion algorithm to the non-cooperative case where each node runs an individual LMS filter without coordination with its neighbors. Selecting $C = I$ simplifies the adaptation step (40) to the case where node k uses only its own data $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$ to perform local adaptation. Choosing $A_1 = I$ and $A_2 = A$, for some left-stochastic matrix A, removes the first combination step and the algorithm reduces to an adaptation step followed by combination (this variant of the algorithm has the Adapt-then-Combine or ATC diffusion structure) [2], [6]. Likewise, choosing $A_1 = A$ and $A_2 = I$ removes the second combination step and the algorithm reduces to a combination step followed by adaptation (this variant has the Combine-then-Adapt (CTA) structure of diffusion [2], [6]). Often in practice, either the ATC or CTA version of the algorithm is used with C set to $C = I$ such as using the following ATC diffusion version described by equations (45)–(47).

Algorithm 3: Diffusion ATC

$$\boldsymbol{\psi}_{k,i} = \boldsymbol{w}_{k,i-1} + \mu_k B_k^T\boldsymbol{u}_{k,i}^T\big(\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}B_k\boldsymbol{w}_{k,i-1}\big) \tag{45}$$

$$\boldsymbol{w}_{k,i} = \sum_{\ell\in\mathcal{N}_k} a_{\ell,k}\boldsymbol{\psi}_{\ell,i} \tag{46}$$

$$\boldsymbol{h}_{k,i} = B_k\boldsymbol{w}_{k,i}. \tag{47}$$

Nevertheless for generality, we shall study the performance of Algorithm 2 for arbitrary matrices $\{A_1, A_2, C\}$ with C right-stochastic and $\{A_1, A_2\}$ left-stochastic. The results can then be specialized to various situations of interest, including ATC, CTA, and the non-cooperative case.

The combination matrices $\{A_1, A_2, C\}$ are normally obtained using some well-known available combination rules such as the Metropolis or uniform combination rules [2], [24], [35]. These matrices can also be treated as free variables in the optimization procedure and used to further enhance the performance of the diffusion strategies. Depending on the network topology and the quality of the communication links between nodes, the optimized values of the combination matrices differ from one case to another [6], [46]–[48].

IV. Performance Analysis

In this section, we analyze the performance of the diffusion strategy (39)–(42) in the mean and mean-square sense and derive expressions to characterize the network mean-square deviation (MSD) and excess mean-square error (EMSE). In the analysis, we need to consider the fact that the covariance matrices $\{\bar{R}_{u,k}\}_{k=1}^{N}$ defined in (32) are now rank-deficient since we have $N_b > 1$. We explain in the sequel the ramifications that follow from this rank-deficiency.

A. Mean Convergence

We introduce the local weight-error vectors:

$$\tilde{\boldsymbol{w}}_{k,i} \triangleq w^o - \boldsymbol{w}_{k,i},\; \tilde{\boldsymbol{\psi}}_{k,i} \triangleq w^o - \boldsymbol{\psi}_{k,i},\; \tilde{\boldsymbol{\phi}}_{k,i} \triangleq w^o - \boldsymbol{\phi}_{k,i} \quad (48)$$

and define the network error vectors:

$$\tilde{\boldsymbol{\phi}}_i \triangleq \mathrm{col}\{\tilde{\boldsymbol{\phi}}_{1,i}, \ldots, \tilde{\boldsymbol{\phi}}_{N,i}\} \quad (49)$$

$$\tilde{\boldsymbol{\psi}}_i \triangleq \mathrm{col}\{\tilde{\boldsymbol{\psi}}_{1,i}, \ldots, \tilde{\boldsymbol{\psi}}_{N,i}\} \quad (50)$$

$$\tilde{\boldsymbol{w}}_i \triangleq \mathrm{col}\{\tilde{\boldsymbol{w}}_{1,i}, \ldots, \tilde{\boldsymbol{w}}_{N,i}\}. \quad (51)$$

We collect the estimates from across the network into the block vector:

$$\boldsymbol{w}_i \triangleq \mathrm{col}\{\boldsymbol{w}_{1,i}, \ldots, \boldsymbol{w}_{N,i}\} \quad (52)$$

and introduce the following extended combination matrices:

$$\mathcal{A}_1 \triangleq A_1 \otimes I_{MN_b} \quad (53)$$

$$\mathcal{A}_2 \triangleq A_2 \otimes I_{MN_b} \quad (54)$$

$$\mathcal{C} \triangleq C \otimes I_{MN_b}. \quad (55)$$

We further define the block diagonal matrices and vectors:

$$\boldsymbol{\mathcal{R}}_i \triangleq \mathrm{diag}\Big\{ \sum_{\ell \in \mathcal{N}_k} c_{\ell,k} B_\ell^T \boldsymbol{u}_{\ell,i}^T \boldsymbol{u}_{\ell,i} B_\ell : k = 1, \ldots, N \Big\} \quad (56)$$

$$\mathcal{M} \triangleq \mathrm{diag}\{\mu_1 I_{MN_b}, \ldots, \mu_N I_{MN_b}\} \quad (57)$$

$$\boldsymbol{t}_i \triangleq \mathrm{col}\Big\{ \sum_{\ell \in \mathcal{N}_k} c_{\ell,k} B_\ell^T \boldsymbol{u}_{\ell,i}^T \boldsymbol{d}_\ell(i) : k = 1, \ldots, N \Big\} \quad (58)$$

$$\boldsymbol{g}_i \triangleq \mathcal{C}^T \,\mathrm{col}\{B_1^T \boldsymbol{u}_{1,i}^T \boldsymbol{V}_1(i), \ldots, B_N^T \boldsymbol{u}_{N,i}^T \boldsymbol{v}_N(i)\} \quad (59)$$

and introduce the expected values of $\boldsymbol{\mathcal{R}}_i$ and $\boldsymbol{t}_i$:

$$\mathcal{R} \triangleq \mathbb{E}[\boldsymbol{\mathcal{R}}_i] = \mathrm{diag}\{R_1, \ldots, R_N\} \quad (60)$$

$$r \triangleq \mathbb{E}[\boldsymbol{t}_i] = \mathrm{col}\{r_1, \ldots, r_N\} \quad (61)$$

where

$$R_k \triangleq \sum_{\ell \in \mathcal{N}_k} c_{\ell,k}\, \bar{R}_{u,\ell} \quad (62)$$

$$r_k \triangleq \sum_{\ell \in \mathcal{N}_k} c_{\ell,k}\, \bar{r}_{du,\ell}. \quad (63)$$

We also introduce an indicator matrix operator, denoted by $\mathrm{Ind}(\cdot)$, such that for any real-valued matrix X with (k,j)-th entry $X_{k,j}$, the corresponding entry of $Y = \mathrm{Ind}(X)$ is:

$$Y_{k,j} = \begin{cases} 1, & \text{if } X_{k,j} > 0 \\ 0, & \text{otherwise} \end{cases}. \quad (64)$$

Now from (39)–(41), we obtain:

$$\boldsymbol{w}_i = \boldsymbol{\mathcal{B}}_i \boldsymbol{w}_{i-1} + \mathcal{A}_2^T \mathcal{M} \boldsymbol{t}_i \quad (65)$$

where

$$\boldsymbol{\mathcal{B}}_i \triangleq \mathcal{A}_2^T (I - \mathcal{M}\boldsymbol{\mathcal{R}}_i) \mathcal{A}_1^T. \quad (66)$$

In turn, making use of (27) in (65), we can verify that the network error vector follows the recursion

$$\tilde{\boldsymbol{w}}_i = \boldsymbol{\mathcal{B}}_i \tilde{\boldsymbol{w}}_{i-1} - \mathcal{A}_2^T \mathcal{M} \boldsymbol{g}_i. \quad (67)$$

By taking the expectation of both sides of (67) and using Assumption 1, we arrive at:

$$\mathbb{E}[\tilde{\boldsymbol{w}}_i] = \mathcal{B}\, \mathbb{E}[\tilde{\boldsymbol{w}}_{i-1}] \quad (68)$$

where in this relation:

$$\mathcal{B} \triangleq \mathbb{E}[\boldsymbol{\mathcal{B}}_i] = \mathcal{A}_2^T (I - \mathcal{M}\mathcal{R}) \mathcal{A}_1^T. \quad (69)$$

To obtain (68), we used the fact that the expectation of the second term in (67), i.e., $\mathbb{E}[\mathcal{A}_2^T \mathcal{M} \boldsymbol{g}_i]$, is zero because $\boldsymbol{v}_k(i)$ is independent of $\boldsymbol{u}_{k,i}$ and $\mathbb{E}[\boldsymbol{v}_k(i)] = 0$. The rank-deficient matrices $\{\bar{R}_{u,k}\}$ appear inside $\mathcal{R}$ in (69). We now verify that despite having rank-deficient matrix $\mathcal{R}$, recursion (68) still guarantees a bounded mean error vector in steady-state.

To proceed, we introduce the eigendecomposition:

$$R_k = Q_k \Lambda_k Q_k^T \quad (70)$$

where $Q_k = [q_{k,1}, \ldots, q_{k,MN_b}]$ is a unitary matrix with column eigenvectors $q_{k,j}$ and $\Lambda_k = \mathrm{diag}\{\lambda_k(1), \ldots, \lambda_k(MN_b)\}$ is a diagonal matrix with eigenvalues $\lambda_k(j) \geq 0$. For this decomposition, we assume that the eigenvalues of R_k are arranged in descending order, i.e, $\lambda_{\max}(R_k) \triangleq \lambda_k(1) \geq \lambda_k(2) \geq \cdots \geq \lambda_k(MN_b)$, and the rank of R_k is $L_k \leq MN_b$. If we define $\mathcal{Q} \triangleq \mathrm{diag}\{Q_1, \ldots, Q_N\}$ and $\Lambda \triangleq \mathrm{diag}\{\Lambda_1, \ldots, \Lambda_N\}$, then the network covariance matrix, $\mathcal{R}$, given by (60) can be expressed as:

$$\mathcal{R} = \mathcal{Q} \Lambda \mathcal{Q}^T. \quad (71)$$

We now note that the mean error vector, $\mathbb{E}[\tilde{\boldsymbol{w}}_i]$, expressed by (68) will be asymptotically unbiased if the spectral radius of $\mathcal{B}$, denoted by $\rho(\mathcal{B})$, is strictly less than one. Let us examine under what conditions this requirement is satisfied. Since A_1 and A_2

are left-stochastic matrices and $\mathcal{R}$ is block-diagonal, we have from [6] that:

$$\rho(\mathcal{B}) = \rho\Big(\mathcal{A}_2^T(I-\mathcal{M}\mathcal{R})\mathcal{A}_1^T\Big) \le \rho\big(I-\mathcal{M}\mathcal{R}\big). \tag{72}$$

Therefore, if $\mathcal{R}$ is positive-definite, then choosing $\mu_k < 2/\lambda_{\max}(R_k)$ ensures convergence of the algorithm in the mean so that $\mathbb{E}[\tilde{\boldsymbol{w}}_i] \to 0$ as $i \to \infty$. However, when $\mathcal{R}$ is singular, it may hold that $\rho(\mathcal{B}) = 1$, in which case choosing the step-sizes according to the above bound guarantees the boundedness of the mean error, $\mathbb{E}[\tilde{\boldsymbol{w}}_i]$, but not necessarily that it converges to zero. The following result clarifies these observations.

Theorem 1: If the step-sizes are chosen to satisfy

$$0 < \mu_k < \frac{2}{\lambda_{\max}(R_k)} \tag{73}$$

then, under Assumption 1, the diffusion algorithm is stable in the mean in the following sense: (a) If $\rho(\mathcal{B}) < 1$, then $\mathbb{E}[\tilde{\boldsymbol{w}}_i]$ converges to zero and (b) if $\rho(\mathcal{B}) = 1$ then

$$\lim_{i\to\infty} \big\|\mathbb{E}[\tilde{\boldsymbol{w}}_i]\big\|_{b,\infty} \le \|I - \mathrm{Ind}(\Lambda)\|_{b,\infty} \big\|\mathbb{E}[\tilde{\boldsymbol{w}}_{-1}]\big\|_{b,\infty} \tag{74}$$

where $\|\cdot\|_{b,\infty}$ stands for the block-maximum norm, as defined in [6], [47].

Proof: See Appendix A. ■

In what follows, we examine recursion (65) and derive an expression for the asymptotic value of $\mathbb{E}[\boldsymbol{w}_i]$—see (89) further ahead. Before doing so, we first comment on a special case of interest, namely, result (76) below.

Special case: Consider a network with $A_1 = A_2 = I$ and an arbitrary right stochastic matrix C satisfying (37). Using (27) and (62)–(63), it can be verified that the following linear system of equations holds at each node k:

$$R_k w^o = r_k. \tag{75}$$

We show in Appendix B that under condition (73) the mean estimate of the diffusion LMS algorithm at each node k will converge to:

$$\lim_{i\to\infty} \mathbb{E}[\boldsymbol{w}_{k,i}] = R_k^{\dagger} r_k + \sum_{n=L_k+1}^{MN_b} q_{k,n} q_{k,n}^T \mathbb{E}[\boldsymbol{w}_{k,-1}] \tag{76}$$

where $R_k^{\dagger}$ represents the pseudo-inverse of R_k, and $\boldsymbol{w}_{k,-1}$ is the node initial value. This result is consistent with the mean estimate of the stand-alone LMS filter with rank-deficient input data (which corresponds to the situation $A_1 = A_2 = C = I$) [49]. Note that $R_k^{\dagger} r_k$ in (76) corresponds to the minimum-norm solution of $R_k w = r_k$. Therefore, the second term on the right hand side of (76) is the deviation of the node estimate from this minimum-norm solution. The presence of this term after convergence is due to the zero eigenvalues of R_k. If R_k were full-rank so that $L_k = MN_b$, then this term would disappear and the node estimate will converge, in the mean, to its optimal value, w^o. We point out that even though the matrices $\bar{R}_{u,\ell}$ are rank deficient since $N_b > 1$, it is still possible for the matrices R_k to be full rank owing to the linear combination operation in (62). This illustrates one of the benefits of employing the right-stochastic matrix C. However, if despite using C, R_k still remains rank-deficient, the second term on the right-hand side of (76) can be annihilated by proper node initialization (e.g., by setting $\mathbb{E}[\boldsymbol{w}_{k,-1}] = 0$). By doing so, the mean estimate of each node will then approach the unique minimum-norm solution, $R_k^{\dagger} r_k$.

General case: Let us now find the mean estimate of the network for arbitrary left-stochastic matrices A_1 and A_2. Considering definitions (60)–(61) and relation (75) and noting that $\mathcal{A}_1^T(\mathbb{1}\otimes w^o) = \mathcal{A}_2^T(\mathbb{1}\otimes w^o) = (\mathbb{1}\otimes w^o)$, it can be verified that $(\mathbb{1}\otimes w^o)$ satisfies the following linear system of equations:

$$(I-\mathcal{B})(\mathbb{1}\otimes w^o) = \mathcal{A}_2^T \mathcal{M} r. \tag{77}$$

This is a useful intermediate result that will be applied in our argument.

Next, if we iterate recursion (65) and apply the expectation operator, we then obtain

$$\mathbb{E}[\boldsymbol{w}_i] = \mathcal{B}^{i+1}\mathbb{E}[\boldsymbol{w}_{-1}] + \sum_{j=0}^{i} \mathcal{B}^j \mathcal{A}_2^T \mathcal{M} r. \tag{78}$$

The mean estimate of the network can be found by computing the limit of this expression for $i \to \infty$. To find the limit of the first term on the right hand side of (78), we evaluate $\lim_{i\to\infty}\mathcal{B}^i$ and find conditions under which it converges. For this purpose, we introduce the Jordan decomposition of matrix $\mathcal{B}$ as [50]:

$$\mathcal{B} = \mathcal{Z}\Gamma\mathcal{Z}^{-1} \tag{79}$$

where $\mathcal{Z}$ is an invertible matrix, and Γ is a block diagonal matrix of the form

$$\Gamma = \mathrm{diag}\Big\{\Gamma_1, \Gamma_2, \ldots, \Gamma_s\Big\} \tag{80}$$

where the l-th Jordan block, $\Gamma_l \in \mathbb{C}^{m_l \times m_l}$, can be expressed as:

$$\Gamma_l = \gamma_l I_{m_l} + N_{m_l}. \tag{81}$$

In this relation, N_{m_l} is some nilpotent matrix of size $m_l \times m_l$. Using decomposition (79), we can express $\mathcal{B}^i$ as

$$\mathcal{B}^i = \mathcal{Z}\Gamma^i\mathcal{Z}^{-1}. \tag{82}$$

Since Γ is block diagonal, we have

$$\Gamma^i = \mathrm{diag}\Big\{\Gamma_1^i, \Gamma_2^i, \ldots, \Gamma_s^i\Big\}. \tag{83}$$

From this relation, it is deduced that $\lim_{i\to\infty}\mathcal{B}^i$ exists if $\lim_{i\to\infty}\Gamma_l^i$ exists for all $l \in \{1,\ldots,s\}$. Using (81), we can write [50]:

$$\lim_{i\to\infty}\Gamma_l^i = \lim_{i\to\infty}\gamma_l^{i-m_l}\left(\gamma_l^{m_l} I_{m_l} + \sum_{p=1}^{m_l-1}\binom{i}{p}\gamma_l^{m_l-p}N_{m_l}^p\right). \tag{84}$$

When $i \to \infty$, $\gamma_l^{i-m_l}$ becomes the dominant factor in this expression. Note that under condition (73), we have $\rho(\mathcal{B}) \le 1$ which in turn implies that the magnitude of the eigenvalues of $\mathcal{B}$ are bounded as $0 \le |\gamma_n| \le 1$. Without loss of generality, we assume that the eigenvalues of $\mathcal{B}$ are arranged as $|\gamma_1| \le \cdots \le |\gamma_L| < |\gamma_{L+1}| = \cdots = |\gamma_s| = 1$. Now we examine the limit (84) for every $|\gamma_l|$ in this range. Clearly for $|\gamma_l| < 1$, the limit is zero (an obvious conclusion since in this case Γ_l is a stable matrix). For $|\gamma_l| = 1$, the limit is the identity matrix if $\gamma_l = 1$ and $m_l = 1$. However, the limit does not exist for unit magnitude complex eigenvalues and eigenvalues with value -1, even when $m_l = 1$. Motivated by these observations, we introduce the following definition.

Definition: We refer to matrix $\mathcal{B}$ as *power convergent* if (a) its eigenvalues γ_n satisfy $0 \leq |\gamma_n| \leq 1$, (b) its unit magnitude eigenvalues are all equal to one, and (c) its Jordan blocks associated with $\gamma_n = 1$ are all of size 1×1. ■

1) Example 1: Assume $N_b = 1$, $B_k = I_M$, and uniform step-sizes and covariance matrices across the agents, i.e., $\mu_k \equiv \mu$, $R_{u,k} \equiv R_u$ for all k. Assume further that C is doubly-stochastic (i.e., $C^T\mathbb{1} = \mathbb{1} = C\mathbb{1}$) and R_u is singular. Then, in this case, the matrix $\mathcal{B}$ can be written as the Kronecker product $\mathcal{B} = A_2^T A_1^T \otimes (I_M - \mu R_u)$. For strongly-connected networks where $A_1 A_2$ is a primitive matrix, it follows from the Perron-Frobenius Theorem [51] that $A_1 A_2$ has a single unit-magnitude eigenvalue at one, while all other eigenvalues have magnitude less than one. We conclude in this case, from the properties of Kronecker products and under condition (73), that $\mathcal{B}$ is a power-convergent matrix. ■

2) Example 2: Assume $M = 2$, $N = 3$, $N_b = 1$, $B_k = I_M$, and uniform step-sizes and covariance matrices across the agents again. Let $A_2 = I = C$ and select

$$A_1 = A = \begin{bmatrix} 1/2 & 0 & 0 \\ 1/2 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \tag{85}$$

which is not primitive. Let further $R_u = \mathrm{diag}\{\beta, 0\}$ denote a singular covariance matrix. Then, it can be verified in this case the corresponding matrix $\mathcal{B}$ will have an eigenvalue with value -1 and is not power convergent. ■

Returning to the above definition and assuming $\mathcal{B}$ is power convergent, then this means that the Jordan decomposition (79) can be rewritten as:

$$\mathcal{B} = \underbrace{[\mathcal{Z}_1\ \mathcal{Z}_2]}_{\mathcal{Z}} \underbrace{\begin{bmatrix} J & 0 \\ 0 & I \end{bmatrix}}_{\Gamma} \underbrace{\begin{bmatrix} \bar{\mathcal{Z}}_1 \\ \bar{\mathcal{Z}}_2 \end{bmatrix}}_{\mathcal{Z}^{-1}} \tag{86}$$

where J is a Jordan matrix with all eigenvalues strictly inside the unit circle, and the identity matrix inside Γ accounts for the eigenvalues with value one. In (86) we further partition $\mathcal{Z}$ and $\mathcal{Z}^{-1}$ in accordance with the size of J. Using (86), it is straightforward to verify that

$$\lim_{i\to\infty} \mathcal{B}^{i+1} = \mathcal{Z}_2\bar{\mathcal{Z}}_2 \tag{87}$$

and if we multiply both sides of (77) from the left by $\bar{\mathcal{Z}}_2$, it also follows that

$$\bar{\mathcal{Z}}_2\mathcal{A}_2^T\mathcal{M}r = 0. \tag{88}$$

Using these relations, we can now establish the following result, which describes the limiting behavior of the weight vector estimate.

Theorem 2: If the step-sizes $\{\mu_1, \ldots, \mu_N\}$ satisfy (73) and matrix $\mathcal{B}$ is power convergent, then the mean estimate of the network given by (78) asymptotically converges to:

$$\lim_{i\to\infty} \mathbb{E}[\boldsymbol{w}_i] = (\mathcal{Z}_2\bar{\mathcal{Z}}_2)\,\mathbb{E}[\boldsymbol{w}_{-1}] + (I - \mathcal{B})^{-}\mathcal{A}_2^T\mathcal{M}r \tag{89}$$

where the notation X^- denotes a (reflexive) generalized inverse for the matrix X. In this case, the generalized inverse for $I - \mathcal{B}$ is given by

$$(I - \mathcal{B})^{-} = \mathcal{Z}_1(I - J)^{-1}\bar{\mathcal{Z}}_1 \tag{90}$$

which is in terms of the factors $\{\mathcal{Z}_1, \bar{\mathcal{Z}}_1, J\}$ defined in (86).

Proof: See Appendix C. ■

We also argue in Appendix C that the quantity on the right-hand side of (89) is invariant under basis transformations for the Jordan factors $\{\mathcal{Z}_1, \bar{\mathcal{Z}}_1, \mathcal{Z}_2, \bar{\mathcal{Z}}_2\}$. It can be verified that if $A_1 = A_2 = I$ then $\mathcal{B}$ will be symmetric and the result (89) will reduce to (76). Now note that the first term on the right hand side of (89) is due to the zero eigenvalues of $I - \mathcal{B}$. From this expression, we observe that different initialization values generally lead to different estimates. However, if we set $\mathbb{E}[\boldsymbol{w}_{-1}] = 0$, the algorithm converges to:

$$\lim_{i\to\infty} \mathbb{E}[\boldsymbol{w}_i] = (I - \mathcal{B})^{-}\mathcal{A}_2^T\mathcal{M}r. \tag{91}$$

In other words, the diffusion LMS algorithm will converge on average to a generalized inverse solution of the linear system of equations defined by (77).

When matrix $\mathcal{B}$ is stable so that $\rho(\mathcal{B}) < 1$ then the factorization (86) reduces to the form $\mathcal{B} = \mathcal{Z}_1 J\bar{\mathcal{Z}}_1$ and $I - \mathcal{B}$ will be full-rank. In that case, the first term on the right hand side of (89) will be zero and the generalized inverse will coincide with the actual matrix inverse so that (89) becomes

$$\lim_{i\to\infty} \mathbb{E}[\boldsymbol{w}_i] = (I - \mathcal{B})^{-1}\mathcal{A}_2^T\mathcal{M}r. \tag{92}$$

Comparing (92) with (77), we conclude that:

$$\lim_{i\to\infty} \mathbb{E}[\boldsymbol{w}_i] = \mathbb{1} \otimes w^o \tag{93}$$

which implies that the mean estimate of each node will be w^o. This result is in agreement with the previously developed mean-convergence analysis of diffusion LMS when the regression data have full rank covariance matrices [6].

B. Mean-Square Error Convergence

We now examine the mean-square stability of the error recursion (67) in the rank-deficient scenario. We begin by deriving an error variance relation as in [52], [53]. To find this relation, we form the weighted square "norm" of (67), and compute its expectation to obtain:

$$\mathbb{E}\|\tilde{\boldsymbol{w}}_i\|_\Sigma^2 = \mathbb{E}\left(\|\tilde{\boldsymbol{w}}_{i-1}\|_{\Sigma'}^2\right) + \mathbb{E}[\boldsymbol{g}_i^T\mathcal{M}\mathcal{A}_2\Sigma\mathcal{A}_2^T\mathcal{M}\boldsymbol{g}_i] \tag{94}$$

where $\|x\|_\Sigma^2 = x^T\Sigma x$ and $\Sigma \geq 0$ is an arbitrary weighting matrix of compatible dimension that we are free to choose. In this expression,

$$\boldsymbol{\Sigma}' = \mathcal{A}_1(I - \mathcal{M}\boldsymbol{\mathcal{R}}_i)^T\mathcal{A}_2\Sigma\mathcal{A}_2^T(I - \mathcal{M}\boldsymbol{\mathcal{R}}_i)\mathcal{A}_1^T. \tag{95}$$

Under the temporal and spatial independence conditions on the regression data from Assumption 1, we can write:

$$\mathbb{E}\big(\|\tilde{\boldsymbol{w}}_{i-1}\|^2_{\boldsymbol{\Sigma}'}\big) = \mathbb{E}\|\tilde{\boldsymbol{w}}_{i-1}\|^2_{\mathbb{E}[\boldsymbol{\Sigma}']} \tag{96}$$

so that (94) becomes:

$$\mathbb{E}\|\tilde{\boldsymbol{w}}_i\|^2_{\Sigma} = \mathbb{E}\|\tilde{\boldsymbol{w}}_{i-1}\|^2_{\Sigma'} + \mathrm{Tr}[\Sigma\mathcal{A}_2^T\mathcal{M}\mathcal{G}\mathcal{M}\mathcal{A}_2] \tag{97}$$

where $\mathcal{G} \triangleq \mathbb{E}[\boldsymbol{g}_i\boldsymbol{g}_i^T]$ is given by

$$\mathcal{G} = \mathcal{C}^T\mathrm{diag}\{\sigma^2_{v,1}\bar{R}_{u,1}, \ldots, \sigma^2_{v,N}\bar{R}_{u,N}\}\mathcal{C} \tag{98}$$

and

$$\Sigma' \triangleq \mathbb{E}[\boldsymbol{\Sigma}'] = \mathcal{B}^T\Sigma\mathcal{B} + O(\mathcal{M}^2) \approx \mathcal{B}^T\Sigma\mathcal{B}. \tag{99}$$

We shall employ (99) under the assumption of sufficiently small step-sizes where terms that depend on higher-order powers of the step-sizes are ignored. We next introduce

$$\mathcal{Y} \triangleq \mathcal{A}_2^T\mathcal{M}\mathcal{G}\mathcal{M}\mathcal{A}_2 \tag{100}$$

and use (97) to write:

$$\mathbb{E}\|\tilde{\boldsymbol{w}}_i\|^2_{\Sigma} = \mathbb{E}\|\tilde{\boldsymbol{w}}_{i-1}\|^2_{\Sigma'} + \mathrm{Tr}(\Sigma\mathcal{Y}). \tag{101}$$

From (101), we arrive at

$$\mathbb{E}\|\tilde{\boldsymbol{w}}_i\|^2_{\Sigma} = \mathbb{E}\|\tilde{\boldsymbol{w}}_{-1}\|^2_{(\mathcal{B}^T)^{i+1}\Sigma\mathcal{B}^{i+1}} + \sum_{j=0}^{i}\mathrm{Tr}\Big((\mathcal{B}^T)^j\Sigma\mathcal{B}^j\mathcal{Y}\Big). \tag{102}$$

To prove the convergence and stability of the algorithm in the mean-square sense, we examine the convergence of the terms on the right hand side of (102).

In a manner similar to (88), it is shown in Appendix D that the following property holds:

$$\bar{\mathcal{Z}}_2\mathcal{Y} = 0, \qquad \mathcal{Y}\bar{\mathcal{Z}}_2^T = 0. \tag{103}$$

Exploiting this result, we can arrive at the following statement, which establishes that relation (102) converges as $i \to \infty$ and determines its limiting value.

Theorem 3: Assume the step-sizes are sufficiently small and satisfy (73). Assume also that $\mathcal{B}$ is power convergent. Under these conditions, relation (102) converges to

$$\lim_{i\to\infty}\mathbb{E}\|\tilde{\boldsymbol{w}}_i\|^2_{\Sigma} = \mathbb{E}\|\tilde{\boldsymbol{w}}_{-1}\|^2_{(\mathcal{Z}_2\bar{\mathcal{Z}}_2)^T\Sigma\mathcal{Z}_2\bar{\mathcal{Z}}_2} + \big(\mathrm{vec}(\mathcal{Y})\big)^T(I-\mathcal{F})^{-1}\mathrm{vec}(\Sigma) \tag{104}$$

where

$$\mathcal{F} \triangleq \big((\mathcal{Z}_1\otimes\mathcal{Z}_1)(J\otimes J)(\bar{\mathcal{Z}}_1\otimes\bar{\mathcal{Z}}_1)\big)^T \tag{105}$$

and factors $\{\mathcal{Z}_1, \bar{\mathcal{Z}}_1, J\}$ are defined in (86).

Proof: See Appendix D. ■

In a manner similar to the proof at the end of Appendix C, the term on the right hand side of (104) is invariant under basis transformations on the factors $\{\mathcal{Z}_1, \bar{\mathcal{Z}}_1, \mathcal{Z}_2, \bar{\mathcal{Z}}_2\}$. Note that the first term on the right hand side of (104) is the network penalty due to rank-deficiency. When the node covariance matrices are full rank, then choosing step-sizes according to (73) leads to $\rho(\mathcal{B}) < 1$. When this holds, then $\mathcal{B} = \mathcal{Z}_1 J\bar{\mathcal{Z}}_1$. In this case, the first term on the right hand side of (104) will be zero, and $\mathcal{F} = (\mathcal{B}\otimes\mathcal{B})^T$. In this case, we obtain:

$$\lim_{i\to\infty}\mathbb{E}\|\tilde{\boldsymbol{w}}_i\|^2_{\Sigma} = \big(\mathrm{vec}(\mathcal{Y})\big)^T(I-\mathcal{F})^{-1}\mathrm{vec}(\Sigma) \tag{106}$$

which is in agreement with the mean-square analysis of diffusion LMS strategies for regression data with full rank covariance matrices given in [2], [6].

C. Learning Curves

For each k, the MSD and EMSE measures are defined as:

$$\eta_k = \lim_{i\to\infty}\mathbb{E}\|\tilde{\boldsymbol{h}}_{k,i}\|^2 = \lim_{i\to\infty}\mathbb{E}\|\tilde{\boldsymbol{w}}_{k,i}\|^2_{B_k^TB_k} \tag{107}$$

$$\zeta_k = \lim_{i\to\infty}\mathbb{E}\|\boldsymbol{u}_{k,i}\tilde{\boldsymbol{h}}_{k,i-1}\|^2 = \lim_{i\to\infty}\mathbb{E}\|\tilde{\boldsymbol{w}}_{k,i-1}\|^2_{\bar{R}_{u,k}} \tag{108}$$

where $\tilde{\boldsymbol{h}}_{k,i} = h_k^o - \boldsymbol{h}_{k,i}$. These parameters can be computed from the network error vector (104) through proper selection of the weighting matrix Σ as follows:

$$\begin{aligned}\eta_k &= \lim_{i\to\infty}\mathbb{E}\|\tilde{\boldsymbol{w}}_i\|^2_{\Sigma_{\mathrm{msd}_k}},\\ \zeta_k &= \lim_{i\to\infty}\mathbb{E}\|\tilde{\boldsymbol{w}}_{i-1}\|^2_{\Sigma_{\mathrm{emse}_k}},\end{aligned} \tag{109}$$

where

$$\begin{aligned}\Sigma_{\mathrm{msd}_k} &= \mathrm{diag}(e_k)\otimes(B_k^TB_k),\\ \Sigma_{\mathrm{emse}_k} &= \mathrm{diag}(e_k)\otimes\bar{R}_{u,k}\end{aligned} \tag{110}$$

and $\{e_k\}_{k=1}^N$ denote the vectors of a canonical basis set in N dimensional space. The network MSD and EMSE measures are defined as

$$\eta_{\mathrm{net}} = \frac{1}{N}\sum_{k=1}^{N}\eta_k, \qquad \zeta_{\mathrm{net}} = \frac{1}{N}\sum_{k=1}^{N}\zeta_k. \tag{111}$$

We can also define MSD and EMSE measures over time as

$$\eta_k(i) = \mathbb{E}\|\tilde{\boldsymbol{h}}_{k,i}\|^2 = \mathbb{E}\|\tilde{\boldsymbol{w}}_i\|^2_{\Sigma_{\mathrm{msd}_k}} \tag{112}$$

$$\zeta_k(i) = \mathbb{E}\|\boldsymbol{u}_{k,i}\tilde{\boldsymbol{h}}_{k,i-1}\|^2 = \mathbb{E}\|\tilde{\boldsymbol{w}}_{i-1}\|^2_{\Sigma_{\mathrm{emse}_k}}. \tag{113}$$

Using (102), it can be verified that these measures evolve according to the following dynamics:

$$\eta_k(i) = \eta_k(i-1) - \|w^o\|_{\mathcal{H}^i(I-\mathcal{H})\sigma_{\mathrm{msd}_k}} + \alpha^T\mathcal{H}^i\sigma_{\mathrm{msd}_k} \tag{114}$$

$$\zeta_k(i) = \zeta_k(i-1) - \|w^o\|_{\mathcal{H}^i(I-\mathcal{H})\sigma_{\mathrm{emse}_k}} + \alpha^T\mathcal{H}^i\sigma_{\mathrm{emse}_k} \tag{115}$$

where

$$\mathcal{H} = (\mathcal{B}\otimes\mathcal{B})^T \tag{116}$$

$$\alpha = \mathrm{vec}(\mathcal{Y}) \tag{117}$$

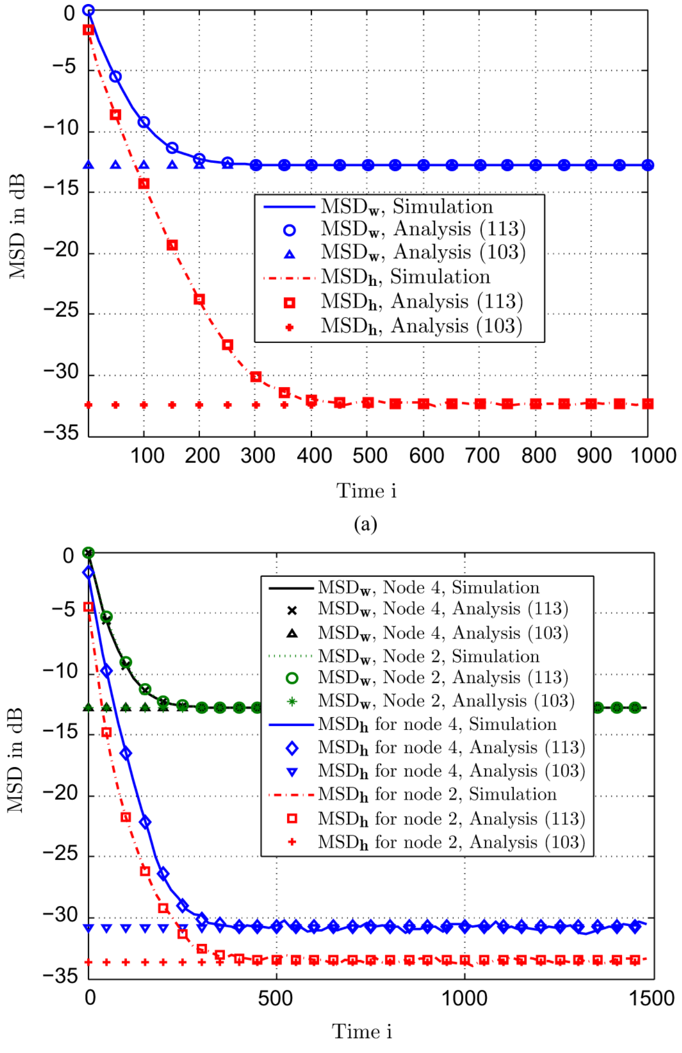
$$\sigma_{\mathrm{msd}_k} = \mathrm{vec}(\Sigma_{\mathrm{msd}_k}) \tag{118}$$

$$\sigma_{\mathrm{emse}_k} = \mathrm{vec}(\Sigma_{\mathrm{emse}_k}). \tag{119}$$

To obtain (114) and (115), we set $\mathbb{E}[\boldsymbol{w}_{k,-1}] = 0$ for all k.

V. Computer Experiments

In this section, we examine the performance of the diffusion strategy (39)–(42) and compare the simulation results with the analytical findings. In addition, we present a simulation example that shows the application of the proposed algorithm in the estimation of space-varying parameters for a physical phenomenon modeled by a PDE system over two spatial dimensions.



(a)

(b)

Fig. 2. The network MSD learning curve for $N = 4$ (a) The network MSD (b) The MSD at some individual nodes.

A. Performance of the Distributed Solution

We consider a one-dimensional network topology, illustrated by Fig. 1, with $L = 1$ and equally spaced nodes along the x direction. We choose A_1 as the identity matrix, and compute A_2 and C based on the uniform combination and Metropolis rules [2], [6], respectively. We choose $M = 2$ and $N_b = 5$ and generate the unknown global parameter w^o randomly for each experiment. We obtain B_k using the shifted Chebyshev polynomials given by (17) and compute the space varying parameters h_k^o according to (26). The measurement data $\boldsymbol{d}_k(i)$, $k \in \{1, 2, \ldots, N\}$ are generated using the regression model (12). The SNR for each node k is computed as $\text{SNR}_k = \mathbb{E}\|\boldsymbol{u}_{k,i} h_k^o\|^2/\sigma_{v,k}^2$. The noise and the entries of the regression data are white Gaussian and satisfy Assumption 1. The noise variances, $\{\sigma_{v,k}^2\}$, and the trace of the covariance matrices, $\{\text{Tr}(R_{u,k})\}$, are uniformly distributed between $[0.05, 0.1]$ and $[1, 5]$, respectively.

Fig. 2 illustrates the simulation results for a network with $N = 4$ nodes. For this experiment, we set $\mu_k = 0.01$ for all k and initialize each node at zero. In the legend of the figure, we use the subscript h to denote the MSD for $\tilde{\boldsymbol{h}}_{k,i}$ and the subscript w to refer to the MSD of $\tilde{\boldsymbol{w}}_{k,i}$. The simulation curves are obtained by averaging over 300 independent runs. It can be seen that the simulated and theoretical results match well in all cases. To obtain the analytical results, we use expression (104) to assess the steady-state values and expression (114) to generate the theoretical learning curves.

Two important points in Fig. 2 need to be highlighted. First, note from the top plot that the network MSD for $\tilde{\boldsymbol{w}}_{k,i}$ is larger than that for $\tilde{\boldsymbol{h}}_{k,i}$. This is because

$$\mathbb{E}\|\tilde{\boldsymbol{h}}_{k,i}\|^2 = \mathbb{E}\|\tilde{\boldsymbol{w}}_{k,i}\|^2_{B_k^T B_k} \tag{120}$$

so that the MSD of $\tilde{\boldsymbol{h}}_{k,i}$ is a weighted version of the MSD of $\tilde{\boldsymbol{w}}_{k,i}$. In this experiment, the weighting leads to a lower estimation error. Second, note from the bottom plot that while the MSD values of $\tilde{\boldsymbol{w}}_{k,i}$ are largely independent of the node index, the same is not true for the MSD values of $\tilde{\boldsymbol{h}}_{k,i}$. In previous studies on diffusion LMS strategies, it has been shown that, for strongly-connected networks, the network nodes approach a uniform MSD performance level [36]. The result in Fig. 2(b) supports this conclusion where it is seen that the MSD of $\tilde{\boldsymbol{w}}_{k,i}$ for nodes 2 and 4 converge to the same MSD level. However, note that the MSD of $\tilde{\boldsymbol{h}}_{k,i}$ is different for nodes 2 and 4. This difference in behavior is due to the difference in weighting across nodes from (120).

B. Comparison With Centralized Solution

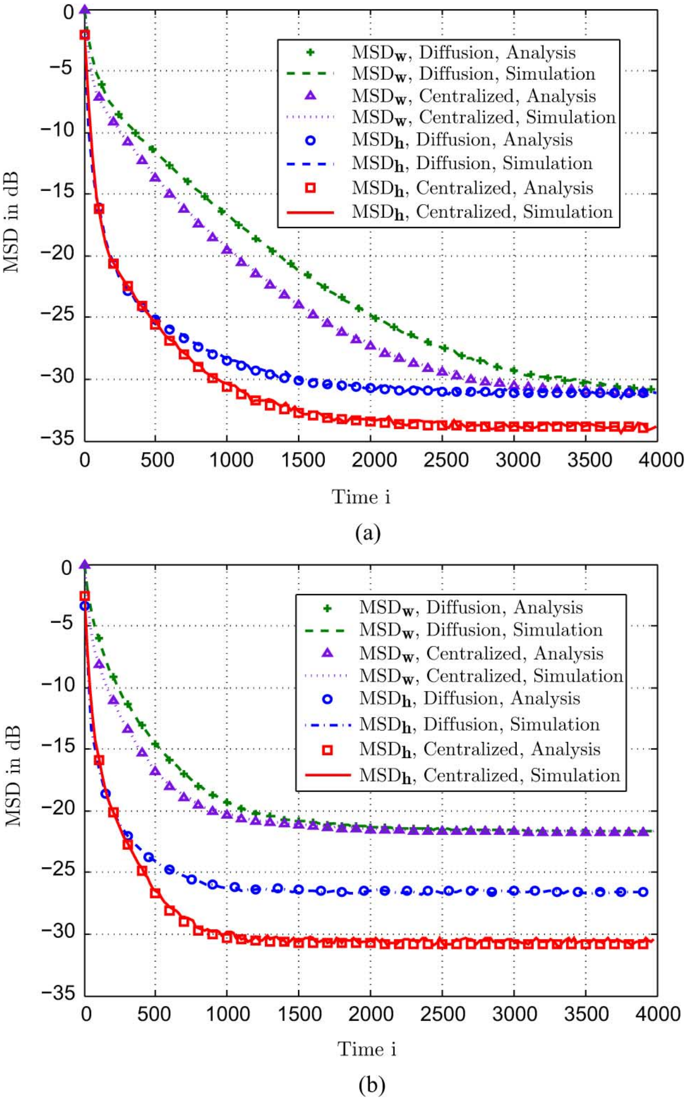
We next compare the performance of the diffusion strategy (34)–(35) with the centralized solution (39)–(42). We consider a network with $N = 10$ nodes with the topology illustrated by Fig. 1. In this experiment, we set $\mu_k = 0.02$ for all k, while the other network parameters are obtained following the same construction described for Fig. 2. As the results in Fig. 3 indicate, the diffusion and centralized LMS solutions tend to the same MSD performance level in the w domain. This conclusion is consistent with prior studies on the performance of diffusion strategies in the full-rank case over strongly-connected networks [36]. However, discrepancies in performance are seen between the distributed and centralized implementations in the h domain, and the discrepancy tends to become larger for larger values of N. This is because, in moving from the w domain to the h domain, the inherent aggregation of information that is performed by the centralized solution leads to enhanced estimates for the h_k^o variables. For example, if the estimates $\boldsymbol{w}_{k,i}$ which are generated by the distributed solution are averaged prior to computing the $\boldsymbol{h}_{k,i}$, then it can be observed that the MSD values of $\tilde{\boldsymbol{h}}_{k,i}$ for both the centralized and the distributed solution will be similar.

In these experiments, we also observe that if we increase the number of basis functions, N_b, then both the centralized and diffusion algorithms will converge faster but their steady-state MSD performance will degrade. Therefore, in choosing the number of basis functions, N_b, there is a trade off between convergence speed and MSD performance.

C. Example: Two-Dimensional Process Estimation

In this example, we consider a two-dimensional network with 13×13 nodes that are equally spaced over the unit square $(x, y) \in [0, 1] \times [0, 1]$ with $\Delta x = \Delta y = 1/12$ (see Fig. 4(a). This network monitors a physical process $f(x, y)$ described by the Poisson PDE:

$$\frac{\partial^2 f(x, y)}{\partial x^2} + \frac{\partial^2 f(x, y)}{\partial y^2} = h(x, y) \tag{121}$$



(a)

(b)

Fig. 3. The network MSD learning curve for $N = 10$ (a) $N_b = 5$ (b) $N_b = 10$.

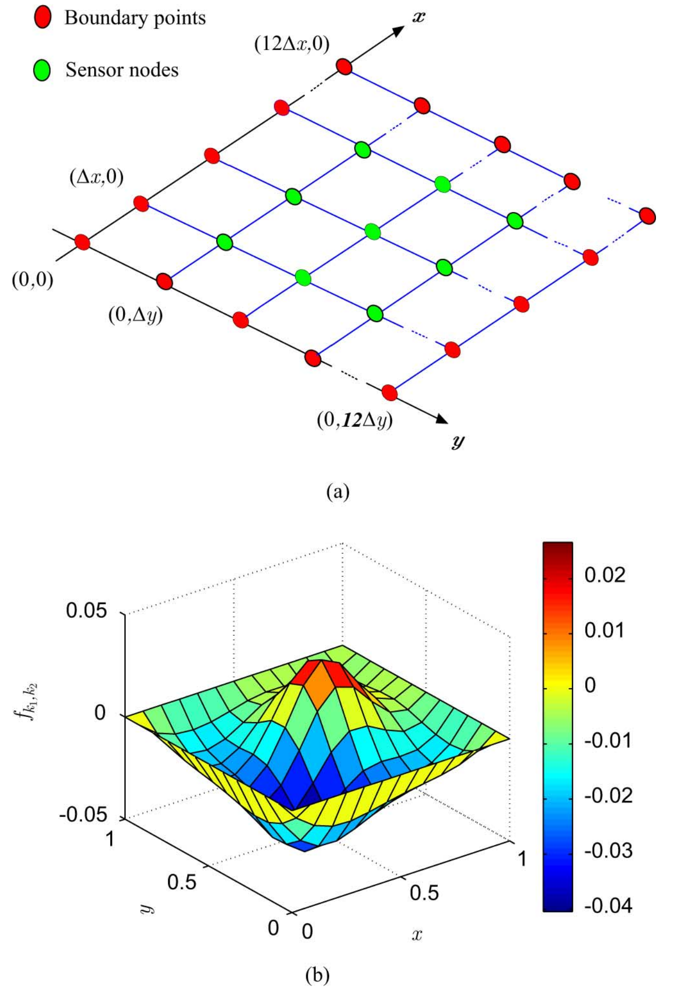
where $h(x, y) : [0, 1]^2 \rightarrow \mathbb{R}$ is an unknown input function. The PDE satisfies the following boundary conditions:

$$f(x, 0) = f(0, y) = f(x, 1) = f(1, y) = 0.$$

For this problem, the objective is to estimate $h(x, y)$, given noisy measurements collected by $N = N_x \times N_y = 11 \times 11$ nodes corresponding to the *interior points* of the network. To discretize the PDE, we employ the finite difference method (FDM) with uniform spacing of Δx and Δy. We define $x_{k_1} \triangleq k_1 \Delta x$, $y_{k_2} \triangleq k_2 \Delta y$ and introduce the sampled values $f_{k_1,k_2} \triangleq f(x_{k_1}, y_{k_2})$ and $h^o_{k_1,k_2} \triangleq h(x_{k_1}, y_{k_2})$. We use the central difference scheme [39] to approximate the second order partial derivatives:

$$\frac{\partial^2 f(x, y, t)}{\partial x^2} \approx \frac{1}{\Delta x^2}[f_{k_1+1,k_2} - 2f_{k_1,k_2} + f_{k_1-1,k_2}] \quad (122)$$

$$\frac{\partial^2 f(x, y, t)}{\partial y^2} \approx \frac{1}{\Delta y^2}[f_{k_1,k_2+1} - 2f_{k_1,k_2} + f_{k_1,k_2-1}]. \quad (123)$$



(a)

(b)

Fig. 4. Spatial distribution of $f(x, y)$ over the network grid $\{(x_{k_1}, y_{k_2})\}$ (a) Network topology (b) f_{k_1,k_2} over the space.

This leads to the following discretized input function:

$$h^o_{k_1,k_2} = \frac{1}{\Delta x^2}\big(f_{k_1+1,k_2} + f_{k_1,k_2+1} + f_{k_1-1,k_2} + f_{k_1,k_2-1} - 4f_{k_1,k_2}\big). \quad (124)$$

For this example, the unknown input process is

$$h^o_{k_1,k_2} = e^{-\kappa\left((k_1-4)^2+(k_2-4)^2\right)} - 5e^{-\kappa\left((k_1-8)^2+(k_2-8)^2\right)} + 1 \quad (125)$$

where $\kappa = (N_x - 1)^2/4$.

To obtain f_{k_1,k_2}, we solve (121) using the Jacobi over-relaxation method [45]. Fig. 4(b) illustrates the values of f_{k_1,k_2} over the spatial domain. For the estimation of h_{k_1,k_2}, the given information are the noisy measurement samples $\boldsymbol{z}_{k_1,k_2}(i) = f_{k_1,k_2} + \boldsymbol{n}_{k_1,k_2}(i)$. In this relation, the noise process $\boldsymbol{n}_{k_1,k_2}(i)$ is zero mean, temporally white and independent over space. For this network, the two dimensional reference signal is the distorted version of $h^o_{k_1,k_2}$ which is represented by $\boldsymbol{d}_{k_1,k_2}(i)$. The reference signal is obtained from (124) with f_{k_1,k_2} replaced by their noisy measured samples $\boldsymbol{z}_{k_1,k_2}(i)$, i.e.,

$$\boldsymbol{d}_{k_1,k_2}(i) = \frac{1}{\Delta x^2}\Big(\boldsymbol{z}_{k_1+1,k_2}(i) + \boldsymbol{z}_{k_1,k_2+1}(i) + \boldsymbol{z}_{k_1-1,k_2}(i) + \boldsymbol{z}_{k_1,k_2-1}(i) - 4\boldsymbol{z}_{k_1,k_2}(i)\Big). \quad (126)$$

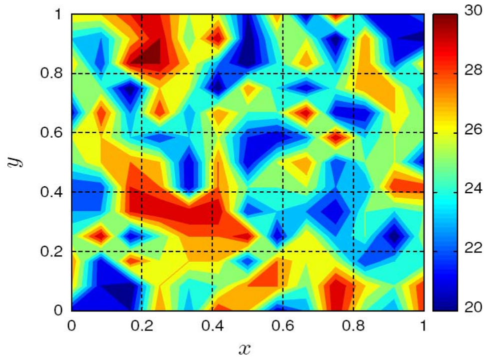
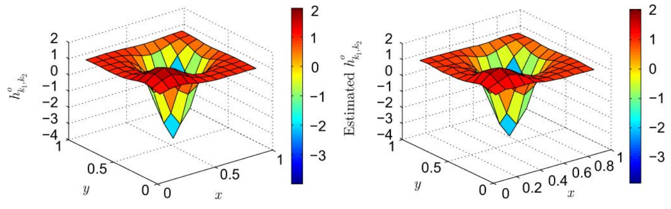


Fig. 5. Spatial distribution of SNR over the network.



(a)

(b)

Fig. 6. True and estimated $h^o_{k_1,k_2}$ by diffusion LMS (a) True parameters (b) Estimated parameters.

According to (126), the linear regression model for this problem takes the following form:

$$\boldsymbol{d}_{k_1,k_2}(i) = \boldsymbol{u}_{k_1,k_2}(i)h^o_{k_1,k_2} + \boldsymbol{v}_{k_1,k_2}(i) \tag{127}$$

where $\boldsymbol{u}_{k_1,k_2}(i) = 1$. Therefore, in this example, we are led to a linear model (127) with *deterministic* as opposed to random regression data. Although we only studied the case of random regression data in this article, this example is meant to illustrate that the diffusion strategy can still be applied to models involving deterministic data in a manner similar to [1], [54].

To represent $h^o_{k_1,k_2}$ as a space-invariant parameter vector, we use two-dimensional shifted Chebyshev basis functions [55]. Using this representation, $h^o_{k_1,k_2}$ can be expressed as:

$$h^o_{k_1,k_2} = \sum_{n=1}^{N_b} w^o_n \, p_{n,k_1,k_2} \tag{128}$$

where each element of the two-dimensional basis set is:

$$p_{n,k_1,k_2} = b_{n_1,k_1} b_{n_2,k_2} \tag{129}$$

where $\{b_{n_1,k_1}\}$ and $\{b_{n_2,k_2}\}$ are the one-dimensional shifted Chebyshev polynomials in the x and y directions, respectively-recall (21).

In the network, each interior node communicates with its four immediate neighbors. We use $A_1 = I$ and compute C and A_2 by using the Metropolis and relative degree rules [2], [6], [35]. All nodes are initialized at zero and $\mu_k = 0.01$ for all k. The signal-to-noise ratio (SNR) of the network is uniformly distributed in the range $[20, 30]$ dB and is shown in Fig. 5.

Figs. 6(a) and 6(b) show three dimensional views of the true and estimated input process using the proposed diffusion LMS algorithm after 3000 iterations. Fig. 7 illustrates the MSD of the estimated source, i.e., $\lim_{i\to\infty} \mathbb{E}\|h^o_{k_1,k_2} - \boldsymbol{h}_{k_1,k_2}(i)\|^2$.

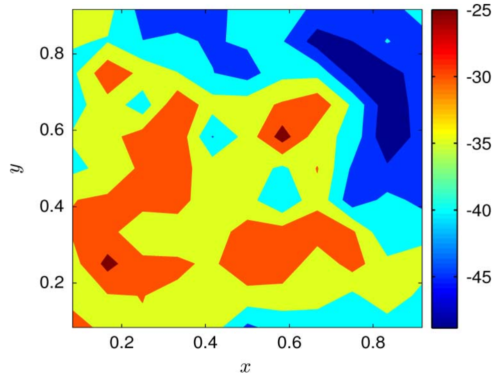


Fig. 7. Network steady-state MSD performance in dB.

VI. Conclusion

By combining interpolation and distributed adaptive optimization, we proposed a diffusion LMS strategy for estimation and tracking of space-time varying parameters over networks. The proposed algorithm can find the space-varying parameters not only at the node locations but also at spaces where no measurement is collected. We showed that if the network experiences data with rank-deficient covariance matrices, the non-cooperative LMS algorithm will converge to different solutions at different nodes. In contrast, the diffusion LMS algorithm is able to alleviate the rank-deficiency problem through its use of combination matrices especially since, as shown by (72), $\rho(\mathcal{B}) \leq \rho(I - \mathcal{MR})$, where $I - \mathcal{MR}$ is the coefficient matrix that governs the dynamics of the non-cooperative solution. Nevertheless, if these mechanisms fail to mitigate the deleterious effect of the rank-deficient data, then the algorithm converges to a solution space where the error is bounded. We analyzed the performance of the algorithm in transient and steady-state regimes, and gave conditions under which the algorithm is stable in the mean and mean-square sense.

Appendix A
Mean Error Convergence

Based on the rank of $\mathcal{R} = \text{diag}\{R_1, \ldots, R_N\}$, we have two possible cases:

a) $R_k > 0 \ \forall k \in \{1, \ldots, N\}$: As (68) implies, $\mathbb{E}[\tilde{\boldsymbol{w}}_i]$ converges to zero if $\rho(\mathcal{B}) < 1$. In [6], it was shown that when $\mathcal{R} > 0$, choosing the step-sizes according to (73) guarantees $\rho(\mathcal{B}) < 1$.

b) $\exists k \in \{1, \ldots, N\}$ for which R_k is rank-deficient: For this case, we first show that

$$\left\|\mathcal{B}^{i+1}\right\|_{b,\infty} \leq \left\|\left(I - \mathcal{M}\Lambda\right)^{i+1}\right\|_{b,\infty} \tag{130}$$

where $\|\cdot\|_{b,\infty}$ denotes the block-maximum norm for block vectors with block entries of size $MN_b \times 1$ and block matrices with blocks of size $MN_b \times MN_b$. To this end, we note that for the left-stochastic matrices A_1 and A_2, we have $\|\mathcal{A}_1^T\|_{b,\infty} = \|\mathcal{A}_2^T\|_{b,\infty} = 1$[6], and use the sub-multiplicative property of the block maximum norm [46] to write:

$$\begin{aligned}\left\|\mathcal{B}^{i+1}\right\|_{b,\infty} \leq & \|\mathcal{A}_2^T\|_{b,\infty} \, \|I - \mathcal{MR}\|_{b,\infty} \, \|\mathcal{A}_1^T\|_{b,\infty} \times \cdots \\ & \times \|\mathcal{A}_2^T\|_{b,\infty} \, \|I - \mathcal{MR}\|_{b,\infty} \, \|\mathcal{A}_1^T\|_{b,\infty} \\ = & \left\|I - \mathcal{MR}\right\|^{i+1}_{b,\infty}. \end{aligned} \tag{131}$$

If we introduce the (block) eigendecomposition of $\mathcal{R}$ (71) into (131) and consider the fact that the block-maximum norm is invariant under block-diagonal unitary matrix transformations [6], [47], then inequality (131) takes the form:

$$\left\|\mathcal{B}^{i+1}\right\|_{b,\infty} \le \left\|I - \mathcal{M}\Lambda\right\|_{b,\infty}^{i+1}. \tag{132}$$

Using the property $\|X\|_{b,\infty} = \rho(X)$ for a block diagonal Hermitian matrix X[6], we obtain:

$$\begin{aligned}\left\|(I - \mathcal{M}\Lambda)^{i+1}\right\|_{b,\infty} &= \rho\Big((I - \mathcal{M}\Lambda)^{i+1}\Big)\\ &= \max_{\substack{1\le k\le N\\ 1\le n\le MN_b}} \left|\big(1-\mu_k\lambda_k(n)\big)^{i+1}\right|\\ &= \Big(\max_{\substack{1\le k\le N\\ 1\le n\le MN_b}} \left|1-\mu_k\lambda_k(n)\right|\Big)^{i+1}\\ &= \Big(\rho(I - \mathcal{M}\Lambda)\Big)^{i+1}\\ &= \left\|I - \mathcal{M}\Lambda\right\|_{b,\infty}^{i+1}. \end{aligned}\tag{133}$$

Using (133) in (132), we arrive at (130). We now proceed to show the boundedness of the mean error for case (b). We iterate (68) to get:

$$\mathbb{E}[\tilde{\boldsymbol{w}}_i] = \mathcal{B}^{i+1}\mathbb{E}[\tilde{\boldsymbol{w}}_{-1}]. \tag{134}$$

Applying the block maximum norm to (134) and using inequality (130), we obtain:

$$\lim_{i\to\infty} \left\|\mathbb{E}[\tilde{\boldsymbol{w}}_i]\right\|_{b,\infty} \le \lim_{i\to\infty} \left\|(I - \mathcal{M}\Lambda)^{i+1}\right\|_{b,\infty} \left\|\mathbb{E}[\tilde{\boldsymbol{w}}_{-1}]\right\|_{b,\infty}. \tag{135}$$

The value of $\lim_{i\to\infty} \|(I - \mathcal{M}\Lambda)^{i+1}\|_{b,\infty}$ can be computed by evaluating the limits of its diagonal entries. Considering the step-sizes as in (73), the diagonal entries are computed as:

$$\lim_{i\to\infty} \big(1-\mu_k\lambda_k(n)\big)^{i+1} = \begin{cases} 1, & \text{if } \lambda_k(n) = 0\\ 0, & \text{otherwise}\end{cases}. \tag{136}$$

Therefore, (135) reads as:

$$\lim_{i\to\infty} \left\|\mathbb{E}[\tilde{\boldsymbol{w}}_i]\right\|_{b,\infty} \le \|I - \mathrm{Ind}(\Lambda)\|_{b,\infty} \left\|\mathbb{E}[\tilde{\boldsymbol{w}}_{-1}]\right\|_{b,\infty}. \tag{137}$$

Appendix B
Mean Behavior When $(A_1 = A_2 = I)$

Setting $A_1 = A_2 = I$ in the diffusion recursions (39)–(42) and subtracting w^o from both sides of (40), we get:

$$\tilde{\boldsymbol{w}}_{k,i} = \tilde{\boldsymbol{w}}_{k,i-1} - \mu_k \sum_{\ell\in\mathcal{N}_k} c_{\ell,k} B_\ell^T \boldsymbol{u}_{\ell,i}^T(\boldsymbol{d}_\ell(i) - \boldsymbol{u}_{\ell,i}B_\ell \boldsymbol{w}_{k,i-1}). \tag{138}$$

Under Assumption 1 and using $\boldsymbol{d}_\ell(i) = \boldsymbol{u}_{\ell,i}B_\ell w^o + \boldsymbol{v}_\ell(i)$, we obtain:

$$\mathbb{E}[\tilde{\boldsymbol{w}}_{k,i}] = Q_k[I - \mu_k\Lambda_k]Q_k^T\,\mathbb{E}[\tilde{\boldsymbol{w}}_{k,i-1}]. \tag{139}$$

We define $\boldsymbol{p}_{k,i} \triangleq Q_k^T\tilde{\boldsymbol{w}}_{k,i}$ and start from some initial conditions to arrive at

$$\mathbb{E}[\boldsymbol{p}_{k,i}] = [I - \mu_k\Lambda_k]\mathbb{E}[\boldsymbol{p}_{k,i-1}] = [I - \mu_k\Lambda_k]^{i+1}\mathbb{E}[\boldsymbol{p}_{k,-1}].$$

If we choose the step-sizes according to (73) then we get:

$$\lim_{i\to\infty} \mathbb{E}[\boldsymbol{p}_{k,i}] = \big[I - \mathrm{Ind}(\Lambda_k)\big]\mathbb{E}[\boldsymbol{p}_{k,-1}]. \tag{140}$$

Equivalently, this can be written as:

$$\lim_{i\to\infty} \mathbb{E}[\tilde{\boldsymbol{w}}_{k,i}] = Q_k\big[I - \mathrm{Ind}(\Lambda_k)\big]Q_k^T\,\mathbb{E}[\tilde{\boldsymbol{w}}_{k,-1}]. \tag{141}$$

This result indicates that the mean error does not grow unbounded. Now from (75), we can verify that:

$$Q_k\mathrm{Ind}(\Lambda_k)Q_k^T w^o = R_k^\dagger r_k. \tag{142}$$

Then, upon substitution of $\tilde{\boldsymbol{w}}_{k,i} = w^o - \boldsymbol{w}_{k,i}$ into (141), we obtain:

$$\begin{aligned}\lim_{i\to\infty} \mathbb{E}[\boldsymbol{w}_{k,i}] &= Q_k\mathrm{Ind}(\Lambda_k)Q_k^T w^o\\ &\quad + Q_k[I - \mathrm{Ind}(\Lambda_k)]Q_k^T\mathbb{E}[\boldsymbol{w}_{k,-1}]\\ &= R_k^\dagger r_k + \sum_{n=L_k+1}^{MN_b} q_{k,n}q_{k,n}^T\mathbb{E}[\boldsymbol{w}_{k,-1}].\end{aligned} \tag{143}$$

Appendix C
Proof of Lemma 2

From (87), we readily deduce that

$$\lim_{i\to\infty} \mathcal{B}^{i+1}\mathbb{E}[\boldsymbol{w}_{-1}] = (\mathcal{Z}_2\bar{\mathcal{Z}}_2)\,\mathbb{E}[\boldsymbol{w}_{-1}]. \tag{144}$$

On the other hand, from (86), we have

$$\lim_{i\to\infty}\sum_{j=0}^{i} \mathcal{B}^j\mathcal{A}_2^T\mathcal{M}r = \lim_{i\to\infty}\sum_{j=0}^{i} \big(\mathcal{Z}_1J^j\bar{\mathcal{Z}}_1 + \mathcal{Z}_2\bar{\mathcal{Z}}_2\big)\mathcal{A}_2^T\mathcal{M}r. \tag{145}$$

Using (88), the term involving $\bar{\mathcal{Z}}_2$ cancels out and the above reduces to

$$\begin{aligned}\lim_{i\to\infty}\sum_{j=0}^{i} \mathcal{B}^j\mathcal{A}_2^T\mathcal{M}r &= \lim_{i\to\infty}\sum_{j=0}^{i} \big(\mathcal{Z}_1J^j\bar{\mathcal{Z}}_1\big)\mathcal{A}_2^T\mathcal{M}r\\ &= \mathcal{Z}_1(I-J)^{-1}\bar{\mathcal{Z}}_1\mathcal{A}_2^T\mathcal{M}r\end{aligned} \tag{146}$$

since $\rho(J) < 1$. We now verify that the matrix

$$X^- = \mathcal{Z}_1(I-J)^{-1}\bar{\mathcal{Z}}_1 \tag{147}$$

is a (reflexive) generalized inverse for the matrix $X = (I - \mathcal{B})$. Recall that a (reflexive) generalized inverse for a matrix Y is any matrix Y^- that satisfies the two conditions [56]:

$$YY^-Y = Y \tag{148}$$

$$Y^-YY^- = Y^-. \tag{149}$$

To verify these conditions, we first note from $\mathcal{Z}\mathcal{Z}^{-1} = I$ and $\mathcal{Z}^{-1}\mathcal{Z} = I$ in (86) that the following relations hold:

$$\mathcal{Z}_1\bar{\mathcal{Z}}_1 + \mathcal{Z}_2\bar{\mathcal{Z}}_2 = I \tag{150}$$

$$\bar{\mathcal{Z}}_1\mathcal{Z}_2 = 0 \tag{151}$$

$$\bar{\mathcal{Z}}_2\mathcal{Z}_1 = 0 \tag{152}$$

$$\bar{\mathcal{Z}}_1\mathcal{Z}_1 = I \tag{153}$$

$$\bar{\mathcal{Z}}_2\mathcal{Z}_2 = I. \tag{154}$$

We further note that X can be expressed as:

$$X = (I - \mathcal{B}) = \mathcal{Z}_1(I - J)\bar{\mathcal{Z}}_1. \tag{155}$$

It is then easy to verify that the matrices $\{X, X^-\}$ satisfy conditions (148) and (149), as claimed. Therefore, (146) can be expressed as:

$$\lim_{i\to\infty}\sum_{j=0}^{i}\mathcal{B}^j\mathcal{A}_2^T\mathcal{M}r = (I - \mathcal{B})^-\mathcal{A}_2^T\mathcal{M}r. \tag{156}$$

Substituting (144) and (156) into (78) leads to (89).

Let us now verify that the right-hand side of (89) remains invariant under basis transformations for the Jordan factors $\{\mathcal{Z}_1, \bar{\mathcal{Z}}_1, \mathcal{Z}_2, \bar{\mathcal{Z}}_2\}$. To begin with, the Jordan decomposition (86) is not unique. Let us assume, however, that we fix the central term $\mathrm{diag}\{J, I\}$ to remain invariant and allow the Jordan factors $\{\mathcal{Z}_1, \bar{\mathcal{Z}}_1, \mathcal{Z}_2, \bar{\mathcal{Z}}_2\}$ to vary. It follows from (86) that

$$\bar{\mathcal{Z}}_2\mathcal{B} = \bar{\mathcal{Z}}_2, \quad \mathcal{B}\mathcal{Z}_2 = \mathcal{Z}_2 \tag{157}$$

so that the columns of $\mathcal{Z}_2$ and the rows of $\bar{\mathcal{Z}}_2$ correspond to right and left-eigenvectors of $\mathcal{B}$, respectively, associated with the eigenvalues with value one. If we replace $\mathcal{Z}_2$ by any transformation of the form $\mathcal{Z}_2\mathcal{X}_2$, where $\mathcal{X}_2$ is invertible, then by (154), $\bar{\mathcal{Z}}_2$ should be replaced by $\mathcal{X}_2^{-1}\bar{\mathcal{Z}}_2$. This conclusion can also be seen as follows. The new factor $\mathcal{Z}$ is given by

$$\mathcal{Z} \triangleq [\,\mathcal{Z}_1 \quad \mathcal{Z}_2\mathcal{X}_2\,] = [\,\mathcal{Z}_1 \quad \mathcal{Z}_2\,]\begin{bmatrix} I & 0 \\ 0 & \mathcal{X}_2 \end{bmatrix} \tag{158}$$

and, hence, the new $\mathcal{Z}^{-1}$ becomes

$$\mathcal{Z}^{-1} = \begin{bmatrix} \bar{\mathcal{Z}}_1 \\ \mathcal{X}_2^{-1}\bar{\mathcal{Z}}_2 \end{bmatrix} \tag{159}$$

which confirms that $\bar{\mathcal{Z}}_2$ is replaced by $\mathcal{X}_2^{-1}\bar{\mathcal{Z}}_2$. It follows that the product $\mathcal{Z}_2\bar{\mathcal{Z}}_2$ remains invariant under arbitrary invertible transformations $\mathcal{X}_2$. Moreover, from (86) we also have that

$$\bar{\mathcal{Z}}_1\mathcal{B} = J\bar{\mathcal{Z}}_1, \quad \mathcal{B}\mathcal{Z}_1 = \mathcal{Z}_1 J. \tag{160}$$

Assume we replace $\mathcal{Z}_1$ by any transformation of the form $\mathcal{Z}_1\mathcal{X}_1$, where $\mathcal{X}_1$ is invertible, then by (153), $\bar{\mathcal{Z}}_1$ should be replaced by $\mathcal{X}_1^{-1}\bar{\mathcal{Z}}_1$. However, since we want to maintain J invariant, then this implies that the transformation $\mathcal{X}_1$ must also satisfy

$$\mathcal{X}_1^{-1}J\mathcal{X}_1 = J. \tag{161}$$

It follows that the product $\mathcal{Z}_1(I - J)^{-1}\bar{\mathcal{Z}}_1$ remains invariant under such invertible transformations $\mathcal{X}_1$, since

$$\begin{aligned}\mathcal{Z}_1(I - J)^{-1}\bar{\mathcal{Z}}_1 &= \mathcal{Z}_1\mathcal{X}_1\mathcal{X}_1^{-1}(I - J)^{-1}\mathcal{X}_1\mathcal{X}_1^{-1}\bar{\mathcal{Z}}_1 \\ &= \mathcal{Z}_1\mathcal{X}_1(I - \mathcal{X}_1^{-1}J\mathcal{X}_1)^{-1}\mathcal{X}_1^{-1}\bar{\mathcal{Z}}_1 \\ &= \mathcal{Z}_1\mathcal{X}_1(I - J)^{-1}\mathcal{X}_1^{-1}\bar{\mathcal{Z}}_1. \end{aligned} \tag{162}$$

APPENDIX D
PROOF OF LEMMA 3

We first establish that $\bar{\mathcal{Z}}_2\mathcal{Y}$ and $\mathcal{Y}\bar{\mathcal{Z}}_2^T$ are both equal to zero. Indeed, we start by replacing r in (88) by its expression from (61) and (63) as $r = \mathcal{C}^T\mathrm{col}\{\bar{r}_{du,1}, \ldots, \bar{r}_{du,N}\}$ that leads to:

$$\bar{\mathcal{Z}}_2\mathcal{A}_2^T\mathcal{M}\mathcal{C}^T\mathrm{col}\{\bar{r}_{du,1}, \ldots, \bar{r}_{du,N}\} = 0. \tag{163}$$

By further replacing $\bar{r}_{du,k}$ by their values from (32), we obtain:

$$\bar{\mathcal{Z}}_2\mathcal{A}_2^T\mathcal{M}\mathcal{C}^T\mathrm{diag}\{B_1^T, \ldots, B_N^T\}\mathrm{col}\{r_{du,1}, \ldots, r_{du,N}\} = 0. \tag{164}$$

This relation must hold regardless of the cross-correlation vectors $\{r_{du,k}\}$. Therefore,

$$\bar{\mathcal{Z}}_2\mathcal{A}_2^T\mathcal{M}\mathcal{C}^T\mathrm{diag}\{B_1^T, \ldots, B_N^T\} = 0. \tag{165}$$

We now define

$$\mathcal{V} = \mathrm{diag}\{\sigma_{v,1}^2 I_{MN_b}, \ldots, \sigma_{v,N}^2 I_{MN_b}\} \tag{166}$$

and rewrite expression (100) as

$$\begin{aligned}\mathcal{Y} = \mathcal{A}_2^T\mathcal{M}\mathcal{C}^T&\mathrm{diag}\{B_1^T, \ldots, B_N^T\}\mathrm{diag}\{R_{u,1}, \ldots, R_{u,N}\} \\ &\times\mathrm{diag}\{B_1, \ldots, B_N\}\,\mathcal{V}\mathcal{C}\mathcal{M}\mathcal{A}_2. \end{aligned} \tag{167}$$

Multiplying this from the left by $\bar{\mathcal{Z}}_2$ and comparing the result with (165), we conclude that

$$\bar{\mathcal{Z}}_2\mathcal{Y} = 0. \tag{168}$$

Noting that $\mathcal{Y}$ is symmetric, we then obtain:

$$\mathcal{Y}\bar{\mathcal{Z}}_2^T = 0. \tag{169}$$

Returning to recursion (102), we note first from (86) that $\mathcal{B}$ can be rewritten as

$$\mathcal{B} = \mathcal{Z}_1 J\bar{\mathcal{Z}}_1 + \mathcal{Z}_2\bar{\mathcal{Z}}_2. \tag{170}$$

Since $\mathcal{B}$ is power convergent, the first term on the right hand side of (102) converges to

$$\lim_{i\to\infty}\mathbb{E}\|\tilde{\boldsymbol{w}}_{-1}\|^2_{(\mathcal{B}^T)^{i+1}\Sigma\mathcal{B}^{i+1}} = \mathbb{E}\|\tilde{\boldsymbol{w}}_{-1}\|^2_{(\mathcal{Z}_2\bar{\mathcal{Z}}_2)^T\Sigma\mathcal{Z}_2\bar{\mathcal{Z}}_2}. \tag{171}$$

Substituting (170) into the second term on the right hand side of (102) and using (168) and (169), we arrive at

$$\begin{aligned}\lim_{i\to\infty}\sum_{j=0}^{i}&\mathrm{Tr}\Big((\mathcal{B}^T)^j\Sigma\mathcal{B}^j\mathcal{Y}\Big) \\ &= \mathrm{Tr}\Big(\lim_{i\to\infty}\sum_{j=0}^{i}(\mathcal{Z}_1 J^j\bar{\mathcal{Z}}_1)^T \times \Sigma(\mathcal{Z}_1 J^j\bar{\mathcal{Z}}_1)\mathcal{Y}\Big). \end{aligned} \tag{172}$$

If matrices X_1, X_2 and Σ are of compatible dimensions, then the following relations hold [6]:

$$\mathrm{Tr}(X_1X_2) = \big(\mathrm{vec}(X_2^T)\big)^T\mathrm{vec}(X_1) \tag{173}$$

$$\mathrm{vec}(X_1\Sigma X_2) = (X_2^T \otimes X_1)\mathrm{vec}(\Sigma). \tag{174}$$

Using these relations in (172), we obtain

$$\mathrm{Tr}\Big(\lim_{i\to\infty}\sum_{j=0}^{i}(\mathcal{B}^T)^j\Sigma\mathcal{B}^j\mathcal{Y}\Big) = \Big(\mathrm{vec}(\mathcal{Y}^T)\Big)^T \times\Big(\lim_{i\to\infty}\sum_{j=0}^{i}(\mathcal{Z}_1 J^j\bar{\mathcal{Z}}_1)^T\otimes(\mathcal{Z}_1 J^j\bar{\mathcal{Z}}_1)^T\Big)\mathrm{vec}(\Sigma). \quad (175)$$

This is equivalent to:

$$\mathrm{Tr}\Big(\sum_{j=0}^{\infty}(\mathcal{B}^T)^j\Sigma\mathcal{B}^j\mathcal{Y}\Big) = \big(\mathrm{vec}(\mathcal{Y})\big)^T\Big(\sum_{j=0}^{\infty}\mathcal{F}^j\Big)\mathrm{vec}(\Sigma) \quad (176)$$

where

$$\mathcal{F} = \Big((\mathcal{Z}_1\otimes\mathcal{Z}_1)(J\otimes J)(\bar{\mathcal{Z}}_1\otimes\bar{\mathcal{Z}}_1)\Big)^T. \quad (177)$$

Since $\rho(J\otimes J)<1$, the series converges and we obtain:

$$\mathrm{Tr}\Big(\lim_{i\to\infty}\sum_{j=0}^{i}(\mathcal{B}^T)^j\Sigma\mathcal{B}^j\mathcal{Y}\Big) = \Big(\mathrm{vec}(\mathcal{Y})\Big)^T(I-\mathcal{F})^{-1}\mathrm{vec}(\Sigma). \quad (178)$$

Upon substitution of (171) and (178) into (102), we arrive at (104).

REFERENCES

[1] F. S. Cattivelli, C. G. 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] A. H. Sayed, "Diffusion adaptation over networks," in *Academic Press Library in Signal Processing, 2013.* , R. Chellapa and S. Theodoridis, Eds. New York, NY, USA: Academic, Elsevier, 2012, pp. 323–454 [Online]. Available: arXiv:1205.4220v2

[7] S. Y. Tu and A. H. Sayed, "Adaptive decision-making over complex networks," in *Proc. Asilomar Conf. Signals, Syst., Comput.*, Pacific Grove, CA, USA, Nov. 2012, pp. 525–530.

[8] P. D. Lorenzo, S. Barbarossa, and A. H. Sayed, "Decentralized resource assignment in cognitive networks based on swarming mechanisms over random graphs," *IEEE Trans. Signal Process.*, vol. 60, no. 7, pp. 3755–3769, Jul. 2012.

[9] T. Lee and J. Seinfeld, "Estimation of two-phase petroleum reservoir properties by regularization," *J. Computat. Phys.*, vol. 69, no. 2, pp. 397–419, Feb. 1987.

[10] E. Holmes, M. Lewis, J. Banks, and R. Veit, "Partial differential equations in ecology: Spatial interactions and population dynamics," *Ecology*, vol. 75, no. 1, pp. 17–29, Jan. 1994.

[11] M. Alpay and M. Shor, "Model-based solution techniques for the source localization problem," *IEEE Trans. Control Syst. Technol.*, vol. 8, no. 6, pp. 895–904, Jun. 2000.

[12] N. V. Kampen, "Diffusion in inhomogeneous media," *J. Phys. Chem. Solids*, vol. 49, no. 1, pp. 673–677, Jun. 1988.

[13] C. Chung and C. Kravaris, "Identification of spatially discontinuous parameters in second-order parabolic systems by piecewise regularization," *Inverse Problems*, vol. 4, pp. 973–994, Feb. 1988.

[14] G. Richter, "Numerical identification of a spatially-varying diffusion coefficient," *Math. Comput.*, vol. 36, no. 154, pp. 375–386, Apr. 1981.

[15] V. Isakov and S. Kindermann, "Identification of the diffusion coefficient in a one-dimensional parabolic equation," *Inverse Problems*, vol. 16, pp. 665–680, Jun. 2000.

[16] M. Demetriou, "Process estimation and moving source detection in 2-D diffusion processes by scheduling of sensor networks," in *Proc. Amer. Control Conf.*, New York, Jul. 2007, pp. 3432–3437.

[17] M. Demetriou and I. Hussein, "Estimation of spatially distributed processes using mobile spatially distributed sensor network," *SIAM J. Contr. Optimiz.*, vol. 48, no. 1, pp. 266–291, Feb. 2009.

[18] 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.

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

[20] 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.

[21] 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.

[22] L. Li, J. A. Chambers, C. G. Lopes, and A. H. Sayed, "Distributed estimation over an adaptive incremental network based on the affine projection algorithm," *IEEE Trans. Signal Process.*, vol. 58, no. 1, pp. 151–164, Jan. 2010.

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

[24] L. Xiao, S. Boyd, and S. Lall, "A space-time diffusion scheme for peer-to-peer least-squares estimation," in *Proc. 5th Int. Conf. Inf. Process. Sens. Netw. (IPSN)*, Nashville, TN, USA, Apr. 2006, pp. 168–176.

[25] S. S. Stankovic, M. S. Stankovic, and D. M. Stipanovic, "Decentralized parameter estimation by consensus based stochastic approximation," *IEEE Trans. Autom. Control*, vol. 56, no. 3, pp. 531–543, Mar. 2011.

[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] T. Aysal, M. Yildiz, A. Sarwate, and A. Scaglione, "Broadcast gossip algorithms for consensus," *IEEE Trans. Signal Process.*, vol. 57, no. 7, pp. 2748–2761, Jul. 2009.

[29] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2508–2530, Jun. 2006.

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

[31] K. Srivastava and A. Nedic, "Distributed asynchronous constrained stochastic optimization," *IEEE Sel. Topics Signal Process.*, vol. 5, no. 4, pp. 772–790, Aug. 2011.

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

[33] P. D. 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.

[34] Y. Hu and A. Ribeiro, "Adaptive distributed algorithms for optimal random access channels," *IEEE Trans. Wireless Commun.*, vol. 10, no. 8, pp. 2703–2715, Aug. 2010.

[35] 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.

[36] 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, pp. 155–171, May 2013.

[37] 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.

[38] R. Mattheij, S. Rienstra, and J. ten Thije Boonkkamp, *Partial Differential Equations: Modeling, Analysis, Computation*. Philadelphia, PA, USA: SIAM, 2005.

[39] J. Thomas, *Numerical Partial Differential Equations: Finite Difference Methods*. New York, NY, USA: Springer-Verlag, 1995.

[40] J. C. Mason and D. C. Handscomb, *Chebyshev Polynomials*. Boca Raton, FL, USA: Chapman & Hall/CRC, 2003.

[41] E. Isaaks and R. Srivastava, *An Introduction to Applied Geostatistics*. New York, NY, USA: Oxford Univ. Press, 1989.

[42] S.-J. Kim, E. Dall'Anese, and G. B. Giannakis, "Cooperative spectrum sensing for cognitive radios using Kriged Kalman filtering," *IEEE Sel. Topics Signal Process.*, vol. 5, no. 1, pp. 24–36, Jan. 2011.

[43] N. A. Cressie, *Statistics for Spatial Data*. New York, NY, USA: Wiley, 1993.

[44] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, *Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers*. New York, NY, USA: Now , 2011.

[45] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation*. Old Tappan, NJ, USA: Prentice-Hall, 1989.

[46] X. Zhao, S. Y. Tu, and A. H. 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.

[47] 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.

[48] R. Abdolee and B. Champagne, "Diffusion LMS algorithms for sensor networks over non-ideal inter-sensor wireless channels," in *Proc. Int. Conf. on Dist. Computing in Sens. Syst. Workshops*, Barcelona, Spain, Jun. 2011, pp. 1–6.

[49] D. McLernon, M. Lara, and A. Orozco-Lugo, "On the convergence of the LMS algorithm with a rank-deficient input autocorrelation matrix," *Signal Process.*, vol. 89, pp. 2244–2250, Nov. 2009.

[50] C. D. Meyer, *Matrix Analysis and Applied Linear Algebra*. Philadelphia, PA, USA: SIAM, 2000.

[51] R. A. Horn and C. R. Johnson, *Matrix analysis*. Cambridge, U.K.: Cambridge Univ. Press , 2003.

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

[53] T. Al-Naffouri and A. H. Sayed, "Transient analysis of data-normalized adaptive filters," *IEEE Trans. Signal Process.*, vol. 51, no. 3, pp. 639–652, Mar. 2003.

[54] F. Cattivelli and A. H. Sayed, "Distributed detection over adaptive networks using diffusion adaptation," *IEEE Trans. Signal Process.*, vol. 59, no. 5, pp. 1917–1932, May 2011.

[55] R. Mukundan, S. Ong, and P. A. Lee, "Image analysis by tchebichef moments," *IEEE Trans. Image Process.*, vol. 10, no. 9, pp. 1357–1364, Sep. 2001.

[56] A. Ben-Israel and T. N. Greville, *Generalized Inverses: Theory and Applications*. New York, NY, USA: Springer, 2003.

Reza Abdolee (S'13) is currently a Ph.D. candidate at the Department of Electrical and Computer Engineering, McGill University, Montreal, Canada. In 2012, he was a research scholar at the Bell Labs, Alcatel-Lucent, Stuttgart, Germany. In 2011, he was a visiting Ph.D. student at the Department of Electrical Engineering, University of California, Los Angeles (UCLA). From 2006 to 2008, he was as a staff engineer at the Wireless Communication Center, University of Technology, Malaysia (UTM), 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, 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, ReSMiQ International Doctoral Scholarship, Graduate Student Support Award (Concordia University), and International Tuition Fee Remission Award (Concordia University).

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

From 1990 to 1999, he was an Assistant and then Associate Professor at INRS-Telecommunications, Université du Quebec, Montréal. In 1999, he joined McGill University, Montreal, where he is now a Full Professor within 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.

Dr. Champagne has been an Associate Editor for the IEEE Signal Processing Letters, the IEEE Transactions on Signal Processing and the EURASIP *Journal on Applied Signal Processing*. He has also served on the Technical Committees of several international conferences in the fields of communications and signal processing.

Ali H. Sayed (S'90–M'92–SM'99–F'01) is Professor and former Chairman of electrical engineering at the University of California, Los Angeles, where he directs the UCLA Adaptive Systems Laboratory. An author of more than 430 scholarly publications and five books, his research involves several areas including adaptation and learning, information processing theories, statistical signal processing, network science, and biologically-inspired designs.

Dr. Sayed's work has been recognized with several awards including the 2012 Technical Achievement Award from the IEEE Signal Processing Society, the 2005 Terman Award from the American Society for Engineering Education, a 2005 Distinguished Lecturer from the IEEE Signal Processing Society, the 2003 Kuwait Prize, and the 1996 IEEE Donald G. Fink Prize. He has also been awarded several paper awards from the IEEE Signal Processing Society (2002,2005,2012) and is a Fellow of the American Association for the Advancement of Science (AAAS). He has been active in serving the Signal Processing community in various roles. Among other activities, he served as Editor-in-Chief of the IEEE Transactions on Signal Processing (2003–2005), Editor-in-Chief of the EURASIP *Journal on Advances in Signal Processing* (2006–2007), General Chairman of ICASSP (Las, Vegas, 2008), and Vice-President of Publications of the IEEE Signal Processing Society (2009–2011). He also served as member of the Board of Governors (2007–2011), Awards Board (2005), Publications Board (2003–2005, 2009–2011), Conference Board (2007–2011), and Technical Directions Board (2008–2009) of the same Society and as General Chairman of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP) 2008.