This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCNS.2016.2578044, IEEE Transactions on Control of Network Systems

Tracking Performance and Optimal Adaptation Step-Sizes of Diffusion-LMS Networks

Reza Abdolee, *Member, IEEE,* Vida Vakilian, *Member, IEEE,* Benoit Champagne, *Senior Member, IEEE,*

***Abstract*—In this paper, we investigate the effect of adaptation step-sizes on the tracking performance of diffusion least-mean squares (DLMS) algorithms in networks under non-stationary signal conditions. We assume that the network parameter vector being estimated varies over time according to a first-order random walk model. To find the optimal adaptation step-sizes over the network, we formulate a constrained nonlinear optimization problem and solve it through a log-barrier Newton algorithm in an iterative manner. Our studies reveal that the optimal step-size of each node in the network not only depends on the statistics of the random walk and the energy profile of the node itself, but also on the energy and statistical profile of its neighboring nodes. The results show that the optimal step-sizes can substantially improve the performance of DLMS algorithms in tracking time-varying parameters over networks. We also find that the DLMS algorithms have faster tracking ability and superior steady-state mean-square deviation (MSD) performance than the DLMS in non-corporative mode since with the diffusion mode of cooperation, each node at each iteration can take a larger step toward the network optimal parameters.**

***Index Terms*—Diffusion adaptation, sensor networks, LMS algorithms, time-varying parameters, distributed estimation.**

I. Introduction

IN-NETWORK distributed adaptive signal processing is emerging as a key enabling technology for sensor networks to support the implementation of flexible cooperative learning and information processing schemes [1], [2]. This type of processing and learning mechanism, in which nearby nodes cooperate over wired or wireless links in the achievement of network-wide computational and control objectives without using a fusion center, will probably form one of the cornerstones of the future generation of data communications and control networks [3]–[9]. The potential applications of distributed adaptive processing are diverse, and currently, they are being considered in several areas, including factory automation, robotics, intelligent transportation, health care monitoring, precision agriculture, smart spaces and telecommunications [10]–[15], [37].

There are several variant forms of distributed adaptive algorithms for parameter estimation over networks, including the two well-known types of techniques, i.e., consensus [16]–[28] and diffusion [29]–[33]. It was shown in [34] that for constant step-size adaptation, network states can grow unbounded due to an inherent asymmetry in the consensus dynamics. The same problem does not occur for diffusion strategies [34], and for this reason, we focus on these algorithms in this work. The diffusion least mean-squares (DLMS) algorithm is one of the efficient algorithms in the diffusion class, besides the diffusion recursive least-squares (DRLS) [35], [36], that can demonstrate an excellent performance in the estimation of time-varying parameters over networks [34], [37]. This algorithm operates using a simple mode of cooperation, endow the network with adaptive and learning abilities and can exhibit agile tracking performance [29], [30], [37].

The mean and mean-square convergence behavior of DLMS algorithms in stationary signal environments have been extensively investigated in many previous works, including [30], [37], [38]. However, there are only a few studies that examined the performance of this algorithm in non-stationary signal environments. The latest works on this topic are [34] and [39], where the authors have compared the performance of DLMS with distributed consensus algorithms in time-varying environments. The results reported in these references confirmed that the DLMS algorithms outperform the consensus ones and exhibit a superior tracking performance under such conditions. Nevertheless these works only used some arbitrary adaptation step-sizes that satisfy the mean and mean-square stability conditions of the algorithms. So far however, the critically important effects of the step-size parameters on the convergence and tracking properties of the DLMS algorithms in non-stationary signal environments have not been thoroughly investigated. Furthermore, the link between the choice of optimal step-sizes and the dynamic model characterizing the time evolution of the underlying parameter vector subject to adaptive estimation has not been explored.

A. Contributions

In this paper, we study the tracking performance of DLMS algorithms in non-stationary signal environments, where the underlying parameter vector of interest evolves according to a first-order random walk model. To the best of our knowledge, such a key study of the DLMS algorithm along with the interpretation of the results from the perspective of control and networking engineering has not been attempted before in the literature. In particular, our main contributions can be summarized as follows in terms of novelty and importance:

- Investigating the effects of adaptation step-sizes on tracking performance of DLMS algorithms in networks with non-stationary signal conditions.

R. Abdolee is with Qualcomm, Inc., San Diego,. CA, 92121 (e-mail: reza.abdolee@gmail.com).

B. Champagne is with the Department of Electrical and Computer Engineering, McGill University, Montreal, QC, H3A 0E9 Canada (e-mail: benoit.champagne@mcgill.ca).

V. Vakilian is with the Department of Computer and Electrical Engineering & Computer Science, California State University, CA, 93311 (e-mail: vida.vakilian.ca@ieee.org).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCNS.2016.2578044, IEEE Transactions on Control of Network Systems

- Formulating a constrained nonlinear optimization problem to find nearly optimal step-size parameter vector within the stability range of DLMS algorithms.
- Finding the effect of the network energy profile on the network optimal adaptation step-size parameter.
- Implementing a log-barrier interior point method and gradient projection algorithm to find adaptation step-sizes in adaptive networks for time-varying model parameters.

To be more specific, we first show how an improper choice of adaptation step-sizes can substantially degrade the steady-state and tracking performance of DLMS algorithms. We then examine the mean-square error (MSE) performance of these algorithms with respect to changes in the adaptation step-sizes, and formulate an optimization problem to find their optimal values over the network, which enhance MSE performance of the algorithms in the steady-state. Furthermore, we learn that, similar to a stand-alone LMS algorithm in a non-stationary environment, the steady-state performance of DLMS does not continue to improve by decreasing the adaptation step-sizes, but instead starts deteriorating past a certain point. In addition, we find that compared to LMS, DLMS algorithms demonstrate faster tracking ability and superior performance in the steady-state mean-square deviation (MSD) and excess mean-square deviation (EMSE). According to our results, this is because with the diffusion mode of cooperation, each node at each iteration can take a larger step-size toward the network optimal point. Our analysis and simulation results also indicate that by using the optimal step-size at each node, we can substantially improve the steady-state MSE performance of the DLMS algorithms, and enhance their tracking speed.

The paper is organized as follows. In Section II, we briefly review DLMS algorithms and explain their operation in the estimation of time-varying parameters over networks. In Section III, we characterize the performance of these algorithms in time-varying signal environments. In Section IV, we examine the tracking performance of DLMS algorithms with respect to the choice of step-size parameters. We present our numerical results in Section V, and conclude the paper in Section VI.

Notation: We use boldface letters for random variables and normal letters for deterministic variables. Symbol $(\cdot)^T$ denotes transposition for real vectors and matrices, and $(\cdot)^*$ denotes conjugate transposition for complex vectors and matrices. We show the trace operator with $\mathrm{Tr}(\cdot)$, the spectral radius of a matrix with $\rho(\cdot)$ and the expectation by $\mathbb{E}[\cdot]$. We use $\mathrm{diag}\{\cdot\}$ to extract the diagonal entries of a matrix, or to construct a diagonal matrix from a vector. The $\mathrm{vec}(\cdot)$ operator vectorizes a matrix by stacking its columns on top of each other and $\mathrm{bvec}(\cdot)$ is the block-vectorization operator [29]. We denote the Kronecker products by $\otimes$, and *block* Kronecker products by $\otimes_b$.

II. Diffusion LMS Strategies

We consider a collection of N nodes which are distributed over a geographical area to estimate the unknown time-varying parameter vector $\boldsymbol{w}_i^o \in \mathbb{C}^{M\times 1}$ where i is the time index. Each node $k \in \{1, 2, \cdots, N\}$ at time instant $i \in \mathbb{N}$ collects the measurement sample $\boldsymbol{d}_k(i)$ that is related to the time-varying parameter vector through the following regression model:

$$\boldsymbol{d}_k(i) = \boldsymbol{u}_{k,i}\boldsymbol{w}_i^o + \boldsymbol{v}_{k,i} \tag{1}$$

where $\boldsymbol{u}_{k,i} \in \mathbb{C}^{1\times M}$ represents the regression data and $\boldsymbol{v}_{k,i} \in \mathbb{C}$ represents the output measurement noise. We assume that $\boldsymbol{w}_i^o$ varies over time according to a first-order random walk model [40]:

$$\boldsymbol{w}_i^o = \boldsymbol{w}_{i-1}^o + \boldsymbol{q}_i \tag{2}$$

where $\boldsymbol{q}_i \in \mathbb{C}^{M\times 1}$ introduces a random perturbation to the parameter vector at time i. In this model, the random-walk sequence, $\{\boldsymbol{q}_i\}$, is assumed to be zero-mean independent and identically distributed (i.i.d.) with positive-definite covariance matrix $Q = \mathbb{E}[\boldsymbol{q}_i\boldsymbol{q}_i^*]$. Considering these properties of the random sequence $\{\boldsymbol{q}_i\}$ and relation (2), we observe that $\boldsymbol{w}_i^o$ has a constant mean and hence $\mathbb{E}[\boldsymbol{w}_i^o] = \mathbb{E}[\boldsymbol{w}_{i-1}^o]$. We assume that the regression data $\boldsymbol{u}_{k,i}$ at each node k are zero-mean wide-sense stationary with covariance $R_{u,k} = \mathbb{E}[\boldsymbol{u}_{k,i}^*\boldsymbol{u}_{k,i}]$. For this model, however, the cross-covariance vector at each node k is time-varying and we, therefore, denote it by $\boldsymbol{r}_{du,k,i} = \mathbb{E}[\boldsymbol{u}_{k,i}^*\boldsymbol{d}_k(i)]$. For mathematical tractability, in our analysis, we assume that $\{\boldsymbol{u}_{k,i}\}$ are i.i.d over time and independent over space with positive definite covariance matrices. The measurement noises $\{\boldsymbol{v}_k(i)\}$ are zero-mean random processes, i.i.d over time and independent over space with variances $\{\sigma_{v,k}^2\}$. The noise $\{\boldsymbol{v}_k(i)\}$ are independent of the regressors $\{\boldsymbol{u}_{m,j}\}$ for all i, j and k, m. These are customary assumptions that have been normally used in the context of distributed adaptive filtering [29], [30], [34], [37], [38].

It can be verified that if the moments $R_{u,k}$ and $\boldsymbol{r}_{du,k,i}$ are available at the fusion center, the minimum mean square error (MMSE) estimate of the parameter vector through a centralized solution, at each time instance i, will be:

$$\boldsymbol{w}_i^o = \Big(\sum_{k=1}^{N} R_{u,k}\Big)^{-1} \sum_{k=1}^{N} \boldsymbol{r}_{du,k,i} \tag{3}$$

If we assume that the parameter vector $\boldsymbol{w}_i^o$ varies slowly, the following centralized LMS algorithm can be used to estimate and track its value:

$$\boldsymbol{w}_i = \boldsymbol{w}_{i-1} + \mu \sum_{k=1}^{N} \boldsymbol{u}_{k,i}^*\big(\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}\boldsymbol{w}_{i-1}\big) \tag{4}$$

where $\mu > 0$ is the step-size parameter.

Motivated by the advantages of distributed algorithm such as energy and communication bandwidth efficiency, the DLMS algorithms [30], [37], [38] can be used to estimate and track the parameter vector $\boldsymbol{w}_i^o$ over the network. The Adapt-then-Combine(ATC) variant of DLMS algorithm will take a similar form to that for the estimation of time-invariant parameters, i.e.:

$$\boldsymbol{\psi}_{k,i} = \boldsymbol{w}_{k,i-1} + \mu_k \sum_{\ell\in\mathcal{N}_k} c_{\ell,k}\boldsymbol{u}_{\ell,i}^*\big(\boldsymbol{d}_\ell(i) - \boldsymbol{u}_{\ell,i}\boldsymbol{w}_{k,i-1}\big) \tag{5}$$

$$\boldsymbol{w}_{k,i} = \sum_{\ell\in\mathcal{N}_k} a_{\ell,k}\boldsymbol{\psi}_{\ell,i} \tag{6}$$

where $\mu_k > 0$ is the local step-size, the vectors $\boldsymbol{\psi}_{k,i}$ and $\boldsymbol{w}_{k,i}$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCNS.2016.2578044, IEEE Transactions on Control of Network Systems

are the intermediate estimates of $\boldsymbol{w}_i^o$ at node k, and $\mathcal{N}_k$ is the set of neighboring nodes (including node k itself) with which node k can share information. In (5) coefficients $c_{\ell,k}$ are non-negatives entries of the right-stochastic matrix $C \in \mathbb{R}^{N\times N}$ where $c_{\ell,k} = 0$ if $\ell \notin \mathcal{N}_k$, and $\sum_{k=1}^{N} c_{\ell,k} = 1$. In (6), $a_{\ell,k}$ are nonnegative coefficients of a left-stochastic matrix $A \in \mathbb{R}^{N\times N}$, which satisfy the following conditions: $a_{\ell,k} = 0$ if $\ell \notin \mathcal{N}_k$, and $\sum_{l\in\mathcal{N}_k} a_{\ell,k} = 1$. There are various methods to obtain matrices C and A [30].

The Combine-then-Adapt (CTA) forms of DLMS [30], [37], [38] can be obtained by reversing the order of adapt and combine steps in (5) and (6) and can be written as:

$$\boldsymbol{\psi}_{k,i-1} = \sum_{\ell\in\mathcal{N}_k} a_{\ell,k}\boldsymbol{w}_{\ell,i-1} \tag{7}$$

$$\boldsymbol{w}_{k,i} = \boldsymbol{\psi}_{k,i-1} - \mu_k \sum_{\ell\in\mathcal{N}_k} c_{\ell,k}\boldsymbol{u}_{\ell,i}^*\big(\boldsymbol{d}_\ell(i) - \boldsymbol{u}_{\ell,i}\boldsymbol{\psi}_{k,i-1}\big) \tag{8}$$

Detailed information about the operation and computational complexity of these algorithms can be found in previous studies, including [30], [34], [37], [38].

III. Network Performance Characterization

In this section, we derive expressions to characterize the network MSD and EMSE of the DLMS algorithms in rapidly time-varying model parameter vector. We will apply the same analysis strategy as used in [30], [38]. The new expressions are however different from those in previous works in that they can characterize the network MSE for relatively large step-size values. As it will be explained in the sequel, the use of large step-sizes in DLMS algorithms is sometimes necessary in order to maintain fast tracking ability should the parameter vector of interest changes quickly.

The general form of DLMS algorithms for time-varying parameter can be written as:

$$\boldsymbol{\phi}_{k,i-1} = \sum_{\ell\in\mathcal{N}_k} a_{\ell,k}^{(1)}\boldsymbol{w}_{\ell,i-1} \tag{9}$$

$$\boldsymbol{\psi}_{k,i} = \boldsymbol{\phi}_{k,i-1} + \mu_k \sum_{\ell\in\mathcal{N}_k} c_{\ell,k}\,\boldsymbol{u}_{\ell,i}^*\big(\boldsymbol{d}_\ell(i) - \boldsymbol{u}_{\ell,i}\boldsymbol{\phi}_{k,i-1}\big) \tag{10}$$

$$\boldsymbol{w}_{k,i} = \sum_{\ell\in\mathcal{N}_k} a_{\ell,k}^{(2)}\boldsymbol{\psi}_{\ell,i} \tag{11}$$

where $a_{\ell,k}^{(1)}$ and $a_{\ell,k}^{(2)}$ are the (ℓ,k) element of left-stochastic combination matrices A_1 and A_2, respectively. We use the general form of DLMS since it includes the CTA and ATC algorithms as special cases. In particular, once the analytical expressions are obtained for all A_1 and A_2, setting $A_1 = I$ and $A_2 = A$ yields the analysis results for ATC and $A_1 = A$ and $A_2 = I$ gives the results for CTA algorithm. In addition, setting $A_1 = A_2 = C = I$ generates the analysis results for DLMS networks in non-cooperative mode.

We begin by introducing the local error vectors at node k, i.e., $\tilde{\boldsymbol{w}}_{k,i} \triangleq \boldsymbol{w}_i^o - \boldsymbol{w}_{k,i}$, $\tilde{\boldsymbol{\psi}}_{k,i} \triangleq \boldsymbol{w}_i^o - \boldsymbol{\psi}_{k,i}$ and $\tilde{\boldsymbol{\phi}}_{k,i} \triangleq \boldsymbol{w}_i^o - \boldsymbol{\phi}_{k,i}$. These are used to define the global error vectors of the network, i.e., $\tilde{\boldsymbol{\phi}}_i \triangleq \text{col}\{\tilde{\boldsymbol{\phi}}_{1,i},\cdots,\tilde{\boldsymbol{\phi}}_{N,i}\}$, $\tilde{\boldsymbol{\psi}}_i \triangleq \text{col}\{\tilde{\boldsymbol{\psi}}_{1,i}\cdots,\tilde{\boldsymbol{\psi}}_{N,i}\}$, and $\tilde{\boldsymbol{w}}_i \triangleq \text{col}\{\tilde{\boldsymbol{w}}_{1,i},\cdots,\tilde{\boldsymbol{w}}_{N,i}\}$. We further define the extended weighting matrices $\mathcal{A}_2 \triangleq A_2\otimes I_M$ $\mathcal{A}_1 \triangleq A_1\otimes I_M$ and $\mathcal{C} \triangleq C\otimes I_M$. Next, we introduce the following matrices and vectors:

$$\boldsymbol{\mathcal{R}}_i \triangleq \sum_{\ell=1}^{N} \text{diag}\Big\{c_{\ell,1}\,\boldsymbol{u}_{\ell,i}^*\boldsymbol{u}_{\ell,i},\cdots,c_{\ell,N}\,\boldsymbol{u}_{\ell,i}^*\boldsymbol{u}_{\ell,i}\Big\} \tag{12}$$

$$\mathcal{M} \triangleq \text{diag}\Big\{\mu_1 I_M,\cdots,\mu_N I_M\Big\} \tag{13}$$

$$\boldsymbol{g}_i \triangleq \mathcal{C}^T\text{col}\big\{\boldsymbol{u}_{1,i}^*\boldsymbol{v}_1(i),\cdots,\boldsymbol{u}_{N,i}^*\boldsymbol{v}_N(i)\big\} \tag{14}$$

$$\boldsymbol{p}_i \triangleq \mathbb{1}_N\otimes\boldsymbol{q}_i \tag{15}$$

where $\mathbb{1}_N$ is a column vector with size N and unit entries. Using these definitions and subtracting $\boldsymbol{w}_i^o$ from both sides (9)-(11), we then arrive at:

$$\begin{aligned}\tilde{\boldsymbol{w}}_i =&\mathcal{A}_2^T(I-\mathcal{M}\boldsymbol{\mathcal{R}}_i)\mathcal{A}_1^T\tilde{\boldsymbol{w}}_{i-1} - \mathcal{A}_2^T\mathcal{M}\boldsymbol{g}_i\\ &+\mathcal{A}_2^T(I-\mathcal{M}\boldsymbol{\mathcal{R}}_i)\mathcal{A}_1^T\boldsymbol{p}_i\end{aligned} \tag{16}$$

which shows the evolution of the network error vector over time. We note that the instantaneous network error vector (16) has one additional error term compared with the standard DLMS that was reported in [30], [38], i.e., $\boldsymbol{e}_i = \mathcal{A}_2^T(I - \mathcal{M}\boldsymbol{\mathcal{R}}_i)\mathcal{A}_1^T\boldsymbol{p}_i$. Under previous assumption that $\boldsymbol{q}_i$ and $\boldsymbol{v}_k(i)$ are zero-mean and independent of $\boldsymbol{u}_{k,i}$ for all k, we obtain $\mathbb{E}[\boldsymbol{p}_i] = \mathbb{E}[\boldsymbol{e}_i] = \mathbb{E}[\boldsymbol{g}_i] = 0$. Considering these equalities and taking the expectation of the global error vector $\tilde{\boldsymbol{w}}_i$ in (16), we arrive at:

$$\mathbb{E}[\tilde{\boldsymbol{w}}_i] = \mathcal{A}_2^T(I-\mathcal{M}\mathcal{R})\mathcal{A}_1^T\mathbb{E}[\tilde{\boldsymbol{w}}_{i-1}] \tag{17}$$

where $\mathcal{R} \triangleq \mathbb{E}[\boldsymbol{\mathcal{R}}_i] = \sum_{\ell=1}^{N}\text{diag}\big\{c_{\ell,1}\,R_{u,\ell},\ldots,c_{\ell,N}\,R_{u,\ell}\big\}$. The expression (17) is similar to the expression of the mean error vector of standard DLMS. Therefore, the mean stability condition of standard DLMS algorithm, as developed in [30], [38], remains valid under the non-stationary condition considered in this work. Therefore, for the algorithm to be stable in the mean, the step-size at each node k must satisfy

$$0 < \mu_k < \frac{2}{\rho\Big(\sum_\ell c_{\ell,k}R_{u,\ell}\Big)} \qquad k\in\{1,\ldots,N\} \tag{18}$$

We will use (18) to find the optimal step-size parameters of the network in the following section.

A. Steady-State Mean-Square Performance

To characterize the mean-square performance of the DLMS algorithms in non-stationary signal environments, we use (16) and form the variance relation by computing the weighted-squared norm of (16). Then, taking the expectations on both sides:

$$\begin{aligned}\text{E}\|\tilde{\boldsymbol{w}}_i\|_\Sigma^2 =&\mathbb{E}\|\tilde{\boldsymbol{w}}_{i-1}\|_{\boldsymbol{\Sigma}'}^2 + \mathbb{E}\|\boldsymbol{p}_i\|_{\boldsymbol{\Sigma}'}^2\\ &+\mathbb{E}[\boldsymbol{g}_i^*\mathcal{M}^T\mathcal{A}_2\Sigma\mathcal{A}_2^T\mathcal{M}\boldsymbol{g}_i]\end{aligned} \tag{19}$$

where $\Sigma > 0$ is a weighting matrix with compatible dimension that we are free to choose and $\boldsymbol{\Sigma}' = \mathcal{A}_1(I-\mathcal{M}\boldsymbol{\mathcal{R}}_i)^*\mathcal{A}_2\Sigma\mathcal{A}_2^T(I-\mathcal{M}\boldsymbol{\mathcal{R}}_i)\mathcal{A}_1^T$, $\mathbb{E}\|\boldsymbol{p}_i\|_{\Sigma'}^2 = \mathbb{E}[\boldsymbol{p}_i^*\Sigma'\boldsymbol{p}_i] = \text{Tr}(\Sigma'\mathcal{P})$, $\Sigma' = \mathbb{E}[\boldsymbol{\Sigma}']$ and $\mathcal{P} = \mathbb{E}[\boldsymbol{p}_i\boldsymbol{p}_i^*]$. We obtain (19) by using the fact that $\tilde{\boldsymbol{w}}_{i-1}$ is independent of $\boldsymbol{g}_i$ and $\boldsymbol{p}_i$, and $\mathbb{E}[\boldsymbol{g}_i] = \mathbb{E}[\boldsymbol{p}_i] = 0$ according to the network data statistic presented in Section II. Relation (19) can be written

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCNS.2016.2578044, IEEE Transactions on Control of Network Systems

as:

$$\mathbb{E}\|\tilde{\boldsymbol{w}}_i\|_{\Sigma}^2 = \mathbb{E}\|\tilde{\boldsymbol{w}}_{i-1}\|_{\Sigma'}^2 + \mathrm{Tr}(\Sigma\mathcal{A}_2^T\mathcal{M}\mathcal{G}\mathcal{M}\mathcal{A}_2) + \mathrm{Tr}(\Sigma'\mathcal{P}) \tag{20}$$

where $\mathcal{G} \triangleq \mathbb{E}[\boldsymbol{g}_i^*\boldsymbol{g}_i]$ which can be written as $\mathcal{G} = \mathcal{C}^T\mathrm{diag}\{\sigma_{v,1}^2R_{u,1},\ldots,\sigma_{v,N}^2R_{u,N}\}\mathcal{C}$. Introducing $\sigma \triangleq \mathrm{bvec}(\Sigma)$, and $\sigma' \triangleq \mathrm{bvec}(\Sigma')$, the variance relation in (20), becomes:

$$\begin{aligned}\mathbb{E}\|\tilde{\boldsymbol{w}}_i\|_{\sigma}^2 =& \mathbb{E}\|\tilde{\boldsymbol{w}}_{i-1}\|_{\sigma'}^2 + \big[\mathrm{bvec}(\mathcal{A}_2^T\mathcal{M}\mathcal{G}\mathcal{M}\mathcal{A}_2)\big]^T\sigma \\ &+ \big[\mathrm{bvec}(\mathcal{P}^T)\big]^T\sigma'\end{aligned} \tag{21}$$

where $\sigma' = \mathcal{F}\sigma$ and $\mathcal{F}$ is computed below for two special cases: small step-size approximation and exact closed-form evaluation using Gaussian distribution assumption. For the first case, using the property of conditional expectation together with the independence of $\tilde{\boldsymbol{w}}_{i-1}$ and $\boldsymbol{\mathcal{R}}_i$, we can extend the multiplication terms in $\boldsymbol{\Sigma}'$ to obtain:

$$\begin{aligned}\mathbb{E}[\boldsymbol{\Sigma}'] =& \mathcal{A}_1\mathcal{A}_2\Sigma\mathcal{A}_2^T\mathcal{A}_1^T - \mathcal{A}_1\mathcal{R}\mathcal{M}\mathcal{A}_2\Sigma\mathcal{A}_2^T\mathcal{A}_1^T \\ &- \mathcal{A}_1\mathcal{A}_2\Sigma\mathcal{A}_2^T\mathcal{M}\mathcal{R}\mathcal{A}_1^T + O(\mathcal{M}^2)\end{aligned} \tag{22}$$

where

$$O(\mathcal{M}^2) = \mathbb{E}[\mathcal{A}_1\boldsymbol{\mathcal{R}}_i\mathcal{M}\mathcal{A}_2\Sigma\mathcal{A}_2^T\mathcal{M}\boldsymbol{\mathcal{R}}_i\mathcal{A}_1^T] \tag{23}$$

For small step-sizes this term can be neglected, since it depends on $\{\mu_k^2\}$. Considering this and using $\sigma' = \mathcal{F}\sigma$, we obtain:

$$\mathcal{F} \approx (\mathcal{A}_1 \otimes_b \mathcal{A}_1)(I - I \otimes_b \mathcal{R}\mathcal{M} - \mathcal{R}^T\mathcal{M} \otimes_b I)(\mathcal{A}_2 \otimes_b \mathcal{A}_2) \tag{24}$$

As will be demonstrated below, however, this approximation causes large errors if relatively large step-sizes are chosen. In what follows, to circumvent this limitation, we derive an exact closed-form expression for $\mathcal{F}$ that is required to characterize the network behavior in the estimation and tracking of rapidly varying parameter vectors. To make the derivation simpler we assume that the distribution of the regression data $\boldsymbol{u}_{k,i}$ is Gaussian.

Remark 1. *In the estimation of time-invariant parameters in adaptive networks, it has been shown that the MSD decreases as the optimization step-size parameters become smaller [30], [38]. Moreover, choosing small step-sizes will only lead to a lower convergence speed. The situation is however different if the the parameter vector of interest varies over time. In this case, the step-sizes cannot be chosen arbitrarily small. This can be explained using the adaptation step (10) in general DLMS algorithms. If at each node k, the step-size μ_k is chosen to be very small then the innovation term*

$$\Delta\boldsymbol{\phi}_{k,i} = \mu_k \sum_{\ell\in\mathcal{N}_k} c_{\ell,k}\,\boldsymbol{u}_{\ell,i}^*\big(\boldsymbol{d}_\ell(i) - \boldsymbol{u}_{\ell,i}\boldsymbol{\phi}_{k,i-1}\big) \tag{25}$$

which is used to update $\boldsymbol{\phi}_{k,i}$ in (10) also becomes very small. In turn, this adversely affects the tracking performance of the algorithm when the perturbation term $\boldsymbol{q}_i$ in (2) is relatively large. This is because if the change from $\boldsymbol{w}_{i-1}$ to $\boldsymbol{w}_i$ is large due to large random walk $\boldsymbol{q}_i$, the DLMS algorithms will need to proportionally add large $\Delta\boldsymbol{\phi}_{k,i}$ to track the change. However, when the step-sizes μ_k are chosen very small, the innovation term, $\Delta\boldsymbol{\phi}_{k,i}$, become small and the algorithms fail to track changes. Therefore, in the estimation of time-varying parameters in adaptive networks, the step-sizes of DLMS algorithms cannot be chosen arbitrarily small such that term $O(\mathcal{M}^2)$ in (22) can be neglected. This motivates the derivation of an exact closed-form expression for $\mathcal{F}$ to properly characterize the MSE of DLMS algorithms in tracking time-varying parameters.

To find an exact closed-form expression to evaluate $\mathcal{F}$, we need to compute the term given in (23). As shown in Appendix A, if we assume that the regressors $\boldsymbol{u}_{k,i}$ are zero-mean circular complex Gaussian random vectors, $\mathcal{F}$ will be:

$$\begin{aligned}\mathcal{F} =& (\mathcal{A}_1 \otimes_b \mathcal{A}_1)\Big(I - I \otimes_b \mathcal{R}\mathcal{M} - \mathcal{R}^T\mathcal{M} \otimes_b I\Big) \\ &\times (\mathcal{A}_2 \otimes_b \mathcal{A}_2) + \Delta\mathcal{F}\end{aligned} \tag{26}$$

where $\Delta\mathcal{F}$ is given in (27). In this expression, $\beta = 2$ for the real-valued data, while $\beta = 1$ for the complex-valued data. Now the steady-state MSD and EMSE expressions of the network can be obtained from (21) as follows. By definition, the MSD at node k is: $\eta(k) = \lim_{i\to\infty}\mathbb{E}\|\tilde{\boldsymbol{w}}_{k,i}\|^2$, which alternatively, can be retrieved from the global error vector $\tilde{\boldsymbol{w}}_i$ as

$$\eta(k) = \lim_{i\to\infty}\mathbb{E}\|\tilde{\boldsymbol{w}}_i\|_{\mathrm{diag}(e_k)\otimes I_M}^2 \tag{28}$$

where $e_k \in \mathbb{Z}^{N\times 1}$ is a column vector with one at position k and zeros elsewhere. We use (20) to write:

$$\begin{aligned}\lim_{i\to\infty}\mathbb{E}\|\tilde{\boldsymbol{w}}_i\|_{(I-\mathcal{F})\sigma}^2 =& \\ &\Big(\big[\mathrm{bvec}(\mathcal{A}_2^T\mathcal{M}\mathcal{G}^T\mathcal{M}\mathcal{A}_2)\big]^T + \big[\mathrm{bvec}(\mathcal{P}^T)\big]^T\mathcal{F}\Big)\sigma\end{aligned} \tag{29}$$

Using (28) and (29), we obtain:

$$\eta(k) = \Big(\mathrm{bvec}(\mathcal{A}_2^T\mathcal{M}\mathcal{G}^T\mathcal{M}\mathcal{A}_2)^T + \mathrm{bvec}(\mathcal{P}^T)^T\mathcal{F}\Big)\sigma_{\mathrm{msd}_k} \tag{30}$$

where $\sigma_{\mathrm{msd}_k} = (I-\mathcal{F})^{-1}\mathrm{bvec}(\mathrm{diag}(e_k)\otimes I_M)$. Note that $(I-\mathcal{F})$ is nonsingular since $\rho(\mathcal{F}) < 1$. In the same way, we can compute the EMSE for each node k, starting from its

$$\begin{aligned}\Delta\mathcal{F} = (\mathcal{A}_1 \otimes_b \mathcal{A}_1)\Bigg\{&(\mathcal{R}^T \otimes_b \mathcal{R}) + \sum_{m=1}^{N}\Big[\mathrm{diag}\{\mathrm{vec}(C_m\mathbb{1}_{N\times N}C_m)\}\Big] \\ &\otimes\Big[(\beta-1)(R_{u,m}^T\otimes R_{u,m}) + r_m r_m^*\Big]\Bigg\}(\mathcal{M}\otimes_b\mathcal{M})(\mathcal{A}_2\otimes_b\mathcal{A}_2)\end{aligned} \tag{27}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCNS.2016.2578044, IEEE Transactions on Control of Network Systems

definition, i.e., $\zeta(k) = \lim_{i\to\infty} \mathbb{E}\|\tilde{\boldsymbol{w}}_i\|^2_{\text{diag}(e_k)\otimes R_{u,k}}$, which finally lead us to

$$\zeta(k) = \Big(\text{bvec}(\mathcal{A}_2^T \mathcal{M}\mathcal{G}^T \mathcal{M}\mathcal{A}_2)^T + \text{bvec}(\mathcal{P}^T)^T \mathcal{F}\Big)\sigma_{\text{emse}_k} \tag{31}$$

with $\sigma_{\text{emse}_k} = (I-\mathcal{F})^{-1}\text{bvec}(\text{diag}(e_k)\otimes R_{u,k})$.

B. Mean-Square Transient Analysis

Starting from (21) we obtain the following expression to characterize the mean-square behavior of the algorithm in transient-state:

$$\mathbb{E}\|\tilde{\mathbf{w}}_i\|^2_\sigma = \mathbb{E}\|\tilde{\mathbf{w}}_{i-1}\|^2_\sigma + \|w^o_{-1}\|^2_{\mathcal{F}^i(I-\mathcal{F})\sigma} + s^T\mathcal{F}^i\sigma. \tag{32}$$

where $s = \big[\text{bvec}(\mathcal{A}_2^T\mathcal{M}\mathcal{G}^T\mathcal{M}\mathcal{A})\big] + \mathcal{F}^T\big[\text{bvec}(\mathcal{P}^T)\big]$. By, respectively, replacing σ with $\text{bvec}\big(\text{diag}(e_k\otimes I_m)\big)$ and $\text{bvec}\big(\text{diag}(e_k\otimes R_{u,k})\big)$ into (32), we then arrive at two expressions for the evolution of MSD and EMSE, at each node k. Note that to obtain these expressions, we consider zero initialization at each node over the network, i.e., $\boldsymbol{w}_{k,-1} = 0$ for all k. The steady-state and transient network MSD are, respectively, defined as the average of the steady-state and transient MSD of the nodes. The same holds for the network steady-state and transient EMSE. These results will be used in Section V to demonstrate the transient MSD and EMSE performance of diffusion-LMS networks.

IV. Tracking Performance and Optimal Step-sizes

In this section, we examine the tracking performance of DLMS algorithms in non-stationary signal environments for both cooperative and non-cooperative networks. We also investigate how an optimal choice of step-size parameter vector can enhance the tracking ability of these algorithms and minimize their steady-state MSD performance. In particular, we obtain optimal step-sizes of the DLMS algorithms by defining a constrained nonlinear optimization problem and solving it through a log-barrier Newton algorithm.

A. Optimal Step-Size in a Non-cooperative Network

The MSD and EMSE of each node k in a non-cooperative network can be obtained from (30) and (31) by setting

$$A_1 = A_2 = C = I_N \tag{33}$$

For this scenario, these expressions more explicitly can be rewritten as:

$$\eta(k) = \big(\mu_k^2\,\sigma_{v,k}^2\,r_k^* + b^*\bar{\mathcal{F}}_k\big)(I-\bar{\mathcal{F}}_k)^{-1}m_k \tag{34}$$

$$\zeta(k) = \big(\mu_k^2\,\sigma_{v,k}^2\,r_k^* + b^*\bar{\mathcal{F}}_k\big)(I-\bar{\mathcal{F}}_k)^{-1}r_k \tag{35}$$

where $b = \text{vec}(Q)$, $r_k = \text{vec}(R_{u,k})$, $m_k = \text{vec}(I_M)$ and

$$\bar{\mathcal{F}}_k = I - 2\mu_k\,R_{u,k}^T\otimes I_M + \mu_k^2\big[\beta(R_{u,k}^T\otimes R_{u,k}) + r_k r_k^*\big] \tag{36}$$

We call signal $\boldsymbol{d}_k(i)$ slowly time-varying if $\sigma_q^2 \ll \sigma_{v,k}^2$ otherwise refer to it rapidly time-varying.

We now proceed to explain why the exact closed-form expression (26) needs to be taken into account in finding the optimal value of the step-sizes over networks with rapidly time-varying signals. For simplicity, we only present the MSD performance results for a single node over the network. To perform these numerical experiments, we use the DLMS algorithm in the non-cooperative mode ($i.e., A_1 = A_2 = C = I$). The regression data and the measurement noise are circular complex-valued Gaussian. Fig. 1 and 2 show the steady-state MSD of node k after 6000 iteration against the step-size μ_k. For small step-size approximation, we have

$$\bar{\mathcal{F}}_k \approx I - 2\mu_k\,R_{u,k}^T\otimes I_M \tag{37}$$

while the exact $\bar{\mathcal{F}}_k$ value is given by (36). As it is shown in Fig. 1, in slowly time-varying signal environment where $\text{Tr}(Q) = M\sigma_q^2 = 0.005\sigma_{v,k}^2$, the optimal values of the step-size that give the minimum MSD at node k are identical for the exact value and small step-size approximations of $\bar{\mathcal{F}}_k$. This will be also the case for the optimal choice of the step-size in the steady-state EMSE curve which is not shown here. In contrast, as shown in Fig. 2, in the rapidly time-varying signal environment where $\text{Tr}(Q) = M\sigma_q^2 = 0.5\sigma_{v,k}^2$, the optimal values of the step-sizes obtained from the small step-size approximation are far away from the correct values. Therefore, to compute the optimal step-size parameters μ_k in rapidly

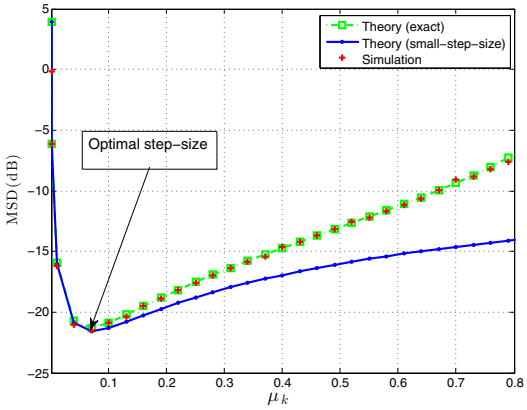

Fig. 1: Steady-state MSD performance of node k in a slowly time-varying environment where $\sigma_q^2 = 0.0025\sigma_{v,k}^2$.

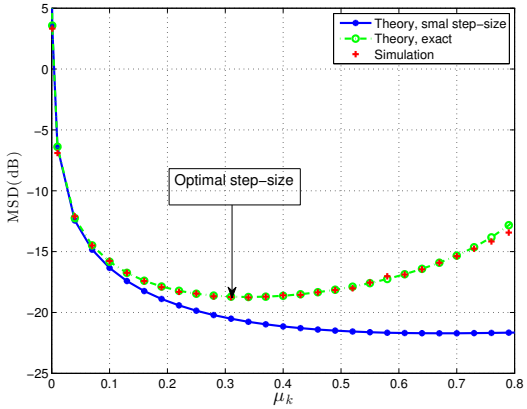

Fig. 2: Steady-state MSD performance of node k in rapidly time-varying environments where $\sigma_q^2 = 0.25\sigma_{v,k}^2$. The minimum point of the solid curve (theory with approximation) does not match with that of cross-marked curve (simulation result).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCNS.2016.2578044, IEEE Transactions on Control of Network Systems

varying signal environments, the terms depending on μ_k^2 in $\bar{\mathcal{F}}_k$ cannot be ignored.

As it is seen from (34) and (35), in fast non-stationary signal environments, the random walk $\boldsymbol{q}_i$ and $\Delta\mathcal{F}$ depend on μ_k^2 in $\bar{\mathcal{F}}_k$, which make the MSD and EMSE of each node k nonlinear functions of the step-sizes μ_k[1]. Therefore, unlike the stationary signal environment, the MSD and EMSE performance does not improve as step-sizes approach zero. In fact, in non-stationary scenario, there exists an optimal step-size in the stability range of the algorithm for which the MSD and EMSE are minimized. The optimal step-size that minimizes the MSD of each node can be found by solving the following optimization problem:

$$\mu^o_{\mathrm{msd}_k} = \arg\min_{\mu_k} \eta(k,\mu) \quad \text{s.t} \quad 0 < \mu_k < \frac{2}{\mathrm{Tr}\Big(\sum_{\ell\in\mathcal{N}_k} c_{\ell,k} R_{u,\ell}\Big)} \tag{38}$$

The constraint in (38) is obtained from the DLMS stability range (18). For slowly time-varying signals, a closed-form solution can be obtained by using the small step-size approximation in the evaluation of $\bar{\mathcal{F}}_k$. To this end, we use (34) and (35) to obtain:

$$\eta(k) = \mu_k\,\sigma^2_{v,k}\,r^*_k\big(R^T_{u,k}\otimes R_{u,k}\big)^{-1}m_k/2 + b^*\bar{\mathcal{F}}_k\big(R^T_{u,k}\otimes R_{u,k}\big)^{-1}m_k/2\mu_k \tag{39}$$

$$\zeta(k) = \mu_k\,\sigma^2_{v,k}\,r^*_k\big(R^T_{u,k}\otimes R_{u,k}\big)^{-1}r_k/2 + b^*\bar{\mathcal{F}}_k\big(R^T_{u,k}\otimes R_{u,k}\big)^{-1}r_k/2\mu_k \tag{40}$$

It can be verified that (39) and (40) are convex with respect to $\mu_k \in \{\mu_1,\ldots,\mu_N\}$[2]. Therefore, computing the derivatives of MSD and EMSE with respect to μ_k and forcing them to zero yields, respectively:

$$\mu^o_{\mathrm{msd}_k} = \sqrt{\frac{b^*(R^T_{u,k}\otimes I_M)^{-1}m_k}{\sigma^2_{v,k}r^*_k(R^T_{u,k}\otimes I_M)^{-1}m_k}} \tag{41}$$

$$\mu^o_{\mathrm{emse}_k} = \sqrt{\frac{b^*(R^T_{u,k}\otimes I_M)^{-1}r_k}{\sigma^2_{v,k}r^*_k(R^T_{u,k}\otimes I_M)^{-1}r_k}} \tag{42}$$

These expressions give the optimal step-size values that lead to minimum steady-state MSD and EMSE at a given node k in slowly-varying signal environments in non-cooperative mode. These results are in agreement with the optimal step-size parameter obtained for the stand-alone LMS filter in [41]. In rapidly varying signal environments, these expressions are invalid as explained. Alternatively, we propose to use a constrained iterative optimization method, specifically, the log-barrier Newton method [42], to obtain the local optimal step-sizes over the network.

B. Optimal Step-Sizes of DLMS Algorithms

We now propose an algorithm to obtain the optimal step-sizes of DLMS algorithms that leads to lower estimation error and faster tracking ability in the network. We begin this part by illustrating the MSD performance of a simple network with $N = 2$ against variation of the step-size parameters (see Fig. 3). As shown in this figure, there exists an optimal step-size parameter vector at which DLMS algorithm attains its minimum MSD. Similar to the non-cooperative LMS algorithm, in non-stationary signal environments, the steady-state MSD and EMSE of DLMS algorithms increase as the step-sizes move away from their optimal points. A similar behavior is seen in network with multiple nodes. We can find the optimal adaptation step-size vector of the network by solving the following optimization problem:

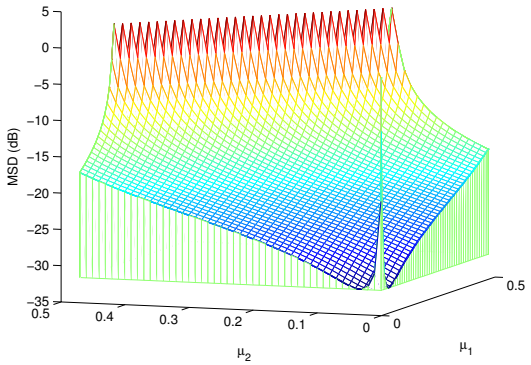

Fig. 3: Steady-State MSD performance of a two-node network as a function of μ_1 and μ_2.

$$(\mu^o_1,\cdots,\mu^o_N) = \arg\min_{(\mu_1,\cdots,\mu_N)} \Big\{(\gamma+\alpha\mathcal{F})\frac{1}{N}\sum_{k=1}^{N}\sigma_{\mathrm{msd}_k}\Big\} \quad \text{s.t} \quad 0<\mu_k<\frac{2}{\mathrm{Tr}\Big(\sum_{\ell\in\mathcal{N}_k} c_{\ell,k}R_{u,\ell}\Big)} \tag{43}$$

where $k = \{1,2,\ldots,N\}$, $\alpha = \mathrm{bvec}(\mathcal{P}^T)^T$, and $\gamma = \mathrm{bvec}(\mathcal{A}_2^T\mathcal{M}\mathcal{G}^T\mathcal{M}\mathcal{A}_2)^T$. Note that the constraint in (43) guarantees that the obtained solution will satisfy the stability condition (18). A similar optimization problem can be formed for minimization of the EMSE over the network. The nonlinear constrained optimization problem (43) can be converted to an unconstrained optimization problem by adding a log-barrier function to the objective, i.e.,

$$(\mu^o_1,\cdots,\mu^o_N) = \arg\min_{(\mu_1,\cdots,\mu_N)} L(\boldsymbol{\mu}) \tag{44}$$

where

$$L(\boldsymbol{\mu}) = (\gamma+\alpha\mathcal{F})\frac{1}{N}\sum_{k=1}^{N}\sigma_{\mathrm{msd}_k} - \frac{1}{t}\sum_{k=1}^{N}\Big[\log(\mu_k)+\log\Big(\frac{2}{\mathrm{Tr}\sum_{\ell\in\mathcal{N}_k}c_{\ell,k}R_{u,\ell}}-\mu_k\Big)\Big] \tag{45}$$

and $t > 0$ controls the accuracy of the solution. The solution of the log-barrier method, denoted by $\boldsymbol{\mu}$, is suboptimal and it is away from the optimal point by $\frac{N}{t}$. Therefore, as $t\to\infty$, it

[1]It can be verified that if we ignore μ_k^2 in $\mathcal{F}_k$, the MSD and EMSE will be linear functions of μ_k whose value monotonically decreases as μ_k become smaller.

[2]In general, $\eta(k)$ and $\zeta(k)$ may not be convex for all network topologies and signal conditions.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCNS.2016.2578044, IEEE Transactions on Control of Network Systems

approaches the solution[1] of the original problem (43) denoted by $\mu^o = (\mu_1^o, \ldots, \mu_N^o)$. Here, we introduce the threshold ϵ as a desired stoping criterion of the following interior point method optimization algorithm to find the solution. The nominal values for ϵ and t are, respectively, 10^{-7} and 0.01. In this approach, summarized under Algorithm 1, $f(\boldsymbol{\mu}, t)$ can be any iterative gradient algorithms, such as the steepest descent or the Newton algorithm. In our implementation of Algorithm 1, we employ

Algorithm 1 : Step-size optimization

while $\dfrac{N}{t} > \epsilon$

$\boldsymbol{\mu}_i = f(\boldsymbol{\mu}_{i-1}, t)$ (Update the estimated step-size)

$t = 2t$ (Increase the threshold)

Newton algorithm as it offers a fast convergence speed. Note that at each iteration i the solution is away from the optimal point by $\frac{N}{t}$. We double the value of t at each iteration and use the previous solution of the Newton algorithm to eventually approach the optimal step-size parameters. In this implementation, for each i, several Newton iterations, denoted by j, are performed before doubling the value of t. The criterion to exit Newton iteration will be given at the end of this section. The iterative Newton algorithm can be represented as:

$$\boldsymbol{\mu}_j = \boldsymbol{\mu}_{j-1} - \alpha\Big(\nabla_\mu^2 L(\boldsymbol{\mu}_{j-1})\Big)^{-1} \nabla_\mu L(\boldsymbol{\mu}_{j-1}) \tag{46}$$

where $\alpha > 0$ is the Newton step-size, and $\nabla_\mu L(\boldsymbol{\mu}_{j-1})$ and $\nabla_\mu^2 L(\boldsymbol{\mu}_{j-1})$, respectively, are the gradient and Hessian matrices of the the augmented MSD function, $L(\boldsymbol{\mu})$, with respect to $\boldsymbol{\mu}$:

$$\nabla_\mu L(\boldsymbol{\mu}) \triangleq [\frac{\partial L(\boldsymbol{\mu})}{\partial \mu_1}, \frac{\partial L(\boldsymbol{\mu})}{\partial \mu_2}, \cdots, \frac{\partial L(\boldsymbol{\mu})}{\partial \mu_N}]^T \tag{47}$$

$$\nabla_\mu^2 L(\boldsymbol{\mu}) \triangleq \begin{bmatrix} \frac{\partial^2 L(\boldsymbol{\mu})}{\partial^2 \mu_1} & \frac{\partial^2 L(\boldsymbol{\mu})}{\partial \mu_1 \partial \mu_2} & \cdots & \frac{\partial^2 L(\boldsymbol{\mu})}{\partial \mu_1 \partial \mu_N} \\ \frac{\partial^2 L(\boldsymbol{\mu})}{\partial \mu_2 \mu_1} & \frac{\partial^2 L(\boldsymbol{\mu})}{\partial^2 \mu_2} & \cdots & \frac{\partial^2 L(\boldsymbol{\mu})}{\partial \mu_2 \partial \mu_N} \\ \vdots & \cdots & \cdots & \vdots \\ \frac{\partial^2 L(\boldsymbol{\mu})}{\partial \mu_N \mu_1} & \frac{\partial^2 L(\boldsymbol{\mu})}{\partial \mu_N \partial \mu_1} & \cdots & \frac{\partial^2 L(\boldsymbol{\mu})}{\partial^2 \mu_N} \end{bmatrix} \tag{48}$$

To improve the convergence speed of the Newton algorithm, a line search algorithm such as Armigo rule [43] can be implemented to compute the value of α at each iteration. Nevertheless, implementing a line search algorithm will be more useful for algorithms with slow convergence speed such as the gradient descent. Introducing $l_k \triangleq 2/(\mathrm{Tr}\sum_{\ell \in \mathcal{N}_k} c_{\ell,k} R_{u,\ell})$, the enteries of the first order gradient are computed as:

$$\frac{\partial L(\boldsymbol{\mu})}{\partial \mu_n} = \frac{1}{N}\Big\{\varrho \frac{\partial \mathcal{F}(\boldsymbol{\mu})}{\partial \mu_n} + \frac{\partial \gamma(\boldsymbol{\mu})}{\partial \mu_n}\Big\} \sum_{k=1}^{N} \sigma_{\mathrm{msd}_k} - \frac{1}{t}\big(\frac{1}{\mu_n} - \frac{1}{l_n - \mu_n}\big) \tag{49}$$

[1]The optimization problem (43) is not convex in general. The term "solution" is used to refer to one of the local stationary point of this function within the stability range of DLMS that leads to a lower MSD.

where $\varrho = \alpha + (\alpha\mathcal{F} + \gamma)(I - \mathcal{F})^{-1}$ and the partial derivatives are:

$$\frac{\partial \mathcal{F}(\boldsymbol{\mu})}{\partial \mu_n} = -(\mathcal{A}_1 \otimes_b \mathcal{A}_1)\Big[(\mathcal{R}^T \frac{\partial \mathcal{M}(\boldsymbol{\mu})}{\partial \mu_n}) \otimes_b (I - \mathcal{R}\mathcal{M}) + (I - \mathcal{R}^T\mathcal{M}) \otimes_b \mathcal{R}\frac{\partial \mathcal{M}(\boldsymbol{\mu})}{\partial \mu_n}\Big](\mathcal{A} \otimes_b \mathcal{A}) \tag{50}$$

$$\frac{\partial \gamma(\boldsymbol{\mu})}{\partial \mu_n} = \big[\mathrm{bvec}(\mathcal{A}_2^T \frac{\partial \mathcal{M}(\boldsymbol{\mu})}{\partial \mu_n} \mathcal{G}^T \mathcal{M}\mathcal{A}) + \mathrm{bvec}(\mathcal{A}_2^T \mathcal{M}\mathcal{G}^T \frac{\partial \mathcal{M}(\boldsymbol{\mu})}{\partial \mu_n}\mathcal{A}_2)\big]^T \tag{51}$$

$$\frac{\partial \mathcal{M}(\boldsymbol{\mu})}{\partial \mu_n} = \mathrm{diag}\{\mathbf{0}, \cdots, I_M, \cdots, \mathbf{0}\} \tag{52}$$

The elements of the Hessian matrix are computed as:

$$\frac{\partial^2 L(\boldsymbol{\mu})}{\partial \mu_m \partial \mu_n} = \Big\{\frac{\partial \varrho(\boldsymbol{\mu})}{\partial \mu_m}\frac{\partial \mathcal{F}(\boldsymbol{\mu})}{\partial \mu_n} + \varrho\frac{\partial^2 \mathcal{F}(\boldsymbol{\mu})}{\partial \mu_m \mu_n} + \frac{\partial^2 \gamma(\boldsymbol{\mu})}{\partial \mu_m \mu_n} - (\varrho\frac{\partial \mathcal{F}(\boldsymbol{\mu})}{\partial \mu_n} + \frac{\partial \gamma(\boldsymbol{\mu})}{\partial \mu_n})(I - \mathcal{F})^{-1}\frac{\partial \mathcal{F}(\boldsymbol{\mu})}{\partial \mu_m}\Big\}\frac{1}{N}\sum_{k=1}^{N}\sigma_{\mathrm{msd}_k} + \delta_{nm}\frac{1}{t}\big(\frac{l}{\mu_n^2} - \frac{1}{(l_n - \mu_n)^2}\big) \tag{53}$$

where

$$\frac{\partial \varrho(\boldsymbol{\mu})}{\partial \mu_m} = (\alpha\frac{\partial \mathcal{F}(\boldsymbol{\mu})}{\partial \mu_m} + \frac{\partial \gamma(\boldsymbol{\mu})}{\partial \mu_m})(I - \mathcal{F})^{-1} - (\alpha\mathcal{F} + \gamma)(I - \mathcal{F})^{-1}\frac{\partial \mathcal{F}(\boldsymbol{\mu})}{\partial \mu_m}(I - \mathcal{F})^{-1} \tag{54}$$

$$\frac{\partial^2 \mathcal{F}(\boldsymbol{\mu})}{\partial \mu_m \mu_n} = (\mathcal{A}_1 \otimes_b \mathcal{A}_1)\big[(\mathcal{R}^T \frac{\partial \mathcal{M}(\boldsymbol{\mu})}{\partial \mu_n}) \otimes_b (\mathcal{R}\frac{\partial \mathcal{M}(\boldsymbol{\mu})}{\partial \mu_m}) + (\mathcal{R}\frac{\partial \mathcal{M}(\boldsymbol{\mu})}{\partial \mu_m}) \otimes_b \mathcal{R}\frac{\partial \mathcal{M}(\boldsymbol{\mu})}{\partial \mu_n}\big](\mathcal{A} \otimes_b \mathcal{A}) \tag{55}$$

$$\frac{\partial^2 \gamma(\boldsymbol{\mu})}{\partial \mu_m \mu_n} = \Big[\mathrm{bvec}(\mathcal{A}_2^T \frac{\partial \mathcal{M}(\boldsymbol{\mu})}{\partial \mu_n}\mathcal{G}^T \frac{\partial \mathcal{M}(\boldsymbol{\mu})}{\partial \mu_m}\mathcal{A}) + \mathrm{bvec}(\mathcal{A}_2^T \frac{\partial \mathcal{M}(\boldsymbol{\mu})}{\partial \mu_m}\mathcal{G}^T \frac{\partial \mathcal{M}(\boldsymbol{\mu})}{\partial \mu_n}\mathcal{A}_2)\Big]^T \tag{56}$$

The stopping criterion for the Newton iteration will be

$$\nabla_\mu L(\boldsymbol{\mu}_j)\nabla_\mu^2 L(\boldsymbol{\mu}_j)\nabla_\mu L(\boldsymbol{\mu}_j) < \epsilon \tag{57}$$

As shown, the optimal step-size of each node not only depends on the energy profile of the node itself but also on that of its neighbors. The initial value of the parameter vector i.e., $\mu_k(-1)$ can impact the final value of the converging point of Algorithms 1 if chosen arbitrarily. Specifically, our experiments suggest that these initial values should be chosen within the stability range (18). In particular, the following nominal choice is recommended:

$$\mu_k(-1) = \frac{1}{5\mathrm{Tr}(\sum_{k=1}^{N} c_{\ell,k} R_{u,\ell})} \tag{58}$$

V. Numerical Results

In this section, we present the results of our computer simulations to demonstrate the tracking performance of DLMS algorithms and to verify the theoretical findings. We consider a connected network with $N = 10$ nodes that are placed

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCNS.2016.2578044, IEEE Transactions on Control of Network Systems

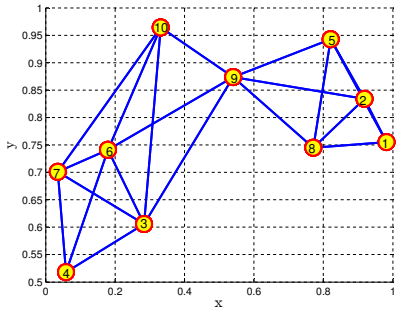

Fig. 4: Network topology.

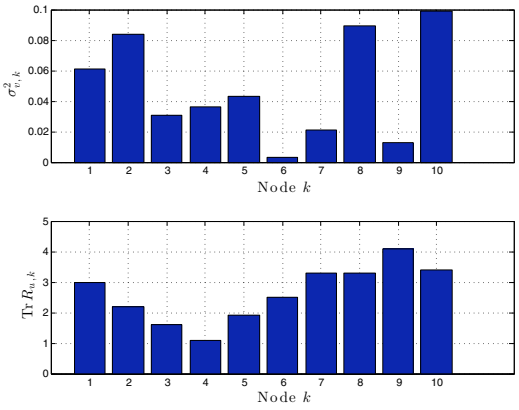

Fig. 5: Network noise variance $\sigma^2_{v,k}$ and regressor power $\mathrm{Tr}(R_{u,k})$.

randomly on a unit square area $(x, y) \in [0\ \ 1] \times [0\ \ 1]$, as shown in Fig. 4. The maximum communication range of each node is 0.4 unit length, i.e., two nodes are considered as neighbors if their distance is less than 0.4. The network energy profile is shown in Fig. 5. The objective of the network is to cooperatively estimate and track a time-varying parameter vector $\boldsymbol{w}_i^o$ of size $M = 2$ that evolves with time according to the first order random walk model given in (2). For this model, we consider both rapidly and slowly time-varying parameters. In this simulation, we choose $\mathrm{Tr}(Q) = 0.0025 \min(\sigma^2_{v,1}, \cdots, \sigma^2_{v,N})$ for slowly varying parameters. We set $A_1 = I$, compute matrix C according to metropolis rule [29] and obtain the combination matrix A_2 according to the relative degree criterion [30]. In the results, we use C_{Met} and A_{Rel} to denote Metropolis and relative degree combination matrices, respectively. We use Algorithm 1 to find the optimal step-sizes of the network. For this algorithm, we use $\epsilon = 10^{-7}$, and initialize t at 0.01. We choose the initial value $\mu_k(-1)$ according to (58) for all k.

Figs. 6 and 7 show that the algorithm finds the nodes' optimal step sizes after about 30 iterations. These figures also shows how the steady-state MSD and EMSE of the network improve at each iteration of this algorithm.

Fig. 8 shows the MSD of the network versus the mean of the step-sizes, at each iteration j, for cooperative and non-cooperative networks. These results indicate that the network

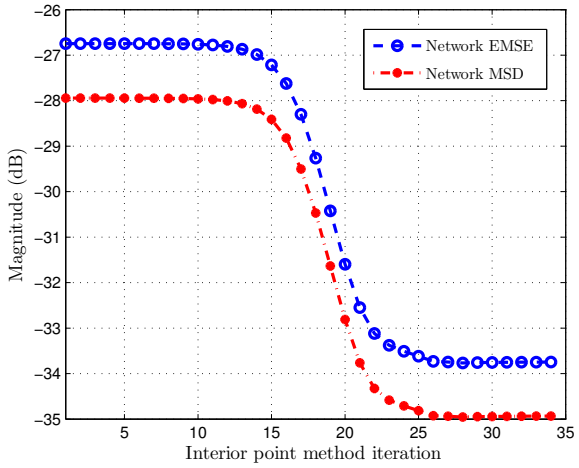

Fig. 6: Network performance improvement in terms of MSD and EMSE versus iteration number in step-step-size optimization.

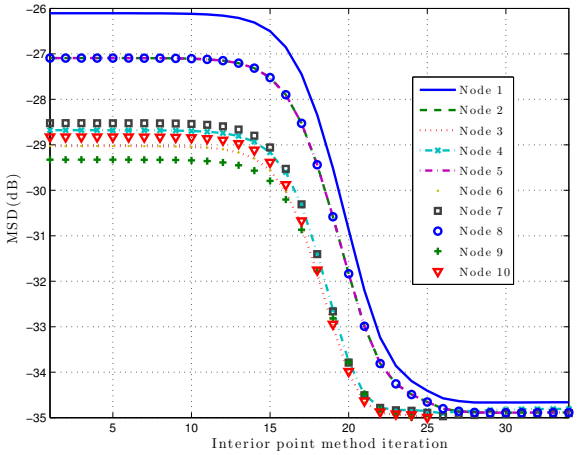

Fig. 7: Node performance improvement in terms of MSD versus iteration number in step-size optimization.

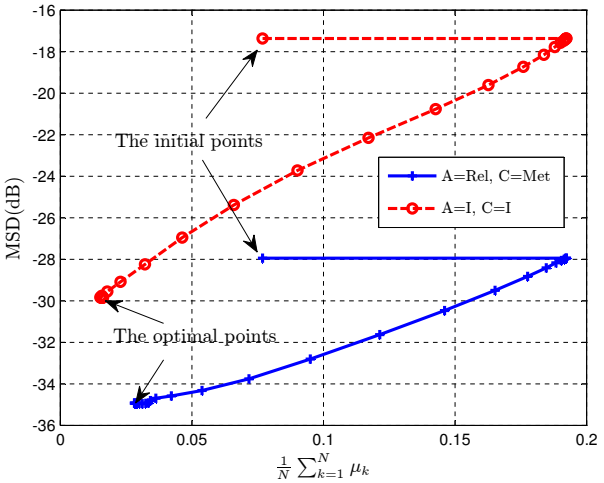

Fig. 8: The network MSD versus average step-size parameters.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCNS.2016.2578044, IEEE Transactions on Control of Network Systems

TABLE I: Performance comparison of the step-size optimization methods for slowly-varying network environments.

	Method	
Step-size	Formula (41)	Algorithm 1
μ_1^o	0.00905	0.00918
μ_2^o	0.00900	0.00910
μ_3^o	0.01739	0.01760
μ_4^o	0.01946	0.01967
μ_5^o	0.01355	0.01373
μ_6^o	0.04472	0.04780
μ_7^o	0.01471	0.01510
μ_8^o	0.00715	0.00724
μ_9^o	0.01678	0.01748
μ_{10}^o	0.00667	0.00675

with DLMS algorithms has higher tracking agility than the network with non-cooperative LMS.

In Table (I), we compare the results of Algorithm 1 and the closed form expression (41) in finding the optimal step-sizes of the network in the non-cooperative mode where the signal is slowly time-varying. We observe that the numerical difference between the results of these two methods is insignificant and the Algorithm 1 performs well.

In Fig. 9, we evaluate the performance of DLMS algorithms in tracking time-varying parameters and examine the impact of step-size parameters on their performance. We observe that the network estimates the underlying parameters of interest after few samples and keeps tracking the parameter changes over time.

Figs. 10 and 11 show the steady-state and transient behavior of DLMS algorithms in terms of MSD with optimal and non-optimal step sizes. We use the optimal step-sizes obtained from previous simulations and use equal step-sizes within the stability range of the algorithm as non-optimal step-size values. For this simulation, the non-optimal values are chosen as $\mu_k = 0.01$ for all k. The algorithm is tested for two network modes, i.e, cooperative and non-cooperative. As these results indicate, the network in the cooperative mode with A_{Rel} and C_{Met} outperforms the network in non-cooperative case. We also observe that the MSD performance of all nodes over the network is almost identical in cooperative mode while in the non-cooperative mode, the performance discrepancy between nodes is more than 5dB. This implies that nodes reach a consensus on the estimated system parameters despite having different energy and noise profile. In all scenarios, the simulations and analysis results coincide well.

From Figs. 10 and 11, we also observe that the optimal step-size parameters, $\{\mu_k^o\}$, significantly increase the convergence speed of the algorithm and enhance the accuracy of the estimation.

Fig. 12 shows the performance of DLMS [5] with fixed step-sizes and with the optimal step-sizes proposed in rapidly varying signal environment. For this simulation, we increase the value of $\mathrm{Tr}(Q)$ by 10 times to model the fast-varying signal condition. All the other parameters are chosen the same as the previous case. The nearly optimal step-sizes were obtained using the proposed interior point method. The computed values

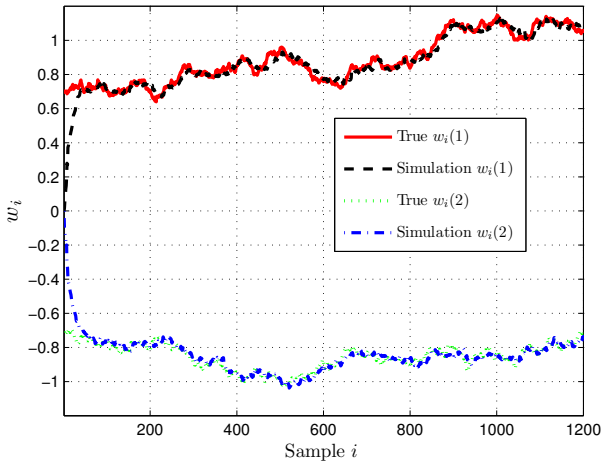

Fig. 9: Time evolution of adaptive parameter estimates over the network.

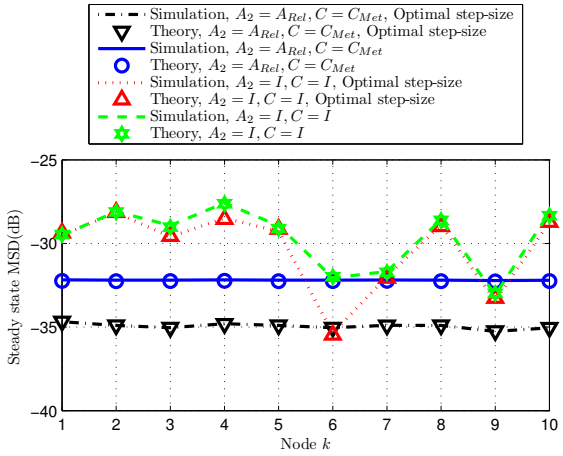

Fig. 10: The steady-state MSD performance of DLMS in different modes, i.e., ATC and non-cooperative modes.

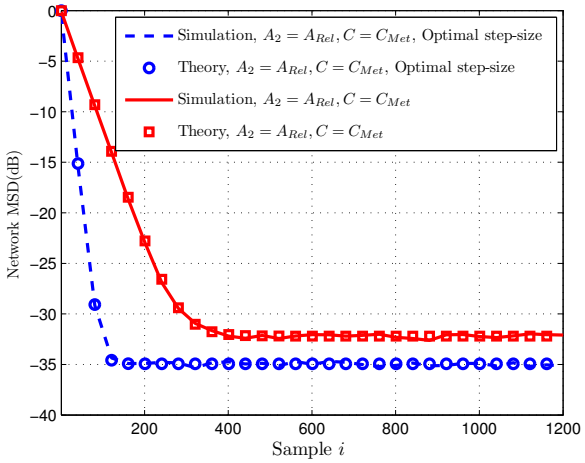

Fig. 11: Network transient MSD performance using DLMS algorithm with different choices of the combination matrices and optimal and non-optimal step-size parameters.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCNS.2016.2578044, IEEE Transactions on Control of Network Systems

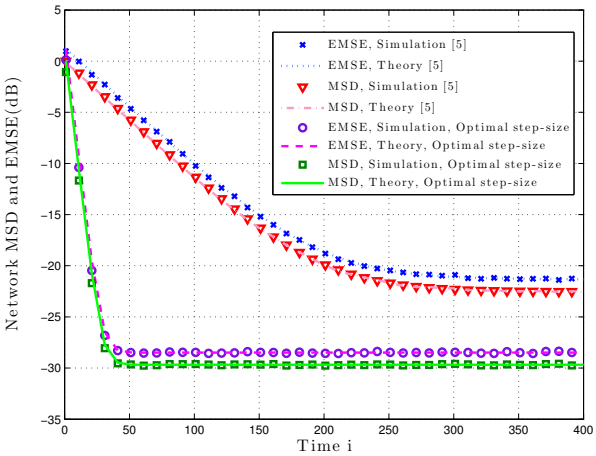

Fig. 12: The MSE performance of conventional DLMS algorithm [5] and DLMS with optimal step-sizes in fast-varying signal environement.

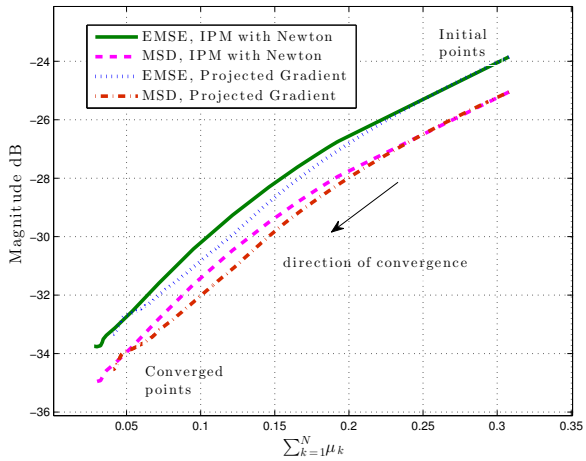

Fig. 13: Trajectory of PG and IPM with Newton versus average value of the adaptation step-sizes.

are $\mu_1 = 0.0542$, $\mu_2 = 0.0440$, $\mu_3 = 0.0583$, $\mu_4 = 0.1630$ $\mu_5 = 0.1099$, $\mu_6 = 0.2445$, $\mu_7 = 0.0566$, $\mu_8 = 0.0419$, $\mu_9 = 0.0917$, and $\mu_{10} = 0.0279$. As expected, for the fast-varying case, the computed step-sizes are larger than the ones obtained for the slow-varying case. As shown in Fig. 12, the DLMS algorithm with the optimal step-sizes not only converges faster but also achieves a lower error rate that the DLMS with non-optimal step-size values. We also note that the theory and simulation results coincide well for all cases.

To find the optimal step-size values, we have also implemneted the gradient projection method, also refered to as projected gradient (PG) algorithm. The numerical results for this algorithm and the interior point method (IPM) with Newton iterations are presented in Figs. 13 and 14. Fig. 13 shows the convergence bahavior of the PG algorithm and compare it with that of the IPM algorithm. As illustrated in this figure, both algorithms take almost similar trajectory for $\sum_{k=1}^{N} \mu_k$ to reach the desired step-size parameter vector. However, as Fig. 14 shows the IPM converges after 35 iterations whereas PG algorithm takes about 2000 iterations. It should be noted that

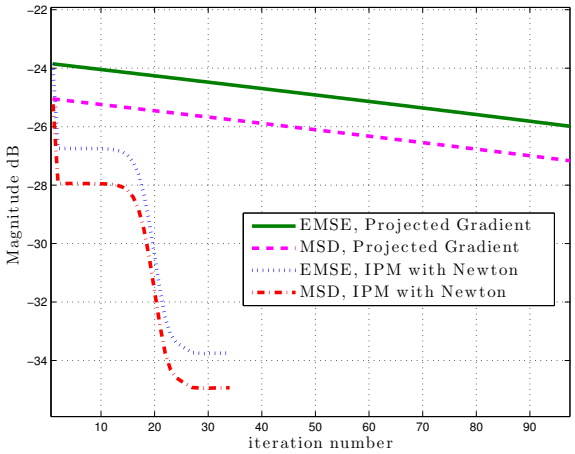

Fig. 14: The MSD and EMSE convergence bahvior versus iterations with PG and IPM with Newton.

the PG algorithm has lower computational complexity than the IPM. Our numerical simulation reveals that the performance of the PG algorithm highly depends on the step-size it uses. Therefore, a line search algorithm should be implemented along with it. In contrast, the performance of IPM with Newton algorithm is satisfactory even with no line search algorithm.

VI. Conclusions

We examined the performance of DLMS algorithms in non-stationary signal environment where the underlying model parameters of interest vary over time. We showed that there exists an optimal step-size parameter vector by which the DLMS algorithms reach their minimum MSE in the steady-state. We found this optimal step-size parameter vector by formulating a constrained nonlinear optimization problem and solving it through a log-barrier Newton algorithm. Our studies show that the optimal step-size parameter at each node in the network not only depends on the statistics of the random walk and the energy profile of the node itself but that is also depends on the energy and statistical profile of neighboring nodes. The results show that the optimal step-size parameters obtained through constrained optimization can substantially improve the performance of DLMS algorithms in tracking time-varying parameters over networks.

Appendix A
Derivation of (27)

We first note that when $\boldsymbol{u}_{k,i}$ are zero mean circular complex-valued Gaussian random vectors, then for any Hermitian matrix Γ of compatible dimensions it holds that [41]:

$$\mathbb{E}[\boldsymbol{u}_{k,i}^{*}\boldsymbol{u}_{k,i}\Gamma\boldsymbol{u}_{k,i}^{*}\boldsymbol{u}_{k,i}] = \beta(R_{u,k}\Gamma R_{u,k}) + R_{u,k}\mathrm{Tr}(\Gamma R_{u,k}) \tag{59}$$

where $\beta = 1$ for complex regressors and $\beta = 2$ when the regressors are real. Based on this result, we obtain:

$$\mathbb{E}[\boldsymbol{u}_{k,i}^{*}\boldsymbol{u}_{k,i}\Gamma\boldsymbol{u}_{\ell,i}^{*}\boldsymbol{u}_{\ell,i}] = R_{u,k}\Gamma R_{u,\ell} + \delta_{kl}(\beta-1)R_{u,k}\Gamma R_{u,k} + \delta_{kl}R_{u,k}\mathrm{Tr}(\Gamma R_{u,k}) \tag{60}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCNS.2016.2578044, IEEE Transactions on Control of Network Systems

δ_{kl} is Kronecker delta function. We now proceed to compute F using this result. The block vectorization of the first three terms in (22) is straightforward. We only present the computation of the fourth term. Tacking the block vectorization of the fourth term yields:

$$\mathrm{bvec}(\mathrm{E}[\mathcal{A}_1\boldsymbol{\mathcal{R}}_i^*Q\boldsymbol{\mathcal{R}}_i\mathcal{A}_1^T]) =(\mathcal{A}_1 \otimes_b \mathcal{A}_1)\mathrm{bvec}(\Pi), \qquad (61)$$

where here $Q = \mathcal{M}\mathcal{A}\Sigma\mathcal{A}^T\mathcal{M}$, and $\Pi = \mathrm{E}[\boldsymbol{\mathcal{R}}_i^*Q\boldsymbol{\mathcal{R}}_i]$. The $[k,l]$-th block of Π is given by:

$$\begin{aligned}\Pi_{k,l} =&\sum_m\sum_n c_{m,k}c_{n,\ell}\mathbb{E}[\boldsymbol{u}_{m,i}^*\boldsymbol{u}_{m,i}Q_{k,l}\boldsymbol{u}_{n,i}\boldsymbol{u}_{n,i}^*]\\ =&\sum_m c_{m,k}c_{m,\ell}\mathbb{E}[\boldsymbol{u}_{m,i}^*\boldsymbol{u}_{m,i}Q_{k,l}\boldsymbol{u}_{m,i}\boldsymbol{u}_{m,i}^*]\\ &+\sum_m\sum_n(1-\delta_{mn})c_{m,k}c_{n,\ell}\mathbb{E}[\boldsymbol{u}_{m,i}^*\boldsymbol{u}_{m,i}Q_{k,l}\boldsymbol{u}_{n,i}\boldsymbol{u}_{n,i}^*]\end{aligned} \qquad (62)$$

Using this expression, we obtain:

$$\begin{aligned}\Pi_{k,l} =&\sum_m c_{m,k}c_{m,\ell}\big[(\beta-1)R_{u,m}Q_{k,l}R_{u,m}\\ &+R_{u,m}\mathrm{Tr}(Q_{k,l}R_{u,m})\big]+\sum_m\sum_n c_{m,k}c_{n,\ell}[R_{u,m}Q_{k,l}R_{u,n}]\end{aligned} \qquad (63)$$

Some algebra and data rearrangement then give:

$$\begin{aligned}\mathrm{bvec}(\Pi) =&\sum_{m=1}^N\big[\mathrm{diag}\{\mathrm{vec}(C_m\mathbb{1}_{N\times N}C_m)\}\big]\\ &\otimes\big[(\beta-1)(R_{u,m}^T\otimes R_{u,m})+r_mr_m^*\big]\mathrm{bvec}(Q)\\ &+(\mathcal{R}^T\otimes_b\mathcal{R})\mathrm{bvec}(Q)\end{aligned} \qquad (64)$$

where in this relation $C_m = \mathrm{diag}(e_m^TC)$, $r_m = \mathrm{vec}(R_{u,m})$ and $\mathbb{1}_{N\times N}$ is the $N\times N$ matrix with unit entries. Finally, by computing the block vectorization of the first three terms in (22) and using (64), we arrive at (26).

REFERENCES

[1] R. Abdolee, B. Champagne, and A. H. Sayed, "Diffusion LMS strategies for parameter estimation over fading wireless channels," in *Proc. of IEEE Int. Conf. on Communications (ICC)*, Budapest, Hungary, June 2013, pp. 1926–1930.

[2] R. Abdolee, B. Champagne, and A. Sayed, "Estimation of space-time varying parameters using a diffusion LMS algorithm," *IEEE Trans. on Signal Processing*, vol. 62, no. 2, pp. 403– 418, Jan. 2014.

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

[4] S. S. Ram, V. Veeravalli *et al.*, "Distributed subgradient projection algorithm for convex optimization," in *Proc. IEEE Int. Conf. on Acoust., Speech, Signal Process.*, Taipei, Taiwan, Apr. 2009, pp. 3653–3656.

[5] K. Srivastava and A. Nedić, "Distributed asynchronous constrained stochastic optimization," *IEEE J. Selected Topics on Signal Processing*, vol. 5, no. 4, pp. 772–790, Apr. 2011.

[6] S. Lee and A. Nedic, "Distributed random projection algorithm for convex optimization," *IEEE J. Selected Topics on Signal Processing*, vol. 7, no. 2, pp. 221–229, Apr. 2013.

[7] R. Abdolee, B. Champagne, and A. H. Sayed, "A Diffusion LMS Strategy for Parameter Estimation in Noisy Regressor Applications," in *Proc. European Signal Processing Conference (EUSIPCO), Bucharest, Romania*, Aug. 2012, pp. 749–753.

[8] ——, "Diffusion LMS for Source and Process Estimation in Sensor Networks," in *Statistical Signal Processing Workshop (SSP), 2012 IEEE*, Aug. 2012, pp. 165–168.

[9] ——, "Diffusion Adaptation over Multi-Agent Networks with Wireless Link Impairments," *IEEE Trans. Mobile Computing*, vol. 15, no. 6, pp. 1362–1376, June 2016.

[10] D. Estrin, L. Girod, G. Pottie, and M. Srivastava, "Instrumenting the world with wireless sensor networks," in *Proc. IEEE Int. Conf. on Acoust., Speech, Signal Process.*, Salt Lake City, Utah, USA, May 2001, pp. 2033–2036.

[11] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," *IEEE Commun. Mag.*, vol. 38, pp. 393–422, Mar. 2002.

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

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

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

[15] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, May 2010.

[16] M. Rabbat and R. Nowak, "Distributed optimization in sensor networks," in *Proc. Int. Symp. on Inf. Proc. in Sensor Networks (IPSN)*, 2004, pp. 20–27.

[17] G. Mateos, I. D. Schizas, and G. B. Giannakis, "Performance analysis of the consensus-based distributed LMS algorithm," *EURASIP J. Adv. Signal Process.*, pp. 1–19, Nov. 2009.

[18] D. Jakovetic, J. Xavier, and J. M. Moura, "Weight optimization for consensus algorithms with correlated switching topology," *IEEE Trans. on Signal Processing*, vol. 58, no. 7, pp. 3788–3801, July 2010.

[19] K. Chan, A. Swami, Q. Zhao, and A. Scaglione, "Consensus algorithms over fading channels," in *Military Communications Conference*, San Jose, CA, Nov. 2010, pp. 549–554.

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

[21] ——, "Distributed consensus algorithms in sensor networks: Quantized data and random link failures," *IEEE Trans. on Signal Processing*, vol. 58, no. 3, pp. 1383–1400, Mar. 2010.

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

[23] S. Barbarossa and G. Scutari, "Bio-inspired sensor network design," *IEEE Signal Processing Magazine*, vol. 24, no. 3, pp. 26–35, May 2007.

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

[25] U. A. Khan and J. M. Moura, "Distributing the Kalman filter for large-scale systems," *IEEE Trans. on Signal Processing*, vol. 56, no. 10, pp. 4919–4935, Oct. 2008.

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

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

[28] A. Bertrand and M. Moonen, "Consensus-based distributed total least squares estimation in ad hoc wireless sensor networks," *IEEE Trans. on Signal Processing*, pp. 2320–2330, May 2011.

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

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

[31] L. Li and J. A. Chambers, "Distributed adaptive estimation based on the APA algorithm over diffusion networks with changing topology," *Proc. IEEE Statist. Signal Process. Workshop*, pp. 757–760, Cardiff, Wales, Sep. 2009.

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

[33] A. Khalili, M. Tinati, A. Rastegarnia, and J. Chambers, "Steady-state analysis of diffusion LMS adaptive networks with noisy links," *IEEE Trans. on Signal Processing*, vol. 60, no. 2, pp. 974–979, Feb. 2012.
[34] S. Y. Tu and A. H. Sayed, "Diffusion strategies outperform consensus strategies for distributed estimation over adaptive networks," *IEEE Trans. on Signal Processing*, vol. 60, no. 12, pp. 6217–6234, Dec. 2012.
[35] F. S. Cattivelli, C. G. Lopes, and A. H. Sayed, "Diffusion recursive least-squares for distributed estimation over adaptive networks," *IEEE Trans. on Signal Processing*, vol. 56, pp. 1865–1877, May 2008.
[36] R. Abdolee and B. Champagne, "Distributed blind adaptive algorithms based on constant modulus for wireless sensor networks," in *Proc. IEEE Int. Conf. on Wireless and Mobile Commun.*, Sept. 2010, pp. 303–308.
[37] J. Chen and A. H. Sayed, "Diffusion adaptation strategies for distributed optimization and learning over networks," *IEEE Trans. on Signal Processing*, vol. 60, no. 8, pp. 4289–4305, Aug. 2012.
[38] A. H. Sayed, "Diffusion adaptation over networks," in *E-Reference Signal Processing,* vol. 3, R. Chellapa and S. Theodoridis, *Eds.,* pp 323–454, Academic Press, 2014. Also available as arXiv:1205.4220v1[cs.MA], May 2012.
[39] X. Zhao, S. Y. Tu, and A. H. Sayed, "Diffusion adaptation over networks under imperfect information exchange and non-stationary data," *IEEE Trans. on Signal Processing*, vol. 60, no. 7, pp. 3460–3475, July 2012.
[40] A. Leon-Garcia, *Probability and Random Processes for Electrical Engineering*. 2nd edition, Addison-Wesley, 1994.
[41] A. H. Sayed, *Adaptive Filters*. Wiley-IEEE Press, 2008.
[42] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2009.
[43] J. Nocedal and S. Wright, *Numerical optimization*. Springer Science & Business Media, 2006.

Reza Abdolee received his Ph.D. from the Department of Electrical and Computer Engineering, McGill University, Montreal, Canada in 2014. He is currently with Qualcomm, Inc., San Diego,. CA. In 2012, he was a research scholar at the Bell Labs, Alcatel-Lucent, Stuttgart, Germany. In 2011, he was a visiting scholar at the Department of Electrical Engineering, University of California, Los Angeles (UCLA). From 2006 to 2008, Dr. Abdolee worked 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 digital hardware design, communication theory, statistical signal processing and optimization. Dr. Abdolee was a recipient of several awards and scholarships, including, NSERC Postgraduate Scholarship, FQRNT Doctoral Research Scholarship, McGill Graduate Research Mobility Award and DAAD-RISE International Internship scholarship from Germany.

Vida Vakilian received her Ph.D. degree from ÉcolePolytechnique de Montréal, Canada, in Electrical Engineering in 2014. She is currently an Assistant Professor at California State University, Bakersfield (CSUB). Before joining CSUB, she has been working on several research topics in collaboration with industry and R&D laboratories. From Feb. 2013 to Feb. 2014, she was a co-op Engineer at InterDigital Communications Inc., where she worked on design and development of advanced signal processing algorithms for the 3rd Generation Partnership Project (3GPP) standards-based cellular systems. From Aug. to Dec. 2012, she was a research intern at Bell Labs, Alcatel-Lucent, Germany. During this period, she contributed to 5GNOW project aiming to develop new robust PHY layer concepts, suited for future wireless systems. From Apr. to Sep. 2011, she was a visiting scholar at University of California, Davis, and worked on analyzing and investigating the performance of reconfigurable MIMO systems in spatially correlated frequency-selective fading channels. Her research interests include communication theory, statistical signal processing and hardware design and integration.

Benoit Champagne received the B.Ing. degree in Engineering Physics from the Ecole Polytechnique de Montral in 1983, the M.Sc. degree in Physics from the Universit de Montral 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, Universit du Quebec, Montral. 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 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. He has been an Associate Editor for the IEEE Signal Processing Letters, the IEEE Trans. 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. He is currently a Senior Member of IEEE.