MEAN-SQUARE ANALYSIS OF CONTINUOUS-TIME DISTRIBUTED ESTIMATION STRATEGIES

Vítor H. Nascimento[†] *and Ali H. Sayed*[‡]

[†] Dept. of Electronic Systems Eng.
Univ. of São Paulo, Brazil
vitor@lps.usp.br

[‡] Dept. of Electrical Engineering
Univ. of California, Los Angeles
sayed@ee.ucla.edu

ABSTRACT

In this work, we study the mean-square-error performance of a diffusion strategy for continuous-time estimation over networks. We derive differential equations that describe the evolution of the mean and correlation of the weight-error vector, and provide expressions for the steady-state mean-square deviation and excess mean-square error measures. Simulation results illustrate a good match between the theoretical model and the practical results.

1. INTRODUCTION

Distributed estimation and control algorithms are useful in several contexts involving decentralized inference and decision tasks; they are also useful in the modeling and analysis of biological and social networks [1–11]. Although many recent works treat discrete-time distributed techniques, continuous-time (CT) strategies are also generating interest [2, 5, 11]. For example, many systems operate in continuous-time and, therefore, models for naturally-occurring distributed networks should take continuous-time effects into account.

We recently developed continuous-time distributed learning strategies that enable a network of nodes to cooperate through in-network processing to solve a global estimation problem [12, 13] by exploiting useful discrete-time diffusion learning strategies from [6,8]. Using deterministic arguments, we examined the stability of the continuous-time diffusion strategies. In this work we examine the CT strategies from a stochastic point of view by evaluating the influence of noise on the quality of the solution computed by the network. Although CT estimation algorithms such as the stand-alone CT least-mean square (LMS) algorithm [14–16] and the CT Kalman filter [17] have been studied before in the literature, mean-square analyses of continuous-time parameter estimation algorithms that take into account the stochastic nature of both the measurement noise and the regressor vector are not as prevalent, and more so in the context of distributed solutions.

The work of V. H. Nascimento was partly supported by CNPq and FAPESP grants 303921/2010-2 and 2011/06994-2. The work of A. H. Sayed was supported in part by NSF grant CCF-1011918.

There is one important difference between the Kalman filtering set-up and the adaptive filtering set-up that does not allow us to directly apply Kalman filtering results to the current problem. Specifically, in the context of adaptation, the measurement equation takes the form:

$$d(t) = \boldsymbol{u}^T(t)\boldsymbol{w}_{\rm o} + v(t), \tag{1}$$

where $\boldsymbol{u}(t) \in \mathbb{R}^M$ is the regressor vector, $d(t)$ is the (scalar) measured output (known as the *desired* signal [19]), $\boldsymbol{w}_{\rm o} \in \mathbb{R}^M$ is the parameter vector we wish to estimate, and $v(t)$ is noise. In the Kalman filtering framework, $\boldsymbol{u}(t)$ is modeled as a known, deterministic function, whereas in the adaptive filtering framework, $\{\boldsymbol{u}(t)\}$ is modeled as a stochastic process, usually assumed to be stationary.

Much of the previous work on continuous-time adaptation considers worst-case deterministic models [20]. Fewer works develop stochastic models for the CT LMS. In [14], the proposed model takes into account only the steady-state effect of the offsets in the filter implementation. Reference [15] proves almost-sure stability of the CT LMS algorithm, but does not provide expressions for performance measures such as mean-square deviation (MSD) or excess mean-square error (EMSE). Finally, [16] considers deterministic (sinusoidal) regressors.

The mean-square error analysis in this work studies mean-square performance in the presence of both gradient noise and stochastic regressors. In addition, the analysis derives expressions that quantify the MSD and EMSE performance of the CT solution for both cases of network adaptation and, by specialization, of stand-alone adaptation. In the following section we develop our mean-square model for the stand-alone CT LMS algorithm. Later, in Section 3, we extend the analysis to distributed diffusion estimation strategies. Simulation examples verifying the accuracy of the model are presented in Section 4, and Section 5 concludes the paper.

2. MEAN-SQUARE ANALYSIS FOR THE STAND-ALONE CT LMS ALGORITHM

The continuous-time LMS algorithm forms estimates $\boldsymbol{w}(t)$ for the unknown parameter vector $\boldsymbol{w}_{\rm o}$ from (1) through the

update law

$$\begin{aligned} e(t) &= d(t) - \boldsymbol{u}^T(t)\boldsymbol{w}(t), \\ \dot{\boldsymbol{w}}(t) &= \gamma e(t)\boldsymbol{u}(t), \end{aligned} \tag{2}$$

where the initial condition to the differential equation is $\boldsymbol{w}(0)$, assumed to be deterministic, $d(t)$ and $\boldsymbol{u}(t)$ are as in (1), $e(t)$ is the estimation error, $\gamma > 0$ is a constant and $\dot{\boldsymbol{w}}(t) = \mathrm{d}\boldsymbol{w}(t)/\,\mathrm{d}\,t$.

Using (1), we can rewrite (2) in terms of the parameter error vector $\tilde{\boldsymbol{w}}(t) = \boldsymbol{w}_{\mathrm{o}} - \boldsymbol{w}(t)$ as

$$\dot{\tilde{\boldsymbol{w}}}(t) = -\gamma \boldsymbol{u}(t)\boldsymbol{u}^T(t)\tilde{\boldsymbol{w}}(t) - \gamma \boldsymbol{u}(t)v(t), \tag{3}$$

where we assume that $v(t)$ is zero-mean white noise, independent of $\boldsymbol{u}(\tau)$ for all t, τ, with covariance $\mathrm{E}\{v(t)v(t-\tau)\} = N_v\delta(\tau)$, where $\mathrm{E}\{\cdot\}$ represents expectation, and $\delta(\tau)$ is Dirac's delta function. We further assume that $\{\boldsymbol{u}(t)\}$ is a zero-mean stationary process with continuous autocorrelation function $\boldsymbol{R}_u(\tau) = \mathrm{E}\{\boldsymbol{u}(t)\boldsymbol{u}(t-\tau)\}$ and finite fourth-order moments.

Given our assumption that $v(t)$ is white noise, relation (3) should in principle be treated using stochastic calculus tools [21]. However, since we are interested only in finding the mean and covariance of $\tilde{\boldsymbol{w}}(t)$, we can follow a simpler route, described in [17, pp. 618–620]. The idea is to discretize (3) using Euler's rule, replacing the continuous-time variables by discrete-time approximations, such as

$$v_n \triangleq \frac{1}{\Delta}\int_{n\Delta}^{(n+1)\Delta} v(t)\,\mathrm{d}\,t, \tag{4}$$

and similarly for $\boldsymbol{u}_n$ and $\tilde{\boldsymbol{w}}_n$. It can be shown [17] that under these conditions, we have

$$\begin{aligned} \mathrm{E}\{v_n\} &= 0, \qquad & \mathrm{E}\{v_m v_n\} &= \frac{N_v}{\Delta}\delta_{mn}, \\ \mathrm{E}\{\boldsymbol{u}_n\} &= \boldsymbol{0}, \qquad & \mathrm{E}\{\boldsymbol{u}_m\boldsymbol{u}_n\} &\approx \boldsymbol{R}_u\big((m-n)\Delta\big), \end{aligned}$$

where now $\delta_{mn} = 1$ if $m = n$ and zero otherwise. The last approximation holds for processes with continuous autocorrelation functions.

To proceed, we use Euler's rule to discretize (3), replacing the variables by their discrete-time approximations

$$\frac{\tilde{\boldsymbol{w}}_{n+1} - \tilde{\boldsymbol{w}}_n}{\Delta} \approx -\gamma \boldsymbol{u}_n\boldsymbol{u}_n^T\tilde{\boldsymbol{w}}_n - \gamma \boldsymbol{u}_n\boldsymbol{v}_n, \tag{5}$$

which gives

$$\tilde{\boldsymbol{w}}_{n+1} \approx \tilde{\boldsymbol{w}}_n - \Delta\gamma \boldsymbol{u}_n\boldsymbol{u}_n^T\tilde{\boldsymbol{w}}_n - \Delta\gamma \boldsymbol{u}_n\boldsymbol{v}_n. \tag{6}$$

Since (6) is now a difference equation, we can more easily evaluate the evolution of its mean and correlation. Let us denote $\boldsymbol{R} \triangleq \boldsymbol{R}_u(0)$ for simplicity, and define

$$\boldsymbol{\omega}(t) = \mathrm{E}\{\tilde{\boldsymbol{w}}(t)\}, \qquad \boldsymbol{K}(t,\tau) = \mathrm{E}\{\tilde{\boldsymbol{w}}(t)\tilde{\boldsymbol{w}}^T(\tau)\}, \tag{7}$$

$$\boldsymbol{\omega}_n = \mathrm{E}\{\tilde{\boldsymbol{w}}_n\}, \qquad \boldsymbol{K}_{m,n} = \mathrm{E}\{\tilde{\boldsymbol{w}}_m\tilde{\boldsymbol{w}}_n^T\}. \tag{8}$$

In order to proceed, we must consider the correlation between $\tilde{\boldsymbol{w}}_n$ and $\boldsymbol{u}_n$. These variables will be in general correlated, but it is known that for small step-size $\gamma\Delta$, the correlation can be disregarded in the analysis of the discrete-time LMS [19] (the results shown below can be precisely justified using the ODE method [22, 23], in which the performance of a discrete-time filter is approximated by an ordinary differential equation). For small Δ and using the independence between $\boldsymbol{u}(t)$ and $v(t)$, we can then write from (6),

$$\boldsymbol{\omega}_{n+1} \approx \boldsymbol{\omega}_n - \Delta\gamma \boldsymbol{R}\boldsymbol{\omega}_n.$$

Rearranging this expression and taking $\Delta \to 0$, we obtain:

$$\dot{\boldsymbol{\omega}}(t) = -\gamma \boldsymbol{R}\boldsymbol{\omega}(t), \quad \boldsymbol{\omega}(0) = \tilde{\boldsymbol{w}}(0) = \boldsymbol{w}_{\mathrm{o}} - \boldsymbol{w}(0). \tag{9}$$

Since $\boldsymbol{R}$ is a positive semi-definite matrix, the origin $\boldsymbol{\omega} = \boldsymbol{0}_M$ is a stable equilibrium point of (9). If $\boldsymbol{R} > 0$ (positive-definite), the equilibrium point will be exponentially stable, and $\boldsymbol{\omega}(t) \to \boldsymbol{0}$ as $t \to \infty$.

For the autocorrelation we obtain

$$\begin{aligned} \boldsymbol{K}_{n+1} \approx \boldsymbol{K}_n &- \Delta\gamma \boldsymbol{K}_n\boldsymbol{R} - \Delta\gamma \boldsymbol{R}\boldsymbol{K}_n + \Delta\gamma^2 N_v\boldsymbol{R} \\ &+ \Delta^2\gamma^2\,\mathrm{E}\{\boldsymbol{u}_n\boldsymbol{u}_n^T\boldsymbol{K}_n\boldsymbol{u}_n\boldsymbol{u}_n^T\}, \end{aligned}$$

and rearranging terms,

$$\begin{aligned} \frac{\boldsymbol{K}_{n+1} - \boldsymbol{K}_n}{\Delta} \approx &-\gamma \boldsymbol{K}_n\boldsymbol{R} - \gamma \boldsymbol{R}\boldsymbol{K}_n + \gamma^2 N_v\boldsymbol{R} \\ &+ \Delta\gamma^2\,\mathrm{E}\{\boldsymbol{u}_n\boldsymbol{u}_n^T\boldsymbol{K}_n\boldsymbol{u}_n\boldsymbol{u}_n^T\}, \end{aligned}$$

Taking $\Delta \to 0$, we obtain a differential equation for $\boldsymbol{K}(t)$,

$$\dot{\boldsymbol{K}}(t) = -\gamma \boldsymbol{R}\boldsymbol{K}(t) - \gamma \boldsymbol{K}(t)\boldsymbol{R} + \gamma^2 N_v\boldsymbol{R}, \tag{10}$$

with initial condition $\boldsymbol{K}(0) = \tilde{\boldsymbol{w}}(0)\tilde{\boldsymbol{w}}^T(0)$. Note that we used the assumption of finite fourth-order moments of $\boldsymbol{u}(t)$ in the last step. Note also that the fourth-order term in $\boldsymbol{u}(t)$ disappears in the continuous-time case, which does not happen in the discrete-time case.

One approach to prove the stability of (10) is to exploit some useful Kronecker product relations [24]. Defining

$$\boldsymbol{k}(t) = \mathrm{vec}(\boldsymbol{K}(t)) \in \mathbb{R}^{MN}, \qquad \boldsymbol{r} = \mathrm{vec}(\boldsymbol{R}), \tag{11}$$

where $\mathrm{vec}(\boldsymbol{A})$ is a vector obtained by stacking the columns of matrix $\boldsymbol{A}$ in order, one on top of the other. Using standard properties of the Kronecker product [24], we can write

$$\dot{\boldsymbol{k}}(t) = -\gamma\left[\boldsymbol{I}_M \otimes \boldsymbol{R} + \boldsymbol{R} \otimes \boldsymbol{I}_M\right]\boldsymbol{k}(t) + \gamma^2 N_v\boldsymbol{r}, \tag{12}$$

where $\otimes$ denotes the Kronecker product. If $\boldsymbol{R} > 0$, the only equilibrium point of (12) is

$$\bar{\boldsymbol{k}} \triangleq \gamma N_v\left[\boldsymbol{I}_M \otimes \boldsymbol{R} + \boldsymbol{R} \otimes \boldsymbol{I}_M\right]^{-1}\boldsymbol{r}. \tag{13}$$

This equilibrium point is exponentially stable, since the eigenvalues of $\boldsymbol{B} = \boldsymbol{I}_M \otimes \boldsymbol{R} + \boldsymbol{R} \otimes \boldsymbol{I}_M$ are $\lambda_i + \lambda_j$, where λ_i, λ_j are eigenvalues of $\boldsymbol{R}$. Since $\boldsymbol{R} > 0$, the eigenvalues of $-\boldsymbol{B}$ lie all in the left-half complex plane.

The excess mean-square error (EMSE) is defined as $\zeta(t) = \mathrm{E}\{\big(\boldsymbol{u}^T(t)\tilde{\boldsymbol{w}}(t)\big)^2\}$, while the mean-square deviation is defined as $\xi(t) = \mathrm{E}\{\tilde{\boldsymbol{w}}^T(t)\tilde{\boldsymbol{w}}(t)\}$. Both can be obtained easily from $\boldsymbol{K}(t)$:

$$\xi(t) = \mathrm{Tr}(\boldsymbol{K}(t)), \qquad \zeta(t) = \mathrm{Tr}(\boldsymbol{R}\boldsymbol{K}(t))\}$$

where $\mathrm{Tr}(\boldsymbol{A})$ represents the trace of $\boldsymbol{A}$. Their steady-state values can be obtained from (10) by setting $\dot{\boldsymbol{K}}(t) = 0$. Denote $\lim_{t\to\infty} \zeta(t) = \bar{\zeta}$, and similarly for $\bar{\xi}$. In steady-state, taking the trace of (10), we obtain

$$\bar{\zeta} = \frac{\gamma N_v \,\mathrm{Tr}(\boldsymbol{R})}{2}, \tag{14}$$

and taking the trace after multiplying (10) from the left by $\boldsymbol{R}^{-1}$ and recalling that $\mathrm{Tr}(\boldsymbol{AB}) = \mathrm{Tr}(\boldsymbol{BA})$ if both products are well-defined, we obtain in steady-state

$$\bar{\xi} = \frac{\gamma N_v M}{2}. \tag{15}$$

In the next section, these results are extended to the CT diffusion strategies of [13] for adaptation over networks.

3. MEAN-SQUARE ANALYSIS FOR DISTRIBUTED CT ADAPTIVE DIFFUSION STRATEGIES

Consider now a network of N nodes that cooperate to estimate a common vector $\boldsymbol{w}_{\mathrm{o}}$. Each node k has access to measurements $\boldsymbol{u}_k(t) \in \mathbb{R}^M$ and $d_k(t) \in \mathbb{R}$, that are assumed related to each other through

$$d_k(t) = \boldsymbol{u}_k^T(t)\boldsymbol{w}_{\mathrm{o}} + v_k(t), \tag{16}$$

where $v_k(t)$ is zero-mean white noise with autocovariance $\mathrm{E}\{v_k(t)v_k(t-\tau) = N_{v,k}\delta(\tau)$, independent of $\boldsymbol{u}_k(t)$. We assume that $\boldsymbol{u}_k(t)$ is a zero-mean stationary vector process with continuous autocorrelation matrix $\boldsymbol{R}_k(\tau)$. For simplicity, we also assume that the noise and regressor variables for different nodes are independent of each other.

In the distributed estimation scheme proposed in [13], each node k computes a local estimate $\boldsymbol{w}_k(t)$, and transmits it to its local neighborhood $\mathcal{N}_k(t)$. Each node then combines the estimates from its neighborhood, forming a combined estimate $\psi_k(t)$, which is used in the update law as follows.

$$\boldsymbol{\psi}_k(t) = \sum_{\ell=1}^{N} a_{\ell k}(t)\boldsymbol{w}_\ell(t), \tag{17a}$$

$$e_k(t) = d_k(t) - \boldsymbol{u}_k^T(t)\boldsymbol{w}_k(t), \tag{17b}$$

$$\dot{\boldsymbol{w}}_k(t) = -\gamma_0\left(\boldsymbol{w}_k(t) - \boldsymbol{\psi}_k(t)\right) + \gamma_k e_k(t)\boldsymbol{u}_k(t), \tag{17c}$$

where the $a_{\ell k}(t)$ are nonnegative weights satisfying

$$\sum_{\ell=0}^{N} a_{\ell k}(t) = 1, \tag{18}$$

$\mathcal{N}_k(t)$ contains the nodes ℓ for which $a_{\ell k}(t) \neq 0$, and γ_0 and γ_k are positive constants. Introduce the mixing matrix $\boldsymbol{A}(t) = [a_{\ell k}(t)] \in \mathbb{R}^{N\times N}$. We assume that $\boldsymbol{A}(t)$ is a stationary process, independent of $\boldsymbol{u}_k(t)$ and of $v_k(t)$ for all k and with mean $\bar{\boldsymbol{A}}$. Note that (18) implies that $\boldsymbol{A}^T(t)\mathbb{1} = \bar{\boldsymbol{A}}^T\mathbb{1} = \mathbb{1}$, where $\mathbb{1} \in \mathbb{R}^N$ has all entries equal to one.

The stability of (17c) was analyzed using deterministic arguments in [13]. We now examine its mean-square-error performance. Define

$$\tilde{\boldsymbol{w}}(t) \triangleq \mathrm{col}\{\boldsymbol{w}_{\mathrm{o}} - \boldsymbol{w}_1(t), \ldots, \boldsymbol{w}_{\mathrm{o}} - \boldsymbol{w}_N(t)\}.$$

The overall error equation is then given by

$$\dot{\tilde{\boldsymbol{w}}}(t) = -\boldsymbol{B}(t)\tilde{\boldsymbol{w}}(t) - \boldsymbol{U}(t)\boldsymbol{\Gamma}\boldsymbol{v}(t), \tag{19}$$

with

$$\boldsymbol{B}(t) \triangleq \gamma_0\left(\boldsymbol{I}_{MN} - \boldsymbol{A}^T(t)\otimes\boldsymbol{I}_M\right) + \boldsymbol{U}(t)\boldsymbol{\Gamma}\boldsymbol{U}^T(t), \tag{20}$$

where $\boldsymbol{v}(t) = \begin{bmatrix} v_1(t) & v_2(t) & \ldots & v_N(t)\end{bmatrix}^T$, $\boldsymbol{\Gamma} = \mathrm{diag}\{\gamma_k\}$ and the regressor matrix $\boldsymbol{U}(t) \in \mathbb{R}^{MN\times N}$ is constructed as

$$\boldsymbol{U}(t) = \mathrm{diag}\{\boldsymbol{u}_1(t), \boldsymbol{u}_2(t), \ldots, \boldsymbol{u}_N(t)\}. \tag{21}$$

Introduce the average error vector $\omega(t) = \mathrm{E}\{\tilde{\boldsymbol{w}}(t)\}$ and the correlation matrix $\boldsymbol{K}(t) = \mathrm{E}\{\tilde{\boldsymbol{w}}(t)\tilde{\boldsymbol{w}}^T(t)\}$, as well as $\bar{\boldsymbol{B}} = \mathrm{E}\{\boldsymbol{B}(t)\}$. Note that, since $\boldsymbol{A}$ is not necessarily symmetric, $\boldsymbol{B}(t)$ and $\bar{\boldsymbol{B}}$ in general will not be symmetric matrices.

Using similar arguments as in the previous section, we can argue that

$$\dot{\omega}(t) = -\bar{\boldsymbol{B}}\omega(t), \tag{22}$$

$$\dot{\boldsymbol{K}}(t) = -\bar{\boldsymbol{B}}\boldsymbol{K}(t) - \boldsymbol{K}(t)\bar{\boldsymbol{B}}^T + \boldsymbol{C}, \tag{23}$$

with initial conditions $\omega(0) = \tilde{\boldsymbol{w}}(0)$, $\boldsymbol{K}(0) = \tilde{\boldsymbol{w}}(0)\tilde{\boldsymbol{w}}^T(0)$, and

$$\bar{\boldsymbol{B}} = \gamma_0\left(\boldsymbol{I}_{MN} - \bar{\boldsymbol{A}}\otimes\boldsymbol{I}_M\right) + \mathrm{diag}\{\gamma_k\boldsymbol{R}_k\}, \tag{24}$$

$$\boldsymbol{C} = \mathrm{diag}\{\gamma_k^2 N_{v,k}\boldsymbol{R}_k\}. \tag{25}$$

In order to show that $\omega(t)$ exponentially converges to $\boldsymbol{0}_{MN}$, we consider the Lyapunov function $V(t) = \omega^T(t)\omega(t)$. Its derivative is

$$\dot{V}(t) = -\omega^T(t)\left(\bar{\boldsymbol{B}} + \bar{\boldsymbol{B}}^T\right)\boldsymbol{\omega}(t). \tag{26}$$

If $\boldsymbol{A}(t)$ is doubly-stochastic, that is, if $\boldsymbol{A}(t)\mathbb{1} = \boldsymbol{A}^T(t)\mathbb{1} = \mathbb{1}$, then $2\boldsymbol{I}_{MN} - \boldsymbol{A}(t) - \boldsymbol{A}^T(t) \geq 0$ [13]. Therefore, if $\boldsymbol{A}(t)$ is doubly-stochastic for all t, then $\bar{\boldsymbol{A}}$ is also doubly-stochastic,

and $2\boldsymbol{I}_{MN} - \bar{\boldsymbol{A}} - \bar{\boldsymbol{A}}^T$ is positive semi-definite. This in turn implies that $\bar{\boldsymbol{B}} + \bar{\boldsymbol{B}}^T > 0$. This result and (26) imply that $\omega(t) \to \boldsymbol{0}_{MN}$ exponentially fast.

A similar approach can be used for the correlation. Define as before $\boldsymbol{k}(t) = \mathrm{vec}(\boldsymbol{K}(t))$, $\boldsymbol{c} = \mathrm{vec}(\boldsymbol{C})$. We have then

$$\dot{\boldsymbol{k}}(t) = -\left[\bar{\boldsymbol{B}} \otimes \boldsymbol{I}_{MN} + \boldsymbol{I}_{MN} \otimes \bar{\boldsymbol{B}}\right] \boldsymbol{k}(t) + \boldsymbol{c}. \tag{27}$$

Let $V_1(t) = \tilde{\boldsymbol{k}}^T(t)\tilde{\boldsymbol{k}}(t)$, where $\tilde{\boldsymbol{k}}(t) = \boldsymbol{k}_{\mathrm{eq}} - \boldsymbol{k}(t)$ and $\boldsymbol{k}_{\mathrm{eq}}$ is the equilibrium point of (27). Its derivative is

$$\begin{aligned}\dot{V}_1(t) = -\tilde{\boldsymbol{k}}^T(t) \Big[&(\bar{\boldsymbol{B}} + \bar{\boldsymbol{B}}^T) \otimes \boldsymbol{I}_{MN} \\ &+ \boldsymbol{I}_{MN} \otimes (\bar{\boldsymbol{B}} + \bar{\boldsymbol{B}}^T)\Big] \tilde{\boldsymbol{k}}(t).\end{aligned} \tag{28}$$

Since $\bar{\boldsymbol{B}} + \bar{\boldsymbol{B}}^T > 0$ implies $(\bar{\boldsymbol{B}} + \bar{\boldsymbol{B}}^T) \otimes \boldsymbol{I}_{MN} + \boldsymbol{I}_{MN} \otimes (\bar{\boldsymbol{B}} + \bar{\boldsymbol{B}}^T) > 0$, we conclude that $\boldsymbol{k}(t)$ converges exponentially to the steady-state value (equilibrium point)

$$\boldsymbol{k}_{\mathrm{eq}} = \left[(\bar{\boldsymbol{B}} + \bar{\boldsymbol{B}}^T) \otimes \boldsymbol{I}_{MN} + \boldsymbol{I}_{MN} \otimes (\bar{\boldsymbol{B}} + \bar{\boldsymbol{B}}^T)\right]^{-1} \boldsymbol{c}. \tag{29}$$

The EMSE and MSD for each node and for the overall network can be obtained directly from (27) and (29) as follows. We re-arrange the solution $\boldsymbol{k}(t)$ from (27) to form $\boldsymbol{K}(t)$, and then, to obtain the overall MSD and EMSE, compute

$$\xi(t) = \mathrm{Tr}(\boldsymbol{K}(t)), \quad \zeta(t) = \mathrm{Tr}\left(\mathrm{diag}\{\boldsymbol{R}_k\}\boldsymbol{K}(t)\right). \tag{30}$$

4. SIMULATIONS

In this section we verify the theoretical results through simulated examples. All simulations were made in Simulink, using the "band-limited white noise" block to approximate white noise. This block approximates white noise by generating discrete-time random sequences with a high sampling rate, compared to the time constants in the system being simulated. In our simulations, this sampling rate was $f_s = 10^3$Hz. Since the generated noise is actually band-limited, its variance is finite, and set by Simulink to $f_s\sigma^2$, where σ^2 is the desired power spectrum density (the "noise power" parameter in Simulink) of the ideal white noise.

When using these blocks to generate a regressor vector, one caution is necessary, because the simulation explicitly computes the squares of the regressors' entries. Given the sampling rate correction explained above, if the power set in Simulink for an entry $u_{k,i}(t)$ of $\boldsymbol{u}_k(t)$ is σ_u^2, the average $\mathrm{E}\{u_{k,i}^2(t)\}$ considered in the theoretical models should be $f_s\sigma_u^2$. The `randseed` function should be used as seed for the white noise blocks, in order to evaluate ensemble-average learning curves.

Our first example is the ten-node network shown in Figure 1 (weights a_{kk} are not drawn; their values are such that (18) is satisfied). In the simulations shown below, the unknown

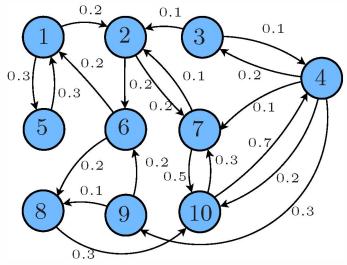

Fig. 1. 10-node network. Self-connections are not drawn.

weight vector is $\boldsymbol{w}_{\mathrm{o}} = \begin{bmatrix}0.5 & -0.1\end{bmatrix}^T$. The regressors are either band-limited noise (nodes 3, 5–9, 10) or sinusoids with random initial phase (node 1, frequency $f_1 = 10$Hz, nodes 2 and 4, $f_2 = 5$Hz, node 10, $f_10 = 6$Hz), as shown below. We only list the covariance matrices in (31). The values for the noise powers for nodes $k = 1, 5\text{–}10$ are $N_{v,k} = 10^{-4}$, while $N_{v,2} = N_{v,4} = 10^{-3}$, $N_{v,3} = 2 \times 10^{-3}$. We set $\gamma_i = 1$, $i = 0 \ldots N$.

$$\boldsymbol{R}_1 = \begin{bmatrix}2 & 0\\ 0 & 0\end{bmatrix}, \qquad \boldsymbol{R}_2 = \begin{bmatrix}0.18 & 0.45\\ 0.45 & 1.125\end{bmatrix}, \tag{31a}$$

$$\boldsymbol{R}_4 = \begin{bmatrix}0.5 & 0.25\\ 0.25 & 0.125\end{bmatrix}, \qquad \boldsymbol{R}_{10} = \begin{bmatrix}0 & 0\\ 0 & 1.125\end{bmatrix}, \tag{31b}$$

$$\boldsymbol{R}_k = r_k \begin{bmatrix}1 & 1\\ 1 & 1\end{bmatrix}, \qquad \text{for } k = 3, 5\text{–}9, \tag{31c}$$

where $r_3 = r_7 = 3$, $r_5 = 0.6$, $r_6 = 0.4$, $r_8 = 4$, $r_9 = 0.5$. Note that all the $\boldsymbol{R}_k$ are singular, which means that stand-alone LMS filters would not converge to a neighborhood of $\boldsymbol{w}_{\mathrm{o}}$. However, as Figure 2 shows, the MSD does converge to a small value for each node (we show only results for the first three nodes, the behavior for the other nodes is similar). Note also how the theoretical model follows closely the simulated results, even following the slight increase in MSD in the initial transient for node 1. Figure 3 shows a simulation

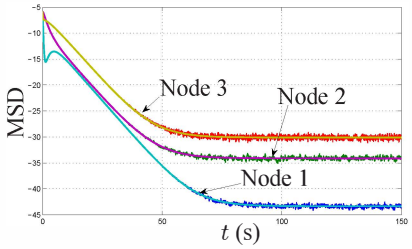

Fig. 2. Mean-square deviation (MSD) for the first three nodes in the 10-node network. The smooth curves correspond to the theoretical model. Experimental MSD obtained from the average of $L = 500$ realizations.

in which $a_{46}, a_{48}, a_{49}, a_{96}, a_{98}$ vary. The parameters follow a sinusoid with frequency 1Hz and random phase, saturated between 0 and the maximum values of $0.2, 0.1, 0.3, 0.2, 0.1$, respectively, so that $\boldsymbol{A}^T(t)\mathbb{1} = \boldsymbol{A}(t)\mathbb{1} = \mathbb{1}$ at all times. The average values are half the maxima.

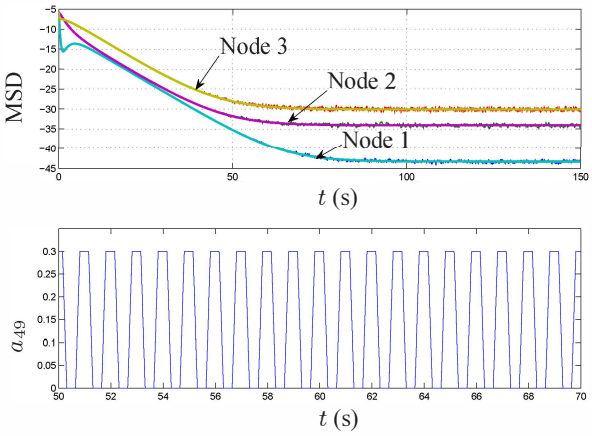

Fig. 3. (a) Mean-square deviation (MSD) for nodes 1–3 for time-variant $\boldsymbol{A}(t)$ — the smooth curves correspond to the theoretical model. (b) $a_{49}(t)$ in the interval $50 \leq t \leq 70$.

5. CONCLUSION

We performed a mean-square-error analysis of a continuous-time learning strategy and derived expressions for the MSD and EMSE performance of the network. We verified the results through simulations showing good agreement between theory and practice.

6. REFERENCES

[1] B. Sinopoli, C. Sharp, L. Schenato, S. Schaffert, and S. S. Sastry, "Distributed control applications within sensor networks," *Proc. IEEE*, vol. 91, no. 8, pp. 1235–1246, 2003.

[2] W. Ren and R. Beard, "Consensus seeking in multiagent systems under dynamically changing interaction topologies," *IEEE Trans. Autom. Control*, vol. 50, no. 5, pp. 655 – 661, May 2005.

[3] 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.

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

[5] R. Olfati-Saber, "Distributed Kalman filtering for sensor networks," in *Proc. IEEE Conference on Decision and Control*, Dec. 2007, pp. 5492 –5498.

[6] 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, 2008.

[7] 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.

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

[9] S.-Y. Tu and A. H. Sayed, "Mobile adaptive networks," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 4, pp. 649–664, 2011.

[10] 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, 2011.

[11] F. Dorfler, F. Pasqualetti, and F. Bullo, "Distributed detection of cyber-physical attacks in power networks: A waveform relaxation approach," in *Proc. 49th Allerton Conf. Communication, Control, and Computing*, 2011, pp. 1486–1491.

[12] V. H. Nascimento and A. H. Sayed, "Continuous-time distributed estimation," in *Proc. 45th Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, CA, Nov. 2011, pp. 1–5.

[13] ——, "Continuous-time distributed estimation with asymmetric mixing," in *Proc. IEEE Statistical Signal Processing Workshop (SSP)*, Ann Arbor, MI, 2012, pp. 528–531.

[14] D. A. Johns, W. M. Snelgrove, and A. S. Sedra, "Continuous-time LMS adaptive recursive filters," *IEEE Trans. Circuits Syst.*, vol. 38, no. 7, pp. 769–778, 1991.

[15] P. J. Voltz and F. Kozin, "Almost-sure convergence of the continuous-time LMS algorithm," *IEEE Trans. Signal Process.*, vol. 40, no. 2, pp. 395–401, 1992.

[16] T. K. Rawat and H. Parthasarathy, "Convergence analysis of a continuous-time nonlinear LMS-type algorithm using stochastic calculus," *AEU - International Journal of Electronics and Communications*, vol. 63, no. 7, pp. 576–583, Jul. 2009.

[17] T. Kailath, A. H. Sayed, and B. Hassibi, *Linear Estimation*. NJ: Prentice Hall, 2000.

[18] A. H. Sayed and T. Kailath, "A state-space approach to adaptive RLS filtering," *IEEE Signal Process. Mag.*, vol. 11, no. 3, pp. 18–60, Jul. 1994.

[19] A. H. Sayed, *Adaptive Filters*. Wiley, NJ, 2008.

[20] P. Ioannou and B. Fidan, *Adaptive Control Tutorial*. SIAM, PA, 2006.

[21] J. L. Speyer and W. H. Chung, *Stochastic Processes, Estimation and Control*. SIAM, PA, 2008.

[22] H. Kushner, "Stochastic approximation: a survey," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, no. 1, pp. 87–96, 2010.

[23] A. Benveniste and G. Ruget, "A measure of the tracking capability of recursive stochastic algorithms with constant gains," *IEEE Transactions on Automatic Control*, vol. 27, no. 3, pp. 639 – 649, Jun. 1982.

[24] A. J. Laub, *Matrix Analysis for Scientists and Engineers*. SIAM, PA, 2005.