CONTINUOUS-TIME DISTRIBUTED ESTIMATION

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

Adaptive diffusion models endow networks with distributed learning and cognitive abilities. These models have been applied recently to emulate various forms of complex and self-organized patterns of behavior encountered in biological networks. In diffusion adaptation, nodes share information with their neighbors in real-time, and the network evolves towards a common objective through decentralized coordination and in-network processing. Current models are based on discrete-time adaptive diffusion strategies. However, physical phenomena usually are governed by continuous-time dynamics. In this paper, we derive continuous-time diffusion adaptive algorithms, which can help provide more accurate models for exchanges of information, and also for systems with large variations in their time constants.

1. INTRODUCTION

Distributed learning and cognition algorithms have been applied recently to model various forms of complex and self-organized behavior encountered in biological networks, such as bird flight formations [1] and fish schooling [2]. Adaptive diffusion models assume that each node in the network is able to exchange information with its neighbors, and that through local cooperation and processing, the network is able to evolve towards a common objective.

Current models are based on discrete-time adaptive diffusion strategies [3, 4]. Nevertheless, most physical phenomena exhibit continuous-time (CT) dynamics. In this paper, we motivate and derive continuous-time adaptive diffusion strategies; such schemes would help provide more accurate models for complex systems with large variations in their time constants. In addition, CT models provide a useful framework for studying networks in which the exchange of information between nodes is non-synchronous (i.e., for networks whose nodes may exchange information at any instant). Continuous-time adaptive schemes have been studied before in the context of control systems [5] and numerical methods [6]. These schemes, however, were limited to single and stand-alone filters. In this paper, we develop continuous-time schemes for adaptation over networks. We derive a stochastic gradient diffusion method and prove its convergence, under certain conditions on the system inputs.

This paper is organized as follows. In the next section, we briefly review the discrete-time diffusion LMS strategy. In Sec. 3, we describe the stand-alone continuous-time LMS algorithm and its relation to discrete-time LMS. Section 4 uses the results of the previous two sections to motivate and derive a continuous-time version of the diffusion LMS algorithm. The stability properties of the CT-diffusion LMS algorithm are studied in Sec. 5. Section 6 provides simulations to illustrate the performance of the proposed algorithm, and Sec. 7 concludes the work. In the following, we use brackets (as in $\boldsymbol{w}[n]$) to denote discrete-time variables, and parentheses (as in $\boldsymbol{w}(t)$) to denote continuous-time variables.

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 grants CCF-1011918 and CCF-0942936.

2. DISCRETE-TIME DIFFUSION ALGORITHMS

Consider a connected network with N nodes, each of which measures at every time instant a scalar $d_k[n]$ and a (column) regressor $\boldsymbol{u}_k[n] \in \Re^M$. We assume the data are related through the linear model:

$$d_k[n] = \boldsymbol{u}_k^T[n]\boldsymbol{w}_{\rm o} + v_k[n], \tag{1}$$

where $\boldsymbol{w}_{\rm o}$ is a *common* unknown parameter vector that the network wishes to estimate, and $v_k[n]$ is zero-mean temporally and spatially-white noise. The notation $(\cdot)^T$ represents vector or matrix transposition. Diffusion algorithms [3, 4] are distributed methods for the estimation of $\boldsymbol{w}_{\rm o}$. There are several variants of diffusion LMS. Here, we restrict our discussions to the so-called *combine-then-adapt* (CTA) diffusion LMS strategy without measurement exchange [4]. In this implementation, each node k first receives the current estimates $\boldsymbol{w}_\ell[n]$ from its neighbors, and computes a combined intermediate estimate $\boldsymbol{\psi}_k[n]$ through the convex combination:

$$\boldsymbol{\psi}_k[n] = \sum_{\ell=1}^{N} a_{\ell k}\boldsymbol{w}_\ell[n], \tag{2}$$

where the $a_{\ell k}$ are nonnegative weights associated with each link; these weights add up to one:

$$\sum_{\ell=1}^{N} a_{\ell k} = 1 \qquad (3)$$

for all k. Moreover, only the weights that correspond to nearby nodes that are connected to node k are nonzero. For each node k, we define its neighborhood $\mathcal{N}_k$ as the set of nodes for which $a_{\ell k}$ is nonzero (this set can include node k itself).

After the combination step (2), each node then uses its current measurements to improve the intermediate estimate $\boldsymbol{\psi}_k[n]$ through an adaptation step, and subsequently shares the improved estimate with its neighbors. The adaptation step is an LMS-like update:

$$\begin{aligned} e_k[n] &= d_k[n] - \boldsymbol{u}_k^T[n]\boldsymbol{\psi}_k[n], \\ \boldsymbol{w}_k[n+1] &= \boldsymbol{\psi}_k[n] + \mu_k \boldsymbol{u}_k[n] e_k[n], \end{aligned} \qquad (4)$$

where μ_k is a step-size parameter.

In order to derive a continuous-time version of (2)-(4), we first review the relation between the classical CT and DT stand-alone LMS algorithms.

3. CONTINUOUS-TIME LMS ADAPTATION

The continuous-time LMS filter follows the differential equation

$$\dot{\boldsymbol{w}}(t) = \gamma e(t)\boldsymbol{u}(t), \qquad (5)$$

where $\boldsymbol{u}(t)$ is the continuous-time regressor, $e(t) = d(t) - \boldsymbol{u}^T(t)\boldsymbol{w}(t)$ is the error, $d(t)$ is the desired signal, and γ is a scalar positive constant (a positive-definite matrix could also be used) [5]. As in the discrete-time case, we consider a model of the form

$$d(t) = \boldsymbol{u}^T(t)\boldsymbol{w}_o + v(t)$$

for the desired signal. An important feature of this filter is that in the absence of noise, its single equilibrium point is globally exponentially stable for all $\gamma > 0$, if the input satisfies a *persistence of excitation* (PE) condition, i.e., if for all t it holds that

$$\alpha_1 \boldsymbol{I} \le \int_{t_0}^{t_0+T} \boldsymbol{u}(t)\boldsymbol{u}^T(t)dt \le \alpha_2 \boldsymbol{I}, \qquad (6)$$

for some constants $0 < T, \alpha_1, \alpha_2 < \infty$ [5].

The discrete-time LMS algorithm can be obtained from (5) via discretization by using the Euler approximation:

$$\dot{\boldsymbol{w}}(t) \approx \frac{\boldsymbol{w}((n+1)\Delta T) - \boldsymbol{w}(n\Delta T)}{\Delta T},$$

and substituting into (5), we obtain

$$\boldsymbol{w}((n+1)\Delta T) \approx \boldsymbol{w}(n\Delta T) + \gamma \Delta T e(n\Delta T)\boldsymbol{u}(n\Delta T).$$

Denoting the discrete-time sequence that results from the above recursion by $\boldsymbol{w}[n]$ (note that $\boldsymbol{w}[n]$ approximates, but is not equal to $\boldsymbol{w}(n\Delta T)$) and letting $\mu = \gamma\Delta T$, $\boldsymbol{u}[n] = \boldsymbol{u}(n\Delta T)$, $d[n] = d(n\Delta T)$, we obtain the conventional LMS recursion:

$$\begin{aligned} e[n] &= d[n] - \boldsymbol{u}^T[n]\boldsymbol{w}[n], \\ \boldsymbol{w}[n+1] &= \boldsymbol{w}[n] + \mu e[n]\boldsymbol{u}[n]. \end{aligned} \qquad (7)$$

Next, we use this relation in the reverse direction to derive a continuous-time diffusion LMS strategy from (2) and (4).

4. CONTINUOUS-TIME DIFFUSION ADAPTATION

It is convenient to rewrite (2)–(4) as a single recursion considering all nodes. Define $\boldsymbol{w}[n]$ by stacking the $\boldsymbol{w}_k[n]$ one on top of the other, i.e., $\boldsymbol{w}[n] = \text{col}\{\boldsymbol{w}_1[n], \dots \boldsymbol{w}_N[n]\}$, $\boldsymbol{a}_k = \begin{bmatrix} a_{1k} & \dots & a_{Nk} \end{bmatrix}^T$, and $\boldsymbol{A} = \begin{bmatrix} \boldsymbol{a}_1 & \dots & \boldsymbol{a}_N \end{bmatrix}$. Then we can write (2) as

$$\boldsymbol{\psi}_k[n] = \sum_{\ell=1}^{N} a_{\ell k}\boldsymbol{w}_\ell[n] = \left(\boldsymbol{a}_k^T \otimes \boldsymbol{I}_M\right)\boldsymbol{w}[n], \qquad (8)$$

where $\boldsymbol{A} \otimes \boldsymbol{B}$ is the Kronecker product of $\boldsymbol{A}$ and $\boldsymbol{B}$ [7] and $\boldsymbol{I}_M$ is the $M \times M$ identity matrix. Substituting (8) into (4), we obtain

$$\boldsymbol{w}_k[n+1] = \left(\boldsymbol{a}_k^T \otimes \boldsymbol{I}_M\right)\boldsymbol{w}[n] + \mu_k \boldsymbol{u}_k[n] e_k[n].$$

Defining now $\boldsymbol{d}[n] = \left[d_1[n] \dots d_N[n]\right]^T$ (similarly for $\boldsymbol{e}[n]$) and

$$\boldsymbol{U}[n] = \begin{bmatrix} \boldsymbol{u}_1[n] & \boldsymbol{0}_M & \dots & \boldsymbol{0}_M \\ \boldsymbol{0}_M & \boldsymbol{u}_2[n] & \dots & \boldsymbol{0}_M \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0}_M & \boldsymbol{0}_M & \dots & \boldsymbol{u}_N[n] \end{bmatrix},$$

where $\boldsymbol{0}_M$ is an $M \times 1$ vector of zeros, and $\boldsymbol{\mathcal{M}} = \text{diag}\{\mu_k\}$ we can write

$$\begin{aligned} \boldsymbol{e}[n] &= \boldsymbol{d}[n] - \boldsymbol{U}^T[n]\left(\boldsymbol{A}^T \otimes \boldsymbol{I}_M\right)\boldsymbol{w}[n], \\ \boldsymbol{w}[n+1] &= (\boldsymbol{A}^T \otimes \boldsymbol{I}_M)\boldsymbol{w}[n] + \boldsymbol{U}[n]\boldsymbol{\mathcal{M}}\boldsymbol{e}[n]. \end{aligned} \qquad (9)$$

Adding and subtracting $\boldsymbol{w}[n]$ from the right-hand side, we obtain a recursion in the form of (7):

$$\boldsymbol{w}[n+1] = \boldsymbol{w}[n] - \left(\boldsymbol{I}_{MN} - \boldsymbol{A}^T \otimes \boldsymbol{I}_M\right)\boldsymbol{w}[n] + \boldsymbol{U}[n]\boldsymbol{\mathcal{M}}\boldsymbol{e}[n].$$

Comparing this equation to (7), we infer that the continuous-time equivalent to the diffusion LMS algorithm should be

$$\dot{\boldsymbol{w}} = -\gamma_0\left(\boldsymbol{I}_{MN} - \boldsymbol{A}^T \otimes \boldsymbol{I}_M\right)\boldsymbol{w}(t) + \boldsymbol{U}(t)\boldsymbol{\Gamma}\boldsymbol{e}(t), \qquad (10a)$$

$$\boldsymbol{e}(t) = \boldsymbol{d}(t) - \boldsymbol{U}^T(t)\left(\boldsymbol{A}^T \otimes \boldsymbol{I}_M\right)\boldsymbol{w}(t), \qquad (10b)$$

$$U(t) = \begin{bmatrix} u_1(t) & 0_M & \dots & 0_M \\ 0_M & u_2(t) & \dots & 0_M \\ \vdots & \vdots & \ddots & \vdots \\ 0_M & 0_M & \dots & u_N(t) \end{bmatrix}, \tag{10c}$$

$$d(t) = \begin{bmatrix} d_1(t) & d_2(t) & \dots & d_N(t) \end{bmatrix}^T, \tag{10d}$$

$$\Gamma = \text{diag}\{\gamma_1, \dots, \gamma_N\}. \tag{10e}$$

with initial condition $w(0) = w_{\text{init}}$ and positive constants $\{\gamma_0, \dots, \gamma_N\}$. Rewriting (10) in terms of the individual nodes, we obtain

$$\psi_k(t) = \sum_{\ell=1}^{N} a_{\ell k} w_\ell(t), \tag{11a}$$

$$e_k(t) = d_k(t) - u_k^T(t)\psi_k(t), \tag{11b}$$

$$\dot{w}_k(t) = -\gamma_0 \left(w_k(t) - \psi_k(t)\right) + \gamma_k e_k(t) u_k(t). \tag{11c}$$

Note that (11c) reduces to (5) if $a_{\ell k} = 1$ for $\ell = k$ and zero otherwise. We study next the convergence properties of (10).

5. CONVERGENCE AND STABILITY

We first seek the equilibrium points of (10). For that, we need a model for the relation between $d(t)$ and $U(t)$. We assume, as in the discrete-time case (1), that

$$d_k(t) = u_k^T(t) w_o + v_k(t),\ k = 1 \dots N,$$

where w_o is the common parameter vector, and $v_k(t)$ is zero-mean temporally and spatially white-noise that is independent of the regression data for all times. Defining $\mathbb{1} = \begin{bmatrix} 1 & \dots & 1 \end{bmatrix}^T$ and $v(t) = \begin{bmatrix} v_1(t) & \dots & v_N(t) \end{bmatrix}^T$, the overall relation is

$$d(t) = U^T(t) \left(\mathbb{1} \otimes w_o\right) + v(t). \tag{12}$$

Consider first the case of zero noise, i.e., let $v(t) = 0_N$. We show next that in this case, $\mathbb{1} \otimes w_o$ is an equilibrium point of (10). Note from (3) that $A^T \mathbb{1} = \mathbb{1}$. We thus obtain

$$\left(A^T \otimes I_M\right)\left(\mathbb{1} \otimes w_o\right) = \left(A^T \mathbb{1}\right) \otimes \left(I_M w_o\right) = \mathbb{1} \otimes w_o.$$

Substituting this result into the expression for $\dot{w}(t)$ in (10), we conclude that in the absence of noise, $\dot{w}(t) = 0_{MN}$ when $w(t) = \mathbb{1} \otimes w_o$.

We must now study the stability of this equilibrium point. Define the weight error vector $\tilde{w}(t) = \mathbb{1} \otimes w_o - w(t)$. If we add

$$0_{MN} = \gamma_0 \left(I_{MN} - A^T \otimes I_M\right)\left(\mathbb{1} \otimes w_o\right)$$

to the first equation in (10), we obtain

$$\dot{w}(t) = \gamma_0 \left(I_{MN} - A^T \otimes I_M\right) \tilde{w}(t) + U(t)\Gamma e(t). \tag{13}$$

Without noise, substituting $d(t)$ from (12) into (10b) we obtain $e(t) = U^T(t)\left(A^T \otimes I_M\right)\tilde{w}(t)$. Noting also that $\dot{\tilde{w}} = -\dot{w}$, and substituting both results into (13), we obtain the error equation

$$\dot{\tilde{w}}(t) = B(t)\tilde{w}(t), \quad \text{where} \tag{14}$$

$$B(t) = -\gamma_0 I_{MN} + \gamma_0 A^T \otimes I_M - U(t)\Gamma U^T(t)\left(A^T \otimes I_M\right).$$

We prove stability of (14) for the special case of A being symmetric and positive-definite. We call upon a few properties of Kronecker products and Markov matrices, namely [7]:

1. For any F, G, it holds that $(F \otimes G)^T = F^T \otimes G^T$.
2. If λ_i are the eigenvalues of F and ν_j are the eigenvalues of G, then the eigenvalues of $F \otimes G$ are $\lambda_i \nu_j$, for all combinations of i, j.
3. If the entries of $F \in \mathbb{R}^{N \times N}$ are nonnegative and $F\mathbb{1} = \mathbb{1}$, then 1 is an eigenvalue of F, and the other eigenvalues λ_i of F satisfy $\varrho(F) \triangleq \max_{1 \le i \le N} |\lambda_i| = 1$.
4. If a matrix $F \in \mathbb{R}^{N \times N}$ is symmetric, then for any $u \in \mathbb{R}^N$,
$$\lambda_{\min}(F) u^T u \le u^T F u \le \lambda_{\max}(F) u^T u.$$

From properties 1 and 2, if $A = A^T$ is positive-definite, then $A \otimes I_M$ is also symmetric and positive-definite. Define then the candidate Lyapunov function

$$V(t) = \tilde{w}^T(t)\left(A \otimes I_M\right)\tilde{w}(t). \tag{15}$$

From properties 2 and 4,

$$\lambda_{\min}(A)\|\tilde{w}(t)\|^2 \le V(t) \le \lambda_{\max}(A)\|\tilde{w}(t)\|^2,$$

so $V(t)$ may indeed be used as a candidate Lyapunov function. In the absence of noise, its derivative is

$$\begin{aligned} \dot{V}(t) &= \tilde{w}^T(t)\left(A \otimes I_M\right)\dot{\tilde{w}}(t) + \dot{\tilde{w}}^T(t)\left(A \otimes I_M\right)\tilde{w}(t) \\ &= \tilde{w}^T(t)\left[\left(A \otimes I_M\right)B(t) + B^T(t)\left(A \otimes I_M\right)\right]\tilde{w}(t). \end{aligned}$$

We now find conditions under which the matrix between brackets in the last expression is nonpositive definite (and, consequently, 0_{MN} is a stable equilibrium of (14).) Now,

$$\begin{aligned} &\left(A \otimes I_M\right)B(t) + B^T(t)\left(A \otimes I_M\right) \\ &= -2\gamma_0\left(A \otimes I_M - A^2 \otimes I_M\right) \\ &\quad - 2\left(A \otimes I_M\right)U(t)\Gamma U^T(t)\left(A \otimes I_M\right). \end{aligned}$$

We see that $\dot{V}(t) \le 0$ if both

$$D \triangleq \left(I_{MN} - A \otimes I_M\right)\left(A \otimes I_M\right) \quad \text{and}$$

$$F \triangleq \left(A^T \otimes I_M\right)U(t)\Gamma U^T(t)\left(A^T \otimes I_M\right)$$

are nonnegative definite. Consider first $\boldsymbol{D}$. Since $\boldsymbol{A}$ is positive-definite, its eigenvalues λ_i, $1 \le i \le N$ are real and nonnegative. In addition, property 3, implies that

$$0 < \lambda_i \le 1. \tag{16}$$

Let $\boldsymbol{Q}$ be a unitary matrix that diagonalizes $\boldsymbol{A}$, i.e., $\boldsymbol{Q}^T\boldsymbol{Q} = \boldsymbol{Q}\boldsymbol{Q}^T = \boldsymbol{I}_N$, and $\boldsymbol{Q}^T\boldsymbol{A}\boldsymbol{Q} = \text{diag}(\lambda_i) = \boldsymbol{\Lambda}$. Then we have

$$\left(\boldsymbol{Q}^T \otimes \boldsymbol{I}_M\right) \boldsymbol{D} \left(\boldsymbol{Q} \otimes \boldsymbol{I}_M\right) = (\boldsymbol{\Lambda} - \boldsymbol{\Lambda}^2) \otimes \boldsymbol{I}_M,$$

which is nonnegative-definite from (16).

The other matrix, $\boldsymbol{F}$, is always nonnegative-definite, since $\boldsymbol{\Gamma}$ is positive-definite by construction. We can therefore conclude that $\dot{V}(t) \le 0$. From Lyapunov theory, this means that the origin $\boldsymbol{0}_{MN}$ will be a stable equilibrium point of (14) when the noise is zero [5].

When the regressors $\boldsymbol{u}_k(t)$ satisfy a persistence of excitation condition, the system will remain stable in the presence of noise. This can be seen as follows. Assume that the matrix

$$\boldsymbol{\Phi}(t_0, T) \triangleq \int_{t_0}^{t_0+T} (\boldsymbol{A} \otimes \boldsymbol{I}_M)\, \boldsymbol{U}(t) \boldsymbol{\Gamma} \boldsymbol{U}^T(t) \,(\boldsymbol{A} \otimes \boldsymbol{I}_M)\, dt$$

is such that there exist $0 < \alpha_1 \le \alpha_2 < \infty$ and $0 < T < \infty$ for which, for all $t_0 \ge 0$,

$$\alpha_1 \boldsymbol{I}_{MN} \le \boldsymbol{\Phi}(t_0, T) \le \alpha_2 \boldsymbol{I}_{MN}. \tag{17}$$

In this case, the stability proof in [5] can be modified to show that, in the absence of noise, the origin will be exponentially stable. Exponential stability implies, through the total stability theorem [8, Lemma 5.2, p. 213], that the origin in (14) will remain stable in the presence of sufficiently small bounded noise.

Note that the CT-diffusion LMS may remain stable even for a non-definite $\boldsymbol{A}$, as the first example in next section shows. However, the condition of positive-definite $\boldsymbol{A}$ is necessary to guarantee stability for any input regressor $\boldsymbol{U}(t)$ and any values of γ_0 and $\boldsymbol{\Gamma}$.

6. SIMULATIONS

We tested the proposed estimation method using Simulink, in an example with three nodes and $\boldsymbol{w}_o = \begin{bmatrix} 0.5 & -0.2 \end{bmatrix}^T$. The input sequences are

$$\boldsymbol{u}_1(t) = \begin{bmatrix} \sin(2\pi 10 t) & 0.5 \sin(2\pi 10 t + \pi/3) \end{bmatrix}^T,$$
$$\boldsymbol{u}_2(t) = \begin{bmatrix} 0.2 \sin(2\pi 5 t) & 0.8 \sin(2\pi 5 t - \pi/2) \end{bmatrix}^T,$$
$$\boldsymbol{u}_3(t) = \begin{bmatrix} u_{31}(t) & u_{32}(t) \end{bmatrix}^T,$$

in which $u_{31}(t)$ is a Gaussian white noise with power equal to 1, and $u_{32}(t)$ is a second independent Gaussian white noise with power 0.2. The noises $v_1(t)$, $v_2(t)$ and $v_3(t)$ are also Gaussian and white, independent of other signals, with powers 10^{-4}, 10^{-2}, $2 \cdot 10^{-4}$, respectively. The combination matrix used was

$$\boldsymbol{A} = \begin{bmatrix} 0.6 & 0 & 0.4 \\ 0 & 0.7 & 0.3 \\ 0.4 & 0.3 & 0.3 \end{bmatrix}.$$

Note that this matrix is not positive-definite. The resulting estimates for each node can be seen in Figs. 1–3 (in all examples, we plot the results from a single realization of the filter.) Note how all estimates gravitate around the true values, for all nodes. Note how the combined estimates $\boldsymbol{\psi}_k(t)$ tend to have faster convergence and more similar performance across nodes. The combined estimate $\boldsymbol{\psi}_k(t)$ for the third node converges very quickly (the covariance matrix for $\boldsymbol{u}_3(t)$ has a small eigenvalue spread.)

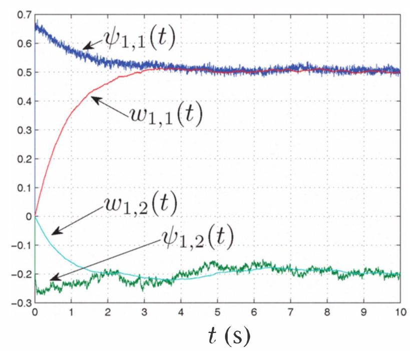

Fig. 1. Parameter estimates $\boldsymbol{w}_1(t)$ and $\psi_1(t)$ for node 1.

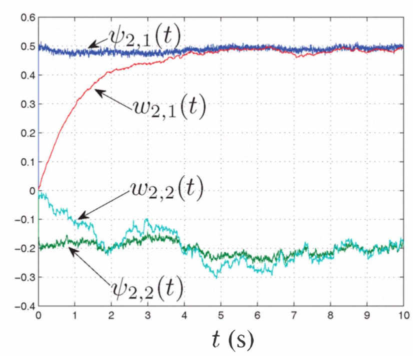

Fig. 2. Parameter estimates $\boldsymbol{w}_2(t)$ and $\psi_2(t)$ for node 2.

For comparison, we plot in Fig. 4 the first elements $w_{k,1}(t)$ of the estimates at each node for the case of independent filters, i.e., when $\boldsymbol{A} = \boldsymbol{I}$. It can be seen that the cooperative estimation scheme tends to equalize the responses across all nodes despite their different SNR conditions. The third node converges very quickly, but the convergence rate of the second node is much slower without cooperation.

As an example of a larger network, we simulated the 10-node network shown in Fig. 5 using diffusion (the entries of

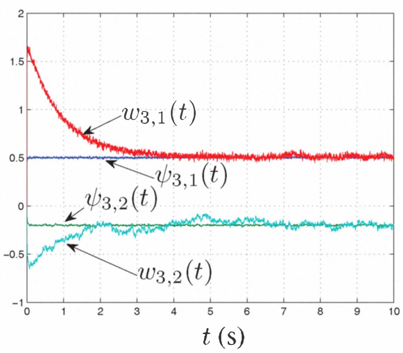

Fig. 3. Parameter estimates $\boldsymbol{w}_3(t)$ and $\psi_3(t)$ for node 3.

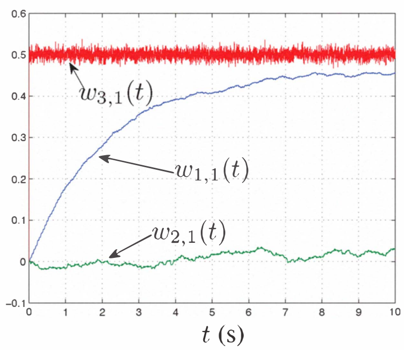

Fig. 4. Parameter estimates $w_{i,1}(t)$ without diffusion.

$a_{\ell k}$ are given next to the edges in Fig. 5 — the values of a_{kk} are chosen so that condition (3) is satisfied). Note that, in this case, $\boldsymbol{A}$ is positive-definite. We also plot the results obtained without diffusion ($\boldsymbol{A} = \boldsymbol{I}$). The inputs are similar to those in the previous example: sinusoids and random noise (in this case, low-pass filtered Gaussian noise). We compare in Fig. 6 the estimates obtained for $\boldsymbol{w}_6(t)$.

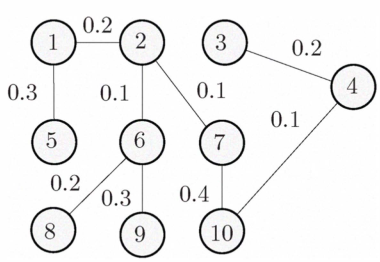

Fig. 5. 10-node network.

7. CONCLUSION

We proposed a continuous-time adaptive diffusion strategy for distributed estimation, and determined conditions under which the algorithm is exponentially stable in the absence of

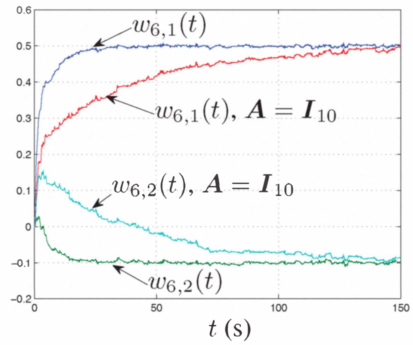

Fig. 6. Parameter estimates $\boldsymbol{w}_6(t)$ for node 6, for two choices of $\boldsymbol{A}$ (with and without cooperation).

noise, and stable in the presence of bounded noise. Our stability proof requires the weight matrix $\boldsymbol{A}$ to be positive-definite. We are currently working to relax this positiveness condition in our proof.

8. REFERENCES

[1] F. S. Cattivelli and A. H. Sayed, "Modeling bird flight formations using diffusion adaptation," *IEEE Trans. Signal Process.*, vol. 59, no. 5, pp. 2038–2051, 2011.

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

[3] 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.

[4] 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.

[5] P. A. Ioannou and J. Sun, *Robust Adaptive Control.* Prentice Hall, NJ, 1996.

[6] A. Bhaya and E. Kaszkurewicz, "A control-theoretic approach to the design of zero finding numerical methods," *IEEE Trans. Autom. Control*, vol. 52, no. 6, pp. 1014–1026, 2007.

[7] R. A. Horn and C. R. Johnson, *Matrix Analysis.* Cambridge University Press, 1987.

[8] H. K. Khalil, *Nonlinear Systems*, 2nd ed. Prentice Hall, NJ, 1996.