STEADY-STATE PERFORMANCE OF ADAPTIVE DIFFUSION LEAST-MEAN SQUARES

Cassio G. Lopes and Ali H. Sayed

Department of Electrical Engineering
University of California
Los Angeles, CA, 90095.
Email: {cassio, sayed@ee.ucla.edu}

ABSTRACT

We study the mean-square performance of a diffusion least mean-squares protocol proposed in recent work to address the problem of distributed estimation [1, 2]. By relying on energy conservation arguments [8] we derive closed form expressions for the mean-square deviation (MSD) and the excess mean-square error (EMSE) of the adaptive network. Examples show a good agreement between simulations and theory.

Index Terms— Adaptive filters, distributed estimation, adaptive estimation, adaptive networks, cooperative systems, mean-square analysis, diffusion algorithm.

1. INTRODUCTION

Recently, distributed and cooperative estimation algorithms have been proposed to address the problem of distributed estimation based on the concept of adaptive networks [1, 3, 6]. The problem arises in several applications, such as environment monitoring, target localization and sensor network applications [7]. In [1] static and dynamic diffusion protocols that solve the global estimation problem in a cooperative fashion have been proposed, and the mean behavior of the static diffusion algorithm has been studied. The algorithms formulated therein solve the estimation problem in a fully distributed manner by sharing the computational burden among the individual nodes. Every node is equipped with local learning abilities and cooperates *only* with its direct neighbors, giving rise to peer-to-peer protocols that are able to respond in real time to changes in the environment. In this work we extend the analysis performed in [1] and study the mean-square steady-state performance of an adaptive network driven by a diffusion protocol. As a result, we provide closed form expressions for the mean-square deviation (MSD) and excess mean-square error (EMSE). Examples show a good match between simulations and theory.

2. DIFFUSION LMS

The global estimation problem consists of a network of interconnected nodes pursuing the same $M \times 1$ unknown vector w^o from measurements collected at the N nodes (see Fig.1). Each node k has access to time realizations $\{d_k(i), u_{k,i}\}$, $k = 1, \ldots, N$, of zero-mean random data $\{\boldsymbol{d}_k, \boldsymbol{u}_k\}$, with $d_k(i)$ a scalar measurement and $u_{k,i}$ a regression row vector; both at time i. In order to accomplish the estimation task we rely on adaptive networks embedded with a peer-to-peer diffusion learning protocol as proposed in [1, 2]. An adaptive network results from equipping the nodes of the network with local learning rules. The available communication topology is then employed to implement a cooperation protocol among the nodes in order to efficiently exploit spatial and temporal information. Different learning rules allied with different cooperation protocols give rise to different adaptive networks. The diffusion protocol is comprised of an aggregate step followed by an adaptation procedure based on an LMS rule (or other similar rules). Since in general every node k in a neighborhood tends to have a different neighborhood for connected topologies, the aggregation step helps fuse information from nodes across the network into node k. Assuming each node k has an unbiased estimate $\psi_k^{(i-1)}$ of w^o at time $i-1$, in the aggregate step a vector combining estimates from the nearby nodes, i.e.,

$$\phi_k^{(i-1)} = f_k\left(\psi_\ell^{(i-1)}; \ell \in \mathcal{N}_{k,i-1}\right) \quad (1)$$

is generated, for some local combiner $f_k(\cdot)$, and where $\mathcal{N}_k$ is the set of nodes connected to node k, including itself. The aggregate estimate is then fed into the local adaptive filter and the new local estimate $\psi_k^{(i)}$ is generated after an adaptive rule, say of the LMS type, as

$$\psi_k^{(i)} = \phi_k^{(i-1)} + \mu_k u_{k,i}^* \left(d_k(i) - u_{k,i}\phi_k^{(i-1)}\right) \quad (2)$$

where μ_k is a local step-size. The combiners $f_k(\cdot)$ might be nonlinear or even time variant, to reflect, for instance, changing topologies or to respond to non-stationary environments.

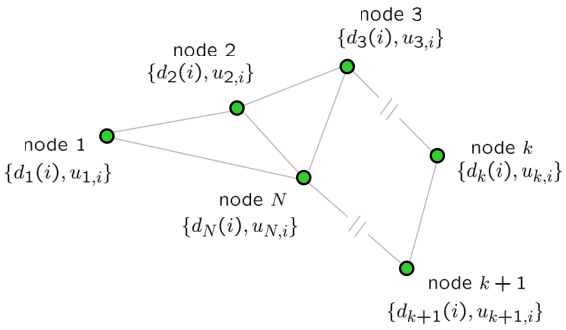

Fig. 1. A distributed network with N nodes.

This material was based on work supported in part by the National Science Foundation under award ECS-0601266. The work of Mr. Lopes was also supported by a fellowship from CAPES, Brazil, under award 1168/01-0.

In order to illustrate the technique, we explore a linear combiner model. At node k, the aggregated estimate $\phi_k^{(i-1)}$ is generated by linearly combining the neighbors' estimates, i.e., by using

$$\begin{aligned}\phi_k^{(i-1)} &= \sum_{\ell\in\mathcal{N}_k} c(k,\ell)\,\psi_\ell^{(i-1)} \\ \psi_k^{(i)} &= \phi_k^{(i-1)} + \mu_k u_{k,i}^*\left(d_k(i) - u_{k,i}\phi_k^{(i-1)}\right) \end{aligned} \quad (3)$$

for a set of *local* combiners $c(k,\ell)$ satisfying

$$\sum_{\ell\in\mathcal{N}_k} c(k,\ell) = 1 \quad (4)$$

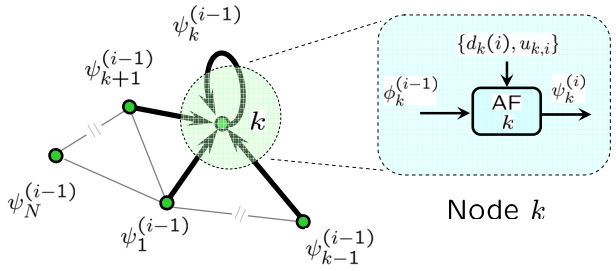

Fig. 2. A network with diffusion cooperation strategy.

One such combiner is the Metropolis rule. Let n_k and n_ℓ be the degree for nodes k and ℓ, i.e., $n_k = |\mathcal{N}_k|$. We have

$$\begin{cases} c(k,\ell) = 1/\max(n_k, n_\ell) & \text{if } k\neq\ell \text{ are } \textit{linked} \\ c(k,\ell) = 0 & \text{if } k \text{ and } \ell \text{ are not } \textit{linked} \\ c(k,k) = 1 - \sum_{\ell\in\mathcal{N}_k/k} c(k,\ell) & \text{for } k=\ell \end{cases} \quad (5)$$

Other possible rules are the Laplacian and the nearest neighbor rules. The Laplacian is given by

$$C = I_N - \kappa L\,, \qquad L = \mathcal{D} - A_d \quad (6)$$

where L is the network Laplacian, A_d is the network adjacency matrix, $\mathcal{D} = \mathrm{diag}\{n_1,\ldots,n_N\}$, and $\kappa = 1/n_{\max}$. The nearest neighbor rule is given by

$$c(k,\cdot) = \frac{1}{|\mathcal{N}_k|} \quad (7)$$

3. ANALYSIS FRAMEWORK

Algorithm (3) embeds the combined effect of several adaptive filter updates, in addition to the network topology. Hence, performance analysis is challenging. We resort to state-space representations and introduce the global quantities:

$$\boldsymbol{\psi}^i \triangleq \mathrm{col}\{\boldsymbol{\psi}_1^{(i)},\ldots,\boldsymbol{\psi}_N^{(i)}\}\,, \quad \boldsymbol{\phi}^{i-1} \triangleq \mathrm{col}\{\boldsymbol{\phi}_1^{(i-1)},\ldots,\boldsymbol{\phi}_N^{(i-1)}\}$$
$$\boldsymbol{U}_i \triangleq \mathrm{diag}\{\boldsymbol{u}_{1,i},\ldots,\boldsymbol{u}_{N,i}\}\,, \quad \boldsymbol{d}_i \triangleq \mathrm{col}\{\boldsymbol{d}_1(i),\ldots,\boldsymbol{d}_N(i)\}$$

in terms of the stochastic quantities whose realizations appear in (3). Now let

$$D = \mathrm{diag}\{\mu_1 I_M, \mu_2 I_M, \ldots, \mu_N I_M\} \quad (8)$$

be a diagonal matrix collecting the local step-sizes. In addition, the measurements are assumed to obey the linear model

$$\boldsymbol{d}_k(i) = \boldsymbol{u}_{k,i} w^o + \boldsymbol{v}_k(i) \quad (9)$$

where $\boldsymbol{v}_k(i)$ is the background noise, assumed independent over time and space and with variance $\sigma_{v,k}^2$. A global data model is given by

$$\boldsymbol{d}_i = \boldsymbol{U}_i w^{(o)} + \boldsymbol{v}_i \quad (10)$$

where

$$\boldsymbol{v}_i = \mathrm{col}\{\boldsymbol{v}_1(i),\ldots,\boldsymbol{v}_N(i)\}$$

and

$$w^{(o)} = q_N \otimes w^o\,, \quad q_N = \mathrm{col}\{1,\ldots,1\} \quad (N\times 1)$$

Let

$$G = C \otimes I_M$$

where $C = [c(k,\ell)]$ is a diffusion combination matrix satisfying $Cq_N = q_N$ and it has information about the network topology: a nonzero entry $c(k,\ell)$ means that node k is connected to node ℓ. Since $Gw^{(o)} = w^{(o)}$, it can be verified that

$$\begin{aligned}\boldsymbol{e}_i &= \boldsymbol{d}_i - \boldsymbol{U}_i G\boldsymbol{\psi}^{i-1} \\ &= \boldsymbol{e}_{a,i}^G + \boldsymbol{v}_i \end{aligned} \quad (11)$$

where

$$\boldsymbol{e}_{a,i}^G = \boldsymbol{U}_i G\widetilde{\boldsymbol{\psi}}^{i-1} \quad \text{and} \quad \widetilde{\boldsymbol{\psi}}^{i-1} \triangleq w^{(o)} - \boldsymbol{\psi}^{i-1} \quad (12)$$

The local rule (3) and the related weight error vector $\widetilde{\boldsymbol{\psi}}^{(i)}$ can be represented globally in a compact state-space form as:

$$\begin{aligned}\boldsymbol{\psi}^i &= G\boldsymbol{\psi}^{i-1} + D\boldsymbol{U}_i^*\left(\boldsymbol{d}_i - \boldsymbol{U}_i G\boldsymbol{\psi}^{i-1}\right) \\ &= G\widetilde{\boldsymbol{\psi}}^{i-1} - D\boldsymbol{U}_i^*\boldsymbol{e}_i \end{aligned} \quad (13)$$

3.1. Weighted Energy and Variance Relation

Initially we characterize the network performance in steady-state by the global quantities MSD and EMSE, which are defined as

$$\eta = \frac{1}{N}\lim_{i\to\infty} E\|\widetilde{\boldsymbol{\psi}}^{i-1}\|^2 \qquad \text{(MSD)} \quad (14)$$

$$\zeta = \frac{1}{N}\lim_{i\to\infty} E\|\widetilde{\boldsymbol{\psi}}^{i-1}\|_{R_u}^2 \qquad \text{(EMSE)} \quad (15)$$

We will evaluate the performance of the diffusion network by extending the energy conservation approach of [4, 8] to this case. Introduce the global *a priori* and *a posteriori* weighted estimation errors:

$$\boldsymbol{e}_{a,i}^{D\Sigma G} = \boldsymbol{U}_i D\Sigma G\widetilde{\boldsymbol{\psi}}^{i-1} \quad \text{and} \quad \boldsymbol{e}_{p,i}^{D\Sigma} = \boldsymbol{U}_i D\Sigma\widetilde{\boldsymbol{\psi}}^{i} \quad (16)$$

for some arbitrary $\Sigma \geq 0$. Substituting (11) into (13) and performing weighted energy balance on both sides, the cross terms cancel out and we are left with the following energy relation:

$$\begin{aligned}\|\widetilde{\boldsymbol{\psi}}^i\|_\Sigma^2 + \left(\boldsymbol{e}_{a,i}^{D\Sigma G}\right)^*\boldsymbol{e}_{a,i}^G + \left(\boldsymbol{e}_{a,i}^G\right)^*\boldsymbol{e}_{a,i}^{D\Sigma G} &= \|\widetilde{\boldsymbol{\psi}}^{i-1}\|_{G^*\Sigma G}^2 \\ + \left(\boldsymbol{e}_{a,i}^G\right)^*\boldsymbol{U}_i D\Sigma D\boldsymbol{U}_i^*\boldsymbol{e}_{a,i}^G + \boldsymbol{v}_i^*\boldsymbol{U}_i D\Sigma D\boldsymbol{U}_i^*\boldsymbol{v}_i \end{aligned} \quad (17)$$

Substituting the error definitions (12) into (17) yields

$$\begin{aligned} E\|\widetilde{\boldsymbol{\psi}}^{i}\|^2_{\Sigma} &= E\|\widetilde{\boldsymbol{\psi}}^{i-1}\|^2_{\boldsymbol{\Sigma}'} + E\boldsymbol{v}_i^*\boldsymbol{U}_i D\Sigma D\boldsymbol{U}_i^*\boldsymbol{v}_i \quad (18)\\ \boldsymbol{\Sigma}' &= G^*\Sigma G - G^*\Sigma D\boldsymbol{U}_i^*\boldsymbol{U}_i G - G^*\boldsymbol{U}_i^*\boldsymbol{U}_i D\Sigma G \\ &+ G^*\boldsymbol{U}_i^*\boldsymbol{U}_i D\Sigma D\boldsymbol{U}_i^*\boldsymbol{U}_i G \quad (19)\end{aligned}$$

Note that no assumptions are needed to arrive at (18)–(19). Nevertheless, we need to rely on simplifying assumptions to make the analysis tractable. We proceed by assuming temporal independence of the regression data, as is common in the analysis of traditional adaptive schemes [8]. In addition, we assume spatial independence, as it is more likely to hold in the distributed domain. An important remark is that these assumptions do not compromise the spatial-temporal nature of the algorithm, neither its distributiveness [4]. It is assumed *only* for analysis purposes. As a result, after taking expectations, (18) yields

$$\begin{aligned} E\|\widetilde{\boldsymbol{\psi}}^{i}\|^2_{\Sigma} &= E\|\widetilde{\boldsymbol{\psi}}^{i-1}\|^2_{\Sigma'} + E\boldsymbol{v}_i^*\boldsymbol{U}_i D\Sigma D\boldsymbol{U}_i^*\boldsymbol{v}_i \quad (20)\\ \Sigma' &= G^*\Sigma G - G^*\Sigma D R_u G - G^* R_u D\Sigma G \\ &+ G^* E\boldsymbol{U}_i^*\boldsymbol{U}_i D\Sigma D\boldsymbol{U}_i^*\boldsymbol{U}_i G \quad (21)\end{aligned}$$

where $R_u = E\boldsymbol{U}_i^*\boldsymbol{U}_i = \mathrm{diag}\{R_{u,1},\ldots,R_{u,N}\}$, with $R_{u,k} = E\boldsymbol{u}_{k,i}^*\boldsymbol{u}_{k,i}$.

3.2. Gaussian Signals

In (20) and (21), one needs to calculate data moments that may be challenging in the general case. In this work we treat the Gaussian case. Thus, we assume the regressors arise from circular Gaussian sources [8]. By performing the eigen-decomposition $R_u = T\Lambda T^*$, where T is unitary, $\Lambda = \mathrm{diag}\{\Lambda_1,\ldots,\Lambda_N\}$, $\Lambda_k > 0$ and diagonal, and defining the transformed quantities

$$\overline{\boldsymbol{\psi}}^{(i)} = T^*\widetilde{\boldsymbol{\psi}}^{(i)} \qquad \overline{\boldsymbol{U}}_i = \boldsymbol{U}_i C \qquad \overline{\Sigma} = T^*\Sigma T$$
$$\overline{\Sigma}' = T^*\Sigma' T \qquad \overline{G} = T^* G T \qquad \overline{D} = T^* D T = D$$

the variance relation (20) becomes

$$\begin{aligned} E\|\overline{\boldsymbol{\psi}}^{(i)}\|^2_{\overline{\Sigma}} &= E\|\overline{\boldsymbol{\psi}}^{(i-1)}\|^2_{\overline{\Sigma}'} + E\boldsymbol{v}_i^*\overline{\boldsymbol{U}}_i D\overline{\Sigma} D\overline{\boldsymbol{U}}_i^*\boldsymbol{v}_i \quad (22)\\ \overline{\Sigma}' &= \overline{G}^*\overline{\Sigma}\overline{G} - \overline{G}^*\overline{\Sigma}D\Lambda\overline{G} - \overline{G}^*\Lambda D\overline{\Sigma}\overline{G} \\ &+ \overline{G}^*\left(E\overline{\boldsymbol{U}}_i^*\overline{\boldsymbol{U}}_i D\overline{\Sigma}D\overline{\boldsymbol{U}}_i^*\overline{\boldsymbol{U}}_i\right)\overline{G} \quad (23)\end{aligned}$$

The transformed variance relation (22)–(23) requires the calculation of data moments, which can be accomplished by exploiting the block structure of the quantities therein. To do so, we resort to the block vector operator $\sigma = \mathrm{bvec}\{\Sigma\}$, which converts a block matrix Σ into a single column vector σ in *two steps* as follows. Let Σ be a $NM \times NM$ block matrix

$$\Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} & \cdots & \Sigma_{1\ell} & \cdots & \Sigma_{1N} \\ \Sigma_{21} & \Sigma_{22} & \cdots & \Sigma_{2\ell} & \cdots & \Sigma_{2N} \\ \vdots & \vdots & & \vdots & \ddots & \vdots \\ \Sigma_{N1} & \Sigma_{N2} & \cdots & \Sigma_{N\ell} & \cdots & \Sigma_{NN} \end{bmatrix} \quad (24)$$

where each block $\Sigma_{k\ell}$ is $M \times M$. First, the block columns

$$\Sigma_\ell = \mathrm{col}\{\Sigma_{1\ell},\ldots,\Sigma_{N\ell}\}, \quad \ell = 1,\ldots,N$$

are stacked on top of each other, yielding the matrix

$$\Sigma^c = \begin{bmatrix} \Sigma_1 \\ \Sigma_2 \\ \vdots \\ \Sigma_N \end{bmatrix} \quad (25)$$

Note that Σ^c is $N^2M \times M$. Subsequently, each $\Sigma_{k\ell}$ is vectorized by using the standard vec$\{\}$ operator:

$$\sigma_\ell = \mathrm{col}\{\sigma_{1\ell},\ldots,\sigma_{N\ell}\}, \ \text{with} \quad \sigma_{k\ell} = \mathrm{vec}\{\Sigma_{k\ell}\} \quad (26)$$

and we finally let

$$\sigma = \mathrm{col}\{\sigma_1,\sigma_2,\ldots,\sigma_N\} \quad (27)$$

We thus write $\sigma = \mathrm{bvec}\{\Sigma\}$ to denote the conversion of Σ into a single column. We also write $\Sigma = \mathrm{bvec}\{\sigma\}$ to recover the original block matrix form of the column vector σ. We further define the *block Kronecker product* of two block matrices A and B, which is denoted by $A \odot B$. Its kl-block is defined as

$$[A\odot B]_{k\ell} = \begin{bmatrix} A_{k\ell}\otimes B_{11} & \cdots & A_{k\ell}\otimes B_{1N} \\ \vdots & \ddots & \vdots \\ A_{k\ell}\otimes B_{N1} & \cdots & A_{k\ell}\otimes B_{NN} \end{bmatrix} \quad (28)$$

for $k,\ell = 1,\ldots,N$. The block vector operator (27) and the block Kronecker product (28) are related via [5]:

$$\mathrm{bvec}\{A\Sigma B\} = (B \odot A^T)\sigma \quad (29)$$

We now use these notations to evaluate the required data moments in (22)–(23), namely,

$$E\boldsymbol{v}_i^*\overline{\boldsymbol{U}}_i D\overline{\Sigma}D\overline{\boldsymbol{U}}_i^*\boldsymbol{v}_i \qquad \text{and} \qquad E\overline{\boldsymbol{U}}_i^*\overline{\boldsymbol{U}}_i D\overline{\Sigma}D\overline{\boldsymbol{U}}_i^*\overline{\boldsymbol{U}}_i \quad (30)$$

The first term in (30) can be verified to be

$$E\boldsymbol{v}_i^*\overline{\boldsymbol{U}}_i D\overline{\Sigma}D\overline{\boldsymbol{U}}_i^*\boldsymbol{v}_i = \mathrm{Tr}\{\Lambda_v E\overline{\boldsymbol{U}}_i D\overline{\Sigma}D\overline{\boldsymbol{U}}_i^*\} \quad (31)$$

where $\Lambda_v > 0$ is a diagonal matrix given by

$$\Lambda_v = \{\sigma_{v,1}^2,\sigma_{v,2}^2,\ldots,\sigma_{v,N}^2\}$$

To arrive at (31), first note that the $k\ell$ entry of $E\overline{\boldsymbol{U}}_i D\overline{\Sigma}D\overline{\boldsymbol{U}}_i^*$ is given by

$$[E\overline{\boldsymbol{U}}_i D\overline{\Sigma}D\overline{\boldsymbol{U}}_i^*]_{k\ell} = \begin{cases} 0, & \text{for } k\neq\ell \\ \mu_k^2\mathrm{Tr}(\Lambda_k\overline{\Sigma}_{kk}) = \mu_k^2\lambda_k^T\overline{\sigma}_{kk}, & \text{for } k=\ell \end{cases}$$

where $\lambda_k = \mathrm{vec}\{\Lambda_k\}$ and $\overline{\sigma}_{k\ell} = \mathrm{vec}\{\overline{\Sigma}_{k\ell}\}$. Hence, we may write (31) as

$$\boxed{E\boldsymbol{v}_i^*\overline{\boldsymbol{U}}_i^* D\overline{\Sigma}D\overline{\boldsymbol{U}}_i\boldsymbol{v}_i = b^T\overline{\sigma}} \quad (32)$$

with $b = \mathrm{bvec}\{R_v D^2\Lambda\}$, $R_v = \Lambda_v \odot I_M$ and $\overline{\sigma} = \mathrm{bvec}\{\overline{\Sigma}\}$.

The fourth order term in (30) is challenging, but it can be handled by appealing to a Gaussian factorization theorem [8]. We start by using (29) and write

$$\mathrm{bvec}\{\overline{G}^* E\overline{\boldsymbol{U}}_i^*\overline{\boldsymbol{U}}_i D\overline{\Sigma}D\overline{\boldsymbol{U}}_i^*\overline{\boldsymbol{U}}_i\overline{G}\} = \left(\overline{G}\odot\overline{G}^{*T}\right)\mathrm{bvec}\{E\overline{\boldsymbol{U}}_i^*\overline{\boldsymbol{U}}_i D\overline{\Sigma}D\overline{\boldsymbol{U}}_i^*\overline{\boldsymbol{U}}_i\}$$

Now considering that $\overline{\boldsymbol{U}}_i^*\overline{\boldsymbol{U}}_i$ and D are block diagonal and defining the quantity $A = E\overline{\boldsymbol{U}}_i^*\overline{\boldsymbol{U}}_i\overline{\Sigma}\overline{\boldsymbol{U}}_i^*\overline{\boldsymbol{U}}_i$ we have

$$\begin{aligned} \mathrm{bvec}\{\overline{G}^* E\overline{\boldsymbol{U}}_i^*\overline{\boldsymbol{U}}_i D\overline{\Sigma}D\overline{\boldsymbol{U}}_i^*\overline{\boldsymbol{U}}_i\overline{G}\} &= \left(\overline{G}\odot\overline{G}^{*T}\right)(D\odot D)\,\mathrm{bvec}\{A\} \\ &= \left(\overline{G}\odot\overline{G}^{*T}\right)(D\odot D)\,\mathcal{A}\overline{\sigma} \quad (33)\end{aligned}$$

where $\mathcal{A} = \mathrm{diag}\{\mathcal{A}_1,\mathcal{A}_2,\ldots,\mathcal{A}_N\}$ with

$$\mathcal{A}_\ell = \mathrm{diag}\left\{\Lambda_1\otimes\Lambda_\ell,\ldots,\lambda_\ell\lambda_\ell^T + \gamma\Lambda_\ell\otimes\Lambda_\ell,\ldots,\Lambda_N\otimes\Lambda_\ell\right\}$$

and $\overline{\sigma} = \overline{\Sigma}$. The intermediate steps in (33) were omitted for space considerations.

Grouping the results and substituting into (22) yields

$$\boxed{\begin{aligned} E\|\overline{\psi}^{i}\|^2_{\overline{\sigma}} &= E\|\overline{\psi}^{i-1}\|^2_{\overline{F}\,\overline{\sigma}} + b^T\overline{\sigma} \qquad (34)\\ \overline{F} &= \left(\overline{G}\odot\overline{G}^{*T}\right)\left[I-(I\odot\Lambda D)-(\Lambda D\odot I)+(D\odot D)\mathcal{A}\right]\overline{\sigma} \qquad (35)\end{aligned}}$$

3.3. Steady-State Performance

When the network is operating in steady-state, we get from (34)

$$E\|\overline{\psi}^{\infty}\|^2_{(I-\overline{F})\overline{\sigma}} = b^T\overline{\sigma}, \quad \text{as } i\to\infty \qquad (36)$$

To calculate the global MSD and the EMSE we need to evaluate the weighted norms

$$E\|\overline{\psi}^{\infty}\|^2_I \quad \text{(MSD)} \quad \text{and} \quad E\|\overline{\psi}^{\infty}\|^2_\Lambda \quad \text{(EMSE)} \qquad (37)$$

which are equivalent to $E\|\overline{\psi}^{\infty}\|^2_q$ and $E\|\overline{\psi}^{\infty}\|^2_\lambda$, by using vectorization notation [4, 8]. Thus the global quantities MSD and EMSE define two underlying linear systems in (36):

$$\left(I-\overline{F}\right)\overline{\sigma}_\eta = q \qquad \text{and} \qquad \left(I-\overline{F}\right)\overline{\sigma}_\zeta = \lambda \qquad (38)$$

solving (38) for $\overline{\sigma}_\eta$ and $\overline{\sigma}_\zeta$ and substituting the results back into (36) yields

$$\boxed{\eta = \frac{1}{N}b^T\left(I-\overline{F}\right)^{-1}q \qquad \text{(MSD)}} \qquad (39)$$

$$\boxed{\zeta = \frac{1}{N}b^T\left(I-\overline{F}\right)^{-1}\lambda \qquad \text{(EMSE)}} \qquad (40)$$

3.4. Local Node Performance

We may also retrieve the individual node steady-state quantities from the global quantity $E\|\overline{\psi}^{i}\|^2_{\overline{\sigma}}$. To do so, we define the following spatial filtering matrices whose purpose is to extract the local quantities from the global expressions:

$$J_{q,k} = \text{diag}\{\mathbf{0}_{(k-1)M}, I_M, \mathbf{0}_{(N-k)M}\} \qquad \text{(MSD)} \qquad (41)$$

$$J_{\lambda,k} = \text{diag}\{\mathbf{0}_{(k-1)M}, \Lambda_k, \mathbf{0}_{(N-k)M}\} \qquad \text{(EMSE)} \qquad (42)$$

where Λ_k is the diagonal matrix with the eigenvalues corresponding to node k and $\mathbf{0}_L$ is a block of $L\times L$ zeros. It is possible to extract local node performance by exploiting the degree of freedom in selecting the weighting matrices $\overline{\sigma}$ in (36). To begin with, note that the local mean-square performance of node k is defined as

$$\eta_k = E\|\overline{\psi}_k^{(\infty)}\|^2 \quad \text{and} \quad \zeta_k = E\|\overline{\psi}_k^{(\infty)}\|^2_{\lambda_k} \qquad (43)$$

in terms of the local stationary vectors $\overline{\psi}_k^{(\infty)}$ and where $\lambda_k = \text{vec}\{\Lambda_k\}$. Now, inspecting the global steady-state quantities (14) and (15) and considering the block diagonal structure of Λ, we get

$$\eta_k = E\|\overline{\psi}^{\infty}\|^2_{J_{q,k}} \qquad \text{and} \qquad \zeta_k = E\|\overline{\psi}^{\infty}\|^2_{J_{\lambda,k}}$$

Thus we select the $\overline{\sigma}$ in (36) as the solution to the (local) linear systems of equations

$$\left(I-\overline{F}\right)\overline{\sigma}_\eta = \text{bvec}\{J_{q,k}\} \qquad \text{(MSD)} \qquad (44)$$

$$\left(I-\overline{F}\right)\overline{\sigma}_\zeta = \text{bvec}\{J_{\lambda,k}\} \qquad \text{(EMSE)} \qquad (45)$$

so that

$$\boxed{\eta_k = b^T\left(I-\overline{F}\right)^{-1}\text{bvec}\{J_{q,k}\} \qquad \text{(MSD)}} \qquad (46)$$

and

$$\boxed{\zeta_k = b^T\left(I-\overline{F}\right)^{-1}\text{bvec}\{J_{\lambda,k}\} \qquad \text{(EMSE)}} \qquad (47)$$

4. SIMULATIONS AND CONCLUDING REMARKS

We run a simulation example to illustrate the efficiency of the diffusion protocol as compared with no cooperation, as well as to validate the analysis. For simplicity, a relatively small network has been selected, only to illustrate the algorithm. However, networks with arbitrary size have been tested. In this example, the signals are Gaussian and follow a first order Markov process [1, 4], i.e., The regressors are generated as

$$u_k(i) = \alpha_k u_k(i-1) + \sqrt{\sigma^2_{u,k}(1-\alpha^2_k)}\cdot z_k(i) \qquad (48)$$

where $\sigma^2_{u,k}$ is the node regressor variance, α_k is the correlation index and $z_k(i)$ is a realization of a local white noise process with $\sigma^2_{z,k} = 1$, independent from other nodes as well. The network has $N = 8$ nodes and the local adaptive filters have order $M = 4$. Figures 3 and 4 depict the network topology as well as its statistical profile, respectively.

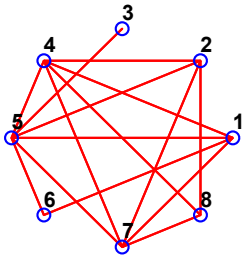

Fig. 3. Network topology.

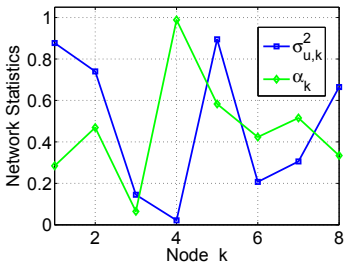

Fig. 4. Network statistics for Example 1.

The background noise was set to $\sigma^2_{v,k} = 10^{-3}$. One can see in Figs. 5 and 6 that the network operating under the cooperative diffusion protocol presents a global improvement in performance, as compared to the non-cooperative case. The curves also show that the

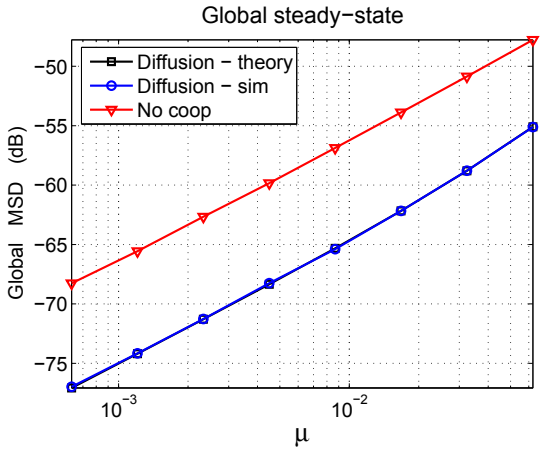

Fig. 5. Global MSD.

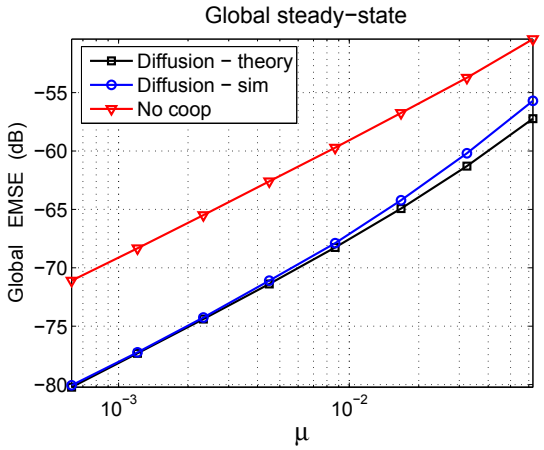

Fig. 6. Global EMSE.

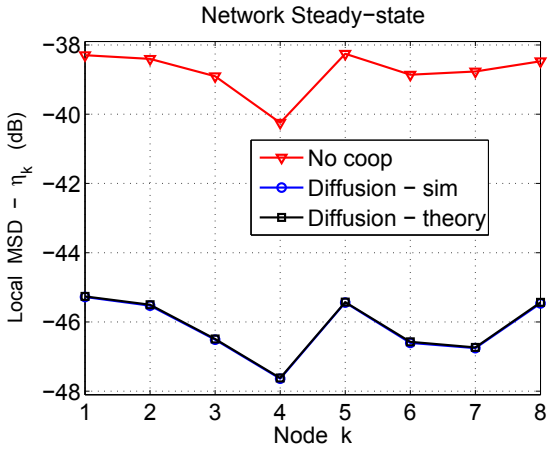

Fig. 7. Local MSD (per node).

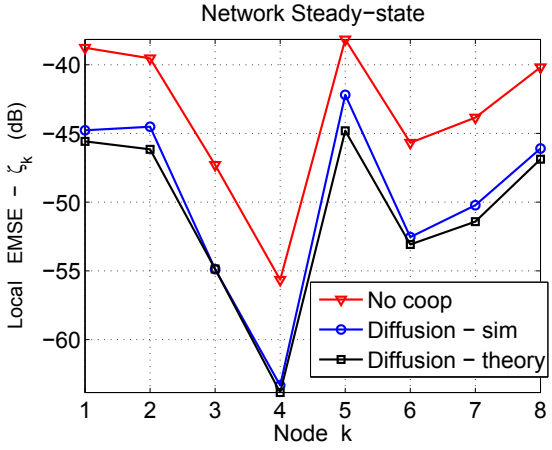

Fig. 8. Local EMSE (per node).

analytical models match well the simulations, especially for diminishing step-sizes.

Figures 7 and 8 present the network performance at the individual nodes. Note how the individual nodes perform better when driven by the cooperative protocol presented in this work. The match between theory and simulations is also quite good.

The analysis provided here can be extended to treat the transient performance as well as more general settings, such as adaptive networks operating under more sophisticated learning rules, observing non-Gaussian data.

5. REFERENCES

[1] C. G. Lopes and A. H. Sayed, "Diffusion least mean-squares over adaptive networks," *Proc. ICASSP*, vol.3, pp. 917–920, Honolulu, Hawaii, April 2007.

[2] A. H. Sayed and C. G. Lopes "Adaptive processing over distributed networks," to appear in *IEICE Transactions,* 2007.

[3] C. Lopes and A. H. Sayed, "Distributed adaptive incremental strategies: formulation and performance analysis," *Proc. ICASSP*, vol.3, pp. 584–587, Toulouse, France, May 2006.

[4] C. G. Lopes and A. H. Sayed, "Incremental adaptive strategies over distributed networks," *IEEE Transactions on Signal Processing*, to appear.

[5] R. H. Koning, H. Neudecker, T. Wansbeek " Block Kronecker products and the VECB operator," *Research Memorandum* nr 351, Institute of Economics Research, Dept. of Economics, University of Groningen, January, 1990.

[6] A. H. Sayed and C. G. Lopes, "Distributed recursive least-squares strategies over adaptive networks," *Proc. 40th Asilomar Conference on Signals, Systems and Computers*, pp. 233–237, Pacific Grove, CA, October 2006.

[7] D. Estrin, G. Pottie and M. Srivastava. "Intrumenting the world with wireless sensor networks," *Proc. IEEE ICASSP*, pp. 2033-2036, Salt Lake City, UT, May 2001.

[8] A. H. Sayed, *Fundamentals of Adaptive Filtering*, Wiley, NY, 2003.