

# Performance analysis of diffusion LMS in multitask networks

Jie Chen, Cédric Richard

Laboratoire Lagrange, Université de Nice Sophia-Antipolis, France  
 {jie.chen, cedric.richard}@unice.fr

**Abstract**—Diffusion LMS algorithm has been extensively studied during the last few years. This efficient approach allows to address distributed optimization problems over sensor networks in the case where the nodes have to collaboratively estimate a single parameter vector. Nevertheless, real-life problems are often multitask-oriented in the sense that the optimum parameter vector may not be the same for every node. In this paper, we conduct a theoretical analysis on the stochastic behavior of diffusion LMS when applied to multitask problems, that is, in a situation where the founding hypothesis of this algorithm is violated. Simulation results validate our theoretical model. Surprisingly, when applied to multitask problems in the presence of poor signal-to-noise ratio, we observe that diffusion LMS may exhibit better performance than non-cooperative LMS. This work provides a theoretical justification for the need to derive new cooperative algorithms specifically dedicated to multitask problems.

## I. INTRODUCTION

Distributed adaptive estimation has become an attractive and challenging problem with the rapid expansion of sensor networks. The interconnected nodes are supposed to conduct parameter estimation from collected observations in a collaborative and autonomous manner, with consideration of robustness and resource limitation. Research works have mainly focused on scenarios where the entire network is required to collectively estimate a same parameter vector. Two distributed strategies, the incremental strategy [1] and the diffusion strategy [2], [3] have been the subject of most discussions in recent years. Clearly, research efforts are now directed toward the latter because the former needs a cyclic covering of all the nodes, which is a severe constraint and a potential cause of failure. The diffusion strategy only requires that each node shares information with its neighbors. The use of local information makes such algorithms robust and adaptive to dynamic variations of the environment. Diffusion-based algorithms have been widely studied, including the pioneering work [2], diffusion LMS [3], diffusion RLS [4], sparsity-inducing diffusion [5], [6], performance analysis with imperfect information exchange and non-stationarity [7]. See [8] for an overview. Several interesting works have also explored distributed estimation of nonlinear input-output relationships defined in a reproducing kernel Hilbert space, e.g., for distributed field estimation problems [9]–[13].

An overview of the existing algorithms clearly shows that works have mostly focused on the case where all the sensor nodes have to collaboratively estimate a single parameter vector for the whole network. In the following, we shall refer to them as single-task problems. Unfortunately, real-life problems are often multitask-oriented in the sense that the optimum parameter vector may not be the same for every node. Diffusion LMS algorithm is a single-task algorithm. Running diffusion LMS to solve a multitask problem will lead to a biased result, which may be unacceptable. However, cooperative strategy may lead to better performance than uncooperative strategies provided that the optimums are close. The aim of this paper is to conduct an analysis on the stochastic behavior of diffusion LMS when applied to multitask problems, that is, in a situation where the founding hypothesis of diffusion LMS is violated.

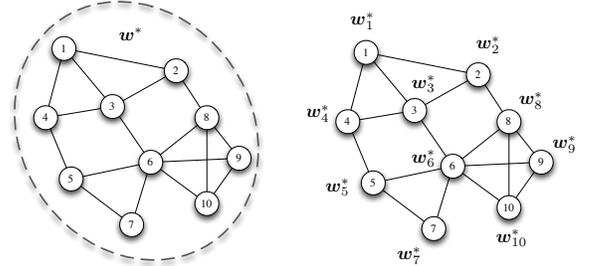


Fig. 1. Single-task network (left) and multitask network (right)

## II. DIFFUSION LMS AND PARETO SOLUTION

Consider a network composed of  $N$  nodes with some interconnection relations. Each node  $k$  has access to temporal measurement sequences  $\{d_k(n), \mathbf{x}_k(n)\}$  dominated by the linear model

$$d_k(n) = \mathbf{x}_k^\top(n) \mathbf{w}_k^* + z_k(n). \quad (1)$$

where the  $\mathbf{x}_k(n)$  are  $(L \times 1)$  regressors with covariance matrix  $\mathbf{R}_{x,k}$ ,  $\mathbf{w}_k^*$  are  $(L \times 1)$  weight vectors, and  $z_k(n)$  is an additive noise at node  $k$  and time instant  $n$ . Let  $J_k(\mathbf{w})$  be the mean-square-error cost function associated with node  $k$ , that is,

$$J_k(\mathbf{w}) = E\{|d_k(n) - \mathbf{x}_k^\top(n) \mathbf{w}\|^2\} \quad (2)$$

The global cost function for the network can be written as

$$J^{\text{glob}}(\mathbf{w}) = \sum_{k=1}^N J_k(\mathbf{w}) \quad (3)$$

For single-task networks, we consider that  $\mathbf{w}_k^* = \mathbf{w}_\ell^*$  for all  $(k, \ell)$ , as shown in the left of Figure 1. The adapt-then-combine (ATC) diffusion LMS algorithm has been proposed to solve problem (3) in an adaptive and distributed way [8]. It is characterized by the following two stages at each instant  $n$ , for each sensor  $k$ ,

$$\begin{aligned} \psi_k(n) &= \mathbf{w}_k(n) + \mu_k \sum_{\ell \in \mathcal{N}_k} c_{\ell k} \mathbf{x}_\ell(n) \left( d_\ell(n) - \mathbf{x}_k^\top(n) \mathbf{w}_k(n) \right) \\ \mathbf{w}_k(n+1) &= \sum_{\ell \in \mathcal{N}_k} a_{\ell k} \psi_\ell(n) \end{aligned} \quad (4)$$

with  $\mathcal{N}_k$  the neighbors of node  $k$ , including  $k$  itself. In the above expressions, non-negative coefficients  $a_{\ell k}$  and  $c_{\ell k}$  are the  $(\ell, k)$ -th entries of matrices  $\mathbf{A}$  and  $\mathbf{C}$ , respectively. They are required to satisfy

$$\begin{aligned} \mathbf{A}^\top \mathbf{1}_N &= \mathbf{1}_N, \quad \mathbf{C} \mathbf{1}_N = \mathbf{1}_N \\ a_{\ell k} &= 0, \quad c_{\ell k} = 0 \text{ if } \ell \notin \mathcal{N}_k \end{aligned} \quad (5)$$

where  $\mathbf{1}_N$  is an  $(N \times 1)$  all-one vector. For multitask problems, we can have  $\mathbf{w}_k^* \neq \mathbf{w}_\ell^*$  as illustrated in the right of Figure 1. As each cost function  $J_k(\mathbf{w})$  may not have the same optimum vector  $\mathbf{w}_k^*$ , the solution of problem (3) is a Pareto optimum for [14]

$$(\mathbf{J}_1(\mathbf{w}), \dots, \mathbf{J}_N(\mathbf{w})) \in \mathbb{R}^N \quad (6)$$

When applying the diffusion LMS (4) to the global cost (3) with sufficiently small step sizes, the network converges to the associated Pareto optimum with a bias at the level  $\mathcal{O}(\max(\mu_k)^2)$  [15].

In this paper, rather than studying the problem of seeking a Pareto solution that is optimal in some sense for the entire network, we shall analytically study how the algorithm (4) behaves with respect to the optimum parameter vectors  $\mathbf{w}_k^*$ .

### III. PERFORMANCE ANALYSIS OF ATC DIFFUSION LMS FOR MULTITASK NETWORKS

In order to analyze the stochastic behavior of the diffusion strategy (4), we collect information into block vectors and matrices. In particular, we collect the weight vectors at instant  $n$  and the optimum weight vector  $\mathbf{w}_k^*$  into  $(LN \times 1)$  vectors defined as

$$\mathbf{w}(n) = (\mathbf{w}_1^\top(n), \dots, \mathbf{w}_N^\top(n))^\top \quad (7)$$

$$\mathbf{w}^* = (\mathbf{w}_1^{*\top}, \dots, \mathbf{w}_N^{*\top})^\top \quad (8)$$

Let us define the weight-error vector as follows

$$\mathbf{v}_k(n) = \mathbf{w}_k(n) - \mathbf{w}_k^*. \quad (9)$$

We also stack the weight-error vectors into an  $(LN \times 1)$  vector

$$\mathbf{v}(n) = (\mathbf{v}_1^\top(n), \dots, \mathbf{v}_N^\top(n))^\top \quad (10)$$

To derive the theoretical analysis, we need to introduce the following independence assumption.

**Independence assumption** The inputs  $\mathbf{x}_k(n)$  are temporally stationary, white and spatially independent with  $\mathbf{R}_{x,k} = E\{\mathbf{x}_k(n)\mathbf{x}_k^\top(n)\}$ .

It directly follows from this assumption that  $\mathbf{x}_k(n)$  is independent of  $\mathbf{v}_\ell(m)$  for all  $\ell, k$ , and  $m \leq n$ . Although not true in general, this assumption is commonly used for adaptive filter analysis since it helps simplify the derivations. Usually, the results are not sensitive to this approximation.

#### A. Mean weight behavior analysis

With the definition of  $\mathbf{v}_k(n)$ , the estimation error that appears in the first step of (4) can be rewritten as follows

$$\begin{aligned} d_\ell(n) - \mathbf{x}_\ell^\top(n)\mathbf{w}_k(n) &= \mathbf{x}_\ell^\top(n)\mathbf{w}_\ell^* + z_\ell(n) - \mathbf{x}_\ell^\top(n)\mathbf{w}_k(n) \\ &= z_\ell(n) - \mathbf{x}_\ell^\top(n)(\mathbf{w}_k(n) - \mathbf{w}_\ell^*) \end{aligned} \quad (11)$$

Note that the term  $\mathbf{w}_k(n) - \mathbf{w}_\ell^*$  appears in the above expression. In usual analysis of diffusion LMS for single-task networks, this term is simply  $\mathbf{v}_k(n)$  since  $\mathbf{w}_k^* = \mathbf{w}_\ell^*$ . However, in this multitask context, we have

$$\begin{aligned} \mathbf{w}_k(n) - \mathbf{w}_\ell^* &= (\mathbf{w}_k(n) - \mathbf{w}_k^*) + (\mathbf{w}_k^* - \mathbf{w}_\ell^*) \\ &= \mathbf{v}_k(n) + \mathbf{u}_{k\ell}^* \end{aligned} \quad (12)$$

where  $\mathbf{u}_{k\ell}^*$  is the difference between the optimum weight vectors  $\mathbf{w}_k^*$  and  $\mathbf{w}_\ell^*$ . This relation enables us to express the error relation (11) as

$$d_\ell(n) - \mathbf{x}_\ell^\top(n)\mathbf{w}_k(n) = z_\ell(n) - \mathbf{x}_\ell^\top(n)\mathbf{v}_k(n) - \mathbf{x}_\ell^\top(n)\mathbf{u}_{k\ell}^* \quad (13)$$

Subtracting  $\mathbf{w}_k^*$  from both sides of the first step in (4), we have

$$\begin{aligned} \boldsymbol{\psi}_k(n) - \mathbf{w}_k^* &= \mathbf{v}_k(n) - \mu_k \sum_{\ell \in \mathcal{N}_k} c_{\ell k} \mathbf{x}_\ell(n) \mathbf{x}_\ell^\top(n) \mathbf{v}_k(n) \\ &\quad + \mu_k \sum_{\ell \in \mathcal{N}_k} c_{\ell k} \mathbf{x}_\ell(n) z_\ell(n) - \mu_k \sum_{\ell \in \mathcal{N}_k} c_{\ell k} \mathbf{x}_\ell(n) \mathbf{x}_\ell^\top(n) \mathbf{u}_{k\ell}^* \end{aligned} \quad (14)$$

Let us define the following  $(N \times N)$  block diagonal matrices with blocks of size  $(L \times L)$

$$\mathbf{U} = \text{diag}\{\mu_1 \mathbf{I}_L, \dots, \mu_N \mathbf{I}_L\} \quad (15)$$

$$\mathbf{H}(n) = \text{diag}\left\{ \sum_{\ell \in \mathcal{N}_1} c_{\ell 1} \mathbf{x}_\ell(n) \mathbf{x}_\ell^\top(n), \dots, \sum_{\ell \in \mathcal{N}_N} c_{\ell N} \mathbf{x}_\ell(n) \mathbf{x}_\ell^\top(n) \right\} \quad (16)$$

$$\begin{aligned} \mathbf{h}_u(n) &= \left( \sum_{\ell \in \mathcal{N}_1} c_{\ell 1} \mathbf{x}_\ell(n) \mathbf{x}_\ell^\top(n) \mathbf{u}_{k\ell}^*, \dots, \sum_{\ell \in \mathcal{N}_N} c_{\ell N} \mathbf{x}_\ell(n) \mathbf{x}_\ell^\top(n) \mathbf{u}_{k\ell}^* \right)^\top \end{aligned} \quad (17)$$

Let us also define the  $(LN \times 1)$  vector  $\mathbf{p}_{zx}(n)$

$$\mathbf{p}_{zx}(n) = \left( \sum_{\ell \in \mathcal{N}_1} c_{\ell 1} \mathbf{x}_\ell^\top(n) z_\ell(n), \dots, \sum_{\ell \in \mathcal{N}_N} c_{\ell N} \mathbf{x}_\ell^\top(n) z_\ell(n) \right)^\top. \quad (18)$$

With these matrices and vectors, the update relation for the stacked weight error vector of  $\boldsymbol{\psi}(n)$  can be expressed as

$$\boldsymbol{\psi}(n) - \mathbf{w}^* = \mathbf{v}(n) - \mathbf{U}\mathbf{H}(n)\mathbf{v}(n) + \mathbf{U}\mathbf{p}_{zx}(n) - \mathbf{U}\mathbf{h}_u(n) \quad (19)$$

Let the matrix  $\mathbf{A}_I = \mathbf{A} \otimes \mathbf{I}_L$ . The combination step in (4) can be written as follows

$$\mathbf{w}(n+1) = \mathbf{A}_I^\top \boldsymbol{\psi}(n) \quad (20)$$

Subtracting  $\mathbf{w}^*$  from both sides of the above expression, the weight error relation is given by

$$\mathbf{v}(n+1) = \mathbf{A}_I^\top (\boldsymbol{\psi}(n) - \mathbf{w}^*) + (\mathbf{A}_I^\top - \mathbf{I}_{NL}) \mathbf{w}^* \quad (21)$$

Combining (19) and (21), the update relation for  $\mathbf{v}(n)$  can be written in a single expression

$$\begin{aligned} \mathbf{v}(n+1) &= \mathbf{A}_I^\top (\mathbf{I}_{NL} - \mathbf{U}\mathbf{H}(n)) \mathbf{v}(n) + \mathbf{A}_I^\top \mathbf{U}\mathbf{p}_{zx}(n) \\ &\quad - \mathbf{A}_I^\top \mathbf{U}\mathbf{h}_u(n) + (\mathbf{A}_I^\top - \mathbf{I}_{NL}) \mathbf{w}^*. \end{aligned} \quad (22)$$

Taking the expectation of (22), the independence assumption yields

$$\begin{aligned} E\{\mathbf{v}(n+1)\} &= \mathbf{A}_I^\top (\mathbf{I}_{NL} - \mathbf{U}\mathbf{H}) E\{\mathbf{v}(n)\} \\ &\quad - \mathbf{A}_I^\top \mathbf{U}\mathbf{h}_u + (\mathbf{A}_I^\top - \mathbf{I}_{NL}) \mathbf{w}^* \end{aligned} \quad (23)$$

where  $\mathbf{H}$  is the expected value of  $\mathbf{H}(n)$  given by

$$\mathbf{H} = \text{diag}\{\mathbf{R}_1, \dots, \mathbf{R}_N\}, \quad (24)$$

with  $\mathbf{R}_k = \sum_{\ell \in \mathcal{N}_k} c_{\ell k} \mathbf{R}_{x,\ell}$  for simplicity. The matrix  $\mathbf{h}_u$  is the expected value of  $E\{\mathbf{h}_u(n)\}$

$$\mathbf{h}_u = \left( \sum_{\ell \in \mathcal{N}_1} c_{\ell 1} \mathbf{R}_{x,\ell} \mathbf{u}_{1\ell}^*, \dots, \sum_{\ell \in \mathcal{N}_N} c_{\ell N} \mathbf{R}_{x,\ell} \mathbf{u}_{N\ell}^* \right)^\top \quad (25)$$

*Theorem 1:* (Convergence in the mean) Assume data model (1) and Assumption 1 hold. Then for any initial condition, the ATC diffusion LMS applied to multitask networks asymptotically converges in the mean if the step size is chosen to satisfy

$$0 < \mu_k < \frac{2}{\lambda_{\max}(\mathbf{R}_k)} \quad (26)$$

where  $\lambda_{\max}(\cdot)$  denotes the maximum eigenvalue of the matrix. The estimation bias in the steady state tends to<sup>1</sup>

$$\begin{aligned} E\{\mathbf{v}(\infty)\} &= \left( \mathbf{I}_{NL} - \mathbf{A}_I^\top (\mathbf{I}_{NL} - \mathbf{U}\mathbf{H}) \right)^{-1} \\ &\quad \times \left( -\mathbf{A}_I^\top \mathbf{U}\mathbf{h}_u + (\mathbf{A}_I^\top - \mathbf{I}_{NL}) \mathbf{w}^* \right). \end{aligned} \quad (27)$$

<sup>1</sup>Due to space limitations, proofs are omitted in this article.

### B. Second-order behavior analysis

For clearance, we denote several terms in the weight error expression (22) as follows

$$\mathbf{B}(n) = \mathbf{A}_I^\top (\mathbf{I}_{NL} - \mathbf{U}\mathbf{H}(n)) \quad (28)$$

$$\mathbf{g}(n) = \mathbf{A}_I^\top \mathbf{U}\mathbf{p}_{zx}(n) \quad (29)$$

$$\mathbf{r}(n) = \mathbf{A}_I^\top \mathbf{U}\mathbf{h}_u(n) - (\mathbf{A}_I^\top - \mathbf{I}_{NL})\mathbf{w}^* \quad (30)$$

Following the independence assumption, the mean-square of the weight vector  $\mathbf{v}(n+1)$  weighted by the metric  $\Sigma$  satisfies

$$\begin{aligned} E\{\|\mathbf{v}(n+1)\|_{\Sigma}^2\} &= E\{\|\mathbf{v}(n)\|_{\Sigma'}^2\} + \text{trace}\left\{\Sigma E\{\mathbf{g}(n)\mathbf{g}^\top(n)\}\right\} \\ &+ E\{\|\mathbf{r}(n)\|_{\Sigma}^2\} - E\{\mathbf{r}^\top(n)\Sigma\mathbf{A}_I^\top (\mathbf{I}_{NL} - \mathbf{U}\mathbf{H}(n))\mathbf{v}(n)\} \end{aligned} \quad (31)$$

with  $\Sigma$  any nonnegative-definite matrix that allows to derive several performance metrics, and  $\Sigma' = E\{\mathbf{B}^\top(n)\Sigma\mathbf{B}(n)\}$ . Let  $\mathbf{G}$  denote the expected value in the second term on the right hand side of (31). It is expressed as

$$\begin{aligned} \mathbf{G} &= E\{\mathbf{g}(n)\mathbf{g}^\top(n)\} \\ &= \mathbf{A}_I^\top \mathbf{U}\mathbf{C}_I^\top \text{diag}\{\sigma_{z,1}^2 \mathbf{R}_{x,1}, \dots, \sigma_{z,N}^2 \mathbf{R}_{x,N}\} \mathbf{C}_I \mathbf{U} \mathbf{A}_I \end{aligned} \quad (32)$$

with the matrix  $\mathbf{C}_I = \mathbf{C} \otimes \mathbf{I}_L$ . For clearance, we also define

$$\begin{aligned} f(\mathbf{r}(n), \Sigma, \mathbf{v}(n)) &= \|\mathbf{r}(n)\|_{\Sigma}^2 - \mathbf{r}^\top(n)\Sigma\mathbf{A}_I^\top (\mathbf{I}_{NL} - \mathbf{U}\mathbf{H}(n))\mathbf{v}(n) \end{aligned} \quad (33)$$

The relation (31) is then written by

$$\begin{aligned} E\{\|\mathbf{v}(n+1)\|_{\Sigma}^2\} &= E\{\|\mathbf{v}(n)\|_{\Sigma'}^2\} + \text{trace}\{\Sigma\mathbf{G}\} + E\{f(\mathbf{r}(n), \Sigma, \mathbf{v}(n))\} \end{aligned} \quad (34)$$

Let  $\text{vec}(\cdot)$  denote the operator that stacks the columns of a matrix on top of each other. Vectorizing the matrix  $\Sigma$  and  $\Sigma'$  by  $\sigma = \text{vec}(\Sigma)$  and  $\sigma' = \text{vec}(\Sigma')$ , the relation between them can be expressed by the linear transformation

$$\sigma' = \mathbf{K}\sigma \quad (35)$$

with the  $(LN)^2 \times (LN)^2$  matrix

$$\begin{aligned} \mathbf{K} &= \mathbf{A}_I \otimes \mathbf{A}_I - \mathbf{H}\mathbf{U}\mathbf{A}_I \otimes \mathbf{A}_I - \mathbf{A}_I \otimes \mathbf{H}\mathbf{U}\mathbf{A}_I \\ &+ E\{\mathbf{H}(n)^\top \mathbf{U}\mathbf{A}_I \otimes \mathbf{H}(n)^\top \mathbf{U}\mathbf{A}_I\} \end{aligned} \quad (36)$$

If step sizes are sufficiently small,  $\mathbf{K}$  can be approximated by

$$\mathbf{K} \approx \mathbf{B}^\top \otimes \mathbf{B}^\top \quad (37)$$

with  $\mathbf{B} = E\{\mathbf{B}(n)\}$ , and

$$\begin{aligned} E\{f(\mathbf{r}(n), \Sigma, \mathbf{v}(n))\} &\approx \|\mathbf{r}\|_{\Sigma}^2 - \mathbf{r}^\top \Sigma \mathbf{A}_I^\top (\mathbf{I}_{NL} - \mathbf{U}\mathbf{H}) E\{\mathbf{v}(n)\} \\ &= f(\mathbf{r}, \Sigma, E\{\mathbf{v}(n)\}) \end{aligned} \quad (38)$$

with  $\mathbf{r} = E\{\mathbf{r}(n)\}$ . The relation (34) can then be expressed as follows

$$\begin{aligned} E\{\|\mathbf{v}(n+1)\|_{\sigma}^2\} &= E\{\|\mathbf{v}(n)\|_{\mathbf{K}\sigma}^2\} + \text{vec}(\mathbf{G}^\top)^\top \sigma \\ &+ f(\mathbf{r}, \Sigma, E\{\mathbf{v}(n)\}) \end{aligned} \quad (39)$$

where we consider the notations  $\|\cdot\|_{\Sigma}$  and  $\|\cdot\|_{\sigma}$  are interchangeable.

**Theorem 2:** (Mean-square stability) Assume data model (1) and Assumption 1 hold. If the step sizes  $\{\mu_k\}$  are sufficiently small such that (26) is satisfied, the diffusion LMS algorithm applied in multitask networks is mean-square stable if the matrix  $\mathbf{K}$  is stable. Moreover, if terms that depend on second-order power of step sizes are ignored, the stability of  $\mathbf{K}$  depends on that of  $E\{\mathbf{B}(n)\}$  guaranteed by (26).

**Theorem 3:** (Transient MSD) Considering small step sizes that satisfy *Theorem 1* and *Theorem 2*, and the metric  $\sigma_I = \text{vec}(\mathbf{I}_{NL})$ , the MSD learning curve of the diffusion LMS applied over multitask networks is described by the following recursions over  $n \geq 0$

$$\begin{aligned} \zeta(n+1) &= \zeta(n) + \frac{1}{N} \left( \text{vec}(\mathbf{G}^\top)^\top \mathbf{K}^n \sigma_I \right. \\ &\quad \left. - \|\mathbf{v}(0)\|_{(\mathbf{I}_{(NL)^2} - \mathbf{K})\mathbf{K}^n \sigma_I}^2 + \|\mathbf{r}\|_{\mathbf{K}^n \sigma_I}^2 \right) \\ &\quad - 2(\Upsilon(n) + (\mathbf{B} E\{\mathbf{v}(n)\})^\top \otimes \mathbf{r}^\top) \sigma_I \\ \Upsilon(n+1) &= \Upsilon(n) \mathbf{K} + ((\mathbf{B} E\{\mathbf{v}(n)\})^\top \otimes \mathbf{r}^\top) (\mathbf{K} - \mathbf{I}_{(NL)^2}) \end{aligned} \quad (40)$$

with the initial condition  $\zeta(0) = \frac{1}{N} \|\mathbf{v}(0)\|^2$  and  $\Upsilon(0) = \mathbf{0}_{(NL)^2}$ .

**Theorem 4:** (Steady state MSD) Considering sufficiently small step sizes that satisfy *Theorem 1* and *Theorem 2*, the steady-state MSD for the diffusion LMS algorithm applied over multitask networks is determined by the following expression

$$\begin{aligned} \zeta^* &= \frac{1}{N} \text{vec}(\mathbf{G}^\top)^\top (\mathbf{I}_{(NL)^2} - \mathbf{K})^{-1} \text{vec}(\mathbf{I}_{NL}) \\ &\quad + f(\mathbf{r}, \frac{1}{N} (\mathbf{I}_{(NL)^2} - \mathbf{K})^{-1} \text{vec}(\mathbf{I}_{NL}), E\{\mathbf{v}(\infty)\}) \end{aligned} \quad (42)$$

with  $E\{\mathbf{v}(\infty)\}$  determined by (27).

It can be seen that whether diffusion LMS is able to outperform non-cooperative LMS in a network depends on the sign of the term  $f(\mathbf{r}, \frac{1}{N} (\mathbf{I}_{(NL)^2} - \mathbf{K})^{-1} \text{vec}(\mathbf{I}_{NL}), E\{\mathbf{v}(\infty)\})$ .

## IV. SIMULATIONS

In this section we provide simulations to validate the derived analytical results and discuss the behavior of ATC diffusion LMS in multitask networks. We considered a network consisting of 10 nodes with connections shown in Figure 2 (left). The coefficient vectors were 2-dimensional with  $\mathbf{w}_k^* = \mathbf{w}_o + \Delta \mathbf{w}_k$  and  $\mathbf{w}_o = [0.5, -0.4]^\top$ . The regression inputs  $\mathbf{x}_k(n)$  were zero-mean ( $L \times 1$ ) Gaussian random vectors with covariance matrices  $\mathbf{R}_{x,k} = \sigma_{x,k}^2 \mathbf{I}_L$ , with  $\sigma_{x,k}^2$  shown in the top right of Figure 2. Noises  $z_k(n)$  were i.i.d. zero-mean Gaussian random variables that were independent of any other signals. We considered the diffusion algorithm (4) with the measurement exchange using a uniform matrix  $\mathbf{C}$  such that  $c_{\ell k} = |\mathcal{N}_\ell|^{-1}$ , and a combination matrix  $\mathbf{A}$  simply averaging the estimates from neighbors, i.e.,  $a_{\ell k} = |\mathcal{N}_k|^{-1}$ .

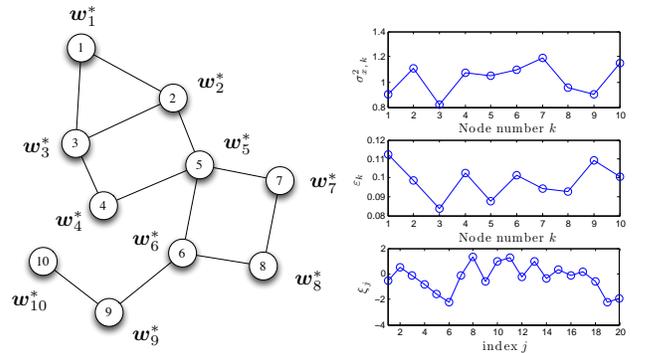


Fig. 2. Network topology and settings. Left: topology. Right: input variance  $\sigma_{x,k}^2$  (top), realization of  $\epsilon$  (middle), realization of  $\xi$  (bottom).

Let  $\epsilon_j$  be a uniform random variable drawn from  $\mathcal{U}(0.18, 0.22)$ , and  $\xi_k$  be a value drawn from the normal distribution  $\mathcal{N}(0, 1)$ . The realizations of  $\epsilon$  and  $\xi$  used in simulations are shown in the middle

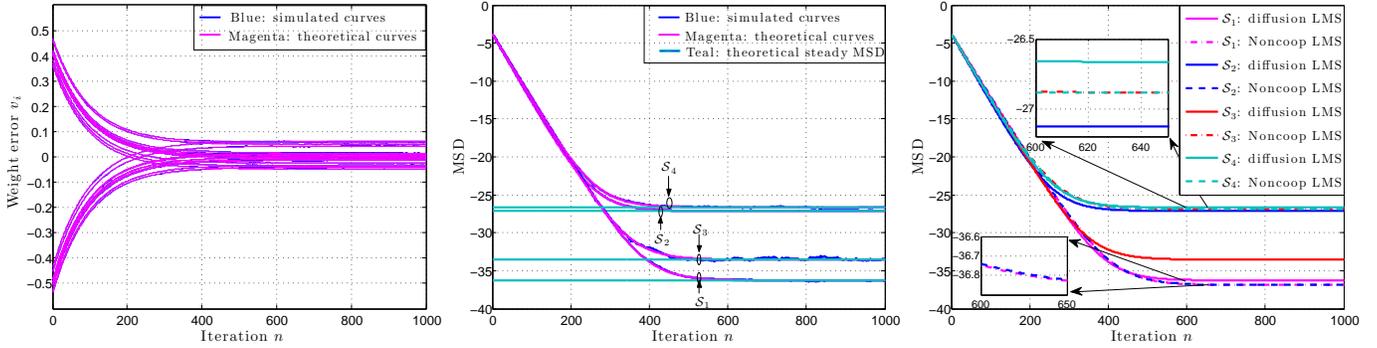


Fig. 3. Network behavior illustration. Left: weight error behavior of  $S_4$ . Middle: Simulated and theoretical MSD. Right: Comparison with non-cooperative LMS.

and bottom of Figure 2 (right). The following network configurations were considered

- $S_1 : \{\sigma_{z,k}^2 = 0.1\varepsilon_k, \Delta w_j = 0.01\xi_j\}$
- $S_2 : \{\sigma_{z,k}^2 = 0.1\varepsilon_k, \Delta w_j = 0.03\xi_j\}$
- $S_3 : \{\sigma_{z,k}^2 = \varepsilon_k, \Delta w_j = 0.01\xi_j\}$
- $S_4 : \{\sigma_{z,k}^2 = \varepsilon_k, \Delta w_j = 0.03\xi_j\}$

The ATC diffusion LMS (4) was conducted over this network with a unique step size  $\mu_k = 0.01$  for all node  $k$ . Simulated results were obtained by averaging over 100 Monte-Carlo runs. Theoretical evolution curves of weight errors were obtained by (23). The left of Figure 3 shows the weight error behavior of the algorithm with the configuration  $S_4$ . It can be observed that  $v_k(n)$  did not converge to zero, i.e., the weights were clearly biased, due to significant difference between optimum  $w_k^*$ . The model (23) accurately matched the simulated curves. The middle of Figure 3 shows the transient MSD behavior and steady-state MSD for all these four configurations. The theoretical learning curves were obtained by (40) and (41). The theoretical steady state MSD was calculated using (42). Again, the derived models accurately predicted the MSD behavior of the algorithm in such a network.

After validating the models, we compared the performance between ATC diffusion LMS and non-cooperative LMS. As theoretical models have been proved accurate enough, we only used theoretical curves in this comparison for clearance of the illustration, shown in the right of Figure 3. Firstly, note that for the case  $S_2$ , diffusion LMS has significantly lower performance compared with non-cooperative LMS, as optimums of nodes were significantly different and noise variances were minor. It can be considered in this case that diffusion LMS was improperly used and estimation bias dominated the error. Situation  $S_3$  is on the contrary to that of  $S_2$ , as with this setting the effect of noise dominated the effect of bias. Nodes benefitted from cooperation of diffusion LMS that provided better estimation by smoothing away noise, even though the estimate was biased. In cases  $S_1$  and  $S_4$ , the effect of noise rivaled that of bias so that diffusion LMS and non-cooperative LMS had similar performance. Moreover, the performance of non-cooperative LMS with the same noise level was almost the same with different optimums ( $S_1$  vs.  $S_2$ ,  $S_3$  vs.  $S_4$ ), as its estimates were unbiased.

## V. CONCLUSION AND PERSPECTIVES

In this paper we examined the performance of ATC diffusion LMS applied to multitask networks, where its design assumption is

violated. Accurate theoretical analysis of algorithm weight behavior and MSD behavior was provided. Further work may include studies of connection or re-weighting strategies to make the cooperation more adaptive to multitask problems.

## REFERENCES

- [1] C. G. Lopes and A. H. Sayed, "Incremental adaptive strategies over distributed networks," *IEEE Trans. Signal Process.*, vol. 55, no. 8, pp. 4064–4077, Aug. 2007.
- [2] C. G. Lopes and A. H. Sayed, "Diffusion least-mean squares over adaptive networks: Formulation and performance analysis," *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 3122–3136, Jul. 2008.
- [3] F. Cattivelli and A. H. Sayed, "Diffusion LMS strategies for distributed estimation," *IEEE Trans. Signal Process.*, vol. 50, no. 3, pp. 1035–1048, Mar. 2010.
- [4] A. Bertrand, M. Moonen, and A. H. Sayed, "Diffusion bias-compensated RLS estimation over adaptive networks," *IEEE Trans. Signal Process.*, vol. 59, no. 11, pp. 5212–5224, Nov. 2011.
- [5] P. Di Lorenzo and A. H. Sayed, "Sparse distributed learning based on diffusion adaptation," *IEEE Trans. Signal Process.*, vol. 61, no. 6, pp. 1419–1433, Mar. 2013.
- [6] P. Chainais and C. Richard, "Distributed dictionary learning over a sensor network," in *Proc. of Cap2013*, Lille, France, Jul. 2013.
- [7] X. Zhao, S.-Y. Tu, and A. H. Sayed, "Diffusion adaptation over networks under imperfect information exchange and non-stationary data," *IEEE Trans. Signal Process.*, vol. 60, no. 7, pp. 3460–3475, Jul. 2012.
- [8] A. H. Sayed, "Diffusion adaptation over networks," *E-Reference Signal Processing*, 2013.
- [9] J. Predd, S. Kulkarni, and H. Vincent Poor, "Distributed learning in wireless sensor networks," *IEEE Signal Process. Mag.*, vol. 23, no. 4, pp. 59–69, Jul. 2006.
- [10] P. Honeine, C. Richard, J.-C. M. Bermudez, and H. Snoussi, "Distributed prediction of time series data with kernels and adaptive filtering techniques in sensor networks," in *Proc. of ASILOMAR*, 2008, pp. 246–250.
- [11] P. Honeine, C. Richard, J.-C. M. Bermudez, H. Snoussi, M. Essoloh, and F. Vincent, "Functional estimation in hilbert space for distributed learning in wireless sensor networks," in *Proc. of ICASSP*, 2009, pp. 2861–2864.
- [12] P. Honeine, C. Richard, H. Snoussi, J. C. M. Bermudez, and J. Chen, "A decentralized approach for non-linear prediction of time series data in sensor networks," *EURASIP JWCN*, vol. 2010, no. 1, pp. 6247372, April 2010.
- [13] J. Chen, C. Richard, P. Honeine, and J.-C. M. Bermudez, "Non-negative distributed regression for data inference in wireless sensor networks," in *Proc. of ASILOMAR*, Pacific Grove (CA), USA, Nov. 2010.
- [14] S. P. Boyd and L. Vandenberghe, *Convex optimization*, Cambridge university press, 2004.
- [15] J. Chen and A. H. Sayed, "Distributed pareto optimization via diffusion strategies," *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 2, pp. 205–220, Aug. 2013.