Performance analysis of multitask diffusion adaptation over asynchronous networks

Roula Nassif[†], Cédric Richard[†], André Ferrari[†], Ali H. Sayed[‡]
[†] Université de Nice Sophia-Antipolis, CNRS, France
Email: {roula.nassif, cedric.richard, andre.ferrari}@oca.eu
[‡]University of California, Los Angeles, USA
Email: sayed@ee.ucla.edu

***Abstract*—The multitask diffusion LMS algorithm is an efficient strategy to address distributed estimation problems that are multitask-oriented in the sense that the optimum parameter vector may not be the same for every cluster of nodes. In this work, we explore the adaptation and learning behavior of the algorithm under asynchronous conditions when networks are subject to various sources of uncertainties, including random link failures and agents turning on and off randomly. We conduct a mean-square-error performance analysis and examine how asynchronous events interfere with the learning performance.**

I. Introduction

Distributed optimization enables the solution of inference problems in a decentralized manner over networks [1]. Depending on the number of parameter vectors to estimate, we distinguish between two types of networks: single-task networks and multitask networks. Several strategies have been proposed for single-task scenarios in the literature where the entire network is employed to collectively estimate a single parameter vector. Among these techniques, we mention consensus strategies, incremental strategies, and diffusion strategies (e.g., [2]–[5]). Diffusion strategies are particularly attractive due to their enhanced adaptation performance and stability.

In some application scenarios, however, there is need to employ distributed algorithms that can handle clustered models with multiple parameter vectors. In this work, we are therefore interested in distributed and collaborative estimation over clustered multitask networks where agents are grouped into clusters, and each cluster has to estimate its own parameter vector. Existing strategies for these cases tend to depend on how the tasks relate to each other and on the availability or not of prior information. For instance, the scenarios studied in [6], [7] do not assume any prior information. In particular, nodes do not know which other nodes share similar objectives. The scenarios described in [8]–[11], on the other hand, assume that the local parameter vectors share a common latent signal subspace. Another way to exploit and model relationships among tasks is to formulate local optimization problems with appropriate regularizers. The multitask diffusion LMS algorithm developed in [12] uses co-regularization between neighboring clusters to enhance estimation accuracy.

The work of C. Richard and A. Ferrari was supported in part by ANR and DGA grant ANR-13-ASTR-0030 (ODISSEE). The work of A. H. Sayed was supported in part by NSF grants CCF-1011918 and ECCS-1407712.

The strategy developed in [12] assumes that all agents act synchronously. Nevertheless, in many real-world applications, networks are subject to uncertainties, such as random agent and link failures, changing topologies, or agents turning on/off randomly for energy conservation. An extensive study on the performance of the diffusion strategies in the presence of asynchronous events or changing topologies has been developed in [13]–[15] for single-task adaptation. In this work, we extend the analysis to multitask scenarios involving mean-square-error designs.

Notation. We use normal font letters to denote scalars, boldface lowercase letters to denote column vectors and boldface uppercase letters to denote matrices. We use the symbol $\otimes$ to denote Kronecker operation and the symbol tr$(\cdot)$ to denote the trace operator.

II. Asynchronous Multitask Diffusion Adaptation

We consider a connected network consisting of N nodes grouped into Q clusters. At each time instant i, node k collects a scalar zero-mean measurement $d_k(i)$ and a zero-mean $L \times 1$ regression vector $\boldsymbol{x}_k(i)$ with positive-definite covariance matrix $\boldsymbol{R}_{x,k} = \mathbb{E}\{\boldsymbol{x}_k(i)\,\boldsymbol{x}_k^\top(i)\}$. We assume that the temporal measurement sequence $\{d_k(i), \boldsymbol{x}_k(i)\}$ is related to the unknown parameter vector $\boldsymbol{w}_k^\star$ via the linear regression model:

$$d_k(i) = \boldsymbol{x}_k^\top(i)\boldsymbol{w}_k^\star + z_k(i), \tag{1}$$

with $z_k(i)$ a zero-mean measurement noise of variance $\sigma_{z,k}^2$. The noise process is assumed to be temporally white and spatially independent of any other signal. The optimum parameter vectors are only constrained to be equal within each cluster, namely, $\boldsymbol{w}_k^\star = \boldsymbol{w}_{\mathcal{C}_j}^\star$ whenever node k belongs to cluster $\mathcal{C}_j$. However, if cluster $\mathcal{C}_p$ is connected to cluster $\mathcal{C}_q$, that is, there exists at least one edge connecting a node of $\mathcal{C}_p$ to a node of $\mathcal{C}_q$, then their optimum parameter vectors are allowed to jointly satisfy certain properties. In [12], smoothness of the graph signal $\mathcal{W}^\star = \{\boldsymbol{w}_1^\star, \dots, \boldsymbol{w}_N^\star\}$ is enforced by regularizing the estimation problem with the squared ℓ_2-norm of the graph gradient at each node k, namely, by using

$$\|\nabla_k \mathcal{W}\|^2 = \sum_{\ell \in \mathcal{N}_k} \rho_{k\ell} \|\boldsymbol{w}_k - \boldsymbol{w}_\ell\|^2 \tag{2}$$

where $\mathcal{N}_k$ denotes the neighborhood of node k, and $\rho_{k\ell}$ is the nonnegative weight assigned to the edge between nodes k

and ℓ. In a manner similar to [12], combining the mean-square error criterion and the regularizer (2) at each node to estimate the unknown parameter vector $\boldsymbol{w}^\star_{\mathcal{C}_j}$ at the level of each cluster leads to a Nash equilibrium problem [16] defined by the Q subproblems $(\mathcal{P}_j)$:

$$(\mathcal{P}_j): \begin{cases} \min\limits_{\boldsymbol{w}_{\mathcal{C}_j}} J_{\mathcal{C}_j}(\boldsymbol{w}_{\mathcal{C}_j}, \boldsymbol{w}_{-\mathcal{C}_j}) \\ \text{with } J_{\mathcal{C}_j}(\boldsymbol{w}_{\mathcal{C}_j}, \boldsymbol{w}_{-\mathcal{C}_j}) \\ \qquad = \sum\limits_{k\in\mathcal{C}_j} \mathbb{E}\{|d_k(i) - \boldsymbol{x}_k^\top(i)\, \boldsymbol{w}_{\mathcal{C}_j}|^2\} \\ \qquad + \eta \sum\limits_{k\in\mathcal{C}_j} \sum\limits_{\ell\in\mathcal{N}_k\setminus\mathcal{C}_j} \rho_{k\ell} \|\boldsymbol{w}_{\mathcal{C}_j} - \boldsymbol{w}_{\mathcal{C}(\ell)}\|^2 \end{cases} \tag{3}$$

where $\mathcal{C}(\ell)$ is the cluster to which node ℓ belongs, and η is the regularization strength. The notation $\boldsymbol{w}_{-\mathcal{C}_i}$ denotes the collection of weight vectors estimated by the other clusters, namely, $\boldsymbol{w}_{-\mathcal{C}_i} = \{\boldsymbol{w}_{\mathcal{C}_k} : k = 1, \ldots, Q\} - \{\boldsymbol{w}_{\mathcal{C}_i}\}$. Note in (3) that the regularizer excludes those neighbors of node k that belong to its cluster. This is because these particular neighbors will be pursuing the same vector as node k.

An adapt-then-combine diffusion algorithm is derived in [12] for solving (3). Following the same procedure as [14], which provides a general framework for single-task asynchronous networks, we introduce the following multitask diffusion LMS algorithm for asynchronous networks:

$$\begin{cases} \boldsymbol{\psi}_k(i+1) = \boldsymbol{w}_k(i) + \mu_k(i)\big[d_k(i) - \boldsymbol{x}_k^\top(i)\boldsymbol{w}_k(i)\big]\, \boldsymbol{x}_k(i) \\ \qquad\qquad + \eta\mu_k(i) \sum\limits_{\ell\in\mathcal{N}_k(i)\setminus\mathcal{C}(k)^-} \rho_{k\ell}(i)(\boldsymbol{w}_\ell(i) - \boldsymbol{w}_k(i)) \\ \boldsymbol{w}_k(i+1) = \sum\limits_{\ell\in\mathcal{N}_k(i)\cap\mathcal{C}(k)} a_{\ell k}(i)\, \boldsymbol{\psi}_\ell(i+1) \end{cases} \tag{4}$$

where $\boldsymbol{w}_k(i)$ is the estimate of $\boldsymbol{w}_k^\star$ at time i, $\boldsymbol{\psi}_k(i)$ is an intermediate estimate, and $\mathcal{C}(k)^-$ is the cluster to which node k belongs, excluding k. To model the asynchronous behavior of agent k at time i, we allow its step-size parameter to be a bounded *random* variable $\mu_k(i) \in [0, \mu_{\max,k}]$. Furthermore, we model uncertainties in the links within and among clusters by the use of nonnegative *random* combination coefficients $\{a_{\ell k}(i)\}$ and regularization factors $\{\rho_{k\ell}(i)\}$. The notation $\mathcal{N}_k(i)$ denotes the *random* neighborhood of agent k at time i.

At each time i, the random coefficients $a_{\ell k}(i)$ and $\rho_{k\ell}(i)$ are required to satisfy the following constraints:

$$\sum_{\ell\in\mathcal{N}_k(i)\cap\mathcal{C}(k)} a_{\ell k}(i) = 1, \text{ and } \begin{cases} a_{\ell k}(i) > 0, \text{ if } \ell \in \mathcal{N}_k(i)\cap\mathcal{C}(k), \\ a_{\ell k}(i) = 0, \text{ otherwise.} \end{cases} \tag{5}$$

$$\sum_{\ell\in\mathcal{N}_k(i)\setminus\mathcal{C}(k)^-} \rho_{k\ell}(i) = 1, \text{ and } \begin{cases} \rho_{k\ell}(i) > 0, \text{ if } \ell \in \mathcal{N}_k(i)\setminus\mathcal{C}(k), \\ \rho_{kk}(i) \geq 0, \\ \rho_{k\ell}(i) = 0, \text{ otherwise.} \end{cases} \tag{6}$$

Let $\boldsymbol{M}(i)$ be the diagonal matrix with entries $\mu_k(i)$, $\boldsymbol{A}(i)$ the left-stochastic matrix whose (ℓ, k)-th entry is $a_{\ell k}(i)$, and $\boldsymbol{P}(i)$ the right-stochastic matrix whose (k, ℓ)-th element is $\rho_{k\ell}(i)$. The random matrices $\boldsymbol{M}(i)$, $\boldsymbol{A}(i)$, and $\boldsymbol{P}(i)$ are assumed to be mutually independent, and independent of any other random variables. We further assume that $\{\boldsymbol{M}(i)\}$, $\{\boldsymbol{A}(i)\}$ and $\{\boldsymbol{P}(i)\}$ are weakly stationary processes with means $\overline{\boldsymbol{M}}$, $\overline{\boldsymbol{A}}$ and $\overline{\boldsymbol{P}}$, respectively. Let $\boldsymbol{C}_M$, $\boldsymbol{C}_A$ and $\boldsymbol{C}_P$ be their Kronecker covariance matrices defined as

$$\boldsymbol{C}_M \triangleq \mathbb{E}\{(\boldsymbol{M}(i) - \overline{\boldsymbol{M}}) \otimes (\boldsymbol{M}(i) - \overline{\boldsymbol{M}})\} \tag{7}$$
$$\boldsymbol{C}_A \triangleq \mathbb{E}\{(\boldsymbol{A}(i) - \overline{\boldsymbol{A}}) \otimes (\boldsymbol{A}(i) - \overline{\boldsymbol{A}})\} \tag{8}$$
$$\boldsymbol{C}_P \triangleq \mathbb{E}\{(\boldsymbol{P}(i) - \overline{\boldsymbol{P}}) \otimes (\boldsymbol{P}(i) - \overline{\boldsymbol{P}})\}. \tag{9}$$

III. Stochastic performance analysis

Let us denote by $\boldsymbol{w}(i)$ and $\boldsymbol{w}^\star$ the block weight estimate vector and the block optimum weight vector, namely,

$$\boldsymbol{w}(i) \triangleq \text{col}\{\boldsymbol{w}_1(i), \ldots, \boldsymbol{w}_N(i)\} \tag{10}$$
$$\boldsymbol{w}^\star \triangleq \text{col}\{\boldsymbol{w}_1^\star, \ldots, \boldsymbol{w}_N^\star\} \tag{11}$$

where $\text{col}\{\cdot\}$ stacks its column vector arguments on top of each other. Let us define the block weight error vector as:

$$\widetilde{\boldsymbol{w}}(i) \triangleq \boldsymbol{w}^\star - \boldsymbol{w}(i). \tag{12}$$

To perform the theoretical analysis, we introduce the following independence assumption.

Assumption 1: (Independent regressors) The regression vectors $\boldsymbol{x}_k(i)$ arise from a zero-mean random process that is temporally stationary, white, and independent over space. ■

Using data model (1), the error recursion can be written in the following form:

$$\widetilde{\boldsymbol{w}}(i+1) = \boldsymbol{\mathcal{B}}(i)\widetilde{\boldsymbol{w}}(i) - \boldsymbol{g}(i) + \eta\boldsymbol{r}(i) \tag{13}$$

where

$$\boldsymbol{\mathcal{B}}(i) \triangleq \boldsymbol{\mathcal{A}}^\top(i)[\boldsymbol{I}_{NL} - \boldsymbol{\mathcal{M}}(i)(\boldsymbol{\mathcal{R}}_x(i) + \eta\boldsymbol{\mathcal{Q}}(i))] \tag{14}$$
$$\boldsymbol{g}(i) \triangleq \boldsymbol{\mathcal{A}}^\top(i)\boldsymbol{\mathcal{M}}(i)\text{col}\{\boldsymbol{x}_k(i)z_k(i)\}_{k=1}^N \tag{15}$$
$$\boldsymbol{r}(i) \triangleq \boldsymbol{\mathcal{A}}^\top(i)\boldsymbol{\mathcal{M}}(i)\boldsymbol{\mathcal{Q}}(i)\boldsymbol{w}^\star \tag{16}$$

with

$$\boldsymbol{\mathcal{A}}(i) \triangleq \boldsymbol{A}(i) \otimes \boldsymbol{I}_L \tag{17}$$
$$\boldsymbol{\mathcal{M}}(i) \triangleq \boldsymbol{M}(i) \otimes \boldsymbol{I}_L \tag{18}$$
$$\boldsymbol{\mathcal{Q}}(i) \triangleq \boldsymbol{I}_{NL} - \boldsymbol{P}(i) \otimes \boldsymbol{I}_L \tag{19}$$

and $\boldsymbol{\mathcal{R}}_x(i)$ is the $N \times N$ block diagonal matrix whose k-th block is the $L \times L$ matrix $\boldsymbol{x}_k(i)\boldsymbol{x}_k^\top(i)$.

A. Mean behavior analysis

Taking the expectation of both sides of (13) and using Assumption 1 yields the following condition for stability. The result follows from the analysis of the spectral radius of $\boldsymbol{\mathcal{B}} \triangleq \mathbb{E}\{\boldsymbol{\mathcal{B}}(i)\}$ defined in (22).

Lemma 1: (Mean stability) Assume that all agents have the same step-size expectation, that is, $\mathbb{E}\{\mu_k(i)\} = \bar{\mu}$ for all k. For any initial conditions, the asynchronous multitask diffusion algorithm (4) converges in the mean if $\bar{\mu}$ satisfies:

$$0 < \bar{\mu} < \frac{2}{\max_{1\leq k\leq N}\{\rho(\boldsymbol{R}_{x,k})\} + 2\eta} \tag{20}$$

where ρ denotes the spectral radius of its matrix argument. The asymptotic mean bias is given by:

$$\lim_{i\to\infty} \mathbb{E}\{\widetilde{\boldsymbol{w}}(i)\} = \eta(\boldsymbol{I}_{NL} - \boldsymbol{\mathcal{B}})^{-1}\boldsymbol{r} \tag{21}$$

where

$$\begin{aligned} \boldsymbol{\mathcal{B}} &\triangleq \boldsymbol{\mathcal{A}}^\top[\boldsymbol{I}_{NL} - \boldsymbol{\mathcal{M}}(\boldsymbol{\mathcal{R}}_x + \eta\, \boldsymbol{\mathcal{Q}})] && (22)\\ \boldsymbol{r} &\triangleq \boldsymbol{\mathcal{A}}^\top \boldsymbol{\mathcal{M}}\, \boldsymbol{\mathcal{Q}}\, \boldsymbol{w}^\star && (23) \end{aligned}$$

with $\boldsymbol{\mathcal{A}} \triangleq \overline{\boldsymbol{A}} \otimes \boldsymbol{I}_L$, $\boldsymbol{\mathcal{M}} \triangleq \overline{\boldsymbol{M}} \otimes \boldsymbol{I}_L$, $\boldsymbol{\mathcal{Q}} \triangleq \boldsymbol{I}_{NL} - \overline{\boldsymbol{P}} \otimes \boldsymbol{I}_L$, and $\boldsymbol{\mathcal{R}}_x$ is the block diagonal matrix whose k-th block is $\boldsymbol{R}_{x,k}$.■

B. Mean-square behavior analysis

We shall now use the block Kronecker product $\otimes_b$, and the block vectorization operation bvec$(\cdot)$, since these operators allow to exploit the block structure of matrices [15], [17]. Before proceeding, let us introduce some useful matrices:

$$\boldsymbol{\mathcal{M}}_{\rm I} \triangleq \mathbb{E}\{\boldsymbol{\mathcal{M}}(i) \otimes_b \boldsymbol{\mathcal{M}}(i)\} = (\overline{\boldsymbol{M}} \otimes \overline{\boldsymbol{M}} + \boldsymbol{C}_M) \otimes \boldsymbol{I}_{L^2} \tag{24}$$

$$\boldsymbol{\mathcal{A}}_{\rm I} \triangleq \mathbb{E}\{\boldsymbol{\mathcal{A}}(i) \otimes_b \boldsymbol{\mathcal{A}}(i)\} = (\overline{\boldsymbol{A}} \otimes \overline{\boldsymbol{A}} + \boldsymbol{C}_A) \otimes \boldsymbol{I}_{L^2} \tag{25}$$

$$\begin{aligned} \boldsymbol{\mathcal{Q}}_{\rm I} &\triangleq \mathbb{E}\{\boldsymbol{\mathcal{Q}}(i) \otimes_b \boldsymbol{\mathcal{Q}}(i)\} && (26)\\ &= (\boldsymbol{I}_{N^2} - \boldsymbol{I}_N \otimes \overline{\boldsymbol{P}} - \overline{\boldsymbol{P}} \otimes \boldsymbol{I}_N + \overline{\boldsymbol{P}} \otimes \overline{\boldsymbol{P}} + \boldsymbol{C}_P) \otimes \boldsymbol{I}_{L^2} \end{aligned}$$

By Assumption 1 and (13), the mean-square of the weight error vector $\widetilde{\boldsymbol{w}}(i+1)$, weighted by any positive semi-definite matrix $\boldsymbol{\Sigma}$, satisfies the following relation:

$$\begin{aligned} \mathbb{E}\{\|\widetilde{\boldsymbol{w}}(i+1)\|^2_{\boldsymbol{\Sigma}}\} = \mathbb{E}\{\|\widetilde{\boldsymbol{w}}(i)\|^2_{\boldsymbol{\Sigma}'}\} + \mathbb{E}\{\|\boldsymbol{g}(i)\|^2_{\boldsymbol{\Sigma}}\} +\\ 2\eta\mathbb{E}\{\boldsymbol{r}^\top(i)\boldsymbol{\Sigma}\boldsymbol{\mathcal{B}}(i)\widetilde{\boldsymbol{w}}(i)\} + \eta^2\mathbb{E}\{\|\boldsymbol{r}(i)\|^2_{\boldsymbol{\Sigma}}\} \end{aligned} \tag{27}$$

with $\|\boldsymbol{x}\|^2_{\boldsymbol{\Sigma}} = \boldsymbol{x}^\top\boldsymbol{\Sigma}\boldsymbol{x}$ and $\boldsymbol{\Sigma}' = \mathbb{E}\{\boldsymbol{\mathcal{B}}^\top(i)\boldsymbol{\Sigma}\boldsymbol{\mathcal{B}}(i)\}$. The freedom in selecting $\boldsymbol{\Sigma}$ will allow us to derive several performance metrics. Let $\boldsymbol{\sigma} \triangleq$ bvec$(\boldsymbol{\Sigma})$ and $\boldsymbol{\sigma}' \triangleq$ bvec$(\boldsymbol{\Sigma}')$. Using that bvec$(\boldsymbol{U}\boldsymbol{V}\boldsymbol{W}) = (\boldsymbol{W}^\top \otimes_b \boldsymbol{U})$bvec$(\boldsymbol{V})$, it can be checked that $\boldsymbol{\sigma}$ and $\boldsymbol{\sigma}'$ are related by the following relationship $\boldsymbol{\sigma}' = \boldsymbol{\mathcal{F}}^\top\boldsymbol{\sigma}$, where $\boldsymbol{\mathcal{F}}$ is the $(NL)^2 \times (NL)^2$ matrix given by:

$$\begin{aligned} \boldsymbol{\mathcal{F}} &\triangleq \mathbb{E}\{\boldsymbol{\mathcal{B}}(i) \otimes_b \boldsymbol{\mathcal{B}}(i)\}\\ &\approx \boldsymbol{\mathcal{A}}_{\rm I}^\top\big[\boldsymbol{I}_{(NL)^2} - \boldsymbol{I}_{NL} \otimes_b \boldsymbol{\mathcal{M}}(\boldsymbol{\mathcal{R}}_x + \eta\boldsymbol{\mathcal{Q}}) -\\ &\qquad \boldsymbol{\mathcal{M}}(\boldsymbol{\mathcal{R}}_x + \eta\boldsymbol{\mathcal{Q}}) \otimes_b \boldsymbol{I}_{NL}\big] \end{aligned} \tag{28}$$

where, considering the case of sufficiently small step-sizes, terms involving higher order moments of the step-sizes have been ignored.

By expressing the second term on the RHS of equation (27) as tr$(\boldsymbol{\Sigma}\mathbb{E}\{\boldsymbol{g}(i)\boldsymbol{g}^\top(i)\})$ and using tr$(\boldsymbol{\Sigma}\boldsymbol{W}) =$ bvec$(\boldsymbol{W}^\top)^\top\boldsymbol{\sigma}$, we obtain:

$$\mathbb{E}\{\|\boldsymbol{g}(i)\|^2_{\boldsymbol{\Sigma}}\} = \boldsymbol{g}_b^\top\boldsymbol{\sigma} \tag{29}$$

with $\boldsymbol{g}_b \triangleq \boldsymbol{\mathcal{A}}_{\rm I}^\top\boldsymbol{\mathcal{M}}_{\rm I}\,$bvec$(\boldsymbol{\mathcal{S}})$ and $\boldsymbol{\mathcal{S}} \triangleq$ diag$\{\sigma^2_{z,k}\boldsymbol{R}_{x,k}\}_{k=1}^N$. In the same way, we get:

$$\mathbb{E}\{\|\boldsymbol{r}(i)\|^2_{\boldsymbol{\Sigma}}\} = \boldsymbol{r}_b^\top\boldsymbol{\sigma} \tag{30}$$

where $\boldsymbol{r}_b \triangleq \boldsymbol{\mathcal{A}}_{\rm I}^\top\boldsymbol{\mathcal{M}}_{\rm I}\boldsymbol{\mathcal{Q}}_{\rm I}\,$bvec$(\boldsymbol{w}^\star\boldsymbol{w}^{\star\top})$. Finally, the third term on the RHS of (27) is given by:

$$\mathbb{E}\{\boldsymbol{r}^\top(i)\boldsymbol{\Sigma}\boldsymbol{\mathcal{B}}(i)\widetilde{\boldsymbol{w}}(i)\} = \mathbb{E}\{\widetilde{\boldsymbol{w}}(i)\}^\top\mathbb{E}\{\boldsymbol{\mathcal{B}}(i) \otimes_b \boldsymbol{r}(i)\}^\top\boldsymbol{\sigma} \tag{31}$$

where

$$\begin{aligned} \boldsymbol{K} &\triangleq \mathbb{E}\{\boldsymbol{\mathcal{B}}(i) \otimes_b \boldsymbol{r}(i)\}\\ &= \boldsymbol{\mathcal{A}}_{\rm I}^\top\big[(\boldsymbol{I}_{NL} \otimes_b \boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{Q}}\boldsymbol{w}^\star) -\\ &\qquad \boldsymbol{\mathcal{M}}_{\rm I}((\boldsymbol{\mathcal{R}}_x \otimes_b \boldsymbol{\mathcal{Q}}\boldsymbol{w}^\star) + \eta\boldsymbol{\mathcal{Q}}_{\rm I}(\boldsymbol{I}_{NL} \otimes_b \boldsymbol{w}^\star))\big]. \end{aligned} \tag{32}$$

Finally, the weighted variance $\mathbb{E}\{\|\widetilde{\boldsymbol{w}}(i)\|^2_{\boldsymbol{\Sigma}}\}$ can be expressed as:

$$\begin{aligned} \mathbb{E}\{\|\widetilde{\boldsymbol{w}}(i+1)\|^2_{\boldsymbol{\sigma}}\} = \mathbb{E}\{\|\widetilde{\boldsymbol{w}}(i)\|^2_{\boldsymbol{\mathcal{F}}^\top\boldsymbol{\sigma}}\} + \boldsymbol{g}_b^\top\boldsymbol{\sigma} +\\ 2\eta\mathbb{E}\{\widetilde{\boldsymbol{w}}(i)\}^\top\boldsymbol{K}^\top\boldsymbol{\sigma} + \eta^2\boldsymbol{r}_b^\top\boldsymbol{\sigma}. \end{aligned} \tag{33}$$

Note that we use interchangeably $\|.\|^2_{\boldsymbol{\Sigma}}$ and $\|.\|^2_{\boldsymbol{\sigma}}$ to refer to the same square weighted norm using $\boldsymbol{\Sigma}$ or its block vector representation $\boldsymbol{\sigma}$. Iterating expression (33) starting from $i = 0$, it can be shown that $\mathbb{E}\{\|\widetilde{\boldsymbol{w}}(i+1)\|^2_{\boldsymbol{\sigma}}\}$ converges to a bounded value, as i tends to infinity provided that $\boldsymbol{\mathcal{F}}$ in (28) is stable.

Lemma 2: (Mean-square stability) Assume that all agents have the same step-size expectation, that is, $\mathbb{E}\{\mu_k(i)\} = \bar{\mu}$ for all k. Assume further that $\{\mu_{\max,k}\}$ are sufficiently small. The asynchronous multitask diffusion algorithm (4) is mean-square stable if the matrix $\boldsymbol{\mathcal{F}}$ is stable. ■

Iterating equation (33) until time instants i and $i+1$, and comparinfg these expressions, we can relate $\mathbb{E}\{\|\widetilde{\boldsymbol{w}}(i+1)\|^2_{\boldsymbol{\sigma}}\}$ to $\mathbb{E}\{\|\widetilde{\boldsymbol{w}}(i)\|^2_{\boldsymbol{\sigma}}\}$. This leads to the following result.

Corollary 1: (Transient behavior) Consider sufficiently small step-sizes that ensure mean and mean-square stability. Then, the variance curve $\zeta(i+1) = \mathbb{E}\{\|\widetilde{\boldsymbol{w}}(i+1)\|^2_{\boldsymbol{\sigma}}\}$ evolves according to the following recursion for $i \geq 0$:

$$\begin{aligned} \zeta(i+1) =& \zeta(i) - \|\widetilde{\boldsymbol{w}}(0)\|^2_{(\boldsymbol{I}_{(NL)^2} - \boldsymbol{\mathcal{F}}^\top)(\boldsymbol{\mathcal{F}}^\top)^i\boldsymbol{\sigma}} +\\ &\boldsymbol{g}_b^\top(\boldsymbol{\mathcal{F}}^\top)^i\boldsymbol{\sigma} + \eta^2\boldsymbol{r}_b^\top(\boldsymbol{\mathcal{F}}^\top)^i\boldsymbol{\sigma} +\\ &2\eta\mathbb{E}\{\widetilde{\boldsymbol{w}}(i)\}^\top\boldsymbol{K}^\top\boldsymbol{\sigma} + 2\eta\boldsymbol{\Gamma}(i)\boldsymbol{\sigma} \end{aligned} \tag{34}$$

where $\boldsymbol{\Gamma}(i)$ is the $1 \times (NL)^2$ row vector updated as follows:

$$\boldsymbol{\Gamma}(i+1) = \boldsymbol{\Gamma}(i)\boldsymbol{\mathcal{F}}^\top + \mathbb{E}\{\widetilde{\boldsymbol{w}}(i)\}^\top\boldsymbol{K}^\top(\boldsymbol{\mathcal{F}}^\top - \boldsymbol{I}_{(NL)^2}), \tag{35}$$

and $\widetilde{\boldsymbol{w}}(0)$ is the initial condition. ■

Expression (34) allows us to derive several performance metrics through the proper selection of $\boldsymbol{\Sigma}$. For instance, the network MSD value at time instant i, defined by MSD$_{net}(i) \triangleq \frac{1}{N}\mathbb{E}\{\|\widetilde{\boldsymbol{w}}(i)\|^2\}$, is obtained for $\boldsymbol{\Sigma} = \frac{1}{N}\boldsymbol{I}_{NL}$. The MSD of cluster $\mathcal{C}_q$ at time instant i is defined as:

$$\text{MSD}_{\mathcal{C}_q}(i) \triangleq \frac{1}{n_q}\sum_{k\in\mathcal{C}_q}\mathbb{E}\{\|\widetilde{\boldsymbol{w}}_k(i)\|^2\} \tag{36}$$

where n_q is the number of nodes in cluster $\mathcal{C}_q$. This quantity can be obtained by computing $\mathbb{E}\{\|\widetilde{\boldsymbol{w}}(i+1)\|^2_{\boldsymbol{\Sigma}_{\mathcal{C}_q}}\}$ with a block diagonal weighting matrix $\boldsymbol{\Sigma}_{\mathcal{C}_q}$ that has the block $\frac{1}{n_q}\boldsymbol{I}_L$ as k-th entry, for all $k \in \mathcal{C}_q$, and zeros elsewhere.

Corollary 2: (Steady-state variance relation) If convergence is achieved, then

$$\begin{aligned} &\lim_{i\to\infty}\mathbb{E}\{\|\widetilde{\boldsymbol{w}}(i)\|^2_{(\boldsymbol{I}_{(NL)^2} - \boldsymbol{\mathcal{F}}^\top)\boldsymbol{\sigma}}\}\\ &= \boldsymbol{g}_b^\top\boldsymbol{\sigma} + \eta^2\boldsymbol{r}_b^\top\boldsymbol{\sigma} + 2\eta\mathbb{E}\{\widetilde{\boldsymbol{w}}(\infty)\}^\top\boldsymbol{K}^\top\boldsymbol{\sigma}. \end{aligned} \tag{37}$$

■

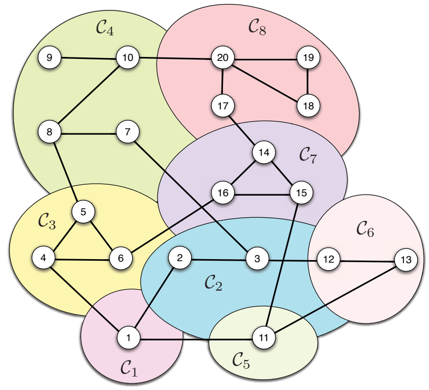

Fig. 1. Network topology.

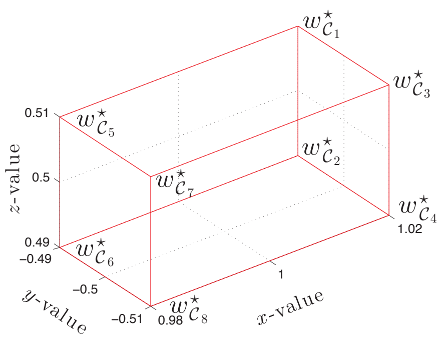

Fig. 2. Network configuration: parameter vector inputs.

To determine the steady-state network MSD from equation (37), we set $\boldsymbol{\sigma}$ to $\frac{1}{N}(\boldsymbol{I}_{(NL)^2} - \boldsymbol{\mathcal{F}}^\top)^{-1}\text{bvec}(\boldsymbol{I}_{NL})$. The steady-state MSD of cluster $\mathcal{C}_q$ is obtained by setting $\boldsymbol{\sigma}$ to $(\boldsymbol{I}_{(LN)^2} - \boldsymbol{\mathcal{F}}^\top)^{-1}\text{bvec}(\boldsymbol{\Sigma}_{\mathcal{C}_q})$.

IV. Simulation Results

The asynchronous ATC model (4) was run over the clustered network shown in Fig. 1, consisting of $N = 20$ nodes divided into 8 clusters. The vectors to estimate, $\boldsymbol{w}^\star_{\mathcal{C}_i}$, were of length $L = 3$, with entries defined as in Fig. 2. As we can see from Figs. 1 and 2, two clusters are connected if their optimum parameter vectors share two identical components. The regression vectors were zero-mean random vectors governed by a Gaussian distribution with covariance matrix $\boldsymbol{R}_{x,k} = \sigma^2_{x,k}\boldsymbol{I}_L$. The background noises $z_k(i)$ were i.i.d. zero-mean Gaussian random variables, independent of any other signal, with variance $\sigma^2_{z,k}$. The variances $\sigma^2_{x,k}$ and $\sigma^2_{z,k}$ are shown in Fig. 3.

We used the Bernoulli asynchronous model [14] with fixed underlying topology. The step-sizes $\mu_k(i)$ were distributed as:

$$\mu_k(i) = \begin{cases} \mu_k, & \text{with probability } q_k \\ 0, & \text{with probability } 1 - q_k \end{cases} \tag{38}$$

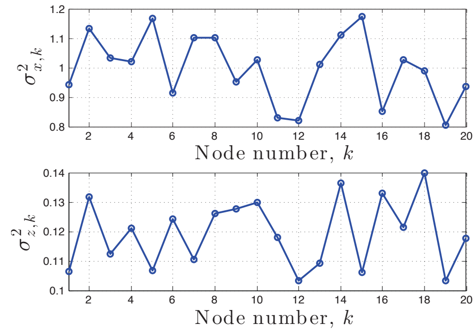

Fig. 3. Network configuration: input and noise variances.

with μ_k a fixed step-size. The combination weights $\{a_{\ell k}(i)\}$ were distributed as follows:

$$a_{\ell k}(i) = \begin{cases} a_{\ell k}, & \text{with probability } p_{\ell k} \\ 0, & \text{with probability } 1 - p_{\ell k} \end{cases} \tag{39}$$

for all $\ell \in \mathcal{N}_k^-(i) \cap \mathcal{C}(k)$, where $0 < a_{\ell k} < 1$ is a fixed coefficient, and $\mathcal{N}_k^-(i)$ denotes $\mathcal{N}_k(i) \setminus \{k\}$. The combination coefficients $\{a_{\ell k}(i)\}$ were spatially uncorrelated for $\ell \neq k$. Each node k was able to set the combination coefficient $a_{kk}(i)$ at each iteration i as follows:

$$a_{kk}(i) = 1 - \sum_{\ell \in \mathcal{N}_k^-(i) \cap \mathcal{C}(k)} a_{\ell k}(i) \tag{40}$$

to ensure condition (5). The weights $\{\rho_{k\ell}(i)\}$ were distributed as follows:

$$\rho_{k\ell}(i) = \begin{cases} \rho_{k\ell}, & \text{with probability } r_{k\ell} \\ 0, & \text{with probability } 1 - r_{k\ell} \end{cases} \tag{41}$$

for all $\ell \in \mathcal{N}_k(i) \setminus \mathcal{C}(k)$, where $0 < \rho_{k\ell} < 1$ is a fixed regularization factor. The factors $\{\rho_{k\ell}(i)\}$ were spatially uncorrelated for $k \neq \ell$. At each iteration i, each node k was able to adjust $\rho_{kk}(i)$ as follows:

$$\rho_{kk}(i) = 1 - \sum_{\ell \in \mathcal{N}_k(i) \setminus \mathcal{C}(k)} \rho_{k\ell}(i) \tag{42}$$

to ensure condition (6).

We set the coefficient $a_{\ell k}$ in (39) such that $a_{\ell k} = |\mathcal{N}_k \cap \mathcal{C}(k)|^{-1}$ for all $\ell \in \mathcal{N}_k \cap \mathcal{C}(k)$, where $|\cdot|$ denotes the cardinality of its argument and the neighborhood $\mathcal{N}_k$ is the union of all possible realizations for the random neighborhood $\mathcal{N}_k(i)$. We set the factors $\rho_{k\ell}$ in (41) to $\rho_{k\ell} = |\mathcal{N}_k \setminus \mathcal{C}(k)|^{-1}$ for $\ell \in \mathcal{N}_k \setminus \mathcal{C}(k)$, and $\rho_{k\ell} = 0$ for any other ℓ. The upper bounds $\mu_{\max,k}$ were uniformly set to 0.03. The regularization strength η was set to 1. The MSD curves were averaged over 100 Monte-Carlo runs. Three different scenarios were considered:

1) 50% idle: $q_k = p_{\ell k} = r_{k\ell} = 0.5$;
2) 30% idle: $q_k = p_{\ell k} = r_{k\ell} = 0.7$;
3) no idle nodes: $q_k = p_{\ell k} = r_{k\ell} = 1$.

It can be observed in Fig. 4 that the simulation results match well the theoretical results.

We also considered the following simulation. We kept the same coefficients $\{a_{\ell k}\}$ and $\{\rho_{k\ell}\}$ as the previous simulation.

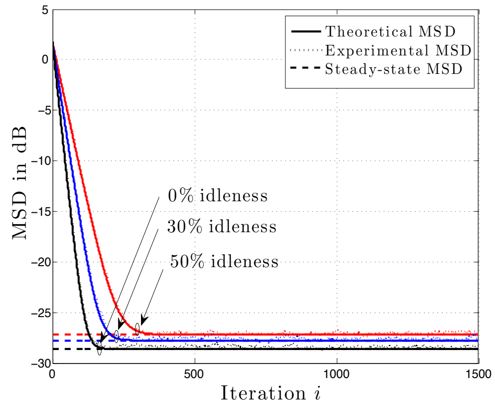

Fig. 4. Network mean-square deviation (MSD).

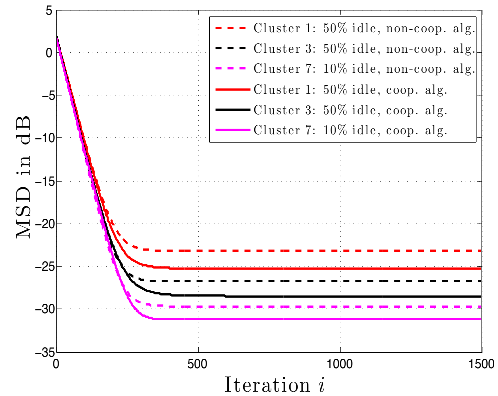

Fig. 5. Cooperative vs. noncooperative multitask learning.

Parameters μ_k in (38) were set to $\mu_k = 0.03$ for clusters $\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3, \mathcal{C}_4$ and to $\mu_k = 0.015$ for the remaining clusters. For the first four clusters, nodes turned off with probability $1 - q_k = 0.5$, and intra-cluster links failed with probability $1 - p_{\ell k} = 0.5$. For the four last clusters, we used $1 - p_{\ell k} = 1 - q_k = 0.1$. The probability that an inter-cluster link between two nodes k and ℓ fails was set to $1 - r_{k\ell} = 0.3$. We compared the asynchronous multitask network with its noncooperative multitask counterpart obtained by setting η to 0. As shown in Fig. 5, the performance improves by exploiting cooperation between clusters. Moreover, for a given cluster, when the number of nodes increases or the probabilities of success associated with the Bernoulli variables increase, the learning is enhanced.

V. Conclusion

In this paper, we studied the performance of the multitask diffusion LMS algorithm over asynchronous networks. We analyzed the behavior of the proposed asynchronous model in the mean and mean-square error sense. Simulations were presented to illustrate our theoretical results. Several open problems still have to be solved for specific applications. For instance, it would be interesting to show which regularization can be advantageously used with our distributed multitask algorithm, and how they can be efficiently implemented in an adaptive manner. It would also be interesting to investigate how nodes can autonomously adjust regularization parameters to optimize the learning performance and how they can learn the structure of the clusters in real-time.

References

[1] A. H. Sayed, "Adaptive networks," *Proc. of IEEE*, vol. 102, no. 4, pp. 460–497, Apr. 2014.

[2] J. Tsitsiklis, D. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Trans. Autom. Control*, vol. 31, no. 9, pp. 803–812, Sept. 1986.

[3] A. G. Dimakis, S. Kar, J. M. F. Moura, M. G. Rabbat, and A. Scaglione, "Gossip algorithms for distributed signal processing," *Proc. of IEEE*, vol. 98, no. 11, pp. 1847–1864, Nov. 2010.

[4] M. G. Rabbat and R. D. Nowak, "Quantized incremental algorithms for distributed optimization," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 4, pp. 798–808, Apr. 2005.

[5] A. H. Sayed, S.-Y. Tu, J. Chen, X. Zhao, and Z. Towfic, "Diffusion strategies for adaptation and learning over networks: an examination of distributed strategies and network behavior," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 155–171, May 2013.

[6] J. Chen, C. Richard, and A. H. Sayed, "Diffusion LMS over multitask networks," *Submitted for publication.* Also available as arXiv:1404.6813, Apr. 2014.

[7] X. Zhao and A. H. Sayed, "Clustering via diffusion adaptation over networks," in *Proc. CIP*, Parador de Baiona, Spain, May 2012, pp. 1–6.

[8] A. Bertrand and M. Moonen, "Distributed adaptive node-specific signal estimation in fully connected sensor networks – Part I: sequential node updating," *IEEE Trans. Signal Process.*, vol. 58, no. 10, pp. 5277–5291, Oct. 2010.

[9] A. Bertrand and M. Moonen, "Distributed adaptive estimation of node-specific signals in wireless sensor networks with a tree topology," *IEEE Trans. Signal Process.*, vol. 59, no. 5, pp. 2196–2210, May 2011.

[10] N. Bogdanović, J. Plata-Chaves, and K. Berberidis, "Distributed diffusion-based LMS for node-specific parameter estimation over adaptive networks," in *Proc. IEEE ICASSP*, Florence, Italy, May 2014, pp. 7223–7227.

[11] J. Chen, C. Richard, A. O. Hero, and A. H. Sayed, "Diffusion LMS for multitask problems with overlapping hypothesis subspaces," in *Proc. IEEE MLSP*, Reims, France, Sept. 2014, pp. 1–6.

[12] J. Chen, C. Richard, and A. H. Sayed, "Multitask diffusion adaptation over networks," *IEEE Trans. Signal Process.*, vol. 62, no. 16, pp. 4129–4144, Aug. 2014.

[13] X. Zhao and A. H. Sayed, "Asynchronous diffusion adaptation over networks," in *Proc. EUSIPCO*, Romania, Aug. 2012, pp. 86–90.

[14] X. Zhao and A. H. Sayed, "Asynchronous adaptation and learning over networks-Part I: Modeling and stability analysis," *available as arXiv:1312.5434*, Dec. 2013.

[15] X. Zhao and A. H. Sayed, "Asynchronous adaptation and learning over networks-Part II: Performance analysis," *available as arXiv:1312.5438*, Dec. 2013.

[16] T. Basar and G. J. Olsder, *Dynamic Noncooperative Game Theory*, SIAM, PA, 1999.

[17] R. H. Koning, H. Neudecker, and T. Wansbeek, "Block Kronecker products and the vecb operator," *Linear Algebra and its Applications*, vol. 149, pp. 165–184, April 1991.