On reducing the communication cost of the diffusion LMS algorithm

Ibrahim El Khalil Harrane, Rémi Flamary, Cédric Richard

On reducing the communication cost of the diffusion LMS algorithm

Ibrahim El Khalil Harrane, Rémi Flamary, Cédric Richard, *Senior Member*, *IEEE*
Université Côte d'Azur, OCA, CNRS, France
ibrahim.harrane@oca.eu, remi.flamary@unice.fr, cedric.richard@unice.fr

***Abstract*—The rise of digital and mobile communications has recently made the world more connected and networked, resulting in an unprecedented volume of data flowing between sources, data centers, or processes. While these data may be processed in a centralized manner, it is often more suitable to consider distributed strategies such as diffusion as they are scalable and can handle large amounts of data by distributing tasks over networked agents. Although it is relatively simple to implement diffusion strategies over a cluster, it appears to be challenging to deploy them in an ad-hoc network with limited energy budget for communication. In this paper, we introduce a diffusion LMS strategy that significantly reduces communication costs without compromising the performance. Then, we analyze the proposed algorithm in the mean and mean-square sense. Next, we conduct numerical experiments to confirm the theoretical findings. Finally, we perform large scale simulations to test the algorithm efficiency in a scenario where energy is limited.**

I. Introduction

Adaptive networks are collections of interconnected agents that continuously learn and adapt from streaming measurements to perform a preassigned task such as parameter estimation. The agents are able to share information besides their own data, and collaborate in order to enhance the solution accuracy. Adaptive networks have proven to be powerful tools for modeling natural and social phenomena, ranging from organized organisms to social networks [1]. They are mainly used for data mining tasks over high dimensional data sets locally collected by distributed agents, in a decentralized and cooperative manner. In such scenarios, among other possible strategies [11]–[13], diffusion strategies are a safer option than centralized strategies due to their robustness and resilience to agent and link failures. In particular, the diffusion LMS plays a central role with its enhanced efficiency and low complexity. It has been extensively studied in the literature in single task [1]–[3] and multitask inference problems [6]–[10]. It has also been considered in other contexts such as nonlinear system identification [14] and dictionary learning [15].

The advent of the Internet of Things and sensor networks has opened new horizons for diffusion strategies but brought up new challenges as well. Indeed, as illustrated in Fig.1 (a), diffusion strategies inherently require all nodes to exchange information with their neighbors at each iteration. In the case of diffusion LMS, as will be detailed in the next section, this information can be either local estimates and gradients of local cost functions, or local estimates only. Even in the latter case, this requirement imposes a substantial burden on communication and energy resources. Reducing the communication cost while maintaining the benefits of cooperation is therefore of major importance for systems with limited energy budget such as wireless sensor networks. In recent years, several strategies were proposed to address this issue. There are mainly two approaches which we illustrate in Fig. 1 (b) and (c). On the one hand, some authors propose to restrict the number of active links between neighboring nodes at each time instant [5]. On the other hand, there are authors that recommend to reduce the communication load by projecting parameter vectors onto lower dimensional spaces before transmission [16], or transmitting only partial parameter vectors [17]–[19]. The literature has mainly focused on the case where only the local estimates are shared at each time instant. However, to achieve faster convergence, it is also of interest to consider the case where both local estimates and gradients of local cost functions can be shared.

In this paper we propose an algorithm where every transmitted parameter vectors, either local estimates or gradients of local cost functions, are partially shared. The network flow is controlled by two parameters, the number of entries of each one of these two types of parameter vectors. Then, we study the stochastic behavior of the algorithm in the mean and mean-square sense. Next, we perform numerical experiments to confirm the theoretical findings. Furthermore, we characterize the algorithm performance for high dimensional data in a large network. We compare our algorithm with the diffusion LMS in a sensor network scenario where energy resource is scarce. Finally, we conclude this paper.

Notation: Boldface small letters denote vectors. All vectors are column vectors. Boldface capital letters denote matrices. The (k,ℓ)-th entry of a matrix is denoted by $(\cdot)_{k\ell}$, and the (k,ℓ)-th block of a block matrix is denoted by $[\,\cdot\,]_{k\ell}$. Matrix trace is denoted by $\mathrm{trace}\{\cdot\}$. The expectation operator is denoted by $\mathbb{E}\{\cdot\}$. The identity matrix of size N is denoted by $\boldsymbol{I}_N$, and the all-one vector of length N is denoted by $\mathbf{1}_N$. We denote by $\mathcal{N}_k$ the set of node indices in the neighborhood of node k, including k itself, and $|\mathcal{N}_k|$ its cardinality. The operator $\mathrm{col}\{\cdot\}$ stacks its vector arguments on the top of each other to generate a connected vector. The notation $\mathrm{diag}\{a,b\}$ denotes a diagonal matrix with entries a and b. Likewise, the notation $\mathrm{diag}\{\boldsymbol{A},\boldsymbol{B}\}$ denotes a block diagonal matrix with block entries $\boldsymbol{A}$ and $\boldsymbol{B}$. The other symbols will be defined in the context where they are used.

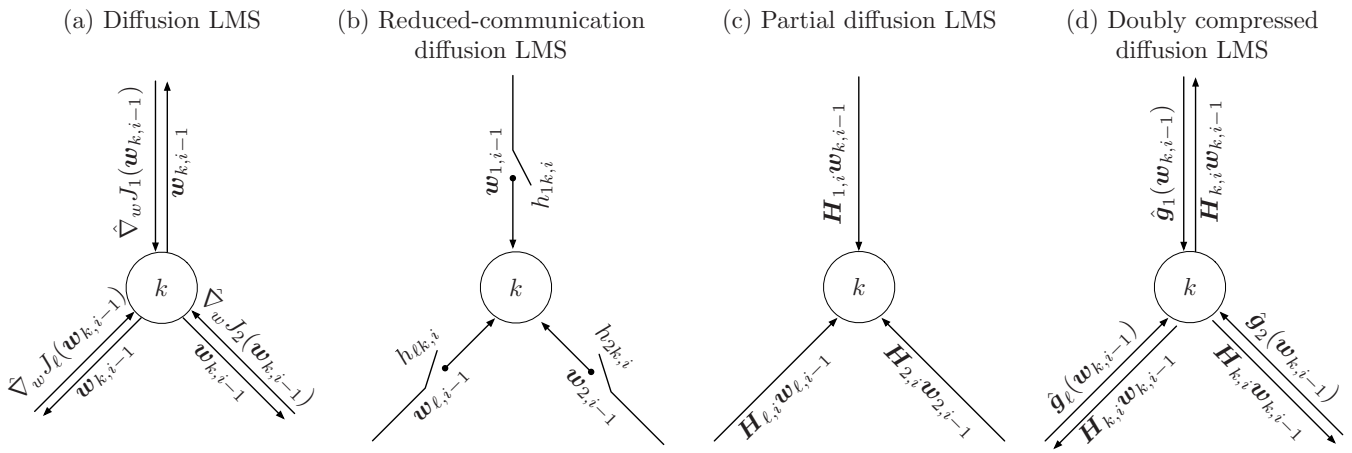

Fig. 1: Illustrative representation of transmitted data for the diffusion LMS and different approaches aiming at reducing the communication load for a node k.

II. Diffusion LMS and resource-saving variants

A. Diffusion LMS

Consider a connected network composed of N nodes. The aim of each node is to estimate an $L \times 1$ unknown parameter vector $\boldsymbol{w}^o$ from collected measurements. Node k has access to local streaming measurements $\{d_k(i), \boldsymbol{u}_{k,i}\}$ where $d_k(i)$ is a scalar zero-mean reference signal, and $\boldsymbol{u}_{k,i}$ is an $L \times 1$ zero-mean regression vector with a positive definite covariance matrix $\boldsymbol{R}_{u_k} = \mathbb{E}\{\boldsymbol{u}_{k,i}\boldsymbol{u}_{k,i}^\top\}$. The data at agent k and time i are assumed to be related via the linear regression model:

$$d_k(i) = \boldsymbol{u}_{k,i}^\top \boldsymbol{w}^o + v_k(i) \tag{1}$$

where $\boldsymbol{w}^o$ is the unknown parameter vector to be estimated, and $v_k(i)$ is a zero-mean i.i.d. noise with variance $\sigma_{v,k}^2$. The noise $v_k(i)$ is assumed to be independent of any other signal. Let $J_k(\boldsymbol{w})$ be a differentiable convex cost function at agent k. In this paper, we shall consider the mean-square-error criterion, namely,

$$J_k(\boldsymbol{w}) = E\{|d_k(i) - \boldsymbol{u}_{k,i}^\top \boldsymbol{w}|^2\} \tag{2}$$

This criterion is strongly convex, second-order differentiable, and minimized at $\boldsymbol{w}^o$.

Diffusion LMS strategies seek the minimizer of the aggregate cost function:

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

in a cooperative manner. Let $\boldsymbol{w}_{k,i}$ denote the estimate of the minimizer of (3) at node k and time instant i. Diffusion LMS algorithm in its Adapt-then-Combine (ATC) form is given by:

$$\boldsymbol{\psi}_{k,i} = \boldsymbol{w}_{k,i-1} - \mu_k \sum_{\ell \in \mathcal{N}_k} c_{\ell k} \hat{\nabla}_w J_\ell(\boldsymbol{w}_{k,i-1}) \tag{4}$$

$$\boldsymbol{w}_{k,i} = \sum_{\ell \in \mathcal{N}_k} a_{\ell k} \boldsymbol{\psi}_{\ell,i} \tag{5}$$

with $\hat{\nabla}_w J_\ell(\boldsymbol{w}_{k,i-1}) = -\boldsymbol{u}_{\ell,i}[d_\ell(i) - \boldsymbol{u}_{\ell,i}^\top \boldsymbol{w}_{k,i-1}]$ the instantaneous approximation of the gradient vector $\nabla_w J_\ell(\boldsymbol{w}_{k,i-1})$, $\mathcal{N}_k$ the neighborhood of node k including k, and μ_k a positive step-size. The nonnegative coefficients $\{a_{\ell k}\}$ and $\{c_{\ell k}\}$ are the (ℓ, k)-th entries of a left-stochastic matrix $\boldsymbol{A}$ and a right-stochastic matrix $\boldsymbol{C}$, respectively.

B. Reducing the communication load of diffusion LMS

We shall now describe the existing techniques for reducing the communication load of the diffusion LMS. We start with the reduced communication diffusion LMS (RCD) [5] where each node k can only communicate with a subset of size m_k of its $|\mathcal{N}_k|$ neighbors. This subset is randomly selected at each node and each iteration. Each agent in the neighborhood of k can be selected with probability:

$$p_k = \tfrac{m_k}{N_k} \tag{6}$$

The algorithm can be formulated as:

$$\begin{cases} \boldsymbol{\psi}_{k,i} &= \boldsymbol{w}_{k,i-1} + \mu_k \boldsymbol{u}_{k,i} \left(d_{k,i} - \boldsymbol{u}_{k,i}^\top \boldsymbol{w}_{k,i-1}\right) \\ \boldsymbol{w}_{k,i} &= h_{kk,i} \boldsymbol{\psi}_{k,i} + \displaystyle\sum_{\ell \in \mathcal{N}_k \setminus \{k\}} h_{\ell k,i} a_{\ell k} \boldsymbol{\psi}_{\ell,i} \end{cases} \tag{7}$$

with $h_{kk,i} = 1 - \sum_{\ell \in \mathcal{N}_k \setminus \{k\}} h_{\ell k,i} a_{\ell k}$. Note that matrix $\boldsymbol{C}$ in (4) has been set to the identity, and $h_{\ell k,i}$ with $\ell \neq k$ is a binary entry depending on whether agent ℓ has been selected or not by agent k.

Similarly to the RCD, in the distributed LMS with partial diffusion [17]–[19], the matrix $\boldsymbol{C}$ is also set to the identity and the combination step (5) is now defined as:

$$\boldsymbol{w}_{k,i} = a_{kk} \boldsymbol{\psi}_{k,i} + \sum_{\ell \in \mathcal{N}_k \setminus \{k\}} a_{\ell k} \Big(\boldsymbol{H}_{\ell,i} \boldsymbol{\psi}_{\ell,i} + [\boldsymbol{I} - \boldsymbol{H}_{\ell,i}] \boldsymbol{\psi}_{k,i} \Big) \tag{8}$$

where $\boldsymbol{H}_{\ell,i}$ is a diagonal entry-selection matrix with M ones and $L - M$ zeros on its diagonal. This means that the nodes can use the entries of their own intermediate estimates in lieu of the ones from the neighbors that have not been communicated.

Matrix $\boldsymbol{H}_{\ell,i}$ can be deterministic, or can randomly select M entries from all entries.

Finally, the projection approach investigated in compressive diffusion LMS [16] consists of sharing a projection of the local estimates. It also introduces an adaptive correction step to compensate the projection error. This leads to the following formulation of the adaptation step (5):

$$\boldsymbol{w}_{k,i} = a_{kk}\boldsymbol{\psi}_{k,i} + \sum_{\ell\in\mathcal{N}_k\setminus\{k\}} a_{\ell k}\boldsymbol{\gamma}_{\ell,i} \tag{9}$$

where $\boldsymbol{\gamma}_{\ell,i} = \boldsymbol{\gamma}_{\ell,i-1} + \eta_\ell \boldsymbol{p}_{\ell,i}\epsilon_{\ell,i}$ is the constructed estimate, with η_ℓ a positive step-size, $\boldsymbol{p}_{\ell,i}$ a projection vector and $\epsilon_{\ell,i} = \boldsymbol{p}_{\ell,i}^\top(\boldsymbol{\psi}_{\ell,i} - \boldsymbol{\gamma}_{\ell,i})$ the reconstruction error. This approach introduces an additional adaptive step which can increase the algorithm complexity.

None of these methods investigates strategies for reducing the communication load induced by the adaptation step (4) and gradient vector $\hat{\nabla}_w J_\ell(\cdot)$ sharing. The *doubly compressed diffusion LMS* (DCD) devised in this paper addresses this issue by considering both the adaptation step (4) and the combination step (5).

Algorithm 1 Local updates at node k for DCD

loop
  randomly generate $\boldsymbol{H}_{k,i}$ and $\boldsymbol{Q}_{k,i}$
  for $\ell \in \mathcal{N}_k \setminus \{k\}$ **do**
    send $\boldsymbol{H}_{k,i}\boldsymbol{w}_{k,i}$ to node ℓ
    receive from node ℓ the partial gradient vector:

$$\boldsymbol{Q}_{\ell,i}\hat{\nabla}_w J_\ell(\boldsymbol{H}_{k,i}\boldsymbol{w}_{k,i-1} + (\boldsymbol{I}_L - \boldsymbol{H}_{k,i})\boldsymbol{w}_{\ell,i-1})$$

    complete the missing entries using those available at node k, which results in $\boldsymbol{g}_{\ell,i}$ defined in (12)
  update the intermediate estimate:

$$\boldsymbol{\psi}_{k,i} = \boldsymbol{w}_{k,i-1} + \mu_k \sum_{\ell\in\mathcal{N}_k} c_{\ell k}\boldsymbol{g}_{\ell,i}$$

  calculate the local estimate:

$$\boldsymbol{w}_{k,i} = a_{kk}\boldsymbol{\psi}_{k,i} + \sum_{\ell\in\mathcal{N}_k\setminus\{k\}} a_{\ell k}\left[\boldsymbol{H}_{\ell,i}\,\boldsymbol{w}_{\ell,i-1} + (\boldsymbol{I}_L - \boldsymbol{H}_{\ell,i})\,\boldsymbol{\psi}_{k,i}\right]$$

III. Doubly-compressed diffusion LMS

We shall now introduce our DCD method and analyze its stochastic behavior.

The DCD algorithm run at each node k is shown in Alg. 1. Matrices $\boldsymbol{H}_{k,i}$ and $\boldsymbol{Q}_{k,i}$ are diagonal entry-selection matrices with M and M_∇ ones on their diagonal, respectively. The other diagonal entries of these two matrices are set to zero. First, we consider the adaptation step. The matrix $\boldsymbol{H}_{k,i}$ selects M entries (over L) of $\boldsymbol{w}_{k,i-1}$ that are sent to node ℓ to approximate $\nabla_w J_\ell(\boldsymbol{w}_{k,i-1})$ in (4). Node ℓ fills the missing entries of $\boldsymbol{H}_{k,i}\boldsymbol{w}_{k,i-1}$ by using its own entries $(\boldsymbol{I}_L - \boldsymbol{H}_{k,i})\boldsymbol{w}_{\ell,i-1}$, and calculates the instantaneous approximation of the gradient vector at this point. Then node ℓ selects M_∇ entries (over L) of this gradient vector using $\boldsymbol{Q}_{k,i}$ and send them to node k. Node k fills the missing entries by using its own local estimate. Finally, we focus on the combination step. Node k considers the partial parameter vectors $\boldsymbol{H}_{\ell,i}\boldsymbol{w}_{\ell,i-1}$ received from its neighbors ℓ during the adaptation step, and fills the missing entries by using its own local estimate $\boldsymbol{\psi}_{k,i}$. Then it aggregate them to obtain the local estimate $\boldsymbol{w}_{k,i}$.

We can formulate the algorithm in the following form:

$$\boldsymbol{\psi}_{k,i} = \boldsymbol{w}_{k,i-1} + \mu_k \sum_{\ell\in\mathcal{N}_k} c_{\ell k}\boldsymbol{g}_{\ell,i} \tag{10}$$

$$\boldsymbol{w}_{k,i} = a_{kk}\boldsymbol{\psi}_{k,i} + \sum_{\ell\in\mathcal{N}_k\setminus\{k\}} a_{\ell k}\left[\boldsymbol{H}_{\ell,i}\,\boldsymbol{w}_{\ell,i-1} + (\boldsymbol{I}_L - \boldsymbol{H}_{\ell,i})\,\boldsymbol{\psi}_{k,i}\right] \tag{11}$$

with

$$\begin{aligned}\boldsymbol{g}_{\ell,i} &= \boldsymbol{Q}_{\ell,i}\boldsymbol{u}_{\ell,i}\big[d_\ell(i) - \boldsymbol{u}_{\ell,i}^\top(\boldsymbol{H}_{k,i}\boldsymbol{w}_{k,i-1} + (\boldsymbol{I}_L - \boldsymbol{H}_{k,i})\,\boldsymbol{w}_{\ell,i-1})\big] \\ &+ (\boldsymbol{I}_L - \boldsymbol{Q}_{\ell,i})\boldsymbol{u}_{k,i}\big[d_k(i) - \boldsymbol{u}_{k,i}^\top\boldsymbol{w}_{k,i-1})\big]\end{aligned} \tag{12}$$

where $\boldsymbol{H}_{\ell,i} = \mathrm{diag}\{\boldsymbol{h}_{\ell,i}\}$ and $\boldsymbol{Q}_{\ell,i} = \mathrm{diag}\{\boldsymbol{q}_{\ell,i}\}$. In this paper, we shall assume that $\boldsymbol{h}_{\ell,i}$ (resp., $\boldsymbol{q}_{\ell,i}$) is an $L\times 1$ binary vector, generated by randomly setting M (resp., M_∇) of its L entries to 1, and the other $L-M$ (resp., $L-M_\nabla$) entries to 0. We shall assume that all possible outcomes for $\boldsymbol{h}_{\ell,i}$ (resp., $\boldsymbol{q}_{\ell,i}$) are equally likely, and i.i.d. over time and space. Then,

$$\mathbb{E}\{\boldsymbol{H}_{\ell,i}\} = \tfrac{M}{L}\boldsymbol{I}_L \qquad \mathbb{E}\{\boldsymbol{Q}_{\ell,i}\} = \tfrac{M_\nabla}{L}\boldsymbol{I}_L \tag{13}$$

We shall now analyze the stochastic behavior of the DCD algorithm. For the sake of simplicity, we shall consider that matrix $\boldsymbol{C}$ is doubly stochastic. We shall also set matrix $\boldsymbol{A}$ to the identity matrix. Focusing in this way on the adaptation step and gradient vector sharing helps simplify the analysis. Note that the distributed LMS with partial diffusion (8), which exclusively addresses how reducing the communication load induced by the combination step, was analyzed in [17]. Combining both analyses into a single general one is challenging and beyond of the scope of this paper. However, in the sequel, we shall illustrate the efficiency of the DCD algorithm in both cases $\boldsymbol{A} = \boldsymbol{I}_L$ and $\boldsymbol{A} \neq \boldsymbol{I}_L$, and compare it with the existing strategies.

Before proceeding with the algorithm analysis, let us introduce the following assumptions on the regression data and selection matrices.

Assumption 1 The regression vectors $\boldsymbol{u}_{k,i}$ arise from a zero-mean random process that is temporally white and spatially independent. A direct consequence of this assumption is that $\boldsymbol{u}_{k,i}$ is independent of $\boldsymbol{w}_{\ell,j}$ for all ℓ and $j < i$.

Assumption 2 The matrices $\boldsymbol{H}_{i,k}$ and $\boldsymbol{Q}_{\ell,i}$ arise from a random process that is temporally white, spatially independent, and independent of each other as well as any other process.

We introduce the $L \times 1$ error vectors:

$$\widetilde{\boldsymbol{w}}_{k,i} = \boldsymbol{w}^o - \boldsymbol{w}_{k,i} \tag{14}$$

and we collect them from across all nodes into the vectors:

$$\widetilde{\boldsymbol{w}}_i = \text{col}\{\widetilde{\boldsymbol{w}}_{1,i}, \widetilde{\boldsymbol{w}}_{2,i}, \ldots, \widetilde{\boldsymbol{w}}_{N,i}\} \tag{15}$$

Let $\boldsymbol{R}_{u_\ell,i} = \boldsymbol{u}_{\ell,i}\boldsymbol{u}_{\ell,i}^\top$. We also introduce:

$$\boldsymbol{\mathcal{M}} = \text{diag}\{\mu_1\boldsymbol{I}_L, \mu_2\boldsymbol{I}_L, \ldots, \mu_N\boldsymbol{I}_L\} \tag{16}$$

$$\boldsymbol{\mathcal{R}}_{Q,i} = \text{diag}\Big\{\sum_{\ell\in\mathcal{N}_1} c_{\ell 1}\boldsymbol{Q}_{\ell,i}\boldsymbol{R}_{u_\ell,i}, \sum_{\ell\in\mathcal{N}_2} c_{\ell 2}\boldsymbol{Q}_{\ell,i}\boldsymbol{R}_{u_\ell,i}, \ldots, \sum_{\ell\in\mathcal{N}_N} c_{\ell N}\boldsymbol{Q}_{\ell,i}\boldsymbol{R}_{u_\ell,i}\Big\} \tag{17}$$

$$\boldsymbol{\mathcal{H}}_i = \text{diag}\{\boldsymbol{H}_{1,i}, \boldsymbol{H}_{2,i}, \ldots, \boldsymbol{H}_{N,i}\} \tag{18}$$

$$\boldsymbol{\mathcal{Q}}'_i = \text{diag}\Big\{\sum_{\ell\in\mathcal{N}_1} c_{\ell 1}(\boldsymbol{I}_L - \boldsymbol{Q}_{\ell,i}), \sum_{\ell\in\mathcal{N}_2} c_{\ell 1}(\boldsymbol{I}_L - \boldsymbol{Q}_{\ell,i})\ldots, \sum_{\ell\in\mathcal{N}_N} c_{\ell N}(\boldsymbol{I}_L - \boldsymbol{Q}_{\ell,i})\Big\} \tag{19}$$

$$\boldsymbol{\mathcal{R}}_{u,i} = \text{diag}\{\boldsymbol{R}_{u_1,i}, \boldsymbol{R}_{u_2,i}, \ldots, \boldsymbol{R}_{u_N,i}\} \tag{20}$$

$$\boldsymbol{\mathcal{Q}}_i = \text{diag}\{\boldsymbol{Q}_{1,i}, \boldsymbol{Q}_{2,i}, \ldots, \boldsymbol{Q}_{N,i}\} \tag{21}$$

$$\boldsymbol{\mathcal{C}} = \boldsymbol{C}\otimes\boldsymbol{I}_L \tag{22}$$

where $\otimes$ denotes the Kronecker product. Finally, we introduce the $N\times N$ block matrix $\boldsymbol{\mathcal{R}}_{Q(I-H),i}$ with each block (k,ℓ) defined as:

$$[\boldsymbol{\mathcal{R}}_{Q(I-H),i}]_{k\ell} = c_{\ell k}\boldsymbol{Q}_{\ell,i}\boldsymbol{R}_{u_\ell,i}(\boldsymbol{I}_L - \boldsymbol{H}_{k,i}) \tag{23}$$

Combining recursion (11) and definition (14), and replacing $d_k(i)$ by its definition (1), we find:

$$\begin{aligned}\widetilde{\boldsymbol{w}}_{k,i} = \widetilde{\boldsymbol{w}}_{k,i-1} &- \mu_k\sum_{\ell\in\mathcal{N}_k} c_{\ell k}\boldsymbol{Q}_{\ell,i}\boldsymbol{u}_{\ell,i}\big[\boldsymbol{u}_{\ell,i}^\top\boldsymbol{w}^o + v_\ell(i) \\ &\qquad - \boldsymbol{u}_{\ell,i}^\top(\boldsymbol{H}_{k,i}\boldsymbol{w}_{k,i-1} + (\boldsymbol{I}_L - \boldsymbol{H}_{k,i})\boldsymbol{w}_{\ell,i-1})\big] \\ &- \mu_k\sum_{\ell\in\mathcal{N}_k} c_{\ell k}(\boldsymbol{I}_L - \boldsymbol{Q}_{\ell,i})\boldsymbol{u}_{k,i}\big[\boldsymbol{u}_{k,i}^\top\boldsymbol{w}^o + v_k(i) \\ &\qquad - \boldsymbol{u}_{k,i}^\top\boldsymbol{w}_{k,i-1}\big]\end{aligned} \tag{24}$$

Note that $\boldsymbol{w}^o = \boldsymbol{H}_{k,i}\boldsymbol{w}^o + (\boldsymbol{I}_L - \boldsymbol{H}_{k,i})\boldsymbol{w}^o$. Replacing $\boldsymbol{w}^o$ by this expression, and using definition (14), leads to:

$$\begin{aligned}\widetilde{\boldsymbol{w}}_{k,i} = \widetilde{\boldsymbol{w}}_{k,i-1} &- \mu_k\sum_{\ell\in\mathcal{N}_k} c_{\ell k}\boldsymbol{Q}_{\ell,i}\boldsymbol{u}_{\ell,i}\big[\boldsymbol{u}_{\ell,i}^\top\boldsymbol{H}_{k,i}\widetilde{\boldsymbol{w}}_{k,i-1} \\ &+ \boldsymbol{u}_{\ell,i}^\top(\boldsymbol{I}_L - \boldsymbol{H}_{k,i})\widetilde{\boldsymbol{w}}_{\ell,i-1} + v_\ell(i)\big] \\ &- \mu_k\sum_{\ell\in\mathcal{N}_k} c_{\ell k}(\boldsymbol{I}_L - \boldsymbol{Q}_{\ell,i})\boldsymbol{u}_{k,i}\big[\boldsymbol{u}_{k,i}^\top\widetilde{\boldsymbol{w}}_{k,i-1} + v_k(i)\big]\end{aligned} \tag{25}$$

Rearranging the terms in (25), and using definitions (15)–(22), leads to:

$$\begin{aligned}\widetilde{\boldsymbol{w}}_i = \big(&\boldsymbol{I}_{NL} - \boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{R}}_{Q,i}\boldsymbol{\mathcal{H}}_i - \boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{Q}}'_i\boldsymbol{\mathcal{R}}_{u,i} \\ &- \boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{R}}_{Q(I-H),i}\big)\widetilde{\boldsymbol{w}}_{i-1} - \big(\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{C}}^\top\boldsymbol{\mathcal{Q}}_i + \boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{Q}}'_i\big)\boldsymbol{s}_i\end{aligned} \tag{26}$$

where

$$\boldsymbol{s}_i = \text{col}\{\boldsymbol{u}_{1,i}v_1(i), \boldsymbol{u}_{2,i}v_2(i), \ldots, \boldsymbol{u}_{N,i}v_N(i)\} \tag{27}$$

A. Mean weight behavior analysis

We shall now examine the convergence in the mean for the DCD algorithm and derive a necessary convergence condition. We start by rewriting the weight-error vector recursion (26) as:

$$\widetilde{\boldsymbol{w}}_i = \boldsymbol{\mathcal{B}}_i\widetilde{\boldsymbol{w}}_{i-1} - \boldsymbol{\mathcal{G}}_i\boldsymbol{s}_i \tag{28}$$

where the coefficient matrices $\boldsymbol{\mathcal{B}}_i$ and $\boldsymbol{\mathcal{G}}_i$ are defined as:

$$\begin{aligned}\boldsymbol{\mathcal{B}}_i = &\boldsymbol{I}_{NL} \\ &- \boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{R}}_{Q,i}\boldsymbol{\mathcal{H}}_i - \boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{Q}}'_i\boldsymbol{\mathcal{R}}_{u,i} - \boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{R}}_{Q(I-H),i}\end{aligned} \tag{29}$$

$$\boldsymbol{\mathcal{G}}_i = \boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{C}}^\top\boldsymbol{\mathcal{Q}}_i + \boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{Q}}'_i \tag{30}$$

Taking expectations of both sides of (28), using Assumptions 1 and 2, and $\mathbb{E}\{\boldsymbol{s}_i\} = 0$, we find:

$$\mathbb{E}\{\widetilde{\boldsymbol{w}}_i\} = \boldsymbol{\mathcal{B}}\,\mathbb{E}\{\widetilde{\boldsymbol{w}}_{i-1}\}$$

where

$$\begin{aligned}\boldsymbol{\mathcal{B}} = \boldsymbol{I}_{NL} &- \tfrac{MM_\nabla}{L^2}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{R}} - \big(1 - \tfrac{M_\nabla}{L}\big)\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{R}}_u \\ &- \tfrac{M_\nabla}{L}\big(1 - \tfrac{M}{L}\big)\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{C}}^\top\boldsymbol{\mathcal{R}}_u\end{aligned} \tag{31}$$

$$\boldsymbol{\mathcal{R}}_u = \mathbb{E}\{\boldsymbol{\mathcal{R}}_{u,i}\} = \text{diag}\{\boldsymbol{R}_{u_1}, \boldsymbol{R}_{u_2}, \ldots, \boldsymbol{R}_{u_N}\} \tag{32}$$

$$\boldsymbol{\mathcal{R}} = \text{diag}\{\boldsymbol{R}_1, \ldots, \boldsymbol{R}_N\} \tag{33}$$

with

$$\boldsymbol{R}_k = \sum_{\ell\in\mathcal{N}_k} c_{\ell,k}\boldsymbol{R}_{u_\ell} \tag{34}$$

From (31), we observe that the algorithm (11) asymptotically converges in the mean toward $\boldsymbol{w}^o$ if, and only if,

$$\rho(\boldsymbol{\mathcal{B}}) < 1 \tag{35}$$

where $\rho(\cdot)$ denotes the spectral radius of its matrix argument. We know that $\rho(\boldsymbol{X}) \le \|\boldsymbol{X}\|$ for any induced norm. Then:

$$\begin{aligned}\rho(\boldsymbol{\mathcal{B}}) &\le \|\boldsymbol{\mathcal{B}}\|_{b,\infty} \\ &\le \max_{k,\ell}\|[\boldsymbol{\mathcal{B}}]_{k\ell}\|\end{aligned} \tag{36}$$

where $\|\cdot\|_{b,\infty}$ denotes the block maximum norm. From (36) and by Weyl's theorem we have:

$$\mu_k < \frac{2}{\lambda_{\max,k}} \tag{37}$$

where

$$\begin{aligned}\lambda_{\max,k} = &\tfrac{MM_\nabla}{L^2}\lambda_{\max}(\boldsymbol{R}_k) + \tfrac{M}{L}\big(1 - \tfrac{M_\nabla}{L}\big)\lambda_{\max}(\boldsymbol{R}_{u_k}) \\ &+ \tfrac{M_\nabla}{L}\big(1 - \tfrac{M}{L}\big)\max_{\ell\in\boldsymbol{\mathcal{N}}_k} c_{\ell k}\,\lambda_{\max}(\boldsymbol{R}_{u_\ell})\end{aligned} \tag{38}$$

and $\lambda_{\max}(\cdot)$ stands for the maximum eigenvalue of its matrix argument.

Note that when $M = M_\nabla = L$, we recover the matrices $\boldsymbol{\mathcal{B}}_i$ and $\boldsymbol{\mathcal{G}}$ of the diffusion LMS in [1, (262)–(263)].

B. Mean-square error behavior analysis

We are now interested in providing a global solution for studying the mean square error. With this aim, we consider the weighted square measure $\mathbb{E}\{\|\widetilde{\boldsymbol{w}}_i\|^2_{\boldsymbol{\Sigma}}\}$ where $\boldsymbol{\Sigma}$ denotes a $N \times N$ block diagonal weighting matrix. By setting $\boldsymbol{\Sigma}$ to different values, we can extract various types of information about the nodes and the network such as the network mean square deviation MSD, or the excess mean square error EMSE.

We start by using the independence Assumption 1 and (28) to write

$$\mathbb{E}\|\widetilde{\boldsymbol{w}}_i\|^2_{\boldsymbol{\Sigma}} = \mathbb{E}\{\widetilde{\boldsymbol{w}}_{i-1}^{\top}\boldsymbol{\mathcal{B}}_i^{\top}\boldsymbol{\Sigma}\boldsymbol{\mathcal{B}}_i\widetilde{\boldsymbol{w}}_{i-1}\} + \mathbb{E}\{\boldsymbol{s}_i^{\top}\boldsymbol{\mathcal{G}}_i^{\top}\boldsymbol{\Sigma}\,\boldsymbol{\mathcal{G}}_i\boldsymbol{s}_i\} \quad (39)$$

On the one hand, the second term on the RHS of (39) can be written as:

$$\begin{aligned}\mathbb{E}\{\boldsymbol{s}_i^{\top}\boldsymbol{\mathcal{G}}_i^{\top}\boldsymbol{\Sigma}\,\boldsymbol{\mathcal{G}}_i\boldsymbol{s}_i\} &= \text{trace}\big(\mathbb{E}\{\boldsymbol{s}_i^{\top}\boldsymbol{\mathcal{G}}_i^{\top}\boldsymbol{\Sigma}\,\boldsymbol{\mathcal{G}}_i\boldsymbol{s}_i\}\big)\\ &= \text{trace}\big(\mathbb{E}\{\boldsymbol{\mathcal{G}}_i^{\top}\boldsymbol{\Sigma}\,\boldsymbol{\mathcal{G}}_i\}\,\mathbb{E}\{\boldsymbol{s}_i\boldsymbol{s}_i^{\top}\}\big)\\ &= \text{trace}\big(\mathbb{E}\{\boldsymbol{\mathcal{G}}_i^{\top}\boldsymbol{\Sigma}\,\boldsymbol{\mathcal{G}}_i\}\,\boldsymbol{\mathcal{S}}\big) \end{aligned} \quad (40)$$

where

$$\boldsymbol{\mathcal{S}} = \mathbb{E}\{\boldsymbol{s}_i\boldsymbol{s}_i^{\top}\} = \text{diag}(\sigma^2_{v,1}\boldsymbol{R}_{u,1},\ldots,\sigma^2_{v,N}\boldsymbol{R}_{u,N}) \quad (41)$$

and

$$\begin{aligned}&\mathbb{E}\{\boldsymbol{\mathcal{G}}_i^{\top}\boldsymbol{\Sigma}\,\boldsymbol{\mathcal{G}}_i\}\\ &= \big(\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{C}}^{\top}\boldsymbol{\mathcal{Q}}_i + \boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{Q}}_i'\big)^{\top}\boldsymbol{\Sigma}\big(\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{C}}^{\top}\boldsymbol{\mathcal{Q}}_i + \boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{Q}}_i'\big)\\ &= \boldsymbol{\Theta}_1 + \boldsymbol{\Theta}_2 + \boldsymbol{\Theta}_2^{\top} + \boldsymbol{\Theta}_3 \end{aligned} \quad (42)$$

with

$$\boldsymbol{\Theta}_1 = \mathbb{E}\{\boldsymbol{\mathcal{Q}}_i\boldsymbol{\mathcal{C}}\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{C}}^{\top}\boldsymbol{\mathcal{Q}}_i\} \quad (43)$$

$$\boldsymbol{\Theta}_2 = \mathbb{E}\{\boldsymbol{\mathcal{Q}}_i\boldsymbol{\mathcal{C}}\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{Q}}_i'\} \quad (44)$$

$$\boldsymbol{\Theta}_3 = \mathbb{E}\{\boldsymbol{\mathcal{Q}}_i'\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{Q}}_i'\} \quad (45)$$

Before proceeding with the calculation of (43)–(45), we introduce some preliminary results.

Given any $L \times L$ matrix $\boldsymbol{\Sigma}$, it can be shown:

$$\mathbb{E}\{\boldsymbol{Q}_{\ell,i}\boldsymbol{\Sigma}\boldsymbol{Q}_{k,i}\} = \quad (46)$$
$$\begin{cases} \frac{M_\nabla}{L}\left(\left(1-\frac{M_\nabla-1}{L-1}\right)\boldsymbol{I}_L\odot\boldsymbol{\Sigma} + \frac{M_\nabla-1}{L-1}\boldsymbol{\Sigma}\right) & \text{if } \ell = k \\ \left(\frac{M_\nabla}{L}\right)^2\boldsymbol{\Sigma} & \text{otherwise} \end{cases}$$

where $\odot$ is the Hadamard entry-wise product.

Consider the block diagonal matrix $\boldsymbol{\mathcal{Q}}_i$ and any $NL \times NL$ matrix $\boldsymbol{\Pi}$. By using (46) for each block $\mathbb{E}\{[\boldsymbol{\mathcal{Q}}_i\boldsymbol{\Pi}\boldsymbol{\mathcal{Q}}_i]_{k\ell}\}$, it follows that:

$$\begin{aligned}&\mathbb{E}\{\boldsymbol{\mathcal{Q}}_i\boldsymbol{\Pi}\boldsymbol{\mathcal{Q}}_i\}\\ &= \alpha_1\,(\boldsymbol{I}_N\otimes\boldsymbol{1}_{LL})\odot\boldsymbol{\Pi} + \alpha_2\,\boldsymbol{I}_{NL}\odot\boldsymbol{\Pi} + \alpha_3\,\boldsymbol{\Pi}\end{aligned} \quad (47)$$

where $\boldsymbol{1}_{LL}$ denotes the all-one $L \times L$ matrix, and:

$$\alpha_1 = \frac{M_\nabla}{L}\left(\frac{M_\nabla-1}{L-1} - \frac{M_\nabla}{L}\right) \quad (48)$$

$$\alpha_2 = \frac{M_\nabla}{L}\left(1-\frac{M_\nabla-1}{L-1}\right) \quad (49)$$

$$\alpha_3 = \left(\frac{M_\nabla}{L}\right)^2 \quad (50)$$

Finally we consider the $NL \times NL$ matrix, say $\boldsymbol{\varphi}_Q(\boldsymbol{\Pi})$, defined by its $L \times L$ blocks:

$$[\boldsymbol{\varphi}_Q(\boldsymbol{\Pi})]_{k\ell} = \mathbb{E}\{\boldsymbol{Q}_{k,i}[\boldsymbol{\Pi}]_{k\ell}\boldsymbol{Q}_{k,i}\} \quad (51)$$

By using (46) for each block, it can be shown that $\boldsymbol{\varphi}_Q(\boldsymbol{\Pi})$ can be expressed as follows:

$$\boldsymbol{\varphi}_Q(\boldsymbol{\Pi}) = \alpha_2\,(\boldsymbol{1}_{NN}\otimes\boldsymbol{I}_L)\odot\boldsymbol{\Pi} + (\alpha_1+\alpha_3)\,\boldsymbol{\Pi} \quad (52)$$

Note that $\boldsymbol{\varphi}_Q(\boldsymbol{\Pi}) = \mathbb{E}\{\boldsymbol{\mathcal{Q}}_i\boldsymbol{\Pi}\boldsymbol{\mathcal{Q}}_i\}$ if $\boldsymbol{\Pi}$ is block diagonal.

We can now proceed with the evaluation of (43)–(45). Matrix $\boldsymbol{\Theta}_1$ calculation follows by setting $\boldsymbol{\Pi} = \boldsymbol{\mathcal{C}}\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{C}}^{\top}$ in (52). Consider now $\boldsymbol{\Theta}_2$ in (42). We have:

$$\begin{aligned}[\boldsymbol{\Theta}_2]_{k\ell} = {}& \tfrac{M_\nabla}{L}[\boldsymbol{\mathcal{C}}\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}]_{k\ell} - c_{k\ell}^2\,\mathbb{E}\{\boldsymbol{Q}_{k,i}[\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}]_{\ell\ell}\boldsymbol{Q}_{k,i}\}\\ &- \left(\tfrac{M_\nabla}{L}\right)^2\left([\boldsymbol{\mathcal{C}}\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}]_{k\ell} - c_{k\ell}^2\,[\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}]_{\ell\ell}\right)\end{aligned} \quad (53)$$

We can use (51) to calculate the second term in the RHS of the above equation since $\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}$ is block diagonal. This yields:

$$\begin{aligned}\boldsymbol{\Theta}_2 = {}& \tfrac{M_\nabla}{L}\boldsymbol{\mathcal{C}}\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}} - \boldsymbol{\mathcal{C}}_2\boldsymbol{\varphi}_Q(\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}})\\ &- \left(\tfrac{M_\nabla}{L}\right)^2(\boldsymbol{\mathcal{C}}\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}} - \boldsymbol{\mathcal{C}}_2\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}})\end{aligned} \quad (54)$$

where $\boldsymbol{\mathcal{C}}_2 = \boldsymbol{\mathcal{C}}\odot\boldsymbol{\mathcal{C}}$.

Finally, we calculate the last term $\boldsymbol{\Theta}_3$ in the RHS of (42). Matrix $\boldsymbol{\Theta}_3$ is block diagonal, with each diagonal block defined as follows:

$$\begin{aligned}&[\boldsymbol{\Theta}_3]_{kk}\\ &= \mathbb{E}\{[\boldsymbol{\mathcal{Q}}_i']_{kk}[\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}]_{kk}[\boldsymbol{\mathcal{Q}}_i']_{kk}\}\\ &= \sum_{m,n=1}^{N} c_{mk}c_{nk}\,\mathbb{E}\{\left(\boldsymbol{I}_L-\boldsymbol{Q}_{m,i}\right)[\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}]_{kk}\left(\boldsymbol{I}_L-\boldsymbol{Q}_{n,i}\right)\}\end{aligned}$$

Using (13), we get:

$$\begin{aligned}[\boldsymbol{\Theta}_3]_{kk} = {}& \left(1-2\tfrac{M_\nabla}{L}\right)[\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}]_{kk}\sum_{m,n=1}^{N}c_{mk}c_{nk}\\ &+ \sum_{m,n=1}^{N}c_{mk}c_{nk}\mathbb{E}\{\boldsymbol{Q}_{m,i}[\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}]_{kk}\boldsymbol{Q}_{n,i}\}\end{aligned}$$

Applying (46) leads to:

$$\begin{aligned}[\boldsymbol{\Theta}_3]_{kk} = {}& \left(1-2\tfrac{M_\nabla}{L}\right)[\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}]_{kk}\\ &+ \sum_{m}^{N}c_{mk}^2\mathbb{E}\{\boldsymbol{Q}_{m,i}[\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}]_{kk}\boldsymbol{Q}_{m,i}\}\\ &+ \left(\tfrac{M_\nabla}{L}\right)^2\left(\sum_{m,n=1}^{N}c_{mk}c_{nk}[\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}]_{kk}\right.\\ &\left.- \sum_{m=1}^{N}c_{mk}^2[\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}]_{kk}\right)\end{aligned} \quad (55)$$

Finally, using (52), we can write (55) in a compact form:

$$\begin{aligned}\boldsymbol{\Theta}_3 = {}& \left(1-2\tfrac{M_\nabla}{L}\right)\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}} + (\boldsymbol{I}_{NL}\odot\boldsymbol{\mathcal{C}}\boldsymbol{\mathcal{C}}^{\top})\,\boldsymbol{\varphi}_Q(\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}})\\ &+ \left(\tfrac{M_\nabla}{L}\right)^2\left(\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}} - (\boldsymbol{I}_{NL}\odot\boldsymbol{\mathcal{C}}\boldsymbol{\mathcal{C}}^{\top})\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}\right)\end{aligned} \quad (56)$$

It is interesting to notice that the second term in the RHS of (39) does not depend on M. Moreover, setting $M_\nabla = L$ results in canceling $\boldsymbol{\Theta}_2$ and $\boldsymbol{\Theta}_3$ since $\boldsymbol{\mathcal{Q}}_i' = \mathbf{0}$.

On the other hand, the first term on the RHS of (39) depends on both parameters M and M_∇ and can be expressed as:

$$\mathbb{E}\{\widetilde{\boldsymbol{w}}_{i-1}^\top \boldsymbol{\mathcal{B}}_i^\top \boldsymbol{\Sigma} \boldsymbol{\mathcal{B}}_i \widetilde{\boldsymbol{w}}_{i-1}\} = \mathbb{E}\|\widetilde{\boldsymbol{w}}_{i-1}\|^2_{\boldsymbol{\Sigma}'} \tag{57}$$

where the weighting matrix $\boldsymbol{\Sigma}'$ is defined as:

$$\boldsymbol{\Sigma}' = \mathbb{E}\{\boldsymbol{\mathcal{B}}_i^\top \boldsymbol{\Sigma} \boldsymbol{\mathcal{B}}_i\} \tag{58}$$

Replacing $\boldsymbol{\mathcal{B}}_i$ by its definition (29) leads to:

$$\begin{aligned}\boldsymbol{\Sigma}' = \boldsymbol{\Sigma} &- \tfrac{MM_\nabla}{L^2}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{R}} - \left(1-\tfrac{M_\nabla}{L}\right)\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{R}}_u \\ &- \tfrac{M_\nabla}{L}\left(1-\tfrac{M}{L}\right)\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{C}}^\top\boldsymbol{\mathcal{R}}_u \\ &- \tfrac{MM_\nabla}{L^2}\boldsymbol{\mathcal{R}}\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma} - \left(1-\tfrac{M_\nabla}{L}\right)\boldsymbol{\mathcal{R}}_u\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma} \\ &- \tfrac{M_\nabla}{L}\left(1-\tfrac{M}{L}\right)\boldsymbol{\mathcal{R}}_u\boldsymbol{\mathcal{C}}^\top\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma} \\ &+ \sum_{j=1}^{6}\boldsymbol{P}_j + \boldsymbol{P}_2^\top + \boldsymbol{P}_3^\top + \boldsymbol{P}_5^\top\end{aligned}$$

where

$$\boldsymbol{P}_1 = \mathbb{E}\{\boldsymbol{\mathcal{H}}_i\boldsymbol{\mathcal{R}}_{Q,i}^\top\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{R}}_{Q,i}\boldsymbol{\mathcal{H}}_i\} \tag{59}$$

$$\boldsymbol{P}_2 = \mathbb{E}\{\boldsymbol{\mathcal{H}}_i\boldsymbol{\mathcal{R}}_{Q,i}^\top\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{Q}}_i'\boldsymbol{\mathcal{R}}_{u,i}\} \tag{60}$$

$$\boldsymbol{P}_3 = \mathbb{E}\{\boldsymbol{\mathcal{H}}_i\boldsymbol{\mathcal{R}}_{Q,i}^\top\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{R}}_{Q(I-H),i}\} \tag{61}$$

$$\boldsymbol{P}_4 = \mathbb{E}\{\boldsymbol{\mathcal{R}}_{u,i}^\top\boldsymbol{\mathcal{Q}}_i'\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{Q}}_i'\boldsymbol{\mathcal{R}}_{u,i}\} \tag{62}$$

$$\boldsymbol{P}_5 = \mathbb{E}\{\boldsymbol{\mathcal{R}}_{u,i}^\top\boldsymbol{\mathcal{Q}}_i'\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{R}}_{Q(I-H),i}\} \tag{63}$$

$$\boldsymbol{P}_6 = \mathbb{E}\{\boldsymbol{\mathcal{R}}_{Q(I-H),i}^\top\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{R}}_{Q(I-H),i}\} \tag{64}$$

Due to the complexity for calculating the terms $\boldsymbol{P}_j$ and the outcomes, we shall not report them in this section. Instead, we provide all the necessary steps and results in the Appendix.

Following the same reasoning as in [1], we express $\boldsymbol{\Sigma}'$ in a vector form as:

$$\boldsymbol{\sigma}' = \boldsymbol{\mathcal{F}}\boldsymbol{\sigma} \tag{65}$$

where

$$\boldsymbol{\sigma} = \mathrm{vec}(\boldsymbol{\Sigma}) \qquad \boldsymbol{\sigma}' = \mathrm{vec}(\boldsymbol{\Sigma}')$$

and the coefficient matrix $\boldsymbol{\mathcal{F}}$ of size $(MN)^2 \times (MN)^2$ is defined as:

$$\begin{aligned}\boldsymbol{\mathcal{F}} = \boldsymbol{I}_{(NM)^2} &- \tfrac{MM_\nabla}{L^2}\boldsymbol{\mathcal{R}}\boldsymbol{\mathcal{M}}\otimes\boldsymbol{I}_{LN} \\ &- \left(1-\tfrac{M_\nabla}{L}\right)\boldsymbol{\mathcal{R}}_u\boldsymbol{\mathcal{M}}\otimes\boldsymbol{I}_{LN} \\ &- \tfrac{M_\nabla}{L}\left(1-\tfrac{M}{L}\right)\boldsymbol{\mathcal{R}}_u\boldsymbol{\mathcal{C}}\boldsymbol{\mathcal{M}}\otimes\boldsymbol{I}_{LN} - \tfrac{MM_\nabla}{L^2}\boldsymbol{I}_{LN}\otimes\boldsymbol{\mathcal{R}}\boldsymbol{\mathcal{M}} \\ &- \left(1-\tfrac{M_\nabla}{L}\right)\boldsymbol{I}_{LN}\otimes\boldsymbol{\mathcal{R}}_u\boldsymbol{\mathcal{M}} \\ &- \tfrac{M_\nabla}{L}\left(1-\tfrac{M}{L}\right)\boldsymbol{I}_{LN}\otimes\boldsymbol{\mathcal{R}}_u\boldsymbol{\mathcal{C}}^\top\boldsymbol{\mathcal{M}} \\ &+ \sum_{j=1}^{6}\boldsymbol{Z}_j + \boldsymbol{Z}_{2^\top} + \boldsymbol{Z}_{3^\top} + \boldsymbol{Z}_{5^\top}\end{aligned} \tag{66}$$

where the matrices $\boldsymbol{Z}_j$ and $\boldsymbol{Z}_{j^\top}$ are obtained when applying the vec$(\cdot)$ operator to $\boldsymbol{P}_j$ and $\boldsymbol{P}_j^\top$, respectively, as it is shown in the Appendix.

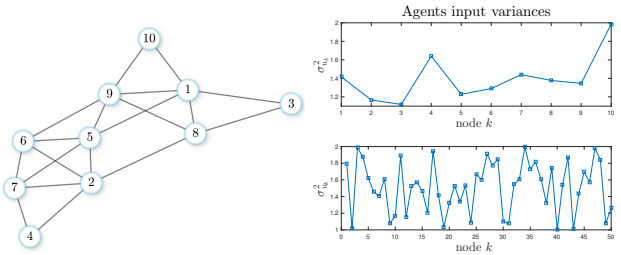

Fig. 2: (left) Network topology. (right) Variance $\sigma^2_{u_k}$ of regressors in Experiment 1 (top) and Experiment 2 (Bottom).

Substituting (40) and (57) into (39), and applying the vec$(\cdot)$ operator to both sides, we get:

$$\mathbb{E}\|\widetilde{\boldsymbol{w}}_i\|^2_{\boldsymbol{\sigma}} = \mathbb{E}\|\widetilde{\boldsymbol{w}}_{i-1}\|^2_{\boldsymbol{\mathcal{F}}\boldsymbol{\sigma}} + \mathrm{trace}\big(\mathbb{E}\{\boldsymbol{\mathcal{G}}^\top\boldsymbol{\Sigma}\,\boldsymbol{\mathcal{G}}\}\,\boldsymbol{\mathcal{S}}\big) \tag{67}$$

Using (67), it is possible to extract useful information about the network or a specific node. For instance, we calculate the network mean square deviation or excess mean square error by setting $\boldsymbol{\Sigma} = \boldsymbol{I}_{LN}$ and $\boldsymbol{\Sigma} = \boldsymbol{\mathcal{R}}_u$, respectively. The DCD can be seen as an extension of the diffusion LMS in the case where the weighting matrix $\boldsymbol{A}$ is the identity matrix. Indeed, it is possible to recover the diffusion LMS, and derive other variants such as the compressed diffusion LMS, by properly setting matrices $\{\boldsymbol{H}_{k,i}, \boldsymbol{Q}_{k,i}\}$ and parameters $\{M, M_\nabla\}$.

IV. Simulation Results

In this section, we shall first evaluate the accuracy of the mean-square error behavior model. Then, we shall perform two experiments to characterize the performance of the DCD algorithm compared to the diffusion LMS algorithm, the reduced-communication diffusion LMS [5], and the partial diffusion LMS [18]. We shall also consider the so-called compressed diffusion LMS (CD) obtained by setting $\boldsymbol{A} = \boldsymbol{I}_L$ and $\boldsymbol{Q}_{\ell,i} = \boldsymbol{I}_L$ in (10)–(11), which means that $M_\nabla = L$ in this case. Before proceeding, note that the compression ratios of the CD and DCD algorithms are equal to $\frac{2L}{M+L}$ and $\frac{2L}{M+M_\nabla}$.

First we considered a small network to validate the theoretical model. Then, we used a larger network and high dimensional measurements with the aim of testing the algorithm in a larger scale setting. Finally, we considered an energy dependent network where the agents alternate between active and inactive states depending on the available energy. For the three experiments, the parameter vectors $\boldsymbol{w}^o$ were generated from a zero-mean Gaussian distribution. The input data $\boldsymbol{u}_{k,i}$ were drawn from zero-mean Gaussian distributions, with covariance $\boldsymbol{R}_{u,k} = \sigma^2_{u,k}\boldsymbol{I}_L$ reported in Fig. 2 (right). The weighting matrices $\boldsymbol{C}$ were generated using the Metropolis rule [1]. Noises $v_k(i)$ were zero-mean, i.i.d. and Gaussian distributed with variance $\sigma^2_{v,k} = 10^{-3}$. Simulation results were averaged over 100 Monte-Carlo runs.

1) Experiment 1: We considered the network with $N = 10$ nodes depicted in Fig. 2 (left). We set the parameters as follows: $\mu_k = 10^{-3}$, $L = 5$, $M = 3$, $M_\nabla = 1$. This resulted in compression ratio of $\frac{10}{8}$ and $\frac{10}{4}$ for compressed diffusion and doubly compressed diffusion LMS, respectively. It can be

observed in Fig. 3 (left) that the theoretical model accurately fits the simulated results. Unsurprisingly, the diffusion LMS algorithm outperformed its compressed counterparts at the expense of a higher communication load.

2) Experiment 2: Since compression is particularly relevant for relatively large data flows, then we considered a network with $N = 50$ agents. We set the algorithm parameters as follows: $\mu_k = 3 \cdot 10^{-2}$, $L = 50$, $M = 5$. Due to the high dimensionality of the matrix $\boldsymbol{\mathcal{F}}$ ($2500^2 \times 2500^2$), we only performed Monte-Carlo simulations using C language scripts. Figure 3 depicts the performance of the algorithms for different compression ratios. The largest compression ratio that can be reached by the CD algorithm equals $\frac{100}{55}$ as it transmits the whole gradient vectors ($\boldsymbol{Q}_{\ell,i} = \boldsymbol{I}_L$). On the other hand, the CDC diffusion LMS offers more flexibility and can adapt to the network communication load by adjusting M and M_∇.

3) Experiment 3: In a realistic wireless sensor network (WSN) implementation, nodes have limited energy reserves and cannot be active all the time. One of the most promising solution for this issue is to adopt an ENO strategy, where ENO stands for Energy Neutral Operation. In other words, the agents consume at most the amount of energy they harvest, hence achieving the neutral energy condition. Theoretically, neutral energy condition guarantees an infinite sensor lifetime. In order to implement an ENO strategy, nodes must be endowed with energy harvesting and storage capabilities. Agents alternate between two phases: a brief active phase and a sleeping phase. During the active phase, each agent k performs its assigned tasks and calculates the duration $T_{s_k,i}$ of the sleeping phase based on the available energy, the consumed energy and an estimate of the energy that will be harvested [21]. For the sake of limiting energy consumption, the agents then switch to sleep mode for a duration of $T_{s_k,i}$. The corresponding DCD based algorithm is presented in Alg. 2.

We considered a solar energy based WSN with Bluetooth capabilities. To calculate $T_{s_k,i}$, we used [21]:

$$T_{s_k,i} = \frac{e_{c_k,i} - \eta\, e_{s_k,i}}{\eta\,(P_{\text{harv},k,i} - P_{\text{leak}}) - P_{\text{sleep}}} \tag{68}$$

where $e_{c_k,i}$ and $e_{s_k,i}$ denote the consumed energy and the stored energy, respectively, η is the power manager efficiency, $P_{\text{harv},k,i}$ is the harvested power, P_{leak} is the capacitor leakage power, and P_{sleep} is the power consumed during sleep phase. These parameters and other parameters used for the experiment are defined in Table I.

Following [21], we estimated $e_{c_k,i}$ as follows:

$$e_{c_k,i} = e_a + P_{\text{sleep}} T_{s_k,i-1} \tag{69}$$

where e_a denotes the consumed energy during the active phase, assumed to be constant and known, and $P_{\text{sleep}} T_{s_k,i-1}$ is a prediction of the consumed energy during the sleep phase i based on the duration of the sleep phase $i-1$. Quantity e_a depends on the algorithm. It is essentially dictated by the volume of transfered data because of the excessive energy consumption of the Bluetooth module. As P_{sleep}, it was determined based on our own measurements and an estimation of the number of frames sent by each algorithm. See Table I.

Algorithm 2 Local updates at node k for the modified DCD

loop
  randomly generate $\boldsymbol{H}_{k,i}$ and $\boldsymbol{Q}_{k,i}$
  for $\ell \in \mathcal{N}_k \setminus \{k\}$ **do**
    send $\boldsymbol{H}_{k,i}\boldsymbol{w}_{k,i}$ to node ℓ
    receive from node ℓ the partial gradient vector:

$$\boldsymbol{Q}_{\ell,i}\hat{\nabla}_w J_\ell(\boldsymbol{H}_{k,i}\boldsymbol{w}_{k,i-1} + (\boldsymbol{I}_L - \boldsymbol{H}_{k,i})\boldsymbol{w}_{\ell,i-1})$$

    complete the missing entries using those available at node k, which results in $\boldsymbol{g}_{\ell,i}$ defined in (12)
  update the intermediate estimate:

$$\boldsymbol{\psi}_{k,i} = \boldsymbol{w}_{k,i-1} + \mu_k \sum_{\ell \in \mathcal{N}_k} c_{\ell k}\boldsymbol{g}_{\ell,i}$$

  calculate the local estimate:

$$\boldsymbol{w}_{k,i} = a_{kk}\boldsymbol{\psi}_{k,i} + \sum_{\ell \in \mathcal{N}_k \setminus \{k\}} a_{\ell k}\left[\boldsymbol{H}_{\ell,i}\,\boldsymbol{w}_{\ell,i-1} + (\boldsymbol{I}_L - \boldsymbol{H}_{\ell,i})\,\boldsymbol{\psi}_{k,i}\right]$$

  switch and stay in sleep mode for $T_{s_k,i}$ seconds

TABLE I: Summary of the parameters used to determine the duration of sleeping phase T_s

parameter	description	value
C_s	super capacitor capacity	0.09 F
P_{leak}	super capacitor leakage power	$3.3 \cdot 10^{-6}$ W
P_{sleep}	consumed power for sleep mode	$3.01 \cdot 10^{-5}$ W
$T_{s_{\min}}$	minimal sleep time duration	1 s
$T_{s_{\max}}$	maximal sleep time duration	300 s
V_{ref}	minimal required voltage	3.5 V
$e_{a,\text{diff}}$	consumed energy for diffusion LMS	$8.58 \cdot 10^{-2}$ J
$e_{a,\text{RCD}}$	consumed energy for red. comm. LMS	$1.61 \cdot 10^{-2}$ J
$e_{a,\text{PM}}$	consumed energy for part. dif. LMS	$5.4 \cdot 10^{-3}$ J
$e_{a,\text{cmp}}$	consumed energy for CD LMS	$7.51 \cdot 10^{-2}$ J
$e_{a,\text{dcmp}}$	consumed energy for DCD LMS	$5.4 \cdot 10^{-3}$ J

Finally, we considered the following law to simulate the amount of harvested energy:

$$E_{\text{harv},k,i} = \max(0, E_0 \sin(2\pi f i) + n(i)) \tag{70}$$

with $E_0 = 0.67J$, $E_{\text{harv},k,i}$ the harvested energy at node k and time i, $f = 10^{-5}$ a frequency, and $n_k(i)$ a zero-mean Gaussian noise with variance $\sigma_n^2 = 10^{-6}$. Note that the additive noise was used to diversify the amount of harvested energy during the Monte-Carlo runs. While it would have been possible to use a constant value over time for the harvested energy, we induced periodicity through the $\sin(\cdot)$ function to roughly model solar energy. We considered the network in Fig. 4 (left). It consists of $N = 80$ agents scattered over a hill with different lighting levels. We set L to 40. To compare the algorithms, we set their compression ratio to $r = 20$. One exception was made for the CD algorithm. As parameter r cannot reach such a large value, it was set to $r = \frac{80}{65}$. Next the step size of each algorithm was set according to Table II in order to reach the same steady-state MSD. When $\boldsymbol{A} \neq \boldsymbol{I}_L$, matrix $\boldsymbol{A}$ was set using the Metropolis rule [1].

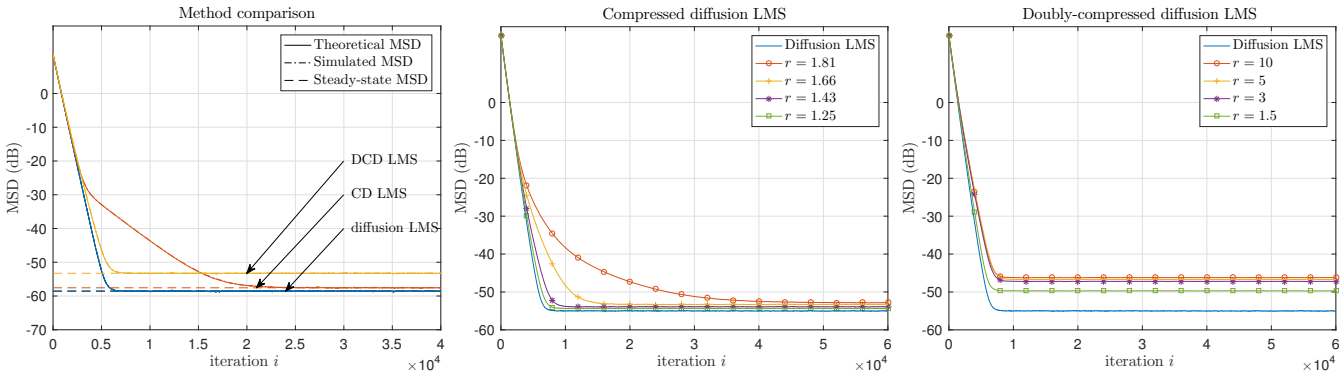

Fig. 3: (left) Theoretical and simulated MSD curves for diffusion LMS and its compressed versions. Evolution of the MSD as a function of the compression ratio for compressed diffusion LMS (center), and doubly-compressed diffusion LMS (right).

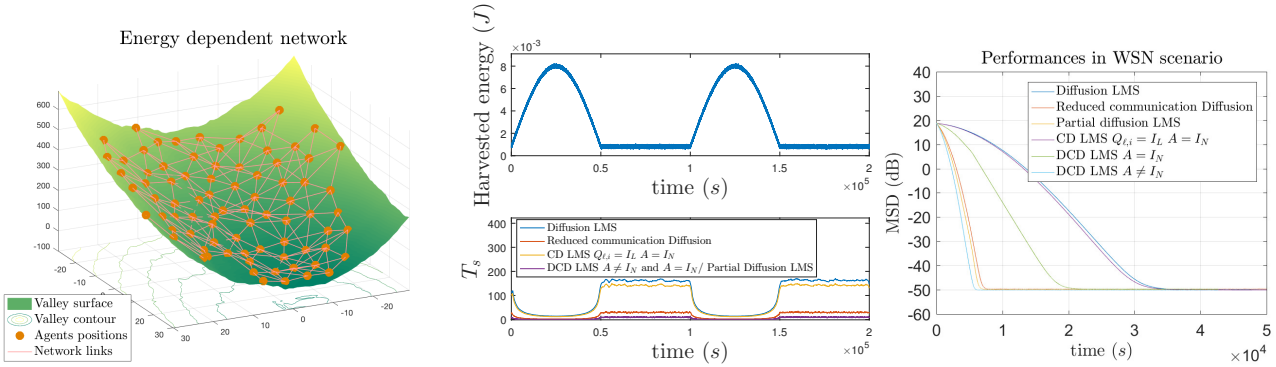

Fig. 4: (left) Network topology for WSN experiment. (center) Harvested energy and sleep periods during the experimentations. (right) Simulated MSD curves.

TABLE II: Step-size and compression settings for the different tested algorithms.

Algorithm	Step-size μ_k	Comp. ratio
Diffusion LMS	$5.4\,10^{-3}$	/
Reduced communication diffusion [5]	$1.14 \cdot 10^{-2}$	20
Partial diffusion LMS [18]	$4.4 \cdot 10^{-3}$	20
Compressed diffusion LMS	$4.8 \cdot 10^{-2}$	$\frac{80}{65}$
Doubly-compressed diffusion LMS	$6 \cdot 10^{-3}$	20

We shall now discuss the results in Fig. 4. Figure 4 (center) shows that the sleep phase duration decreases as the amount of harvested energy increases, and conversely. Also note that, for all the algorithms, the sleep phase is longer at the beginning as a consequence of the limited amount of stored energy that is available. Next, the sleep phase duration drops down until it reaches the minimal sleep duration $T_{s_{\min}}$ if possible. The less energy an algorithm consumes, the faster the super capacitors charge, and the faster the sleep phase duration of the agents drop down. As a consequence, nodes can process larger amounts of data, which makes the convergence of the algorithm faster as confirmed in Fig. 4 (right). This can be observed with the diffusion LMS and the DC algorithm, which are outperformed by the other algorithms. Let us now focus on the partial diffusion LMS and the DCD algorithm ($\boldsymbol{A} \neq \boldsymbol{I}_L$). As their compression ratio r was set to same value for comparison purposes, and their consumed energy during the active phase is almost the same, their sleep phases in Fig. 4 (center) are superimposed. The DCD algorithm however outperformed the partial diffusion LMS, in particular because it is endowed with a gradient sharing mechanism. Both algorithms outperformed the reduced-communication diffusion LMS.

V. Conclusion

Among the challenges brought up by the advent of the Internet of Things and WSN, energy efficiency is a critical one. To address this challenge, we investigated a technique for diffusion LMS that consists of sharing partial data. We carried out an analysis of the stochastic behavior of the proposed algorithm in the mean and mean-square sense. Furthermore, we provided simulation results to illustrate the accuracy of the theoretical models. Finally, we considered a realistic simulation where sensor nodes alternate between active and inactive phases. This experiment confirmed the efficiency of the proposed strategy.

VI. Appendix

Before proceeding with the calculation of the terms $\boldsymbol{P}_j$, we introduce some preliminary results.

Given any $L \times L$ matrix $\boldsymbol{\Sigma}$, it can be shown:

$$\mathbb{E}\{\boldsymbol{H}_{\ell,i}\boldsymbol{\Sigma}\boldsymbol{H}_{k,i}\} = \begin{cases} \frac{M}{L}\left(\left(1-\frac{M-1}{L-1}\right)\boldsymbol{I}_L \odot \boldsymbol{\Sigma} + \frac{M-1}{L-1}\boldsymbol{\Sigma}\right) & \text{if } \ell = k \\ \left(\frac{M}{L}\right)^2 \boldsymbol{\Sigma} & \text{otherwise} \end{cases} \tag{71}$$

where $\odot$ is the Hadamard entry-wise product.

Consider the block diagonal matrix $\boldsymbol{\mathcal{H}}_i$ and any $NL \times NL$ matrix $\boldsymbol{\Pi}$. By using (71) for each block $\mathbb{E}\{[\boldsymbol{\mathcal{H}}_i\boldsymbol{\Pi}\boldsymbol{\mathcal{H}}_i]_{k\ell}\}$, it follows that:

$$\begin{aligned}&\mathbb{E}\{\boldsymbol{\mathcal{H}}_i\boldsymbol{\Pi}\boldsymbol{\mathcal{H}}_i\} \\ &= \beta_1\left(\boldsymbol{I}_N \otimes \boldsymbol{1}_{LL}\right) \odot \boldsymbol{\Pi} + \beta_2\,\boldsymbol{I}_{NL} \odot \boldsymbol{\Pi} + \beta_3\,\boldsymbol{\Pi}\end{aligned} \tag{72}$$

where $\boldsymbol{1}_{LL}$ denotes the all-one $L \times L$ matrix, and:

$$\beta_1 = \frac{M}{L}\left(\frac{M-1}{L-1} - \frac{M}{L}\right) \tag{73}$$

$$\beta_2 = \frac{M}{L}\left(1 - \frac{M-1}{L-1}\right) \tag{74}$$

$$\beta_3 = \left(\frac{M}{L}\right)^2 \tag{75}$$

Finally we consider the $NL \times NL$ matrix, say $\boldsymbol{\varphi}_H(\boldsymbol{\Pi})$, defined by its $L \times L$ blocks:

$$[\boldsymbol{\varphi}_H(\boldsymbol{\Pi})]_{k\ell} = \mathbb{E}\{\boldsymbol{H}_{k,i}[\boldsymbol{\Pi}]_{k\ell}\boldsymbol{H}_{k,i}\} \tag{76}$$

By using (71) for each block, it can be shown that:

$$\boldsymbol{\varphi}_H(\boldsymbol{\Pi}) = \beta_2\left(\boldsymbol{1}_{NN} \otimes \boldsymbol{I}_L\right) \odot \boldsymbol{\Pi} + (\beta_1 + \beta_3)\,\boldsymbol{\Pi} \tag{77}$$

Note that $\boldsymbol{\varphi}_H(\boldsymbol{\Pi}) = \mathbb{E}\{\boldsymbol{\mathcal{H}}_i\boldsymbol{\Pi}\boldsymbol{\mathcal{H}}_i\}$ if $\boldsymbol{\Pi}$ is block diagonal.

A. Terms $\boldsymbol{P}_j$ calculation

1) Term $\boldsymbol{P}_1$ calculation: Matrix $\boldsymbol{P}_1$ is a block diagonal matrix. Its k-th diagonal block is given by:

$$[\boldsymbol{P}_1]_{kk} = \mathbb{E}\{\boldsymbol{H}_{k,i}[\boldsymbol{\mathcal{R}}_{Q,i}]_{kk}^{\top}[\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}]_{kk}[\boldsymbol{\mathcal{R}}_{Q,i}]_{kk}\boldsymbol{H}_{k,i}\} \tag{78}$$

Substituting $\boldsymbol{\mathcal{R}}_{Q,i}$ by its expression (17) leads to:

$$\begin{aligned}[\boldsymbol{P}_1]_{kk} = \sum_{m,n=1}^{N} & c_{mk}c_{nk} \\ & \mathbb{E}\{\boldsymbol{H}_{k,i}\boldsymbol{R}_{u_m,i}\boldsymbol{Q}_{m,i}[\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}]_{kk}\boldsymbol{Q}_{n,i}\boldsymbol{R}_{u_n,i}\boldsymbol{H}_{k,i}\}\end{aligned} \tag{79}$$

We rewrite $[\boldsymbol{P}_1]_{kk}$ as a sum of two terms, one for $m = n$ and one for $m \neq n$. Using (13), we get:

$$\begin{aligned}&[\boldsymbol{P}_1]_{kk} = \\ &\sum_{m=1}^{N} c_{mk}^2\mathbb{E}\{\boldsymbol{H}_{k,i}\boldsymbol{R}_{u_m,i}\boldsymbol{Q}_{m,i}[\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}]_{kk}\boldsymbol{Q}_{m,i}\boldsymbol{R}_{u_m,i}\boldsymbol{H}_{k,i}\} \\ &+ \left(\frac{M_\nabla}{L}\right)^2\Big(\\ &\quad \sum_{m,n=1}^{N} c_{mk}c_{nk}\mathbb{E}\{\boldsymbol{H}_{k,i}\boldsymbol{R}_{u_m,i}[\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}]_{kk}\boldsymbol{R}_{u_n,i}\boldsymbol{H}_{k,i}\} - \\ &\quad \sum_{m=1}^{N} c_{mk}^2\mathbb{E}\{\boldsymbol{H}_{k,i}\boldsymbol{R}_{u_m,i}[\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}]_{kk}\boldsymbol{R}_{u_m,i}\boldsymbol{H}_{k,i}\}\Big)\end{aligned} \tag{80}$$

The terms in (80) depends of higher-order moments of the regression data. While we can continue the analysis by calculating these terms, it is sufficient for the exposition to focus on the case of sufficiently small step-sizes where a reasonable approximation is [1]:

$$\mathbb{E}\{\boldsymbol{R}_{u_m,i}[\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}]_{kk}\boldsymbol{R}_{u_m,i}\} = \boldsymbol{R}_{u_m}[\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}]_{kk}\boldsymbol{R}_{u_m} \tag{81}$$

Note that this approximation will also be used in the sequel.

Finally, using Assumption 2, we can reformulate $\boldsymbol{P}_1$ as:

$$\begin{aligned}\boldsymbol{P}_1 = &\sum_{m=1}^{N}\mathbb{E}\{\boldsymbol{\mathcal{H}}_i\boldsymbol{\mathcal{R}}_{c_m}\boldsymbol{\varphi}_Q(\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}})\boldsymbol{\mathcal{R}}_{c_m}\boldsymbol{\mathcal{H}}_i\} \\ &+ \left(\frac{M_\nabla}{L}\right)^2\Big(\\ &\mathbb{E}\{\boldsymbol{\mathcal{H}}_i\boldsymbol{\mathcal{R}}\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{R}}\boldsymbol{\mathcal{H}}_i\} - \sum_{m=1}^{N}\mathbb{E}\{\boldsymbol{\mathcal{H}}_i\boldsymbol{\mathcal{R}}_{c_m}\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{R}}_{c_m}\}\boldsymbol{\mathcal{H}}_i\Big)\end{aligned} \tag{82}$$

where the matrices $\boldsymbol{\mathcal{R}}_{c_k}$ are defined as:

$$\boldsymbol{\mathcal{R}}_{c_k} = \text{diag}\{c_{k1}\boldsymbol{R}_{u_k}, \ldots, c_{kN}\boldsymbol{R}_{u_k}\} \tag{83}$$

2) Term $\boldsymbol{P}_2$ calculation : Using the same steps as above, we have:

$$[\boldsymbol{P}_2]_{kk} = \mathbb{E}\{\boldsymbol{H}_{k,i}[\boldsymbol{\mathcal{R}}_{Q,i}]_{kk}^{\top}[\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}]_{kk}[\boldsymbol{\mathcal{Q}}_i']_{kk}\boldsymbol{R}_{u_k,i}\}$$

We substitute $[\boldsymbol{\mathcal{R}}_{Q,i}]_{kk}$ and $[\boldsymbol{\mathcal{Q}}_i']_{kk}$ by their respective definitions (17) and (19):

$$\begin{aligned}[\boldsymbol{P}_2]_{kk} = \sum_{m,n=1}^{N} & c_{mk}c_{nk}\mathbb{E}\{\boldsymbol{H}_{k,i}\boldsymbol{R}_{u_m,i}\boldsymbol{Q}_{m,i}[\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}]_{kk} \\ & \left(\boldsymbol{I}_L - \boldsymbol{Q}_{n,i}\right)\boldsymbol{R}_{u_k,i}\}\end{aligned}$$

Using (13) we find that:

$$\begin{aligned}[\boldsymbol{P}_2]_{kk} = &\tfrac{MM_\nabla}{L^2}\textstyle\sum_{m=1}^{N} c_{mk}^2\mathbb{E}\{\boldsymbol{R}_{u_m,i}[\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}]_{kk}\boldsymbol{R}_{u_k,i}\} \\ &- \tfrac{M}{L}\textstyle\sum_{m=1}^{N} c_{mk}^2\mathbb{E}\{\boldsymbol{R}_{u_m,i}\boldsymbol{Q}_{m,i}[\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}]_{kk}\boldsymbol{Q}_{m,i}\boldsymbol{R}_{u_k,i}\} \\ &+ \tfrac{MM_\nabla}{L^2}\left(1 - \tfrac{M_\nabla}{L}\right) \\ &\quad \Big(\sum_{m,n=1}^{N} c_{mk}c_{nk}\mathbb{E}\{\boldsymbol{R}_{u_m,i}[\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}]_{kk}\boldsymbol{R}_{u_k,i}\} \\ &\quad - \sum_{m=1}^{N} c_{mk}^2\mathbb{E}\{\boldsymbol{R}_{u_m,i}[\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}]_{kk}\boldsymbol{R}_{u_k,i}\}\Big)\end{aligned} \tag{84}$$

Finally, we write:

$$\begin{aligned}&\boldsymbol{P}_2 \\ &= \tfrac{M_\nabla M}{L^2}\boldsymbol{\mathcal{R}}_2\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{R}}_u - \tfrac{M}{L}\boldsymbol{\mathcal{R}}_2\boldsymbol{\varphi}_Q(\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}})\boldsymbol{\mathcal{R}}_u \\ &+ \tfrac{MM_\nabla}{L^2}\left(1 - \tfrac{M_\nabla}{L}\right)\left(\boldsymbol{\mathcal{R}}\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{R}}_u - \boldsymbol{\mathcal{R}}_2\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{R}}_u\right)\end{aligned} \tag{85}$$

where

$$\boldsymbol{\mathcal{R}}_2 = \left\{\sum_{m=1}^{N} c_{m1}^2\boldsymbol{R}_{u_m}, \ldots, \sum_{m=1}^{N} c_{mN}^2\boldsymbol{R}_{u_m}\right\} \tag{86}$$

3) Term $\boldsymbol{P}_3$ calculation: Term $\boldsymbol{P}_3$ can be expressed as

$$[\boldsymbol{P}_3]_{k\ell} = \mathbb{E}\{\boldsymbol{H}_{k,i}[\boldsymbol{\mathcal{R}}_{Q,i}]_{kk}^{\top}[\boldsymbol{\mathcal{M}\Sigma\mathcal{M}}]_{kk}[\boldsymbol{\mathcal{R}}_{Q(I-H),i}]_{k\ell}\} \quad (87)$$

Replacing $[\boldsymbol{\mathcal{R}}_{Q,i}]_{kk}$ and $[\boldsymbol{\mathcal{R}}_{Q(I-H),i}]$ by their definitions (17) and (23), respectively, we get:

$$[\boldsymbol{P}_3]_{k\ell} = \quad (88)$$
$$\sum_{m=1}^{N} c_{mk}c_{\ell k}\mathbb{E}\{\boldsymbol{H}_{k,i}\boldsymbol{R}_{u_m,i}\boldsymbol{Q}_{m,i}[\boldsymbol{\mathcal{M}\Sigma\mathcal{M}}]_{kk} \boldsymbol{Q}_{\ell,i}\boldsymbol{R}_{u_\ell,i}\left(\boldsymbol{I}_L - \boldsymbol{H}_{k,i}\right)\}$$

Applying (13) and (46) leads to:

$$\begin{aligned}[\boldsymbol{P}_3]_{k\ell} = {} & \tfrac{M}{L}\, c_{\ell k}^2\, \mathbb{E}\{\boldsymbol{R}_{u_\ell}\boldsymbol{Q}_{\ell,i}[\boldsymbol{\mathcal{M}\Sigma\mathcal{M}}]_{kk}\boldsymbol{Q}_{\ell,i}\boldsymbol{R}_{u_\ell}\} \\ & - c_{\ell k}^2\, \mathbb{E}\{\boldsymbol{H}_{k,i}\boldsymbol{R}_{u_\ell}\boldsymbol{Q}_{\ell,i}[\boldsymbol{\mathcal{M}\Sigma\mathcal{M}}]_{kk}\boldsymbol{Q}_{\ell,i}\boldsymbol{R}_{u_\ell}\boldsymbol{H}_{k,i}\} \\ & + \left(\tfrac{M_\nabla}{L}\right)^2 \Big(\tfrac{M}{L}\textstyle\sum_{m=1}^{N} c_{mk}c_{\ell k}\boldsymbol{R}_{u_m}[\boldsymbol{\mathcal{M}\Sigma\mathcal{M}}]_{kk}\boldsymbol{R}_{u_\ell} \\ & - \sum_{m=1}^{N} c_{mk}c_{\ell k}\mathbb{E}\{\boldsymbol{H}_{k,i}\boldsymbol{R}_{u_m}[\boldsymbol{\mathcal{M}\Sigma\mathcal{M}}]_{kk}\boldsymbol{R}_{u_\ell}\boldsymbol{H}_{k,i}\} \\ & - \tfrac{M}{L}\, c_{\ell k}^2\, \boldsymbol{R}_{u_\ell}[\boldsymbol{\mathcal{M}\Sigma\mathcal{M}}]_{kk}\boldsymbol{R}_{u_\ell} \\ & + c_{\ell k}^2\, \mathbb{E}\{\boldsymbol{H}_{k,i}\boldsymbol{R}_{u_\ell}[\boldsymbol{\mathcal{M}\Sigma\mathcal{M}}]_{kk}\boldsymbol{R}_{u_\ell}\boldsymbol{H}_{k,i}\}\Big) \end{aligned} \quad (89)$$

We have:

$$c_{\ell k}^2\, \boldsymbol{R}_{u_\ell}[\boldsymbol{\mathcal{M}\Sigma\mathcal{M}}]_{kk}\boldsymbol{R}_{u_\ell} = \sum_{m=1}^{N}[\boldsymbol{\mathcal{R}}'_{u_m}\boldsymbol{\mathcal{M}\Sigma\mathcal{M}}\boldsymbol{\mathcal{C}}_2^{\top}\boldsymbol{\mathcal{I}}_m\boldsymbol{\mathcal{R}}_u]_{k\ell} \quad (90)$$

where

$$\boldsymbol{\mathcal{R}}'_{u_m} = \boldsymbol{I}_L \otimes \boldsymbol{R}_{u_m} \quad (91)$$
$$\boldsymbol{\mathcal{I}}_m = \text{diag}\{0, 0, \ldots, \boldsymbol{I}_L, 0, \ldots, 0\} \quad (92)$$

All the entries of the matrix $\boldsymbol{\mathcal{I}}_m$ are equal to zero except the (m,m)-th block which is equal to $\boldsymbol{I}_L$.

Using (90), we find that:

$$\begin{aligned}\boldsymbol{P}_3 = {} & \tfrac{M}{L}\textstyle\sum_{m=1}^{N}\boldsymbol{\mathcal{R}}'_{u_m}\boldsymbol{\varphi}_Q(\boldsymbol{\mathcal{M}\Sigma\mathcal{M}})\boldsymbol{\mathcal{C}}_2^{\top}\boldsymbol{\mathcal{I}}_m\boldsymbol{\mathcal{R}}_u \\ & - \sum_{m=1}^{N}\boldsymbol{\varphi}_H(\boldsymbol{\mathcal{R}}'_{u_m}\boldsymbol{\varphi}_Q(\boldsymbol{\mathcal{M}\Sigma\mathcal{M}})\boldsymbol{\mathcal{C}}_2^{\top}\boldsymbol{\mathcal{I}}_m\boldsymbol{\mathcal{R}}_u) \\ & + \left(\tfrac{M_\nabla}{L}\right)^2\Bigg(\tfrac{M}{L}\boldsymbol{\mathcal{R}}\boldsymbol{\mathcal{M}\Sigma\mathcal{M}}\boldsymbol{\mathcal{C}}^{\top}\boldsymbol{\mathcal{R}}_u - \boldsymbol{\varphi}_H(\boldsymbol{\mathcal{R}}\boldsymbol{\mathcal{M}\Sigma\mathcal{M}}\boldsymbol{\mathcal{C}}^{\top}\boldsymbol{\mathcal{R}}_u) \\ & - \tfrac{M}{L}\textstyle\sum_{m=1}^{N}\boldsymbol{\mathcal{R}}'_{u_m}\boldsymbol{\mathcal{M}\Sigma\mathcal{M}}\boldsymbol{\mathcal{C}}_2^{\top}\boldsymbol{\mathcal{I}}_m\boldsymbol{\mathcal{R}}_u \\ & + \sum_{m=1}^{N}\boldsymbol{\varphi}_H(\boldsymbol{\mathcal{R}}'_{u_m}\boldsymbol{\mathcal{M}\Sigma\mathcal{M}}\boldsymbol{\mathcal{C}}_2^{\top}\boldsymbol{\mathcal{I}}_m\boldsymbol{\mathcal{R}}_u)\Bigg) \end{aligned} \quad (93)$$

4) Term $\boldsymbol{P}_4$ calculation : We express $\boldsymbol{P}_4$ as follows:

$$[\boldsymbol{P}_4]_{kk} = \mathbb{E}\{\boldsymbol{R}_{u_k,i}[\boldsymbol{\mathcal{Q}}'_i]_{kk}[\boldsymbol{\mathcal{M}\Sigma\mathcal{M}}]_{kk}[\boldsymbol{\mathcal{Q}}'_i]_{kk}\boldsymbol{R}_{u_k,i}\}$$

Substituting $[\boldsymbol{\mathcal{Q}}'_i]_{kk}$ by its definition (19) we get:

$$[\boldsymbol{P}_4]_{kk} = \sum_{m,n=1}^{N} c_{mk}c_{nk}\mathbb{E}\{\boldsymbol{R}_{u_k,i}\left(\boldsymbol{I}_L - \boldsymbol{Q}_{m,i}\right)[\boldsymbol{\mathcal{M}\Sigma\mathcal{M}}]_{kk} \left(\boldsymbol{I}_L - \boldsymbol{Q}_{n,i}\right)\boldsymbol{R}_{u_k,i}\}$$

Using (13) leads to

$$\begin{aligned}[\boldsymbol{P}_4]_{kk} = {} & \left(1 - 2\tfrac{M_\nabla}{L}\right)\sum_{m,n=1}^{N} c_{mk}c_{nk}\mathbb{E}\{\boldsymbol{R}_{u_k,i}[\boldsymbol{\mathcal{M}\Sigma\mathcal{M}}]_{kk}\boldsymbol{R}_{u_k,i}\} \\ & + \sum_{m,n=1}^{N} c_{mk}c_{nk}\mathbb{E}\{\boldsymbol{R}_{u_k,i}\boldsymbol{Q}_{m,i}[\boldsymbol{\mathcal{M}\Sigma\mathcal{M}}]_{kk}\boldsymbol{Q}_{n,i}\boldsymbol{R}_{u_k,i}\} \end{aligned}$$

We rearrange the sum as follows:

$$\begin{aligned}[\boldsymbol{P}_4]_{kk} = {} & \left(1 - 2\tfrac{M_\nabla}{L}\right)\sum_{m,n=1}^{N} c_{mk}c_{nk}\mathbb{E}\{\boldsymbol{R}_{u_k,i}[\boldsymbol{\mathcal{M}\Sigma\mathcal{M}}]_{kk}\boldsymbol{R}_{u_k,i}\} \\ & + \sum_{m=1}^{N} c_{mk}^2\mathbb{E}\{\boldsymbol{R}_{u_k,i}\boldsymbol{Q}_{m,i}[\boldsymbol{\mathcal{M}\Sigma\mathcal{M}}]_{kk}\boldsymbol{Q}_{m,i}\boldsymbol{R}_{u_k,i}\} \\ & + \left(\tfrac{M_\nabla}{L}\right)^2\Bigg(\sum_{m,n=1}^{N} c_{mk}c_{nk}\mathbb{E}\{\boldsymbol{R}_{u_k,i}[\boldsymbol{\mathcal{M}\Sigma\mathcal{M}}]_{kk}\boldsymbol{R}_{u_k,i}\} \\ & - \sum_{m=1}^{N} c_{mk}^2\mathbb{E}\{\boldsymbol{R}_{u_k,i}[\boldsymbol{\mathcal{M}\Sigma\mathcal{M}}]_{kk}\boldsymbol{R}_{u_k,i}\}\Bigg) \end{aligned}$$

Finally, we can write $\boldsymbol{P}_4$ in a compact form:

$$\begin{aligned}\boldsymbol{P}_4 = {} & \left(1 - 2\tfrac{M_\nabla}{L}\right)\boldsymbol{\mathcal{R}}_u\boldsymbol{\mathcal{M}\Sigma\mathcal{M}}\boldsymbol{\mathcal{R}}_u \\ & + \boldsymbol{\mathcal{R}}_u(\boldsymbol{I}_{NM} \odot \boldsymbol{\mathcal{C}}\,\boldsymbol{\mathcal{C}}^{\top})\,\boldsymbol{\varphi}_Q(\boldsymbol{\mathcal{M}\Sigma\mathcal{M}})\boldsymbol{\mathcal{R}}_u \\ & + \left(\tfrac{M_\nabla}{L}\right)^2(\boldsymbol{\mathcal{R}}_u\boldsymbol{\mathcal{M}\Sigma\mathcal{M}}\boldsymbol{\mathcal{R}}_u \\ & - \boldsymbol{\mathcal{R}}_u(\boldsymbol{I}_{NM} \odot \boldsymbol{\mathcal{C}}\,\boldsymbol{\mathcal{C}}^{\top})\boldsymbol{\mathcal{M}\Sigma\mathcal{M}}\boldsymbol{\mathcal{R}}_u\Big) \end{aligned} \quad (94)$$

5) Term $\boldsymbol{P}_5$ calculation: Expanding $\boldsymbol{P}_5$ leads to:

$$\boldsymbol{P}_5 = \boldsymbol{P}_{5,1} - \boldsymbol{P}_{5,2} - \boldsymbol{P}_{5,3} + \boldsymbol{P}_{5,4} \quad (95)$$

where:

$$\boldsymbol{P}_{5,1} = \sum_{m=1}^{N} c_{mk}c_{\ell k}\mathbb{E}\{\boldsymbol{R}_{u_k,i}[\boldsymbol{\mathcal{M}\Sigma\mathcal{M}}]_{kk}\boldsymbol{Q}_{\ell,i}\boldsymbol{R}_{u_\ell,i}\} \quad (96)$$

$$\boldsymbol{P}_{5,2} = \sum_{m=1}^{N} c_{mk}c_{\ell k}\mathbb{E}\{\boldsymbol{R}_{u_k,i}\boldsymbol{Q}_{m,i}[\boldsymbol{\mathcal{M}\Sigma\mathcal{M}}]_{kk} \boldsymbol{Q}_{\ell,i}\boldsymbol{R}_{u_\ell,i}\} \quad (97)$$

$$\boldsymbol{P}_{5,3} = \sum_{m=1}^{N} c_{mk}c_{\ell k}\mathbb{E}\{\boldsymbol{R}_{u_k,i}[\boldsymbol{\mathcal{M}\Sigma\mathcal{M}}]_{kk} \boldsymbol{Q}_{\ell,i}\boldsymbol{R}_{u_\ell,i}\boldsymbol{H}_{k,i}\} \quad (98)$$

$$\boldsymbol{P}_{5,4} = \sum_{m=1}^{N} c_{mk}c_{\ell k}\mathbb{E}\{\boldsymbol{R}_{u_k,i}\boldsymbol{Q}_{m,i}[\boldsymbol{\mathcal{M}\Sigma\mathcal{M}}]_{kk} \boldsymbol{Q}_{\ell,i}\boldsymbol{R}_{u_\ell,i}\boldsymbol{H}_{k,i}\} \quad (99)$$

Following the same steps as earlier, and using the results (13) and (46), leads to:

$$\boldsymbol{P}_{5,1} = \tfrac{M_\nabla}{L}\boldsymbol{\mathcal{R}}_u\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{C}}^\top\boldsymbol{\mathcal{R}}_u \tag{100}$$

$$\boldsymbol{P}_{5,2} = \tfrac{M}{L}\Big[\boldsymbol{\mathcal{R}}_u\boldsymbol{\varphi}_Q(\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}})\boldsymbol{\mathcal{C}}_2^\top\boldsymbol{\mathcal{R}}_u \tag{101}$$
$$+\left(\tfrac{M_\nabla}{L}\right)^2\Big(\boldsymbol{\mathcal{R}}_u\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{C}}^\top\boldsymbol{\mathcal{R}}_u - \boldsymbol{\mathcal{R}}_u\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{C}}_2^\top\boldsymbol{\mathcal{R}}_u\Big)\Big]$$

$$\boldsymbol{P}_{5,3} = \tfrac{M}{L}\boldsymbol{\mathcal{R}}_u\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{C}}^\top\boldsymbol{\mathcal{R}}_u \tag{102}$$

$$\boldsymbol{P}_{5,4} = \tfrac{M}{L}\boldsymbol{\varphi}_Q(\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}})\boldsymbol{\mathcal{C}}_2^\top\boldsymbol{R}_u \tag{103}$$
$$+\tfrac{M}{L}\left(\tfrac{M_\nabla}{L}\right)^2\Big(\boldsymbol{\mathcal{R}}_u\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{C}}^\top\boldsymbol{R}_u - \boldsymbol{\mathcal{R}}_u\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{C}}_2^\top\boldsymbol{R}_u\Big)$$

6) Term $\boldsymbol{P}_6$ calculation: Proceeding as previously we find:

$$\boldsymbol{P}_6 = \left(1-\tfrac{2M}{L}\right)\boldsymbol{\mathcal{R}}_u\mathbb{E}\{\boldsymbol{\mathcal{Q}}_i\boldsymbol{\mathcal{C}}\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{C}}^\top\boldsymbol{\mathcal{Q}}_i\}\boldsymbol{\mathcal{R}}_u + \boldsymbol{\varphi}_H(\boldsymbol{\mathcal{R}}_u\mathbb{E}\{\boldsymbol{\mathcal{Q}}_i\boldsymbol{\mathcal{C}}\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{C}}^\top\boldsymbol{\mathcal{Q}}_i\}\boldsymbol{\mathcal{R}}_u) \tag{104}$$

B. Terms $\boldsymbol{P}_j$ vectorization

In order to apply the $\mathrm{vec}(\cdot)$ operator to the terms $\boldsymbol{P}_j$, we use the following transformations:

$$(\boldsymbol{I}_N\otimes\boldsymbol{1}_{LL})\odot\boldsymbol{\Pi} = \sum_{n=1}^{N}\boldsymbol{T}_n\boldsymbol{\Pi}\boldsymbol{T}_n \tag{105}$$

$$\boldsymbol{I}_{NL}\odot\boldsymbol{\Pi} = \sum_{n=1}^{NL}\boldsymbol{D}_n\boldsymbol{\Pi}\boldsymbol{D}_n \tag{106}$$

$$(\boldsymbol{1}_{NN}\otimes\boldsymbol{I}_L)\odot\boldsymbol{\Pi} = \sum_{m=1}^{NL}\sum_{k=1}^{\lfloor\frac{NL-M}{L}\rfloor}(\boldsymbol{D}_m\boldsymbol{\Pi}\boldsymbol{D}_{m+kL}+\boldsymbol{D}_{m+kL}\boldsymbol{\Pi}\boldsymbol{D}_m) - \sum_{m=1}^{N}\boldsymbol{D}_m\boldsymbol{\Pi}\boldsymbol{D}_m \tag{107}$$

where $\boldsymbol{\Pi}$, $\boldsymbol{T}_n$ and $\boldsymbol{D}_n$, are $(NL\times NL)$ matrices and

$$[\boldsymbol{T}_n]_{k\ell} = \delta(k,\ell)\delta(k,n)\boldsymbol{I}_L \tag{108}$$

$$(\boldsymbol{D}_n)_{k\ell} = \delta(k,\ell)\delta(k,n) \tag{109}$$

where $\delta(k,\ell)=1$ if $k=\ell$, and 0 otherwise. Using (72), we find:

$$\begin{aligned}\mathrm{vec}(\boldsymbol{\mathcal{R}}_u\mathbb{E}\{\boldsymbol{\mathcal{Q}}_i\boldsymbol{\mathcal{C}}\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{C}}^\top\boldsymbol{\mathcal{Q}}_i\}\boldsymbol{\mathcal{R}}_u) =& \\ \alpha_1\,\mathrm{vec}(\boldsymbol{\mathcal{R}}_u[(\boldsymbol{I}_N\otimes\boldsymbol{1}_{LL})\odot(\boldsymbol{\mathcal{C}}\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{C}}^\top)]\boldsymbol{\mathcal{R}}_u)& \\ +\alpha_2\,\mathrm{vec}(\boldsymbol{\mathcal{R}}_u[\boldsymbol{I}_N\odot(\boldsymbol{\mathcal{C}}\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{C}}^\top)]\boldsymbol{\mathcal{R}}_u)& \\ +\alpha_3\,\mathrm{vec}(\boldsymbol{\mathcal{R}}_u\boldsymbol{\mathcal{C}}\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{C}}^\top\boldsymbol{\mathcal{R}}_u)&\end{aligned} \tag{110}$$

Using (105) and (106) we have:

$$\begin{aligned}\mathrm{vec}(\boldsymbol{\mathcal{R}}_u\mathbb{E}\{\boldsymbol{\mathcal{Q}}_i\boldsymbol{\mathcal{C}}\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{C}}^\top\boldsymbol{\mathcal{Q}}_i\}\boldsymbol{\mathcal{R}}_u) =& \\ \alpha_1\sum_{n=1}^{N}\mathrm{vec}(\boldsymbol{\mathcal{R}}_u\boldsymbol{T}_n\boldsymbol{\mathcal{C}}\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{C}}^\top\boldsymbol{T}_n\boldsymbol{\mathcal{R}}_u)& \\ +\alpha_2\sum_{m=1}^{NL}\mathrm{vec}(\boldsymbol{\mathcal{R}}_u\boldsymbol{D}_m\boldsymbol{\mathcal{C}}\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{C}}^\top\boldsymbol{D}_m\boldsymbol{\mathcal{R}}_u)& \\ +\alpha_3\,\mathrm{vec}(\boldsymbol{\mathcal{R}}_u\boldsymbol{\mathcal{C}}\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{C}}^\top\boldsymbol{\mathcal{R}}_u)&\end{aligned} \tag{111}$$

Furthermore, we have:

$$\mathrm{vec}(\boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{B}) = (\boldsymbol{B}^\top\otimes\boldsymbol{A})\mathrm{vec}(\boldsymbol{\Sigma}) \tag{112}$$

Applying (112) leads to:

$$\begin{aligned}\mathrm{vec}(\boldsymbol{\mathcal{R}}_u\mathbb{E}\{\boldsymbol{\mathcal{Q}}_i\boldsymbol{\mathcal{C}}\boldsymbol{\mathcal{M}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{C}}^\top\boldsymbol{\mathcal{Q}}_i\}\boldsymbol{\mathcal{R}}_u) =& \\ \alpha_1\sum_{n=1}^{N}\mathrm{vec}(\boldsymbol{\mathcal{R}}_u\boldsymbol{T}_n\boldsymbol{\mathcal{C}}\boldsymbol{\mathcal{M}}\otimes\boldsymbol{\mathcal{R}}_u\boldsymbol{T}_n\boldsymbol{\mathcal{C}}\boldsymbol{\mathcal{M}})\boldsymbol{\sigma}& \\ +\alpha_2\sum_{m=1}^{NL}\mathrm{vec}(\boldsymbol{\mathcal{R}}_u\boldsymbol{D}_m\boldsymbol{\mathcal{C}}\boldsymbol{\mathcal{M}}\otimes\boldsymbol{\mathcal{R}}_u\boldsymbol{D}_m\boldsymbol{\mathcal{C}}\boldsymbol{\mathcal{M}})\boldsymbol{\sigma}& \\ +\alpha_3\,\mathrm{vec}(\boldsymbol{\mathcal{R}}_u\boldsymbol{\mathcal{C}}\boldsymbol{\mathcal{M}}\otimes\boldsymbol{\mathcal{R}}_u\boldsymbol{\mathcal{C}}\boldsymbol{\mathcal{M}})\boldsymbol{\sigma}&\end{aligned} \tag{113}$$

REFERENCES

[1] A. H. Sayed, "Diffusion adaptation over networks," in *Academic Press Libraray in Signal Processing*, R. Chellapa and S. Theodoridis, Eds. Elsevier, 2014. Also available as arXiv:1205.4220 [cs.MA], May 2012., pp. 322–454.

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

[3] A. H. Sayed, "Adaptive networks," *Proceedings of the IEEE*, vol. 102, no. 4, pp. 460–497, 2014.

[4] X. Zhao, S.-Y. Tu, and A. H. Sayed, "Diffusion adaptation over networks under imperfect information exchange and non-stationary data," *IEEE Transactions on Signal Processing*, vol. 60, no. 7, pp. 3460–3475, 2012.

[5] R. Arablouei, S. Werner, K. Doğançay, and Y.-F. Huang, "Analysis of a reduced-communication diffusion LMS algorithm," *Signal Processing*, vol. 117, pp. 355–361, 2015.

[6] J. Chen, C. Richard, A. O. Hero, and A. H. Sayed, "Diffusion lms for multitask problems with overlapping hypothesis subspaces," in *Proc. IEEE MLSP'14*, Reims, France, 2014, pp. 1–6.

[7] J. Chen, C. Richard, and A. H. Sayed, "Multitask diffusion adaptation over networks," *IEEE Transactions on Signal Processing*, vol. 62, no. 16, pp. 4129–4144, 2014.

[8] ——, "Diffusion LMS over multitask networks," *IEEE Transactions on Signal Processing*, vol. 63, no. 11, pp. 2733–2748, 2015.

[9] R. Nassif, C. Richard, A. Ferrari, and A. H. Sayed, "Multitask diffusion adaptation over asynchronous networks," *IEEE Transactions on Signal Processing*, vol. 64, no. 11, pp. 2835–2850, 2016.

[10] ——, "Proximal multitask learning over networks with sparsity-inducing coregularization," *IEEE Transactions on Signal Processing*, vol. 64, no. 23, pp. 6329–6344, 2016.

[11] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, 2009.

[12] 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.

[13] C. G. Lopes and A. H. Sayed, "Incremental adaptive strategies over distributed networks," *IEEE Transactions on Signal Processing*, vol. 55, no. 8, pp. 4064–4077, 2007.

[14] W. Gao, J. Chen, C. Richard, and J. Huang, "Diffusion adaptation over networks with kernel least-mean-square," in *Proc. IEEE CAMSAP'15*, Cancún, Mexico, 2015.

[15] P. Chainais and C. Richard, "Learning a common dictionary over a sensor network," in *Proc. IEEE CAMSAP'13*, Saint Martin, French West Indies, 2013.

[16] M. O. Sayin and S. S. Kozat, "Compressive diffusion strategies over distributed networks for reduced communication load," *IEEE Transactions on Signal Processing*, vol. 62, no. 20, pp. 5308–5323, 2014.

[17] R. Arablouei, S. Werner, Y.-F. Huang, and K. Dogancay, "Distributed least mean-square estimation with partial diffusion," *IEEE Transactions on Signal Processing*, vol. 62, no. 2, pp. 472–484, 2014.

[18] R. Arablouei, K. Dogancay, S. Werner, and Y.-F. Huang, "Adaptive distributed estimation based on recursive least-squares and partial diffusion," *IEEE Transactions on Signal Processing*, vol. 62, no. 14, pp. 3510–3522, 2014.

[19] V. Vadidpour, A. Rastegarnia, A. Khalili, and S. Sanei, "Partial-diffusion least mean-square estimation over networks under noisy information exchange," *arXiv preprint arXiv:1511.09044*, 2015.

[20] I. E. K. Harrane, R. Flamary, and C. Richard, "Doubly compressed diffusion lms over adaptive networks."

[21] T. N. Le, A. Pegatoquet, O. Berder, and O. Sentieys, "Multi-source power manager for super-capacitor based energy harvesting wsn," in *Proceedings of the 1st International Workshop on Energy Neutral Sensing Systems*. ACM, 2013, p. 19.