# Resource-aware event triggered distributed estimation over adaptive networks

Ihsan Utlu [a,b], O. Fatih Kilic [a], Suleyman S. Kozat [a,*]

[a] *Department of Electrical and Electronics Engineering, Bilkent University, Ankara, Turkey*
[b] *ASELSAN Research Center, Ankara 06370, Turkey*

## ARTICLE INFO

## ABSTRACT

We propose a novel algorithm for distributed processing applications constrained by the available communication resources using diffusion strategies that achieves up to a $10^3$ fold reduction in the communication load over the network, while delivering a comparable performance with respect to the state of the art. After computation of local estimates, the information is diffused among the processing elements (or nodes) non-uniformly in time by conditioning the information transfer on level-crossings of the diffused parameter, resulting in a greatly reduced communication requirement. We provide the mean and mean-square stability analyses of our algorithms, and illustrate the gain in communication efficiency compared to other reduced-communication distributed estimation schemes.

© 2017 Elsevier Inc. All rights reserved.

## 1. Introduction

In tandem with the increasing computational capabilities of processing units and the growing amount of generated data, the demand on distributed networks and decentralized data processing algorithms have remained an area of growing interest [1–3]. With intrinsic characteristics such as robustness and scalability, distributed architectures provide enhanced efficiency and performance for a wide variety of applications ranging from adaptive filtering, sequential detection, sensor networks, to distributed resource allocation [4–9]. However, successful implementation of such applications depends on a substantial amount of communication resources. As an example, in smart grid applications, measurement units operating with high frequency put the communication infrastructure of the grid under significant pressure [10]. This calls for resource-efficient, event-triggered distributed estimation solutions that incorporate event-driven communication [11–15]. To this end, in this paper, we construct distributed architectures that have a significantly reduced communication load without compromising performance. We achieve this by introducing novel event triggered communication architectures over distributed networks.

In a distributed processing framework, a group of measurement-capable agents, termed nodes, in a network cooperate with one another in order to estimate an unknown common phenomenon [16].

Among the different approaches for distributed estimation, we specifically consider diffusion-based protocols that exploit the spatial diversity of the network by restricting information sharing to neighboring nodes, without considering any central processing unit or a fusion center [16,17]. Diffusion protocols provide an inherently scalable data processing framework that is resilient to changes in network topology such as link failures as well as changes in the statistical properties of the unknown phenomenon that is measured [16]. However, the requirement for all nodes to exchange their current estimates with their neighbors at each iteration places a heavy burden on the available communication resources [18].

Here, we propose novel event-triggered distributed estimation algorithms for communication-constrained applications that achieve up to a $10^3$ fold reduction in the communication load over the network. We achieve this by leveraging the uneven distribution of the events over time to efficiently reduce the communication load in real life applications. In particular, we condition an information exchange between the neighboring nodes on the level-crossings of the diffused parameter [19], unlike using a fixed rate of diffusion, cf. [16,17]. Furthermore, we show that it is sufficient to only diffuse the information indicating the direction of the change in the levels, which can be handled using only two bits for a slowly-varying parameter.

Reduced communication diffusion is extensively studied in the signal processing literature [18,20–23]. In [18,20,21], the authors restrict the number of active links between neighbors using a probabilistic framework, or by adaptively choosing a single link of communication for each node. In [22], local estimates are ran-
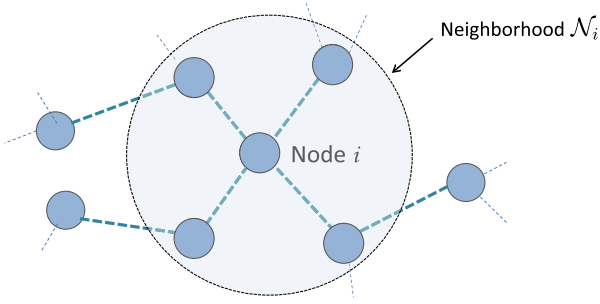
**Fig. 1.** An example distributed network with bidirectional connections. Circular area represents the neighborhood of the $i$th node.

domly projected, and the information transfer between the nodes is reduced to a single bit. In [23], only certain dimensions of the parameter vector are transmitted. On the other hand, in this paper, we reduce the communication load down to only a single bit or a couple of bits, unlike [18,20,21,23], in which authors diffuse parameters in full precision. Furthermore, we regulate the frequency of information exchange depending on the rate of change of the parameter, unlike [22] where the authors transfer information at each single time instant.

Our main contributions are as follows. We introduce algorithms for distributed estimation that i) significantly reduce the communication load on the network, ii) while continuing to provide equal performance with the state of the art. We also perform the mean and mean-square stability analyses of our algorithms. Through numerical examples, we show that our algorithms provide significant reduction in the communication load over the network.

The paper is organized as follows: In Section 2, we introduce the distributed estimation framework and discuss the adapt-then-combine (ATC) diffusion strategy. We further detail our algorithms in Section 3, where we formulate the level-triggered distributed estimation algorithm. In Section 4, we present the algorithmic description of the proposed scheme. In Sections 5 and 6, we provide respectively the mean and mean-square stability analyses of the proposed distributed adaptive filter and state the conditions for stability. We provide experimental verification of the algorithm in Section 7, and concluding remarks in Section 8.

## 2. Problem description

Consider a network with $N$ nodes that are distributed spatially as shown in Fig. 1. Each node sequentially observes a noise-corrupted transformation of an unknown parameter $\boldsymbol{w_o}$ through a linear model

$$d_{i,t} = \boldsymbol{u}_{i,t}^T \boldsymbol{w_o} + v_{i,t}, \quad i = 1, \ldots, N \tag{1}$$

and diffuses information to its neighboring nodes $j \in \mathcal{N}_i$,[1] where $\boldsymbol{w_o} \in \mathbb{R}^M$ is the unknown phenomenon, with $\boldsymbol{u}_{i,t}$ and $v_{i,t}$ representing the regressor and the noise processes, respectively. The additive observation noise $v_{i,t}$ and the regressor $\boldsymbol{u}_{i,t}$ are assumed to be temporally and spatially independent, and independent of one another, with $E\left[\boldsymbol{u}_{i,t}\boldsymbol{u}_{i,t}^T\right] = \sigma_{u,i}^2 \boldsymbol{I}_M$, $E\left[v_{i,t}^2\right] = \sigma_{v,i}^2$. For each node $i$, we assume that at time $t$ only the regressor $\boldsymbol{u}_{i,t}$ and the

observation $d_{i,t}$ along with the parameter estimates from neighboring nodes $\boldsymbol{\phi}_{j,t}$, $j \in \mathcal{N}_i$ are available to it. Therefore each node incurs the cost for the parameter $\boldsymbol{w}$ [17]

$$J_i(\boldsymbol{w}) = \frac{1}{2}E|d_{i,t} - \boldsymbol{u}_{i,t}^T\boldsymbol{w}|^2 + \frac{1}{2}\sum_{j\in\mathcal{N}_i\setminus\{i\}}\alpha_{i,j}\left\|\boldsymbol{w} - \boldsymbol{\phi}_j\right\|_2^2, \tag{2}$$

where $\alpha_{i,j}$ is a non-negative, real coefficient satisfying $\sum_{j=1}^N \alpha_{i,j} = 1$ that assigns different weights to different neighbors. In order to minimize (2) in an online manner, we employ the stochastic gradient approach [24]. To this end, we calculate the gradient for (2) as

$$\left[\nabla_{\boldsymbol{w}} J_i(\boldsymbol{w})\right]^T = \left(\boldsymbol{R}_{\boldsymbol{u},i}\boldsymbol{w} - \boldsymbol{R}_{d\boldsymbol{u},i}\right) + \sum_{j\in\mathcal{N}_i\setminus\{i\}}\alpha_{i,j}(\boldsymbol{w} - \boldsymbol{\phi}_j), \tag{3}$$

where $\boldsymbol{R}_{\boldsymbol{u},i} = E[\boldsymbol{u}_{i,t}\boldsymbol{u}_{i,t}^T]$ and $\boldsymbol{R}_{d\boldsymbol{u},i} = E[\boldsymbol{u}_{i,t}d_{i,t}]$. Using the instantaneous approximations $\boldsymbol{R}_{\boldsymbol{u},i} \approx \boldsymbol{u}_{i,t}\boldsymbol{u}_{i,t}^T$ and $\boldsymbol{R}_{d\boldsymbol{u},i} \approx \boldsymbol{u}_{i,t}d_{i,t}$ in (3), we obtain an approximate expression for the gradient of the cost function in (3) as

$$\left[\nabla_{\boldsymbol{w}} J_i(\boldsymbol{w})\right]^T \approx \boldsymbol{u}_{i,t}(\boldsymbol{u}_{i,t}^T\boldsymbol{w}_{i,t} - d_{i,t}) - \sum_{j\in\mathcal{N}_i\setminus\{i\}}\alpha_{i,j}(\boldsymbol{\phi}_j - \boldsymbol{w}_{i,t}). \tag{4}$$

Considering that we are optimizing a sum of two convex cost functions in (2) with the use of (4), we note that we can carry out the optimization using incremental solutions over (2) where the update is performed in two steps. Since we consider the adapt-then-combine (ATC) diffusion strategy for this paper, first we create an intermediate estimate by using the gradient of the first summand in (2) and then update the estimate using the second summand in (2) as [17]

$$\boldsymbol{\phi}_{i,t+1} = \boldsymbol{w}_{i,t} + \mu_i\boldsymbol{u}_{i,t}(d_{i,t} - \boldsymbol{u}_{i,t}^T\boldsymbol{w}_{i,t}), \tag{5}$$

$$\boldsymbol{w}_{i,t+1} = \boldsymbol{\phi}_{i,t+1} + \eta_i\sum_{j\in\mathcal{N}_i\setminus\{i\}}\alpha_{i,j}(\boldsymbol{\phi}_{j,t+1} - \boldsymbol{\phi}_{i,t+1}), \tag{6}$$

where $\mu_i$ and $\eta_i$ are positive step sizes. Note that we have replaced the estimates coming from neighbors $\boldsymbol{\phi}_j$ with their instantaneous approximations $\boldsymbol{\phi}_{j,t+1}$. Now, we represent the equation in (6) as

$$\boldsymbol{w}_{i,t+1} = \sum_{j\in\mathcal{N}_i} p_{i,j}\boldsymbol{\phi}_{j,t+1}, \tag{7}$$

where we have defined $p_{i,i} = (1 - \sum_{j\in\mathcal{N}_i\setminus\{i\}}\eta_i\alpha_{i,j})$ and $p_{i,j} = \eta_i\alpha_{i,j}$ for $j \neq i$ to obtain (7), yielding the network matrix $\boldsymbol{P} = [p_{i,j}]$ comprised of the combination weights $\sum_{j=1}^N p_{i,j} = 1$ with $p_{i,j} \geq 0$.

## 3. Distributed estimation with level triggered sampling

The well-known ATC full diffusion scheme (7) requires all nodes in the network to communicate their current estimates (i) in their entirety, and (ii) at a fixed rate to all their neighboring nodes [17]. We propose a new scheme, which achieves an increased communication efficiency by conditioning the diffusion of information on the *trigger of an event*, instead of relying on a fixed rate of diffusion. Our approach considerably reduces the load on communication resources since only "significant changes" in the diffused parameter, e.g., an abrupt change in the local estimate, are conveyed based on the particular realization of the signal.

To clarify the framework, we consider the diffusion of a scalar parameter $\xi_{i,t}$ from a given node $i$ to a neighboring node $j$. As an example, this information can be a single component of the estimates [23], or the error associated with an additional estimation layer [22]. In our distributed framework, due to communication constraints, a quantized version of the original parameter, $\xi_{i,t}^q$ is

---

[1] We represent vectors (matrices) by bold lower (upper) case letters. For a vector $\boldsymbol{a}$ (a matrix $\boldsymbol{A}$), $\boldsymbol{a}^T$ ($\boldsymbol{A}^T$) is the transpose. $\|\boldsymbol{a}\|$ represents the Euclidean norm. The diag$\{\boldsymbol{A}\}$ returns a new matrix with only the main diagonal of $\boldsymbol{A}$ while diag$\{a\}$ puts $a$ on the main diagonal of the new matrix. col$\{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_N\}$ produces a column vector formed by column-wise stacking its arguments on top of one another. $\boldsymbol{I}_M$ represents the $M \times M$ identity matrix. $\otimes$ stands for the Kronecker product, Tr$\{\cdot\}$ stands for the trace.
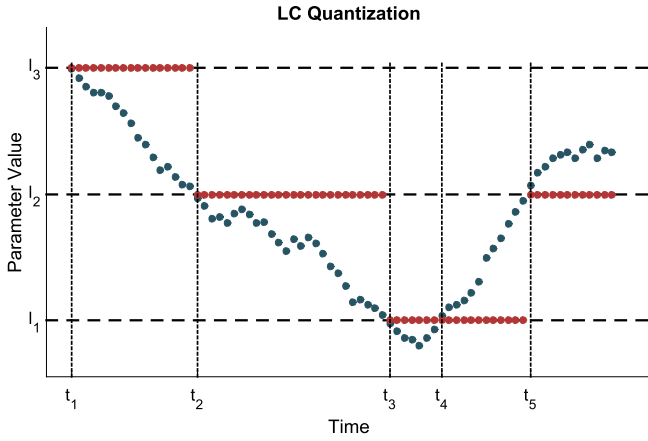
**Fig. 2.** Illustration of the operation of the LC quantizer. Blue dots represent the original node estimates, while red ones represent the quantized version of the corresponding estimates. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

shared. We aim to form a quantization scheme, which guarantees that $\xi_{i,t}$ and $\xi_{i,t}^q$ are approximately equal to each other for all $t$, while at the same time keeping the load on communication resources relatively small.

To solve this problem, we propose an event-triggered communication algorithm where, as the event-triggered approach, we specifically use level crossing (LC) quantization [19]. To clarify the framework, suppose we have a discrete time signal $\xi_{i,t}$ as shown in Fig. 2 that represents the information to be communicated from the node $i$ to the node $j$, e.g., the estimated parameter, or the estimation error. In conventional quantization, at each time instant, we sample and quantize this parameter. On the other hand, in the LC quantization, we consider a set of levels $\mathcal{S} \triangleq \{l_1, \ldots, l_K\}$, which is illustrated in Fig. 2. At each discrete time index $t$, the node $i$ checks whether a level-crossing has occurred on $\xi_{i,t}$. When the parameter $\xi_{i,t}$ crosses a level $l_{i,t}$, i.e.,

$$\left(\xi_{i,t-1} - l_{i,t}\right)\left(\xi_{i,t} - l_{i,t}\right) < 0 \quad \text{for some } l_{i,t} \in \mathcal{S},$$

the node $i$ transmits information to its neighboring nodes. For example, this information can be the direction of the level-crossing [19]. A neighboring node $j$ uses this received information to form an estimate $\xi_{i,t}^q$ for $\xi_{i,t}$.

If there is an information transfer by the node $i$ at time $t$, the receiving node $j$ estimates the parameter as the level through which a level crossing has occurred:

$$\xi_{i,t}^q = l_{i,t}. \tag{8}$$

For the time instants when the node $i$ is silent, the node $j$ infers that no significant change in the parameter has taken place, and uses the estimated parameter value from the previous time instant:

$$\xi_{i,t}^q = \xi_{i,t-1}^q. \tag{9}$$

We note that the set of levels $\mathcal{S}$ is known by all nodes in the network. Hence, as the diffused information, it is sufficient for the node $i$ to only convey how $\xi_{i,t}^q$ changes compared to the previously-crossed level $\xi_{i,t-1}^q$. In particular, we note the following two cases: In the first case, the parameter $\xi_{i,t}$ changes slowly enough such that a crossing through multiple levels do not occur, so that the node $i$ only needs to indicate the *direction* of the change in levels. Therefore, we transmit two bits for this case, one for indicating that the single level crossing occurs and the other for indicating the direction of crossing. In the second case, we may have multiple crossings where we directly code the full location

information of the new level value $\xi_{i,t}^q$ with a flag bit indicating multiple level crossing occurred using $\lceil \log_2(K) \rceil + 1$ bits. As shown, this approach significantly lowers the amount of communication while maintaining estimation performance.

## 4. Algorithm description

In this section, we present the full algorithmic description of the proposed diffusion scheme with the level-crossing quantization [19]. At time $t$, a given node $i$ in the network makes the scalar observation $d_{i,t}$ through the linear model $d_{i,t} = \boldsymbol{u}_{i,t}^T \boldsymbol{w_o} + v_{i,t}$, which is then used to update its intermediary local estimate using the LMS adaptation

$$\boldsymbol{\phi}_{i,t+1} = (\boldsymbol{I}_M - \mu_i \boldsymbol{u}_{i,t} \boldsymbol{u}_{i,t}^T)\boldsymbol{w}_{i,t} + \mu_i \boldsymbol{u}_{i,t} d_{i,t}.$$

Due to the quantized communication framework, a neighboring node $j$ does not have access to the true value of the parameter $\boldsymbol{\phi}_{i,t+1}$, which has $M$ entries. As such, based on the limited information it receives from the node $i$, the node $j$ tries to estimate this parameter as the $M$-entry vector $\boldsymbol{\phi}_{i,t+1}^q$. Specifically, in the LC quantization, the node $j$ receives information about how the current values of the *entries* of the parameter $\boldsymbol{\phi}_{i,t+1}$ have changed relative to the most recent estimate the node $j$ has access to, namely $\boldsymbol{\phi}_{i,t}^q$. The node $i$ records the most recent estimate, $\boldsymbol{\phi}_{i,t}^q$, as a reference and diffuses information to the neighboring nodes $j \in \mathcal{N}_i$ indicating how the current estimate $\boldsymbol{\phi}_{i,t+1}$ compares to this reference on a per-entry basis. In particular, the node $i$ makes this comparison by checking for a level crossing between corresponding entries of the two vector quantities $\boldsymbol{\phi}_{i,t}^q$ and $\boldsymbol{\phi}_{i,t+1}$. If there is a level crossing on an entry, the node $i$ transmits information to its neighbors through a channel frequency allocated to this particular entry. If there is a single level-crossing, this information indicates the direction of the level crossing; otherwise, the transmitted information directly specifies the location of the new level. A neighboring node $j$ then constructs the estimate $\boldsymbol{\phi}_{i,t+1}^q$ using (8) or (9) on a per-entry basis, depending on whether the node $i$ diffuses information or not, respectively, at time $t$.

While diffusing information related to its own local estimate, the node $i$ also receives information from the neighboring nodes $j$ representing their local estimates $\boldsymbol{\phi}_{j,t+1}$. For each neighboring node $j$, the node $i$ uses this diffused information to reconstruct $\boldsymbol{\phi}_{j,t+1}^q$ using (8) or (9). The final estimate $\boldsymbol{w}_{i,t+1}$ is then constructed using the combination

$$\boldsymbol{w}_{i,t+1} = p_{i,i}\boldsymbol{\phi}_{i,t+1} + \sum_{j \in \mathcal{N}_i \setminus \{i\}} p_{i,j}\boldsymbol{\phi}_{j,t+1}^q.$$

**Remark.** In order to keep the presentation clear, we illustrate the special case of $M = 1$ of the proposed algorithm in Algorithm 1, which can be generalized to arbitrary $M$ in a straightforward manner.

**Remark.** We note that an alternative approach to dealing with the $M > 1$ case is to have the nodes in the network transmit only a certain entry of their intermediary estimates $\boldsymbol{\phi}_{i,t}$. As an example, in this case, the nodes can cycle through different entries across time in a round-robin fashion. The non-communicated entries are replaced by the corresponding entries in the local intermediary estimate [23]. This approach is explored in Section 7.

## 5. Mean stability analysis

To continue with the stability analysis of the proposed scheme, we assume that the regressors $\boldsymbol{u}_{i,t}$ are temporally and spatially

**Algorithm 1** ATC diffusion LMS with the LC quantization, $M = 1$.

1: **for** $i = 1$ to $N$ **do**

    *Initialization:*

2:     $w_{i,0} = \phi_{i,0}^q = 0$

3: **end for**

4: **for** $t \geq 0$ **do**

5:     **for** $i = 1$ to $N$ **do**

       *Local adaptation:*

6:        $\phi_{i,t+1} = (1 - \mu_i u_{i,t}^2) w_{i,t} + \mu_i u_{i,t} d_{i,t}$

       *Check for level crossing:*

7:        **if** $\exists l_{i,t} \in \mathcal{S}$ such that

          $(\phi_{i,t}^q - l_{i,t})(\phi_{i,t+1} - l_{i,t}) < 0$ **then**

8:          **if** The crossing is to an adjacent level **then**

9:             Diffuse the direction of the crossing

10:          **else**

11:             Diffuse the location of the new level

12:          **end if**

13:          Locally store $\phi_{i,t+1}^q = l_{i,t}$ in record

14:        **else**

15:          Remain silent

16:          Locally set $\phi_{i,t+1}^q = \phi_{i,t}^q$

17:        **end if**

       *Reconstruction:*

18:        **for all** $j \in \mathcal{N}_i \setminus \{i\}$ **do**

19:          **if** node $j$ is silent **then**

20:             Reconstruct as $\phi_{j,t+1}^q = \phi_{j,t}^q$

21:          **else**

22:             Reconstruct $\phi_{j,t+1}^q$ using the diffused information

23:          **end if**

24:        **end for**

       *Combination:*

25:        $w_{i,t+1} = p_{i,i}\phi_{i,t+1} + \sum_{j \in \mathcal{N}_i \setminus \{i\}} p_{i,j}\phi_{j,t+1}^q$

26:     **end for**

27: **end for**

---

independent, zero mean and white, with covariance matrix $\mathbf{\Lambda}_i \triangleq E\left[\mathbf{u}_{i,t}\mathbf{u}_{i,t}^T\right] = \sigma_{u,i}^2 \mathbf{I}_M$. The observation $d_{i,t}$ at node $i$ is assumed to follow a linear model of the form

$$d_{i,t} = \mathbf{u}_{i,t}^T \mathbf{w_o} + v_{i,t}, \qquad (10)$$

where $\{v_{i,t}\}_{t\geq 1}$ is a zero mean white Gaussian noise process with variance $\sigma_{v,i}^2$, independent of $\{\mathbf{u}_{j,t}\}_{t\geq 1}$ $\forall i, j$.

In our proposed level-triggered estimation framework, at each node $i$, the diffusion LMS update for the ATC strategy takes the form

$$\boldsymbol{\phi}_{i,t+1} = (\mathbf{I}_M - \mu_i \mathbf{u}_{i,t}\mathbf{u}_{i,t}^T)\mathbf{w}_{i,t} + \mu_i \mathbf{u}_{i,t}d_{i,t}, \qquad (11)$$

$$\mathbf{w}_{i,t+1} = p_{i,i}\boldsymbol{\phi}_{i,t+1} + \sum_{j \in \mathcal{N}_i \setminus \{i\}} p_{i,j}\boldsymbol{\phi}_{j,t+1}^q, \qquad (12)$$

where the combination matrix $\mathbf{P}$ is taken to be stochastic, with its rows summing up to unity. We rewrite the expressions (11) and (12) as

$$\boldsymbol{\phi}_{i,t+1} = (\mathbf{I}_M - \mu_i \mathbf{u}_{i,t}\mathbf{u}_{i,t}^T)\mathbf{w}_{i,t} + \mu_i \mathbf{u}_{i,t}d_{i,t}, \qquad (13)$$

$$\mathbf{w}_{i,t+1} = \sum_{j \in \mathcal{N}_i} p_{i,j}\boldsymbol{\phi}_{j,t+1} - \sum_{j \in \mathcal{N}_i \setminus \{i\}} p_{i,j}\boldsymbol{\alpha}_{j,t+1}, \qquad (14)$$

by defining the quantization error for node $j$

$$\boldsymbol{\alpha}_{j,t} \triangleq \boldsymbol{\phi}_{j,t} - \boldsymbol{\phi}_{j,t}^q.$$

We represent the diffusion update over the network $\mathcal{N}$ in state-space form by introducing the following global quantities:

$$\mathbf{d}_t \triangleq \mathrm{col}\{d_{1,t}, \ldots, d_{N,t}\} \qquad \mathbf{v}_t \triangleq \mathrm{col}\{v_{1,t}, \ldots, v_{N,t}\}$$

$$\underline{\mathbf{w}_o} \triangleq \mathrm{col}\{\mathbf{w_o}, \ldots, \mathbf{w_o}\} \qquad \mathbf{U}_t \triangleq \mathrm{diag}\{\mathbf{u}_{1,t}, \ldots, \mathbf{u}_{N,t}\}$$

$$\mathbf{M} \triangleq \mathrm{diag}\{\mu_1 \mathbf{I}_M, \ldots, \mu_N \mathbf{I}_M\} \qquad \mathbf{w}_t \triangleq \mathrm{col}\{\mathbf{w}_{1,t}, \ldots, \mathbf{w}_{N,t}\}$$

$$\boldsymbol{\phi}_t \triangleq \mathrm{col}\{\boldsymbol{\phi}_{1,t}, \ldots, \boldsymbol{\phi}_{N,t}\} \qquad \boldsymbol{\phi}_t^q \triangleq \mathrm{col}\{\boldsymbol{\phi}_{1,t}^q, \ldots, \boldsymbol{\phi}_{N,t}^q\}$$

$$\boldsymbol{\alpha}_t \triangleq \mathrm{col}\{\boldsymbol{\alpha}_{1,t}, \ldots, \boldsymbol{\alpha}_{N,t}\} \qquad \mathbf{G} \triangleq \mathbf{P} \otimes \mathbf{I}_M$$

$$\mathbf{P_C} \triangleq \mathbf{P} - \mathrm{diag}\{\mathbf{P}\} \qquad \mathbf{G_C} \triangleq \mathbf{P_C} \otimes \mathbf{I}_M$$

Using the above-defined quantities, the diffusion updates (13), (14) take the following global state-space form:

$$\boldsymbol{\phi}_{t+1} = (\mathbf{I}_{MN} - \mathbf{M}\mathbf{U}_t\mathbf{U}_t^T)\mathbf{w}_t + \mathbf{M}\mathbf{U}_t\mathbf{d}_t, \qquad (15)$$

$$\mathbf{w}_{t+1} = \mathbf{G}\boldsymbol{\phi}_{t+1} - \mathbf{G_C}\boldsymbol{\alpha}_{t+1}. \qquad (16)$$

Similarly, the data model (10) can be expressed in terms of the global quantities as

$$\mathbf{d}_t = \mathbf{U}_t^T \underline{\mathbf{w}_o} + \mathbf{v}_t. \qquad (17)$$

To facilitate the mean stability analysis, we define the global deviation parameters

$$\tilde{\mathbf{w}}_t \triangleq \underline{\mathbf{w}_o} - \mathbf{w}_t,$$

$$\tilde{\boldsymbol{\phi}}_t \triangleq \underline{\mathbf{w}_o} - \boldsymbol{\phi}_t.$$

After substituting (17) and subtracting both sides of (15), (16) from $\underline{\mathbf{w}_o}$, the diffusion updates in terms of the deviation parameters take the following form:

$$\tilde{\boldsymbol{\phi}}_{t+1} = (\mathbf{I}_{MN} - \mathbf{M}\mathbf{U}_t\mathbf{U}_t^T)\tilde{\mathbf{w}}_t - \mathbf{M}\mathbf{U}_t\mathbf{v}_t, \qquad (18)$$

$$\tilde{\mathbf{w}}_{t+1} = \mathbf{G}\tilde{\boldsymbol{\phi}}_{t+1} + \mathbf{G_C}\boldsymbol{\alpha}_{t+1}, \qquad (19)$$

where we have used the relation $\mathbf{G}\underline{\mathbf{w}_o} = \underline{\mathbf{w}_o}$, which results from the stochastic nature of $\mathbf{P}$.

The expressions (18), (19) can be expressed compactly as

$$\tilde{\mathbf{w}}_{t+1} = \mathbf{G}(\mathbf{I}_{MN} - \mathbf{M}\mathbf{U}_t\mathbf{U}_t^T)\tilde{\mathbf{w}}_t - \mathbf{G}\mathbf{M}\mathbf{U}_t\mathbf{v}_t + \mathbf{G_C}\boldsymbol{\alpha}_{t+1}. \qquad (20)$$

**Assumption.** The quantization error over the network $\boldsymbol{\alpha}_t$ has zero mean. This is a reasonable assumption for the analysis of quantization effects [24]. The applicability of the assumption is verified by our experiments in Section 7.

Taking expectations of both sides of (20) yields

$$E\left[\tilde{\mathbf{w}}_{t+1}\right] = \mathbf{G}(\mathbf{I}_{MN} - \mathbf{M}\mathbf{\Lambda})\, E\left[\tilde{\mathbf{w}}_t\right], \qquad (21)$$

where $\mathbf{\Lambda} \triangleq \mathrm{diag}\{\mathbf{\Lambda}_1, \ldots, \mathbf{\Lambda}_N\}$ is block diagonal. For mean stability and asymptotic unbiasedness of the distributed filter (11)–(12), we require that the spectral radius $|\mathbf{G}(\mathbf{I}_{MN} - \mathbf{M}\mathbf{\Lambda})| < 1$, which, noting that $\mathbf{G}$ is stochastic with nonnegative entries, is equivalent to requiring

$$|(\mathbf{I}_{MN} - \mathbf{M}\mathbf{\Lambda})| < 1, \qquad (22)$$

by the Theorem 4.4 of [25]. Noting that the eigenvalues of the block diagonal matrix $\mathbf{I}_{MN} - \mathbf{M}\mathbf{\Lambda}$ is the union of the eigenvalues of its individual blocks $\mathbf{I}_M - \mu_i\mathbf{\Lambda}_i$ where $\mathbf{\Lambda}_i = \sigma_{u,i}^2\mathbf{I}_M$; we conclude that the distributed filter is mean stable if $|1 - \mu_i\sigma_{u,i}^2| < 1$, $i = 1, \ldots, N$, i.e., if

$$0 < \mu_i < \frac{2}{\sigma_{u,i}^2} \qquad i = 1, \ldots, N,$$

which provides the stability condition of the proposed algorithm.

## 6. Mean-square stability

We utilize the weighted energy relation approach [24] to proceed the mean square transient analysis of the distributed filter. Through a positive-definite weighting matrix $\boldsymbol{\Sigma}$, taking the weighted norm of both sides of (20) yields:

$$
\begin{aligned}
\tilde{\boldsymbol{w}}_{t+1}^T \boldsymbol{\Sigma} \tilde{\boldsymbol{w}}_{t+1} = \\
\quad \tilde{\boldsymbol{w}}_t^T (\boldsymbol{I}_{MN} - \boldsymbol{M} \boldsymbol{U}_t \boldsymbol{U}_t^T)^T \boldsymbol{G}^T \boldsymbol{\Sigma} \boldsymbol{G} (\boldsymbol{I}_{MN} - \boldsymbol{M} \boldsymbol{U}_t \boldsymbol{U}_t^T) \tilde{\boldsymbol{w}}_t \\
\quad - 2 \boldsymbol{v}_t^T \boldsymbol{U}_t^T \boldsymbol{M} \boldsymbol{G}^T \boldsymbol{\Sigma} \boldsymbol{G} (\boldsymbol{I}_{MN} - \boldsymbol{M} \boldsymbol{U}_t \boldsymbol{U}_t^T) \tilde{\boldsymbol{w}}_t \\
\quad + 2 \boldsymbol{\alpha}_{t+1}^T \boldsymbol{G}_{\boldsymbol{C}}^T \boldsymbol{\Sigma} \boldsymbol{G} (\boldsymbol{I}_{MN} - \boldsymbol{M} \boldsymbol{U}_t \boldsymbol{U}_t^T) \tilde{\boldsymbol{w}}_t \\
\quad - 2 \boldsymbol{v}_t^T \boldsymbol{U}_t^T \boldsymbol{M} \boldsymbol{G}^T \boldsymbol{\Sigma} \boldsymbol{G}_{\boldsymbol{C}} \boldsymbol{\alpha}_{t+1} \\
\quad + \boldsymbol{v}_t^T \boldsymbol{U}_t^T \boldsymbol{M} \boldsymbol{G}^T \boldsymbol{\Sigma} \boldsymbol{G} \boldsymbol{M} \boldsymbol{U}_t \boldsymbol{v}_t \\
\quad + \boldsymbol{\alpha}_{t+1}^T \boldsymbol{G}_{\boldsymbol{C}}^T \boldsymbol{\Sigma} \boldsymbol{G}_{\boldsymbol{C}} \boldsymbol{\alpha}_{t+1}.
\end{aligned} \tag{23}
$$

Noting that $\boldsymbol{v}_t$ is zero-mean and independent of $\boldsymbol{U}_t$ and $\tilde{\boldsymbol{w}}_t$, and taking the expected value of both sides of (23) yields the following variance relation:

$$
\begin{aligned}
E \| \tilde{\boldsymbol{w}}_{t+1} \|_{\boldsymbol{\Sigma}}^2 = E \| \tilde{\boldsymbol{w}}_t \|_{\boldsymbol{\Sigma}'}^2 \\
\quad + 2 E \left[ \boldsymbol{\alpha}_{t+1}^T \boldsymbol{G}_{\boldsymbol{C}}^T \boldsymbol{\Sigma} \boldsymbol{G} (\boldsymbol{I}_{MN} - \boldsymbol{M} \boldsymbol{U}_t \boldsymbol{U}_t^T) \tilde{\boldsymbol{w}}_t \right] \\
\quad - 2 E \left[ \boldsymbol{v}_t^T \boldsymbol{U}_t^T \boldsymbol{M} \boldsymbol{G}^T \boldsymbol{\Sigma} \boldsymbol{G}_{\boldsymbol{C}} \boldsymbol{\alpha}_{t+1} \right] \\
\quad + E \left[ \boldsymbol{v}_t^T \boldsymbol{U}_t^T \boldsymbol{M} \boldsymbol{G}^T \boldsymbol{\Sigma} \boldsymbol{G} \boldsymbol{M} \boldsymbol{U}_t \boldsymbol{v}_t \right] \\
\quad + E \left[ \boldsymbol{\alpha}_{t+1}^T \boldsymbol{G}_{\boldsymbol{C}}^T \boldsymbol{\Sigma} \boldsymbol{G}_{\boldsymbol{C}} \boldsymbol{\alpha}_{t+1} \right],
\end{aligned} \tag{24}
$$

where

$$
\begin{aligned}
\boldsymbol{\Sigma}' \triangleq \boldsymbol{G}^T \boldsymbol{\Sigma} \boldsymbol{G} - \boldsymbol{G}^T \boldsymbol{\Sigma} \boldsymbol{G} \boldsymbol{M} \boldsymbol{U}_t \boldsymbol{U}_t^T - \boldsymbol{U}_t \boldsymbol{U}_t^T \boldsymbol{M} \boldsymbol{G}^T \boldsymbol{\Sigma} \boldsymbol{G} \\
\quad + \boldsymbol{U}_t \boldsymbol{U}_t^T \boldsymbol{M} \boldsymbol{G}^T \boldsymbol{\Sigma} \boldsymbol{G} \boldsymbol{M} \boldsymbol{U}_t \boldsymbol{U}_t^T.
\end{aligned}
$$

By the temporal independence of the regressor process $\boldsymbol{U}_t$ and the independence of the noise process $\boldsymbol{v}_t$ from $\boldsymbol{U}_t$, we have the result that $\boldsymbol{U}_t$ is independent of $\tilde{\boldsymbol{w}}_t$. Hence, the random weighting matrix $\boldsymbol{\Sigma}'$ can be replaced by its mean value $\boldsymbol{\Sigma}' \triangleq E \left[ \boldsymbol{\Sigma}' \right]$ in (24). Thus,

$$
\begin{aligned}
\boldsymbol{\Sigma}' = \boldsymbol{G}^T \boldsymbol{\Sigma} \boldsymbol{G} - \boldsymbol{G}^T \boldsymbol{\Sigma} \boldsymbol{G} \boldsymbol{M} \boldsymbol{\Lambda} - \boldsymbol{\Lambda} \boldsymbol{M} \boldsymbol{G}^T \boldsymbol{\Sigma} \boldsymbol{G} \\
\quad + E \left[ \boldsymbol{U}_t \boldsymbol{U}_t^T \boldsymbol{M} \boldsymbol{G}^T \boldsymbol{\Sigma} \boldsymbol{G} \boldsymbol{M} \boldsymbol{U}_t \boldsymbol{U}_t^T \right],
\end{aligned} \tag{25}
$$

where $\boldsymbol{\Lambda} \triangleq E \left[ \boldsymbol{U}_t \boldsymbol{U}_t^T \right]$. Substituting the $\tilde{\phi}_{t+1}$ expression from (18) into (24) yields the following final form of the variance relation

$$
\begin{aligned}
E \| \tilde{\boldsymbol{w}}_{t+1} \|_{\boldsymbol{\Sigma}}^2 = E \| \tilde{\boldsymbol{w}}_t \|_{\boldsymbol{\Sigma}'}^2 \\
\quad + 2 E \left[ \boldsymbol{\alpha}_{t+1}^T \boldsymbol{G}_{\boldsymbol{C}}^T \boldsymbol{\Sigma} \boldsymbol{G} \tilde{\phi}_{t+1} \right] \\
\quad + E \left[ \boldsymbol{\alpha}_{t+1}^T \boldsymbol{G}_{\boldsymbol{C}}^T \boldsymbol{\Sigma} \boldsymbol{G}_{\boldsymbol{C}} \boldsymbol{\alpha}_{t+1} \right] \\
\quad + E \left[ \boldsymbol{v}_t^T \boldsymbol{U}_t^T \boldsymbol{M} \boldsymbol{G}^T \boldsymbol{\Sigma} \boldsymbol{G} \boldsymbol{M} \boldsymbol{U}_t \boldsymbol{v}_t \right].
\end{aligned} \tag{26}
$$

To capture the mean-square behavior of the adaptive network, we express the relations (25), (26) in a compact form by using the convenient vector notation [24]. In particular, we use the bvec$\{\cdot\}$ block vectorization operation [16] which transforms an arbitrary $MN \times MN$ block matrix $\boldsymbol{\Sigma}$ with the $(i, j)$th block $\boldsymbol{\Sigma}_{ij}$ of size $M \times M$ into the vector col$\{\boldsymbol{\sigma}_1, \ldots, \boldsymbol{\sigma}_N\}$, where $\boldsymbol{\sigma}_j \triangleq \text{col} \left\{ \text{vec}\{\boldsymbol{\Sigma}_{1j}\}, \ldots, \text{vec}\{\boldsymbol{\Sigma}_{Nj}\} \right\}$. We also use the block Kronecker product $\boldsymbol{A} \odot \boldsymbol{B}$ defined as having the $(i, j)$th block

$$
[\boldsymbol{A} \odot \boldsymbol{B}]_{ij} = \begin{bmatrix} \boldsymbol{A}_{ij} \otimes \boldsymbol{B}_{11} & \ldots & \boldsymbol{A}_{ij} \otimes \boldsymbol{B}_{1N} \\ \vdots & \ddots & \vdots \\ \boldsymbol{A}_{ij} \otimes \boldsymbol{B}_{N1} & \ldots & \boldsymbol{A}_{ij} \otimes \boldsymbol{B}_{NN} \end{bmatrix}, \tag{27}
$$

which is related to the bvec$\{\cdot\}$ operator via bvec$\{ABC\} = (C^T \odot A) \text{bvec}\{B\}$. Defining $\boldsymbol{\sigma} \triangleq \text{bvec}\{\boldsymbol{\Sigma}\}$ and vectorizing both sides of (25) yields

$$
\begin{aligned}
\text{bvec}\{\boldsymbol{\Sigma}'\} = ((\boldsymbol{I}_{MN} \odot \boldsymbol{I}_{MN}) - (\boldsymbol{\Lambda} \boldsymbol{M} \odot \boldsymbol{I}_{MN}) \\
\quad - (\boldsymbol{I}_{MN} \odot \boldsymbol{\Lambda} \boldsymbol{M})) (\boldsymbol{G}^T \odot \boldsymbol{G}^T) \boldsymbol{\sigma} \\
\quad + \text{bvec}\{E \left[ \boldsymbol{U}_t \boldsymbol{U}_t^T \boldsymbol{M} \boldsymbol{G}^T \boldsymbol{\Sigma} \boldsymbol{G} \boldsymbol{M} \boldsymbol{U}_t \boldsymbol{U}_t^T \right]\}.
\end{aligned} \tag{28}
$$

The term $E \left[ \boldsymbol{U}_t \boldsymbol{U}_t^T \boldsymbol{M} \boldsymbol{G}^T \boldsymbol{\Sigma} \boldsymbol{G} \boldsymbol{M} \boldsymbol{U}_t \boldsymbol{U}_t^T \right]$ on the right-hand side of (28) can be vectorized by resorting to the Gaussian factorization theorem [16,17]. We let $\tilde{\boldsymbol{\Sigma}} = \boldsymbol{M} \boldsymbol{G}^T \boldsymbol{\Sigma} \boldsymbol{G} \boldsymbol{M}$ with $(i, j)$th block $\tilde{\boldsymbol{\Sigma}}_{i,j}$ and with the vectorized form bvec$\{\tilde{\boldsymbol{\Sigma}}\} = \text{col}\left\{ \tilde{\boldsymbol{\sigma}}_1, \ldots, \tilde{\boldsymbol{\sigma}}_j \right\}$ where $\tilde{\boldsymbol{\sigma}}_j = \text{col}\left\{ \tilde{\boldsymbol{\sigma}}_{1j}, \ldots, \tilde{\boldsymbol{\sigma}}_{Nj} \right\}$. Then, the $(k, l)$th block $\boldsymbol{\Gamma}_{kl}$ of $\boldsymbol{\Gamma} \triangleq E \left[ \boldsymbol{U}_t \boldsymbol{U}_t^T \tilde{\boldsymbol{\Sigma}} \boldsymbol{U}_t \boldsymbol{U}_t^T \right]$ is given by

$$
\boldsymbol{\Gamma}_{kl} = \begin{cases} \boldsymbol{\Lambda}_k \tilde{\boldsymbol{\Sigma}}_{kl} \boldsymbol{\Lambda}_l & \text{for } k \neq l, \\ \boldsymbol{\Lambda}_k \tilde{\boldsymbol{\Sigma}}_{kl} \boldsymbol{\Lambda}_k + 2 \boldsymbol{\Lambda}_k \text{Tr}\{\tilde{\boldsymbol{\Sigma}}_{kk} \boldsymbol{\Lambda}_k\} & \text{for } k = l, \end{cases}
$$

with the vectorized form

$$
\boldsymbol{\gamma}_{kl} = \begin{cases} (\boldsymbol{\Lambda}_l \otimes \boldsymbol{\Lambda}_k) \tilde{\boldsymbol{\sigma}}_{kl} & \text{for } k \neq l, \\ ((\boldsymbol{\Lambda}_l \otimes \boldsymbol{\Lambda}_k) + 2 \boldsymbol{r}_k \boldsymbol{r}_k^T) \tilde{\boldsymbol{\sigma}}_{kl} & \text{for } k = l, \end{cases}
$$

by the factorization theorem, where $\boldsymbol{\Lambda}_k \triangleq E \left[ \boldsymbol{u}_{k,t} \boldsymbol{u}_{k,t}^T \right]$, $\boldsymbol{r}_k \triangleq \text{vec}\{\boldsymbol{\Lambda}_k\}$. Letting bvec$\{\boldsymbol{\Gamma}\} = \text{col}\left\{ \boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_j \right\}$ where $\boldsymbol{\gamma}_j = \text{col}\left\{ \boldsymbol{\gamma}_{1j}, \ldots, \boldsymbol{\gamma}_{Nj} \right\}$, we observe that we can express $\boldsymbol{\gamma}_j$ in the form

$$
\boldsymbol{\gamma}_j = \mathcal{A}_j \tilde{\boldsymbol{\sigma}}_j,
$$

where $\mathcal{A}_j \triangleq \text{diag}\left\{ \boldsymbol{\Lambda}_j \otimes \boldsymbol{\Lambda}_1, \ldots (\boldsymbol{\Lambda}_j \otimes \boldsymbol{\Lambda}_j) + 2 \boldsymbol{r}_j \boldsymbol{r}_j^T, \ldots, \boldsymbol{\Lambda}_j \otimes \boldsymbol{\Lambda}_N \right\}$. Further defining $\mathcal{A} \triangleq \text{diag}\{\mathcal{A}_1, \ldots, \mathcal{A}_N\}$, we arrive at the representation

$$
\text{bvec}\{\boldsymbol{\Gamma}\} = \mathcal{A} \text{bvec}\{\tilde{\boldsymbol{\Sigma}}\} = \mathcal{A}(\boldsymbol{M} \odot \boldsymbol{M})(\boldsymbol{G}^T \odot \boldsymbol{G}^T) \boldsymbol{\sigma}. \tag{29}
$$

Substituting (29) to (28) yields

$$
\begin{aligned}
\text{bvec}\{\boldsymbol{\Sigma}'\} = ((\boldsymbol{I}_{MN} \odot \boldsymbol{I}_{MN}) - (\boldsymbol{\Lambda} \boldsymbol{M} \odot \boldsymbol{I}_{MN}) \\
\quad - (\boldsymbol{I}_{MN} \odot \boldsymbol{\Lambda} \boldsymbol{M}) + \mathcal{A}(\boldsymbol{M} \odot \boldsymbol{M})) (\boldsymbol{G}^T \odot \boldsymbol{G}^T) \boldsymbol{\sigma}.
\end{aligned} \tag{30}
$$

The term $E \left[ \boldsymbol{v}_t^T \boldsymbol{U}_t^T \boldsymbol{M} \boldsymbol{G}^T \boldsymbol{\Sigma} \boldsymbol{G} \boldsymbol{M} \boldsymbol{U}_t \boldsymbol{v}_t \right]$ in (26) can be verified to be

$$
\begin{aligned}
E \left[ \boldsymbol{v}_t^T \boldsymbol{U}_t^T \boldsymbol{M} \boldsymbol{G}^T \boldsymbol{\Sigma} \boldsymbol{G} \boldsymbol{M} \boldsymbol{U}_t \boldsymbol{v}_t \right] \\
\quad = E \left[ \text{Tr}\{\boldsymbol{v}_t^T \boldsymbol{U}_t^T \boldsymbol{M} \boldsymbol{G}^T \boldsymbol{\Sigma} \boldsymbol{G} \boldsymbol{M} \boldsymbol{U}_t \boldsymbol{v}_t\} \right] \\
\quad = E \left[ \text{Tr}\{\boldsymbol{\Sigma} \boldsymbol{G} \boldsymbol{M} \boldsymbol{U}_t \boldsymbol{v}_t \boldsymbol{v}_t^T \boldsymbol{U}_t^T \boldsymbol{M} \boldsymbol{G}^T\} \right] \\
\quad = \text{Tr}\{\boldsymbol{\Sigma} \boldsymbol{G} \boldsymbol{M} \boldsymbol{H} \boldsymbol{M} \boldsymbol{G}^T\},
\end{aligned} \tag{31}
$$

where we have defined $\boldsymbol{H} = E \left[ \boldsymbol{U}_t \boldsymbol{v}_t \boldsymbol{v}_t^T \boldsymbol{U}_t^T \right]$. We observe that $\boldsymbol{H}$ has the $(k, l)$th block $\boldsymbol{H}_{kl} = \sigma_{v,k}^2 \boldsymbol{\Lambda}_k \delta_{kl}$, which yields $\boldsymbol{H} = (\boldsymbol{\Lambda}_{\boldsymbol{v}} \otimes \boldsymbol{I}_M) \boldsymbol{\Lambda}$, where $\boldsymbol{\Lambda}_{\boldsymbol{v}} \triangleq E \left[ \boldsymbol{v}_t \boldsymbol{v}_t^T \right]$. Thus (31) becomes

$$
\begin{aligned}
E \left[ \boldsymbol{v}_t^T \boldsymbol{U}_t^T \boldsymbol{M} \boldsymbol{G}^T \boldsymbol{\Sigma} \boldsymbol{G} \boldsymbol{M} \boldsymbol{U}_t \boldsymbol{v}_t \right] \\
\quad = \text{Tr}\{\boldsymbol{\Sigma} \boldsymbol{G} \boldsymbol{M} (\boldsymbol{\Lambda}_{\boldsymbol{v}} \otimes \boldsymbol{I}_M) \boldsymbol{\Lambda} \boldsymbol{M} \boldsymbol{G}^T\} \\
\quad = ((\boldsymbol{G} \boldsymbol{M} \odot \boldsymbol{G} \boldsymbol{M}) \text{bvec}\{(\boldsymbol{\Lambda}_{\boldsymbol{v}} \otimes \boldsymbol{I}_M) \boldsymbol{\Lambda}\})^T \boldsymbol{\sigma}.
\end{aligned} \tag{32}
$$

Similarly the remaining terms in the RHS of (26) can be verified to be

$$E\left[\boldsymbol{\alpha}_{t+1}^T \boldsymbol{G}_C^T \boldsymbol{\Sigma} \boldsymbol{G} \tilde{\boldsymbol{\phi}}_{t+1}\right] = ((\boldsymbol{G} \odot \boldsymbol{G}_C)\, \mathrm{bvec}\{E[\boldsymbol{\alpha}_{t+1}\tilde{\boldsymbol{\phi}}_{t+1}^T]\})^T \boldsymbol{\sigma},$$

$$E\left[\boldsymbol{\alpha}_{t+1}^T \boldsymbol{G}_C^T \boldsymbol{\Sigma} \boldsymbol{G}_C \boldsymbol{\alpha}_{t+1}\right] = ((\boldsymbol{G}_C \odot \boldsymbol{G}_C)\, \mathrm{bvec}\{E[\boldsymbol{\alpha}_{t+1}\boldsymbol{\alpha}_{t+1}^T]\})^T \boldsymbol{\sigma}. \tag{33}$$

Defining the quantities

$$\boldsymbol{b}_t \triangleq (\boldsymbol{GM} \odot \boldsymbol{GM})\, \mathrm{bvec}\{(\boldsymbol{\Lambda_v} \otimes \boldsymbol{I}_M)\, \boldsymbol{\Lambda}\} + (\boldsymbol{G} \odot \boldsymbol{G}_C)\, \mathrm{bvec}\{E[\boldsymbol{\alpha}_t \tilde{\boldsymbol{\phi}}_t^T]\}$$
$$+ (\boldsymbol{G}_C \odot \boldsymbol{G}_C)\, \mathrm{bvec}\{E[\boldsymbol{\alpha}_t \boldsymbol{\alpha}_t^T]\},$$

$$\boldsymbol{F} \triangleq ((\boldsymbol{I}_{MN} \odot \boldsymbol{I}_{MN}) - (\boldsymbol{\Lambda M} \odot \boldsymbol{I}_{MN}) - (\boldsymbol{I}_{MN} \odot \boldsymbol{\Lambda M})$$
$$+ \mathcal{A}\,(\boldsymbol{M} \odot \boldsymbol{M}))\,(\boldsymbol{G}^T \odot \boldsymbol{G}^T), \tag{34}$$

and further using the shorthand $E\|\tilde{\boldsymbol{w}}_t\|_{\boldsymbol{\sigma}}^2$ for $E\|\tilde{\boldsymbol{w}}_t\|_{\mathrm{bvec}^{-1}(\boldsymbol{\sigma})}^2$, yields the following compact form for the weighted energy recursion:

$$E\|\tilde{\boldsymbol{w}}_{t+1}\|_{\boldsymbol{\sigma}}^2 = E\|\tilde{\boldsymbol{w}}_t\|_{\boldsymbol{F}\boldsymbol{\sigma}}^2 + \boldsymbol{b}_{t+1}^T \boldsymbol{\sigma} \tag{35}$$

**Remark.** We note that the expectations $E[\boldsymbol{\alpha}_{t+1}\tilde{\boldsymbol{\phi}}_{t+1}^T]$ and $E[\boldsymbol{\alpha}_{t+1}\boldsymbol{\alpha}_{t+1}^T]$ present some difficulty for further analytical simplifications in closed form, in exact or approximate terms. This is caused by the large degree with which the quantization error term $\boldsymbol{\alpha}_t$ is coupled with itself as well as the intermediary parameter deviation $\tilde{\boldsymbol{\phi}}_t$ nonlinearly through the non-deterministic reference levels $\{\boldsymbol{\phi}_{i,t'}^q\}_{t'\leq t}$ against which the level crossing events are checked, which evolve through (13)–(14). We further note that invoking an approximation based on independence arguments for $E[\boldsymbol{\alpha}_{t+1}\tilde{\boldsymbol{\phi}}_{t+1}^T]$, which captures the covariances between the intermediary parameter deviations and the quantization errors over arbitrary pairs of nodes on the network, is not feasible in general unless further assumptions are made on the number of quantization levels employed so that the deviations become statistically less sensitive on the error terms. We stress that the lack of closed-form expressions for these expectations does not hamper our analysis for the mean-square stability, since requiring that the aforementioned terms remain bounded is sufficient for the purposes of establishing a bound for the (weighted) mean-square deviation $E\|\tilde{\boldsymbol{w}}_t\|_{\boldsymbol{\sigma}}^2$.

Iteration of (35) yields the recursions

$$E\|\tilde{\boldsymbol{w}}_{t+1}\|_{\boldsymbol{\sigma}}^2 = E\|\tilde{\boldsymbol{w}}_t\|_{\boldsymbol{F}\boldsymbol{\sigma}}^2 + \boldsymbol{b}_{t+1}^T \boldsymbol{\sigma}$$

$$E\|\tilde{\boldsymbol{w}}_{t+1}\|_{\boldsymbol{F}\boldsymbol{\sigma}}^2 = E\|\tilde{\boldsymbol{w}}_t\|_{\boldsymbol{F}^2\boldsymbol{\sigma}}^2 + \boldsymbol{b}_{t+1}^T \boldsymbol{F}\boldsymbol{\sigma}$$

$$\vdots$$

$$E\|\tilde{\boldsymbol{w}}_{t+1}\|_{\boldsymbol{F}^{N^2M^2-1}\boldsymbol{\sigma}}^2 = E\|\tilde{\boldsymbol{w}}_t\|_{\boldsymbol{F}^{N^2M^2}\boldsymbol{\sigma}}^2 + \boldsymbol{b}_{t+1}^T \boldsymbol{F}^{N^2M^2-1}\boldsymbol{\sigma}. \tag{36}$$

Using Cayley–Hamilton theorem with characteristic polynomial $p(x)$ for F results in

$$\boldsymbol{F}^{N^2M^2} = -p_{N^2M^2-1}\boldsymbol{F}^{N^2M^2-1} - \ldots - p_1\boldsymbol{F} - p_0.$$

Substituting to (36) then results in the expression

$$E\|\tilde{\boldsymbol{w}}_{t+1}\|_{\boldsymbol{F}^{N^2M^2-1}\boldsymbol{\sigma}}^2$$
$$= -p_{N^2M^2-1}E\|\tilde{\boldsymbol{w}}_t\|_{\boldsymbol{F}^{N^2M^2-1}\boldsymbol{\sigma}}^2 - \ldots - p_0 E\|\tilde{\boldsymbol{w}}_t\|_{\boldsymbol{\sigma}}^2$$
$$+ \boldsymbol{b}_{t+1}^T \boldsymbol{F}^{N^2M^2-1}\boldsymbol{\sigma},$$

which can be placed into the state space form

$$\mathcal{W}_{t+1} = \mathcal{F}\mathcal{W}_t + \mathcal{Y}_{t+1}, \tag{37}$$

where

$$\mathcal{W}_t \triangleq \begin{bmatrix} E\|\tilde{\boldsymbol{w}}_t\|_{\boldsymbol{\sigma}}^2 \\ E\|\tilde{\boldsymbol{w}}_t\|_{\boldsymbol{F}\boldsymbol{\sigma}}^2 \\ \vdots \\ E\|\tilde{\boldsymbol{w}}_t\|_{\boldsymbol{F}^{(N^2M^2-1)}\boldsymbol{\sigma}}^2 \end{bmatrix}, \quad \mathcal{Y}_t \triangleq \begin{bmatrix} \boldsymbol{b}_t^T \boldsymbol{\sigma} \\ \boldsymbol{b}_t^T \boldsymbol{F}\boldsymbol{\sigma} \\ \vdots \\ \boldsymbol{b}_t^T \boldsymbol{F}^{N^2M^2-1}\boldsymbol{\sigma} \end{bmatrix} \tag{38}$$

$$\mathcal{F} \triangleq \begin{bmatrix} 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -p_0 & -p_1 & -p_2 & \ldots & -p_{N^2M^2-1} \end{bmatrix}. \tag{39}$$

To make the mean-square stability analysis more tractable, we introduce the following assumption:

**Assumption.** The quantization error covariances $E\left[\boldsymbol{\alpha}_{t+1}\tilde{\boldsymbol{\phi}}_{t+1}^T\right]$ and $E\left[\boldsymbol{\alpha}_{t+1}\boldsymbol{\alpha}_{t+1}^T\right]$ remain bounded, with $\left\|E\left[\boldsymbol{\alpha}_{t+1}\tilde{\boldsymbol{\phi}}_{t+1}^T\right]\right\|_F$, $\left\|E\left[\boldsymbol{\alpha}_{t+1}\boldsymbol{\alpha}_{t+1}^T\right]\right\|_F < A$ for some $A > 0$ for the Frobenius norms.

Using the assumption, we obtain a bound the norm $\|\boldsymbol{b}_t\|_2$ as

$$\|\boldsymbol{b}_t\|_2 \leq \|(\boldsymbol{GM} \odot \boldsymbol{GM})\, \mathrm{bvec}\{(\boldsymbol{\Lambda_v} \otimes \boldsymbol{I}_M)\, \boldsymbol{\Lambda}\}\|_2$$
$$+ \|\boldsymbol{G} \odot \boldsymbol{G}_C\|_2 \left\|\mathrm{bvec}\{E[\boldsymbol{\alpha}_t \tilde{\boldsymbol{\phi}}_t^T]\}\right\|_2$$
$$+ \|\boldsymbol{G}_C \odot \boldsymbol{G}_C\|_2 \left\|\mathrm{bvec}\{E[\boldsymbol{\alpha}_t \boldsymbol{\alpha}_t^T]\}\right\|_2$$
$$\leq \|(\boldsymbol{GM} \odot \boldsymbol{GM})\, \mathrm{bvec}\{(\boldsymbol{\Lambda_v} \otimes \boldsymbol{I}_M)\, \boldsymbol{\Lambda}\}\|_2$$
$$+ A\,(\|\boldsymbol{P}\|_2 + \|\boldsymbol{P}_C\|_2)\,\|\boldsymbol{P}_C\|_2 \triangleq B.$$

Inspecting (39), we observe that the boundedness of $\|\boldsymbol{b}_t\|_2$ implies the boundedness of $\|\mathcal{Y}_t\|_2$, hence $\exists C > 0$ s.t. $\|\mathcal{Y}_t\|_2 < C \ \forall t$.

The recursion (37) can be solved for $\mathcal{W}_t$ in closed form as

$$\mathcal{W}_t = \mathcal{F}^t \mathcal{W}_0 + \sum_{n=0}^{t-1} \mathcal{F}^n \mathcal{Y}_{t-n}. \tag{40}$$

Using (40), we can obtain a bound for $\|\mathcal{W}_t\|_2$ as

$$\|\mathcal{W}_t\|_2 \leq \|\mathcal{F}\|_2^t \|\mathcal{W}_0\|_2 + \sum_{n=0}^{t-1} \|\mathcal{F}\|_2^n \|\mathcal{Y}_{t-n}\|_2$$
$$\leq \|\boldsymbol{F}\|_2^t \|\mathcal{W}_0\|_2 + C \sum_{n=0}^{t-1} \|\boldsymbol{F}\|_2^n$$
$$= \|\boldsymbol{F}\|_2^t \|\mathcal{W}_0\|_2 + C \frac{1 - \|\boldsymbol{F}\|_2^t}{1 - \|\boldsymbol{F}\|_2} \tag{41}$$

where we have used the fact that since $\mathcal{F}$ is in the form of a companion matrix for $\boldsymbol{F}$, they share the same set of eigenvalues.

We note that requiring that $\|\mathcal{W}_t\|_2$ remains bounded is sufficient to guarantee the mean-square stability of the overall system since doing so ensures that $E\|\tilde{\boldsymbol{w}}_t\|_{\boldsymbol{\sigma}}^2$ remains bounded. Thus, by (41), the mean-square stability condition reduces to the matrix $\boldsymbol{F}$ given by (34) being stable. Hence in order to ensure MS stability, it is sufficient that the step sizes $\mu_i$ are chosen such that the matrix $\boldsymbol{F}$ is stable.

## 7. Experiments

In this section, we demonstrate the significant reduction in the communication load achieved by our algorithms while providing equal performance with respect to the state of the art.

For the first part of the simulations, we consider a sample network consisting of $N = 10$ nodes, where each node makes its observation through the linear model

$$d_{i,t} = \boldsymbol{u}_{i,t}^T \boldsymbol{w_o} + v_{i,t}, \qquad i = 1, \ldots, N. \tag{42}$$

The regressor data $\boldsymbol{u}_{i,t}$ are zero mean i.i.d. Gaussian with standard deviations $\sigma_{u,i}$ chosen randomly from the interval $(0.3, 0.8)$. The observation noises are generated from a Normal distribution with standard deviations $\sigma_{v,i}$ chosen randomly from the interval $(0.1, 0.3)$. In Fig. 3, we depict the network topology and the network's statistical profile to show how the signal power and the noise power vary across the network.

The unknown vector parameter $\boldsymbol{w_o}$ with $M = 10$ components is randomly chosen from a Normal distribution and normalized to have a unit energy. We changed the source statistics in the middle of the simulations to observe how well the proposed algorithm is able to track the sudden changes in the unknown parameter.

We use Metropolis combination rule to generate the network matrix $\boldsymbol{P}$ such that

$$p_{i,j} = \begin{cases} \frac{2}{M^2} \frac{1}{\max(N_i, N_j)} & \text{if } i \neq j \text{ are linked,} \\ 0 & \text{for } i \text{ and } j \text{ not linked,} \\ 1 - \sum_{j \in \mathcal{N}_i \setminus i} p_{i,j} & \text{for } i = j \end{cases}$$

using the randomly selected network adjacency matrix given by

$$\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}.$$

We configure the nodes such that they cycle through the entries of the intermediary estimates $\boldsymbol{\phi}_{i,t}$ in a round-robin fashion, and exchange only this selected $L = 1$ dimension out of $M$ in one time instant. For instance, for a $L = 1, M = 3$ system at time instants $t = 1, \ldots, 4$, the $i$th node will send its entries of the intermediary estimate $\boldsymbol{\phi}_{i,t}$ as in (43)

$$\boldsymbol{\phi}_{1,i} = \begin{bmatrix} \phi_{1,1,i} \\ 0 \\ 0 \end{bmatrix}, \quad \boldsymbol{\phi}_{2,i} = \begin{bmatrix} 0 \\ \phi_{2,2,i} \\ 0 \end{bmatrix}, \quad \boldsymbol{\phi}_{3,i} = \begin{bmatrix} 0 \\ 0 \\ \phi_{3,3,i} \end{bmatrix},$$

$$\boldsymbol{\phi}_{4,i} = \begin{bmatrix} \phi_{1,4,i} \\ 0 \\ 0 \end{bmatrix}, \tag{43}$$

where $\phi_{l,t,i}$ is the $l$th dimension of the intermediary entry $\boldsymbol{\phi}_{i,t}$ of the $i$th node at time $t$ that is sent to the neighbors.

We evaluate the communication reduction performance of the proposed algorithm with respect to the algorithm in [23], where only one entry of intermediate estimates is exchanged by the nodes at each round in a sequential order as explained in (43).

In Fig. 4, the MSD performance of the proposed algorithm is demonstrated, where as a reference, we have considered the algorithm in [23] with an adaptive Lloyd–Max quantizer and with a no-quantization (scalar diffusion) implementation of the system. Note that both in scalar diffusion algorithm and Lloyd–Max quantized algorithm, which is referred as conventionally quantized algorithm later, nodes are exchanging the information of one dimension per communication round. However, in scalar diffusion algorithm, information of the exchanged dimension is diffused with full precision, while in Lloyd–Max case, information of the exchanged dimension is quantized with a finite precision. We selected the quantization interval so that we do not suffer from any
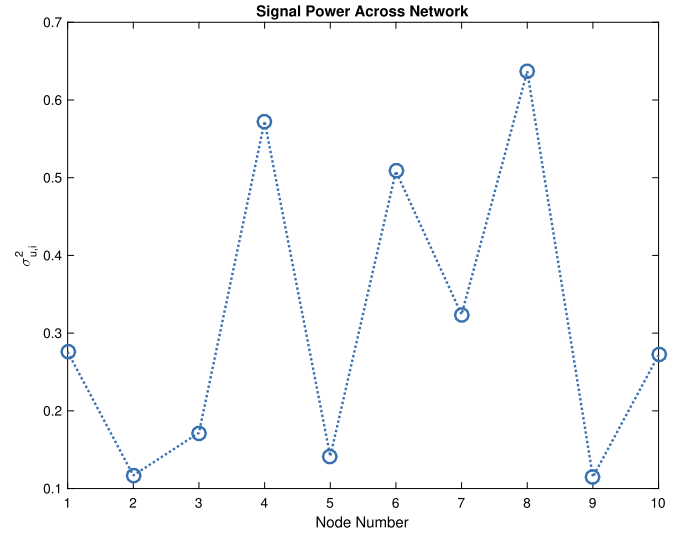


(a) Regressor variances over nodes
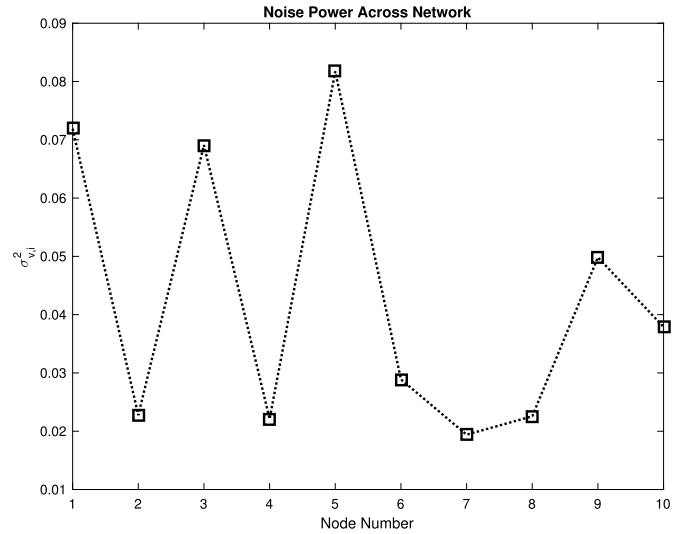


(b) Noise variances over nodes
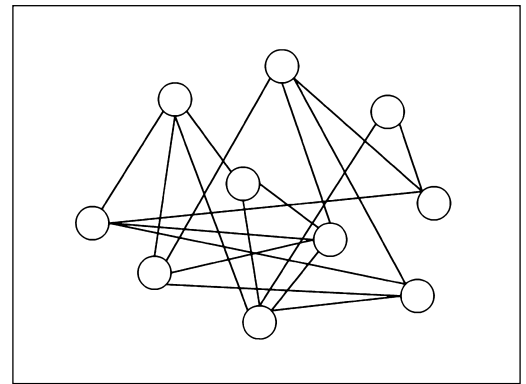


(c) Network topology

**Fig. 3.** Network topology and statistical profile.

saturation effects and also we have chosen the number of quantization levels so that no further significant improvement can be made on the MSD performance of the algorithms by increasing the number of levels. We observed that 53 quantization levels for the LC algorithm and 31 quantization levels for the conventional algorithm were sufficient. We use a step size of $\mu = 0.05$ during the
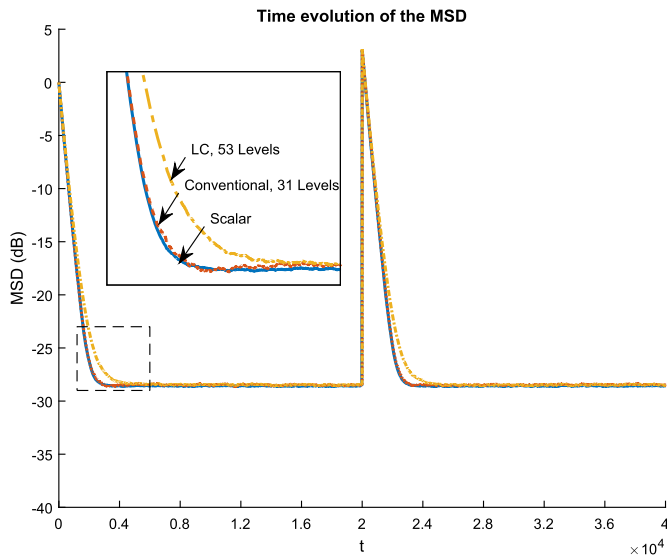
**Fig. 4.** The global MSD curves of the proposed algorithm, displayed with the label 'LC', in comparison with the conventional quantization and the scalar diffusion algorithms ($N = 10$, $M = 10$). Magnified figure provides the transient performance of the algorithms. Source statistics change at time $t = 2 \times 10^4$.
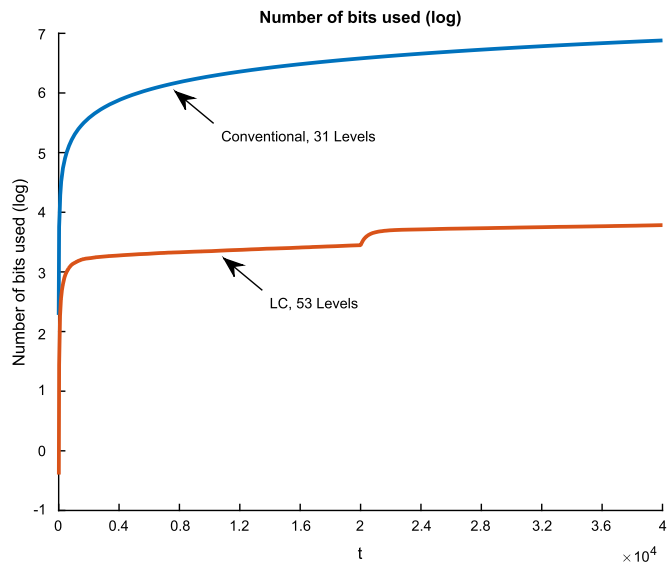


**Fig. 5.** Time evolution of the number of bits transmitted across the network. Sudden increase in the 'LC' curve corresponds to the time where the source statistics are changed.

simulations due to its good learning rate and convergence results. The results that we obtained in the experiments are averaged over 100 independent trials.

From these simulations, we observe that the convergence rate of the scalar diffusion and the conventionally quantized diffusion algorithms are superior compared to the proposed algorithm, while the steady-state MSD values of all three systems are identical. We note that it was our aim to get equal steady-state MSD values allowing a fair comparison in terms of the convergence speeds. Also, it is observed that the proposed algorithm is able to adapt well when faced with a sudden change in the source statistics.

In Fig. 5, we present the communication load that each algorithm incurs on the network. We exclude the scalar (infinite-precision) diffusion algorithm from this comparison since it requires an infinite number of bits to encode the information exchanged among the nodes. We observe a substantial enhancement in the communication efficiency achieved by the proposed algo-

rithm in terms of the total number of bits exchanged between the nodes across the entire adaptive network with respect to the algorithm that uses the conventional quantization. Particularly, for this $N = 10$ node network, we note that the proposed algorithm provides $10^3$ times less communication load over the reference implementation with the same steady state MSD values. We also observe that at the time of change in the source statistics, there is a sudden increase in the number of bits used by the proposed algorithm. This is because there are multiple level crossings occur due to the sudden change in the parameter of interest at that time, which requires more than two bits to encode. However, we observe that the system quickly adapts itself to using two-bits again. The same behavior is not present for the conventional quantization case since it already encodes true values of the levels at every single time instant. We stress further that we achieve this improvement with relatively little complexity since we have shown that using a simple non-adaptive quantizer is sufficient to realize the improvements.

In the second part of the experiments, we aim to observe the performance of the proposed algorithm over high dimensional data. Therefore, we have changed the former setup so that the unknown vector parameter $\boldsymbol{w_o}$ with $M = 100$ components is randomly chosen from a Normal distribution and normalized to have a unit energy. We use the same distributed network with connections given in Fig. 3c. Quantization levels for the algorithms again chosen so that no further significant improvement can be made by increasing the number of levels. We observed that 53 quantization levels for the LC algorithm and 31 quantization levels for the conventional algorithm were sufficient. We again use a step size of $\mu = 0.05$ and the results are averaged over 10 independent trials. We have decreased the number of independent trials to be averaged since processing high dimensional data takes significantly more time.

We present the MSD performance of the proposed algorithm in comparison with the sequential variant of the algorithm in [23] with the parameters $M = 100$, $L = 1$ in Fig. 6. We observe that in the high dimensional data case, the convergence rate of the proposed algorithm is the same as the compared algorithms. They also have the same steady-state MSD values. These results indicate that the adaptation performances of the scalar diffusion algorithm and the conventionally quantized diffusion algorithm decrease for the high dimensional case since the nodes are allowed to share only one dimension per round, which prevents them from quickly sending their entire intermediary estimates to their neighboring nodes. Therefore, we observe that for such systems, the proposed algorithm performs similar to the scalar diffusion and the conventionally quantized algorithms.

In Fig. 7, we illustrate the communication load for each algorithm. We observe an improvement on the communication requirements in a similar vein to the previous experiments. Ultimately, the proposed algorithm incurs $10^2$ times less communication load compared to the baseline, where the number of transmitted bits is significantly reduced. The magnitude of this reduction is of a smaller scale compared to the non-high dimensional case, on the other hand, mainly due to the extra bits required to encode the higher dimensions for multiple level crossings in the LC quantization.

In the third part of the experiments, in order to observe the possible effects of number of quantization levels, we simulate the algorithms within an identical experimental setup – except that the number of quantization levels are no longer optimized as in the previous cases. To this end, we have arbitrarily chosen 25 quantization levels for the LC algorithm and again 25 levels for the conventional algorithm. We use the same distributed network connections given in Fig. 3c. We have used a step size of $\mu = 0.05$ and the results are averaged over 100 independent trials.
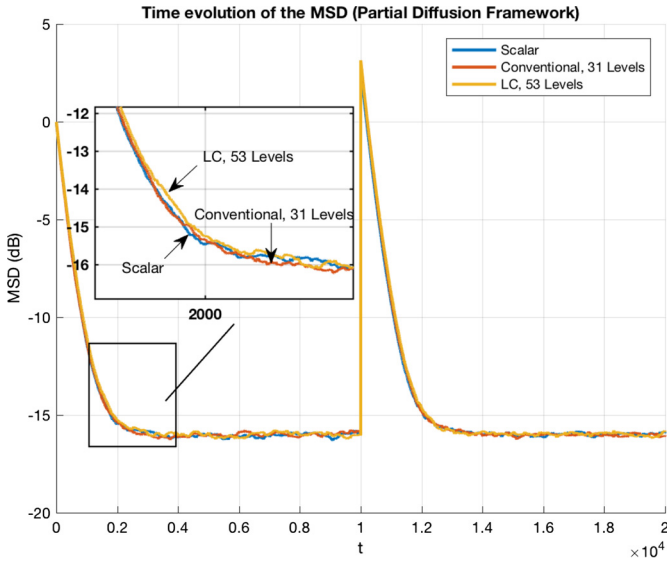
**Fig. 6.** The global MSD curves of the proposed algorithm, displayed with the label 'LC', in comparison with the conventional quantization and the scalar diffusion algorithms over high dimensional data ($N = 10$, $M = 100$). Magnified figure provides the transient performance of the algorithms. Source statistics change at time $t = 10^4$.
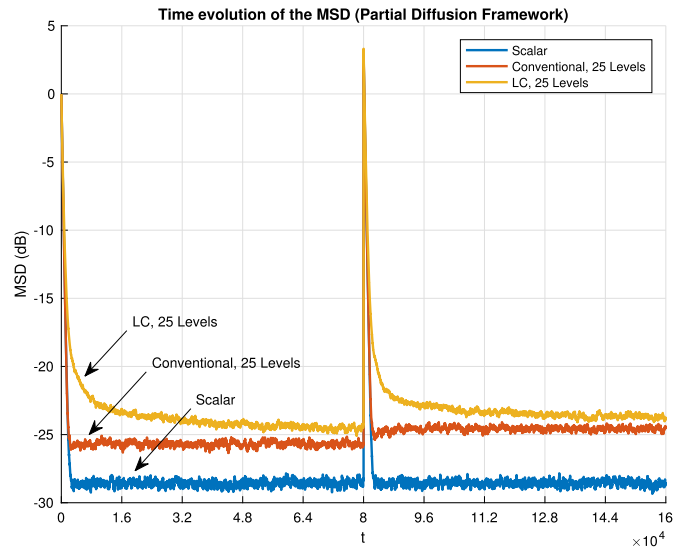


**Fig. 8.** The global MSD curves of the proposed algorithm, displayed with the label 'LC', in comparison with the conventional quantization and the scalar diffusion algorithms with sub-optimal quantization levels ($N = 10$, $M = 10$). Source statistics change at time $t = 10^4$.
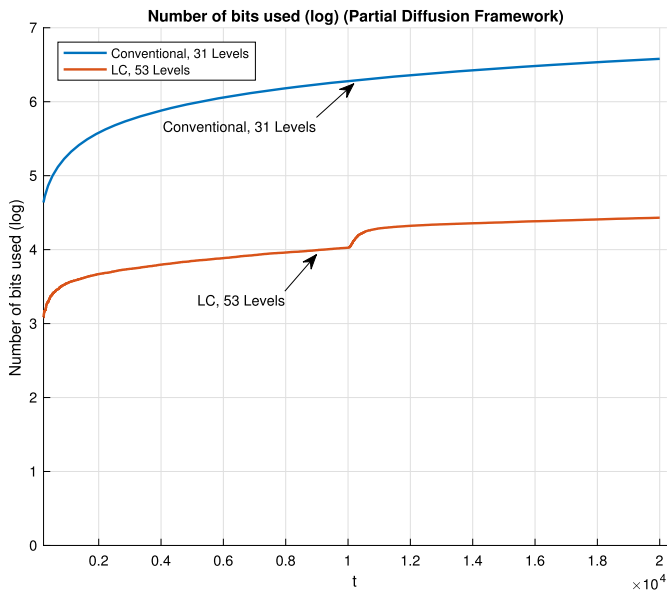


**Fig. 7.** Time evolution of the number of bits transmitted by the algorithms across network over high dimensional data ($N = 10$, $M = 100$). Sudden increase in the 'LC' curve corresponds to the time at which the source statistics are changed.
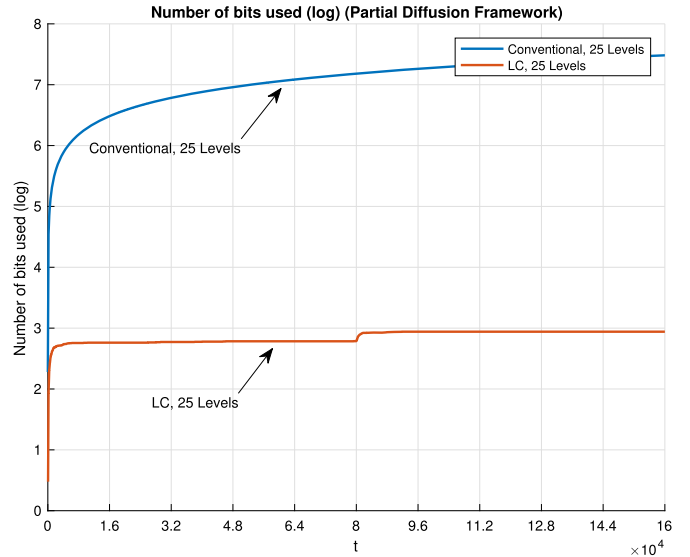


**Fig. 9.** Time evolution of the number of bits transmitted by the algorithms across network with sub-optimal quantization levels ($N = 10$, $M = 10$). Sudden increase in the 'LC' curve corresponds to the time at which the source statistics are changed.

We present the MSD performances of the algorithms in Fig. 8. We observe that when sub-optimal quantization levels are used, the compared algorithms exhibit superior performance compared to the proposed algorithm both in terms of the convergence rate and the steady-state MSD. We also note that the quantized algorithms could not reach the steady-state performance of the scalar diffusion due to the deliberate poor selection of the number of quantization levels.

These results are observed due to a failure on the system's part to satisfy the assumed quantization error model. The statistical model that we used for the quantization error $\phi_i^q$ assumes that it has zero mean such that $E[\phi_i^q] = 0$ [24]. However, when such a low number of quantization levels are selected, this model ceases to be applicable and the quantized algorithms are no longer guar-

anteed to converge to the steady-state MSD values of the scalar diffusion algorithm.

In Fig. 9, we present the communication load of the algorithms over the network for the case of a sub-optimal level selection. We again observe a similar behavior where the proposed algorithm diffuses more than $10^3$ times less bits through network compared to the baseline. We note that the difference in the number of bits exchanged between the two algorithms is larger compared with the previous results. This can be explained by the fact that we use fewer quantization levels for the LC algorithm, which makes the occurrence of multiple level crossings a rarer phenomenon. Thus, it becomes less likely for each node to send our more than two bits of information for a given iteration. Ultimately, this particular experiment illustrates the existence of a trade-off between the estimation performance and the communication load imposed on the network.
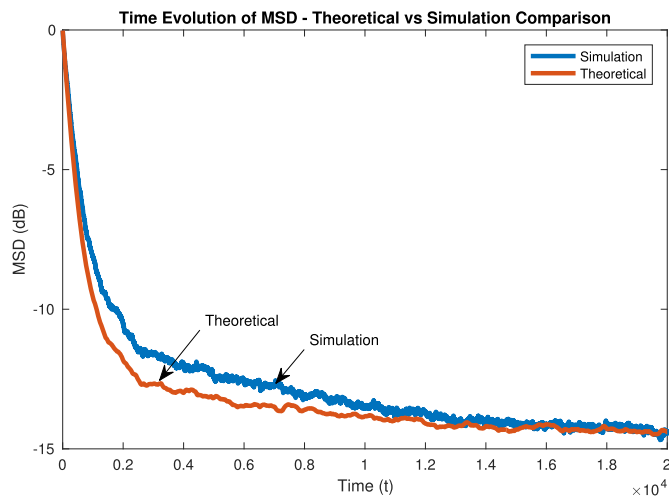
**Fig. 10.** The global MSD curve of the proposed algorithm, displayed with the label 'LC', in comparison with the theoretical MSD results over $M = 7, N = 2$ system.

In the last part of the experiments, in order to demonstrate the theoretical consistency of the proposed algorithm, we simulated the MSD performance of an $M = 7, N = 2$ network and compared it with the expected theoretical results. We have used 53 quantization levels for the simulations. Due to the analytical intractability brought about by the terms $E\left[\alpha_{t+1}\tilde{\phi}_{t+1}^T\right]$ and $E\left[\alpha_{t+1}\alpha_{t+1}^T\right]$ within our framework, we have decided to illustrate the transient behavior predicted by the theoretical analysis by replacing these terms by their empirically obtained counterparts. In addition, we have reduced both the data dimension and the number of nodes in the network due to the memory requirements imposed by the theoretical calculations. Throughout, we use a step size of $\mu = 0.05$ and all the results are averaged over 10 independent trials. In Fig. 10, we present the MSD performance of the proposed algorithm alongside the response predicted by our theoretical analysis. We observe that the simulation results exhibit a consistent behavior with respect to the derived theoretical results. The slight differences in the initial convergence behavior can be explained by the stochastic nature of our estimates for the $E\left[\alpha_{t+1}\tilde{\phi}_{t+1}^T\right]$ and $E\left[\alpha_{t+1}\alpha_{t+1}^T\right]$ terms.

## 8. Conclusion

We introduced an event-triggered distributed estimation algorithm with level-crossing quantization for distributed applications, where an unknown parameter is cooperatively learned by a group of nodes in an adaptive network. We proposed a diffusion-LMS algorithm where at each time instant, a node initiates communication with its neighbors only if the parameter to be communicated goes through a level crossing, which is signified by a single bit that indicates the direction of the level crossing. Consequently, the proposed algorithm required data transfers between the nodes that are much more sparse across time, as compared to a continuous stream of information at each instant. This translated into a much diminished load on available communication resources, which is of crucial importance in applications such as big data, where these resources are constrained, set against the sheer volume of the data. By theoretical analysis and simulations, we showed that the proposed algorithm is convergent in the mean sense, and we demonstrated that it provides up to a $10^3$ fold reduction in the communication load imposed on the network.

## References

[1] M.K. Banavar, J.J. Zhang, B. Chakraborty, H. Kwon, Y. Li, H. Jiang, A. Spanias, C. Tepedelenlioglu, C. Chakrabarti, A. Papandreou-Suppappola, An overview of recent advances on distributed and agile sensing algorithms and implementation, Digit. Signal Process. 39 (2015) 1–14.

[2] N. Assimakis, M. Adam, M. Koziri, S. Voliotis, K. Asimakis, Optimal decentralized Kalman filter and Lainiotis filter, Digit. Signal Process. 23 (2013) 442–452.

[3] N. Li, S. Sun, J. Ma, Multi-sensor distributed fusion filtering for networked systems with different delay and loss rates, Digit. Signal Process. 34 (2014) 29–38.

[4] J.-H. Seo, S.M. Jung, P. Park, A diffusion subband adaptive filtering algorithm for distributed estimation using variable step size and new combination method based on the {MSD}, Digit. Signal Process. 48 (2016) 361–369.

[5] J. Fernandez-Bes, L.A. Azpicueta-Ruiz, J. Arenas-Garcia, M.T. Silva, Distributed estimation in diffusion networks using affine least-squares combiners, Digit. Signal Process. 36 (2015) 1–14.

[6] S. Kumar, U.K. Sahoo, A.K. Sahoo, D.P. Acharya, Diffusion minimum-Wilcoxon-norm over distributed adaptive networks: formulation and performance analysis, Digit. Signal Process. 51 (2016) 156–169.

[7] D. Estrin, L. Girod, G. Pottie, M. Srivastava, Instrumenting the world with wireless sensor networks, in: IEEE International Conference on Acoustics, Speech and Signal Processing Proceedings, 2001, pp. 2033–2036.

[8] J. Chen, A. Sayed, Diffusion adaptation strategies for distributed optimization and learning over networks, IEEE Trans. Signal Process. 60 (2012) 4289–4305.

[9] S. Werner, M. Mohammed, Y.-F. Huang, V. Koivunen, Decentralized set-membership adaptive estimation for clustered sensor networks, in: 2008 IEEE International Conference on Acoustics, Speech and Signal Processing, 2008, pp. 3573–3576.

[10] J.-J. Xiao, A. Ribeiro, Z.-Q. Luo, G. Giannakis, Distributed compression–estimation using wireless sensor networks, IEEE Signal Process. Mag. 23 (2006) 27–41.

[11] S. Werner, Y.-F. Huang, M.L. De Campos, V. Koivunen, Distributed parameter estimation with selective cooperation, in: 2009 IEEE International Conference on Acoustics, Speech and Signal Processing, 2009, pp. 2849–2852.

[12] S. Chouvardas, K. Slavakis, S. Theodoridis, Trading off complexity with communication costs in distributed adaptive learning via Krylov subspaces for dimensionality reduction, IEEE J. Sel. Top. Signal Process. 7 (2013) 257–273.

[13] R. Arablouei, S. Werner, K. Doğançay, Y.-F. Huang, Analysis of a reduced-communication diffusion LMS algorithm, Signal Process. 117 (2015) 355–361.

[14] S. Werner, Y.-F. Huang, Time- and coefficient-selective diffusion strategies for distributed parameter estimation, in: 2010 Conference Record of the Forty Fourth Asilomar Conference on Signals, Systems and Computers, 2010, pp. 696–697.

[15] R. Arablouei, K. Doğançay, S. Werner, Y.-F. Huang, Adaptive distributed estimation based on recursive least-squares and partial diffusion, IEEE Trans. Signal Process. 62 (2014) 3510–3522.

[16] C.G. Lopes, A.H. Sayed, Diffusion least-mean squares over adaptive networks: formulation and performance analysis, IEEE Trans. Signal Process. 56 (2008) 3122–3136.

[17] F.S. Cattivelli, A.H. Sayed, Diffusion LMS strategies for distributed estimation, IEEE Trans. Signal Process. 58 (2010) 1035–1048.

[18] Z. Xiaochuan, A. Sayed, Single-link diffusion strategies over adaptive networks, in: IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, 2012, pp. 3749–3752.

[19] J.W. Mark, T. Todd, A nonuniform sampling approach to data compression, IEEE Trans. Commun. 29 (1981) 24–32.

[20] C. Lopes, A. Sayed, Diffusion adaptive networks with changing topologies, in: IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, 2008, pp. 3285–3288.

[21] N. Takahashi, I. Yamada, Link probability control for probabilistic diffusion least-mean squares over resource-constrained networks, in: IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, 2010, pp. 3518–3521.

[22] T.G.M.O. Sayin, N.D. Vanli, S.S. Kozat, Communication efficient channel estimation over distributed networks, in: IEEE Global Conference on Signal and Information Processing, 2014.

[23] R. Arablouei, S. Werner, Y.F. Huang, K. Dogancay, Distributed least mean square estimation with partial diffusion, IEEE Trans. Signal Process. 62 (2014) 472–483.

[24] A.H. Sayed, Fundamentals of Adaptive Filtering, John Wiley and Sons, 2003.

[25] N. Takahashi, I. Yamada, A.H. Sayed, Diffusion least-mean squares with adaptive combiners: formulation and performance analysis, IEEE Trans. Signal Process. 58 (9) (2010) 4795–4810.

**Ihsan Utlu** received the B.S. degree with high honors in Electrical and Electronics Engineering from Middle East Technical University, Ankara, Turkey in 2014, and the M.S. degree in Electrical and Electronics Engineering from Bilkent University, Ankara, Turkey in 2017. He is currently with the Intelligent Data Analytics Research Program Department, ASELSAN Research Center, Ankara, Turkey, and is working toward a Ph.D. degree in Electrical and Electronics engineering at Bilkent University, Ankara, Turkey. His research interests include statistical signal processing, distributed optimization, and natural language processing.

**O. Fatih Kilic** received his B.S. degree with full scholarship and high honors from Bilkent University. He has graduated from the Electrical and Electronics Engineering Department with a GPA of 3.85/4.00. He is currently pursuing an M.Sc. degree at Bilkent University. His research interests include big data signal processing, online learning and optimization.

**Suleyman S. Kozat** (SM'11) received the B.S. degree with full scholarship and high honors from Bilkent University, Turkey. He received the M.S. and Ph.D. degrees in electrical and computer engineering from University of Illinois at Urbana Champaign, Urbana, IL.

After graduation, he joined IBM Research, T.J. Watson Research Lab, Yorktown, New York, as a Research Staff Member (and later became a Project Leader) in the Pervasive Speech Technologies Group, where he focused on problems related to statistical signal processing and machine learning. While doing his Ph.D., he was also working as a Research Associate at Microsoft Research, Redmond, Washington, in the Cryptography and Anti-Piracy Group. He holds several patent inventions due to his research accomplishments at IBM Research and Microsoft Research. He is currently an Associate Professor at the Electrical and Electronics Engineering Department at Bilkent University. He is the elected President of the IEEE Signal Processing Society, Turkey Chapter. He coauthored more than 100 papers in refereed high impact journals and conference proceedings and has several patent inventions (currently used in several different Microsoft and IBM products such as the MSN and the ViaVoice). He holds many international and national awards. Overall, his research interests include cyber security, anomaly detection, big data, data intelligence, adaptive filtering and machine learning algorithms for signal processing.