EVENT-TRIGGERED DISTRIBUTED ESTIMATION WITH REDUCED COMMUNICATION LOAD

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF
MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

By
İhsan Utlu
January 2017

EVENT-TRIGGERED DISTRIBUTED ESTIMATION WITH REDUCED COMMUNICATION LOAD
By İhsan Utlu
January 2017

We certify that we have read this thesis and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Süleyman Serdar Kozat(Advisor)

Sinan Gezici

Çağatay Candan

Approved for the Graduate School of Engineering and Science:

Ezhan Karasan
Director of the Graduate School

ABSTRACT

EVENT-TRIGGERED DISTRIBUTED ESTIMATION WITH REDUCED COMMUNICATION LOAD

İhsan Utlu
M.S. in Electrical and Electronics Engineering
Advisor: Süleyman Serdar Kozat
January 2017

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 the computation of the 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 the proposed algorithm, and illustrate the gain in communication efficiency compared to other reduced-communication distributed estimation schemes.

Keywords: Distributed estimation, adaptive networks, event-triggered communication, level-crossing quantization.

ÖZET

DÜŞÜK İLETİŞİM YÜKLÜ OLAY-TETİKLİ DAĞITIK KESTİRİM

İhsan Utlu
Elektrik ve Elektronik Mühendisliği, Yüksek Lisans
Tez Danışmanı: Süleyman Serdar Kozat
Ocak 2017

Mevcut bulunan iletişim kaynakları yönünden kısıtlamalara tabi olan dağıtık kestirim uygulamaları için, gelişkin metodlarla kıyaslanabilir bir başarımı korurken ağ üzerindeki iletişim yükü açısından 10^3 kata kadar bir düşüş gerçekleştirebilen, yayınım stratejilerine dayanan yeni bir algoritma öneriyoruz. Önerilen yöntemde bilginin, yerel kestirimlerin hesaplanmasından sonra işlem elemanları (yumrular) arasında yayılması, bilgi aktarımının yayınımı yapılan parametrenin seviye-kesimlerine şartlandırılması yoluyla tekdüze olmayan bir şekilde gerçekleşmekte; bunun sonucunda iletişim gereksinimlerinde büyük bir azalma elde edilmektedir. Önerilen algoritmanın ortalama ve ortalama kare kararlılık analizleri gerçekleştirilmiş ve diğer düşük-iletişimli dağıtık kestirim yöntemlerine kıyasla iletişim veriminde elde edilen kazanç gösterilmiştir.

Anahtar sözcükler: Dağıtık kestirim, uyarlanır ağlar, olay-tetikli iletişim, seviye kesimi nicemlemesi.

Acknowledgement

I acknowledge that this thesis is supported by TÜBİTAK BİDEB 2210-A Scholarship Programme.

Contents

List of Figures

Chapter 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, 2, 3, 4]. 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 [5, 6, 7, 8, 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. To this end, in this thesis, 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 [11]. Among the different approaches, 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 [11, 12]. 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 [11]. 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 [13].

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 [14], unlike using a fixed rate of diffusion, cf. [11, 12]. 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 a single bit for a slowly-varying parameter.

Reduced communication diffusion is extensively studied in the signal processing literature [15, 16, 13, 17, 18]. In [15, 16, 13], 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 [17], local estimates are randomly projected, and the information transfer between the nodes is reduced to a single bit. In [18], only certain dimensions of the parameter vector are transmitted. On the other hand, in this thesis, we reduce the communication load down to only a single bit or a couple of bits, unlike [15, 16, 13, 18], 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 [17] 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 achieve a significant reduction in the communication load over the network.

The thesis is organized as follows: In Chapter 2, we provide the motivation and background for general distributed adaptive filtering problems. In Chapter 3, Section 3.1, we introduce the distributed estimation framework and discuss the adapt-then-combine (ATC) diffusion strategy. We further detail our algorithms in Section 3.2, where we formulate the level-triggered distributed estimation algorithm. In Section 3.3, we present the algorithmic description of the proposed scheme. In Sections 3.4, 3.5 and 3.6 we provide respectively the mean, mean-square and tracking performance analyses of the proposed distributed adaptive filter and state the conditions for stability. We provide experimental verification of the algorithm in Chapter 4, and concluding remarks in Chapter 5.

Notation: 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 $\mathrm{diag}\{\boldsymbol{A}\}$ returns a new matrix with only the main diagonal of $\boldsymbol{A}$ while $\mathrm{diag}\{a\}$ puts a on the main diagonal of the new matrix. $\mathrm{col}\{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_N\}$ 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. $\mathbb{1}_N$ denotes an $N \times 1$ vector of ones. $\otimes$ stands for the Kronecker product, $\mathrm{Tr}\{\cdot\}$ stands for the trace.

Chapter 2

Distributed Estimation

2.1 Motivation and Background

In this chapter, we motivate the distributed estimation framework considered in the succeeding chapters of this thesis. We initiate the discussion by a brief summary of the operation of adaptive filters, followed by the more general structure of combinations of different instances of adaptive filters to improve some desirable property of an end configuration, e.g., an improved convergence behavior or reduced steady-state mean-square deviation. Subsequently, as a precursor to more sophisticated filter combination schemes that operate over a network of filters (agents) forming an arbitrary topology, we consider consensus and diffusion as the two well-known cooperation schemes from the literature. We introduce the problem of consensus over a network and the associated fundamental consensus protocols, briefly discussing the associated convergence results. The chapter is finalized with a discussion of consensus-based distributed optimization and estimation schemes and their stated shortcomings against the diffusion schemes.

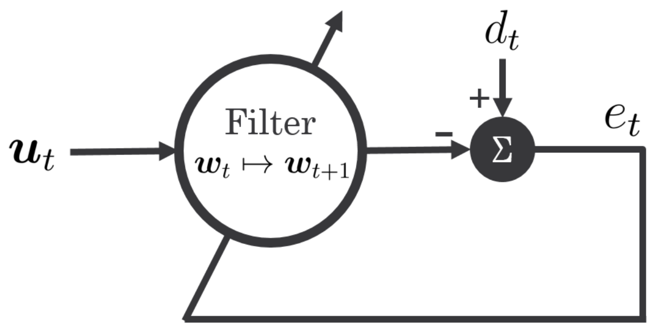


Figure 2.1: The operation of an adaptive filter. The process of adaptation is represented by an oblique arrow

2.2 Adaptive Filtering

Signal processing problems from a wide range of applications have derived substantial benefits from the well-known adaptive filter structures as fundamental building blocks for online linear parameter estimation tasks, owing to their inherent characteristics including its ability to operate with incomplete statistical information, and retain performance under potentially non-stationary environments [19]. In particular, adaptive filter based solutions can naturally fit into the frameworks of fundamental signal processing problems such as system identification, linear prediction and channel equalization [20]. To illustrate the main principles, as illustrated in Fig. 2.1, we introduce two processes $\{d_t\}_{t\geq 1}$, $\{\boldsymbol{u}_t\}_{t\geq 1}$ indexed by the discrete time variable t, representing the reference and regressor (or input) signals under consideration, respectively, which we presume to follow a linear model of the form $d_t \approx \boldsymbol{u}_t^T \boldsymbol{w_o}$ for some unknown parameter $\boldsymbol{w_o}$ that is to be estimated. In this context, an adaptive filter updates its coefficient vector $\boldsymbol{w}_t$ at time t in an online manner by accounting for the new observations d_t, $\boldsymbol{u}_t$, with the objective of driving some metric $f(\cdot)$ of the error $e_t = |d_t - \boldsymbol{u}_t^T \boldsymbol{w}|$ towards zero in the mean sense, equivalently tuning itself such that the updated coefficients best describe the observed $d_t, \boldsymbol{u}_t$ by achieving $d_t \approx \boldsymbol{u}_t^T \boldsymbol{w}_{t+1}$ in expectation. More formally, an adaptive filter aims to iteratively minimize a stochastic approximation [21] to a cost function of the form $J(\boldsymbol{w}) = E\left[f\left(d_t - \boldsymbol{u}_t^T \boldsymbol{w}\right)\right]$. As an example, utilizing the instantaneous approximations $E[\boldsymbol{u}_t \boldsymbol{u}_t^T] \approx \boldsymbol{u}_t \boldsymbol{u}_t^T$ and $E[d_t \boldsymbol{u}_t^T] \approx \boldsymbol{u}_t d_t$, and employing the squared-error metric with a gradient-descent learning scheme

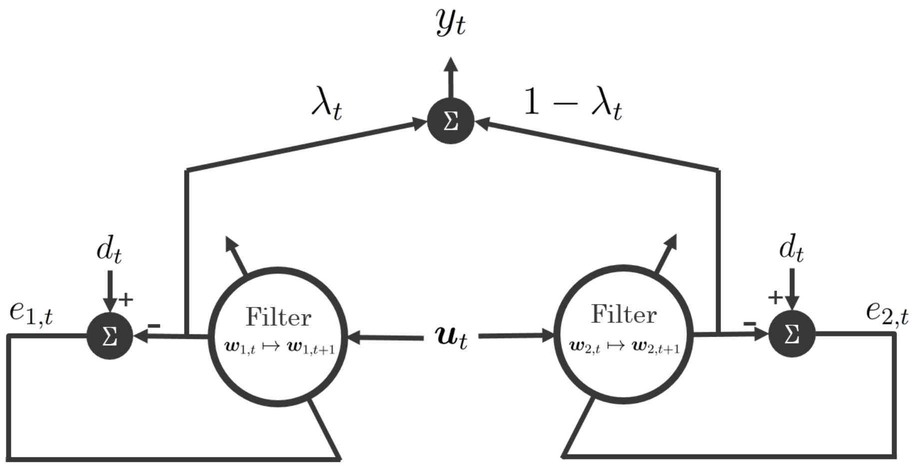


Figure 2.2: An adaptive filter configuration obtained by a convex combination at the output layer

results in the well-known LMS algorithm [19]. Choice of a combination of different loss functions (e.g. a mean fourth error loss [22]), different learning rules (e.g. a Hessian based update, which can be shown to yield the RLS algorithm [19]) and different stochastic approximation rules results in different adaptive filter structures with varying characteristics in error performance, tracking performance and convergence rate. Furthermore, given a particular structure, e.g. LMS with the update rule

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + \mu_t \boldsymbol{u}_t (d_t - \boldsymbol{u}_t^T \boldsymbol{w}_t),$$

the designer is facilitated to adjust for the required specifications by tuning the free parameters in the chosen filter structure. A canonical example is that adjusting the step size μ sets up a trade-off between the steady state MSE and the convergence speed under stationary environments for the LMS algorithm [19].

2.2.1 Combinations of Adaptive Filters

Before we introduce the distributed filtering problem, in order to further motivate the distributed estimation framework investigated in this thesis, it is instructive to consider the literature on systems comprised of a combination of adaptive filters [23]. As illustrated in Fig. 2.2, these systems employ at least two adaptive filters fed with identical reference and regressor processes whose outputs $\boldsymbol{u}_{i,t}^T \boldsymbol{w}_{i,t+1}$,

$i = 1, 2, \ldots$ are combined in a convex [24], affine [25] or unconstrained linear [26] manner to yield an overall filter output y_t, where the combination weights adjusted online according to a, e.g. gradient-based [24], learning scheme. The findings reported in the literature indicate that the combination of a fast and a slow LMS filter in a stationary environment results in an improved steady-state MSE performance in a universal sense, performing as well as the best performer among the constituent filters [24]. Combinations of filters from different filter families is also considered in the literature. A convex combination LMS and RLS filters is shown to result in an improvement in the tracking performance under non-stationary environments [27]. Another result concerning the tracking performance indicates that combining filters from the *same* family (e.g. LMS+LMS) does not yield a better tracking performance compared to the performance of a constituent filter with an optimally tuned step size [28].

While the above laid-out cooperation schemes among different adaptive filters constitute an important milestone in achieving better error or tracking performance with increased robustness, a critical factor associated with their formulation is that the individual filters are set to operate in an isolated manner, with information synthesis only occurring at the output layer. It is in turn in line with intuition to suggest that information dissemination at lower levels of operation, e.g., by exchanging the internal states of the constituent filters and incorporating the received information as the result of the exchange in the adaptation process, could lead to even greater improvements in estimation performance, which is verified by the results from the distributed adaptive filtering literature [12, 29, 30]. As a precursor to its use in general distributed estimation scenarios, this approach is seen to have found application in adaptive filter mixture problems, e.g. in [31] as illustrated in Fig. 2.3, where the speed of convergence of the overall configuration is observed to be improved by employing an update based on shared internal states of the form

$$\boldsymbol{w}_{i,t+1} = \sum_{j=1}^{2} \alpha_{i,j} \boldsymbol{w}_{j,t} - \mu_i e_i \boldsymbol{u}_i,$$

for a two-filter setup. It can be shown that this iteration corresponds to what is called a *consensus* based cooperation scheme, which we explore in more detail in the next section.

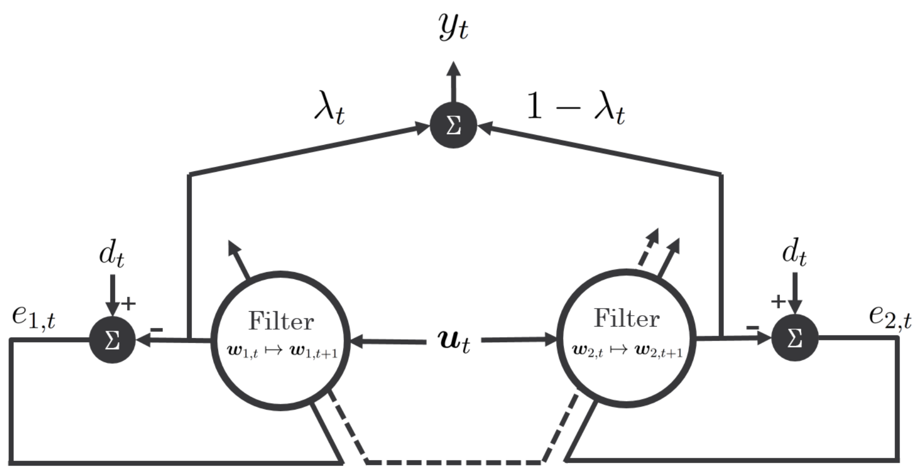


Figure 2.3: A convex combination scheme with shared states influencing adaptation

2.3 Consensus Algorithms

Consensus problems concern the ability of a number of agents organized in a network to reach a common decision or agreement on the value of some parameter that is of common interest, where the individual initial states of the agents are required to asymptotically converge to a common value via local interactions over the network [32]. A consensus algorithm is any method that proposes to solve this problem in a distributed, iterative fashion where each agent shares information it gathers from local computations to its immediate neighbors. Consensus problems have their roots in computer science and statistics literatures, with an interest towards problems such as distributed parallelized computation [33, 34, 35] and obtaining agreement amongst a panel of experts [36].

More formally, given an arbitrary collection of initial states $\{\boldsymbol{w}_{i,0}\}_{i=1}^N$ for agents $i = 1, \ldots, N$, the goal of a consensus algorithm is to specify a local interaction rule such that $\|\boldsymbol{w}_{i,t} - \boldsymbol{w}_{j,t}\| \to 0$ in steady state, for all agents i, j. [37]. The final equilibrium state $\boldsymbol{w}^*$ is in general specified to be an arbitrary function $f(\cdot)$ of the initial states, in which case the protocol is said to solve an f-consensus problem [38, 39]. A canonical example is the average-consensus problem where the nodes try to reach a consensus on the average value of their initial states in a distributed manner. Consensus iterations have enjoyed extensive utilization in many fields including, e.g. applications from control theory where autonomous agents are forced to agree on a common orientation, known as the multi-agent coordination problem, or applications in estimation theory, e.g. a wireless sensor network (WSN) where the sensors are forced to agree on the true value of some parameter.

In the following, we will briefly discuss some fundamental underpinnings of consensus algorithms, which will also serve as a basis for the discussions on distributed estimation in the later chapters. To this end, we begin by considering the organization of the computational units (agents) within the system. The agents in the network are typically represented as nodes in a potentially time-varying directed graph $\mathcal{G}_t = (V, E_t)$, which specifies the pairs of agents that are allowed to communicate as well as the permitted directions of information flow, where V and E_t represent the sets of nodes and directed edges, respectively. We note that in most treatments multiple edges are disallowed, while self-loops are enforced. Generally, the inter-agent communications are allowed to be subject to time delays, and the network connectivity is assumed to be time varying. We further note that the special cases of undirected (with bidirectional communication links) and/or time-varying graphs have also been treated in the literature [40].

Consensus protocols for continuous time and discrete time operations take the following canonical forms

$$\dot{\boldsymbol{w}}_{i,t} = \sum_{j\in\mathcal{N}_i\setminus\{i\}} \alpha_{i,j,t}\left(\boldsymbol{w}_{i,t} - \boldsymbol{w}_{j,t}\right) \tag{2.1}$$

$$\boldsymbol{w}_{i,t+1} = \sum_{j\in\mathcal{N}_i} \alpha_{i,j,t}\boldsymbol{w}_{j,t}, \tag{2.2}$$

which are also called first-order consensus protocols due to the assumed first order integrator/accumulator behaviour for the agents [38]. Here $\mathcal{N}_i$ is defined as the neighborhood of the node i which is the set of nodes that it is receives information from in a single time step, including itself. The potentially time-varying nonnegative weights $\alpha_{i,j,t}$ signify the weight assigned by the node i to the contribution it receives from the node j, where the positivity of the weights associated with an information flow from a node i to a node j is reflected as a directed link in the underlying graph $\mathcal{G}_t$ in the $i-j$ direction. We note that the inclusion of the node i in the set $\mathcal{N}_i$ signifies the frequent assumption in convergence analyses of discrete-time consensus that a node remains in communication with itself for all time steps, i.e., $\alpha_{ii,t} > 0$ for all t [41, 42, 43]. The weights are further required to yield a convex combination, satisfying $\sum_{j \in \mathcal{N}_i} \alpha_{i,j,t} = 1$. This final condition is reflected in the requirement that the matrix $A_t \triangleq [a_{i,j,t}]$ is (right) stochastic, with unity row sums.

Consensus protocols have been supported by a foundational literature treating issues involving design of appropriate interaction schemes for different objectives, and the convergence performance under varying operation modes, e.g. with time-varying (switching) topologies, or operation under delays.

Some fundamental results in the literature concern the requirements for convergence for consensus protocols of the form (2.1)-(2.2) [41, 43, 44, 45]. In many of the reported results, these conditions are stated in terms of the connectivity properties of the underlying graph $\mathcal{G}_t$. In particular, a sufficient condition for convergence under a switching topology has been established as the regular emergence of at least one 'well-connected' node that can reach every node (including itself) in the network using the time-varying communication links, over some finite time interval. Specifically, considering the discrete-time protocol as an example, this condition is stated in terms of the ability to construct a spanning tree using the edges of the graph obtained by the union $\bigcup_{t=t_k}^{t_{k+1}-1} \mathcal{G}_t$ of the graphs $\mathcal{G}_t$ for each consecutive uniformly bounded time interval with index $k \geq 1$. With such an assignment of spanning trees to consecutive time intervals, the root node of an assigned spanning tree is observed to satisfy the above-defined notion of connectedness over its corresponding time interval, ensuring convergence of the consensus protocol [41, 43]. An important point to note is that this condition implies that the edges in the network that spring up infinitely often over the infinite time horizon would also have to be able to support at least one node as the root of a spanning tree, so that the above-mentioned condition can still be satisfied even when all the transient connections have died out. These nodes (called leaders [40]) are shown to be the ones that, via their initial states, determine the final consensus value that the nodes in the network agree upon [37].

We further note that different variants of the above conditions of sufficiency for convergence have been proposed in the literature, albeit with stronger connectivity assumptions on the union $\bigcup_{t=t_k}^{t_{k+1}-1} \mathcal{G}_t$ [32, 42]. Furthermore, we remark that for the case of bidirectional connectivity, it has been shown that the more relaxed condition of requiring a spanning tree in $\bigcup_{t=t'}^{\infty} \mathcal{G}_t$ is sufficient, as opposed to requiring the emergence of a spanning tree with some temporal regularity [43]. We also note that the condition for the existence of a spanning tree can equivalently be stated as requiring a strongly connected graph for the bidirectional communication case [32]. Finally, the effect of time-delays in (2.1)-(2.2) are shown to be tolerable as long as the delays reside within some bounds determined by the connectivity of the network [45].

The equilibrium state $\boldsymbol{w}^*$ for the consensus protocol was previously remarked to be influenced by the initial states of the nodes that are 'well-connected' in the steady state (leader nodes). In particular, for the iteration (2.2), the consensus state is obtained as a weighted average of the initial states of these particular nodes [38]. The weights of this averaging scheme is shown to be amenable to change by controlling $\mathcal{G}_t$; as an example, it has been shown that an exact (unweighted) averaging algorithm can be obtained by requiring that the time-invariant directed graph $\mathcal{G}$ is balanced, i.e., $\sum_i \alpha_{i,j} = \sum_i \alpha_{j,i}$ as in [32], which can be equivalently stated as requiring that the weight matrix $A = [a_{i,j}]$ is doubly-stochastic. Another special case of interest is the scenario where the structure of the graphs $\mathcal{G}_t$ are arranged such that only a single leader node emerges, which can occur if and only if a particular node never accepts incoming traffic but only does broadcasts, in which case the initial state of *only* this node is shown to contribute to the final equilibrium state, with the states of all the other nodes converging towards the initial state of this particular node. This also known as the 'follow-the-leader' scheme [40].

Taking into consideration the fundamental issues that have been laid out in the preceding discussion regarding the convergence behavior of the basic consensus scheme (2.2), the associated consensus iterations have been used as building blocks in sophisticated algorithms from many different fields [32]. Of special interest is the applications to general distributed optimization problems where a sum of local objectives is aimed to be minimized [46], and the use of consensus in distributed parameter estimation problems, which includes distributed minimum mean square type estimators [47] and distributed Kalman filters [48].

A survey of the literature reveals that the evolution of algorithms from the optimization and estimation tracks are interrelated, with advancements from one track amenable to transfer to the other [49]. In particular, the consensus iteration (2.2) and its variants is observed to find utilization in distributed algorithms as an intermediary averaging module [32], which local agents engage in with some temporal regularity determined by the algorithm design, resulting in an interim agreement amongst the local intermediary estimates over the network. A useful taxonomy for these algorithms is identified in the literature to be the timing of these consensus iterations –signifying social learning– set against the timing of the local adaptations based on gradient updates (self-learning) [30]. For the cases where the rate of communication over the network is large enough compared to the acquisition rate of local observations, two-time-scale algorithms have been proposed where the consensus iterations are performed at a slower, distinct time-scale, while local adaptations are performed at a faster rate. It has been established in the literature that these earlier two-time-scale configurations ultimately constitute an unwarranted compromise against their single time-scale counterparts in terms of their adaptation capability to streaming data with high-frequency variations [48]. Accordingly, single time-scale approaches where the local adaptation and cooperation are fused into a single iteration have recently been proposed in the distributed optimization and estimation literatures [46, 47]. Particularly in the optimization domain, various single time-scale primal-domain distributed optimization schemes have been developed such as the ones based on subgradients [50], proximal gradients [51], or dual-averaging [52]. The distributed variant of the algorithm based on the subgradient method, as an example, is given by [50]

$$\boldsymbol{w}_{i,t+1} = \sum_{j=1}^{N} \alpha_{i,j} \boldsymbol{w}_{j,t} - \mu_i \boldsymbol{s}_i,$$

with the iterates $\boldsymbol{w}_{i,t}$ associated with an agent i converging to the optimal value for a deterministic objective of the form $\sum_{i=1}^{N} f_i(\boldsymbol{w})$, with $\boldsymbol{s}_i$ standing for a subgradient for the local cost f_i for the agent i.

A fundamental limitation adaptive filters that are built on top of the above subgradient based methods have been identified in the literature to be a dependence on vanishing step-sizes μ_i for convergence [53], which is noted to limit the adaptation capabilities of the filter for non-stationary environments [48]. Further limitations associated with consensus-based filters in general have been observed to be a significant sensitivity on filter parameters such as combination weights for the stability of the overall configuration [54]. These limitations are addressed in the literature by so-called *diffusion* methods for distributed adaptive filtering, which does away with the requirement of diminishing step sizes for convergence, lending itself well to configurations with enhanced adaptation facilities under non-stationary environments [55]. The diffusion algorithm and its major collaboration protocols (strategies) constitute the foundation for the reduced communication algorithms proposed and analyzed and this thesis, and in turn, will be discussed separately in the chapter that follows.

Chapter 3

Distributed Estimation with Level Triggered Sampling

3.1 Problem Description

We consider a network with N nodes that are distributed spatially as shown in Fig. 3.1. Each node sequentially observes a noise-corrupted transformation of an unknown parameter $\boldsymbol{w_o}$ through a linear model

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

and diffuses information to its neighboring nodes $j \in \mathcal{N}_i$, 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 $\boldsymbol{R}_{\boldsymbol{u},i} \overset{\triangle}{=} E\left[\boldsymbol{u}_{i,t}\boldsymbol{u}_{i,t}^T\right] = \sigma_{u,i}^2 \boldsymbol{I}_M$, $E\left[v_{i,t}^2\right] = \sigma_{v,i}^2$.

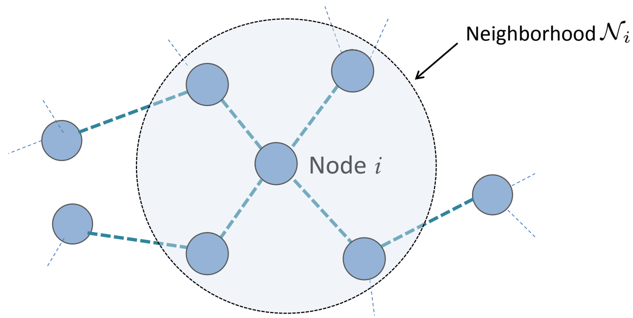


Figure 3.1: A sample network with N nodes.

We follow a stochastic gradient approach to minimizing a cost function of the form

$$J(\boldsymbol{w}) = \sum_{i=1}^{N} J_i(\boldsymbol{w}), \tag{3.1}$$

where $J_i(\boldsymbol{w}) = \frac{1}{2}E|d_{i,t} - \boldsymbol{u}_{i,t}^T\boldsymbol{w}|^2$ is the local mean square error cost incurred by the node i. The diffusion family of distributed solutions to minimizing (3.1) rely on reformulating this total cost in terms of local and extrinsic terms from the perspective of a single node i [12], yielding

$$J(\boldsymbol{w}) = \frac{1}{2}E|d_{i,t} - \boldsymbol{u}_{i,t}^T\boldsymbol{w}|^2 + \sum_{j=1,\, j\neq i}^{N} J_j(\boldsymbol{w}),$$

which, noting the decomposition $E|d_{j,t} - \boldsymbol{u}_{j,t}^T\boldsymbol{w}|^2 = \|\boldsymbol{w} - \boldsymbol{w}_j^o\|^2_{\boldsymbol{R}_{\boldsymbol{u},j}} + J_{\min,j}$ by completing the square, is equivalent to minimizing

$$J(\boldsymbol{w}) = \frac{1}{2}E|d_{i,t} - \boldsymbol{u}_{i,t}^T\boldsymbol{w}|^2 + \sum_{j=1,\, j\neq i}^{N} \frac{1}{2}\|\boldsymbol{w} - \boldsymbol{w}_j^o\|^2_{\boldsymbol{R}_{\boldsymbol{u},j}}, \tag{3.2}$$

where $\boldsymbol{w}_j^o = \boldsymbol{R}_{\boldsymbol{u},j}^{-1}\boldsymbol{R}_{d\boldsymbol{u},j}$ is the local minimum MSE solution, and $\boldsymbol{R}_{d\boldsymbol{u},j} \triangleq E[\boldsymbol{u}_{j,t}d_{j,t}]$.

In the ATC strategy [12], which we use as an example, an approximate cost function is formulated as an alternative to (3.2) for the node i as

$$\hat{J}_i(\boldsymbol{w}) = \frac{1}{2}E|d_{i,t} - \boldsymbol{u}_{i,t}^T\boldsymbol{w}|^2 + \sum_{j\in\mathcal{N}_i\setminus\{i\}} \frac{1}{2}\beta_{i,j}\|\boldsymbol{w} - \boldsymbol{\phi}_j\|^2, \tag{3.3}$$

for some intermediary estimate ϕ_j communicated from the node j to the node i, and for some set of coefficients $\beta_{i,j} \geq 0$ satisfying $\sum_{j=1}^{N} \beta_{i,j} = 1$ that denote the different weights assigned to the different neighbors.

The gradient of the cost (3.3) is given by

$$\left[\nabla_{\boldsymbol{w}}\hat{J}_i(\boldsymbol{w})\right]^T = \left(\boldsymbol{R}_{\boldsymbol{u},i}\boldsymbol{w} - \boldsymbol{R}_{d\boldsymbol{u},i}\right) + \sum_{j\in\mathcal{N}_i\setminus\{i\}} \beta_{i,j}(\boldsymbol{w} - \boldsymbol{\phi}_j). \tag{3.4}$$

To minimize (3.3), we may proceed by directly appealing to a stochastic gradient descent based approach [19]. Using (3.4) and the instantaneous approximations $\boldsymbol{R}_{\boldsymbol{u},i} \approx \boldsymbol{u}_{i,t}\boldsymbol{u}_{i,t}^T$ and $\boldsymbol{R}_{d\boldsymbol{u},i} \approx \boldsymbol{u}_{i,t}d_{i,t}$, would result in a recursion for the estimate of $\boldsymbol{w_o}$ at the node i given by

$$\boldsymbol{w}_{i,t+1} = \boldsymbol{w}_{i,t} + \mu_i\boldsymbol{u}_{i,t}(d_{i,t} - \boldsymbol{u}_{i,t}^T\boldsymbol{w}_{i,t}) + \eta_i \sum_{j\in\mathcal{N}_i\setminus\{i\}} \beta_{i,j}(\boldsymbol{\phi}_j - \boldsymbol{w}_{i,t}),$$

where μ_i and η_i are the positive step sizes.

We note, however, that the cost function (3.3) is the sum of two convex functions, and hence lends itself to incremental solutions [56] where the stochastic gradient update is done in two steps, with the introduction of an intermediary estimate $\boldsymbol{\phi}_{i,t}$ at time t. In the ATC strategy [7] these steps are chosen such that

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

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

Replacing the quantity $\boldsymbol{\phi}_j$ by its instantaneous estimate $\boldsymbol{\phi}_{j,t+1}$, yields

$$\boldsymbol{\phi}_{i,t+1} = \boldsymbol{w}_{i,t} + \mu_i\boldsymbol{u}_{i,t}(d_{i,t} - \boldsymbol{u}_{i,t}^T\boldsymbol{w}_{i,t}), \tag{3.5}$$

$$\boldsymbol{w}_{i,t+1} = \boldsymbol{\phi}_{i,t+1} + \eta_i \sum_{j\in\mathcal{N}_i\setminus\{i\}} \beta_{i,j}(\boldsymbol{\phi}_{j,t+1} - \boldsymbol{\phi}_{i,t+1}). \tag{3.6}$$

Further introducing the weights $p_{i,i} = (1 - \eta_i + \eta_i\beta_{i,i})$ and $p_{i,j} = \eta_i\beta_{i,j}$ for $i \neq j$, (3.6) can be restated as

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

where the condition $\sum_{j=1}^N \beta_{i,j} = 1$ ensures that $\sum_{j=1}^N p_{i,j} = 1$. This yields the final form of the ATC diffusion recursion

$$\boldsymbol{\phi}_{i,t+1} = \boldsymbol{w}_{i,t} + \mu_i \boldsymbol{u}_{i,t}(d_{i,t} - \boldsymbol{u}_{i,t}^T \boldsymbol{w}_{i,t}), \tag{3.7}$$

$$\boldsymbol{w}_{i,t+1} = \sum_{j \in \mathcal{N}_i} p_{i,j} \boldsymbol{\phi}_{j,t+1}, \tag{3.8}$$

with the associated right stochastic combination matrix $\boldsymbol{P} = [p_{i,j}]$ satisfying $\boldsymbol{P}\mathbb{1}_N = \mathbb{1}_N$.

3.2 Distributed Estimation with Level Triggered Sampling

The well-known ATC full diffusion scheme (3.7)-(3.8) 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 [12]. 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 [18], or the error associated with an additional estimation layer [17]. In our distributed framework, due to communication constraints, a quantized version of the original parameter, $\xi_{i,t}^q$ is 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 quantization [14]. To clarify the framework, suppose we have a discrete time signal $\xi_{i,t}$

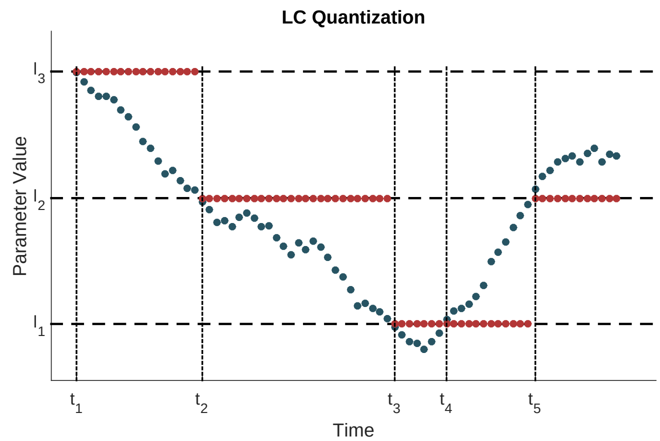


Figure 3.2: Illustration of the operation of the LC quantizer. Blue dots represent the original parameter values, while the red dots stand for the corresponding quantized versions.

as shown in Fig. 3.2 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. 3.2. 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 [14]. 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:

$$\xi_{i,t}^q = l_{i,t}. \tag{3.9}$$

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:

$$\xi_{i,t}^q = \xi_{i,t-1}^q. \tag{3.10}$$

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 $\xi^q_{i,t}$ changes compared to the previously-crossed level $\xi^q_{i,t-1}$. 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, which we represent using a single bit. In the second case, the parameter undergoes multiple level crossings - in which case the full location information of the new level value $\xi^q_{i,t}$ is directly coded, alongside a flag bit indicating the multiple level-crossing event. Hence, this second case results in a total of $\lceil \log_2(K) \rceil + 1$ expended bits. As shown, this approach significantly lowers the amount of communication while maintaining estimation performance.

3.3 Algorithm Description

In this section, we present the full algorithmic description of the proposed diffusion scheme with the level-crossing quantization [14]. 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 $\boldsymbol{\phi}^{\boldsymbol{q}}_{i,t+1}$. 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 $\boldsymbol{\phi}^{\boldsymbol{q}}_{i,t}$. In order to provide this information, the node i also keeps a record of the past estimated parameter values $\{\boldsymbol{\phi}^{\boldsymbol{q}}_i(k)\}_{k=1}^t$ that the neighboring nodes have related to its true $\{\boldsymbol{\phi}_i(k)\}_{k=1}^t$. The node i uses the most recent entry in this record, $\boldsymbol{\phi}^{\boldsymbol{q}}_{i,t}$, 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 $\boldsymbol{\phi}^{\boldsymbol{q}}_{i,t}$ 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 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 $\boldsymbol{\phi}^{\boldsymbol{q}}_{i,t+1}$ using (3.9) or (3.10) 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 $\boldsymbol{\phi}^{\boldsymbol{q}}_{j,t+1}$ using (3.9) or (3.10). The final estimate $\boldsymbol{w}_{i,t+1}$ is then constructed using the combination

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

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 by incorporating a separate LC quantizer for each individual component and maintaining separate communication channels between the nodes for the different components.

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 [18]. This approach is explored in the Experiments section.

Algorithm 1 ATC Diffusion LMS with the LC Quantization, M=1

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

  Initialization:

  $w_{i,0} = \phi^q_{i,0} = 0$

end for

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

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

    Local adaptation:

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

    Check for level crossing:

    if $\exists\, l_{i,t} \in \mathcal{S}$ such that $(\phi^q_{i,t} - l_{i,t})\,(\phi_{i,t+1} - l_{i,t}) < 0$ **then**

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

        Diffuse the direction of the crossing

      else

        Diffuse the location of the new level

      end if

      Locally store $\phi^q_{i,t+1} = l_{i,t}$ in record

    else

      Remain silent

      Locally set $\phi^q_{i,t+1} = \phi^q_{i,t}$

    end if

    Reconstruction:

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

      if node j is silent **then**

        Reconstruct as $\phi^q_{j,t+1} = \phi^q_{j,t}$

      else

        Reconstruct $\phi^q_{j,t+1}$ using the diffused information

      end if

    end for

    Combination:

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

  end for

end for

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

$$d_{i,t} = \boldsymbol{u}_{i,t}^T \boldsymbol{w}_{\boldsymbol{o}} + v_{i,t}, \tag{3.11}$$

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

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

$$\boldsymbol{\phi}_{i,t+1} = (\boldsymbol{I}_M - \mu_i \boldsymbol{u}_{i,t}\boldsymbol{u}_{i,t}^T)\boldsymbol{w}_{i,t} + \mu_i \boldsymbol{u}_{i,t} d_{i,t}, \tag{3.12}$$

$$\boldsymbol{w}_{i,t+1} = p_{i,i}\boldsymbol{\phi}_{i,t+1} + \sum_{j\in\mathcal{N}_i\setminus\{i\}} p_{i,j}\boldsymbol{\phi}_{j,t+1}^{\boldsymbol{q}}, \tag{3.13}$$

where the combination matrix $\boldsymbol{P}$ is taken to be stochastic, with its rows summing up to unity. We rewrite the expressions (3.12) and (3.13) as

$$\boldsymbol{\phi}_{i,t+1} = (\boldsymbol{I}_M - \mu_i \boldsymbol{u}_{i,t}\boldsymbol{u}_{i,t}^T)\boldsymbol{w}_{i,t} + \mu_i \boldsymbol{u}_{i,t} d_{i,t}, \tag{3.14}$$

$$\boldsymbol{w}_{i,t+1} = \sum_{j\in\mathcal{N}_i} p_{i,j}\boldsymbol{\phi}_{j,t+1} - \sum_{j\in\mathcal{N}_i\setminus\{i\}} p_{i,j}\boldsymbol{\alpha}_{j,t+1}, \tag{3.15}$$

by defining the quantization error for node j

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

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

$$\begin{aligned}
\boldsymbol{d}_t &\triangleq \mathrm{col}\{d_{1,t}, \ldots, d_{N,t}\} & \boldsymbol{v}_t &\triangleq \mathrm{col}\{v_{1,t}, \ldots, v_{N,t}\} \\
\underline{\boldsymbol{w}_{\boldsymbol{o}}} &\triangleq \mathrm{col}\{\boldsymbol{w}_{\boldsymbol{o}}, \ldots, \boldsymbol{w}_{\boldsymbol{o}}\} & \boldsymbol{U}_t &\triangleq \mathrm{diag}\{\boldsymbol{u}_{1,t}, \ldots, \boldsymbol{u}_{N,t}\} \\
\boldsymbol{M} &\triangleq \mathrm{diag}\{\mu_1 \boldsymbol{I}_M, \ldots, \mu_N \boldsymbol{I}_M\} & \boldsymbol{w}_t &\triangleq \mathrm{col}\{\boldsymbol{w}_{1,t}, \ldots, \boldsymbol{w}_{N,t}\} \\
\boldsymbol{\phi}_t &\triangleq \mathrm{col}\{\boldsymbol{\phi}_{1,t}, \ldots, \boldsymbol{\phi}_{N,t}\} & \boldsymbol{\phi}_t^{\boldsymbol{q}} &\triangleq \mathrm{col}\left\{\boldsymbol{\phi}_{1,t}^{\boldsymbol{q}}, \ldots, \boldsymbol{\phi}_{N,t}^{\boldsymbol{q}}\right\} \\
\boldsymbol{\alpha}_t &\triangleq \mathrm{col}\{\boldsymbol{\alpha}_{1,t}, \ldots, \boldsymbol{\alpha}_{N,t}\} & \boldsymbol{G} &\triangleq \boldsymbol{P} \otimes \boldsymbol{I}_M \\
\boldsymbol{P}_{\boldsymbol{C}} &\triangleq \boldsymbol{P} - \mathrm{diag}\{\boldsymbol{P}\} & \boldsymbol{G}_{\boldsymbol{C}} &\triangleq \boldsymbol{P}_{\boldsymbol{C}} \otimes \boldsymbol{I}_M
\end{aligned}$$

Using the above-defined quantities, the diffusion updates (3.14), (3.15) take the following global state-space form:

$$\boldsymbol{\phi}_{t+1} = (\boldsymbol{I}_{MN} - \boldsymbol{M}\boldsymbol{U}_t\boldsymbol{U}_t^T)\boldsymbol{w}_t + \boldsymbol{M}\boldsymbol{U}_t\boldsymbol{d}_t, \tag{3.16}$$

$$\boldsymbol{w}_{t+1} = \boldsymbol{G}\boldsymbol{\phi}_{t+1} - \boldsymbol{G}_{\boldsymbol{C}}\boldsymbol{\alpha}_{t+1}. \tag{3.17}$$

Similarly, the data model (3.11) can be expressed in terms of the global quantities as

$$\boldsymbol{d}_t = \boldsymbol{U}_t^T \underline{\boldsymbol{w}}_{\boldsymbol{o}} + \boldsymbol{v}_t. \tag{3.18}$$

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

$$\tilde{\boldsymbol{w}}_t \triangleq \underline{\boldsymbol{w}}_{\boldsymbol{o}} - \boldsymbol{w}_t,$$
$$\tilde{\boldsymbol{\phi}}_t \triangleq \underline{\boldsymbol{w}}_{\boldsymbol{o}} - \boldsymbol{\phi}_t.$$

After substituting (3.18) and subtracting both sides of (3.16), (3.17) from $\underline{\boldsymbol{w}}_{\boldsymbol{o}}$, the diffusion updates in terms of the deviation parameters take the following form:

$$\tilde{\boldsymbol{\phi}}_{t+1} = (\boldsymbol{I}_{MN} - \boldsymbol{M}\boldsymbol{U}_t\boldsymbol{U}_t^T)\tilde{\boldsymbol{w}}_t - \boldsymbol{M}\boldsymbol{U}_t\boldsymbol{v}_t, \tag{3.19}$$

$$\tilde{\boldsymbol{w}}_{t+1} = \boldsymbol{G}\tilde{\boldsymbol{\phi}}_{t+1} + \boldsymbol{G}_{\boldsymbol{C}}\boldsymbol{\alpha}_{t+1}, \tag{3.20}$$

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

The expressions (3.19), (3.20) can be expressed compactly as

$$\tilde{\boldsymbol{w}}_{t+1} = \boldsymbol{G}(\boldsymbol{I}_{MN} - \boldsymbol{M}\boldsymbol{U}_t\boldsymbol{U}_t^T)\tilde{\boldsymbol{w}}_t - \boldsymbol{G}\boldsymbol{M}\boldsymbol{U}_t\boldsymbol{v}_t + \boldsymbol{G}_{\boldsymbol{C}}\boldsymbol{\alpha}_{t+1}. \tag{3.21}$$

Assumption 1: The quantization error over the network $\boldsymbol{\alpha}_t$ has zero mean. This is a reasonable assumption for the analysis of quantization effects [19]. The applicability of the assumption is verified by our experiments in Chapter 4.

Taking expectations of both sides of (3.21) yields

$$E\left[\tilde{\boldsymbol{w}}_{t+1}\right] = \boldsymbol{G}(\boldsymbol{I}_{MN} - \boldsymbol{M}\boldsymbol{\Lambda})\, E\left[\tilde{\boldsymbol{w}}_t\right], \tag{3.22}$$

where $\boldsymbol{\Lambda} \triangleq \text{diag}\{\boldsymbol{\Lambda}_1, \ldots, \boldsymbol{\Lambda}_N\}$ is block diagonal. For mean stability and asymptotic unbiasedness of the distributed filter (3.12)-(3.13), we require that the spectral radius $|\boldsymbol{G}(\boldsymbol{I}_{MN} - \boldsymbol{M}\boldsymbol{\Lambda})| < 1$, which, noting that $\boldsymbol{G}$ is stochastic with non-negative entries, is equivalent to requiring

$$|(\boldsymbol{I}_{MN} - \boldsymbol{M}\boldsymbol{\Lambda})| < 1, \tag{3.23}$$

by Lemma 1 of [12]. Noting that the eigenvalues of the block diagonal matrix $\boldsymbol{I}_{MN} - \boldsymbol{M}\boldsymbol{\Lambda}$ is the union of the eigenvalues of its individual blocks $\boldsymbol{I}_M - \mu_i \boldsymbol{\Lambda}_i$ where $\boldsymbol{\Lambda}_i = \sigma_{u,i}^2 \boldsymbol{I}_M$; 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.

3.5 Mean-Square Stability

We utilize the weighted energy relation approach [19] 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 (3.21) yields:

$$\begin{aligned}
&\tilde{\boldsymbol{w}}_{t+1}^T \boldsymbol{\Sigma} \tilde{\boldsymbol{w}}_{t+1} = \\
&\quad \tilde{\boldsymbol{w}}_t^T (\boldsymbol{I}_{MN} - \boldsymbol{M}\boldsymbol{U}_t \boldsymbol{U}_t^T)^T \boldsymbol{G}^T \boldsymbol{\Sigma} \boldsymbol{G} (\boldsymbol{I}_{MN} - \boldsymbol{M}\boldsymbol{U}_t \boldsymbol{U}_t^T) \tilde{\boldsymbol{w}}_t \\
&\quad - 2\boldsymbol{v}_t^T \boldsymbol{U}_t^T \boldsymbol{M} \boldsymbol{G}^T \boldsymbol{\Sigma} \boldsymbol{G} (\boldsymbol{I}_{MN} - \boldsymbol{M}\boldsymbol{U}_t \boldsymbol{U}_t^T) \tilde{\boldsymbol{w}}_t \\
&\quad + 2\boldsymbol{\alpha}_{t+1}^T \boldsymbol{G}_{\boldsymbol{C}}^T \boldsymbol{\Sigma} \boldsymbol{G} (\boldsymbol{I}_{MN} - \boldsymbol{M}\boldsymbol{U}_t \boldsymbol{U}_t^T) \tilde{\boldsymbol{w}}_t \\
&\quad - 2\boldsymbol{v}_t^T \boldsymbol{U}_t^T \boldsymbol{M} \boldsymbol{G}^T \boldsymbol{\Sigma} \boldsymbol{G}_{\boldsymbol{C}} \boldsymbol{\alpha}_{t+1} \\
&\quad + \boldsymbol{v}_t^T \boldsymbol{U}_t^T \boldsymbol{M} \boldsymbol{G}^T \boldsymbol{\Sigma} \boldsymbol{G} \boldsymbol{M} \boldsymbol{U}_t \boldsymbol{v}_t \\
&\quad + \boldsymbol{\alpha}_{t+1}^T \boldsymbol{G}_{\boldsymbol{C}}^T \boldsymbol{\Sigma} \boldsymbol{G}_{\boldsymbol{C}} \boldsymbol{\alpha}_{t+1}.
\end{aligned} \tag{3.24}$$

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 (3.24) yields the following variance relation:

$$\begin{aligned} E\|\tilde{\boldsymbol{w}}_{t+1}\|^2_{\boldsymbol{\Sigma}} = & E\|\tilde{\boldsymbol{w}}_t\|^2_{\boldsymbol{\Sigma}'} \\ & + 2E\left[\boldsymbol{\alpha}_{t+1}^T \boldsymbol{G}_{\boldsymbol{C}}^T \boldsymbol{\Sigma} \boldsymbol{G}(\boldsymbol{I}_{MN} - \boldsymbol{M}\boldsymbol{U}_t\boldsymbol{U}_t^T)\tilde{\boldsymbol{w}}_t\right] \\ & - 2E\left[\boldsymbol{v}_t^T \boldsymbol{U}_t^T \boldsymbol{M} \boldsymbol{G}^T \boldsymbol{\Sigma} \boldsymbol{G}_{\boldsymbol{C}} \boldsymbol{\alpha}_{t+1}\right] \\ & + E\left[\boldsymbol{v}_t^T \boldsymbol{U}_t^T \boldsymbol{M} \boldsymbol{G}^T \boldsymbol{\Sigma} \boldsymbol{G} \boldsymbol{M} \boldsymbol{U}_t \boldsymbol{v}_t\right] \\ & + E\left[\boldsymbol{\alpha}_{t+1}^T \boldsymbol{G}_{\boldsymbol{C}}^T \boldsymbol{\Sigma} \boldsymbol{G}_{\boldsymbol{C}} \boldsymbol{\alpha}_{t+1}\right], \end{aligned} \tag{3.25}$$

where

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

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 $\boldsymbol{\Sigma}' \triangleq E[\boldsymbol{\Sigma}']$ in (3.25). Thus,

$$\begin{aligned} \boldsymbol{\Sigma}' = & \boldsymbol{G}^T\boldsymbol{\Sigma}\boldsymbol{G} - \boldsymbol{G}^T\boldsymbol{\Sigma}\boldsymbol{G}\boldsymbol{M}\boldsymbol{\Lambda} - \boldsymbol{\Lambda}\boldsymbol{M}\boldsymbol{G}^T\boldsymbol{\Sigma}\boldsymbol{G} \\ & + E\left[\boldsymbol{U}_t\boldsymbol{U}_t^T\boldsymbol{M}\boldsymbol{G}^T\boldsymbol{\Sigma}\boldsymbol{G}\boldsymbol{M}\boldsymbol{U}_t\boldsymbol{U}_t^T\right], \end{aligned} \tag{3.26}$$

where $\boldsymbol{\Lambda} \triangleq E\left[\boldsymbol{U}_t\boldsymbol{U}_t^T\right]$. Substituting the $(\boldsymbol{I}_{MN} - \boldsymbol{M}\boldsymbol{U}_t\boldsymbol{U}_t^T)\tilde{\boldsymbol{w}}_t$ expression from (3.19) into (3.25) yields the following final form of the variance relation

$$\begin{aligned} E\|\tilde{\boldsymbol{w}}_{t+1}\|^2_{\boldsymbol{\Sigma}} = & E\|\tilde{\boldsymbol{w}}_t\|^2_{\boldsymbol{\Sigma}'} \\ & + 2E\left[\boldsymbol{\alpha}_{t+1}^T \boldsymbol{G}_{\boldsymbol{C}}^T \boldsymbol{\Sigma} \boldsymbol{G} \tilde{\boldsymbol{\phi}}_{t+1}\right] \\ & + E\left[\boldsymbol{\alpha}_{t+1}^T \boldsymbol{G}_{\boldsymbol{C}}^T \boldsymbol{\Sigma} \boldsymbol{G}_{\boldsymbol{C}} \boldsymbol{\alpha}_{t+1}\right] \\ & + E\left[\boldsymbol{v}_t^T \boldsymbol{U}_t^T \boldsymbol{M} \boldsymbol{G}^T \boldsymbol{\Sigma} \boldsymbol{G} \boldsymbol{M} \boldsymbol{U}_t \boldsymbol{v}_t\right]. \end{aligned} \tag{3.27}$$

To capture the mean-square behavior of the adaptive network, we express the relations (3.26), (3.27) in a compact form by using the convenient vector notation [19]. In particular, we use the bvec$\{\cdot\}$ block vectorization operation [11] 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 $\mathrm{col}\{\boldsymbol{\sigma}_1, \ldots, \boldsymbol{\sigma}_N\}$, where

$\boldsymbol{\sigma}_j \overset{\triangle}{=} \mathrm{col}\{\mathrm{vec}\{\boldsymbol{\Sigma}_{1j}\}, \ldots, \mathrm{vec}\{\boldsymbol{\Sigma}_{Nj}\}\}$. We also use the block Kronecker product $\boldsymbol{A} \odot \boldsymbol{B}$ defined as having the (i,j)th block

$$[\boldsymbol{A} \odot \boldsymbol{B}]_{ij} = \begin{bmatrix} \boldsymbol{A}_{ij} \otimes \boldsymbol{B}_{11} & \ldots & \boldsymbol{A}_{ij} \otimes \boldsymbol{B}_{1N} \\ \vdots & \ddots & \vdots \\ \boldsymbol{A}_{ij} \otimes \boldsymbol{B}_{N1} & \ldots & \boldsymbol{A}_{ij} \otimes \boldsymbol{B}_{NN} \end{bmatrix}, \tag{3.28}$$

which is related to the bvec$\{\cdot\}$ operator via $\mathrm{bvec}\{ABC\} = (C^T \odot \ A)\,\mathrm{bvec}\{B\}$.

Defining $\boldsymbol{\sigma} \overset{\triangle}{=} \mathrm{bvec}\{\boldsymbol{\Sigma}\}$ and vectorizing both sides of (3.26) yields

$$\begin{aligned} \mathrm{bvec}\{\boldsymbol{\Sigma'}\} &= ((\boldsymbol{I}_{MN} \odot \boldsymbol{I}_{MN}) - (\boldsymbol{\Lambda M} \odot \boldsymbol{I}_{MN}) - (\boldsymbol{I}_{MN} \odot \boldsymbol{\Lambda M}))\,(\boldsymbol{G}^T \odot \boldsymbol{G}^T)\boldsymbol{\sigma} \\ &\quad + \mathrm{bvec}\{E\left[\boldsymbol{U}_t\boldsymbol{U}_t^T\boldsymbol{M}\boldsymbol{G}^T\boldsymbol{\Sigma}\boldsymbol{G}\boldsymbol{M}\boldsymbol{U}_t\boldsymbol{U}_t^T\right]\}. \end{aligned} \tag{3.29}$$

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 (3.29) can be vectorized by resorting to the Gaussian factorization theorem [11, 12]. 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 $\mathrm{bvec}\{\tilde{\boldsymbol{\Sigma}}\} = \mathrm{col}\{\tilde{\boldsymbol{\sigma}}_1, \ldots, \tilde{\boldsymbol{\sigma}}_j\}$ where $\tilde{\boldsymbol{\sigma}}_j = \mathrm{col}\{\tilde{\boldsymbol{\sigma}}_{1j}, \ldots, \tilde{\boldsymbol{\sigma}}_{Nj}\}$. Then, The (k,l)th block $\boldsymbol{\Gamma}_{kl}$ of $\boldsymbol{\Gamma} \overset{\triangle}{=} E\left[\boldsymbol{U}_t\boldsymbol{U}_t^T\tilde{\boldsymbol{\Sigma}}\boldsymbol{U}_t\boldsymbol{U}_t^T\right]$ is given by

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

with the vectorized form

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

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

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

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

$$\mathrm{bvec}\{\boldsymbol{\Gamma}\} = \mathcal{A}\,\mathrm{bvec}\{\tilde{\boldsymbol{\Sigma}}\} = \mathcal{A}(\boldsymbol{M} \odot \boldsymbol{M})(\boldsymbol{G}^T \odot \boldsymbol{G}^T)\boldsymbol{\sigma}. \tag{3.30}$$

Substituting (3.30) to (3.29) yields

$$\begin{aligned}\text{bvec}\{\boldsymbol{\Sigma'}\} = ((&\boldsymbol{I}_{MN} \odot \boldsymbol{I}_{MN}) - (\boldsymbol{\Lambda M} \odot \boldsymbol{I}_{MN}) \\ &- \ (\boldsymbol{I}_{MN} \odot \boldsymbol{\Lambda M}) + \mathcal{A}(\boldsymbol{M} \odot \boldsymbol{M}))\,(\boldsymbol{G}^T \odot \boldsymbol{G}^T)\boldsymbol{\sigma}.\end{aligned} \tag{3.31}$$

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 (3.27) can be verified to be

$$\begin{aligned}&E\left[\boldsymbol{v}_t^T \boldsymbol{U}_t^T \boldsymbol{M}\boldsymbol{G}^T \boldsymbol{\Sigma} \boldsymbol{G}\boldsymbol{M}\boldsymbol{U}_t \boldsymbol{v}_t\right] \\ &\quad = E\left[\text{Tr}\{\boldsymbol{v}_t^T \boldsymbol{U}_t^T \boldsymbol{M}\boldsymbol{G}^T \boldsymbol{\Sigma} \boldsymbol{G}\boldsymbol{M}\boldsymbol{U}_t \boldsymbol{v}_t\}\right] \\ &\quad = E\left[\text{Tr}\{\boldsymbol{\Sigma} \boldsymbol{G}\boldsymbol{M}\boldsymbol{U}_t \boldsymbol{v}_t \boldsymbol{v}_t^T \boldsymbol{U}_t^T \boldsymbol{M}\boldsymbol{G}^T\}\right] \\ &\quad = \text{Tr}\{\boldsymbol{\Sigma} \boldsymbol{G}\boldsymbol{M}\boldsymbol{H}\boldsymbol{M}\boldsymbol{G}^T\},\end{aligned} \tag{3.32}$$

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 $\boldsymbol{H} = (\boldsymbol{\Lambda_v} \otimes \boldsymbol{I}_M)\,\boldsymbol{\Lambda}$, where $\boldsymbol{\Lambda_v} \triangleq E\left[\boldsymbol{v}_t \boldsymbol{v}_t^T\right]$. Thus (3.32) becomes

$$\begin{aligned}&E\left[\boldsymbol{v}_t^T \boldsymbol{U}_t^T \boldsymbol{M}\boldsymbol{G}^T \boldsymbol{\Sigma} \boldsymbol{G}\boldsymbol{M}\boldsymbol{U}_t \boldsymbol{v}_t\right] \\ &\quad = \text{Tr}\{\boldsymbol{\Sigma} \boldsymbol{G}\boldsymbol{M}(\boldsymbol{\Lambda_v} \otimes \boldsymbol{I}_M)\boldsymbol{\Lambda}\boldsymbol{M}\boldsymbol{G}^T\} \\ &\quad = ((\boldsymbol{G}\boldsymbol{M} \odot \boldsymbol{G}\boldsymbol{M})\,\text{bvec}\{(\boldsymbol{\Lambda_v} \otimes \boldsymbol{I}_M)\,\boldsymbol{\Lambda}\})^T \boldsymbol{\sigma}.\end{aligned} \tag{3.33}$$

Similarly the remaining terms in the RHS of (3.27) can be verified to be

$$\begin{aligned}E\left[\boldsymbol{\alpha}_{t+1}^T \boldsymbol{G}_{\boldsymbol{C}}^T \boldsymbol{\Sigma} \boldsymbol{G} \tilde{\boldsymbol{\phi}}_{t+1}\right] &= ((\boldsymbol{G} \odot \boldsymbol{G}_{\boldsymbol{C}})\,\text{bvec}\{E[\boldsymbol{\alpha}_{t+1} \tilde{\boldsymbol{\phi}}_{t+1}^T]\})^T \boldsymbol{\sigma}, \\ E\left[\boldsymbol{\alpha}_{t+1}^T \boldsymbol{G}_{\boldsymbol{C}}^T \boldsymbol{\Sigma} \boldsymbol{G}_{\boldsymbol{C}} \boldsymbol{\alpha}_{t+1}\right] &= ((\boldsymbol{G}_{\boldsymbol{C}} \odot \boldsymbol{G}_{\boldsymbol{C}})\,\text{bvec}\{E[\boldsymbol{\alpha}_{t+1} \boldsymbol{\alpha}_{t+1}^T]\})^T \boldsymbol{\sigma}.\end{aligned} \tag{3.34}$$

Defining the quantities

$$\begin{aligned}\boldsymbol{b}_t &\triangleq (\boldsymbol{G}\boldsymbol{M} \odot \boldsymbol{G}\boldsymbol{M})\,\text{bvec}\{(\boldsymbol{\Lambda_v} \otimes \boldsymbol{I}_M)\,\boldsymbol{\Lambda}\} + (\boldsymbol{G} \odot \boldsymbol{G}_{\boldsymbol{C}})\,\text{bvec}\{E[\boldsymbol{\alpha}_t \tilde{\boldsymbol{\phi}}_t^T]\} \\ &\quad + (\boldsymbol{G}_{\boldsymbol{C}} \odot \boldsymbol{G}_{\boldsymbol{C}})\,\text{bvec}\{E[\boldsymbol{\alpha}_t \boldsymbol{\alpha}_t^T]\}, \\ \boldsymbol{F} &\triangleq ((\boldsymbol{I}_{MN} \odot \boldsymbol{I}_{MN}) - (\boldsymbol{\Lambda M} \odot \boldsymbol{I}_{MN}) - (\boldsymbol{I}_{MN} \odot \boldsymbol{\Lambda M}) \\ &\quad + \ \mathcal{A}\,(\boldsymbol{M} \odot \boldsymbol{M}))\,(\boldsymbol{G}^T \odot \boldsymbol{G}^T),\end{aligned} \tag{3.35}$$

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

$$E\|\tilde{\boldsymbol{w}}_{t+1}\|_{\boldsymbol{\sigma}}^2 = E\|\tilde{\boldsymbol{w}}_t\|_{\boldsymbol{F\sigma}}^2 + \boldsymbol{b}_{t+1}^T \boldsymbol{\sigma} \tag{3.36}$$

Iteration of (3.36) yields

$$\begin{aligned}
E\|\tilde{\boldsymbol{w}}_{t+1}\|^2_{\boldsymbol{\sigma}} &= E\|\tilde{\boldsymbol{w}}_t\|^2_{\boldsymbol{F\sigma}} + \boldsymbol{b}^T_{t+1}\boldsymbol{\sigma} \\
E\|\tilde{\boldsymbol{w}}_{t+1}\|^2_{\boldsymbol{F\sigma}} &= E\|\tilde{\boldsymbol{w}}_t\|^2_{\boldsymbol{F}^2\boldsymbol{\sigma}} + \boldsymbol{b}^T_{t+1}\boldsymbol{F\sigma} \\
&\vdots \\
E\|\tilde{\boldsymbol{w}}_{t+1}\|^2_{\boldsymbol{F}^{N^2M^2-1}\boldsymbol{\sigma}} &= E\|\tilde{\boldsymbol{w}}_t\|^2_{\boldsymbol{F}^{N^2M^2}\boldsymbol{\sigma}} + \boldsymbol{b}^T_{t+1}\boldsymbol{F}^{N^2M^2-1}\boldsymbol{\sigma}. \qquad (3.37)
\end{aligned}$$

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 (3.37) yields

$$E\|\tilde{\boldsymbol{w}}_{t+1}\|^2_{\boldsymbol{F}^{N^2M^2-1}\boldsymbol{\sigma}} = -p_{N^2M^2-1}E\|\tilde{\boldsymbol{w}}_t\|^2_{\boldsymbol{F}^{N^2M^2-1}\boldsymbol{\sigma}} - \ldots - p_0E\|\tilde{\boldsymbol{w}}_t\|^2_{\boldsymbol{\sigma}} + \boldsymbol{b}^T_{t+1}\boldsymbol{F}^{N^2M^2-1}\boldsymbol{\sigma},$$

which can be placed into the state space form

$$\mathcal{W}_{t+1} = \mathcal{F}\mathcal{W}_t + \mathcal{Y}_{t+1}, \qquad (3.38)$$

where

$$\mathcal{W}_t \triangleq \begin{bmatrix} E\|\tilde{\boldsymbol{w}}_t\|^2_{\boldsymbol{\sigma}} \\ E\|\tilde{\boldsymbol{w}}_t\|^2_{\boldsymbol{F\sigma}} \\ \vdots \\ E\|\tilde{\boldsymbol{w}}_t\|^2_{\boldsymbol{F}^{(N^2M^2-1)}\boldsymbol{\sigma}} \end{bmatrix}, \quad \mathcal{Y}_t \triangleq \begin{bmatrix} \boldsymbol{b}^T_t\boldsymbol{\sigma} \\ \boldsymbol{b}^T_t\boldsymbol{F\sigma} \\ \vdots \\ \boldsymbol{b}^T_t\boldsymbol{F}^{N^2M^2-1}\boldsymbol{\sigma} \end{bmatrix} \qquad (3.39)$$

$$\mathcal{F} \triangleq \begin{bmatrix} 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -p_0 & -p_1 & -p_2 & \ldots & -p_{N^2M^2-1} \end{bmatrix}. \qquad (3.40)$$

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

Assumption 2: The quantization error covariances $E\left[\boldsymbol{\alpha}_{t+1}\tilde{\boldsymbol{\phi}}^T_{t+1}\right]$ and $E\left[\boldsymbol{\alpha}_{t+1}\boldsymbol{\alpha}^T_{t+1}\right]$ remain bounded, with $\left\|E\left[\boldsymbol{\alpha}_{t+1}\tilde{\boldsymbol{\phi}}^T_{t+1}\right]\right\|_F, \left\|E\left[\boldsymbol{\alpha}_{t+1}\boldsymbol{\alpha}^T_{t+1}\right]\right\|_F < A$ for some $A > 0$ for the Frobenius norms.

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

$$\begin{aligned}\|\boldsymbol{b}_t\|_2 &\leq \|(\boldsymbol{GM} \odot \boldsymbol{GM})\,\text{bvec}\{(\boldsymbol{\Lambda_v} \otimes \boldsymbol{I}_M)\,\boldsymbol{\Lambda}\}\|_2 \\ &\quad + \|\boldsymbol{G} \odot \boldsymbol{G_C}\|_2 \left\|\text{bvec}\{E[\boldsymbol{\alpha}_t \tilde{\boldsymbol{\phi}}_t^T]\}\right\|_2 + \|\boldsymbol{G_C} \odot \boldsymbol{G_C}\|_2 \left\|\text{bvec}\{E[\boldsymbol{\alpha}_t \boldsymbol{\alpha}_t^T]\}\right\|_2 \\ &\leq \|(\boldsymbol{GM} \odot \boldsymbol{GM})\,\text{bvec}\{(\boldsymbol{\Lambda_v} \otimes \boldsymbol{I}_M)\,\boldsymbol{\Lambda}\}\|_2 + A\,(\|\boldsymbol{P}\|_2 + \|\boldsymbol{P_C}\|_2)\,\|\boldsymbol{P_C}\|_2 \triangleq B.\end{aligned}$$

Inspecting (3.40), 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 (3.38) can be solved for $\mathcal{W}_t$ in closed form as

$$\mathcal{W}_t = \mathcal{F}^t \mathcal{W}_0 + \sum_{n=0}^{t-1} \mathcal{F}^n \mathcal{Y}_{t-n}. \tag{3.41}$$

Using (3.41), we can obtain a bound for $\|\mathcal{W}_t\|_2$ as

$$\begin{aligned}\|\mathcal{W}_t\|_2 &\leq \|\mathcal{F}\|_2^t\,\|\mathcal{W}_0\|_2 + \sum_{n=0}^{t-1} \|\mathcal{F}\|_2^n\,\|\mathcal{Y}_{t-n}\|_2 \\ &\leq \|\boldsymbol{F}\|_2^t\,\|\mathcal{W}_0\|_2 + C\sum_{n=0}^{t-1} \|\boldsymbol{F}\|_2^n \\ &= \|\boldsymbol{F}\|_2^t\,\|\mathcal{W}_0\|_2 + C\,\frac{1 - \|\boldsymbol{F}\|_2^t}{1 - \|\boldsymbol{F}\|_2}\end{aligned} \tag{3.42}$$

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 (3.42), the mean-square stability condition reduces to the matrix $\boldsymbol{F}$ given by (3.35) being stable. Hence in order to ensure MS stability, it is sufficient that the step sizes μ_i are chosen such that the matrix $\boldsymbol{F}$ is stable.

3.6 Tracking Performance

In this section we will consider the tracking performance of the proposed scheme from the perspective of mean-square stability operating under a non-stationary environment. To clarify the framework, we consider a time-varying model for the observations made by a node i to satisfy

$$d_{i,t} = \boldsymbol{u}_{i,t}^T \boldsymbol{w}_{\boldsymbol{o},t} + v_{i,t}, \tag{3.43}$$

where $\boldsymbol{w}_{\boldsymbol{o},\boldsymbol{t}}$ is the time varying parameter to be estimated. This variation that needs to be tracked by the adaptive filter is modeled to follow a random walk model

$$\boldsymbol{w}_{\boldsymbol{o},t+1} = \boldsymbol{w}_{\boldsymbol{o},t} + \boldsymbol{q}_t, \tag{3.44}$$

where the process noise $\boldsymbol{q}_t$ is assumed to be zero-mean stationary and white with covariance $\boldsymbol{Q} = E\left[\boldsymbol{q}_t \boldsymbol{q}_t^T\right]$, independent of the regressor and observation noise processes.

Similarly, the data model (3.11) can be expressed in terms of the global quantities as $\boldsymbol{d}_t = \boldsymbol{U}_t^T \underline{\boldsymbol{w}}_{\boldsymbol{o},t} + \boldsymbol{v}_t$. Similar to the mean-square error analysis in the preceding sections, we define $\underline{\boldsymbol{w}}_{\boldsymbol{o},t} \triangleq \mathrm{col}\left\{\boldsymbol{w}_{\boldsymbol{o},t}, \ldots, \boldsymbol{w}_{\boldsymbol{o},t}\right\}$, and $\underline{\boldsymbol{q}}_t \triangleq \mathrm{col}\left\{\boldsymbol{q}_t, \ldots, \boldsymbol{q}_t\right\}$ in addition to the deviation parameters

$$\tilde{\boldsymbol{w}}_t \triangleq \underline{\boldsymbol{w}}_{\boldsymbol{o},t} - \boldsymbol{w}_t,$$
$$\tilde{\boldsymbol{\phi}}_t \triangleq \underline{\boldsymbol{w}}_{\boldsymbol{o},t} - \boldsymbol{\phi}_t.$$

The diffusion LMS recursions with the time-varying underlying parameter is given by the iterations

$$\boldsymbol{\phi}_{t+1} = (\boldsymbol{I}_{MN} - \boldsymbol{M}\boldsymbol{U}_t\boldsymbol{U}_t^T)\boldsymbol{w}_t + \boldsymbol{M}\boldsymbol{U}_t\boldsymbol{d}_t, \tag{3.45}$$

$$\boldsymbol{w}_{t+1} = \boldsymbol{G}\boldsymbol{\phi}_{t+1} - \boldsymbol{G}_{\boldsymbol{C}}\boldsymbol{\alpha}_{t+1}. \tag{3.46}$$

Subtracting both sides of (3.45)-(3.46) from $\boldsymbol{w}_{\boldsymbol{o},t+1}$ and substituting the random walk model (3.44) as well as the global data model (3.43) yields the diffusion updates in terms of the deviation parameters in the following form:

$$\tilde{\boldsymbol{\phi}}_{t+1} = (\boldsymbol{I}_{MN} - \boldsymbol{M}\boldsymbol{U}_t\boldsymbol{U}_t^T)\tilde{\boldsymbol{w}}_t - \boldsymbol{M}\boldsymbol{U}_t\boldsymbol{v}_t + \underline{\boldsymbol{q}}_t, \tag{3.47}$$

$$\tilde{\boldsymbol{w}}_{t+1} = \boldsymbol{G}\tilde{\boldsymbol{\phi}}_{t+1} + \boldsymbol{G}_{\boldsymbol{C}}\boldsymbol{\alpha}_{t+1}, \tag{3.48}$$

where we have used the relation $\boldsymbol{G}\underline{\boldsymbol{w}}_{\boldsymbol{o},t} = \underline{\boldsymbol{w}}_{\boldsymbol{o},t}$, which results from the stochastic nature of $\boldsymbol{P}$. The pair of equations (3.47)-(3.48) are expressed compactly as

$$\tilde{\boldsymbol{w}}_{t+1} = \boldsymbol{G}(\boldsymbol{I}_{MN} - \boldsymbol{M}\boldsymbol{U}_t\boldsymbol{U}_t^T)\tilde{\boldsymbol{w}}_t - \boldsymbol{G}\boldsymbol{M}\boldsymbol{U}_t\boldsymbol{v}_t + \boldsymbol{G}\underline{\boldsymbol{q}}_t + \boldsymbol{G}_{\boldsymbol{C}}\boldsymbol{\alpha}_{t+1}.$$

The associated energy conservation relation for the deviation parameter yields

$$\begin{aligned}
&\tilde{\boldsymbol{w}}_{t+1}^T\boldsymbol{\Sigma}\tilde{\boldsymbol{w}}_{t+1} = \\
&\quad \tilde{\boldsymbol{w}}_t^T(\boldsymbol{I}_{MN} - \boldsymbol{M}\boldsymbol{U}_t\boldsymbol{U}_t^T)^T\boldsymbol{G}^T\boldsymbol{\Sigma}\boldsymbol{G}(\boldsymbol{I}_{MN} - \boldsymbol{M}\boldsymbol{U}_t\boldsymbol{U}_t^T)\tilde{\boldsymbol{w}}_t \\
&\quad - 2\boldsymbol{v}_t^T\boldsymbol{U}_t^T\boldsymbol{M}\boldsymbol{G}^T\boldsymbol{\Sigma}\boldsymbol{G}(\boldsymbol{I}_{MN} - \boldsymbol{M}\boldsymbol{U}_t\boldsymbol{U}_t^T)\tilde{\boldsymbol{w}}_t \\
&\quad + 2\boldsymbol{\alpha}_{t+1}^T\boldsymbol{G}_{\boldsymbol{C}}^T\boldsymbol{\Sigma}\boldsymbol{G}(\boldsymbol{I}_{MN} - \boldsymbol{M}\boldsymbol{U}_t\boldsymbol{U}_t^T)\tilde{\boldsymbol{w}}_t \\
&\quad + 2\underline{\boldsymbol{q}}_t^T\boldsymbol{G}^T\boldsymbol{\Sigma}\boldsymbol{G}(\boldsymbol{I}_{MN} - \boldsymbol{M}\boldsymbol{U}_t\boldsymbol{U}_t^T)\tilde{\boldsymbol{w}}_t \\
&\quad - 2\boldsymbol{v}_t^T\boldsymbol{U}_t^T\boldsymbol{M}\boldsymbol{G}^T\boldsymbol{\Sigma}\boldsymbol{G}_{\boldsymbol{C}}\boldsymbol{\alpha}_{t+1} - 2\boldsymbol{v}_t^T\boldsymbol{U}_t^T\boldsymbol{M}\boldsymbol{G}^T\boldsymbol{\Sigma}\boldsymbol{G}\underline{\boldsymbol{q}}_t + 2\underline{\boldsymbol{q}}_t^T\boldsymbol{G}^T\boldsymbol{\Sigma}\boldsymbol{G}_{\boldsymbol{C}}\boldsymbol{\alpha}_{t+1} \\
&\quad + \boldsymbol{v}_t^T\boldsymbol{U}_t^T\boldsymbol{M}\boldsymbol{G}^T\boldsymbol{\Sigma}\boldsymbol{G}\boldsymbol{M}\boldsymbol{U}_t\boldsymbol{v}_t + \boldsymbol{\alpha}_{t+1}^T\boldsymbol{G}_{\boldsymbol{C}}^T\boldsymbol{\Sigma}\boldsymbol{G}_{\boldsymbol{C}}\boldsymbol{\alpha}_{t+1} + \underline{\boldsymbol{q}}_t^T\boldsymbol{G}^T\boldsymbol{\Sigma}\boldsymbol{G}\underline{\boldsymbol{q}}_t
\end{aligned} \tag{3.49}$$

Noting that $\boldsymbol{v}_t$ is zero-mean and independent of $\boldsymbol{U}_t$, $\tilde{\boldsymbol{w}}_t$ and $\underline{\boldsymbol{q}}_t$, and taking the expected value of both sides of (3.49) yields the following variance relation:

$$\begin{aligned}
&E\|\tilde{\boldsymbol{w}}_{t+1}\|_{\boldsymbol{\Sigma}}^2 = E\|\tilde{\boldsymbol{w}}_t\|_{\boldsymbol{\Sigma}'}^2 \\
&\quad + 2E\left[\boldsymbol{\alpha}_{t+1}^T\boldsymbol{G}_{\boldsymbol{C}}^T\boldsymbol{\Sigma}\boldsymbol{G}(\boldsymbol{I}_{MN} - \boldsymbol{M}\boldsymbol{U}_t\boldsymbol{U}_t^T)\tilde{\boldsymbol{w}}_t\right] \\
&\quad - 2E\left[\boldsymbol{v}_t^T\boldsymbol{U}_t^T\boldsymbol{M}\boldsymbol{G}^T\boldsymbol{\Sigma}\boldsymbol{G}_{\boldsymbol{C}}\boldsymbol{\alpha}_{t+1}\right] + 2E\left[\underline{\boldsymbol{q}}_t^T\boldsymbol{G}^T\boldsymbol{\Sigma}\boldsymbol{G}_{\boldsymbol{C}}\boldsymbol{\alpha}_{t+1}\right] \\
&\quad + E\left[\boldsymbol{v}_t^T\boldsymbol{U}_t^T\boldsymbol{M}\boldsymbol{G}^T\boldsymbol{\Sigma}\boldsymbol{G}\boldsymbol{M}\boldsymbol{U}_t\boldsymbol{v}_t\right] \\
&\quad + E\left[\boldsymbol{\alpha}_{t+1}^T\boldsymbol{G}_{\boldsymbol{C}}^T\boldsymbol{\Sigma}\boldsymbol{G}_{\boldsymbol{C}}\boldsymbol{\alpha}_{t+1}\right] + E\left[\underline{\boldsymbol{q}}_t^T\boldsymbol{G}^T\boldsymbol{\Sigma}\boldsymbol{G}\underline{\boldsymbol{q}}_t\right].
\end{aligned} \tag{3.50}$$

Substituting the $(\boldsymbol{I}_{MN} - \boldsymbol{M}\boldsymbol{U}_t\boldsymbol{U}_t^T)\tilde{\boldsymbol{w}}_t$ expression from (3.47) into (3.50) yields the following final form of the variance relation

$$
\begin{aligned}
E\|\tilde{\boldsymbol{w}}_{t+1}\|^2_{\boldsymbol{\Sigma}} = {} & E\|\tilde{\boldsymbol{w}}_t\|^2_{\boldsymbol{\Sigma}'} \\
& + 2E\left[\boldsymbol{\alpha}_{t+1}^T \boldsymbol{G}_{\boldsymbol{C}}^T \boldsymbol{\Sigma} \boldsymbol{G} \tilde{\boldsymbol{\phi}}_{t+1}\right] \\
& + E\left[\boldsymbol{\alpha}_{t+1}^T \boldsymbol{G}_{\boldsymbol{C}}^T \boldsymbol{\Sigma} \boldsymbol{G}_{\boldsymbol{C}} \boldsymbol{\alpha}_{t+1}\right] \\
& + E\left[\boldsymbol{v}_t^T \boldsymbol{U}_t^T \boldsymbol{M} \boldsymbol{G}^T \boldsymbol{\Sigma} \boldsymbol{G} \boldsymbol{M} \boldsymbol{U}_t \boldsymbol{v}_t\right] \\
& + E\left[\underline{\boldsymbol{q}}_t^T \boldsymbol{G}^T \boldsymbol{\Sigma} \boldsymbol{G} \underline{\boldsymbol{q}}_t\right].
\end{aligned} \tag{3.51}
$$

We note that the variance relation (3.51) is almost identical to the expression for the case of a stationary environment (3.27). Consequently, we observe that we can establish a mean-square stability result by invoking *Assumption 2* and obtaining a bound on the steady state mean-square deviation, in similar vein to the previous formulation. To this end, we defining $\boldsymbol{F}$ again as in (3.35) and introduce

$$
\boldsymbol{b}_t \triangleq (\boldsymbol{G} \odot \boldsymbol{G})\,\text{bvec}\{\boldsymbol{Q} \otimes \boldsymbol{I}_M\} + (\boldsymbol{G}\boldsymbol{M} \odot \boldsymbol{G}\boldsymbol{M})\,\text{bvec}\{(\boldsymbol{\Lambda}_{\boldsymbol{v}} \otimes \boldsymbol{I}_M)\,\boldsymbol{\Lambda}\} \tag{3.52}
$$

$$
+ (\boldsymbol{G} \odot \boldsymbol{G}_{\boldsymbol{C}})\,\text{bvec}\{E[\boldsymbol{\alpha}_t \tilde{\boldsymbol{\phi}}_t^T]\} + (\boldsymbol{G}_{\boldsymbol{C}} \odot \boldsymbol{G}_{\boldsymbol{C}})\,\text{bvec}\{E[\boldsymbol{\alpha}_t \boldsymbol{\alpha}_t^T]\}, \tag{3.53}
$$

which gives rise to the bound

$$
\begin{aligned}
\|\boldsymbol{b}_t\|_2 \le {} & \|(\boldsymbol{G} \odot \boldsymbol{G})\,\text{bvec}\{\boldsymbol{Q} \otimes \boldsymbol{I}_M\}\|_2 + \|(\boldsymbol{G}\boldsymbol{M} \odot \boldsymbol{G}\boldsymbol{M})\,\text{bvec}\{(\boldsymbol{\Lambda}_{\boldsymbol{v}} \otimes \boldsymbol{I}_M)\,\boldsymbol{\Lambda}\}\|_2 \\
& + \|\boldsymbol{G} \odot \boldsymbol{G}_{\boldsymbol{C}}\|_2 \left\|\text{bvec}\{E[\boldsymbol{\alpha}_t \tilde{\boldsymbol{\phi}}_t^T]\}\right\|_2 + \|\boldsymbol{G}_{\boldsymbol{C}} \odot \boldsymbol{G}_{\boldsymbol{C}}\|_2 \left\|\text{bvec}\{E[\boldsymbol{\alpha}_t \boldsymbol{\alpha}_t^T]\}\right\|_2 \\
& \le B + \|(\boldsymbol{G} \odot \boldsymbol{G})\,\text{bvec}\{\boldsymbol{Q} \otimes \boldsymbol{I}_M\}\|_2 .
\end{aligned}
$$

Following a state space formulation analogous to the stability analysis for the stationary environments, we arrive at a similar state-space representation of the form (3.38), by substituting (3.53) for $\boldsymbol{b}_t$. We note that the arguments leading up to (3.41) (3.42) still hold, yielding a stability condition in terms of the step sizes μ_i such that spectral norm of $\boldsymbol{F}$ is smaller than 1.

Chapter 4

Numerical Experiments

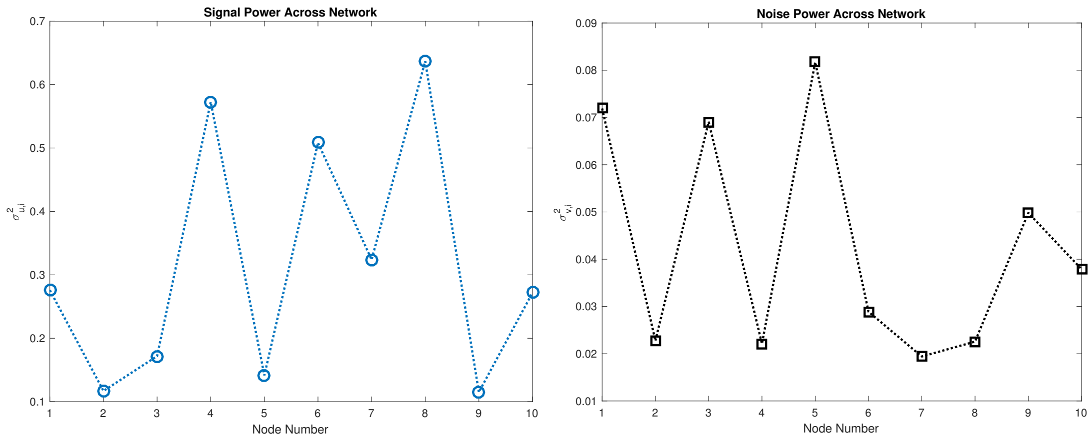
4.1 Reduction of the Load on the Communication Resources

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

$$d_{i,t} = \boldsymbol{u}_{i,t}^T \boldsymbol{w_o} + v_{i,t}, \qquad i = 1, \ldots, N. \tag{4.1}$$

The regressor data over the nodes $\boldsymbol{u}_{i,t}$ are taken to be zero mean i.i.d. Gaussian variables with the associated standard deviations $\sigma_{u,i}$ chosen randomly from the interval $(0.3, 0.8)$. The observation noises are generated from a Normal distribution with the standard deviations $\sigma_{v,i}$ chosen randomly from the interval $(0.1, 0.3)$. In Fig. 4.1 and Fig. 4.2, we depict the network topology and statistical profile illustrating how the signal power and the noise power vary across the network for this configuration.

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 have opted to change the source statistics in the middle of the course of the



(a) Regressor variances over the nodes (b) The noise profile

Figure 4.1: The statistical profile of the network

simulations in order to observe how well the proposed algorithm is able to adapt under concept drifts.

We use a Metropolis combination rule to generate the network matrix $\boldsymbol{P}$ as in

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

where $N_i \triangleq \|\mathcal{N}_i \backslash \{i\}\|$ is the number of neighboring nodes. We also use a randomly selected network adjacency matrix given by

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

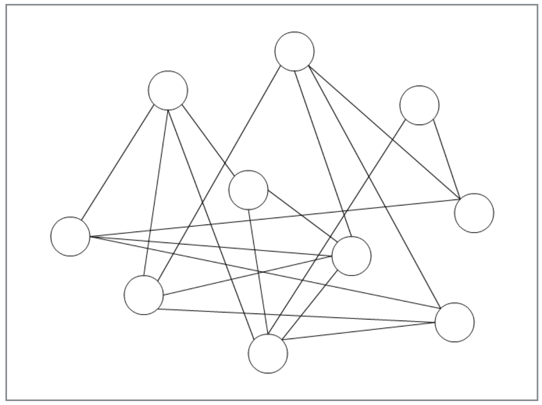


Figure 4.2: The network topology

For the first two parts of the experiments, the proposed event-triggered algorithm is compared against the algorithm detailed in [18]. In order to better illustrate the efficacy of the proposed event-triggered inter-nodal communication scheme, as have been remarked earlier as a way to deal with the case of a non-scalar ($M > 1$) unknown parameter, we have configured the nodes such that they cycle through the entries of their intermediary estimates $\boldsymbol{\phi}_{i,t}$ in a round-robin fashion, and exchange only some given L dimensions out of a total of M in a single time instant. A recipient node $j \in \mathcal{N}_i \backslash \{i\}$ makes up for the non-communicated dimensions of $\boldsymbol{\phi}_{i,t}$ by substituting the corresponding entries of its own intermediary estimate $\boldsymbol{\phi}_{j,t}$. As an example, for an $L = 1, M = 3$ system, a node i communicates the entries of its intermediary estimate $\boldsymbol{\phi}_{i,t}$ at the time instants $t = 1, \ldots, 4$ with the following order:

$$\boldsymbol{\phi}_{i,1}^{\text{com}} = \phi_{i,1}^{(1)}, \qquad \boldsymbol{\phi}_{i,2}^{\text{com}} = \phi_{i,2}^{(2)}, \qquad \boldsymbol{\phi}_{i,3}^{\text{com}} = \phi_{i,3}^{(3)}, \qquad \boldsymbol{\phi}_{i,4}^{\text{com}} = \phi_{i,4}^{(1)}, \tag{4.2}$$

where $\phi_{i,t}^{(k)}$ is the k^{th} dimension of the intermediary estimate $\boldsymbol{\phi}_{i,t}$ of a node i which it communicates to its neighbors at time t.

Within this context, we evaluate the communication reduction performance of the proposed LC-quantization based algorithm with respect to the algorithm proposed in [18]. For this purpose we consider the sequential variant of the algorithm in [18] with the parameters $M = 10$, $L = 1$, where only one entry of intermediate estimates is exchanged by the nodes at each round in a sequential order, according to (4.2).

Throughout, we use the MSD (mean-square deviation) given by

$$\text{MSD} = \frac{1}{N}\sum_{i=1}^{N} E\|\tilde{\boldsymbol{w}}_{i,t}\|^2$$

as the measure of estimation accuracy for the distributed estimation schemes.

In Fig. 4.3, the MSD performance of the proposed algorithm is demonstrated, where as a reference, we have considered an implementation of the algorithm in [18] with an adaptive Lloyd-Max quantizer, alongside a no-quantization (scalar) implementation. We have selected the extent of the quantization levels $|l_K - l_1|$ such that we do not suffer significantly from saturation effects, and similarly the number of quantization levels have been chosen such that no further significant improvement is obtained on the MSD performance by a finer partitioning of the interval $[l_1, l_K]$. With these constraints, we have noted that 53 quantization levels for the LC algorithm and 31 quantization levels for the baseline algorithm were sufficient. We have picked a step size of $\mu = 0.05$ for both systems. Individual runs of the simulation were averaged over 100 independent trials.

Observing the results obtained from these simulations, we note that the convergence rate of the scalar (infinite-precision) 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 practically identical. We note that we have aimed to obtain uniform steady-state MSD values across the three systems, allowing for a fair comparison in terms of the associated convergence speeds. Ultimately, the results indicate that the proposed algorithm incurs a small trade-off in rate of convergence in exchange for gains in communication efficiency, which will be discussed shortly. On another note, it is observed that the proposed algorithm does not compromise on its ability to track an abrupt change in the source statistics.

In Fig. 4.4, we present the communication load that each algorithm incurs over 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 algorithm in terms of the

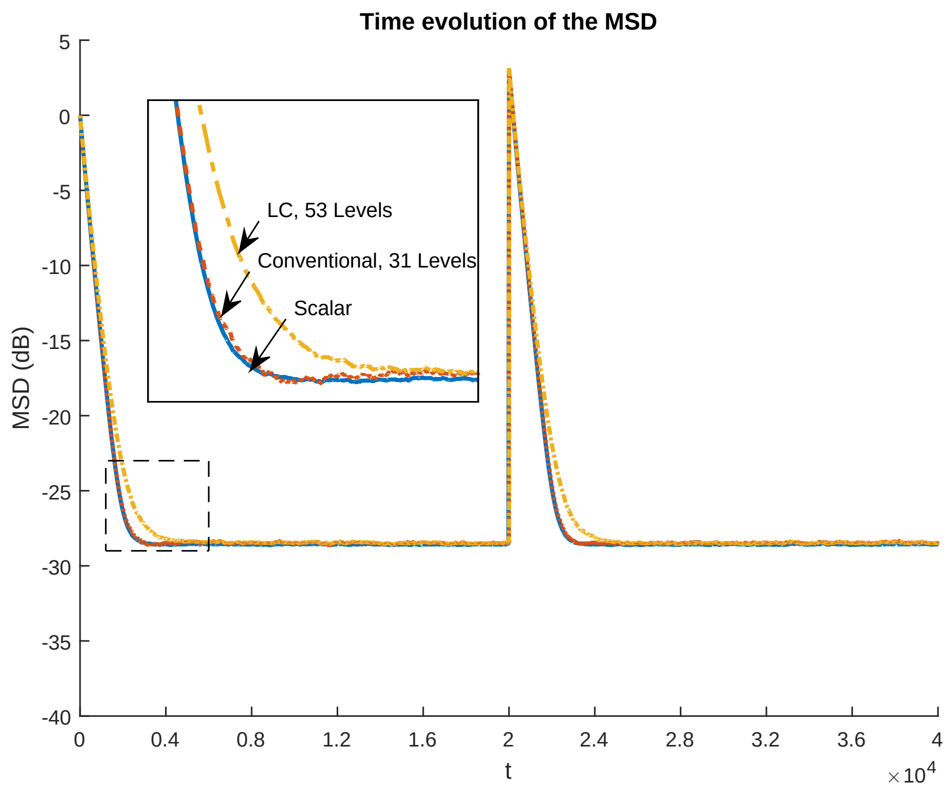


Figure 4.3: The global MSD curves of the proposed algorithm, represented with the label 'LC', in comparison with the conventional quantization and the scalar diffusion algorithms ($N = 10$, $M = 10$). The Magnified section highlights the rates of convergence of the algorithms. Source statistics change at time $t = 2\times10^4$

total number of bits exchanged between the nodes across the entire network with respect to the algorithm that uses the conventional quantization. Particularly, for this $N = 10$ node network, we note that the proposed algorithm induces a communication load which is reduced approximately by a factor of 10^3 compared with the reference implementation with the same steady state MSD value. We remark that this reduction in the LC algorithm is primarily achieved due to the dominance of the single-bit communication mode in the later stages of adaptation, compared with the conventional quantization implementation where the full location information is encoded and transmitted at every time instant regardless of how far along the system is in its adaptation course. We also observe that at the time instant at which the source statistics change, there is a sudden increase in the number of bits used by the proposed algorithm, which can be addressed

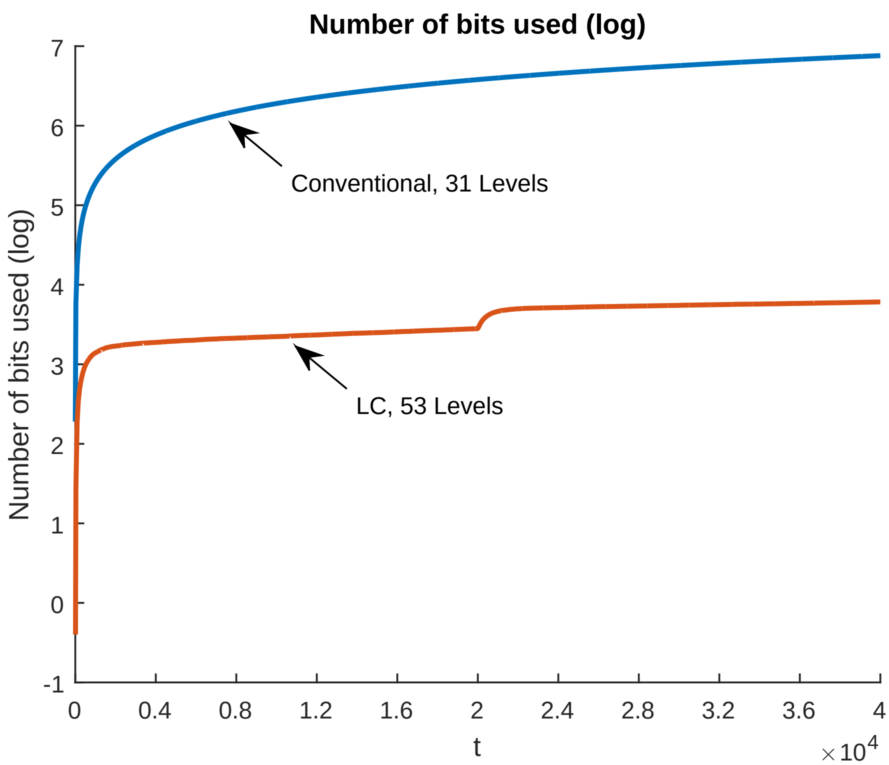


Figure 4.4: Time evolution of the number of bits transmitted across the network. Sudden increase in the 'LC' curve corresponds to the time instant at which the source statistics are changed.

by the outset of frequent multiple level crossings during the initial adjustment to the sudden change in the parameter of interest, which requires multiple bits to encode. We emphasize that the system is seen to quickly move back to a state of operation dominated by single level crossings (a single-bit mode of operation). We stress further that we achieve this improvement with relatively little implementation complexity since we have shown that using a simple non-adaptive quantizer is sufficient to realize the improvements.

4.2 Effect of the Choice of Quantization Levels

For the second part of the experiments, in order to observe the possible effects of the choice quantization levels, we simulate the algorithms within an identical experimental setup - except that the number of quantization levels are no longer optimized as was the case in prior simulations. To this end, we have arbitrarily chosen 25 quantization levels for the both the proposed and the baseline algorithm. We use the same distributed network with connections given in Fig. 4.2. We have used a step size of $\mu = 0.05$ and the results are averaged over 100 independent trials.

We present the MSD performances of the algorithms in Fig. 4.5. We observe that when sub-optimal quantization levels are used, the baseline algorithm exhibits a greater resilience compared to the proposed algorithm both in terms of the convergence rate and the steady-state MSD, indicating that for the estimation performance, the proposed algorithm is characterized by a more pronounced dependence on the choice of levels. We also note that neither of the quantized algorithms were able to reach the steady-state performance of the infinite-precision (scalar) diffusion due to the deliberate poor selection of the number of quantization levels.

These results are partly 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 $\boldsymbol{\alpha}_i$ necessitates that it has zero mean, with $E[\boldsymbol{\alpha}_i] = 0$ [19]. However, when such a low number of quantization levels are selected, this model ceases to be applicable and the quantized algorithms are no longer guaranteed to converge to the steady-state MSD values of the scalar diffusion algorithm.

We also present the communication load comparison of the two schemes in Fig. 4.6, where the proposed algorithm appears to realize ever-greater savings compared to the baseline, which can be accounted for by the simple observation that a coarser set of levels result in a smaller number of level crossings, and in turn, fewer bits transmitted over the network. When viewed alongside the poorer MSD performance, however, it is seen that an improper choice of quantization levels ultimately reflects a trade-off between the communication load imposed on the network and the estimation performance.

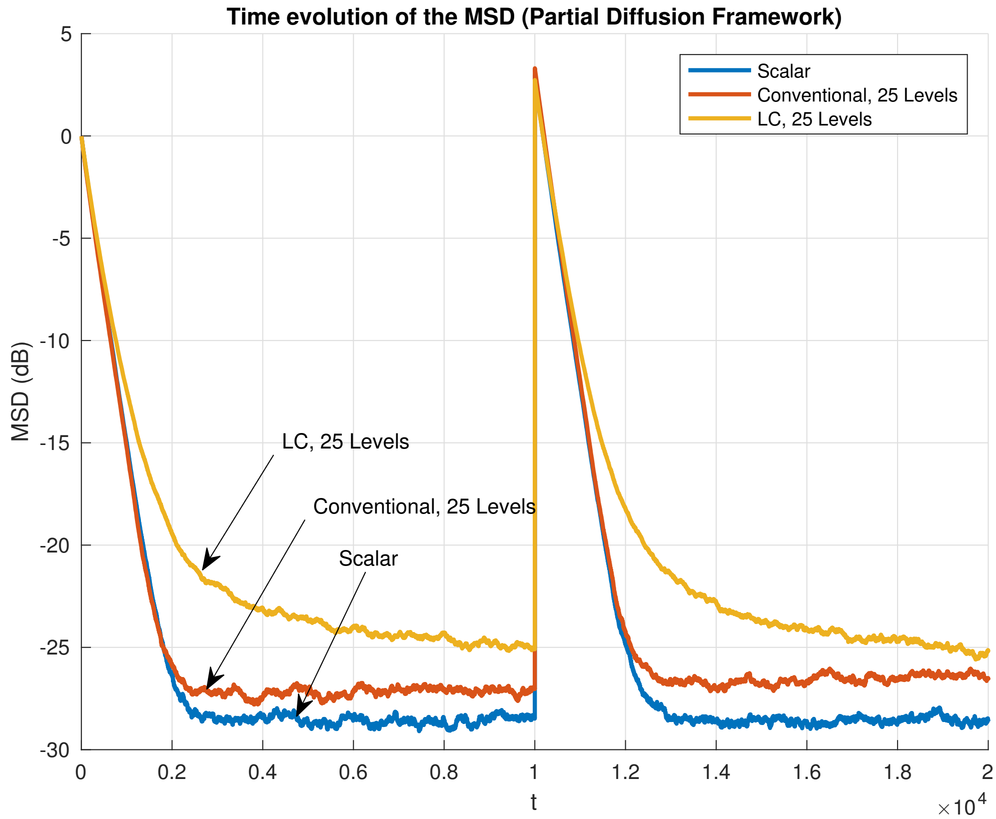


Figure 4.5: The global MSD curves of the proposed algorithm, represented with the label 'LC', in comparison with the conventional quantization and the scalar diffusion algorithms with sub-optimal quantization levels ($N = 10$, $M = 10$). Source statistics change at time $t = 10^4$.

4.3 Theoretical Verification

In this part of the experiments, we have sought to verify that the MSD time-evolution predicted by (3.37) is consistent with the simulation results. To this end, we have simulated the proposed algorithm on an $M = 7, N = 2$ network with 53 quantization levels. Due to the analytical intractability brought about by the terms $E\left[\boldsymbol{\alpha}_{t+1}\tilde{\boldsymbol{\phi}}_{t+1}^T\right]$ and $E\left[\boldsymbol{\alpha}_{t+1}\boldsymbol{\alpha}_{t+1}^T\right]$ within our framework, however, 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.

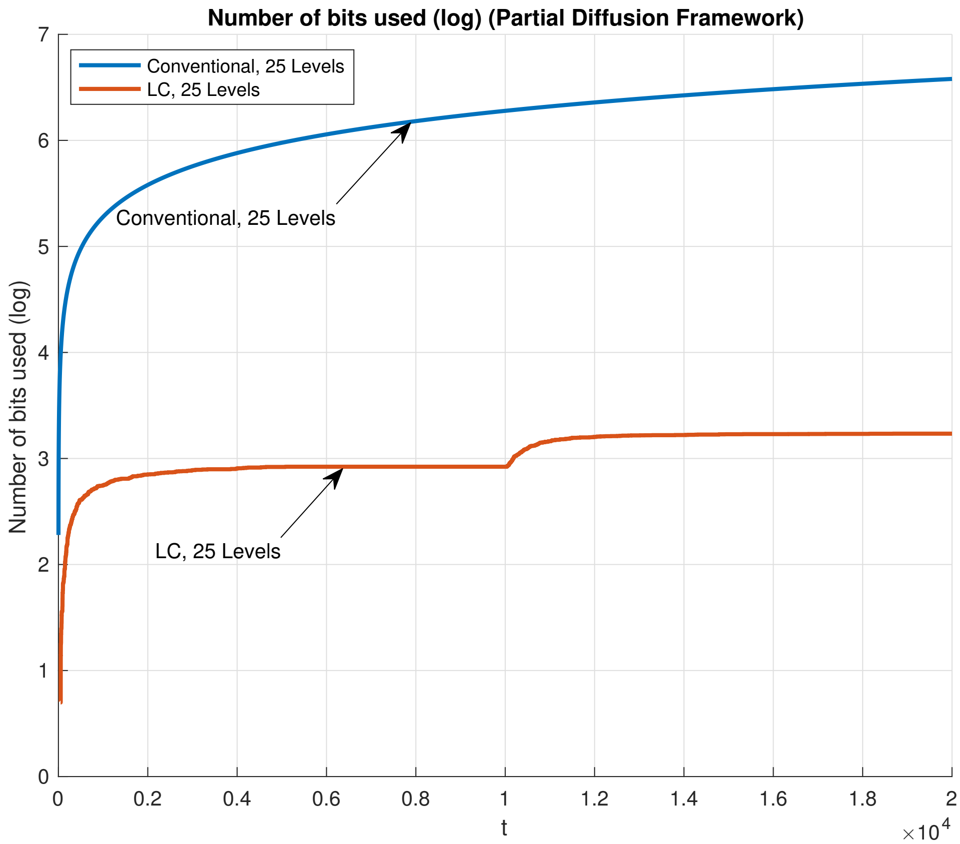


Figure 4.6: Time evolution of the number of bits transmitted by the algorithms across network with sub-optimal quantization levels ($N = 10$, $M = 10$). Sudden increase in the 'LC' curve corresponds to the time at which the source statistics are changed.

Throughout, we have used a step size of $\mu = 0.05$ and all the results are averaged over 10 independent trials. In Fig. 4.7, we present the MSD performance of the proposed algorithm alongside the response predicted by the 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 $E\left[\boldsymbol{\alpha}_{t+1}\tilde{\boldsymbol{\phi}}_{t+1}^T\right]$ and $E\left[\boldsymbol{\alpha}_{t+1}\boldsymbol{\alpha}_{t+1}^T\right]$ terms.

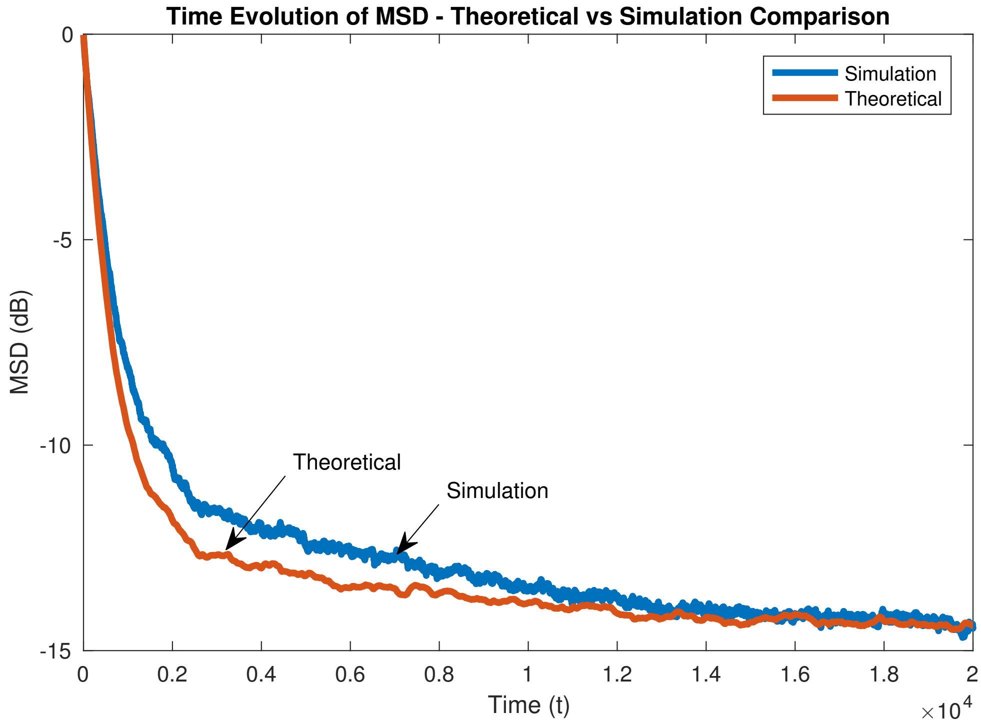


Figure 4.7: The global MSD curve of the proposed algorithm, represented with the label 'LC', in comparison with the theoretical MSD results over the $M = 7, N = 2$ system.

4.4 Operation under High Dimensional Data

In the last 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. 4.2. 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.

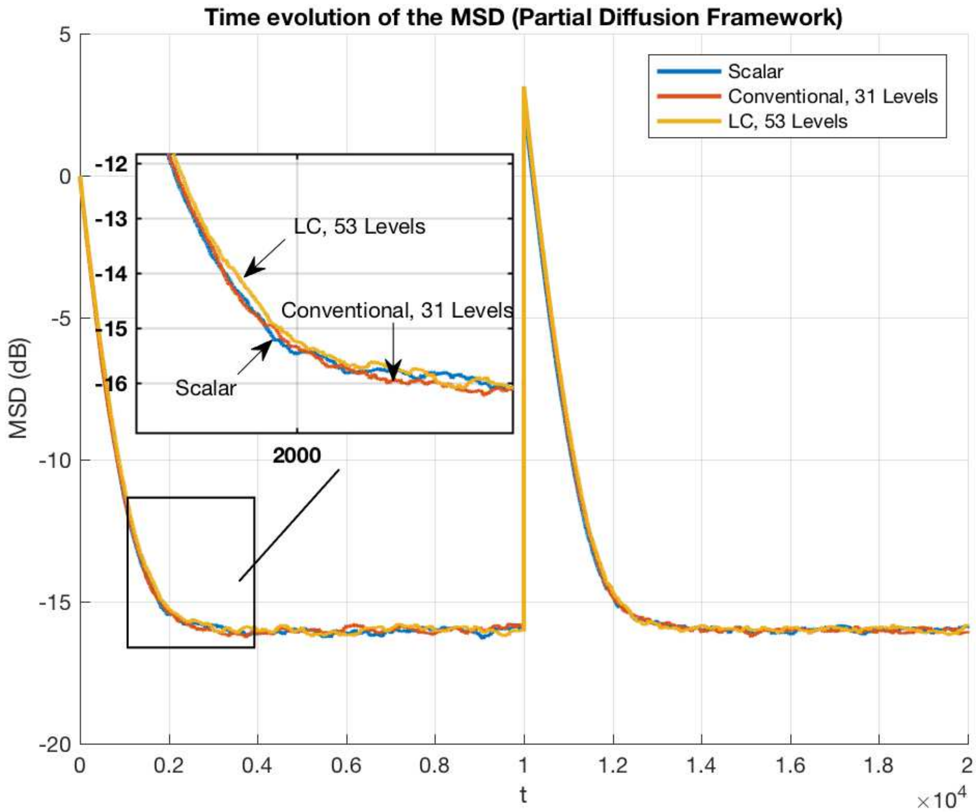


Figure 4.8: The global MSD curves of the proposed algorithm, represented with the label 'LC', in comparison with the conventional quantization and the scalar diffusion algorithms over high dimensional data ($N = 10$, $M = 100$). Magnified figure presents the transient performance of the algorithms. Source statistics change at time $t = 10^4$.

We present the MSD performance of the proposed algorithm in comparison with the sequential variant of the algorithm in [18] with the parameters $M = 100$, $L = 1$ in Fig. 4.8. 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 of transmission, 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

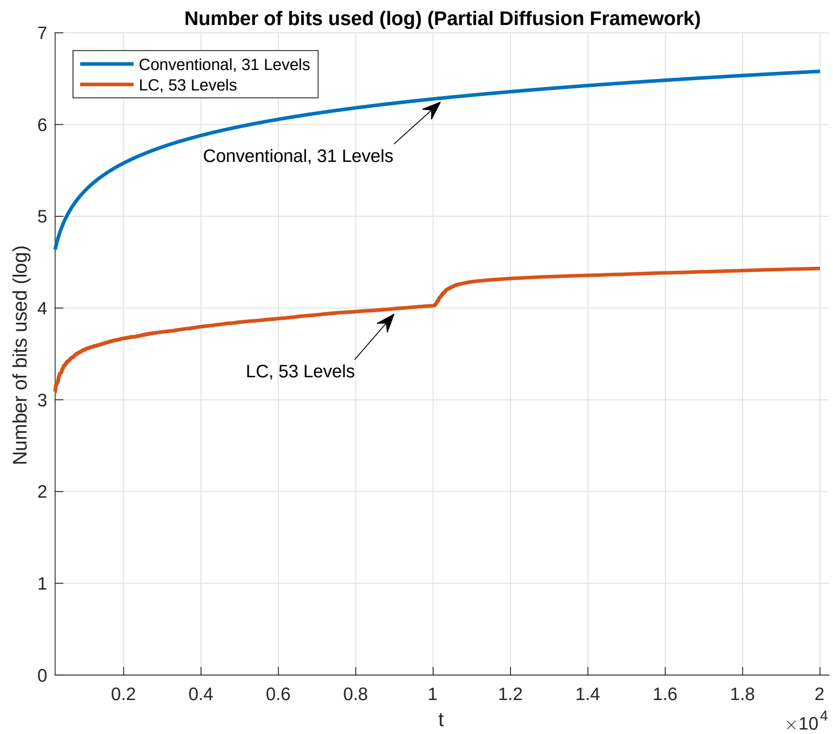


Figure 4.9: Time evolution of the number of bits transmitted by the algorithms across network over high dimensional data ($N = 10$, $M = 100$). Sudden increase in the 'LC' curve corresponds to the time at which the source statistics are changed.

algorithms.

In Fig. 4.9, 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.

Chapter 5

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 neighbours 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 and mean square senses, and we demonstrated that it achieves up to a 10^3 fold reduction in the communication load imposed on the network.

Bibliography

[1] 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, pp. 155–171, May 2013.

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

[3] S. Werner, Y. F. Huang, M. L. R. de Campos, and V. Koivunen, "Distributed parameter estimation with selective cooperation," in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 2849–2852, April 2009.

[4] N. Assimakis, M. Adam, M. Koziri, S. Voliotis, and K. Asimakis, "Optimal decentralized kalman filter and lainiotis filter," *Digital Signal Processing*, vol. 23, no. 1, pp. 442 – 452, 2013.

[5] D. Estrin, L. Girod, G. Pottie, and M. Srivastava, "Instrumenting the world with wireless sensor networks," in *Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01). 2001 IEEE International Conference on*, vol. 4, pp. 2033–2036 vol.4, 2001.

[6] D. Li, K. Wong, Y. H. Hu, and A. Sayeed, "Detection, classification, and tracking of targets," *Signal Processing Magazine, IEEE*, vol. 19, pp. 17–29, Mar 2002.

[7] J. Chen and A. Sayed, “Diffusion adaptation strategies for distributed optimization and learning over networks,” *Signal Processing, IEEE Transactions on*, vol. 60, pp. 4289–4305, Aug 2012.

[8] S. Chouvardas, K. Slavakis, and S. Theodoridis, “Trading off complexity with communication costs in distributed adaptive learning via krylov subspaces for dimensionality reduction,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, pp. 257–273, April 2013.

[9] J. Fernandez-Bes, L. A. Azpicueta-Ruiz, J. Arenas-Garca, and M. T. Silva, “Distributed estimation in diffusion networks using affine least-squares combiners,” *Digital Signal Processing*, vol. 36, pp. 1 – 14, 2015.

[10] J.-J. Xiao, A. Ribeiro, Z.-Q. Luo, and G. Giannakis, “Distributed compression-estimation using wireless sensor networks,” *Signal Processing Magazine, IEEE*, vol. 23, pp. 27–41, July 2006.

[11] C. G. Lopes and A. H. Sayed, “Diffusion least-mean squares over adaptive networks: Formulation and performance analysis,” *IEEE Transactions on Signal Processing*, vol. 56, no. 7, pp. 3122–3136, 2008.

[12] F. S. Cattivelli and A. H. Sayed, “Diffusion lms strategies for distributed estimation,” *IEEE Transactions on Signal Processing*, vol. 58, no. 3, pp. 1035–1048, 2010.

[13] Z. Xiaochuan and A. Sayed, “Single-link diffusion strategies over adaptive networks,” in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pp. 3749–3752, March 2012.

[14] J. W. Mark and T. Todd, “A nonuniform sampling approach to data compression,” *Communications, IEEE Transactions on*, vol. 29, pp. 24–32, Jan 1981.

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

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

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

[18] 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. 472–483, 2014.

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

[20] S. Haykin, *Adaptive Filter Theory (3rd Ed.).* Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1996.

[21] Y. Z. Tsypkin, *Adaptation and Learning in Automatic Systems.* Orlando, FL, USA: Academic Press, Inc., 1971.

[22] E. Walach and B. Widrow, "The least mean fourth (lmf) adaptive algorithm and its family," *IEEE Transactions on Information Theory*, vol. 30, pp. 275–283, Mar 1984.

[23] M. Martinez-Ramon, J. Arenas-Garcia, A. Navia-Vazquez, and A. R. Figueiras-Vidal, "An adaptive combination of adaptive filters for plant identification," in *2002 14th International Conference on Digital Signal Processing Proceedings. DSP 2002 (Cat. No.02TH8628)*, vol. 2, pp. 1195–1198 vol.2, 2002.

[24] J. Arenas-Garcia, A. R. Figueiras-Vidal, and A. H. Sayed, "Mean-square performance of a convex combination of two adaptive filters," *IEEE Transactions on Signal Processing*, vol. 54, pp. 1078–1090, March 2006.

[25] N. J. Bershad, J. C. M. Bermudez, and J. Y. Tourneret, "An affine combination of two lms adaptive filters—transient mean-square analysis," *IEEE Transactions on Signal Processing*, vol. 56, pp. 1853–1864, May 2008.

[26] S. S. Kozat, A. T. Erdogan, A. C. Singer, and A. H. Sayed, "Steady-state mse performance analysis of mixture approaches to adaptive filtering," *IEEE Transactions on Signal Processing*, vol. 58, pp. 4050–4063, Aug 2010.

[27] M. T. M. Silva and V. H. Nascimento, "Improving the tracking capability of adaptive filters via convex combination," *IEEE Transactions on Signal Processing*, vol. 56, pp. 3137–3149, July 2008.

[28] V. H. Nascimento, M. T. M. Silva, L. A. Azpicueta-Ruiz, and J. Arenas-Garca, "On the tracking performance of combinations of least mean squares and recursive least squares adaptive filters," in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 3710–3713, March 2010.

[29] F. S. Cattivelli, C. G. Lopes, and A. H. Sayed, "Diffusion recursive least-squares for distributed estimation over adaptive networks," *IEEE Transactions on Signal Processing*, vol. 56, pp. 1865–1877, May 2008.

[30] J. Chen and A. H. Sayed, "On the learning behavior of adaptive networks—part i: Transient analysis," *IEEE Transactions on Information Theory*, vol. 61, pp. 3487–3517, June 2015.

[31] J. Arenas-Garca, M. Martnez-Ramn, ngel Navia-Vzquez, and A. R. Figueiras-Vidal, "Plant identification via adaptive combination of transversal filters," *Signal Processing*, vol. 86, no. 9, pp. 2430 – 2438, 2006. Special Section: Signal Processing in {UWB} Communications.

[32] A. Olshevsky and J. N. Tsitsiklis, "Convergence speed in distributed consensus and averaging," *SIAM J. Control Optim.*, vol. 48, pp. 33–55, Feb. 2009.

[33] J. N. Tsitsiklis, *Problems in decentralized decision making and computation.* PhD thesis, 1984.

[34] J. N. Tsitsiklis, D. P. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," in *1984 American Control Conference*, pp. 484–489, June 1984.

[35] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1989.

[36] M. H. Degroot, "Reaching a consensus," *Journal of the American Statistical Association*, vol. 69, no. 345, pp. 118–121, 1974.

[37] W. Ren, R. W. Beard, and T. W. McLain, *Coordination Variables and Consensus Building in Multiple Vehicle Systems*, pp. 171–188. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005.

[38] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, pp. 215–233, Jan 2007.

[39] J. Cortes, "Analysis and design of distributed algorithms for x-consensus," in *Proceedings of the 45th IEEE Conference on Decision and Control*, pp. 3363–3368, Dec 2006.

[40] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Transactions on Automatic Control*, vol. 48, pp. 988–1001, June 2003.

[41] W. Ren and R. W. Beard, "Consensus seeking in multiagent systems under dynamically changing interaction topologies," *IEEE Transactions on Automatic Control*, vol. 50, pp. 655–661, May 2005.

[42] D. P. Bertsekas and J. N. Tsitsiklis, "Comments on ldquo;coordination of groups of mobile autonomous agents using nearest neighbor rules rdquo;," *IEEE Transactions on Automatic Control*, vol. 52, pp. 968–969, May 2007.

[43] L. Moreau, "Stability of multiagent systems with time-dependent communication links," *IEEE Transactions on Automatic Control*, vol. 50, pp. 169–182, Feb 2005.

[44] V. D. Blondel, J. M. Hendrickx, A. Olshevsky, and J. N. Tsitsiklis, "Convergence in multiagent coordination, consensus, and flocking," in *Proceedings of the 44th IEEE Conference on Decision and Control*, pp. 2996–3000, Dec 2005.

[45] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, pp. 1520–1533, Sept 2004.

[46] A. Nedi and A. Ozdaglar, "Cooperative distributed multi-agent optimization," in *Convex Optimization in Signal Processing and Communications:* (D. P. Palomar and Y. C. Eldar, eds.), pp. 340–386, Cambridge: Cambridge University Press, 12 2009.

[47] S. Kar and J. M. F. Moura, "Convergence rate analysis of distributed gossip (linear parameter) estimation: Fundamental limits and tradeoffs," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, pp. 674–690, Aug 2011.

[48] A. H. Sayed, "Adaptive networks," *Proceedings of the IEEE*, vol. 102, pp. 460–497, April 2014.

[49] S. Hosseini, A. Chapman, and M. Mesbahi, "Online distributed estimation via adaptive sensor networks," *h ttp://rain. aa. washington. edu/@ api/deki/files/324/= TCNS13*, vol. 914, 2014.

[50] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, pp. 48–61, Jan 2009.

[51] A. I. Chen and A. Ozdaglar, "A fast distributed proximal-gradient method," 2012.

[52] J. C. Duchi, A. Agarwal, and M. J. Wainwright, "Dual averaging for distributed optimization: Convergence analysis and network scaling," *IEEE Transactions on Automatic Control*, vol. 57, pp. 592–606, March 2012.

[53] J. Chen and A. H. Sayed, "Diffusion adaptation strategies for distributed optimization and learning over networks," *IEEE Transactions on Signal Processing*, vol. 60, pp. 4289–4305, Aug 2012.

[54] 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, pp. 155–171, May 2013.

[55] S. Y. Tu and A. H. Sayed, "Diffusion strategies outperform consensus strategies for distributed estimation over adaptive networks," *IEEE Transactions on Signal Processing*, vol. 60, pp. 6217–6234, Dec 2012.

[56] D. P. Bertsekas, "A new class of incremental gradient methods for least squares problems," *SIAM J. on Optimization*, vol. 7, pp. 913–926, Apr. 1997.