UNIVERSITY OF MINNESOTA

This is to certify that I have examined this copy of a Master's thesis by

Gonzalo Mateos Buckstein

and have found that it is complete and satisfactory in all respects, and that any and all revisions required by the final examining committee have been made.

Name of Faculty Advisor(s)

Signature of Faculty Advisor(s)

Date

GRADUATE SCHOOL

Distributed Adaptive Estimation and Tracking using Ad Hoc Wireless Sensor Networks

A THESIS

SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL

OF THE UNIVERSITY OF MINNESOTA

BY

Gonzalo Mateos Buckstein

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

Professor Georgios B. Giannakis, Advisor

July 2009

Acknowledgments

First and foremost, my deepest gratitude goes to my advisor Prof. Georgios Giannakis. I would like to thank him for giving me the opportunity to embark on this journey as a graduate student, a real privilege for which I am really honored. His guidance and constant encouragement has made me become not only a better researcher, but also a better person. This thesis would not have been possible without all his insightful suggestions. He generously shared with me his immense knowledge on statistical signal processing, and his contagious passion towards seeking an understanding of what we still cannot explain.

Due thanks go to Profs. Mihailo Jovanovic, Zhi-Quan (Tom) Luo and Stergios Roumeliotis for agreeing to serve on my committee.

The work in this thesis is a result of a very fruitful side-by-side collaboration with Yannis Schizas. He does not only deserve my gratitude for helping me along my first steps as a researcher, but significant credit for his contributions to the work reported here. The material in this thesis benefited from discussions with current and former members of SPiNCOM: Dr. Daniele Angelosante, Juan-Andrés Bazerque, Dr. Alfonso Cano, Yannis Delis, Shahrokh Farahmand, Pedro Forero, Nikos Gatsis, Dr. Vassilis Kekatos, Dr. Seung-Jun Kim, Prof. Antonio Marques, Eric Msechu, Ketan Rajawat, Dr. Tairan Wang, Yuchen Wu, Dr. Yingqun Yu, and Hao Zhu. I am not forgetting Prof. Alejandro Ribeiro and his family: I want to give them special thanks for – among many other things – their hospitality and help upon my arrival to Minneapolis. At this point I would also like to thank my colleagues at the Instituto de Ingeniería Eléctrica, Universidad de la República, who taught me what electrical engineering is all about; and specially Profs. Gregory Randall and Alicia Fernandez that encouraged me to pursue graduate studies abroad.

My family and friends, some of which I've already mentioned above, are the most important part in making my life a wonderful experience. For this reason I wish to thank those here in Minneapolis, and specially all of my long-time friends that are far way, but actually feel as close to me as in the past. Por último, quiero agradecer a Cachito, Mamá, Fede y Agus por la familia que tengo. Esta tesis va dedicada a ustedes.

Gonzalo Mateos, Minneapolis, July 22 2009.

Distributed Adaptive Estimation and Tracking using Ad Hoc Wireless Sensor Networks

Abstract:

Wireless sensor networks (WSNs), whereby large numbers of inexpensive sensors with constrained resources cooperate to achieve a common goal, constitute a promising technology for applications as diverse and crucial as environmental monitoring, healthcare, fault diagnosis in process industry, protection of critical infrastructure including the smart grid, and surveillance systems for homeland security. The advent of WSNs has created renewed interest in the field of distributed computing, calling for collaborative solutions that enable low-cost estimation of stationary signals as well as reduced-complexity tracking of nonstationary processes. In this thesis, distributed estimation and tracking algorithms using ad hoc WSNs are developed, and analyzed in terms of their stability and performance.

Unique features characterizing the operation of WSNs dictate that often times sensors need to perform estimation in a constantly changing environment, and lacking a data model for the underlying processes of interest. This motivates the distributed counterparts to the least mean-square (LMS) and recursive least-squares (RLS) algorithms developed in this thesis, which are based on in-network processing of distributed sensor observations. An iterative process takes place towards consenting on the desired global estimators: sensors perform simple local tasks to refine their current estimates, and exchange messages with one-hop neighbors over noisy communication channels. New sensor data is incorporated in real time to enrich the estimation process and learn the unknown statistics 'on-the-fly'.

Jointly, the novel distributed (D-)LMS and D-RLS algorithms offer the flexibility to manipulate the complexity versus performance tradeoff. Requiring identical communication resources, D-RLS markedly outperforms D-LMS in terms of convergence rate and steady-state error, at the price of increased computational complexity. Mean-square error (MSE) performance analysis is conducted to provide the means of selecting the best such tradeoff. Under simplifying assumptions, closed-form expressions are derived for the networkwide and sensor-level performance figures of merit in steady-state. Mean and MSE-sense stability of both algorithms are also established, and yield sufficient conditions ensuring that a steady-state is reached.

Contents

List of Figures

Chapter 1

Wireless Sensor Networks

Driven by a wide span of foreseen applications, decentralized estimation of signals based on observations acquired by spatially distributed wireless sensors has attracted much attention recently. A wireless sensor is a device capable of sensing the physical environment, performing signal processing tasks and communicating information using a wireless transceiver. A large collection of these sensors is referred to as a wireless sensor network (WSN). In order to make large scale deployments economically feasible, sensors are supposed to be inexpensive devices with constrained resources, limited computational, storage and communication capabilities. Yet, the fundamental premise that many simple entities working in synergy can still achieve greater goals is the key behind the popularity of WSNs. In other words, a unique feature of WSNs is the cooperative effort of sensor nodes. It is the addition of wireless communication abilities to the sensors that enables such cooperation, hence leading to a whole new dimension of possibilities with regards to the environmental, domestic, and military application areas.

Emergent WSN-based applications include distributed field monitoring, localization, surveillance, power spectrum estimation and target tracking just to name a few. These tasks typically require estimating parameters of interest such as temperature, concentration of certain pollutants in the air, and position of a target. Surveillance applications could also include the detection of critical events, e.g., the start of a fire in a warehouse to trigger a remote alarm. Even though it is the distributed structure of a WSN which provides the means of tackling the aforementioned applications, this is simply not enough to achieve the ultimate goals of performing distributed estimation and detection. Specifically, it has been recognized that sensors should be empowered with appropriate signal processing tools, that explicitly take into account the distributed nature of the sensed data and operational constraints of the WSNs. In this dissertation, novel distributed estimation algorithms are developed and analyzed in terms of their stability and performance. They enable low cost estimation of stationary signals as well as reduced-complexity tracking of nonstationary processes.

1.1 Motivation and Context

Formidable challenges arise as emergent WSN-based estimation applications demand promptly available, yet accurate local estimates under increasingly restrictive and unpredictable operational constraints. Such unique WSN features dictate that often times sensors need to perform estimation in a constantly changing environment, without having available a (statistical) model for the underlying processes of interest. This has motivated the development of distributed *adaptive* estimation schemes, generalizing the notion of adaptive filtering to a setup involving networked sensing/processing devices. In this thesis, distributed counterparts to the least mean-square (LMS) and recursive least-squares (RLS) algorithms are developed for decentralized estimation and tracking in WSNs. Different from schemes that operate in a batch centralized mode, the novel D-LMS and D-RLS algorithms allow for online incorporation and processing of sensor data. This enables learning the unknown process statistics 'on-the-fly', or simply tracking them in the typical case of nonstationary WSN environments.

1.1.1 Two Prevailing WSN Topologies

Two dominant WSN topologies have prevailed as the most suitable for distributed estimation and detection tasks. They can be classified based on the presence or absence of a central processing unit or fusion center (FC).

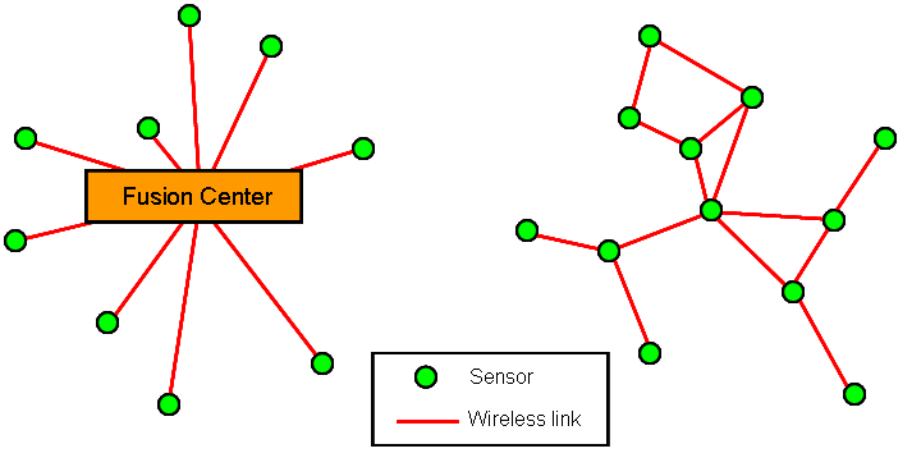

Figure 1.1: (left) An FC-based WSN where many sensors transmit their information to a fusion center via wireless links; (right) An ad hoc topology that is devoid of hierarchies.

When an FC is present, the WSN becomes a two-level hierarchical entity; see Fig 1.1 (left). On the one hand, all deployed sensors are in charge of sensing the environment and then transmitting their acquired data to the central processing unit. It is possible that simple sensor-level processing – such as compression and/or quantization – is performed on the acquired raw data before transmission. Communications take place over the unreliable wireless medium; hence, they are challenged by the effects of shadowing, multipath fading and additive receiver noise. On the other hand, the FC serves the purpose of a data sink collecting the sensor information. Most importantly, it is responsible of performing all necessary signal processing tasks in order to achieve the network's operational objectives. Depending on the application, it may be the case that the FC feeds back the global estimate to the sensors so that they can take specific action. In contrast to the resource-constrained sensors, it is common to assume that an FC is a device with rich processing capabilities and memory capacity.

In the absence of an FC, the WSNs are termed *ad hoc* and are devoid of hierarchies;

see Fig 1.1 (right). It is the network itself, i.e., the collection of sensors, which is responsible for processing the information acquired by the sensors. The so-termed *in-network* processing introduces additional algorithmic challenges, as the information is not centrally available but scattered across a possibly large geographical area under surveillance. Decentralized estimation using ad hoc WSNs is based on successive refinements of local estimates maintained at individual sensors. In a nutshell, each iteration of this broad class of fully distributed algorithms comprises: (i) a communication step where sensors exchange information with their (nearby) neighbors through the shared wireless medium; and (ii) an update step where each sensor uses this information to refine its local estimate. Absence of hierarchy and the purely decentralized nature of in-network processing dictate that local sensor estimates should eventually consent to a global estimate, while fully exploiting spatial correlations to maximize estimation performance. In most cases, consensus can only be attained asymptotically in time. However, a finite number of iterations will suffice to obtain local sensor estimates that are sufficiently accurate for all practical purposes. All in all, different from FC-based topologies sensors actively participate in both the sensing and signal processing tasks, and the network behaves as a self-organized entity in lieu of a centralized controller.

WSN topologies that include an FC usually encounter two limitations. The first one pertains to robustness and WSN survivability, as the FC becomes a critical isolated point of failure. In the eventuality of FC malfunction, the whole WSN operation is interrupted. Secondly, the WSN lifetime does not scale gracefully with the coverage area. Specifically, FC-based operation may challenge communications as the WSN scales over a larger geographic area, since far away sensors will require higher power to reach the FC, thus diminishing their battery lifetime. This problem could be alleviated to some extent by requiring the sensors to relay their data via multi-hop routing, since communications take place between neighbors reachable with reduced power. Nevertheless, the important issue regarding fault tolerance remains. For these reasons, the present thesis deals with distributed estimation algorithms for ad hoc WSNs. In the case of sensor failures, ad hoc WSNs will incur performance loss but still remain operational.

It is usually the case that FC-based topologies benchmark the performance among the class of decentralized estimators that can be implemented using WSNs. This is because all network-wide information is centrally available for processing in a 'single-shot' fashion, whereas in ad hoc WSNs sensor data percolate via single-hop exchanges among neighboring sensors. There is an inherent delay till a given sensor can make use of all the data collected by the WSN. Hence, intermediate local sensor estimates will generally be of lower quality when compared to those formed by the FC. An important goal of this thesis is to develop distributed estimation algorithms for ad hoc WSNs with quantifiable, e.g., mean-square error (MSE) performance; that comes as close as possible to the equivalent solution that involves an FC. Often times throughout the course of the dissertation, the performance of the novel distributed algorithms will be gauged against the appropriate FC-based benchmarking schemes. With a slight abuse of notation, the latter will be referred to as the 'centralized' counterpart since all sensor data – though distributed in nature – is centrally available for processing.

Resource allocation, medium access control and general communication protocols for in-network processing schemes are interesting problems in their own right, but go beyond the scope of the present thesis. Some high-level comments on a feasible protocol to support the proposed algorithms can be found in Remark 2.5 of Chapter 2.

1.1.2 A Motivating Application

A WSN application where the need for linear regression arises, is spectrum estimation for the purpose of environmental monitoring. Suppose sensors comprising a WSN deployed over some area of interest observe a narrowband source to determine its spectral peaks. This information can assist them to disclose hidden periodicities due to a physical phenomenon controlled by e.g., a natural heat or seismic source. The source of interest propagates through multi-path channels and is contaminated with additive noise when sensed at the sensors. The unknown source-sensor channels may introduce deep fades at the frequency band occupied by the source. Thus, having each sensor operating on its own may lead to faulty assessments. The available spatial diversity to effect improved spectral estimates, can only be achieved via sensor collaboration as in the distributed estimation algorithms presented in this dissertation.

Let $\theta(t)$ denote the evolution of the source signal in time, and suppose that $\theta(t)$ can be modeled as an autoregressive (AR) process [55, p. 106]

$$\theta(t) = -\sum_{\tau=1}^{p} \alpha_\tau \theta(t-\tau) + w(t)$$

where p is the order of the AR process, while $\{\alpha_\tau\}$ are the AR coefficients and $w(t)$ denotes white noise. The source propagates to sensor j via a channel modeled as an FIR filter $C_j(z) = \sum_{l=0}^{L_j-1} c_{j,l} z^{-l}$, of unknown order L_j and tap coefficients $\{c_{j,l}\}$ and is contaminated with additive sensing noise $\bar{\epsilon}_j(t)$ to yield the observation

$$x_j(t) = \sum_{l=0}^{L_j-1} c_{j,l}\theta(t-l) + \bar{\epsilon}_j(t).$$

Since $x_j(t)$ is an autoregressive moving average (ARMA) process, it can be written as [55]

$$x_j(t) = -\sum_{\tau=1}^{p} \alpha_\tau x_j(t-\tau) + \sum_{\tau'=1}^{m} \beta_{\tau'} \tilde{\eta}_j(t-\tau'), \quad j \in \mathcal{J} \tag{1.1}$$

where the MA coefficients $\{\beta_{\tau'}\}$ and the variance of the white noise process $\tilde{\eta}_j(t)$ depend on $\{c_{j,l}\}$, $\{\alpha_\tau\}$ and the variance of the noise terms $w(t)$ and $\bar{\epsilon}_j(t)$. For the purpose of determining spectral peaks, the MA term in (1.1) can be treated as observation noise, i.e., $\epsilon_j(t) := \sum_{\tau'=1}^{m} \beta_{\tau'} \tilde{\eta}_j(t-\tau')$. This is very important since sensors do not have to know the source-sensor channel coefficients as well as the noise variances. The spectral content of the source can be estimated provided sensors estimate the coefficients $\{\alpha_\tau\}$. Let $\mathbf{s}_0 := [\alpha_1 \ldots \alpha_p]^T$ be the unknown parameter of interest. From (1.1) the regression vectors are given as $\mathbf{h}_j(t) = [-x_j(t-1) \ldots -x_j(t-p)]^T$, and can be acquired directly from the sensor data $\{x_j(t)\}$ without the need of training/estimation. Distributed spectrum estimation has been considered also in [24] utilizing generalized projection schemes. Assumptions in [24] include ideal any-to-any communications and known source-sensor channels.

As a preview of the results in this thesis, the performance of the distributed algorithms described in the forthcoming chapters is illustrated next, when applied to the aforementioned power spectrum estimation task. For the numerical experiments, an ad hoc WSN

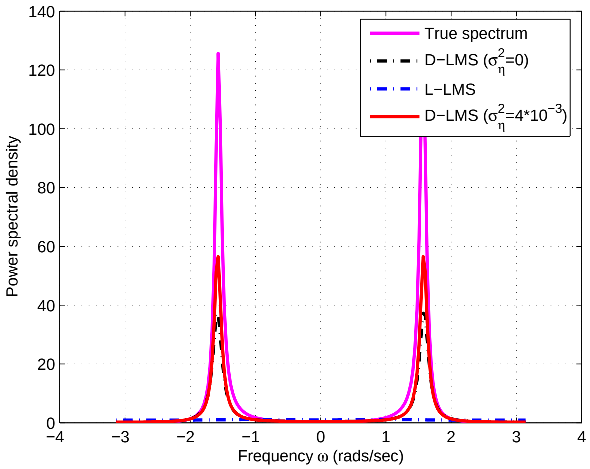

Figure 1.2: D-LMS in a power spectrum estimation task. The true narrowband spectra is compared to the estimated PSD, obtained after the WSN runs the D-LMS and (non-cooperative) L-LMS algorithms. The reconstruction results correspond to a sensor whose multipath channel from the source introduces a null at $\omega = \pi/2 = 1.57$.

with $J = 80$ sensors is simulated as a realization of a random geometric graph; see, e.g., [20]. The source-sensor channels corresponding to a few of the sensors are set so that they have a null at the frequency where the AR source has a peak, namely at $\omega = \pi/2$. Fig. 1.2 depicts the actual power spectral density (PSD) of the source as well as the estimated PSDs for one of the sensors affected by a bad channel. To form the desired estimates in a distributed fashion, the WSN runs local (L-) LMS and the D-LMS algorithm in Chapter 2. The L-LMS is a noncooperative scheme, whereby each sensor, say the jth, independently runs an LMS adaptive filter fed by its local data $\{x_j(t), \mathbf{h}_j(t)\}$ only. In the case of D-LMS, the experiment is performed under ideal and noisy inter-sensor links. Clearly, even in the presence of communication noise D-LMS exploits the spatial diversity available and allows all sensors to estimate accurately the actual spectral peak, whereas L-LMS leads the problematic sensors to misleading estimates.

For the same setup, Fig. 1.3 shows the global learning curve evolution $\text{MSE}(t) = J^{-1}\sum_{j=1}^{J}\|x_j(t) - \mathbf{h}_j^T(t)\mathbf{s}_j(t-1)\|^2$, where $\mathbf{s}_j(t)$ denotes sensor j's estimate of $\mathbf{s}_0$ at time

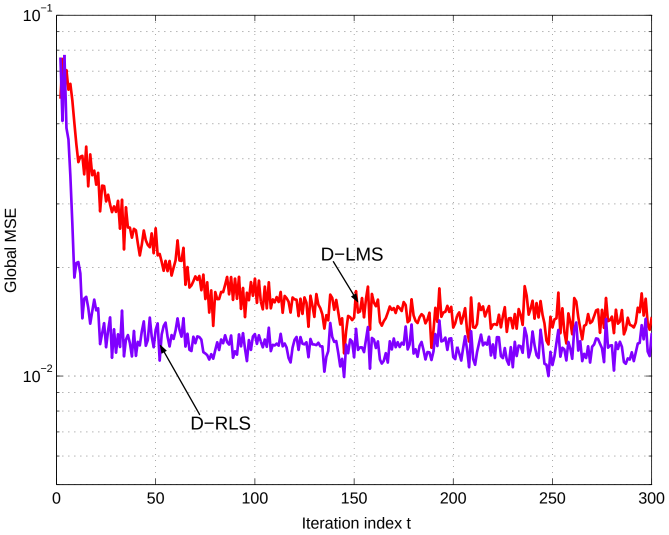

Figure 1.3: Global MSE evolution (network learning curve) for the D-LMS and D-RLS algorithms in a power spectrum estimation task.

instant t. D-LMS and the D-RLS algorithm in Chapter 4 are compared under ideal communication links. It is apparent that D-RLS achieves better performance both in terms of convergence rate and steady state MSE. As discussed in Chapter 4, this comes at the price of increased computational complexity per sensor, while the communication costs incurred are identical.

1.1.3 The Thesis in Context

The advent of WSNs has created renewed interest in the field of distributed computing, calling for collaborative solutions that enable low-cost estimation of stationary signals as well as reduced-complexity tracking of nonstationary processes. Different from WSN topologies that include an FC, ad hoc ones are devoid of hierarchies and rely on in-network processing to effect agreement among sensors on the estimate of interest. A great body of literature has been amassed in recent years, building-up the field of consensus-based distributed signal processing. The uninitiated reader is referred to the tutorial in [36] for general results and a vast list of related works. Achieving consensus across agents was considered in vehicle coordination [23], as well as in distributed sample-averaging of sensor observations [37, 62].

Recently, projecting into a linear subspace (a problem of which sample-averaging is a special case) has been cast in terms of linear distributed iterations [4], and utilized for cooperative spectrum sensing in wireless cognitive radio networks [3]. A general distributed estimation framework was put forth in [45, 49], which does not require the desired estimator to be expressible in closed form in terms of (weighted) sample averages. In the aforementioned schemes sensors acquire data only once and then locally exchange messages to reach consensus.

Extensions for distributed tracking of the sample-average of time-varying signals can be found in e.g., [38, 53]. Sequential in-time incorporation of sensor observations to enrich the estimation process was considered in [63], in the context of linear least-squares parameter estimation. The space-time diffusion algorithm of [63] requires knowledge of the data model and costly exchanges of matrices among neighbors, while the requirement for diminishing step-sizes renders it incapable of tracking time-varying signals. The fundamental problem of distributed state estimation for dynamical systems using ad hoc WSNs has also attracted a lot of attention recently. Distributed Kalman filtering and smoothing approaches have been reported in, e.g., [9, 11, 35, 45, 65], but they are applicable when the state and observation models are known.

In many applications however, sensors need to perform estimation in a constantly changing environment without having available a (statistical) model for the underlying processes of interest. This motivates the development of distributed adaptive estimation schemes, the subject dealt with in the present thesis. The first such approach introduced a sequential scheme, whereby information circulation through a topological cycle in conjunction with LMS-type adaptive filtering per sensor allows the network to account for time variations in the signal statistics [27]. For more general estimators, a similar stochastic incremental gradient descent algorithm was developed in [42], which subsumes [27] as a special case. The incremental LMS schemes in [27, 42] may outperform a centralized implementation of LMS in terms of convergence rate and steady-state error, while entailing a relatively low communication overhead. These features make them appealing, especially for small-size WSNs. However, such schemes inherently require a Hamiltonian cycle through which signal estimates are sequentially circulated from sensor to sensor. In the eventuality of a sensor failure, determination of a new cycle is an NP-hard problem [39], thus challenging the applicability of incremental schemes in medium- to large-size WSNs. Time-critical applications may encounter additional challenges, since the delay for a local estimate update may be significant as the network, hence, the cycle size scales.

Without topological constraints and by fully exploiting the available links in the network, the so-termed (combine-then-adapt) diffusion LMS [28] offers an improved alternative at the price of increasing communication cost. On a per iteration basis, each sensor forms a convex combination of the local estimates acquired from its neighborhood, quantity which is then fed to a local LMS filter to yield the updated estimate. Performance gains result from interchanging the order of the aforementioned steps; i.e., adapt-then-combine [10], leading to the diffusion LMS variant originally proposed in [54]. An alternative to reduce steady-state estimation errors involves diffusing raw sensor observations and regression vectors per neighborhood [10]. This facilitates the flow of new data across the WSN, but can degrade performance in the presence of communication noise and essentially doubles the communication cost. Additional variants of the diffusion LMS algorithm in [28] include the utilization of hierarchical nodes [12] and adaptive combiners [57]. Tailored to applications in which fast convergence is at a premium and increased computational burden per sensor can be afforded, distributed RLS counterparts can be found in [44] and [8].

Several distributed estimation algorithms are rooted on iterative optimization methods, which capitalize upon the separable structure of the cost defining the desired estimator. The sample mean estimator was formulated in [41] as an optimization problem, and was solved in a distributed fashion using a primal dual approach; see, e.g., [7]. Similarly, the incremental schemes in [27,40,42,44] are all based in incremental (sub)gradient methods [6,34]. Even the diffusion LMS algorithm in [28] has been recently shown to have a connection with incremental strategies, when these are applied to optimize an approximate reformulation of the LMS cost [10]. Building on the framework introduced in [45,49], the D-LMS and D-RLS algorithms developed in this thesis are obtained upon recasting the respective decentralized estimation problems as multiple equivalent constrained subproblems. The resulting minimization subtasks are shown to be highly parellizable across sensors, when carried out using the alternating-direction method of multipliers (AD-MoM) [7,17,18]. Similar ideas have been applied in spectrum cartography for cognitive radio networks [5], distributed demodulation [64] and distributed classification [15]. Much related to the AD-MoM is the alternating minimization algorithm (AMA) of [59], which is used in Chapter 4 to derive a reduced complexity D-RLS variant.

Different from [27,42,44], the distributed estimators in this thesis are applicable to general ad hoc WSNs that do not necessarily posses a Hamiltonian cycle, and where inter-sensor communications are challenged by additive noise. The diffusion LMS and RLS algorithms in [8,10] that exchange raw sensor data, as well as the consensus averaging schemes generate local estimates with unbounded variance in the presence of communication noise. Diminishing step-sizes have been proposed to suppress such undesirable effects [21,25], limiting their applicability to stationary environments. The novel D-LMS and D-RLS tracking algorithms utilize constant step-sizes, yet they remain robust to communication noise.

1.2 Thesis Outline and Contributions

A distributed least mean-square (D-LMS) algorithm is developed in Chapter 2, for ad hoc WSN based estimation and tracking applications. It offers simplicity and flexibility whilst solely requiring single-hop communications among sensors. The resultant estimator minimizes a pertinent squared-error cost by resorting to: (i) the AD-MoM so as to gain the desired degree of parallelization; and, (ii) a stochastic approximation iteration to cope with the time-varying statistics of the process under consideration. Information is efficiently percolated across the WSN using a subset of 'bridge' sensors, which further trade-off communication cost for robustness to sensor failures. The resulting in-network processing per sensor is interpreted as a local-LMS adaptation rule superimposed to the output of a tunable proportional-integral (PI) regulator, which drives the local estimate to consensus as dictated by a network-wide information enriched reference.

For a linear data model and under mild assumptions aligned with those considered in the classical (centralized) LMS [43,51,52,60], stability of the novel D-LMS algorithm is established to guarantee that local sensor estimation error norms remain bounded most of the time. Interestingly, this weak stochastic stability result extends to the pragmatic setup where inter-sensor communications are corrupted by additive noise. In the absence of observation and communication noise, consensus is achieved almost surely as local estimates are shown exponentially convergent to the parameter of interest with probability one. Mean-square error (MSE) performance of D-LMS is also assessed, via an approximation that is asymptotically exact as the step-size vanishes. Numerical simulations illustrate that D-LMS outperforms existing alternatives and corroborate the stability and performance analysis results.

Starting from an alternative reformulation of the LMS cost, an improved version of the D-LMS algorithm is developed in Chapter 3 which does not rely on the special type of 'bridge' sensors. As a byproduct, this approach results in a fully distributed algorithm whereby all sensors perform identical tasks, without introducing hierarchies that may require intricate recovery protocols to cope with sensor failures. When error control codes render inter-sensor links virtually noise-free, a third variant of D-LMS incurring lower communication and computational complexity can be developed without sacrificing performance.

An MSE performance analysis of D-LMS tracking is conducted in the presence of a time-varying parameter vector, which adheres to a first-order autoregressive model. For sensor observations that are related to the parameter vector of interest via a linear Gaussian model and after adopting simplifying independence assumptions, exact closed-form expressions are derived for the global and sensor-level MSE evolution as well as its steady-state values. Mean and MSE-sense stability of D-LMS are also established. As in the classical (centralized) LMS algorithm, when tracking slowly time-varying processes there exists an optimum step-size minimizing the limiting MSE. Interestingly, extensive numerical tests demonstrate that for small step-sizes the results accurately extend to the pragmatic setting whereby sensors acquire temporally correlated, not necessarily Gaussian data.

Recursive least-squares (RLS) schemes are of paramount importance for reducing complexity and memory requirements in estimating stationary signals as well as for tracking nonstationary processes, especially when the state and/or data model are not available and fast convergence rates are at a premium. To this end, a fully distributed (D-) RLS algorithm is developed in Chapter 4 for use by WSNs whereby sensors exchange messages with one-hop neighbors to consent on the network-wide estimates adaptively. The WSNs considered here are general in the sense that they do not necessarily possess a Hamiltonian cycle, while the inter-sensor links are challenged by communication noise. The novel algorithm is obtained after judiciously reformulating the exponentially-weighted least-squares cost into a separable form, which is then optimized via the AD-MoM. The exponential weighting effected through a forgetting factor endows D-RLS with tracking capabilities. This is desirable in a constantly changing environment, within which WSNs are envisioned to operate. If powerful error control codes are utilized and communication noise is not an issue, D-RLS is modified to reduce communication overhead when compared to existing noise-unaware alternatives. Numerical simulations demonstrate that D-RLS outperforms existing approaches in terms of estimation performance and noise resilience, while it has the potential of performing efficient tracking.

The second-order D-RLS scheme markedly outperforms the (first-order) stochastic gradient based D-LMS algorithms in Chapters 2 and 3, both in terms of convergence rate and steady-state estimation error. A price is paid, however, in terms of increased computational complexity per sensor, while the communication costs incurred are identical. One of the main goals in this thesis is to introduce several distributed algorithms for in-network adaptive estimation using WSNs, with quantifiable MSE performance and affordable complexity. This, in turn, allows the system designer to select a signal-processing solution according to the best complexity-performance tradeoff, given the system performance specifications and the sensor hardware at hand.

The results in this dissertation have been reported in journal and conference publications [29–32, 46–48]. The D-LMS algorithm with bridge sensors presented in Chapter 2 appeared in [30, 48], while its stability in a stochastic sense was reported in [47]. The fully distributed D-LMS variant in Chapter 3 as well as its tracking performance analysis appeared in [29, 31]. A bridge sensor based precursor to the D-RLS algorithm is introduced in [46], whereas the D-RLS algorithm covered in Chapter 4 can be found in [32].

1.3 Notational conventions

The following notational conventions will be adopted throughout the subsequent chapters. Bold uppercase letters will denote matrices with ij-th entry $[.]_{ij}$, whereas bold lowercase letters will stand for column vectors (i-th entry denoted by $[.]_i$). Whenever the context makes it sufficiently clear, $[.]_{ij}$ will also be used for a matrix to denote block matrix partitioning. Operators $\otimes$, $\circ$, $(.)^T$, $(.)^\dagger$, $\lambda_{\max}(.)$, $\mathrm{tr}(.)$, $\mathrm{diag}(.)$, $\mathrm{bdiag}(.)$, $E\,[.]$, $\mathrm{vec}\,[.]$ will denote Kronecker product, Hadamard product, transposition, matrix pseudo-inverse, spectral radius, matrix trace, diagonal matrix (arguments are scalar diagonal entries), block diagonal matrix (arguments are matrix diagonal entries), expectation, and matrix vectorization, respectively. For both vector and matrices, $\|.\|$ will stand for the $2-$norm and $|.|$ for the cardinality of a set or the magnitude of a scalar. Positive definite matrices will be denoted by $\mathbf{M} \succ \mathbf{0}$. The $n \times n$ identity matrix will be represented by $\mathbf{I}_n$, while $\mathbf{1}_n$ will denote the $n \times 1$ vector of all ones and $\mathbf{1}_{n\times m} := \mathbf{1}_n \mathbf{1}_m^T$. Similar notation will be adopted for vectors (matrices) of all zeros. The i-th vector in the canonical basis for $\mathbb{R}^n$ will be denoted by $\mathbf{b}_{n,i}$, $i = 1, \ldots, n$.

Chapter 2

Distributed LMS for Consensus-Based In-Network Adaptive Processing

2.1 Introduction

In the present chapter we develop a consensus-based distributed (D-)LMS algorithm for in-network adaptive processing using ad hoc WSNs with noisy links. Its simplicity matches well the scarcity of communication and computation resources characterizing WSNs. In contrast with [28], the algorithm is derived from a well-posed estimation criterion optimized using the alternating-direction method of multipliers (AD-MoM) and stochastic approximation techniques. Different from [27, 28] and [42], the novel D-LMS scheme accounts for inter-sensor communication noise, in which setup local estimation errors are shown to be stochastically bounded. In the absence of noise, the local estimates obtained via D-LMS converge exponentially fast to the true parameters. These stability properties, also present in the classical LMS algorithm (see e.g., [51]), are established without invoking the independence and Gaussianity conditions assumed in [27, 28] and [30]. Stochastic averaging arguments are further utilized to approximate the MSE associated with D-LMS. Moreover, D-LMS is shown flexible to trade-off communication cost for robustness to sensor failures by imposing consensus only within a subset of the available sensors. The optimization setup used here to derive the distributed adaptive algorithm resembles the one in [45, 49]. However, as in [27, 28] and [42] D-LMS offers novel and attractive features including: (i) online incorporation and processing of new data across sensors; and (ii) a distributed adaptive estimation scheme for applications where a statistical model of variations is not available (this is needed in e.g., [35, 45]).

In Section 2.2, we introduce the WSN model and the optimization problem defining the desired estimator. Building on [49], we recast the original formulation into an equivalent constrained optimization problem, whose solution becomes available in a distributed fashion using the AD-MoM and stochastic approximation iterations leading to the novel D-LMS (Section 2.3.1). Next, we describe its operation and required communications, and further elaborate on the intuition and flexibility of the resulting algorithm (Section 2.3.2), before demonstrating its merits via numerical simulations in Section 2.3.3. Turning our attention to performance analysis, the challenging problems of stochastic stability and asymptotic MSE characterization are addressed in Sections 2.4.1 and 2.4.2.

2.2 Preliminaries and Problem Statement

Consider an ad hoc WSN comprising J sensors where only single-hop communications are allowed, i.e., sensor j can only communicate with sensors in its neighborhood $\mathcal{N}_j \subseteq \{1, \ldots, J\} := \mathcal{J}$, with the convention $j \in \mathcal{N}_j$. Assuming that inter-sensor links are symmetric, the WSN is modeled as an undirected graph whose vertices are the sensors and its edges represent the available links. Global connectivity information is captured by the symmetric adjacency matrix $\mathbf{E} \in \mathbb{R}^{J \times J}$, where $[\mathbf{E}]_{ij} = 1$ if $i \in \mathcal{N}_j$ and $[\mathbf{E}]_{ij} = 0$ otherwise. This model includes the widely adopted planar random geometric graph $\mathcal{G}^2(J, r)$ [20], where J sensors are randomly placed over the unity square, while connectivity of two nodes is ensured so long as their Euclidean distance is less than a pre-specified communication range r. To ensure that the data from an arbitrary sensor can eventually percolate through the entire network, it is assumed that:

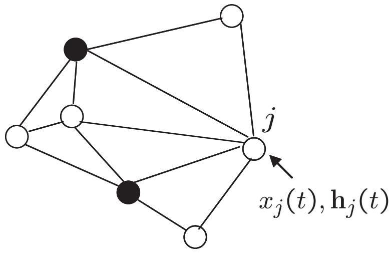

Figure 2.1: An ad hoc WSN with $J = 7$ sensors. The $|\mathcal{B}| = 2$ of them serving as bridge sensors are depicted in black.

(a1) *The WSN graph is connected; i.e., there exists a (possibly) multi-hop communication path connecting any two sensors.*

Different from [27, 28] and [42], the present network model accounts explicitly for non-ideal sensor-to-sensor links, through a zero-mean additive noise vector $\boldsymbol{\eta}_j^i(t)$ with covariance matrix $\mathbf{R}_{\eta_{j,i}} := E[\boldsymbol{\eta}_j^i(t)\boldsymbol{\eta}_j^i(t)^T]$ corrupting signals received at sensor j from sensor i at discrete-time instant t. The noise vectors $\{\boldsymbol{\eta}_j^i(t)\}_{j\in\mathcal{J}}^{i\in\mathcal{N}_j}$ are assumed temporally and spatially uncorrelated. Because the results in this chapter do not depend on the noise pdf, this model incorporates, but is not limited to receiver additive white Gaussian noise (AWGN). A sample ad hoc WSN is depicted in Fig. 2.1.

The WSN is deployed to estimate a signal vector $\mathbf{s}_0 \in \mathbb{R}^{p\times 1}$. Per time instant $t = 0, 1, 2, \ldots$, each sensor has available a regression vector $\mathbf{h}_j(t) \in \mathbb{R}^{p\times 1}$ and acquires a scalar observation $x_j(t)$, both assumed zero-mean without loss of generality. A similar data setting was considered also in [27] and [28]. Introducing the global vector $\mathbf{x}(t) := [x_1(t) \ldots x_J(t)]^T \in \mathbb{R}^{J\times 1}$ and matrix $\mathbf{H}(t) := [\mathbf{h}_1(t) \ldots \mathbf{h}_J(t)]^T \in \mathbb{R}^{J\times p}$, the global LMS estimator of interest can be written as [43, p. 49], [52, p. 14], [27]

$$\hat{\mathbf{s}}(t) = \arg\min_{\mathbf{s}} E\left[\|\mathbf{x}(t) - \mathbf{H}(t)\mathbf{s}\|^2\right] = \arg\min_{\mathbf{s}} \sum_{j=1}^{J} E\left[(x_j(t) - \mathbf{h}_j^T(t)\mathbf{s})^2\right]. \tag{2.1}$$

For jointly stationary $\{\mathbf{x}(t), \mathbf{H}(t)\}$, solving (4.1) leads to the well-known Wiener filter estimate $\hat{\mathbf{s}}_W = \mathbf{R}_H^{-1}\mathbf{r}_{Hx}$, where $\mathbf{R}_H := E[\mathbf{H}^T(t)\mathbf{H}(t)]$ and $\mathbf{r}_{Hx} := E[\mathbf{H}^T(t)\mathbf{x}(t)]$, see e.g., [52, p. 15]. If $\mathbf{R}_H$ and $\mathbf{r}_{Hx}$ were known, then a steepest-descent iteration

$$\hat{\mathbf{s}}(t) = \hat{\mathbf{s}}(t-1) + \mu[\mathbf{r}_{Hx} - \mathbf{R}_H\hat{\mathbf{s}}(t-1)]$$

with sufficiently small step-size μ would converge to $\hat{\mathbf{s}}_W$ while avoiding the burden of inverting $\mathbf{R}_H$. In many linear regression applications involving online processing of data, this covariance information may be either unavailable or time-varying, and thus impossible to update continuously. Targeting low complexity implementations, one often resorts to the centralized (C-) LMS algorithm; see e.g., [52, p. 77]

$$\hat{\mathbf{s}}(t) = \hat{\mathbf{s}}(t-1) + \mu\mathbf{H}^T(t)\left[\mathbf{x}(t) - \mathbf{H}(t)\hat{\mathbf{s}}(t-1)\right] \tag{2.2}$$

which relies on $\mathbf{R}_H \approx \mathbf{H}^T(t)\mathbf{H}(t)$ and $\mathbf{r}_{Hx} \approx \mathbf{H}^T(t)\mathbf{x}(t)$ to coarsely approximate the ensemble averages instantaneously. Considering a constant step-size μ, in order to allow for tracking of a possibly time-varying $\mathbf{s}_0(t)$, the C-LMS algorithm yields stochastic iterates $\hat{\mathbf{s}}(t)$ that do not converge to, but hover around the desired signal of interest. Stability analysis of C-LMS has a long history. General results can be found in [51], [26] and [43] that also include surveys of prior art. One of the main results reported in [51] is that for observations adhering to a linear model, i.e., $\mathbf{x}(t) = \mathbf{H}(t)\mathbf{s}_0 + \boldsymbol{\epsilon}(t)$ with $\mathbf{H}(t)$ assumed stationary ergodic with finite fourth-order moments and $\mathbf{R}_H \succ \mathbf{0}$, recursion (2.2) with sufficiently small step-size μ provably: (i) yields an estimation error whose norm remains most of the time within a finite interval, i.e., $\lim_{\delta\to\infty}\sup_{t\geq 0}\Pr[\|\hat{\mathbf{s}}(t) - \mathbf{s}_0\| \geq \delta] = 0$ even in the presence of noise; and (ii) provides estimates that are almost surely (a.s.) convergent to the true parameter $\mathbf{s}_0$ at an exponential rate in the absence of observation noise. The stability notion described in (i) is referred to as *weak stochastic stability*, and estimation errors are said to be weakly stochastic bounded (WSB) [51]. Note that in (ii) a.s. convergence is with respect to the probability measure induced by the random regressors $\{\mathbf{h}_j(t)\}_{j=1}^J$.

Remark 2.1 (Application to distributed linear regression) An interesting application where the need for linear regression arises is spectrum estimation. Specifically, suppose sensors observe a narrowband source to determine its spectral peaks, which can assist them disclose hidden periodicities due to a physical phenomenon controlled by e.g., a natural heat source. The source of interest propagates through multi-path channels and is contaminated with additive noise when sensed at the sensors. The unknown source-sensor channels may introduce deep fades at the frequency band occupied by the source. Thus, having each sensor operating on its own may lead to faulty assessments. The available spatial diversity to effect improved spectral estimates, can only be achieved via sensor collaboration.

Let $\theta(t)$ denote the narrowband source of interest, which can be modeled as an autoregressive (AR) process [55, p. 106]

$$\theta(t) = -\sum_{\tau=1}^{p} \alpha_\tau \theta(t-\tau) + w(t) \tag{2.3}$$

where p is the order of the AR process, while $\{\alpha_\tau\}$ are the AR coefficients and $w(t)$ denotes white noise. The source propagates to sensor j via a channel modeled as an FIR filter $C_j(z) = \sum_{l=0}^{L_j-1} c_{j,l} z^{-l}$, of unknown order L_j and tap coefficients $\{c_{j,l}\}$ and is contaminated with additive sensing noise $\bar{\epsilon}_j(t)$ to yield the observation

$$x_j(t) = \sum_{l=0}^{L_j-1} c_{j,l} \theta(t-l) + \bar{\epsilon}_j(t). \tag{2.4}$$

Since $x_j(t)$ is an autoregressive moving average (ARMA) process, it can be written as [55]

$$x_j(t) = -\sum_{\tau=1}^{p} \alpha_\tau x_j(t-\tau) + \sum_{\tau'=1}^{m} \beta_{\tau'} \tilde{\eta}_j(t-\tau'), \quad j \in \mathcal{J} \tag{2.5}$$

where the MA coefficients $\{\beta_{\tau'}\}$ and the variance of the white noise process $\tilde{\eta}_j(t)$ depend on $\{c_{j,l}\}$, $\{\alpha_\tau\}$ and the variance of the noise terms $w(t)$ and $\bar{\epsilon}_j(t)$. For the purpose of determining spectral peaks, the MA term in (4.3) can be treated as observation noise, i.e., $\epsilon_j(t) := \sum_{\tau'=1}^{m} \beta_{\tau'} \tilde{\eta}_j(t-\tau')$. This is very important since sensors do not have to know the source-sensor channel coefficients as well as the noise variances. The spectral content of the source can be estimated provided sensors estimate the coefficients $\{\alpha_\tau\}$, so we let $\mathbf{s}_0 := [\alpha_1 \dots \alpha_p]^T$. From (4.3) the regressor vectors are given as $\mathbf{h}_j(t) = [-x_j(t-1) \dots - x_j(t-p)]^T$, directly from the sensor data $\{x_j(t)\}$ without the need of training/estimation. Distributed spectrum estimation has been considered also in [24] utilizing generalized projection schemes. Assumptions in [24] include ideal any-to-any communications and known source-sensor channels.

For different estimation/tracking applications suitable reformulation may be needed in order to acquire linear regressors based on the available information across sensors. For example, in target tracking applications where sensors rely on power or range measurements, the nonlinear data models must be linearized before obtaining regressors as a function of sensor observations; see e.g., [2, p. 137]. Another possibility is to obtain the regression vectors from the physics of the problem, using standard kinematic models with constant velocity or acceleration that are well documented in the tracking literature; see e.g., [2, Ch. 6].

Remark 2.2 (Motivation for in-network processing) Both C-LMS and incremental LMS [27] provide comparable performance benchmarks for distributed LMS-type adaptation rules, as every update encompasses all the information available in the network. Although both the observations $\mathbf{x}(t)$ and regressor rows in $\mathbf{H}(t)$ are actually disseminated across the WSN, in the broad context of sensor network processing one could envision an implementation of the C-LMS using an FC-based topology. This, however, comes at the price of isolating the network's point of failure and may challenge communications as the WSN scales over a larger geographic area, since far away sensors will require higher power to reach the FC, thus diminishing their battery lifetime.

In the context of Remarks 2.1 and 2.2, this chapter aims to develop and analyze in terms of stability and performance, a fully *distributed* (D-) LMS algorithm for in-network adaptive processing using ad hoc WSNs. In a nutshell, the described setup naturally suggests three characteristics that the algorithm should exhibit: (i) stability properties analogous to C-LMS, (ii) processing at the sensor level should be kept as simple as possible; and, (iii) communications among sensors should be confined to single-hop exchanges.

2.3 The D-LMS Algorithm

In this section we introduce the D-LMS algorithm, first going through the algorithm construction and salient features of its operation. The approach followed includes three main building blocks: (i) recast (4.1) into an equivalent form amenable to distributed implementation, (ii) split the optimization problem into smaller and simpler subtasks executed locally at each sensor, and (iii) invoke a stochastic approximation iteration to obtain an adaptive LMS-type of algorithm that can both handle the unavailability/variation of statistical information, and also remain robust to signal variations. We further interpret the resulting D-LMS recursions to gain insights on how local and network-wide information are combined in the learning process, and build intuition on the mechanisms employed to reach consensus among sensors on the adaptive estimate.

To distribute the cost function in (4.1), we replace the global variable $\mathbf{s}$ which couples the per-sensor summands with auxiliary local variables $\{\mathbf{s}_j\}_{j=1}^J$ that represent candidate estimates of $\mathbf{s}$ per sensor. In conjunction with these local variables, consider the convex *constrained* minimization problem

$$\{\hat{\mathbf{s}}_j(t)\}_{j=1}^J = \arg\min_{\{\mathbf{s}_j\}_{j=1}^J} \sum_{i=1}^J E\left[(x_i(t) - \mathbf{h}_i^T(t)\mathbf{s}_i)^2\right], \quad \text{s. t.} \quad \varepsilon_j \mathbf{s}_j = \varepsilon_j \bar{\mathbf{s}}_b,\ b \in \mathcal{B}, j \in \mathcal{N}_b \tag{2.6}$$

where $\mathcal{B} \subseteq \mathcal{J}$ is the *bridge* sensor set introduced in [49], and the additional set of consensus-enforcing variables $\{\bar{\mathbf{s}}_b\}_{b\in\mathcal{B}}$ are maintained at each of the bridge sensors comprising $\mathcal{B}$. Regarding the positive constants ε_j, though they do not cause any effect whatsoever on the constraints in (2.6), they will play an important role in the performance of the D-LMS algorithm (see Remark 2.6). Two simple conditions define a valid set $\mathcal{B}$: (i) for every sensor j there exists at least one bridge sensor $b \in \mathcal{B}$ such that $b \in \mathcal{N}_j$ (the bridge neighbors of sensor j will be denoted by $\mathcal{B}_j := \mathcal{N}_j \cap \mathcal{B}$); and, (ii) for every two bridge sensors b_1 and b_2 there exists a path connecting them which is devoid of edges that link two non-bridge sensors. Multiple sensor assignments will qualify as valid bridge subsets for a given WSN. For instance, the set of all sensors $\mathcal{J}$ is a valid one with maximum cardinality; see also Fig. 2.1 where sensors in black depict $\mathcal{B}$. An upper bound on the number of bridge neighbors per sensor is provided by the maximum connectivity degree in the WSN, namely $D := \max_{j\in\mathcal{J}} |\mathcal{N}_j|$. Note that typically D is much smaller than the total number of sensors J. From a practical viewpoint, $\mathcal{B}$ can be determined and maintained in a distributed fashion using e.g., the simple and efficient polynomial time algorithm in [61].

The WSN connectivity assumption (a1) along with the defining characteristics of $\mathcal{B}$ provide necessary and sufficient conditions to assure that the equality constraints in (2.6) imply $\mathbf{s}_j = \mathbf{s}_{j'}\ \forall\, j, j' \in \mathcal{J}$ [49, Proposition 1]. This establishes the equivalence between (4.1) and (2.6) in the sense that their optimal solutions coincide; i.e., $\hat{\mathbf{s}}_j(t) = \hat{\mathbf{s}}(t) \, \forall \, j \in \mathcal{J}$. Two important structural properties of (2.6) should be appreciated, as they will be instrumental in the development of a distributed algorithm to compute $\{\hat{\mathbf{s}}_j(t)\}_{j=1}^J$: (i) the separable structure of the objective function; and, (ii) the constraints which involve variables of neighboring sensors only.

2.3.1 Algorithm Construction

In order to solve (2.6), we associate Lagrange multipliers $\{\mathbf{v}_j^b\}_{j\in\mathcal{J}}^{b\in\mathcal{B}_j}$ with the corresponding equality constraints and consider the quadratically augmented Lagrangian function given by

$$\begin{aligned}\mathcal{L}_a\left[\boldsymbol{s}, \bar{\mathbf{s}}, \mathbf{v}\right] =& \sum_{j=1}^{J} E\left[(x_j(t+1) - \mathbf{h}_j^T(t+1)\mathbf{s}_j)^2\right] \\ &+ \sum_{b\in\mathcal{B}} \sum_{j\in\mathcal{N}_b} \left[\varepsilon_j \left(\mathbf{v}_j^b\right)^T (\mathbf{s}_j - \bar{\mathbf{s}}_b) + \frac{c_j \varepsilon_j^2}{2} \|\mathbf{s}_j - \bar{\mathbf{s}}_b\|^2\right] \qquad (2.7)\end{aligned}$$

where $\boldsymbol{s} := \{\mathbf{s}_j\}_{j=1}^J$, $\bar{\mathbf{s}} := \{\bar{\mathbf{s}}_b\}_{b\in\mathcal{B}}$, $\mathbf{v} := \{\mathbf{v}_j^b\}_{j\in\mathcal{J}}^{b\in\mathcal{B}_j}$ and $c_j > 0$ are coefficients penalizing the violation of the constraints $\varepsilon_j \mathbf{s}_j = \varepsilon_j \bar{\mathbf{s}}_b \, , \forall \, b \in \mathcal{B}$. The Lagrange multipliers $\{\mathbf{v}_j^b\}_{b\in\mathcal{B}_j}$ are maintained at sensor j. We will now resort to the AD-MoM [7, p. 253] to iteratively minimize (2.7) through a set of simple recursions that update $\{\boldsymbol{s}, \bar{\mathbf{s}}, \mathbf{v}\}$ in a fully distributed fashion. Because the D-LMS algorithm is designed for online estimation, the recursions will run in real-time and hence the iteration index will coincide with the time index t.

The first step consists of locally updating the Lagrange multipliers via dual gradient ascent iterations, as it is customary in the various methods of multipliers [7, Ch. 3]. The pertinent recursions are

$$\mathbf{v}_j^b(t) = \mathbf{v}_j^b(t-1) + \varepsilon_j c_j \left(\mathbf{s}_j(t) - \bar{\mathbf{s}}_b(t)\right), \quad j \in \mathcal{J}, \, b \in \mathcal{B}_j. \qquad (2.8)$$

The second step involves recursions of the local estimates $\boldsymbol{s}$ obtained by minimizing (2.7) using block coordinate descent, i.e., $\mathcal{L}_a\left[\boldsymbol{s}, \bar{\mathbf{s}}, \mathbf{v}\right]$ is minimized with regards to $\boldsymbol{s}$ assuming all other variables $\bar{\mathbf{s}}(t) := \{\bar{\mathbf{s}}_b(t)\}_{b\in\mathcal{B}}$, and $\mathbf{v}(t) := \{\mathbf{v}_j^b(t)\}_{j\in\mathcal{J}}^{b\in\mathcal{B}_j}$ from (2.8) are fixed. The separable structure of (2.6) is inherited by the augmented Lagrangian, and therefore

$$\boldsymbol{s}(t+1) = \arg\min_{\boldsymbol{s}} \mathcal{L}_a[\boldsymbol{s}, \bar{\mathbf{s}}(t), \mathbf{v}(t)]$$

decouples into J simpler minimization sub-problems:

$$\mathbf{s}_j(t+1) = \arg\min_{\mathbf{s}_j} \Bigg(E\left[(x_j(t+1) - \mathbf{h}_j^T(t+1)\mathbf{s}_j)^2\right] + \sum_{b\in\mathcal{B}_j}\left[\varepsilon_j \left(\mathbf{v}_j^b(t)\right)^T \mathbf{s}_j + \frac{c_j\varepsilon_j^2}{2}\|\mathbf{s}_j - \bar{\mathbf{s}}_b(t)\|^2\right]\Bigg). \quad (2.9)$$

Since the cost in (2.9) is convex and differentiable, the first-order necessary condition is also sufficient for optimality. Computing the gradient with respect to $\mathbf{s}_j$ and setting the result equal to zero, yields

$$E\left[-2\mathbf{h}_j(t+1)\left(x_j(t+1) - \mathbf{h}_j^T(t+1)\mathbf{s}_j\right) + \sum_{b\in\mathcal{B}_j}\varepsilon_j\mathbf{v}_j^b(t) + \sum_{b\in\mathcal{B}_j}\varepsilon_j^2 c_j\left(\mathbf{s}_j - \bar{\mathbf{s}}_b(t)\right)\right] = \mathbf{0}. \quad (2.10)$$

Thus, the local estimate update $\mathbf{s}_j(t+1)$ can be obtained as the root of an equation of the form $\mathbf{f}(\mathbf{s}_j) := E[\boldsymbol{\varphi}(\mathbf{s}_j, x_j(t+1), \mathbf{h}_j(t+1))] = \mathbf{0}$, where $\boldsymbol{\varphi}(\cdot)$ stands for the function inside the expectation in (2.10). In lieu of local (cross-) covariance information, namely $\mathbf{r}_{h_j x_j} := E[\mathbf{h}_j(t+1)x_j(t+1)]$ and $\mathbf{R}_{h_j} := E[\mathbf{h}_j(t+1)\mathbf{h}_j^T(t+1)]$, the root of $\mathbf{f}(\mathbf{s}_j) = \mathbf{0}$ is not computable in closed form since the function $\mathbf{f}(\mathbf{s}_j)$ is unknown. Hence, motivated by stochastic approximation techniques (such as the Robbins-Monro algorithm [26, Ch. 1]) which find the root of an unknown function $\mathbf{f}(\mathbf{s}_j)$ given a time-series of noisy observations $\{\boldsymbol{\varphi}(\mathbf{s}_j, x_j(t+1), \mathbf{h}_j(t+1))\}_{t=0}^{\infty}$, the proposed recursion for all $j = 1, \ldots, J$ is

$$\mathbf{s}_j(t+1) = \mathbf{s}_j(t) + \mu_j\left[2\mathbf{h}_j(t+1)e_j(t+1) - \varepsilon_j^2 c_j|\mathcal{B}_j|\mathbf{s}_j(t) - \sum_{b\in\mathcal{B}_j}\left(\varepsilon_j\mathbf{v}_j^b(t) - \varepsilon_j^2 c_j\bar{\mathbf{s}}_b(t)\right)\right] \quad (2.11)$$

where μ_j is a constant step-size and $e_j(t+1) := x_j(t+1) - \mathbf{h}_j^T(t+1)\mathbf{s}_j(t)$ is the local *a priori* error.

The final step entails updating the consensus-imposing variables $\bar{\mathbf{s}}_b$ kept at the bridge sensors. The corresponding recursions are obtained by minimizing (2.7) with $\boldsymbol{s}(t+1) :=$

$\{\mathbf{s}_j(t+1)\}_{j\in\mathcal{J}}$ and $\mathbf{v}(t) := \{\mathbf{v}_j^b(t)\}_{j\in\mathcal{J}}^{b\in\mathcal{B}_j}$ fixed. The separability of the Lagrangian is crucial again as the general problem

$$\bar{\mathbf{s}}(t+1) = \arg\min_{\bar{\mathbf{s}}} \mathcal{L}_a[\boldsymbol{s}(t+1), \bar{\mathbf{s}}, \mathbf{v}(t)]$$

separates into $|\mathcal{B}|$ convex and differentiable equivalent sub-problems:

$$\bar{\mathbf{s}}_b(t+1) = \arg\min_{\bar{\mathbf{s}}_b} \sum_{j\in\mathcal{N}_b} \left[-\varepsilon_j \left(\mathbf{v}_j^b(t)\right)^T \bar{\mathbf{s}}_b + \frac{c_j\varepsilon_j^2}{2}\|\mathbf{s}_j(t+1) - \bar{\mathbf{s}}_b\|^2 \right]. \tag{2.12}$$

It should be noted that the expectation term in (2.7) has been discarded in the process of obtaining (2.12), since it is not dependent on $\bar{\mathbf{s}}_b$ and thus inconsequential for the minimization. Applying the first-order optimality condition explicitly yields

$$\bar{\mathbf{s}}_b(t+1) = \left(\sum_{r\in\mathcal{N}_b} \varepsilon_r^2 c_r\right)^{-1} \sum_{j\in\mathcal{N}_b} \left(\varepsilon_j \mathbf{v}_j^b(t) + \varepsilon_j^2 c_j \mathbf{s}_j(t+1)\right), \quad b \in \mathcal{B}. \tag{2.13}$$

Recursions (2.8), (2.11) and (2.13) constitute the D-LMS algorithm, which can be arbitrarily initialized. At the beginning of the t-th iteration, sensor j receives the consensus variables $\bar{\mathbf{s}}_b(t)$ from its bridge neighbors $b \in \mathcal{B}_j$. With this information and using (2.8), it is able to update its Lagrange multipliers $\{\mathbf{v}_j^b(t)\}_{b\in\mathcal{B}_j}$ which are then jointly used along with the newly acquired local data $\{x_j(t+1), \mathbf{h}_j(t+1)\}$ to compute $\mathbf{s}_j(t+1)$ via (2.11). Then sensor j transmits the vector $(\varepsilon_j c_j)^{-1}\mathbf{v}_j^b(t) + \mathbf{s}_j(t+1)$ to all bridge sensors in its neighborhood $\mathcal{B}_j$. Subsequently, each sensor $b \in \mathcal{B}$ receives the vectors $(\varepsilon_j c_j)^{-1}\mathbf{v}_j^b(t) + \mathbf{s}_j(t+1)$ and scales them with $\varepsilon_j^2 c_j$ in order to find the weighted average in (2.13) and obtain $\bar{\mathbf{s}}_b(t+1)$, thus completing the $t-$th iteration. Further, observe that in order to compute the weights in (2.13), bridge sensor b should acquire $\{\varepsilon_j^2 c_j\}_{j\in\mathcal{N}_b}$ only from its neighbors during the start-up phase of the WSN.

Communications take place among single-hop neighboring sensors only, at a resulting cost that scales linearly in p, the dimensionality of $\mathbf{s}_0$. Incorporating also the effects of additive communication noise, Fig. 2.2 depicts the vector exchanges required by D-LMS on a per iteration basis, and explicitly shows the additional tasks performed by the sensors in $\mathcal{B}$. The modified D-LMS recursions accounting for the noise corrupted variables exchanged

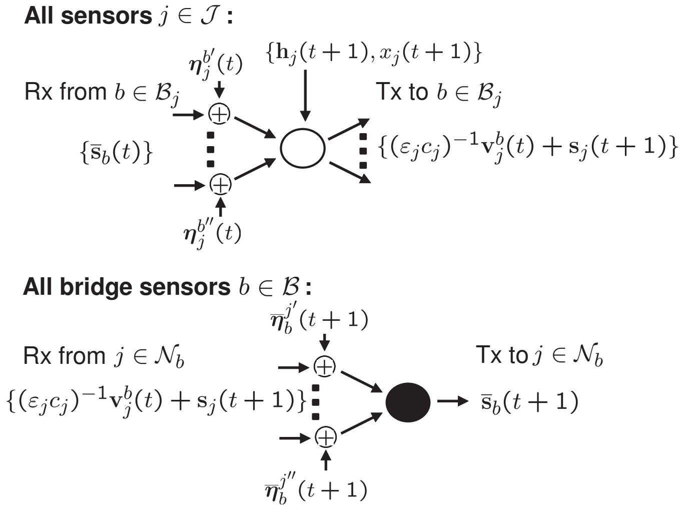
All sensors $j \in \mathcal{J}$:
$\boldsymbol{\eta}_j^{b'}(t)$
$\{\mathbf{h}_j(t+1), x_j(t+1)\}$
Rx from $b \in \mathcal{B}_j$
Tx to $b \in \mathcal{B}_j$
$\{\bar{\mathbf{s}}_b(t)\}$
$\{(\varepsilon_j c_j)^{-1}\mathbf{v}_j^b(t) + \mathbf{s}_j(t+1)\}$
$\boldsymbol{\eta}_j^{b''}(t)$
All bridge sensors $b \in \mathcal{B}$:
$\bar{\boldsymbol{\eta}}_b^{j'}(t+1)$
Rx from $j \in \mathcal{N}_b$
Tx to $j \in \mathcal{N}_b$
$\{(\varepsilon_j c_j)^{-1}\mathbf{v}_j^b(t) + \mathbf{s}_j(t+1)\}$
$\bar{\mathbf{s}}_b(t+1)$
$\bar{\boldsymbol{\eta}}_b^{j''}(t+1)$

Figure 2.2: D-LMS communications over noisy links.

among sensors are summarized below, and tabulated as Algorithm 1. For all sensors $j \in \mathcal{J}$ and $b \in \mathcal{B}$, the D-LMS algorithm in the noisy setup becomes

$$\mathbf{v}_j^b(t) = \mathbf{v}_j^b(t-1) + \varepsilon_j c_j \left(\mathbf{s}_j(t) - (\bar{\mathbf{s}}_b(t) + \boldsymbol{\eta}_j^b(t))\right), \quad b \in \mathcal{B}_j \tag{2.14}$$

$$\mathbf{s}_j(t+1) = \mathbf{s}_j(t) + \mu_j \Bigg[2\mathbf{h}_j(t+1)e_j(t+1) - \varepsilon_j^2 c_j |\mathcal{B}_j| \mathbf{s}_j(t) - \sum_{b \in \mathcal{B}_j} \left(\varepsilon_j \mathbf{v}_j^b(t) - \varepsilon_j^2 c_j (\bar{\mathbf{s}}_b(t) + \boldsymbol{\eta}_j^b(t))\right) \Bigg] \tag{2.15}$$

$$\bar{\mathbf{s}}_b(t+1) = \left(\sum_{r \in \mathcal{N}_b} \varepsilon_r^2 c_r\right)^{-1} \sum_{j \in \mathcal{N}_b} \left(\varepsilon_j \mathbf{v}_j^b(t) + \varepsilon_j^2 c_j (\mathbf{s}_j(t+1) + \bar{\boldsymbol{\eta}}_b^j(t+1))\right). \tag{2.16}$$

D-LMS entails $|\mathcal{B}_j|$ more recursions per sensor when compared to diffusion LMS in [28]. However, since $|\mathcal{B}_j|$ is typically much smaller than J the increase in computational complexity is relatively low. On the other hand, this additional cost and introduced hierarchy among sensors pays off with improved convergence rates as will become apparent in the numerical examples of Section 2.3.3.

Algorithm 1 : D-LMS

```
Arbitrarily initialize $\{\mathbf{s}_j(0)\}_{j=1}^J$, $\{\bar{\mathbf{s}}_b(0)\}_{b\in\mathcal{B}}$ and $\{\mathbf{v}_j^b(-1)\}_{j\in\mathcal{J}}^{b\in\mathcal{B}_j}$.
for $t = 0, 1, \ldots$ do
    Bridge sensors $b \in \mathcal{B}$: transmit $\bar{\mathbf{s}}_b(t)$ to neighbors in $\mathcal{N}_b$.
    All $j \in \mathcal{J}$: update $\{\mathbf{v}_j^b(t)\}_{b\in\mathcal{B}_j}$ using (2.14).
    All $j \in \mathcal{J}$: update $\mathbf{s}_j(t+1)$ using (2.15).
    All $j \in \mathcal{J}$: transmit $(\varepsilon_j c_j)^{-1}\mathbf{v}_j^b(t) + \mathbf{s}_j(t+1)$ to each $b \in \mathcal{B}_j$.
    Bridge sensors $b \in \mathcal{B}$: compute $\bar{\mathbf{s}}_b(t+1)$ using (2.16).
end for
```

Remark 2.3 (Comparison with [45, 49]) In contrast to the D-MLE, D-BLUE and D-LMMSE schemes in [49] and [45], it is apparent that D-LMS in (2.14)-(2.16) allows online incorporation and processing of sensor data. It is an adaptive estimation/tracking algorithm, whereas the distributed estimation schemes in [45, 49] operate in batch mode. Further, note that in D-LMS the requirement for statistical information is bypassed in the stochastic approximation step where the process statistics are learnt 'on-the-fly'. This is not the case in [45, 49] where all proposed schemes are applicable as long as data models are available across sensors.

Remark 2.4 (Versatility through the use of bridge sensors) The bridge sensor set provides flexibility to trade-off communication cost for robustness to sensor failures. In D-LMS each sensor transmits $|\mathcal{B}_j|p$ scalars whereas in diffusion LMS each sensor transmits p scalars. However, note that in D-LMS the non-bridge sensors have to communicate (transmit and receive) with approximately half of their neighbors, namely those in $\mathcal{B}_j$. This follows since the two conditions defining the set $\mathcal{B}$ are satisfied when $|\mathcal{B}| \approx J/2$. Intuitively, this holds because if a sensor is designated to serve as a bridge then its neighbors do not have to be in $\mathcal{B}$, whereas if a sensor does not have bridge neighbors then it turns itself into one; see also the numerical tests in [61]. With reference to Fig. 2.2, a non-bridge sensor j has to remain active over $2|\mathcal{B}_j|$ time slots in order to send and receive information from $b \in \mathcal{B}_j$. Among these slots, $|\mathcal{B}_j|$ are required to transmit $(\varepsilon_j c_j)^{-1}\mathbf{v}_j^b(t) + \mathbf{s}_j(t+1)$ to its bridge neighbors, while the remaining $|\mathcal{B}_j|$ to receive $\bar{\mathbf{s}}_b(t)$ from them. Bridge sensor b

remains active over $2(|\mathcal{B}_b|-1)$ time slots, as any other sensor, plus $|\mathcal{N}_b|$ additional time slots: one to transmit $\bar{\mathbf{s}}_b(t)$ to its neighbors in $\mathcal{N}_b$, and the rest to receive $(\varepsilon_j c_j)^{-1}\mathbf{v}_j^b(t)+\mathbf{s}_j(t+1)$ from them. Assuming equal battery capacities across sensors, bridge sensors are expected to fail first. In diffusion LMS, each sensor has to be active over $|\mathcal{N}_j|$ time slots among which one slot is spent for transmission and the rest for listening. Now, consider the WSN in Fig. 2.1 where bridge sensors are disconnected. Thus, $|\mathcal{B}_b| = 1$ and the communication cost of a bridge sensor is also $|\mathcal{N}_b|$. Because typically $2|\mathcal{B}_j| \leq |\mathcal{N}_j|$, when diffusion LMS is applied to the same WSN the total number of required active time slots will be larger (31 versus 26 in this example). Thus, utilization of bridge sensors offers the potential of increasing the life expectancy of the network.

Regarding recovery from sensor failures, D-LMS remains operational so long as each sensor adjusts its local recursions (2.14)-(2.16) to the modified neighborhood structure, and the overall network graph remains connected. In a possible bridge sensor failure, it might be the case that some sensors need to be promoted to $\mathcal{B}$ using the algorithm in e.g., [61]. Thus, the network as an autonomous entity is capable of adapting to changes in the topology. The steps of the simple recovery process are given in [45, Remark 2].

Remark 2.5 (Consensus and communication protocols) Similar to all consensus-based schemes, D-LMS requires an underlying communication protocol that controls information exchanges among sensors. One feasible choice (not necessarily the most efficient) could be a time division multiple-access (TDMA) system, where each sensor is allocated a time slot during which it can transmit data to its neighbors that operate in reception mode. Consider a TDMA system with $J+|\mathcal{B}|$ time slots. During the first $|\mathcal{B}|$ slots each bridge sensor b transmits to all its neighbors its consensus variable $\bar{\mathbf{s}}_b(t)$ required in (2.14) and (2.15). Each of the p scalars in $\bar{\mathbf{s}}_b(t)$ can be transmitted using e.g., multi-carrier modulation. Then, during the $(|\mathcal{B}|+j)$th time slot only sensor j is active and broadcasts $(\varepsilon_j c_j)^{-1}\mathbf{v}_j^b(t)+\mathbf{s}_j(t+1)$ to each of its bridge neighbors. Recalling that $|\mathcal{B}_j| \ll J$, sensor j could either use different frequency bands to transmit information to each of its bridge neighbors, or, could devote $(1/|\mathcal{B}_j|)$th fraction of the time slot for each of its $|\mathcal{B}_j|$ bridge neighbors. Such a scheme requires: (i) unique sensor indexing established prior to the WSN deployment; and (ii) global synchronization across the WSN, for which there are available algorithms in the existing literature e.g., see [56].

2.3.2 Consensus Controller Interpretation

Even though recursions (2.14)-(2.16) clearly suggest simplicity as an asset of the proposed algorithm, they may somehow obscure the essential mechanisms operating on the available information to yield the estimates $\mathbf{s}_j$. Here we derive a set of equivalent recursions which turn out to be insightful about these issues, despite being less appropriate for online implementation than (2.14)-(2.16).

For arbitrary $j \in \mathcal{J}$ and $b \in \mathcal{B}_j$, consider the noise-free Lagrange multiplier update recursion (2.8) with initial condition $\mathbf{v}_j^b(-1) = \mathbf{0}$. By recognizing $\mathbf{v}_j^b(t)$ as the output of an accumulator system whose input $\varepsilon_j c_j \left[\mathbf{s}_j(t) - \bar{\mathbf{s}}_b(t)\right]$ is the sequence of scaled constraint violations, the zero initial condition yields the equivalent non-recursive form [cf. (2.8)]

$$\mathbf{v}_j^b(t) = \sum_{n=0}^{t} \varepsilon_j c_j \left[\mathbf{s}_j(n) - \bar{\mathbf{s}}_b(n)\right]. \tag{2.17}$$

Arguing by induction as in [49, Lemma 3], the consensus variables for all $b \in \mathcal{B}$ and $t \geq 0$ can be expressed as [cf. (2.13)]

$$\bar{\mathbf{s}}_b(t) = \left(\sum_{r \in \mathcal{N}_b} \varepsilon_r^2 c_r\right)^{-1} \sum_{j \in \mathcal{N}_b} \varepsilon_j^2 c_j \mathbf{s}_j(t). \tag{2.18}$$

Equation (2.18) establishes that the consensus variables $\bar{\mathbf{s}}_b(t)$ are simply obtained as a weighted average of the local estimates gathered from sensor b's neighborhood.

Consider now the vector $\mathbf{q}_j(t) := \mathbf{s}_j(t) - |\mathcal{B}_j|^{-1} \sum_{b \in \mathcal{B}_j} \bar{\mathbf{s}}_b(t)$, which represents the *instantaneous consensus error* at sensor j, as measured with respect to the *consensus reference* given by the average $|\mathcal{B}_j|^{-1} \sum_{b \in \mathcal{B}_j} \bar{\mathbf{s}}_b(t)$. Setting the penalty coefficients as $c_j = 1/|\mathcal{B}_j|$ and using (2.17) to eliminate the Lagrange multipliers from (2.11) yields

$$\mathbf{s}_j(t+1) = \mathbf{s}_j(t) + \mu_j 2\mathbf{h}_j(t+1) e_j(t+1) - \mu_j \varepsilon_j^2 \mathbf{q}_j(t) - \mu_j \varepsilon_j^2 \textstyle\sum_{n=0}^{t} \mathbf{q}_j(n). \tag{2.19}$$

Equations (2.18) and (2.19) are equivalent to D-LMS under ideal links, when $\mathbf{v}_j^b(-1) = \mathbf{0}$. As they stand, the new recursions are not suitable for real-time implementation because the

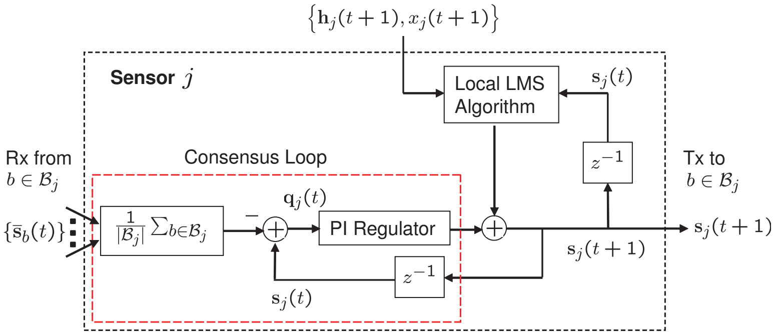
$\{\mathbf{h}_j(t+1), x_j(t+1)\}$
Sensor j
Local LMS Algorithm
$\mathbf{s}_j(t)$
z^{-1}
Rx from $b \in \mathcal{B}_j$
Consensus Loop
Tx to $b \in \mathcal{B}_j$
$\{\bar{\mathbf{s}}_b(t)\}$
$\frac{1}{|\mathcal{B}_j|}\sum_{b\in\mathcal{B}_j}$
$-$
$\mathbf{q}_j(t)$
PI Regulator
$\mathbf{s}_j(t+1)$
$\mathbf{s}_j(t+1)$
z^{-1}
$\mathbf{s}_j(t)$

Figure 2.3: D-LMS sensor processing to obtain local estimates $\mathbf{s}_j$.

sum term in (2.19) requires storing the entire history of $\mathbf{q}_j(t)$. Nonetheless, they shed light into the signal processing taking place at each sensor, which turns out to be remarkably intuitive as discussed next.

The right hand side of (2.19) readily suggests that the local estimate $\mathbf{s}_j(t+1)$ is obtained as the superposition of three terms: (a) the sum $\mathbf{s}_j(t)+\mu_j 2\mathbf{h}_j(t+1)e_j(t+1)$ represents a *local* LMS adaptation based on the new information $\{\mathbf{h}_j(t+1), x_j(t+1)\}$ available at sensor j; (b) an update based on a proportional correction $\mu_j\varepsilon_j^2\mathbf{q}_j(t)$ due to the instantaneous consensus error $\mathbf{q}_j(t)$; and, (c) a correction sum due to the accumulated consensus error (discrete-time integral). A term like (a) is expected, whereas the rest should explain the mechanisms employed to incorporate the extra information gathered from the whole WSN. In fact, (b) and (c) show that a proportional-integral (PI) discrete-time controller, see e.g. [16, p. 605], is used to drive the local estimate $\mathbf{s}_j(t)$ to consensus, as dictated by the computed time-varying set-point $|\mathcal{B}_j|^{-1}\sum_{b\in\mathcal{B}_j}\bar{\mathbf{s}}_b(t)$; see also Fig. 2.3. It is exclusively throughout this reference that global information is percolated to improve the local estimate $\mathbf{s}_j$.

The first closed-loop system interpretation for consensus schemes was given in [23]. Let $\tilde{\mathbf{q}}_j(t) := |\mathcal{B}_j|\mathbf{q}_j(t) = \sum_{b\in\mathcal{B}_j}[\mathbf{s}_j(t) - \bar{\mathbf{s}}_b(t)]$ and eliminate $\bar{\mathbf{s}}_b(t)$ using (2.18). This leads to the global representation $\tilde{\mathbf{q}}(t) = \mathbf{A}\mathbf{s}(t)$ with $\tilde{\mathbf{q}}(t) := [\tilde{\mathbf{q}}_1^T(t)\ldots\tilde{\mathbf{q}}_J^T(t)]^T$, $\mathbf{s}(t) := [\mathbf{s}_1^T(t)\ldots\mathbf{s}_J^T(t)]^T$, and the generalized "two-hop range" Laplacian is given by

$$\mathbf{A} := \text{bdiag}(|\mathcal{B}_1|\mathbf{I}_p,\ldots,|\mathcal{B}_J|\mathbf{I}_p) - \sum_{b\in\mathcal{B}}\frac{(\mathbf{e}_b\otimes\mathbf{I}_p)(\mathbf{e}_b\otimes\mathbf{I}_p)^T}{\sum_{r\in\mathcal{N}_b}\varepsilon_r^2 c_r}\text{bdiag}\left(\varepsilon_1^2 c_1\mathbf{I}_p,\ldots,\varepsilon_J^2 c_J\mathbf{I}_p\right) \quad (2.20)$$

where $\mathbf{e}_b$ represents the b-th column of the adjacency matrix $\mathbf{E}$. Indeed, under (a1) $\mathbf{A}$ shares fundamental properties with an undirected graph Laplacian, i.e., it is symmetric positive semi-definite and its null-space is the consensus subspace (vectors with equal entries). The network-wide feedback $-\mu\mathbf{A}\mathbf{s}(t)$ [cf. (2.19) and Fig. 2.3] links D-LMS with the standard Laplacian-based consensus protocol studied in e.g., [23, 37], whereas the difference stems from the use of bridge sensors. This may not be surprising if one recalls that the Laplacian-based protocol for undirected graphs can be derived using an iterative procedure minimizing a suitable disagreement potential [37, eq. (5)]. The latter strongly resembles the quadratic term augmenting the Lagrangian in (2.7). The extended two-hop information range enjoyed by D-LMS should be also contrasted with diffusion LMS [28], which only spans the single-hop neighborhood. Though, as clarified in Remark 2.4 a small price is paid in terms of the amount of data needed to be transmitted from each sensor.

Remark 2.6 (Consensus loop tuning) The constant ε_j is only affecting the PI gains of the consensus regulator [cf. (2.19)]. For $\varepsilon_j = 1$ these gains boil down to μ_j, a generally small constant attenuating the influence that the information embedded in (b) and (c) has on the estimate $\mathbf{s}_j(t+1)$. The presence of ε_j is thus intuitively justified as a compensator for this effect, gaining an additional degree of freedom to attain potentially faster convergence and/or better estimation performance. Indeed, our simulation results in [30] corroborate considerable improvements when selecting $\varepsilon_j^2\mu_j \approx 1$. For a given step-size and contrasting with $\varepsilon_j = 1$, the steady-state estimation error is markedly reduced at a modest price slightly decreasing the convergence rate of the MSE cost in (4.1). On the other hand, if the D-LMS step-size is increased to the point that there is no gain in estimation error, then the MSE reaches steady-state much faster without a noticeable misadjustment. Based on a suitable performance criterion, a problem falling outside the scope of this thesis, it would be interesting to optimally design the ε_j coefficients; see [62] for a related weight optimization approach in the context of consensus averaging problems.

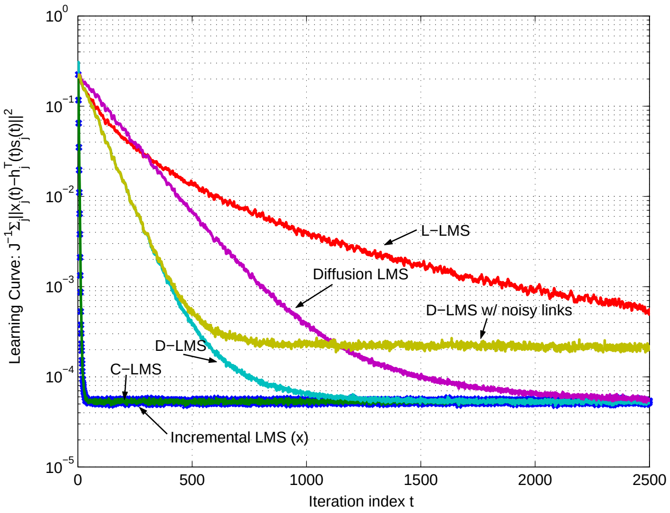

Figure 2.4: Normalized global MSE (learning curve).

2.3.3 Numerical Examples

Here we test the novel D-LMS algorithm, and compare its global MSE performance with: (i) diffusion LMS using Metropolis weights [28]; (ii) local (L-) LMS whereby each sensor runs an independent LMS filter using its local information only (no communications); (iii) centralized incremental LMS [27]; and (iv) C-LMS [cf. (2.2)]. The WSN is simulated as a $\mathcal{G}^2(80, 0.6)$ graph, and for the examples with noisy links receiver AWGN with variance $\sigma_\eta^2 = 10^{-3}$ is added. The signal vector $\mathbf{s}_0 = \mathbf{1}_p$ has dimensionality $p = 8$, and for all $j \in \mathcal{J}$ the regressors $\mathbf{h}_j(t) = [h_j(t) \ldots h_j(t-p+1)]^T$ have entries which evolve according to $h_j(t) = (1-\rho)u_{1,j}h_j(t-1) + \sqrt{\rho}\nu_j(t)$. We choose $\rho = 2 \times 10^{-1}$, the $u_{1,j} \sim \mathcal{U}[0,1]$ (uniformly distributed) are i.i.d. in space, and the driving white noise $\nu_j(t) \sim \mathcal{U}[-\sqrt{3}\sigma_{\nu_j}, \sqrt{3}\sigma_{\nu_j}]$ has a spatial variance profile given by $\sigma_{\nu_j}^2 = 10^{-1}u_{2,j}$ with $u_{2,j} \sim \mathcal{U}[0,1]$ and i.i.d. A linear model $\mathbf{x}(t) = \mathbf{H}(t)\mathbf{s}_0 + \boldsymbol{\epsilon}(t)$ is adopted with observation WGN of spatial variance profile $\sigma_{\epsilon_j}^2 = 10^{-4}u_{3,j}$, with i.i.d. $u_{3,j} \sim \mathcal{U}[0,1]$. For all four algorithms the step-size is set to $\mu = 9 \times 10^{-2}$, and in particular for D-LMS $c_j = \varepsilon_j = 1\,\forall\, j \in \mathcal{J}$.

Fig. 2.4 compares the normalized MSE evolution (learning curve) obtained as $J^{-1}\sum_{j=1}^{J} E[e_j(t)^2]$ for the distributed schemes, where the expectation is approximated by averaging 50 Monte Carlo realizations. Both incremental LMS and C-LMS provide a com-

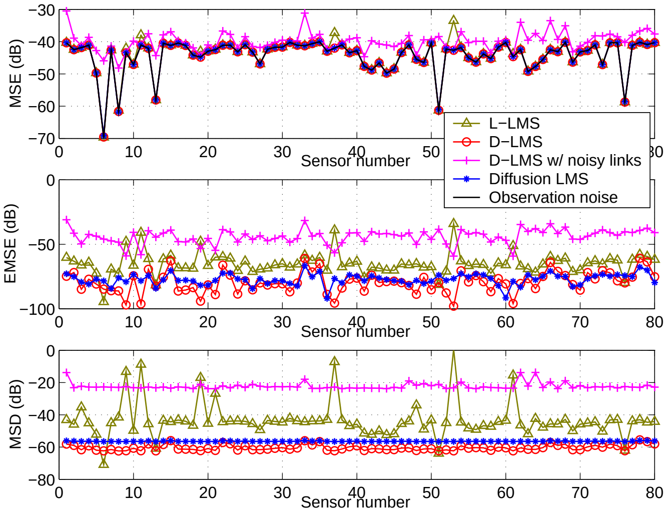

Figure 2.5: Local performance figures of merit: MSE, EMSE and MSD.

parable performance benchmark while L-LMS stands on the other extreme. For both distributed approaches in the (communication) noise-free setting, the resultant misadjustment is negligible thus matching the performance (in this sense) of its centralized counterparts. Furthermore, D-LMS outperforms diffusion LMS whereas its MSE remains bounded even when channel links are corrupted by reception noise; with an inflated steady-state MSE level, as expected.

To gauge local performance, we evaluate the figures of merit which are customary in the adaptive literature [27,28]: (i) MSE $E[e_j(t)^2]$, (ii) excess-MSE (EMSE) $E[e_j(t)^2] - \sigma^2_{\epsilon_j}$, and (ii) mean-square deviation (MSD) $E[\|\mathbf{s}_j(t) - \mathbf{s}_0\|^2]$. For the previous setup, the steady-state values of these metrics are depicted in Fig. 2.5. Good performance for D-LMS is apparent from the MSE curve which comes very close to the local noise levels; observe also the small EMSE. Sensor collaboration, despite the diverse noise statistical profile across the WSN, smoothens network-wide EMSE/MSD values. In comparison with diffusion LMS, D-LMS exhibits a slight edge on MSD while EMSE levels are comparable.

Under the same WSN setup, we illustrate the capabilities of D-LMS when it comes to tracking a time-varying signal vector $\mathbf{s}_0(t)$. The large amplitude slowly time-varying process model $\mathbf{s}_0(t) = (1-\rho)\mathbf{s}_0(t-1) + \sqrt{\rho}\boldsymbol{\nu}(t)$ is simulated, with $\rho = 10^{-2}$ and $\boldsymbol{\nu}(t) \sim \mathcal{N}(\mathbf{0}, 2 \times$

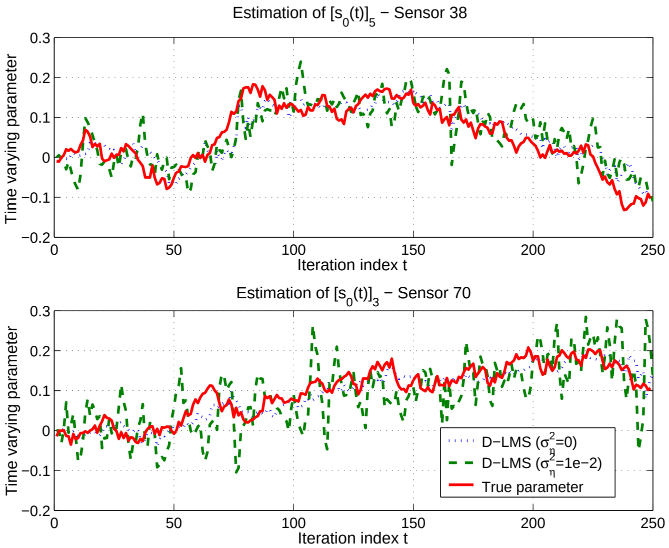

Figure 2.6: Tracking with D-LMS.

$10^{-2}\mathbf{I}_8$). Fig. 2.6 depicts the fifth and third entries of the true time-varying parameter $\mathbf{s}_0(t)$, and the respective estimates from sensors 38 and 70 that closely follow the true variations. Both sensors and parameter entries were chosen uniformly at random, in the interest of showing the representative behavior across the WSN. In addition, we also plot the estimates $[\mathbf{s}_{38}(t)]_5, [\mathbf{s}_{70}(t)]_3$ obtained when the WSN model accounts for communication noise with $\sigma_\eta^2 = 10^{-2}$. Larger estimate fluctuations are a direct manifestation of the increased MSE.

Next, we examine the D-LMS performance in the spectrum estimation application described earlier in Remark 2.1 for the aforementioned WSN setting. The AR source has order $p = 4$ and coefficients $\mathbf{s}_0 = [-0.31, 1.14, 0.275, 0.222]^T$. The source signal propagates via multi-path channels of order $L_j = 2$ and arrives at the sensors where it gets contaminated with sensing noise having variance 10^{-4} [cf. (2.4)]. The channels corresponding to sensors $3, 7, 15, 27, 37, 57, 67$ are set so that they have a null at the frequency where the AR source has a peak, namely at $\omega = \pi/2$. Fig. 2.7 depicts the actual power spectral density of the source as well as the estimated ones at sensor 15 using L-LMS and D-LMS under ideal and noisy inter-sensor links. The step-size is $\mu = 0.98 \times 10^{-3}$, while $\varepsilon_j = c_j = 1$. Clearly, even in the presence of communication noise D-LMS exploits the spatial diversity available and

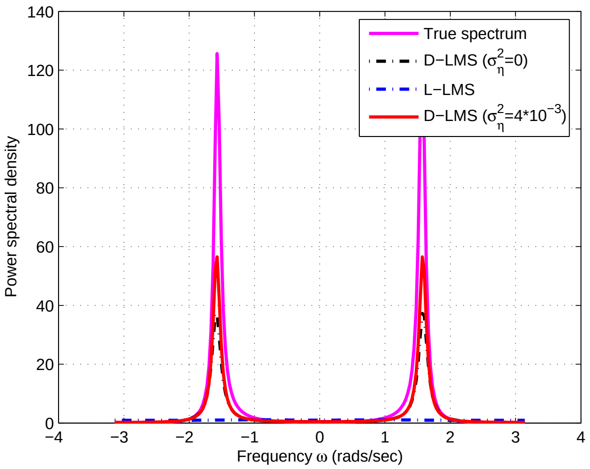

Figure 2.7: Spectral estimation with D-LMS.

allows all sensors to estimate accurately the actual spectral peak, whereas L-LMS leads the problematic sensors, e.g., sensor 15 in Fig. 2.7, to misleading estimates. The latter corroborates the ability of D-LMS to percolate information across the WSN.

2.4 Stability and Performance Analysis

An attractive feature of D-LMS is that it can be applied to a wide class of signals. Indeed, D-LMS requires no assumption on the statistics of $\{x_j(t), \mathbf{h}_j(t)\}_{j=1}^J$. When it comes to stability and performance evaluation however, a meaningful 'ground-truth' model should be adopted to carry out the analysis and enable fair comparison among competing alternatives. Toward this end, we adopt the standard data model, commonly used throughout the adaptive signal processing literature, e.g., [52, Ch. 5,9], [51], [27, 28]:

(a2) *The sensor observations adhere to the linear model*

$$x_j(t) = \mathbf{h}_j^T(t)\mathbf{s}_0 + \epsilon_j(t), \quad j \in \mathcal{J} \tag{2.21}$$

where the white noise $\epsilon_j(t)$ is zero-mean with variance $\sigma_{\epsilon_j}^2$.

In order to facilitate stability analysis, an important preliminary step is to express D-LMS as a linear time-varying (LTV) stochastic difference-equation. Specifically, starting from (2.14)-(2.16) and applying simple algebraic manipulations we will obtain recursions for the local estimation errors $\mathbf{y}_{1,j}(t) := \mathbf{s}_j(t) - \mathbf{s}_0$ and the local sum of multipliers $\mathbf{y}_{2,j}(t) := \sum_{b\in\mathcal{B}_j} \mathbf{v}_j^b(t-1)\ \forall\ j \in \mathcal{J}$. For simplicity in exposition, set $\varepsilon_j^2 c_j = \mu_j^{-\gamma} c_1/2$, $\varepsilon_j c_j = \mu_j^{-\gamma/2} c_2$ for any $\gamma \in (0,1)$, and $\mu_j = \mu\ \forall\ j \in \mathcal{J}$. Such a selection of ε_j is well motivated, since it gives more emphasis to the information gathered from the neighborhood [cf. (2.19)]. This is desirable in WSN-based applications including the one described in Remark 2.1, where sensor collaboration is essential to efficiently estimate the parameters of interest. Further, stack $\{\mathbf{y}_{1,j}(t), \mathbf{y}_{2,j}(t)\}_{j=1}^J$ vectors to form the $Jp \times 1$ supervectors $\mathbf{y}_1(t) := [\mathbf{y}_{1,1}^T(t) \dots \mathbf{y}_{1,J}^T(t)]^T$ and $\mathbf{y}_2(t) := [\mathbf{y}_{2,1}^T(t) \dots \mathbf{y}_{2,J}^T(t)]^T$, respectively; and let $\mathbf{y}(t) := [\mathbf{y}_1^T(t)\ \mathbf{y}_2^T(t)]^T \in \mathbb{R}^{2Jp\times 1}$. The Lagrange multipliers $\mathbf{v}_j^b(-1)$ are initialized so that

$$\sum_{j\in\mathcal{N}_b} \mathbf{v}_j^b(-1) = \mathbf{0}, \quad \forall b \in \mathcal{B}. \tag{2.22}$$

Equation (2.22) is easy to satisfy since sensors can initialize arbitrarily their multipliers, e.g., through zero initial conditions whereby sensor j sets $\mathbf{v}_j^b(-1) = \mathbf{0}$ with $b \in \mathcal{B}_j$.

The next step is to rewrite the consensus variable recursion in (2.16) in a way that will allow us later on to derive a first-order recursion for $\mathbf{y}(t)$. Specifically, we prove in Appendix A that:

Lemma 2.1 *If the Lagrange multipliers $\mathbf{v}_j^b(t)$ are initialized as in (2.22), then the consensus variables $\bar{\mathbf{s}}_b(t)$ can be expressed for $t \geq -1$ as*

$$\bar{\mathbf{s}}_b(t+1) = \frac{1}{|\mathcal{N}_b|}\sum_{j\in\mathcal{N}_b} \mathbf{s}_j(t+1) + \frac{1}{|\mathcal{N}_b|}\sum_{j\in\mathcal{N}_b}\left[\bar{\boldsymbol{\eta}}_b^j(t+1) - \bar{\boldsymbol{\eta}}_b^j(t)\right] - \frac{1}{|\mathcal{N}_b|}\sum_{j\in\mathcal{N}_b} \boldsymbol{\eta}_j^b(t) \tag{2.23}$$

with $\boldsymbol{\eta}_j^b(-1) = \bar{\boldsymbol{\eta}}_b^j(-1) = \mathbf{0}$, and $\bar{\boldsymbol{\eta}}_b^b(t) = \boldsymbol{\eta}_b^b(t) = \mathbf{0}$.

Using Lemma 2.1, as well as the multiplier update rule in (2.14) we wish to derive first-order recursions for $\mathbf{y}_{1,j}(t) := \mathbf{s}_j(t) - \mathbf{s}_0$ and $\mathbf{y}_{2,j}(t) := \sum_{b\in\mathcal{B}_j} \mathbf{v}_j^b(t-1)$. To this end, let us define the noise vectors

$$\bar{\boldsymbol{\eta}}_j(t) := \sum_{b\in\mathcal{B}_j}\sum_{j'\in\mathcal{N}_b} \frac{\bar{\boldsymbol{\eta}}_b^{j'}(t)}{|\mathcal{N}_b|}, \quad j \in \mathcal{J} \tag{2.24}$$

that depend on the reception noise at the bridge neighbors. Further, consider two more noise vectors

$$\boldsymbol{\eta}_j^{\alpha}(t) := \sum_{b\in\mathcal{B}_j} \boldsymbol{\eta}_j^{b}(t), \qquad \boldsymbol{\eta}_j^{\beta}(t) := \sum_{b\in\mathcal{B}_j}\sum_{j'\in\mathcal{N}_b} \frac{\boldsymbol{\eta}_{j'}^{b}(t)}{|\mathcal{N}_b|}. \tag{2.25}$$

Based on definitions (2.24) and (2.25), it is shown in Appendix B that:

Lemma 2.2 *Under (a2) and with* $\{\mathbf{v}_j^b(-1)\}_{j\in\mathcal{N}_b}$ *selected to satisfy (2.22), local state vectors* $\{\mathbf{y}_{i,j}(t)\}_{j=1}^{J}$ *obey the recursions*

$$\begin{aligned}
\mathbf{y}_{1,j}(t+1) = {} & \left(\mathbf{I}_p - 2\mu\mathbf{h}_j(t+1)\mathbf{h}_j^T(t+1) - \mu^{1-\gamma}c_1|\mathcal{B}_j|\mathbf{I}_p\right)\mathbf{y}_{1,j}(t) \\
& + \mu^{1-\gamma}c_1\left[\sum_{b\in\mathcal{B}_j}\left(\sum_{j'\in\mathcal{N}_b}\frac{\mathbf{y}_{1,j'}(t)}{|\mathcal{N}_b|}\right) - \frac{\mu^{\gamma/2}}{2c_2}\mathbf{y}_{2,j}(t) + \left(\bar{\boldsymbol{\eta}}_j(t) - \bar{\boldsymbol{\eta}}_j(t-1)\right)\right. \\
& \left. + \left(\boldsymbol{\eta}_j^{\alpha}(t) - \boldsymbol{\eta}_j^{\beta}(t-1)\right)\right] + 2\mu\mathbf{h}_j(t+1)\epsilon_j(t+1)
\end{aligned} \tag{2.26}$$

$$\begin{aligned}
\mathbf{y}_{2,j}(t+1) = \mathbf{y}_{2,j}(t) + \mu^{-\gamma/2}c_2 & \left[|\mathcal{B}_j|\mathbf{y}_{1,j}(t) - \sum_{b\in\mathcal{B}_j}\left(\sum_{j'\in\mathcal{N}_b}\frac{\mathbf{y}_{1,j'}(t)}{|\mathcal{N}_b|}\right) - \left(\bar{\boldsymbol{\eta}}_j(t) - \bar{\boldsymbol{\eta}}_j(t-1)\right)\right. \\
& \left. - \left(\boldsymbol{\eta}_j^{\alpha}(t) - \boldsymbol{\eta}_j^{\beta}(t-1)\right)\right].
\end{aligned} \tag{2.27}$$

Aiming at a first-order recursion for $\mathbf{y}(t)$, consider concatenating the noise terms in (2.24) and (2.25) for $j = 1, \ldots, J$ to form the $Jp \times 1$ supervectors $\bar{\boldsymbol{\eta}}(t)$, $\boldsymbol{\eta}^{\alpha}(t)$ and $\boldsymbol{\eta}^{\beta}(t)$, respectively; and also define the global observation noise vector

$$\boldsymbol{\epsilon}(t) := 2\mu[\mathbf{h}_1^T(t)\epsilon_1(t)\ldots\mathbf{h}_J^T(t)\epsilon_J(t)]^T. \tag{2.28}$$

Upon stacking $\mathbf{y}_{1,j}(t+1)$ and $\mathbf{y}_{2,j}(t+1)$ from Lemma 2.2, for $j = 1, \ldots, J$, in $\mathbf{y}(t+1)$, it is shown in Appendix C that the D-LMS recursions (2.14)-(2.16) can be compactly written in matrix form as

$$\mathbf{y}(t+1) = \boldsymbol{\Psi}(t+1,\mu)\mathbf{y}(t) + \begin{bmatrix} \mu^{1-\gamma}c_1\mathbf{I}_{Jp} & \mu^{1-\gamma}c_1\mathbf{I}_{Jp} \\ -\mu^{\gamma/2}c_2\mathbf{I}_{Jp} & -\mu^{\gamma/2}c_2\mathbf{I}_{Jp} \end{bmatrix} \begin{bmatrix} \bar{\boldsymbol{\eta}}(t) - \bar{\boldsymbol{\eta}}(t-1) \\ \boldsymbol{\eta}^{\alpha}(t) - \boldsymbol{\eta}^{\beta}(t-1) \end{bmatrix} + \begin{bmatrix} \boldsymbol{\epsilon}(t+1) \\ \mathbf{0} \end{bmatrix} \tag{2.29}$$

where for $t \geq 0$ the $2Jp\times 2Jp$ transition matrix $\boldsymbol{\Psi}(t,\mu)$ consists of four $Jp\times Jp$ matrix blocks given by $[\boldsymbol{\Psi}(t,\mu)]_{11} = \mathbf{I}_{Jp} - 2\mu\mathbf{R}_h(t) - \mu^{1-\gamma}c_1\mathbf{A}$, $[\boldsymbol{\Psi}(t,\mu)]_{12} = -\frac{\mu^{1-\gamma/2}c_1}{2c_2}\mathbf{I}_{Jp}$, $[\boldsymbol{\Psi}(t,\mu)]_{21} =$

$\mu^{-\gamma/2}c_2\mathbf{A}$ and $[\boldsymbol{\Psi}(t,\mu)]_{22} = \mathbf{I}_{Jp}$, with $\mathbf{R}_h(t) := \text{bdiag}\left(\mathbf{h}_1(t)\mathbf{h}_1^T(t), \ldots, \mathbf{h}_J(t)\mathbf{h}_J^T(t)\right)$ while the structure of $\mathbf{A}$ is given in (2.20).

The linear dynamical system described by (2.29) is indeed time-varying since $\boldsymbol{\Psi}(t,\mu)$ as well as $\mathbf{R}_h(t)$ are time dependent. It is also random due to the noise terms as well as the regression vectors $\mathbf{h}_j(t)$. In order to satisfy the initialization requirement in (2.22), $\mathbf{y}_{2,j}(0)$ should be set to $\mathbf{y}_{2,j}(0) = \sum_{b\in\mathcal{B}_j} \mathbf{v}_j^b(-1) = \mu^{-\gamma/2}c_2\mathbf{A}_j\mathbf{y}_2'(0)$, where $\mathbf{A}_j$ is the j-th $p \times Jp$ block of the decomposition $\mathbf{A} = [\mathbf{A}_1^T \ldots \mathbf{A}_J^T]^T$ and $\mathbf{y}_2'(0)$ can be chosen arbitrarily. Hence, using

$$\mathbf{y}_{2,j}(0) = \mu^{-\gamma/2}c_2 \sum_{b\in\mathcal{B}_j} \left[\mathbf{y}_{2,j}'(0) - \frac{1}{|\mathcal{N}_b|}\sum_{j'\in\mathcal{N}_b} \mathbf{y}_{2,j'}'(0)\right] \tag{2.30}$$

and setting $\mathbf{v}_j^b(-1) = \mathbf{y}_{2,j}'(0) - |\mathcal{N}_b|^{-1}\sum_{j'\in\mathcal{N}_b} \mathbf{y}_{2,j'}'(0)$ ensures that (2.22) holds true, while $\mu^{-\gamma/2}c_2$ is placed for normalization.

The LTV system in (2.29) is not yet ready for stability analysis since $E_{\mathbf{h}}[\boldsymbol{\Psi}(t,\mu)]$ does not have all its eigenvalues inside the unit circle. Towards reformulating (2.29), consider the $p(\sum_{b\in\mathcal{B}} |\mathcal{N}_b|) \times 1$ supervector

$$\boldsymbol{\eta}(t) := \left[\{\boldsymbol{\eta}_{j'}^{b_1}(t)\}_{j'\in\mathcal{N}_{b_1}}^T \ldots \{\boldsymbol{\eta}_{j'}^{b_{|\mathcal{B}|}}(t)\}_{j'\in\mathcal{N}_{b_{|\mathcal{B}|}}}^T\right]^T \tag{2.31}$$

comprising the receiver noise of the bridge sensors' transmissions to their neighbors; i.e., the first $|\mathcal{N}_{b_1}|$ vectors in $\boldsymbol{\eta}(t)$ correspond to the reception noise at the neighbors of bridge sensor b_1 and so on. Using $\boldsymbol{\eta}(t)$, the noise supervectors $\boldsymbol{\eta}^\alpha(t)$ and $\boldsymbol{\eta}^\beta(t)$ can be written as $\boldsymbol{\eta}^\alpha(t) = \mathbf{P}_\alpha\boldsymbol{\eta}(t)$ and $\boldsymbol{\eta}^\beta(t) = \mathbf{P}_\beta\boldsymbol{\eta}(t)$, where the structure of the time-invariant matrices $\mathbf{P}_\alpha$ and $\mathbf{P}_\beta$ can be found in Appendix D, which establishes:

Lemma 2.3 *The LTV system in (2.29) can be equivalently written as*

$$\mathbf{y}(t+1) = \mathit{bdiag}(\mathbf{I}_{Jp}, \mu^{-\gamma/2}c_2\mathbf{A})\mathbf{z}(t+1) + \begin{bmatrix} \mu^{1-\gamma}c_1\mathbf{I}_{Jp} \\ -\mu^{-\gamma/2}c_2\mathbf{I}_{Jp} \end{bmatrix} (\bar{\boldsymbol{\eta}}(t) + \boldsymbol{\eta}^\alpha(t)) \tag{2.32}$$

where the state $\mathbf{z}(t) := [\mathbf{z}_1^T(t)\ \mathbf{z}_2^T(t)]^T$ *is arbitrarily initialized as* $\mathbf{z}(0) := [\mathbf{y}_1^T(0)\ \ (\mathbf{y}_2'(0))^T]^T$*; updated according to*

$$\mathbf{z}(t+1) = \boldsymbol{\Phi}(t+1,\mu)\mathbf{z}(t) + \mathbf{R}_h^\alpha(t+1)\bar{\boldsymbol{\eta}}(t-1) + \mathbf{R}_h^\beta(t+1)\boldsymbol{\eta}(t-1) + [\boldsymbol{\epsilon}^T(t+1)\quad \mathbf{0}^T]^T, \tag{2.33}$$

and the transition matrix $\mathbf{\Phi}(t+1,\mu)$ *consists of the submatrices* $[\mathbf{\Phi}(t,\mu)]_{11} = \mathbf{I}_{Jp} - 2\mu\mathbf{R}_h(t) - \mu^{1-\gamma}c_1\mathbf{A}$, $[\mathbf{\Phi}(t,\mu)]_{12} = -\frac{\mu^{1-\gamma}c_1}{2}\mathbf{A}$ *and* $[\mathbf{\Phi}(t,\mu)]_{21} = [\mathbf{\Phi}(t,\mu)]_{22} = \mathbf{A}\mathbf{A}^\dagger$. *Matrices* $\mathbf{R}_h^\alpha(t+1)$ *and* $\mathbf{R}_h^\beta(t+1)$ *are defined as*

$$\mathbf{R}_h^\alpha(t+1) := \left[\frac{\mu^{1-\gamma}c_1}{2}\mathbf{I}_{Jp} - \mu^{2(1-\gamma)}c_1^2\mathbf{A}^T - 2\mu^{2-\gamma}c_1\mathbf{R}_h^T(t+1), \quad \mu^{1-\gamma}c_1\mathbf{I}_{Jp}\right]^T \tag{2.34}$$

$$\mathbf{R}_h^\beta(t+1) := \left[\mu^{1-\gamma}c_1\left(\frac{3}{2}\mathbf{P}_\alpha^T - \mu^{1-\gamma}c_1(\mathbf{A}\mathbf{P}_\alpha)^T - 2\mu(\mathbf{R}_h(t+1)\mathbf{P}_\alpha)^T - \mathbf{P}_\beta^T\right),\right.$$
$$\left.\mu^{1-\gamma}c_1\mathbf{P}_\alpha^T + 2\mathbf{C}_R^T\right]^T \tag{2.35}$$

with $\mathbf{C}_R$ *chosen such that* $\mathbf{A}\mathbf{C}_R = \mathbf{P}_\beta - \mathbf{P}_\alpha$.

2.4.1 Stability Analysis

This subsection deals with stability analysis of D-LMS based on the equivalent LTV system derived in Lemma 2.3. Specifically, it will be shown that under mild conditions the error norm $\|\mathbf{y}(t)\|$ remains most of the time in a finite interval, i.e., errors are weakly stochastic bounded (WSB) [51]. This WSB stability guarantees that for any $\theta > 0$ there exists a $\delta > 0$ such that $\Pr[\|\mathbf{y}(t)\| < \delta] = 1 - \theta$ uniformly in t. As a consequence, it will be shown that in the absence of observation and inter-sensor communication noise the local estimation errors in D-LMS converge exponentially fast to zero with probability one. Therefore, consensus is achieved a.s., as all local sensor estimates agree on the true parameter $\mathbf{s}_0$. This establishes a strong connection with the known behavior of C-LMS [51], further validating the importance of D-LMS in a distributed setting.

The first step in proving that $\|\mathbf{y}(t)\|$ is bounded in probability is to show that the same holds for $\|\mathbf{z}(t)\|$. This will be established under the following assumptions:

(a3) *The regressor vectors* $\mathbf{h}_j(t)$ *are strictly stationary with* $\mathbf{R}_{h_j} := E[\mathbf{h}_j(t)\mathbf{h}_j^T(t)] < \infty$, *and* $\mathbf{R}_h := bdiag(\mathbf{R}_{h_1}, \ldots, \mathbf{R}_{h_J}) \succ \mathbf{0}$. *Regressors are ergodic and satisfy (a.s.)*

$$\lim_{t\to\infty} t^{-1}\sum_{\tau=1}^{t}\|\mathbf{R}_h - \mathbf{R}_h(\tau)\| = E[\|\mathbf{R}_h - \mathbf{R}_h(t)\|] := m_{\tilde{h}} < \infty \tag{2.36}$$

(a4) *Communication and observation noise vector norms are bounded in the mean; i.e.,*

$$E[\|\epsilon_j(t)\|] = m_\epsilon < \infty, \;\; E[\|\boldsymbol{\eta}_j^b(t)\|] = m_\eta < \infty \text{ and } E[\|\bar{\boldsymbol{\eta}}_b^j(t)\|] = m_{\bar{\eta}} < \infty. \tag{2.37}$$

Under assumption (a4), typically met in practice, the estimation error does not grow unbounded.

Necessary for proving boundedness of $\mathbf{z}(t)$ in (2.33) is to show that the norm of $\mathbf{\Phi}(1 : t, \mu) := \prod_{\tau=1}^{t} \mathbf{\Phi}(\tau, \mu)$ converges to zero as $t \to \infty$, a.s.. To this end, let $\lambda^{+}(\mathbf{A})$ denote the minimum positive eigenvalue of $\mathbf{A} \succeq \mathbf{0}$. Specifically, it is established in Appendix E that:

Lemma 2.4 *If (a3) holds true, c_1 is selected such that $c_1\lambda^{+}(\mathbf{A}) > 4$ and the step-size $\mu \in (0, \mu_u')$, with $\mu_u' < \mu_u$ is chosen as*

$$\mu_u := \left[2 \min \left(\lambda_{\max}^{-1}(\mathbf{R}_h + (c_1/2)\mathbf{A}), \lambda_{\max}^{-1}(2\mathbf{R}_h + (3c_1/4)\mathbf{A})\right)\right]^{1/(1-\gamma)} \tag{2.38}$$

to guarantee that the eigenvalues of $E_{\mathbf{h}}[\mathbf{\Phi}(t, \mu)]$ are less than one, then with $\lambda \in (0, 1)$

$$\limsup\nolimits_{t\to\infty} t^{-1} \log \|\mathbf{\Phi}(1 : t, \mu)\| \leq \log(\lambda) < 0, \qquad a.s.. \tag{2.39}$$

In words, (2.39) establishes that $\|\mathbf{\Phi}(1 : t, \mu)\|$ will converge to zero exponentially fast with probability one. This property is necessary to prove later on that $\|\mathbf{z}(t)\|$ satisfies the WSB property, which in turn will lead us to the ultimate goal of establishing stochastic boundedness of $\|\mathbf{y}(t)\|$. Thus, using Lemma 2.4, we prove in Appendix F the following result.

Lemma 2.5 *Under (a3)-(a4), $c_1\lambda^{+}(\mathbf{A}) > 4$ and if $\mu \in (0, \mu_u')$ with $\mu_u' < \mu_u$, then $\mathbf{z}(t)$ satisfies the WSB property; i.e.,*

$$\lim_{\delta\to\infty} \sup_{t\geq 0} \Pr[\|\mathbf{z}(t)\| \geq \delta] = 0. \tag{2.40}$$

Careful inspection of (2.32), and exploitation of Lemmas 2.4 and 2.5, along with (a4) reveals that $\mathbf{y}(t)$ is also WSB. As a result, it is possible to prove the following main result (see Appendix G).

Proposition 2.1 *Under (a2)-(a4), $c_1\lambda^{+}(\mathbf{A}) > 4$ and if $\mu \in (0, \mu_u')$ with $\mu_u' < \mu_u$, then $\mathbf{y}(t)$ is WSB; i.e.,*

$$\lim_{\delta\to\infty} \sup_{t\geq 0} \Pr[\|\mathbf{y}(t)\| \geq \delta] = 0. \tag{2.41}$$

Proposition 2.1 asserts that no probability mass of $\mathbf{y}(t)$, and consequently of the local estimation errors $\mathbf{y}_{1,j}(t) := \mathbf{s}_j(t) - \mathbf{s}_0$, escapes to infinity. This is very important since even in the presence of observation and communication noise the local estimation errors remain bounded. Interestingly, in the absence of noise the local estimates $\mathbf{s}_j(t)$ provided by D-LMS converge exponentially fast to $\mathbf{s}_0$ with probability one. Thus, D-LMS exhibits behavior similar to its centralized counterpart when it comes to stationary ergodic signals [51]. Actually, it follows readily from Lemma 2.5 that (cf. [50, Proof of Lemma 3]):

Corollary 2.1 *If $c_1\lambda^+(\mathbf{A}) > 4$ and $\mu \in (0, \mu'_u)$ with $\mu'_u < \mu_u$, (a2) and (a3) hold true, while $\epsilon_j(t) = 0$ and $\boldsymbol{\eta}_j^b(t) = \bar{\boldsymbol{\eta}}_b^j(t) = \mathbf{0}$ for $j \in \mathcal{J}$ and $b \in \mathcal{B}$, then there exists $t_0 < \infty$, a random variable $B < \infty$ and $\lambda(\mu) \in (0,1)$ such that*

$$\|\mathbf{y}(t)\| \leq B\lambda^t(\mu), \;\; a.s. \;\; \forall \;\; t > t_0. \tag{2.42}$$

Corollary 1 demonstrates that the WSN achieves consensus in the sense that local estimation errors $\{\mathbf{s}_j(t) - \mathbf{s}_0\}_{j=1}^J$ converge to zero exponentially fast on a per realization basis. Interestingly, μ_u resembles the stability bound for C-LMS, namely $2/\lambda_{\max}(\mathbf{R}_h)$ [52, Ch. 9]. The main difference here is that this bound is also affected by the topology of the WSN, via $\mathbf{A}$, due to the distributed nature of the algorithm and the information exchanges among sensors.

2.4.2 Performance Analysis

In this subsection the estimation performance of D-LMS is analyzed by approximating the error covariance matrix. Since the recursion (2.32) involved in D-LMS is time-varying, a closed-form expression for the error covariance matrix is difficult, if not impossible, to obtain. Specifically, the estimation MSE associated with the D-LMS recursions in (2.32)-(2.33) evolves according to

$$e_{mse}(t) := J^{-1}E[\mathbf{y}_1^T(t)\mathbf{y}_1(t)] = J^{-1}E[\mathbf{z}_1^T(t)\mathbf{z}_1(t)] + (\mu^{1-\gamma}c_1)^2 J^{-1}\text{trace}(\mathbf{R}_{\bar{\eta}} + \mathbf{P}_\alpha\mathbf{R}_\eta\mathbf{P}_\alpha^T), \tag{2.43}$$

where $E[\mathbf{z}_1^T(t)\mathbf{z}_1(t)]$ denotes the trace of the upper left $Jp \times Jp$ submatrix of the covariance matrix $E[\mathbf{z}(t)\mathbf{z}^T(t)]$ that evolves according to

$$\begin{aligned} E[\mathbf{z}(t)\mathbf{z}^T(t)] &= E[\boldsymbol{\Phi}(t,\mu)\mathbf{z}(t-1)\mathbf{z}^T(t-1)\boldsymbol{\Phi}^T(t,\mu)] + E[\mathbf{R}_h^{\alpha}(t)\mathbf{R}_{\bar{\eta}}(\mathbf{R}_h^{\alpha}(t))^T] \\ &\quad + E[\mathbf{R}_h^{\beta}(t)\mathbf{R}_{\eta}(\mathbf{R}_h^{\beta}(t))^T] + \mathrm{bdiag}(E[\boldsymbol{\epsilon}(t)\boldsymbol{\epsilon}^T(t)], \mathbf{0}_{Jp\times Jp}) \\ &\quad + E[\boldsymbol{\Phi}(t,\mu)\mathbf{z}(t-1)[\boldsymbol{\epsilon}^T(t)\ \mathbf{0}^T]^T] + (E[\boldsymbol{\Phi}(t,\mu)\mathbf{z}(t-1)[\boldsymbol{\epsilon}^T(t)\ \mathbf{0}^T]^T])^T. \end{aligned} \tag{2.44}$$

The first expectation in the right hand side of (2.44) is impossible to split because the regressors are temporally correlated. Thus, under (a3)-(a4) it appears impossible to evaluate $E[\mathbf{z}(t)\mathbf{z}^T(t)]$. One possible alternative is to consider an appropriate time-invariant 'average' system approximating the LTV system in (2.32) and recursively evaluate its corresponding error covariance matrix. Then, using stochastic averaging arguments, see e.g., [52, Ch. 9], the estimation error $\bar{\mathbf{y}}_1(t)$ associated with the 'average' system can be shown convergent in probability to $\mathbf{y}_1(t)$ as the step-size μ approaches zero. This approach allows approximating the estimation MSE of D-LMS with that of the average system

$$\bar{\mathbf{y}}(t+1) = \mathrm{bdiag}(\mathbf{I}_{Jp}, \mu^{-\gamma/2}c_2\mathbf{A})\bar{\mathbf{z}}(t+1) + \begin{bmatrix} \mu^{1-\gamma}c_1\mathbf{I}_{Jp} \\ -\mu^{-\gamma/2}c_2\mathbf{I}_{Jp} \end{bmatrix} (\bar{\boldsymbol{\eta}}(t) + \boldsymbol{\eta}^{\alpha}(t)) \tag{2.45}$$

$$\bar{\mathbf{z}}(t+1) = \boldsymbol{\Phi}(\mu)\bar{\mathbf{z}}(t) + \mathbf{R}_h^{\alpha}\bar{\boldsymbol{\eta}}(t-1) + \mathbf{R}_h^{\beta}\boldsymbol{\eta}(t-1) + [\boldsymbol{\epsilon}^T(t+1)\quad \mathbf{0}^T]^T \tag{2.46}$$

where $\boldsymbol{\Phi}(\mu) := E_{\mathbf{h}}[\boldsymbol{\Phi}(t,\mu)]$, $\mathbf{R}_h^{\alpha} := E_{\mathbf{h}}[\mathbf{R}_h^{\alpha}(t+1)]$ and $\mathbf{R}_h^{\beta} := E_{\mathbf{h}}[\mathbf{R}_h^{\beta}(t+1)]$. Note that the average system in (2.45) is not constructed by taking expectations on both sides of (2.32)-(2.33). Instead, it is formed starting from the primary system and replacing the time-varying transition matrix $\boldsymbol{\Phi}(t,\mu)$, and the matrices $\mathbf{R}_h^{\alpha}(t+1), \mathbf{R}_h^{\beta}(t+1)$ with their time-invariant counterparts $\boldsymbol{\Phi}(\mu)$, and $\mathbf{R}_h^{\alpha}, \mathbf{R}_h^{\beta}$ respectively. This average system plays a key role in the stochastic averaging approach of [52, Ch. 9].

Next, we see how the local estimation errors in $\mathbf{y}_1(t)$ are statistically related with the average state vector $\bar{\mathbf{y}}_1(t)$. Recall that both $\mathbf{y}_1(t)$ and $\bar{\mathbf{y}}_1(t)$ depend on μ. Actually, it is shown in Appendix H that:

Proposition 2.2 *If $\mathbf{z}(0) = \bar{\mathbf{z}}(0)$ and (a2)-(a4) are satisfied, while the joint moments of $\{\mathbf{h}_j(t)\}_{j=1}^J$ are bounded, then given finite $T > 0$ and for any $\delta > 0$ and $\beta > 0$ arbitrarily* *small, it holds that*

$$\lim_{\mu\to 0}\sup_{0\le t\le T/\mu^{1-\gamma-\beta}}\Pr[\|\bar{\mathbf{y}}_1(t)-\mathbf{y}_1(t)\|\ge\delta]=0. \tag{2.47}$$

Proposition 2.2 shows that the probability of the estimation error $\mathbf{y}_1(t)$ being close to $\bar{\mathbf{y}}_1(t)$ approaches unity with vanishing step-sizes, so long as the D-LMS and its 'average' version are initialized with the same local estimates and multipliers, i.e., $\mathbf{z}(0)=\bar{\mathbf{z}}(0)$. This type of result is referred to as *trajectory locking*, because the trajectory of the primary system hovers around and locks to the trajectory of its average counterpart. The time horizon for which the two systems remain 'locked' goes to infinity as $\mu\to 0$. These locking results are applicable even when regressors exhibit temporal correlations.

Now, observe that the 'average' D-LMS algorithm in (2.45) has a time-invariant transition matrix. As a result, the 'average' estimation error covariance $\mathbf{R}_{\bar{y}_1}(t):=E[\bar{\mathbf{y}}_1(t)\bar{\mathbf{y}}_1^T(t)]$ can be found in closed form. Specifically, we prove in Appendix I that

$$\mathbf{R}_{\bar{y}_1}(t+1)=[\mathbf{R}_{\bar{z}}(t+1)]_{11}+(\mu^{1-\gamma}c_1)^2\left(\mathbf{R}_{\bar{\eta}}+\mathbf{P}_\alpha\mathbf{R}_\eta\mathbf{P}_\alpha^T\right) \tag{2.48}$$

where $[\mathbf{R}_{\bar{z}}(t+1)]_{11}$ is the $Jp\times Jp$ upper left submatrix of

$$\mathbf{R}_{\bar{z}}(t+1)=\mathbf{\Phi}^{t+1}(\mu)\mathbf{R}_{\bar{z}}(0)(\mathbf{\Phi}^{t+1}(\mu))^T+\sum_{\tau=0}^{t}\mathbf{\Phi}^\tau(\mu)\mathbf{R}_{\breve{\eta}}(\mathbf{\Phi}^\tau(\mu))^T \tag{2.49}$$

while

$$\mathbf{R}_{\breve{\eta}}=\mathbf{R}_h^\alpha\mathbf{R}_{\bar{\eta}}(\mathbf{R}_h^\alpha)^T+\mathbf{R}_h^\beta\mathbf{R}_\eta(\mathbf{R}_h^\beta)^T+\text{bdiag}(E[\boldsymbol{\epsilon}(t+1)\boldsymbol{\epsilon}^T(t+1)],\mathbf{0}_{Jp\times Jp}) \tag{2.50}$$

with $\mathbf{R}_{\bar{\eta}}:=E[\bar{\boldsymbol{\eta}}(t)\bar{\boldsymbol{\eta}}^T(t)]$ and $\mathbf{R}_\eta:=E[\boldsymbol{\eta}(t)\boldsymbol{\eta}^T(t)]$; see also Appendix I for the structure of $\mathbf{R}_{\bar{\eta}}$ and $\mathbf{R}_\eta$.

Based on Proposition 2.2, the covariance matrix in (2.48) can be viewed as an approximation to $\mathbf{R}_{y_1}(t):=E[\mathbf{y}_1(t)\mathbf{y}_1^T(t)]$. Then, the global normalized estimation error of D-LMS at time instant t can be approximated as

$$e_{mse}(t):=\frac{1}{J}\sum_{j=1}^{J}E[\|\mathbf{s}_j(t)-\mathbf{s}_0\|^2]\approx J^{-1}\text{trace}(\mathbf{R}_{\bar{y}_1}(t)), \tag{2.51}$$

while local approximate MSE performance across sensors can be acquired from the corresponding diagonal entries of $\mathbf{R}_{\bar{y}_1}(t)$. Proposition 2.2 implies that this approximation improves as $\mu \to 0$. Intuitively, this happens because a vanishing step-size suppresses temporal correlations present in the regressors thus making D-LMS behave as the 'average system' in (2.45). In the interest of tractability, the 'average system' does not take into account temporal correlations. Thus, for a small step-size the MSE corresponding to the 'average' system in (2.45) can be used to approximate efficiently the one associated with D-LMS, which will be corroborated via simulations.

Remark 2.7 (Comparison with existing results) The stochastic stability results presented in Section 2.4.1 allow for (non-) Gaussian distributed and spatio-temporally correlated regressors. As for C-LMS, WSB has been established under conditions similar to those in [51]. Communication noise has not been considered earlier, and stochastic boundedness of D-LMS is a consequence of the inherent noise-robustness of the method of multipliers; see also [49] for related claims in single-shot non-adaptive distributed estimation. Mean and MSE convergence results for diffusion LMS [28] were established under the widely assumed white Gaussian setting [52, Ch. 5]; similar mean-stability results were reported for D-LMS in [30]. A comparison between the stability results here and those in [28] is not possible since they are different in nature. Regarding performance analysis, steady-state closed-form expressions of the relevant figures of merit have been derived for both incremental and diffusion LMS [27,28]; when regressors are Gaussian and independent in space and time by relying on an energy conservation framework [43]. This should be contrasted with the alternative approach delineated in Section 2.4.2, that utilizes stochastic averaging arguments to approximate the MSE associated with D-LMS. This approximation becomes increasingly accurate for a vanishing step-size, since the regressor's temporal correlations are suppressed making D-LMS behave as in a white data setting.

2.4.3 Numerical Example

Here again, we test a WSN generated as a $\mathcal{G}^2(80, 0.6)$ graph yielding $\lambda^+(\mathbf{A}) = 0.904$. With $p = 8$ and $\mathbf{s}_0 = \mathbf{1}_p$, observations obey the linear model (2.21), where regressors are $\mathbf{h}_j(t) = [h_j(t) \dots h_j(t-p+1)]^T$ with $h_j(t)$ evolving according to an AR(1) process as in Section 2.3.3. We choose $\rho = 0.7$, the $u_{1,j} \sim \mathcal{U}[0, 1]$ are i.i.d. in space and the uniformly

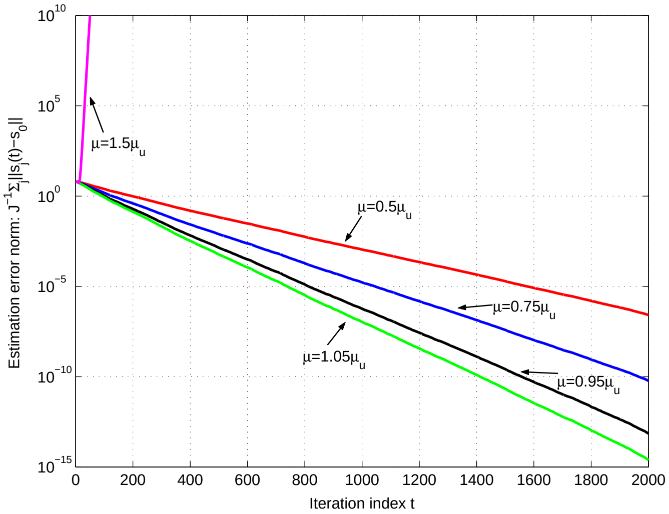

Figure 2.8: Normalized estimation error for D-LMS in the absence of noise.

distributed white driving noise has a spatial variance profile given by $\sigma^2_{\nu_j} = 2u_{2,j}$ with $u_{2,j} \sim \mathcal{U}[0,1]$ and i.i.d.. First, we corroborate the result in Corollary 2.1, by running D-LMS with $c_j = 2.25$, $\varepsilon_j = \mu^{-1/4}$ for all $j \in \mathcal{J}$ in a noise-free setup and computing the sample paths of the normalized estimation error $J^{-1}\sum_{j=1}^{J} \|\mathbf{s}_j(t) - \mathbf{s}_0\|$. Results are depicted in Fig. 2.8 for different step-sizes related to the upper bound $\mu_u = 8 \times 10^{-3}$. When $\mu \in (0, \mu_u)$, the error norm converges to zero exponentially fast with a decay rate increasing with μ. As per Corollary 2.1, convergence cannot be claimed for step-sizes larger than μ_u, though simulations indicate that the stability region may be larger than $(0, \mu_u)$.

Next, we validate the approximation (2.51) by plotting the empirically estimated MSE achieved by D-LMS in (2.32) (averaged over 50 Monte Carlo runs) and comparing it with the MSE achieved by the average system [cf. right hand side of (2.51)] in a $\mathcal{G}^2(60, 0.25)$ setting. Space-time i.i.d. observation noise $\epsilon_j(t) \sim \mathcal{N}(0, 10^{-3})$ is now added as well as receiver AWGN of variance $\sigma^2_\eta = 10^{-2}$. Fig. 2.9 confirms that the theoretical MSE obtained from the 'average' D-LMS in (2.45) approximates well the primary MSE in (2.32), especially as μ becomes smaller. Note that D-LMS is derived using the AD-MoM. Convergence in MoM is not necessarily monotonic and the same holds for the true and approximated 'average' behavior of D-LMS, as depicted in Fig. 2.9.

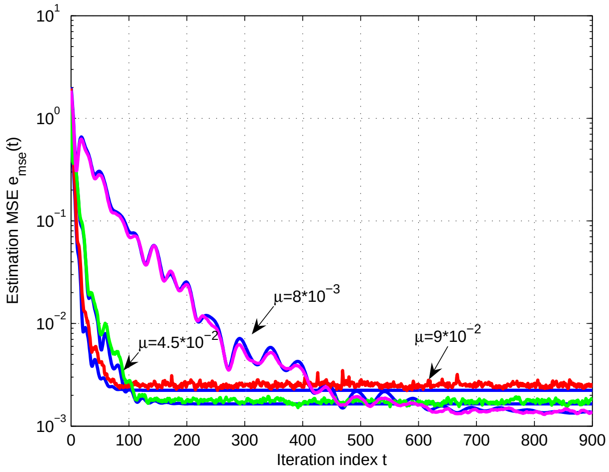

Figure 2.9: Empirical normalized estimation MSE for D-LMS and theoretical approximation (2.51) for the 'average' D-LMS.

2.5 Appendices

2.5.1 Proof of Lemma 2.1

Recall that in the presence of noise, bridge variables obey [cf. (2.16)]

$$\bar{\mathbf{s}}_b(t+1) = \left(|\mathcal{N}_b|\mu^{-\gamma/2}c_2\right)^{-1}\sum_{j\in\mathcal{N}_b}\left[\mathbf{v}_j^b(t)+\mu^{-\gamma/2}c_2(\mathbf{s}_j(t+1)+\bar{\boldsymbol{\eta}}_b^j(t+1))\right]. \tag{2.52}$$

Substituting (2.14) into (2.52), while adding and subtracting $\sum_{j\in\mathcal{N}_b}\bar{\boldsymbol{\eta}}_b^j(t)$, yields

$$\begin{aligned}\bar{\mathbf{s}}_b(t+1) = {} & \left(|\mathcal{N}_b|\mu^{-\gamma/2}c_2\right)^{-1}\sum_{j\in\mathcal{N}_b}\left[\mathbf{v}_j^b(t-1)+\mu^{-\gamma/2}c_2(\mathbf{s}_j(t)+\bar{\boldsymbol{\eta}}_b^j(t))\right]-\bar{\mathbf{s}}_b(t)\\ & +|\mathcal{N}_b|^{-1}\sum_{j\in\mathcal{N}_b}\mathbf{s}_j(t+1)+|\mathcal{N}_b|^{-1}\sum_{j\in\mathcal{N}_b}\left[\bar{\boldsymbol{\eta}}_b^j(t+1)-\bar{\boldsymbol{\eta}}_b^j(t)\right]-|\mathcal{N}_b|^{-1}\sum_{j\in\mathcal{N}_b}\boldsymbol{\eta}_j^b(t).\end{aligned} \tag{2.53}$$

Equations (2.22) and (2.52) imply that the first sum in (2.53) equals $\bar{\mathbf{s}}_b(t)$ for $t\geq 0$. Thus, (2.23) follows. □

2.5.2 Proof of Lemma 2.2

Summing (2.14) over $b \in \mathcal{B}_j$, we can write $\mathbf{y}_{2,j}(t+1)$ as

$$\begin{aligned}\mathbf{y}_{2,j}(t+1) &= \sum_{b\in\mathcal{B}_j}\left[\mathbf{v}_j^b(t-1) + \mu^{-\gamma/2}c_2(\mathbf{s}_j(t) - (\bar{\mathbf{s}}_b(t) + \boldsymbol{\eta}_j^b(t)))\right]\\ &= \mathbf{y}_{2,j}(t) + \mu^{-\gamma/2}c_2|\mathcal{B}_j|\mathbf{s}_j(t) - \mu^{-\gamma/2}c_2\sum_{b\in\mathcal{B}_j}\left(\sum_{j'\in\mathcal{N}_b}\frac{\mathbf{s}_{j'}(t)}{|\mathcal{N}_b|}\right)\\ &\quad - \mu^{-\gamma/2}c_2\left(\bar{\boldsymbol{\eta}}_j(t) - \bar{\boldsymbol{\eta}}_j(t-1)\right) - \mu^{-\gamma/2}c_2\left(\boldsymbol{\eta}_j^\alpha(t) - \boldsymbol{\eta}_j^\beta(t-1)\right)\end{aligned} \tag{2.54}$$

where the last equality follows after splitting the sum in the first equality into four individual terms and invoking Lemma 2.1. Adding and subtracting $\mu^{-\gamma/2}c_2|\mathcal{B}_j|\mathbf{s}_0$ from the right hand side of (2.54), yields (2.27) readily.

To prove (2.26), recall that the local estimate $\mathbf{s}_j(t)$ is updated as [cf. (2.15)]

$$\begin{aligned}\mathbf{s}_j(t+1) &= \mathbf{s}_j(t) + 2\mu\mathbf{h}_j(t+1)e_j(t+1) - \frac{\mu^{1-\gamma}c_1}{2}|\mathcal{B}_j|\mathbf{s}_j(t)\\ &\quad - \frac{\mu^{1-\gamma}c_1}{2}\left[\frac{\mu^{\gamma/2}}{c_2}\mathbf{y}_{2,j}(t+1) - \sum_{b\in\mathcal{B}_j}\bar{\mathbf{s}}_b(t) - \sum_{b\in\mathcal{B}_j}\boldsymbol{\eta}_j^b(t)\right].\end{aligned} \tag{2.55}$$

Upon: (i) using $e_j(t) := x_j(t) - \mathbf{h}_j^T(t)\mathbf{s}_j(t-1)$; (ii) substituting $x_j(t)$ from (a2) into (2.55); (iii) subtracting $\mathbf{s}_0$ from both sides of (2.55); and (iv) replacing $\bar{\mathbf{s}}_b(t)$ and $\mathbf{y}_{2,j}(t+1)$ from (2.23) and (2.27), respectively, we arrive at

$$\begin{aligned}\mathbf{y}_{1,j}(t+1) &= \mathbf{y}_{1,j}(t) - 2\mu\mathbf{h}_j(t+1)\mathbf{h}_j^T(t+1)\mathbf{y}_{1,j}(t) - \mu^{1-\gamma}c_1\Bigg[|\mathcal{B}_j|\mathbf{s}_j(t) + \frac{\mu^{\gamma/2}}{2c_2}\mathbf{y}_{2,j}(t)\\ &\quad - \sum_{b\in\mathcal{B}_j}\left(\sum_{j'\in\mathcal{N}_b}\frac{\mathbf{s}_{j'}(t)}{|\mathcal{N}_b|}\right) - \left(\bar{\boldsymbol{\eta}}_j(t) - \bar{\boldsymbol{\eta}}_j(t-1)\right) - \left(\boldsymbol{\eta}_j^\alpha(t) - \boldsymbol{\eta}_j^\beta(t-1)\right)\Bigg]\\ &\quad + 2\mu\mathbf{h}_j(t+1)\epsilon_j(t+1)\\ &= \mathbf{y}_{1,j}(t) - 2\mu\mathbf{h}_j(t+1)\mathbf{h}_j^T(t+1)\mathbf{y}_{1,j}(t) - \mu^{1-\gamma}c_1\Bigg[|\mathcal{B}_j|\mathbf{y}_{1,j}(t) + \frac{\mu^{\gamma/2}}{2c_2}\mathbf{y}_{2,j}(t)\\ &\quad - \sum_{b\in\mathcal{B}_j}\left(\sum_{j'\in\mathcal{N}_b}\frac{\mathbf{y}_{1,j'}(t)}{|\mathcal{N}_b|}\right) - \left(\bar{\boldsymbol{\eta}}_j(t) - \bar{\boldsymbol{\eta}}_j(t-1)\right) - \left(\boldsymbol{\eta}_j^\alpha(t) - \boldsymbol{\eta}_j^\beta(t-1)\right)\Bigg]\\ &\quad + 2\mu\mathbf{h}_j(t+1)\epsilon_j(t+1)\end{aligned}$$

where the last equality follows after adding and subtracting $|\mathcal{B}_j|\mathbf{s}_0$ from the quantity inside the square brackets in the right hand side of the first equality. □

2.5.3 Proof of Equation (2.29)

Consider first the noise vectors in (2.29). Stacking the channel noise terms from (2.26) and (2.27) and scaling with $\mu^{1-\gamma}c_1$ and $\mu^{-\gamma/2}c_2$, respectively, yields the first noise term in (2.29). Likewise, stacking the noise terms $2\mu\mathbf{h}_j(t+1)\epsilon_j(t+1)$ in (2.26) for $j = 1, \ldots, J$ yields the second noise term in (2.29) corresponding to the observation noise. Note that (2.27) contains no observation noise, which explains the zero vector at the lower part of the second noise term in (2.29).

The second term within the square brackets in (2.26) explains why $[\mathbf{\Psi}(t,\mu)]_{12} = -(\mu^{1-\gamma/2}c_1/2c_2)\mathbf{I}_{Jp}$. To specify the structure of $[\mathbf{\Psi}(t,\mu)]_{11}$, notice that

$$\sum_{b\in\mathcal{B}_j} |\mathcal{N}_b|^{-1} \sum_{j'\in\mathcal{N}_b} \mathbf{y}_{1,j'}(t) = \sum_{b\in\mathcal{B}} \mathbf{e}_b(j)|\mathcal{N}_b|^{-1}(\mathbf{e}_b \otimes \mathbf{I}_p)^T\mathbf{y}_1(t). \tag{2.56}$$

The supervector formed by concatenating the first term within the square brackets in (2.26), for $j = 1, \ldots, J$ can be written as

$$\sum_{b\in\mathcal{B}} \frac{(\mathbf{e}_b \otimes \mathbf{I}_p)(\mathbf{e}_b \otimes \mathbf{I}_p)^T}{|\mathcal{N}_b|}\mathbf{y}_1(t). \tag{2.57}$$

Stack the first term in (2.26) for $j = 1, \ldots, J$ and add the resulting supervector to the one in (2.57), to obtain

$$\left(\mathbf{I}_{Jp} - 2\mu\mathbf{R}_h(t+1) - \mu^{1-\gamma}c_1 \left[\text{bdiag}(|\mathcal{B}_1|\mathbf{I}_p, \ldots, |\mathcal{B}_J|\mathbf{I}_p) - \sum_{b\in\mathcal{B}} \frac{(\mathbf{e}_b \otimes \mathbf{I}_p)(\mathbf{e}_b \otimes \mathbf{I}_p)^T}{|\mathcal{N}_b|}\right]\right)\mathbf{y}_1(t)$$
$$= (\mathbf{I}_{Jp} - 2\mu\mathbf{R}_h(t+1) - \mu^{1-\gamma}c_1\mathbf{A})\mathbf{y}_1(t)$$

from which we can readily deduce that $[\mathbf{\Psi}(t,\mu)]_{11}$ is equal to the matrix multiplying $\mathbf{y}_1(t)$. Also, it follows immediately from the first term in (2.27) that $[\mathbf{\Psi}(t,\mu)]_{22} = \mathbf{I}_{Jp}$. Further, note that the second term within the square brackets in (2.27) has the same structure as the second term in (2.26). Thus, after: (i) stacking the first and second terms within the square brackets in (2.27), scaling them with $\mu^{-\gamma/2}c_2$ and subtracting them; and (ii) using (2.56) and (2.57), we obtain $[\mathbf{\Psi}(t,\mu)]_{21} = \mu^{-\gamma/2}c_2\mathbf{A}$. □

2.5.4 Proof of Lemma 2.3

We will argue by induction. For $t = 0$ we have from (2.33) that $\mathbf{z}(1) = \mathbf{\Phi}(1,\mu)\mathbf{z}(0) + [\boldsymbol{\epsilon}^T(1)\ \ \mathbf{0}^T]^T$, where $\mathbf{z}(0) := [\mathbf{y}_1^T(0)\ \ (\mathbf{y}_2'(0))^T]^T$; and after substituting $\mathbf{z}(1)$ into (2.32), we find

$$\mathbf{y}(1) = \text{bdiag}(\mathbf{I}_{Jp}, \mu^{-\gamma/2}c_2\mathbf{A})\mathbf{\Phi}(1,\mu)\mathbf{z}(0) + \begin{bmatrix} \mu^{1-\gamma}c_1\mathbf{I}_{Jp} \\ -\mu^{-\gamma/2}c_2\mathbf{I}_{Jp} \end{bmatrix}(\bar{\boldsymbol{\eta}}(0) + \boldsymbol{\eta}^{\alpha}(0)) + \begin{bmatrix} \boldsymbol{\epsilon}(1) \\ \mathbf{0} \end{bmatrix}. \tag{2.58}$$

Note that $\text{bdiag}(\mathbf{I}_{Jp}, \mu^{-\gamma/2}c_2\mathbf{A})\mathbf{\Phi}(t,\mu) = \mathbf{\Psi}(t,\mu)\text{bdiag}(\mathbf{I}_{Jp}, \mu^{-\gamma/2}c_2\mathbf{A})$ for $t = 1, 2, \ldots$; and $\mathbf{y}(0) = \text{bdiag}(\mathbf{I}_{Jp}, \mu^{-\gamma/2}c_2\mathbf{A})\mathbf{z}(0)$. Thus, the right hand side of (2.58) is equal to the right hand side of (2.29) for $t = 0$.

Suppose next that (2.32) and (2.33) hold true for $\mathbf{y}(t)$ and $\mathbf{z}(t)$. The same will be shown for $\mathbf{y}(t+1)$ and $\mathbf{z}(t+1)$. To this end, replace $\mathbf{y}(t)$ with the right hand side of (2.32) evaluated at time instant t, into (2.29) to obtain

$$\begin{aligned}\mathbf{y}(t+1) = {}& \text{bdiag}(\mathbf{I}, \mu^{-\gamma/2}c_2\mathbf{A})\mathbf{\Phi}(t+1,\mu)\mathbf{z}(t) + \begin{bmatrix} \mu^{1-\gamma}c_1\mathbf{I}_{Jp} \\ -\mu^{-\gamma/2}c_2\mathbf{I}_{Jp} \end{bmatrix}(\bar{\boldsymbol{\eta}}(t) + \boldsymbol{\eta}^{\alpha}(t)) \\ & + (\mathbf{\Psi}(t+1,\mu) - \mathbf{I}_{Jp})\begin{bmatrix} \mu^{1-\gamma}c_1\mathbf{I}_{Jp} \\ -\mu^{-\gamma/2}c_2\mathbf{I}_{Jp} \end{bmatrix}\bar{\boldsymbol{\eta}}(t-1) \qquad (2.59) \\ & + \mathbf{\Psi}(t+1,\mu)\begin{bmatrix} \mu^{1-\gamma}c_1\mathbf{I}_{Jp} \\ -\mu^{-\gamma/2}c_2\mathbf{I}_{Jp} \end{bmatrix}\mathbf{P}_{\alpha}\boldsymbol{\eta}(t-1) - \begin{bmatrix} \mu^{1-\gamma}c_1\mathbf{I}_{Jp} \\ -\mu^{-\gamma/2}c_2\mathbf{I}_{Jp} \end{bmatrix}\mathbf{P}_{\beta}\boldsymbol{\eta}(t-1) \\ & \qquad (2.60) \\ & + \begin{bmatrix} \boldsymbol{\epsilon}(t+1) \\ \mathbf{0} \end{bmatrix} \qquad (2.61)\end{aligned}$$

where $\mathbf{P}_{\alpha}$ and $\mathbf{P}_{\beta}$ are defined as $\mathbf{P}_{\alpha} := [\mathbf{p}_1, \ldots, \mathbf{p}_J]^T$, $\mathbf{P}_{\beta} := [\mathbf{p}_1', \ldots, \mathbf{p}_J']^T$; and the $p(\sum_{b\in\mathcal{B}} |\mathcal{N}_b|) \times p$ submatrices $\mathbf{p}_j$ and $\mathbf{p}_j'$ are given by $(\mathbf{p}_j)^T := [(\mathbf{p}_{j,1})^T, \ldots, (\mathbf{p}_{j,|\mathcal{B}|})^T]$,

$(\mathbf{p}'_j)^T := [(\mathbf{p}'_{j,1})^T, \ldots, (\mathbf{p}'_{j,|\mathcal{B}|})^T]$ with $\mathbf{p}_{j,r}, \mathbf{p}_{j',r}$ defined for $r = 1, \ldots, |\mathcal{B}|$ as

$$\mathbf{p}_{j,r}^T = \begin{cases} \mathbf{b}^T_{|\mathcal{N}_{b_r}|,r(j)} \otimes \mathbf{I}_p & \text{if } j \in \mathcal{N}_{b_r} \\ \mathbf{0}_{p \times |\mathcal{N}_{b_r}|p} & \text{if } j \notin \mathcal{N}_{b_r} \end{cases}, \quad (\mathbf{p}'_{j,r})^T = \begin{cases} |\mathcal{N}_{b_r}|^{-1}\mathbf{1}_{1\times|\mathcal{N}_{b_r}|} \otimes \mathbf{I}_p & \text{if } j \in \mathcal{N}_{b_r} \\ \mathbf{0}_{p \times |\mathcal{N}_{b_r}|p} & \text{if } j \notin \mathcal{N}_{b_r} \end{cases}.$$

Note that $r(j) = 1, \ldots, |\mathcal{N}_{b_r}|$ denotes the order in which $\boldsymbol{\eta}_j^{b_r}(t)$ appears in $\{\boldsymbol{\eta}_{j'}^{b_r}(t)\}_{j' \in \mathcal{N}_{b_r}}$.

Coming back to prove that $\mathbf{y}(t+1)$ is updated according to Lemma 2.3, observe that the noise term in (2.59) can be easily written as $\text{bdiag}(\mathbf{I}_{Jp}, \mu^{-\gamma/2}c_2\mathbf{A})\mathbf{R}_h^\alpha(t+1)\bar{\boldsymbol{\eta}}(t-1)$. Then, after algebraic manipulations the noise terms in (2.60) can be expressed as $\check{\mathbf{R}}_h^\beta(t+1)\boldsymbol{\eta}(t-1)$, where the $2Jp \times p(\sum_{b\in\mathcal{B}}|\mathcal{N}_b|)$ matrix $\check{\mathbf{R}}_h^\beta(t+1) := [[\check{\mathbf{R}}_h^\beta(t+1)]_{11}^T \quad [\check{\mathbf{R}}_h^\beta(t+1)]_{21}^T]^T$ is given by

$$[\check{\mathbf{R}}_h^\beta(t+1)]_{11} := \frac{3\mu^{1-\gamma}c_1}{2}\mathbf{P}_\alpha - (\mu^{1-\gamma}c_1)^2\mathbf{A}\mathbf{P}_\alpha - 2\mu^{2-\gamma}c_1\mathbf{R}_h(t+1)\mathbf{P}_\alpha - \mu^{1-\gamma}c_1\mathbf{P}_\beta$$

$$[\check{\mathbf{R}}_h^\beta(t+1)]_{21} := \mu^{1-3\gamma/2}c_1c_2\mathbf{A}\mathbf{P}_\alpha + \mu^{-\gamma/2}c_2(\mathbf{P}_\beta - \mathbf{P}_\alpha).$$

The remaining step is to show that $\check{\mathbf{R}}_h^\beta(t+1) = \text{bdiag}(\mathbf{I}_{Jp}, \mu^{-\gamma/2}c_2\mathbf{A})\mathbf{R}_h^\beta(t+1)$. If the latter holds, then: (i) we group the noise terms in (2.59), (2.60) and (2.61); (ii) take out $\text{bdiag}(\mathbf{I}_{Jp}, \mu^{-\gamma/2}c_2\mathbf{A})$ as a common factor; and (iii) conclude that $\mathbf{y}(t+1)$ is given by (2.32), while $\mathbf{z}(t+1)$ is provided by (2.33).

To show that $\check{\mathbf{R}}_h^\beta(t+1) = \text{bdiag}(\mathbf{I}_{Jp}, \mu^{-\gamma/2}c_2\mathbf{A})\mathbf{R}_h^\beta(t+1)$, it suffices to prove that there exists matrix $\mathbf{C}_R$ such that $\mathbf{A}\mathbf{C}_R = \mathbf{P}_\beta - \mathbf{P}_\alpha$. To this end, it can be shown that (details omitted due to space limitations) $\text{nullspace}(\mathbf{P}_\beta^T - \mathbf{P}_\alpha^T) = \text{nullspace}(\mathbf{A}) = \text{span}(\bar{\boldsymbol{\nu}}_i)$, where $\bar{\boldsymbol{\nu}}_i := [\boldsymbol{\nu}_{p,i}^T \ldots \boldsymbol{\nu}_{p,i}^T]^T$, $i = 1, \ldots, p$. Since $\mathbf{A}$ is symmetric, we have $\text{nullspace}(\mathbf{A}) \perp \text{range}(\mathbf{A})$. As $\text{nullspace}(\mathbf{P}_\beta^T - \mathbf{P}_\alpha^T) \perp \text{range}(\mathbf{P}_\beta - \mathbf{P}_\alpha)$, it follows that $\text{range}(\mathbf{P}_\beta - \mathbf{P}_\alpha) \subseteq \text{range}(\mathbf{A})$, which further implies that we can find $\mathbf{C}_R$ such that $\mathbf{A}\mathbf{C}_R = \mathbf{P}_\beta - \mathbf{P}_\alpha$. □

2.5.5 Proof of Lemma 2.4

We will specify first the step-size values for which $\boldsymbol{\Phi}(\mu) := E_{\mathbf{h}}[\boldsymbol{\Phi}(t+1, \mu)]$ has its eigenvalues inside the unit circle.

Lemma 2.6 *If c_1 is selected such that $c_1\lambda^+(\mathbf{A}) > 4$ and $\mu \in (0, \mu_u)$, where μ_u is defined in (2.38), then all the eigenvalues of $\mathbf{\Phi}(\mu)$ lie inside the unit circle; i.e., $|\lambda_i(\mathbf{\Phi}(\mu))| < 1$ for $i = 1, \ldots, 2Jp$.*

Proof: Following steps similar to those in [49, Appendix H], we express the eigenvalues as roots of a second-order polynomial to determine bounds on μ that ensure $|\lambda_i(\mathbf{\Phi}(\mu))| < 1$. Further, the spectral radius of $\lambda_{\max}(\mu) := \lambda_{\max}(\mathbf{\Phi}(\mu))$, can be expressed as $\lambda_{\max}(\mu) = 1 - \mu^{1-\gamma} b_1 + \mu b_2 < 1$, where b_1, b_2 are constants with $b_1 > 0$. □

Using Lemma 2.6, we can apply the results in [52, Section C6, pg. 321] to infer that for $\mu \in [0, \mu_u)$ there exists a finite constant $0 < \kappa(\mu)$, such that

$$\|\mathbf{\Phi}_1^t(\mu)\| \leq \kappa(\mu)\lambda_{\max}^t(\mu), \tag{2.62}$$

where $\mathbf{\Phi}_1(\mu)$ and $\mathbf{\Phi}_2(\mu)$ denote the $Jp \times 2Jp$ upper and lower block matrices of $\mathbf{\Phi}(\mu)$ obtained by keeping the upper Jp or lower Jp rows of $\mathbf{\Phi}(\mu)$, respectively.

In order to upper bound $\|\mathbf{\Phi}(1:t,\mu)\|$ in (2.39) we will establish a recursive inequality for $\|\mathbf{\Phi}(1:t,\mu)\|$ and then apply the discrete Bellman-Gronwall lemma [52, pg. 315]. To this end, rewrite $\mathbf{\Phi}(t,\mu)$ as $\mathbf{\Phi}(t,\mu) = \mathbf{\Phi}(\mu) + \tilde{\mathbf{\Phi}}(t,\mu)$, where $\tilde{\mathbf{\Phi}}(t,\mu) := \text{bdiag}(2\mu(\mathbf{R}_h - \mathbf{R}_h(t)), \mathbf{0}_{Jp})$. Then,

$$\begin{aligned}
\mathbf{\Phi}(1:t,\mu) &= \mathbf{\Phi}(t,\mu)\mathbf{\Phi}(1:t-1,\mu) \\
&= \mathbf{\Phi}(\mu)\mathbf{\Phi}(1:t-1,\mu) + \tilde{\mathbf{\Phi}}(t,\mu)\mathbf{\Phi}(1:t-1,\mu) \\
&= \mathbf{\Phi}^t(\mu) + \sum_{\tau=1}^{t} \mathbf{\Phi}^{t-\tau}(\mu)\tilde{\mathbf{\Phi}}(\tau,\mu)\mathbf{\Phi}(1:\tau-1,\mu)
\end{aligned} \tag{2.63}$$

where $\mathbf{\Phi}(1:0,\mu) = \mathbf{I}_{Jp}$. Next, let $\mathbf{\Phi}_1(1:t,\mu)$ and $\mathbf{\Phi}_2(1:t,\mu)$ denote the $Jp \times 2Jp$ upper and lower block matrices of $\mathbf{\Phi}(1:t,\mu)$, respectively. Taking norm on both sides of (2.63) leads to the recursive inequality

$$\|\mathbf{\Phi}_1(1:t,\mu)\| \leq \|\mathbf{\Phi}_1^t(\mu)\| + \sum_{\tau=1}^{t} \|\mathbf{\Phi}_1^{t-\tau}(\mu)\|\|\tilde{\mathbf{\Phi}}(\tau,\mu)\|\|\mathbf{\Phi}_1(1:\tau-1,\mu)\| \tag{2.64}$$

$$\leq \kappa(\mu)\lambda_{\max}^t(\mu) + \sum_{\tau=1}^{t} \kappa(\mu)\lambda_{\max}^{t-\tau}(\mu)\|\tilde{\mathbf{\Phi}}(\tau,\mu)\|\|\mathbf{\Phi}_1(1:\tau-1,\mu)\|, \tag{2.65}$$

where the second inequality is obtained after using (2.62). Then, multiplying both sides of (2.64) with $\lambda_{\max}^{-t}(\mu)$, and applying the the discrete Bellman-Gronwall lemma, leads to the following *non-recursive* inequality:

$$\begin{aligned}\|\mathbf{\Phi}_1(1:t,\mu)\| &\leq \kappa(\mu)\lambda_{\max}^t(\mu)\prod_{\tau=1}^{t}(1+\lambda_{\max}^{-1}(\mu)\kappa(\mu)\|\tilde{\mathbf{\Phi}}(\tau,\mu)\|)\\ &= \kappa(\mu)\prod_{\tau=1}^{t}(\lambda_{\max}(\mu)+\kappa(\mu)\|\tilde{\mathbf{\Phi}}(\tau,\mu)\|). \end{aligned} \tag{2.66}$$

Raising both sides of (2.66) to the power of $1/t$ and applying the arithmetic-mean geometric-mean inequality for the product term we arrive at

$$\|\mathbf{\Phi}_1(1:t,\mu)\|^{1/t} \leq \kappa^{1/t}(\mu)\left(\lambda_{\max}(\mu)+\mu\kappa(\mu)t^{-1}\sum_{\tau=1}^{t}\|\tilde{\mathbf{\Phi}}(\tau)\|\right), \tag{2.67}$$

where $\tilde{\mathbf{\Phi}}(\tau) = \text{bdiag}(2(\mathbf{R}_h - \mathbf{R}_h(\tau)), \mathbf{0}_{Jp})$.

Note at this point that $\lim_{t\to\infty}(\kappa(\mu))^{1/t} = 1$, while from (a3) the strong of large numbers implies that $\lim_{t\to\infty} t^{-1}\sum_{\tau=1}^{t}\|\tilde{\mathbf{\Phi}}(\tau)\| = E_{\mathbf{h}}[\|\tilde{\mathbf{\Phi}}(t)\|] = 2m_{\tilde{h}}$ exists a.s., and is bounded. The latter limits when combined with (2.67) give

$$\begin{aligned}\limsup_{t\to\infty}\|\mathbf{\Phi}_1(1:t,\mu)\|^{1/t} &\leq \lambda_{\max}(\mu)+2\mu\kappa(\mu)m_{\tilde{h}}\\ &= 1-\mu^{1-\gamma}[b_1-\mu^{\gamma}(b_2+2\kappa(\mu)m_{\tilde{h}})] := \lambda'(\mu),\end{aligned} \tag{2.68}$$

a.s.. Now, given that $b_1 > 0$ we can always find a $\mu_p > 0$ such that $b_1 - \mu^{\gamma}(b_2+2\kappa(\mu)m_{\tilde{h}}) > 0$ and $\lambda'(\mu) < 1$ for all $\mu \in (0, \mu_p]$. Thus, for $\mu \in (0, \mu_u')$ with $\mu_u' := \min(\mu_u, \mu_p)$ we ensure that (2.68) is satisfied, while $\lambda'(\mu) < 1$.

Next, we show that for $\mu \in (0, \mu_u')$ matrix $\mathbf{\Phi}_2(1:t,\mu)$ also satisfies an inequality of the form given in (2.68). Given that $\mu \in (0, \mu_u')$ it follows from (2.68) that there exists $t_0 < \infty$ and positive, a.s. finite random variable R such that

$$\|\mathbf{\Phi}_1(1:t,\mu)\| \leq (\lambda'(\mu))^t R, \text{ for all } t \geq t_0. \tag{2.69}$$

Recall that for $\mu \in (0, \mu_u')$ matrix $\mathbf{\Phi}(\mu)$ is stable (cf. Lemma 6); thus, similarl to (2.62) we have $\|\mathbf{\Phi}_2^t(\mu)\| \leq \kappa'(\mu)\lambda_{\max}^t(\mu)$, where $\kappa'(\mu)$ is a positive constant.

Focusing on the lower part $\mathbf{\Phi}_2(1:t,\mu)$ and taking norms in (2.63) we obtain

$$\begin{aligned}\|\mathbf{\Phi}_2(1:t,\mu)\| &\le \kappa'(\mu)\lambda_{\max}^t(\mu)+S(1:t;t)\\ &=\kappa'(\mu)\lambda_{\max}^t(\mu)+S(1:t_0;t)+S(t_0+1:t;t) \qquad (2.70)\end{aligned}$$

where $S(t_1:t_2;t) := \sum_{\tau=t_1}^{t_2}\kappa'(\mu)\lambda_{\max}^{t-\tau}(\mu)\|\tilde{\mathbf{\Phi}}(\tau,\mu)\|\|\mathbf{\Phi}_1(1:\tau-1,\mu)\|$. Next, we provide bounds for the terms in (2.70) and examine how they behave as $t\to\infty$. To this end, let $\lambda(\mu) := \max(\lambda_{\max}(\mu),\lambda'(\mu))$, and use (2.69) to upper bound the third term in (2.70) as

$$\begin{aligned}S(t_0+1:t;t) &\le \kappa'(\mu)\sum_{\tau=t_0+1}^{t}\lambda^{t-\tau}(\mu)\|\tilde{\mathbf{\Phi}}(\tau,\mu)\|\lambda^{\tau-1}(\mu)R \qquad (2.71)\\ &\le \kappa'(\mu)R\lambda^{t-1}(\mu)(t-t_0)\left[(t-t_0)^{-1}\sum_{\tau=t_0+1}^{t}\|\tilde{\mathbf{\Phi}}(\tau,\mu)\|\right]. \qquad (2.72)\end{aligned}$$

The quantity inside the square brackets in (2.72) converges a.s. to $2\mu m_{\tilde{h}}<\infty$. Further, $R<\infty$ a.s. and $\lim_{t\to\infty}\lambda^{t-1}(\mu)(t-t_0)=0$ since $\lambda(\mu)<1$. Given these properties, it follows readily that $S(t_0+1:t;t)$ converges to zero a.s. as $t\to\infty$. The second term in (2.70) can be rewritten as

$$\begin{aligned}S(1:t_0;t) &= \kappa'(\mu)\lambda_{\max}^t(\mu)\left[\sum_{\tau=1}^{t_0}\lambda_{\max}^{-\tau}(\mu)\|\tilde{\mathbf{\Phi}}(\tau,\mu)\|\|\mathbf{\Phi}_1(1:\tau-1,\mu)\|\right]\\ &=\kappa'(\mu)\lambda_{\max}^t(\mu)R' \le \kappa'(\mu)\lambda^t(\mu)R', \qquad (2.73)\end{aligned}$$

where R' is the quantity within the square brackets. Next, it suffices to show that R' is finite a.s. Since $t_0<\infty$ we have to show that each of the summands within R' is finite a.s. Observe first that $\lambda_{\max}^{-\tau}(\mu)<\infty$ for $\tau=1,\ldots,t_0$. Also, since $E[\|\tilde{\mathbf{\Phi}}(t,\mu)\|]<\infty$ [cf. (a3)], it follows that $\|\tilde{\mathbf{\Phi}}(t,\mu)\|<\infty$ a.s. Next, notice that $\|\mathbf{\Phi}(1:\tau,\mu)\|\le\prod_{i=1}^{\tau}\|\mathbf{\Phi}(i,\mu)\|$ and recall that $E[\mathbf{h}_j(t)\mathbf{h}_j^T(t)]<\infty$ [cf. (a3)] which further implies that $\|\mathbf{R}_h(t)\|<\infty$ a.s. Thus, $\|\mathbf{\Phi}(i,\mu)\|<\infty$ a.s. and consequently $\|\mathbf{\Phi}(1:\tau,\mu)\|\le\infty$ a.s., for $\tau=1,\ldots,t_0$. Using the previous bounds we can upper bound the right hand side in (2.70). Then, raising this upper bound and the left hand side of (2.70) to the power of $1/t$ we have

$$\|\mathbf{\Phi}_2(1:t,\mu)\|^{1/t}\le\left(\kappa'(\mu)+R'\kappa'(\mu)+R\kappa'(\mu)\lambda^{-1}(\mu)(t-t_0)\left[\sum_{\tau=t_0+1}^{t}\frac{\|\tilde{\mathbf{\Phi}}(\tau,\mu)\|}{(t-t_0)}\right]\right)^{1/t}\lambda(\mu).$$

Since $R, R' < \infty$ a.s., while the sample average within the square brackets converges to $2\mu m_{\tilde{h}}$ a.s., it follows readily that the quantity multiplying $\lambda(\mu)$ converges to one as $t \to \infty$. Thus, $\limsup_{t\to\infty} \|\mathbf{\Phi}_2(1:t,\mu)\|^{1/t} \leq \lambda(\mu)$ a.s., while $\lambda(\mu) < 1$ for all $\mu \in (0, \mu'_u)$. Combining the latter result with the one in (2.68), we deduce that $\limsup_{t\to\infty} \|\mathbf{\Phi}(1:t,\mu)\|^{1/t} \leq \lambda(\mu) < 1$, and consequently $\limsup_{t\to\infty} \log\left(\|\mathbf{\Phi}(1:t,\mu)\|^{1/t}\right) \leq \log(\lambda(\mu)) < 0$, a.s. □

2.5.6 Proof of Lemma 2.5

The proof follows readily from the result in [51, Section VI, (A13)]. Specifically, (a3) implies that: (i) $\mathbf{\Phi}(t,\mu)$ is stationary and ergodic; (ii) it holds that [cf. (a3)-(a4)]

$$E[\|\mathbf{R}_h^{\alpha}(t+1)\bar{\boldsymbol{\eta}}(t-1) + \mathbf{R}_h^{\beta}(t+1)\boldsymbol{\eta}(t-1) + [\boldsymbol{\epsilon}^T(t+1) \quad \mathbf{0}^T]^T\|] < \infty$$

since $E[\|\mathbf{R}_h^{\alpha}(t)\|] < \infty$, $E[\|\mathbf{R}_h^{\beta}(t)\|] < \infty$ and $E[\|\bar{\boldsymbol{\eta}}_j(t)\|] < \infty$; and (iii) Lemma 2.4 shows that $\limsup_{t\to\infty} t^{-1}\log(\|\mathbf{\Phi}(1:t,\mu)\|) \leq \log(\lambda(\mu)) < 0$. Conditions (i)-(iii) guarantee that $\mathbf{z}(t)$ is weakly stochastically bounded [51, Section VI]. □

2.5.7 Proof of Proposition 2.1

Taking norms on both sides of (2.32) yields

$$\begin{aligned}\|\mathbf{y}(t+1)\| \leq \|\text{bdiag}(\mathbf{I}_{Jp}, \mu^{-\gamma/2}c_2\mathbf{A})\|\|\mathbf{z}(t+1)\| + \|[\mu^{1-\gamma}c_1\mathbf{I}_{Jp}^T \ - \mu^{-\gamma/2}c_2\mathbf{I}_{Jp}^T]^T\|\|\bar{\boldsymbol{\eta}}(t)\| \\ + \|[\mu^{1-\gamma}c_1\mathbf{I}_{Jp}^T \ - \mu^{-\gamma/2}c_2\mathbf{I}_{Jp}^T]^T\|\|\boldsymbol{\eta}^{\alpha}(t)\|. \qquad (2.74)\end{aligned}$$

For brevity, let $\xi(t)$ denote the sum of the last two terms in (2.74). Now recall that if Y_1, Y_2 are random variables and $Y_1 \leq Y_2$, then $\Pr[Y_1 \geq \delta] \leq \Pr[Y_2 \geq \delta]$; hence

$$\Pr[\|\mathbf{y}(t+1)\| \geq \delta] \leq \Pr[\|\text{bdiag}(\mathbf{I}_{Jp}, \mu^{-\gamma/2}c_2\mathbf{A})\|\|\mathbf{z}(t+1)\| + \xi(t) \geq \delta]. \qquad (2.75)$$

Another property needed in the remainder of the proof is that if Y_1, Y_2 are positive random variables, then $\Pr[Y_1 + Y_2 \geq \delta] \leq \Pr[Y_1 \geq \delta/2] + \Pr[Y_2 \geq \delta/2]$. Applying this property to the right hand side of (2.75) yields

$$\Pr[\|\mathbf{y}(t+1)\| \geq \delta] \leq \Pr[\|\text{bdiag}(\mathbf{I}_{Jp}, \mu^{-\gamma/2}c_2\mathbf{A})\|\|\mathbf{z}(t+1)\| \geq \delta/2] + \Pr[\xi(t) \geq \delta/2].$$

Markov's inequality can now be used to obtain the upper bound $\Pr[\xi(t) \geq \delta/2] \leq 2\delta^{-1}E[\xi(t)]$. Boundedness of $E[\|\mathbf{R}_h^{\alpha}(t)\|]$, $E[\|\mathbf{R}_h^{\beta}(t)\|]$, $E[\|\bar{\boldsymbol{\eta}}(t)\|]$ and $\mu^{-\gamma/2}$, since $\mu \in (0, \mu_u')$ along with (a4) imply that $2E[\xi(t)] = c_\xi < \infty$. Thus,

$$\sup_{t\geq 0} \Pr[\|\mathbf{y}(t+1)\| \geq \delta] \leq \sup_{t\geq 0} \Pr[\|\mathbf{z}(t+1)\| \geq (\delta/2)\|\mathrm{bdiag}(\mathbf{I}_{Jp}, \mu^{-\gamma/2}c_2\mathbf{A})\|^{-1}] + c_\xi\delta^{-1}. \quad (2.76)$$

But from Lemma 2.5 we have that $\lim_{\delta\to\infty}\sup_{t\geq 0}\Pr[\|\mathbf{z}(t+1)\| \geq \delta/2\|\mathrm{bdiag}(\mathbf{I}_{Jp}, \mu^{-\gamma/2}c_2\mathbf{A})\|^{-1}] = 0$ for any $\mu \in (0, \mu_u')$. Hence, letting $\delta \to \infty$ on both sides of (2.76) we find that $\mathbf{y}(t)$ is WSB; i.e.,

$$\lim_{\delta\to\infty}\sup_{t\geq 0}\Pr[\|\mathbf{y}(t+1)\| \geq \delta] = 0. \ \square$$

2.5.8 Proof of Proposition 2.2

Since $\mathbf{y}_1(t+1) - \bar{\mathbf{y}}_1(t+1) = (\mathbf{z}_1(t+1) - \bar{\mathbf{z}}_1(t+1))$, it suffices to show that

$$\lim_{\mu\to 0}\sup_{0\leq t\leq T/\mu^{1-\gamma-\beta}}\Pr[\|\bar{\mathbf{z}}_1(t) - \mathbf{z}_1(t)\| \geq \delta] = 0.$$

To this end, subtract $\bar{\mathbf{z}}_1(t+1)$ from $\mathbf{z}_1(t+1)$ and recursively substitute $\mathbf{z}_1(t) - \bar{\mathbf{z}}_1(t)$ to obtain

$$\begin{aligned}\mathbf{z}_1(t+1) - \bar{\mathbf{z}}_1(t+1) = {} & [[\boldsymbol{\Phi}(1:t+1,\mu)]_{11} \ \ \mathbf{0}](\mathbf{z}(0) - \bar{\mathbf{z}}(0)) \\ & + 2\mu\textstyle\sum_{\tau=0}^{t}[[\boldsymbol{\Phi}(\tau+2:t+1,\mu)]_{11} \ \ \mathbf{0}]\mathrm{diag}(\mathbf{R}_h(\tau+1) - \mathbf{R}_h, \mathbf{0})\bar{\mathbf{z}}(\tau) \\ & + 2c_1\mu^{2-\gamma}\textstyle\sum_{\tau=0}^{t}[\boldsymbol{\Phi}(\tau+2:t+1,\mu)]_{11}(\mathbf{R}_h - \mathbf{R}_h(\tau+1))\bar{\boldsymbol{\eta}}(\tau-1) \\ & + 2c_1\mu^{2-\gamma}\textstyle\sum_{\tau=0}^{t}[\boldsymbol{\Phi}(\tau+2:t+1,\mu)]_{11}(\mathbf{R}_h - \mathbf{R}_h(\tau+1))\mathbf{P}_\alpha\boldsymbol{\eta}(\tau-1) \quad (2.77)\end{aligned}$$

where $[\boldsymbol{\Phi}(\tau+2:t+1,\mu)]_{11}$ is the $Jp \times Jp$ upper left submatrix contained in $\boldsymbol{\Phi}(\tau+2:t+1,\mu)$. Now, let $T_1(0,t), T_2(0,t), T_3(0,t)$ denote the norm of each of the last three summands in (2.77), and $T(0,t) := \sum_{i=1}^{3}T_i(0,t)$. Since $\mathbf{z}(0) = \bar{\mathbf{z}}(0)$, it holds that $\|\mathbf{z}_1(t+1) - \bar{\mathbf{z}}_1(t+1)\| \leq T(0, t-t_0) + T(t-t_0+1, t)$, from which it follows that

$$\Pr[\|\mathbf{z}_1(t+1) - \bar{\mathbf{z}}_1(t+1)\| \geq \delta] \leq \Pr[T(0,t-t_0) \geq \delta/2] + \Pr[T(t-t_0+1,t) \geq \delta/2] \quad (2.78)$$

where $t_0 < \infty$ will be selected appropriately later on.

Thus, it suffices to prove that the two terms in (2.78) converge to zero as $\mu \to 0$. Toward this objective, the first term in (2.78) can be upper bounded as

$$\Pr[T(0,t-t_0) \geq \delta/2] \leq \Pr[T_1(0,t-t_0) \geq \delta/4]+\Pr[T_2(0,t-t_0) \geq \delta/8]+\Pr[T_3(0,t-t_0) \geq \delta/8]. \tag{2.79}$$

In order to show that the right hand side in (2.79) goes to zero, recall from (2.62) that $\|[\mathbf{\Phi}^t(\mu)]_{11}\| \leq \kappa(\mu)\lambda^t_{\max}(\mu)$, where $\lambda_{\max}(\mu) = 1 - \mu^{1-\gamma}b_1 + \mu b_2$, with $b_1 > 0$ and $\kappa(\mu)$ finite positive constant. Note that (a3) ensures that there exists finite t_0' such that $|(t_0')^{-1}\sum_{m=1}^{t_0'}\|\tilde{\mathbf{\Phi}}(m)\| - E[\|\tilde{\mathbf{\Phi}}\|]| \leq \Delta$ a.s. Then, setting $t_0 = t_0'$ and for $t \geq t_0'$, it follows from (2.67) that

$$\begin{aligned}\|[\mathbf{\Phi}(1:t,\mu)]_{11}\| &\leq \kappa(\mu)\left[\lambda_{\max}(\mu) + \mu\kappa(\mu)(E[\|\tilde{\mathbf{\Phi}}(t)\|] + \kappa_\Delta)\right]^t \\ &\leq \kappa(\mu)\Big(1-\mu^{1-\gamma}(b_1-\mu^\gamma(b_2+\kappa(\mu)E[\|\tilde{\mathbf{\Phi}}\|]+\kappa(\mu)\kappa_\Delta))\Big)^t,\end{aligned} \tag{2.80}$$

where $\kappa_\Delta \in [-\Delta,\Delta]$. We now contend that $\sup_{0\leq t\leq T/\mu^{1-\gamma-\beta}}\Pr[T_1(0,t-t_0) \geq \delta/4] \to 0$ as $\mu \to 0$. Indeed, $T_1(0,t-t_0)$ can be bounded as [cf. (2.45)-(46)]

$$\begin{aligned}T_1(0,t-t_0) \leq 2\mu\sum_{\tau=0}^{t-t_0}&\|[\mathbf{\Phi}(\tau+2:t+1,\mu)]_{11}\|\|\tilde{\mathbf{\Phi}}(\tau+1)\|\|[[\mathbf{\Phi}^\tau(\mu)]_{11}\ [\mathbf{\Phi}^\tau(\mu)]_{12}]\|\|\bar{\mathbf{z}}(0)\| \\ &+2\mu^{1+\beta}\sum_{\tau=0}^{t-t_0}\|[\mathbf{\Phi}(\tau+2:t+1,\mu)]_{11}\|\|\tilde{\mathbf{\Phi}}(\tau+1)\|\left(\|\boldsymbol{\zeta}_1(\tau)\|+\|\boldsymbol{\zeta}_2(\tau)\|+\|\boldsymbol{\zeta}_3(\tau)\|\right)\end{aligned} \tag{2.81}$$

where

$$\begin{aligned}\boldsymbol{\zeta}_1(\tau) &:= \mu^{-\beta}\sum_{m=0}^{\tau-1}[[\mathbf{\Phi}^m(\mu)]_{11}\ [\mathbf{\Phi}^m(\mu)]_{12}]\mathbf{R}_h^\alpha\bar{\boldsymbol{\eta}}(\tau-m-2) \\ \boldsymbol{\zeta}_2(\tau) &:= \mu^{-\beta}\sum_{m=0}^{\tau-1}[[\mathbf{\Phi}^m(\mu)]_{11}\ [\mathbf{\Phi}^m(\mu)]_{12}]\mathbf{R}_h^\beta\boldsymbol{\eta}(\tau-m-2) \\ \boldsymbol{\zeta}_3(\tau) &:= \mu^{-\beta}\sum_{m=0}^{\tau-1}[\mathbf{\Phi}^m(\mu)]_{11}\boldsymbol{\epsilon}(\tau-m).\end{aligned} \tag{2.82}$$

Next, recall that $\|[\mathbf{\Phi}^m(\mu)]_{1i}\| \leq \kappa(\mu)\lambda^t_{\max}(\mu)$ with $\lambda_{\max}(\mu) = 1-\mu^{1-\gamma}b_1+\mu b_2$ and $\kappa(\mu) > 0$. Upon considering the expected norms of the noise terms in (2.82) and using (a4), we arrive after tedious but straightforward manipulations at

$$\sup_{0\le\tau\le T/\mu^{1-\gamma-\beta}} E[\|\tilde{\mathbf{\Phi}}(\tau+1)\|\|\boldsymbol{\zeta}_i(\tau)\|] = m_{\zeta_i} < \infty, \;\; i=1,2,3. \tag{2.83}$$

Using the result in (2.83) in conjunction with (2.80) leads to

$$\lim_{\mu\to 0} \sup_{0\le t\le T/\mu^{1-\gamma-\beta}} E[T_1(0,t-t_0)] = 0. \tag{2.84}$$

Applying once more Markov's inequality for $\Pr(T_1(0,t-t_0)\ge\delta/4)$ yields

$$\lim_{\mu\to 0} \sup_{0\le t\le T/\mu^{1-\gamma-\beta}} \Pr[T_1(0,t-t_0)\ge\delta/4] = 0. \tag{2.85}$$

Using similar arguments we can bound the second and third probability terms in (2.79). Since $\sup_{0\le\tau\le T/\mu^{1-\gamma-\beta}} E[\|\tilde{\mathbf{\Phi}}(\tau+1)\|\|\bar{\boldsymbol{\eta}}_b(\tau-1)\|] < \infty$ and $\sup_{0\le\tau\le T/\mu^{1-\gamma-\beta}} E[\|\tilde{\mathbf{\Phi}}(\tau+1)\|\|\boldsymbol{\eta}(\tau-1)\|] < \infty$, use of (2.80) to bound $\|\mathbf{\Phi}(\tau+2:t+1,\mu)\|$, as well as Markov's inequality (as in $T_1(0,t-t_0)$) implies that $\lim_{\mu\to 0}\sup_{0\le t\le T/\mu^{1-\gamma-\beta}} \Pr[T_i(0,t-t_0)\ge\delta/8]=0$, for $i=2,3$. Combining these limits with the one in (2.85) establishes that

$$\lim_{\mu\to 0} \sup_{0\le t\le T/\mu^{1-\gamma-\beta}} \Pr[T(0,t-t_0)\ge\delta/2] = 0. \tag{2.86}$$

Consider next, the second probability term in (2.78). The three summands in (2.77) comprising $T_1(t-t_0+1,t)$ contain a finite number of terms, namely t_0. Hence Markov's inequality yields $\Pr[T(t-t_0+1,t)\ge\delta/2]\le\frac{2}{\delta}E[T(t-t_0+1,t)]$. Boundedness of the regressor moments further ensures that the expectation in the right hand side of the last inequality converges to zero as $\mu\to 0$. We have already shown that the supremum of the two probability terms in (2.78) goes to zero as $\mu\to 0$ over the time interval $[0,T/\mu^{1-\gamma-\beta}]$; thus, the supremum of the left hand side in (2.78) also goes to zero for vanishing μ. □

2.5.9 Proof of Equation (2.48)

The covariance in (2.48) follows readily from (2.46) after recalling that $\boldsymbol{\eta}^\alpha(t) := \mathbf{P}_\alpha\boldsymbol{\eta}(t)$. Similarly, it is possible to find the covariance matrix of $\bar{\mathbf{z}}(t)$ in (2.49) using the recursive formula for $\bar{\mathbf{z}}(t)$ in (2.45) and setting $\breve{\boldsymbol{\eta}}(t+1) := \mathbf{R}_h^\alpha\bar{\boldsymbol{\eta}}(t-1)+\mathbf{R}_h^\beta\boldsymbol{\eta}(t-1)+[\boldsymbol{\epsilon}^T(t+1)\;\;\mathbf{0}^T]^T$.

Thus, we can readily obtain $\mathbf{R}_{\breve{\eta}}$. Next, focus on the structure of $\mathbf{R}_{\bar{\eta}}$ and $\mathbf{R}_{\eta}$. From the definition in (2.24) it follows that $\mathbf{R}_{\bar{\eta}}$ consists of $p \times p$ submatrices of the form

$$[\mathbf{R}_{\bar{\eta}}]_{jj'} = \begin{cases} \displaystyle\sum_{b \in \mathcal{B}_j} \frac{\sum_{j'' \in \mathcal{N}_b \setminus \{b\}} \mathbf{R}_{\bar{\eta}_{b,j''}}}{|\mathcal{N}_b|^2} & \text{if } j' = j, \text{ and } j = 1, \ldots, J \\ \displaystyle\sum_{b \in \mathcal{B}_j \cap \mathcal{B}_{j'}} \frac{\sum_{j'' \in \mathcal{N}_b \setminus \{b\}} \mathbf{R}_{\bar{\eta}_{b,j''}}}{|\mathcal{N}_b|^2} & \text{if } j' \neq j, \text{ and } j, j' = 1, \ldots, J \end{cases}$$

where $\mathbf{R}_{\bar{\eta}_{b,j}} := E[\bar{\boldsymbol{\eta}}_b^j(t)(\bar{\boldsymbol{\eta}}_b^j(t))^T]$ denotes the covariance of the channel noise at bridge sensor b when receiving from sensor j. In the same way it follows from (2.31) that $\mathbf{R}_{\eta}$ is a block diagonal matrix with $(\sum_{b \in \mathcal{B}} |\mathcal{N}_b|)$ diagonal blocks of size $p \times p$. Each of these blocks is set equal to $\mathbf{R}_{\eta_{j,b}}$ of the corresponding channel noise vector $\boldsymbol{\eta}_j^b(t)$. Note also that $\mathbf{R}_{\eta_{j,j}} = \mathbf{0}$. □

Chapter 3

Tracking Performance Analysis of the Distributed LMS Algorithm

3.1 Introduction

This chapter deals with a fully-distributed version of the D-LMS algorithm, with major focus towards quantifying its tracking performance in nonstationary environments. Relative to the D-LMS variant in Chapter 2, the reformulation of the LMS cost introduced in this chapter circumvents the requirement of the special type of sensors comprising the bridge sensor subset. As a byproduct, this approach results in a fully distributed algorithm whereby all sensors perform identical tasks, without introducing hierarchies that may require intricate recovery protocols to cope with sensor failures. Utilization of a constant step-size endows D-LMS with tracking capabilities, without hurting its resilience to communication noise.

A main contribution of the present chapter pertains to a detailed mean-square error (MSE) *tracking performance analysis* for D-LMS (Section 3.5). For a *time-varying* parameter fluctuating as a first-order autoregressive [AR(1)] process, and sensor observations that are linearly related to it, the simplifying independence Gaussian setting assumptions [52, pg. 110], [43, pg. 448] are key enablers towards deriving exact closed-form expressions for the MSE evolution and its steady-state value (Section 3.5.2). Mean and MSE stability are also established, revealing sufficient conditions under which steady-state is attained. The AR(1) model subsumes a time-invariant parameter as a special case, and performance results for the stationary case are readily obtained as a byproduct. Hence, the estimation/tracking capabilities of D-LMS in the presence of: i) time-invariant; and ii) time-varying parameters can be contrasted in a unified fashion. Of particular interest in these two scenarios are the corresponding steady-state MSE versus step-size characteristics, which reveal fundamental insights and differences similar to those observed in the classic LMS algorithm (Section 3.5.3). All in all, the importance of the aforementioned results is threefold: i) an exact tracking MSE characterization is provided for D-LMS; ii) for the stationary case and ideal inter-sensor links, similar results for the diffusion LMS algorithm [28] lay a common ground for fair comparisons; and iii) for small step-sizes the conclusions extend to temporally correlated (non-)Gaussian data. Numerical tests corroborating the theoretical findings of this chapter are presented in Section 3.6.

3.2 Network Model and Estimation Problem Statement

Consider an ad hoc WSN with sensors $\{1, \ldots, J\} := \mathcal{J}$. To effect energy-aware communications, sensor j communicates only with its (nearby) single-hop neighbors in $\mathcal{N}_j \subseteq \mathcal{J}$. Under the natural assumption of symmetric inter-sensor links, the WSN is modeled as an undirected graph $G(\mathcal{V}, \mathcal{E})$. The vertices in $\mathcal{V}$ correspond to the sensors and hence are biunivocally mapped to the elements of $\mathcal{J}$, while the edges in $\mathcal{E}$ represent the available wireless links. Global connectivity information is captured by the symmetric adjacency matrix $\mathbf{E} \in \mathbb{R}^{J \times J}$, where $[\mathbf{E}]_{ij} = 1$ if $i \in \mathcal{N}_j$ and $[\mathbf{E}]_{ij} = 0$ otherwise. By convention $[\mathbf{E}]_{jj} = 1 \ \forall\, j \in \mathcal{J}$ so that G is not simple. The graph Laplacian $\mathbf{L} \in \mathbb{R}^{J \times J}$ will be useful in the sequel, where $\mathbf{L} := \mathbf{D} - \mathbf{E}$, and $\mathbf{D} := \text{diag}(|\mathcal{N}_1|, \ldots, |\mathcal{N}_J|)$. The adjustment of the sensors' transmission power as well as the initial WSN deployment are assumed to render G connected. This is a minimum requirement ensuring that sensors' data can percolate across the whole WSN. Different from [27, 28, 42], the present network model accounts explicitly for non-ideal sensor-to-sensor links, through a zero-mean additive noise vector $\boldsymbol{\eta}_j^i(t)$ with covariance matrix $\mathbf{R}_{\eta_{j,i}} := E[\boldsymbol{\eta}_j^i(t)\boldsymbol{\eta}_j^i(t)^T]$ corrupting signals received at sensor j from sensor i at discrete-time instant t. The noise vectors $\{\boldsymbol{\eta}_j^i(t)\}_{j\in\mathcal{J}}^{i\in\mathcal{N}_j}$ are assumed temporally and spatially uncorrelated.

The WSN is deployed to estimate a signal vector $\mathbf{s}_0(t) \in \mathbb{R}^{p\times 1}$ in a collaborative fashion subject to single-hop communication constraints, by resorting to the linear LMS criterion; see e.g. [43, pg. 171]. Per time instant $t = 0, 1, 2, \ldots$, each sensor has available a regression vector $\mathbf{h}_j(t) \in \mathbb{R}^{p\times 1}$ and a scalar observation $x_j(t)$, both assumed zero-mean without loss of generality. The network-wide snapshot of data at time instant t can be compactly represented in the global vector $\mathbf{x}(t) := [x_1(t) \ldots x_J(t)]^T \in \mathbb{R}^{p\times 1}$ and regression matrix $\mathbf{H}(t) := [\mathbf{h}_1(t) \ldots \mathbf{h}_J(t)]^T \in \mathbb{R}^{J\times p}$. A similar data setting was considered in [28]. The global LMS estimator of interest can be written as [28], [27], [48]

$$\hat{\mathbf{s}}(t) = \arg\min_{\mathbf{s}} E\left[\|\mathbf{x}(t) - \mathbf{H}(t)\mathbf{s}\|^2\right] = \arg\min_{\mathbf{s}} \sum_{j=1}^{J} E\left[(x_j(t) - \mathbf{h}_j^T(t)\mathbf{s})^2\right]. \tag{3.1}$$

Suitable reformulation may be needed in order to acquire $\{\mathbf{h}_j(t)\}_{j\in\mathcal{J}}$ based on the available information across sensors. There are no general guidelines to this end, which is dictated by the specific estimation/tracking problem at hand. For example, in target tracking applications where sensors rely on power or range measurements, the nonlinear data models must be linearized before obtaining regressors as a function of sensor observations; see, e.g., [2, pg. 137]. Another possibility is to obtain the regression vectors from the physics of the problem, using standard kinematic models; see, e.g., [2, Ch. 6]. A distributed power spectrum estimation problem was described in Remark 2.1 in Chapter 2, where regressors are formed by stacking the last p local sensor observations.

Remark 3.1 (Motivation for adaptive processing) The gradient of the differentiable cost in (3.1) depends on the (cross-) covariances $\mathbf{R}_H(t) := E[\mathbf{H}^T(t)\mathbf{H}(t)]$ and $\mathbf{r}_{Hx}(t) := E[\mathbf{H}^T(t)\mathbf{x}(t)]$. Any attempt to iteratively obtain $\hat{\mathbf{s}}(t)$ via gradient-based optimization algorithms will require knowledge of $\mathbf{R}_H(t)$ and $\mathbf{r}_{Hx}(t)$. In many WSN applications however, this information may be either unavailable or time-varying, and thus impossible to acquire continuously. Tuned with the reduced-complexity requirements of WSNs, the arguably simplest approach involves approximating the expectations coarsely via instantaneous realizations of the sensor data as in the classical LMS, i.e., $\mathbf{R}_H(t) \approx \mathbf{H}^T(t)\mathbf{H}(t)$ and $\mathbf{r}_{Hx}(t) \approx \mathbf{H}^T(t)\mathbf{x}(t)$.

This exemplification of the adaptive step in turn leads to centralized stochastic (noisy) gradient iterations, which one expects to perform well after sufficient data are acquired and the unknown statistics are learnt. Still, the challenge is to enable such learning mechanisms even when data is not centrally available, i.e., when entries of $\mathbf{x}(t)$ and rows of $\mathbf{H}(t)$ are scattered across the WSN. In this context, the present chapter aims to develop a fully distributed LMS-type algorithm, which relies on in-network, adaptive processing of the available information across the WSN.

3.3 The D-LMS Algorithm

In this section we introduce the D-LMS algorithm, first going through the algorithmic construction steps and salient features of its operation. The approach followed includes three main building blocks: i) recast (3.1) into an equivalent separable form which facilitates distributed implementation; ii) split the optimization problem into simpler subtasks executed locally at each sensor; and iii) invoke a stochastic approximation iteration to obtain an adaptive LMS-type algorithm that can both handle the unavailability of (cross-) covariance information, and also remain robust to signal variations. Important differences with respect to the related approach in [48] are encountered in steps i) and ii); see also Remark 3.2 for a summary of the merits of the present contribution relative to [48].

To distribute the cost in (3.1), replace the global variable $\mathbf{s}$ which couples the per-sensor summands with auxiliary local variables $\boldsymbol{s} := \{\mathbf{s}_j\}_{j=1}^J$ representing candidate estimates of $\mathbf{s}$ per sensor. These local estimates are utilized to reformulate (3.1) as the following convex *constrained* minimization problem:

$$\{\hat{\mathbf{s}}_j(t)\}_{j=1}^J = \arg\min_{\boldsymbol{s}} \sum_{i=1}^J E\left[(x_i(t) - \mathbf{h}_i^T(t)\mathbf{s}_i)^2\right], \quad \text{s. t.} \quad \mathbf{s}_j = \mathbf{s}_{j'},\ j \in \mathcal{J},\ j' \in \mathcal{N}_j. \quad (3.2)$$

The set of equality constraints in (3.2) involves variables of neighboring sensors only, forcing an agreement across each sensor's neighborhood. If the WSN graph G is connected, these constraints impose *network-wide* consensus a fortiori, i.e., $\mathbf{s}_j = \mathbf{s}_j' \ \forall j, j' \in \mathcal{J}$. As an immediate consequence, one finds that the optimization problems (3.1) and (3.2) are equivalent in the sense that their optimal solutions coincide; i.e., $\hat{\mathbf{s}}_j(t) = \hat{\mathbf{s}}(t),\ \forall\, j \in \mathcal{J}$.

3.3.1 Algorithm Construction

In order to tackle (3.2) in a distributed fashion, we will resort to the AD-MoM [7, p. 253]. Towards this end, consider the auxiliary variables $\boldsymbol{z} := \{\mathbf{z}_j^{j'}\}_{j\in\mathcal{J}}^{j'\in\mathcal{N}_j}$ and replace the constraints in (3.2) with the equivalent ones

$$\mathbf{s}_j = \mathbf{z}_j^{j'} \text{ and } \mathbf{s}_{j'} = \mathbf{z}_j^{j'},\;\; j \in \mathcal{J},\;\; j' \in \mathcal{N}_j,\;\; j \neq j'. \tag{3.3}$$

The sole purpose of the variables $\boldsymbol{z}$ is to facilitate application of the AD-MoM, and they will be eventually eliminated due to their inherent redundancy. Next, associate Lagrange multipliers $[\boldsymbol{v}, \boldsymbol{u}] := \{\mathbf{v}_j^{j'}, \mathbf{u}_j^{j'}\}_{j\in\mathcal{J}}^{j'\in\mathcal{N}_j}$ with the constraints in (3.3), and form the quadratically augmented Lagrangian function

$$\begin{aligned}\mathcal{L}_a\left[\boldsymbol{s}, \boldsymbol{z}, \boldsymbol{v}, \boldsymbol{u}\right] = &\sum_{j=1}^{J} E\left[(x_j(t+1) - \mathbf{h}_j^T(t+1)\mathbf{s}_j)^2\right] \\ &+ \sum_{j=1}^{J}\sum_{j'\in\mathcal{N}_j}\left[(\mathbf{v}_j^{j'})^T(\mathbf{s}_j - \mathbf{z}_j^{j'}) + (\mathbf{u}_j^{j'})^T(\mathbf{s}_j - \mathbf{z}_{j'}^{j})\right] \\ &+ \sum_{j=1}^{J}\sum_{j'\in\mathcal{N}_j}\frac{c}{2}\left[\|\mathbf{s}_j - \mathbf{z}_j^{j'}\|^2 + \|\mathbf{s}_j - \mathbf{z}_{j'}^{j}\|^2\right]\end{aligned} \tag{3.4}$$

where $c > 0$ is a penalty coefficient. Sensor j will locally store and update a total of $3|\mathcal{N}_j| + 1$ vectors in $\mathbb{R}^{p\times 1}$, namely $\mathbf{s}_j$ and $\{\mathbf{z}_j^{j'}, \mathbf{v}_j^{j'}, \mathbf{u}_j^{j'}\}_{j'\in\mathcal{N}_j}$. We re-iterate however, that in the process of deriving the local updating recursions many of these variables will turn out to be redundant.

The AD-MoM entails an iterative process comprising three steps per time instant $t = 0, 1, 2, \ldots$

[S1] Multiplier updates:

$$\mathbf{v}_j^{j'}(t) = \mathbf{v}_j^{j'}(t-1) + c(\mathbf{s}_j(t) - \mathbf{z}_j^{j'}(t)), \quad j \in \mathcal{J},\; j' \in \mathcal{N}_j \tag{3.5}$$

$$\mathbf{u}_j^{j'}(t) = \mathbf{u}_j^{j'}(t-1) + c(\mathbf{s}_j(t) - \mathbf{z}_{j'}^{j}(t)), \quad j \in \mathcal{J},\; j' \in \mathcal{N}_j. \tag{3.6}$$

[S2] Local estimate updates:

$$\boldsymbol{s}(t+1) = \arg\min_{\boldsymbol{s}} \mathcal{L}_a\left[\boldsymbol{s}, \boldsymbol{z}(t), \boldsymbol{v}(t), \boldsymbol{u}(t)\right]. \tag{3.7}$$

[S3] **Auxiliary variable updates:**

$$\boldsymbol{z}(t+1) = \arg\min_{\boldsymbol{z}} \mathcal{L}_a\left[\boldsymbol{s}(t+1), \boldsymbol{z}, \boldsymbol{v}(t), \boldsymbol{u}(t)\right]. \tag{3.8}$$

The multiplier recursions in [S1] correspond to gradient ascent iterations seeking the optimal dual prices, and are customary in various methods of multipliers [7, Ch. 3]. On the other hand, [S2] and [S3] represent block coordinate descent steps which respectively update $\boldsymbol{s}$ and $\boldsymbol{z}$ in a cyclic fashion. At each step while minimizing the augmented Lagrangian, the variables not being updated are treated as fixed constants and substituted with their most up to date values. Interestingly, it is shown in Appendix A that [S1]-[S3] boil down to the following simple set of subtasks carried out locally at each sensor $j \in \mathcal{J}$,

$$\mathbf{v}_j^{j'}(t) = \mathbf{v}_j^{j'}(t-1) + \frac{c}{2}(\mathbf{s}_j(t) - \mathbf{s}_{j'}(t)), \quad j' \in \mathcal{N}_j, \tag{3.9}$$

$$\mathbf{s}_j(t+1) = \arg\min_{\mathbf{s}_j} \Bigg[E\left[(x_j(t+1) - \mathbf{h}_j^T(t+1)\mathbf{s}_j)^2\right] + \sum_{j' \in \mathcal{N}_j} [\mathbf{v}_j^{j'}(t) - \mathbf{v}_{j'}^{j}(t)]^T \mathbf{s}_j \\ + c \sum_{j' \in \mathcal{N}_j} \left\| \mathbf{s}_j - \frac{1}{2}(\mathbf{s}_j(t) + \mathbf{s}_{j'}(t)) \right\|^2 \Bigg]. \tag{3.10}$$

As promised, the set of auxiliary variables $[\boldsymbol{z}, \boldsymbol{u}]$ have been eliminated; and each sensor, say the j-th, has to store and update only $(|\mathcal{N}_j| + 1)p$ scalars. To carry out the unconstrained minimization in (3.10), observe that the cost is convex and differentiable. Thus, the first-order necessary condition is also sufficient for optimality. Computing the gradient with respect to $\mathbf{s}_j$ and setting the result equal to zero, yields

$$E\Bigg[-2\mathbf{h}_j(t+1)\left(x_j(t+1) - \mathbf{h}_j^T(t+1)\mathbf{s}_j\right) + \sum_{j' \in \mathcal{N}_j} \left(\mathbf{v}_j^{j'}(t) - \mathbf{v}_{j'}^{j}(t)\right) \\ + 2c \sum_{j' \in \mathcal{N}_j} \left(\mathbf{s}_j - \frac{1}{2}(\mathbf{s}_j(t) + \mathbf{s}_{j'}(t))\right) \Bigg] = \mathbf{0}_p. \tag{3.11}$$

It is now apparent that $\mathbf{s}_j(t+1)$ can be obtained as the root of an equation of the form $\mathbf{f}(\mathbf{s}_j) := E[\boldsymbol{\varphi}(\mathbf{s}_j, x_j(t+1), \mathbf{h}_j(t+1))] = \mathbf{0}_p$, where $\boldsymbol{\varphi}$ stands for the function inside the expectation in (3.11) and corresponds to the stochastic gradient of the cost in (3.10). In lieu of local (cross-) covariance information, namely $\mathbf{r}_{h_j x_j} := E[\mathbf{h}_j(t+1)x_j(t+1)]$ and

$\mathbf{R}_{h_j} := E[\mathbf{h}_j(t+1)\mathbf{h}_j^T(t+1)]$, the root of $\mathbf{f}(\mathbf{s}_j) = \mathbf{0}_p$ is not computable in closed form since $\mathbf{f}$ is unknown. Hence, motivated by stochastic approximation techniques[1] which iteratively find the root of an unknown function $\mathbf{f}(\mathbf{s}_j)$ given a time-series of noisy observations $\{\boldsymbol{\varphi}(\mathbf{s}_j(t), x_j(t+1), \mathbf{h}_j(t+1))\}_{t=0}^{\infty}$, the proposed recursion for every $j \in \mathcal{J}$ is

$$\mathbf{s}_j(t+1) = \mathbf{s}_j(t) + \mu_j \left[2\mathbf{h}_j(t+1)e_j(t+1) - \sum_{j' \in \mathcal{N}_j} (\mathbf{v}_j^{j'}(t) - \mathbf{v}_{j'}^{j}(t)) - c \sum_{j' \in \mathcal{N}_j} (\mathbf{s}_j(t) - \mathbf{s}_{j'}(t))\right] \tag{3.12}$$

where μ_j is a constant step-size and $e_j(t+1) := x_j(t+1) - \mathbf{h}_j^T(t+1)\mathbf{s}_j(t)$ is the local *a priori* error. Recursions (3.9)-(3.12) are tabulated as Algorithm 2, and constitute the D-LMS algorithm that can be arbitrarily initialized. To capture the effects of receiving noise corrupting the variables exchanged among neighboring sensors, the D-LMS recursions are modified to yield

$$\mathbf{v}_j^{j'}(t) = \mathbf{v}_j^{j'}(t-1) + \frac{c}{2}(\mathbf{s}_j(t) - (\mathbf{s}_{j'}(t) + \boldsymbol{\eta}_j^{j'}(t))), \quad j' \in \mathcal{N}_j \tag{3.13}$$

$$\mathbf{s}_j(t+1) = \mathbf{s}_j(t) + \mu_j \left[2\mathbf{h}_j(t+1)e_j(t+1) - \sum_{j' \in \mathcal{N}_j} (\mathbf{v}_j^{j'}(t) - (\mathbf{v}_{j'}^{j}(t) + \bar{\boldsymbol{\eta}}_j^{j'}(t))) \right. \\ \left. - c \sum_{j' \in \mathcal{N}_j} (\mathbf{s}_j(t) - (\mathbf{s}_{j'}(t) + \boldsymbol{\eta}_j^{j'}(t)))\right] \tag{3.14}$$

where $\boldsymbol{\eta}_j^{j'}(t)$ and $\bar{\boldsymbol{\eta}}_j^{j'}(t)$ denote the additive communication noise present in the reception of $\mathbf{s}_{j'}(t)$ and $\mathbf{v}_{j'}^{j}(t)$ at sensor j, respectively. In detail, during time instant $t+1$ sensor j receives the local estimates $\{\mathbf{s}_{j'}(t) + \boldsymbol{\eta}_j^{j'}(t)\}_{j' \in \mathcal{N}_j}$ and plugs them into (3.13) to evaluate $\mathbf{v}_j^{j'}(t)$ for $j' \in \mathcal{N}_j$. Each one of the updated local Lagrange multipliers $\{\mathbf{v}_j^{j'}(t)\}_{j' \in \mathcal{N}_j}$ is subsequently transmitted to the corresponding neighbor $j' \in \mathcal{N}_j$. Then, upon reception of $\{\mathbf{v}_{j'}^{j}(t) + \bar{\boldsymbol{\eta}}_j^{j'}(t)\}_{j' \in \mathcal{N}_j}$, the multipliers are jointly used along with $\{\mathbf{s}_{j'}(t) + \boldsymbol{\eta}_j^{j'}(t)\}_{j' \in \mathcal{N}_j}$ and the newly acquired local data $\{x_j(t+1), \mathbf{h}_j(t+1)\}$ to obtain $\mathbf{s}_j(t+1)$ via (3.14). The $(t+1)$-st iteration is concluded after sensor j broadcasts $\mathbf{s}_j(t+1)$ to its neighbors.

The use of a constant step-size μ_j endows D-LMS with tracking capabilities. This is

[1] Such as the celebrated Robbins-Monro algorithm; see e.g., [26, Ch. 1].

Algorithm 2 : D-LMS

```
Arbitrarily initialize $\{\mathbf{s}_j(0)\}_{j=1}^{J}$ and $\{\mathbf{v}_j^{j'}(-1)\}_{j\in\mathcal{J}}^{j'\in\mathcal{N}_j}$.
for $t = 0, 1, \ldots$ do
    All $j \in \mathcal{J}$: transmit $\mathbf{s}_j(t)$ to neighbors in $\mathcal{N}_j$.
    All $j \in \mathcal{J}$: update $\{\mathbf{v}_j^{j'}(t)\}_{j'\in\mathcal{N}_j}$ using (3.13).
    All $j \in \mathcal{J}$: transmit $\mathbf{v}_j^{j'}(t)$ to each $j' \in \mathcal{N}_j$.
    All $j \in \mathcal{J}$: update $\mathbf{s}_j(t+1)$ using (3.14).
end for
```

desirable in a constantly changing environment, within which WSNs are envisioned to operate. Some related consensus-based estimation approaches compromise adaptability, by introducing a diminishing step-size to suppress the error-propagation effects of communication noise; see, e.g., [21] and [25]. D-LMS is shown to be robust against communication noise in Section 3.5, a property directly inherited from the AD-MoM; see also [49] for related claims in single-shot non-adaptive distributed estimation.

Remark 3.2 (Comparison with the D-LMS algorithm in Chapter 2) To enable task parallelization via AD-MoM while ensuring that estimates agree across the whole WSN, the approach in the previous chapter reformulates (3.1) by relying on the so called bridge sensor subset. Not only setting-up – but also readjusting the bridge sensor set, e.g., when sensors inevitably fail in battery-limited WSN deployments – requires additional coordination among sensors with an associated communication overhead. The approach followed here does not require such a bridge sensor set, and in this sense, it offers a fully distributed, robust, and resource efficient LMS-type algorithm for use in ad hoc WSNs. The contributions in this chapter are also relevant to the D-LMS variant in Chapter 2, as the performance analysis in the forthcoming sections carries over with minor adjustments; see also [31].

3.3.2 D-LMS Algorithm with Ideal Links

Consider an ideal scenario whereby sensors are able to communicate via error-free links. Such an operational setup may arise, e.g., whenever the use of powerful channel codes renders inter-sensor links virtually noise-free. Next, we show that under such assumptions, D-LMS can be simplified to yield a set of local recursions which are equivalent to (3.9) and (3.12), while incurring notably lower communication and reduced computational complexities with respect to the original version of the algorithm.

Specifically, note first that if the Lagrange multipliers $\boldsymbol{v}$ are initialized such that $\mathbf{v}_j^{j'}(-1) = -\mathbf{v}_{j'}^{j}(-1)$ with $j \in \mathcal{J}$ and $j' \in \mathcal{N}_j$, then in the absence of communication noise one finds from (3.9) that $\mathbf{v}_j^{j'}(t) = -\mathbf{v}_{j'}^{j}(t)$ for all $t \geq 0$. By relying on this identity and starting from (3.12), we arrive at a simplified recursion to update the local estimates $\mathbf{s}_j(t+1)$ for all $j \in \mathcal{J}$

$$\mathbf{s}_j(t+1) = \mathbf{s}_j(t) + \mu_j \left[2\mathbf{h}_j(t+1)e_j(t+1) - 2\sum_{j' \in \mathcal{N}_j} \mathbf{v}_j^{j'}(t) - c \sum_{j' \in \mathcal{N}_j} (\mathbf{s}_j(t) - \mathbf{s}_{j'}(t)) \right]. \quad (3.15)$$

The specific initialization requirement for the multipliers is not restrictive, as it can be readily satisfied by selecting $\mathbf{v}_j^{j'}(-1) = \mathbf{0}_p$ for $j \in \mathcal{J}$ and $j' \in \mathcal{N}_j$ without the need of extra coordination among sensors.

The second summand inside the square brackets in (3.15) incorporates only the local multipliers $\{\mathbf{v}_j^{j'}(t)\}_{j \in \mathcal{N}_j}$ stored at sensor j. Hence, multipliers need not be communicated to the neighboring sensors at all. What is more, multipliers enter (3.15) only through their local sum across $j' \in \mathcal{N}_j$, so that there is no need to keep track of them separately. This motivates introducing the new set of local variables $\mathbf{p}_j(t) := 2\sum_{j' \in \mathcal{N}_j} \mathbf{v}_j^{j'}(t)$ for $j = 1, \ldots, J$, which have to be updated in conjunction with $\mathbf{s}_j(t)$ The updating rule for $\mathbf{p}_j(t)$ follows immediately from (3.9), and the final recursions per sensor $j \in \mathcal{J}$ are

$$\mathbf{p}_j(t) = \mathbf{p}_j(t-1) + c \sum_{j' \in \mathcal{N}_j} (\mathbf{s}_j(t) - \mathbf{s}_{j'}(t)) \quad (3.16)$$

$$\mathbf{s}_j(t+1) = \mathbf{s}_j(t) + \mu_j \left[2\mathbf{h}_j(t+1)e_j(t+1) - \mathbf{p}_j(t) - c \sum_{j' \in \mathcal{N}_j} (\mathbf{s}_j(t) - \mathbf{s}_{j'}(t)) \right]. \quad (3.17)$$

Interestingly, (3.16)-(3.17) require each sensor to store and update only $2p$ scalars, regardless of the WSN topology and corresponding neighborhood sizes. While diffusion LMS [28] needs half the number of scalar recursions, in D-LMS [cf. (3.13)-(3.14)] sensor j has to store and update $(|\mathcal{N}_j| + 1)p$ scalars. With regards to communication cost incurred by the

Algorithm 3 : D-LMS with Ideal Links

```
Arbitrarily initialize $\{\mathbf{s}_j(0)\}_{j=1}^{J}$ and set $\{\mathbf{p}_j(-1) = \mathbf{0}_p\}_{j\in\mathcal{J}}$.
for $t = 0, 1, \ldots$ do
    All $j \in \mathcal{J}$: transmit $\mathbf{s}_j(t)$ to neighbors in $\mathcal{N}_j$.
    All $j \in \mathcal{J}$: update $\mathbf{p}_j(t)$ using (3.16).
    All $j \in \mathcal{J}$: update $\mathbf{s}_j(t+1)$ using (3.17).
end for
```

D-LMS variant in (3.16)-(3.17), on a per-iteration basis, sensor j receives $|\mathcal{N}_j|p$ scalars due to $\{\mathbf{s}_{j'}(t)\}_{j'\in\mathcal{N}_j}$ and transmits p scalars corresponding to $\mathbf{s}_j(t+1)$; exactly as in diffusion LMS. Recall that D-LMS in (3.13)-(3.14) incurs an additional cost of $2|\mathcal{N}_j|p$ communicated scalars due to the reception and transmission of $\{\mathbf{v}_{j'}^{j}(t)\}_{j'\in\mathcal{N}_j}$ and $\{\mathbf{v}_{j}^{j'}(t)\}_{j'\in\mathcal{N}_j}$, respectively. Recursions (3.16)-(3.17) (also tabulated as Algorithm 3) are equivalent to D-LMS whenever $\boldsymbol{\eta}_j^{j'}(t) = \bar{\boldsymbol{\eta}}_j^{j'}(t) = \mathbf{0}_p$ in (3.13)-(3.14). Hence, both will achieve identical convergence rates and estimation performance, making Algorithm 3 the most attractive alternative when noise is not an issue as corroborated via the numerical tests in Section 3.6. However, there is a price paid for the reduced amount of communications and computational savings as clarified on the ensuing remark.

Remark 3.3 (Communication noise resilience) The D-LMS variant in (3.16)-(3.17) is only applicable when communication links are ideal. Being equivalent to D-LMS under this assumption, one might still be tempted to replace $\{\mathbf{s}_{j'}(t)\}_{j'\in\mathcal{N}_j}$ with $\{\mathbf{s}_{j'}(t) + \boldsymbol{\eta}_j^{j'}(t)\}_{j'\in\mathcal{N}_j}$ in recursions (3.16)-(3.17) to capture the effects of noise corrupting the exchanged local estimates. As it turns out, in the process of running (3.16)-(3.17) noise will accumulate resulting in local estimates whose variance grows unbounded as $t \to \infty$. The reduced communication overhead is thus counterbalanced by the lack of resilience in the presence of communication noise. As a byproduct, this highlights the key role played by the Lagrange multiplier exchanges in rendering D-LMS – and generally all MoM-based distributed algorithms – robust to communication noise; see also [49] for further details.

3.4 Analysis Preliminaries

Our approach to performance analysis relies on a compact *error-form* representation of D-LMS as a linear time-varying stochastic difference equation. As discussed in this section, the resulting estimation error covariance matrix encompasses all the information needed to evaluate the relevant performance metrics; namely MSE, excess mean-square error (EMSE) and mean-square deviation (MSD). The aforementioned figures of merit ultimately assess the performance of D-LMS, both on a per-sensor basis and collectively by considering the WSN as a whole.

3.4.1 Error-form D-LMS

In this subsection, we start from the D-LMS recursions in (3.13)-(3.14) and characterize the evolution of the local estimation errors $\{\mathbf{y}_{1,j}(t) := \mathbf{s}_j(t) - \mathbf{s}_0(t)\}_{j=1}^J$ and multiplier-based quantities $\{\mathbf{y}_{2,j}(t) := \sum_{j' \in \mathcal{N}_j} (\mathbf{v}_j^{j'}(t-1) - \mathbf{v}_{j'}^j(t-1))\}_{j=1}^J$. It turns out that a convenient global state capturing the spatio-temporal dynamics of D-LMS can be defined as $\mathbf{y}(t) := [\mathbf{y}_1^T(t)\ \mathbf{y}_2^T(t)]^T = [\mathbf{y}_{1,1}^T(t) \ldots \mathbf{y}_{1,J}^T(t)\ \mathbf{y}_{2,1}^T(t) \ldots \mathbf{y}_{2,J}^T(t)]^T$. While the need of the local errors within $\mathbf{y}_1(t)$ is apparent, augmentation with the seemingly unnecessary $\mathbf{y}_2(t)$ will prove useful to obtain a simple, first-order difference equation for $\mathbf{y}(t)$. Otherwise, a first-order recursion for $\mathbf{y}_1(t)$ is impossible. In order to proceed, we shall require for all $j \in \mathcal{J}$ that:

(a1) *Sensor observations obey $x_j(t) = \mathbf{h}_j^T(t)\mathbf{s}_0(t-1) + \epsilon_j(t)$, where the zero-mean white noise $\{\epsilon_j(t)\}$ has variance $\sigma_{\epsilon_j}^2$.*

Linear models are commonly used throughout the adaptive signal processing literature to facilitate stability and performance analysis, e.g., [43], [52, Ch. 5,9], [28], [27], and [48]. Observation noise variances can differ across sensors, accounting for faulty sensing devices presumably leading to larger values of $\sigma_{\epsilon_j}^2$.

Remark 3.4 (Sensor data assumptions) An attractive feature of D-LMS is that it can be applied to a wide class of distributed linear regression problems. Indeed, D-LMS does not require prior knowledge of a data model to describe the sensor observations, as the underlying process statistics are learnt ‘on-the-fly’. In this sense, D-LMS differs from the distributed Kalman filtering approaches in [35], [45], and [9], which are only applicable when state and observation models are available locally at each sensor. When it comes to stability and performance evaluation however, a meaningful ‘ground-truth’ model should be adopted to carry out the analysis and enable fair comparison among competing alternatives. It is true that assumption (a1) delimitates the scope of the forthcoming analysis, though by no means restrains D-LMS from being applied in broader settings.

To concisely capture the effects of both observation and communication noise on the estimation errors across the WSN, define the $Jp \times 1$ noise vectors $\boldsymbol{\epsilon}(t) := 2\mu[\mathbf{h}_1^T(t)\epsilon_1(t)\ldots\mathbf{h}_J^T(t)\epsilon_J(t)]^T$ and $\bar{\boldsymbol{\eta}}(t) := [\bar{\boldsymbol{\eta}}_1^T(t)\ldots\bar{\boldsymbol{\eta}}_J^T(t)]^T$, where vectors $\{\bar{\boldsymbol{\eta}}_j(t)\}_{j\in\mathcal{J}}$ are given by

$$\bar{\boldsymbol{\eta}}_j(t) := \sum_{j'\in\mathcal{N}_j} \bar{\boldsymbol{\eta}}_j^{j'}(t) \tag{3.18}$$

and $\mathbf{R}_{\bar{\boldsymbol{\eta}}} := E[\bar{\boldsymbol{\eta}}(t)\bar{\boldsymbol{\eta}}^T(t)]$. The vector in (3.18) amounts to the accumulated communication noise at sensor j, due to the reception of all required multipliers at time t, namely $\{\mathbf{v}_{j'}^j(t)\}_{j'\in\mathcal{N}_j}$. Next, introduce the $p(\sum_{j=1}^J |\mathcal{N}_j|) \times 1$ vector

$$\boldsymbol{\eta}(t) := \left[\{(\boldsymbol{\eta}_{j'}^1(t))^T\}_{j'\in\mathcal{N}_1}\ldots\{(\boldsymbol{\eta}_{j'}^J(t))^T\}_{j'\in\mathcal{N}_J}\right]^T \tag{3.19}$$

which comprises the receiver noises corrupting transmissions of local estimates across the whole network at time instant t, and define $\mathbf{R}_{\boldsymbol{\eta}} := E[\boldsymbol{\eta}(t)\boldsymbol{\eta}^T(t)]$. Finally, let $\mathbf{L}_c := c\mathbf{L} \otimes \mathbf{I}_p \in \mathbb{R}^{Jp\times Jp}$ be a matrix capturing the WSN topology through the (scaled) graph Laplacian $\mathbf{L}$, and arrange the instantaneous outer products of regression vectors in $\mathbf{R}_h(t) := \text{bdiag}\left(\mathbf{h}_1(t)\mathbf{h}_1^T(t),\ldots,\mathbf{h}_J(t)\mathbf{h}_J^T(t)\right) \in \mathbb{R}^{Jp\times Jp}$. Based on these definitions and assuming for simplicity in exposition that $\mu_j = \mu$ for all $j \in \mathcal{J}$, the following instrumental lemma is established in Appendix B.

Lemma 3.1 *Under (a1) and for $t \geq 0$, the global state $\mathbf{y}(t)$ evolves according to*

$$\mathbf{y}(t+1) = bdiag(\mathbf{I}_{Jp},\mathbf{L}_c)\mathbf{z}(t+1) + \begin{bmatrix} \mu\mathbf{I}_{Jp} \\ \mathbf{0}_{Jp\times Jp} \end{bmatrix} \bar{\boldsymbol{\eta}}(t) + \begin{bmatrix} \mu(3\mathbf{P}_\alpha - \mathbf{P}_\beta) \\ \mathbf{P}_\beta - \mathbf{P}_\alpha \end{bmatrix} \boldsymbol{\eta}(t) \tag{3.20}$$

where the inner state $\mathbf{z}(t) := [\mathbf{z}_1^T(t)\ \mathbf{z}_2^T(t)]^T$ *is arbitrarily initialized and updated according to*

$$\mathbf{z}(t+1) = \mathbf{\Phi}(t+1,\mu)\mathbf{z}(t) + \begin{bmatrix} \boldsymbol{\epsilon}(t+1) \\ \mathbf{0}_{Jp} \end{bmatrix} - \begin{bmatrix} \mathbf{1}_J \otimes (\mathbf{s}_0(t+1) - \mathbf{s}_0(t)) \\ \mathbf{0}_{Jp} \end{bmatrix}$$
$$+ \mathbf{\Phi}(t+1,\mu) \begin{bmatrix} \mu \mathbf{I}_{Jp} \\ \mathbf{0}_{Jp \times Jp} \end{bmatrix} \bar{\boldsymbol{\eta}}(t-1) + \mathbf{\Phi}(t+1,\mu) \begin{bmatrix} \mu(3\mathbf{P}_\alpha - \mathbf{P}_\beta) \\ \mathbf{C} \end{bmatrix} \boldsymbol{\eta}(t-1) \tag{3.21}$$

and the $2Jp \times 2Jp$ *transition matrix* $\mathbf{\Phi}(t,\mu)$ *consists of the blocks* $[\mathbf{\Phi}(t,\mu)]_{11} = \mathbf{I}_{Jp} - 2\mu(\mathbf{R}_h(t) + \mathbf{L}_c)$, $[\mathbf{\Phi}(t,\mu)]_{12} = -\mu\mathbf{L}_c$ *and* $[\mathbf{\Phi}(t,\mu)]_{21} = [\mathbf{\Phi}(t,\mu)]_{22} = \mathbf{L}_c\mathbf{L}_c^\dagger$*. The matrix* $\mathbf{C}$ *is chosen such that* $\mathbf{L}_c\mathbf{C} = \mathbf{P}_\beta - \mathbf{P}_\alpha$*, where the structure of the time-invariant matrices* $\mathbf{P}_\alpha$ *and* $\mathbf{P}_\beta$ *is given in Appendix B.*

The desired state $\mathbf{y}(t)$ is obtained as a rank-deficient linear transformation of the inner state $\mathbf{z}(t)$, plus a stochastic offset due to the effects of communication noise. A linear, time-varying, first-order difference equation describes the dynamics of $\mathbf{z}(t)$, and hence of $\mathbf{y}(t)$, via the algebraic transformation in (3.20). The time-varying nature of (3.20)-(3.21) is inherited from $\mathbf{\Phi}(t,\mu)$ that depends on the regression vectors within $\mathbf{R}_h(t)$. Four stochastic inputs are clearly discernible from (3.21): i) communication noise $\boldsymbol{\eta}(t-1)$ affecting the transmission of local estimates; ii) communication noise $\bar{\boldsymbol{\eta}}(t-1)$ contaminating the Lagrange multipliers; iii) observation noise within $\boldsymbol{\epsilon}(t+1)$; and iv) a forcing term due to the true 'parameter speed' $\mathbf{s}_0(t+1) - \mathbf{s}_0(t)$.

3.4.2 Performance Metrics

When it comes to performance evaluation of adaptive algorithms, it is customary to consider as figures of merit the so-called MSE, excess mean-square error (EMSE) and mean-square deviation (MSD) [43], [52]. In the present setup for distributed adaptive estimation, it is pertinent to address both global (network-wide) and local (per-sensor) performance [28]. After recalling the definitions of the local a priori error $e_j(t) := x_j(t) - \mathbf{h}_j^T(t)\mathbf{s}_j(t-1)$ and local estimation error $\mathbf{y}_{1,j}(t) := \mathbf{s}_j(t) - \mathbf{s}_0(t)$, the per-sensor performance metrics are defined as

$$\begin{aligned}\mathrm{MSE}_j(t) &:= E[e_j^2(t)]\\ \mathrm{EMSE}_j(t) &:= E[(\mathbf{h}_j^T(t)\mathbf{y}_{1,j}(t-1))^2]\\ \mathrm{MSD}_j(t) &:= E[\|\mathbf{y}_{1,j}(t)\|^2]\end{aligned}$$

whereas their global counterparts are defined as the respective averages across sensors, e.g., $\mathrm{MSE}(t) := J^{-1}\sum_{j=1}^{J} E[e_j(t)^2]$ and so on. Assume $\forall\ j \in \mathcal{J}$ that:

(a2) Vectors $\{\mathbf{h}_j(t)\}$ *are spatio-temporally white with covariance matrix* $\mathbf{R}_{h_j} \succ \mathbf{0}_{p\times p}$*; and*

(a3) Vectors $\{\mathbf{h}_j(t)\}$*,* $\{\epsilon_j(t)\}$*,* $\{\boldsymbol{\eta}_j^{j'}(t)\}_{j'\in\mathcal{N}_j}$ *and* $\{\bar{\boldsymbol{\eta}}_j^{j'}(t)\}_{j'\in\mathcal{N}_j}$ *are independent.*

Assumptions (a1)-(a3) comprise the widely adopted *independence setting*, which is instrumental in rendering the subsequent performance analysis tractable; see e.g., [52, pg. 110], [43, pg. 448]. Clearly, (a2) can be violated in, e.g., FIR filtering of signals (regressors) with a shift structure as in the distributed power spectrum estimation problem described in [48, Remark 1]. Nonetheless, for small step-sizes the upshot of the analysis extends to correlated data as will be demonstrated via computer simulations in Section 3.6.

Next, we show that it suffices to evaluate the state covariance matrix $\mathbf{R}_y(t) := E[\mathbf{y}(t)\mathbf{y}^T(t)]$ in order to assess the aforementioned performance metrics. To this end, note that by virtue of (a1) it is possible to write $e_j(t) = -\mathbf{h}_j^T(t)\mathbf{y}_{1,j}(t-1) + \epsilon_j(t)$. Because $\mathbf{y}_{1,j}(t-1)$ is independent of the zero-mean $\{\mathbf{h}_j(t), \epsilon_j(t)\}$ under (a1)-(a3), from the previous relationship between the a priori and estimation errors one finds that $\mathrm{MSE}_j(t) = \mathrm{EMSE}_j(t) + \sigma_{\epsilon_j}^2$. Hence, it suffices to focus on the evaluation of $\mathrm{EMSE}_j(t)$, through which $\mathrm{MSE}_j(t)$ can also be determined. If $\mathbf{R}_{y_{1,j}}(t) := E[\mathbf{y}_{1,j}(t)\mathbf{y}_{1,j}^T(t)]$ denotes the j-th local error covariance matrix, then $\mathrm{MSD}_j(t) = \mathrm{tr}(\mathbf{R}_{y_{1,j}}(t))$; and under (a1)-(a3), a simple manipulation yields

$$\begin{aligned}\mathrm{EMSE}_j(t) &= E[\mathrm{tr}((\mathbf{h}_j^T(t)\mathbf{y}_{1,j}(t-1))^2)] = \mathrm{tr}(E[\mathbf{h}_j(t)\mathbf{h}_j^T(t)\mathbf{y}_{1,j}(t-1)\mathbf{y}_{1,j}^T(t-1)])\\ &= \mathrm{tr}(E[\mathbf{h}_j(t)\mathbf{h}_j^T(t)]E[\mathbf{y}_{1,j}(t-1)\mathbf{y}_{1,j}^T(t-1)]) = \mathrm{tr}(\mathbf{R}_{h_j}\mathbf{R}_{y_{1,j}}(t-1)).\end{aligned}$$

Observe that the global error covariance matrix corresponds to the $Jp \times Jp$ upper left submatrix $[\mathbf{R}_y(t)]_{11}$ of $\mathbf{R}_y(t)$. Further, its j-th $p \times p$ diagonal submatrix ($j = 1, \ldots, J$) denoted by $[\mathbf{R}_y(t)]_{11,j}$ is $\mathbf{R}_{y_{1,j}}(t)$. It follows that with $\mathbf{R}_h := E[\mathbf{R}_h(t)] = \text{bdiag}(\mathbf{R}_{h_1}, \ldots, \mathbf{R}_{h_J})$, the global performance metrics are given by $\text{MSD}(t) = J^{-1}\text{tr}([\mathbf{R}_y(t)]_{11})$ and $\text{EMSE}(t) = J^{-1}\text{tr}(\mathbf{R}_h[\mathbf{R}_y(t-1)]_{11})$. In a nutshell, deriving a closed-form expression for $\mathbf{R}_y(t)$ enables the evaluation of all performance metrics of interest.

3.5 Performance Analysis of D-LMS Tracking

In this section, a performance analysis is conducted for the D-LMS tracking algorithm. For that purpose, we first introduce a commonly adopted model for the time-varying parameter $\mathbf{s}_0(t)$, and resort to a simplifying Gaussian assumption on the regression vectors to complement (a1)-(a3). Main results include deriving an exact closed-form recursion for the global error covariance matrix $[\mathbf{R}_y(t)]_{11}$, and establishing the existence of step-sizes ensuring stability of D-LMS both in the mean and MSE-sense. Expressions for the steady-state local and global figures of merit are also provided. To conclude, the steady-state EMSE is viewed as a function of the step-size, and is compared for time-invariant and time-varying parameters. While in the former case the trend is monotonically increasing, when tracking slowly time-varying processes there exists a non-vanishing optimal step-size minimizing the limiting error.

To characterize fluctuations of the time-varying parameter $\mathbf{s}_0(t)$, consider $\forall\ j \in \mathcal{J}$ a:

(a4) *First-order autoregressive [AR(1)] model, i.e.,*

$$\begin{cases} \mathbf{s}_0(t) &= \mathbf{s}_0 + \check{\mathbf{s}}(t) \\ \check{\mathbf{s}}(t) &= \boldsymbol{\Theta}\check{\mathbf{s}}(t-1) + \boldsymbol{\zeta}(t) \end{cases}$$

where $\boldsymbol{\Theta} \in \mathbb{R}^{p\times p}$ *has eigenvalues with modulus within* $[0,1)$*, the driving noise* $\{\boldsymbol{\zeta}(t)\}$ *is zero-mean, white with covariance matrix* $\mathbf{R}_\zeta \succ \mathbf{0}_{p\times p}$*; and* $E[\check{\mathbf{s}}(-1)] = \mathbf{0}_p$.

Under (a4), the perturbation due to the parameter velocity in (3.21) becomes

$$-\begin{bmatrix} \mathbf{1}_J \otimes (\mathbf{s}_0(t+1) - \mathbf{s}_0(t)) \\ \mathbf{0}_{Jp} \end{bmatrix} = \begin{bmatrix} \mathbf{1}_J \otimes (\mathbf{I}_p - \boldsymbol{\Theta})\check{\mathbf{s}}(t) \\ \mathbf{0}_{Jp} \end{bmatrix} - \begin{bmatrix} \mathbf{1}_J \otimes \boldsymbol{\zeta}(t+1) \\ \mathbf{0}_{Jp} \end{bmatrix}, \tag{3.22}$$

and its expectation vanishes for all $t \geq 0$. Further, (a3) is augmented and replaced by

(a5) Vectors $\{\mathbf{h}_j(t)\}$, $\{\epsilon_j(t)\}$, $\{\boldsymbol{\zeta}(t)\}$, $\breve{\mathbf{s}}(-1)$, $\{\boldsymbol{\eta}_j^{j'}(t)\}_{j'\in\mathcal{N}_j}$ *and* $\{\bar{\boldsymbol{\eta}}_j^{j'}(t)\}_{j'\in\mathcal{N}_j}$ *are independent.*

The model in (a4) provides a simple description of a time-varying parameter, and has been widely adopted to evaluate the performance of (centralized) adaptive filters [52, pg. 121], [43, pg. 360]. The true parameter $\mathbf{s}_0(t)$ has been split into a 'DC level' $\mathbf{s}_0$ which is superimposed to the 'AC component' $\breve{\mathbf{s}}(t)$, with fluctuations adhering to a stable vector AR(1) process. Two other models of interest are obtained by simple modifications to (a4):

(a4.1) *Time-invariant parameter, i.e.,* $\mathbf{s}_0(t) = \mathbf{s}_0$ *is subsumed by (a4), by selecting* $\boldsymbol{\Theta} = \mathbf{R}_\zeta = \mathbf{0}_{p\times p}$.

(a4.2) *Random-walk model, i.e.,* $\mathbf{s}_0(t) = \mathbf{s}_0(t-1) + \boldsymbol{\zeta}(t)$, *where* $\{\boldsymbol{\zeta}(t)\}$ *is zero-mean, white, with covariance matrix* $\mathbf{R}_\zeta \succ \mathbf{0}$; *and* $E[\mathbf{s}_0(-1)] = \mathbf{0}_p$. *Such a model is obtained by letting* $\mathbf{s}_0 = \mathbf{0}_p$ *and* $\boldsymbol{\Theta} = \mathbf{I}_p$ *in (a4).*

A random-walk is the simplest stochastic model to describe variations of $\mathbf{s}_0(t)$, and has been also considered for performance analysis of trackers; see, e.g., [52, pg. 121] and [43, pg. 359]. It could be arguably thought as not meaningful due to its increasing variance, thus violating the sensor's limited dynamical range which requires $E[x_j^2(t)] < \infty$, $\forall\, t$. To circumvent this problem, the forthcoming analysis generalizes [31] by considering the (asymptotically) stationary case in (a4). When $\boldsymbol{\Theta}$ is a stable matrix, i.e., $\lambda_{\max}(\boldsymbol{\Theta}) < 1$, the steady-state covariance matrix of $\mathbf{s}_0(t)$ has finite entries, and obeys the Lyapunov equation $\mathbf{R}_{\mathbf{s}_0}(\infty) = \boldsymbol{\Theta}\mathbf{R}_{\mathbf{s}_0}(\infty)\boldsymbol{\Theta}^T + \mathbf{R}_\zeta$. In any case, the model is simple but well justified as the resulting analysis sheds sufficient light on the key aspects of D-LMS when it comes to tracking.

3.5.1 Mean Stability

From Lemma 3.1 it is straightforward to establish that local estimates obtained via D-LMS are asymptotically unbiased, implying that consensus in the mean-sense is achieved on $\mathbf{s}_0$.

Proposition 3.1 *Under (a1)-(a2) and (a4)-(a5), the D-LMS algorithm achieves consensus in the mean, i.e.,*

$$\lim_{t\to\infty} E[\mathbf{y}_{1,j}(t)] = \mathbf{0}_p, \quad \forall\, j \in \mathcal{J}$$

provided the step-size is chosen such that $\mu \in (0, \mu_u)$ *with*

$$\mu_u := \min\left(\frac{2}{\lambda_{\max}(\mathbf{R}_h + \mathbf{L}_c)}, \frac{2}{\lambda_{\max}(2\mathbf{R}_h + (3/2)\mathbf{L}_c)}\right). \tag{3.23}$$

Proof: Based on the independence setting assumptions (a1)-(a2), (a5) and since the data is zero-mean, one obtains after taking expectations on (3.20) and (3.21) that $E[\mathbf{y}(t)] = \text{bdiag}(\mathbf{I}_{Jp}, \mathbf{L}_c)E[\mathbf{z}(t)]$ and $E[\mathbf{z}(t)] = \boldsymbol{\Phi}(\mu)E[\mathbf{z}(t-1)]$, where $\boldsymbol{\Phi}(\mu) := E[\boldsymbol{\Phi}(t, \mu)]$. Assumption (a4) was also invoked to render the expectation of (3.22) null. The following lemma specifies the step-size values under which $\boldsymbol{\Phi}(\mu)$ is a stable matrix.

Lemma 3.2 *If* $\mu > 0$ *is chosen smaller than (3.23), then* $\boldsymbol{\Phi}(\mu)$ *is a stable matrix, i.e.,* $\lambda_{\max}(\boldsymbol{\Phi}(\mu)) < 1$.

Proof: Following steps similar to those in [49, Appendix H], it is possible to express the eigenvalues of $\boldsymbol{\Phi}(\mu)$ as the roots of a second-order polynomial to determine bounds on μ that ensure $\lambda_{\max}(\boldsymbol{\Phi}(\mu)) < 1$. Further, for sufficiently small μ the eigenvalues with largest modulus correspond to a complex conjugate pair, while the spectral radius scales as $\lambda_{\max}(\boldsymbol{\Phi}(\mu)) \sim 1 - \mu\kappa$, where $\kappa > 0$ is a finite constant. From Lemma 3.2 and the theory of linear time-invariant dynamical systems, $E[\mathbf{z}(t)]$ is exponentially convergent to zero for $\mu \in (0, \mu_u)$. Noting that $E[\mathbf{y}_1(t)] = E[\mathbf{z}_1(t)]$ [cf. (3.20)], the result follows. Interestingly, μ_u resembles the first-order (mean) stability bound for the centralized LMS algorithm, namely $2/\lambda_{\max}(\mathbf{R}_h)$ [52, pg. 111]. The main difference here is that this bound is also affected by the topology of the WSN, via the graph Laplacian matrix within $\mathbf{L}_c$.

3.5.2 MSE Stability and Performance Evaluation

Turning to MSE stability and performance analysis, observe from the upper $Jp \times 1$ block of $\mathbf{y}(t+1)$ in (3.20) that $\mathbf{y}_1(t+1) = \mathbf{z}_1(t+1) + \mu[\bar{\boldsymbol{\eta}}(t) + (3\mathbf{P}_\alpha - \mathbf{P}_\beta)\boldsymbol{\eta}(t)]$. Under (a2) and (a5), $\mathbf{z}_1(t+1)$ is independent of the zero-mean $\{\bar{\boldsymbol{\eta}}(t), \boldsymbol{\eta}(t)\}$; hence,

$$[\mathbf{R}_y(t)]_{11} = [\mathbf{R}_z(t)]_{11} + \mu^2[\mathbf{R}_{\bar{\boldsymbol{\eta}}} + (3\mathbf{P}_\alpha - \mathbf{P}_\beta)\mathbf{R}_{\boldsymbol{\eta}}(3\mathbf{P}_\alpha - \mathbf{P}_\beta)^T] \tag{3.24}$$

based on which we obtain $\mathbf{R}_z(t) := E[\mathbf{z}(t)\mathbf{z}^T(t)]$. From the coupling between $\mathbf{z}(t)$ and $\check{\mathbf{s}}(t)$ entering through (3.22), it is convenient to consider the augmented state $\check{\mathbf{z}}(t) := [\mathbf{z}^T(t) \;\; \mathbf{1}_J^T \otimes \check{\mathbf{s}}^T(t)]^T$ in order to perform covariance calculations [52, pg. 124]. From (3.21), (a4) and (3.22), one finds that $\check{\mathbf{z}}(t)$ can be recursively updated as

$$\begin{aligned}\check{\mathbf{z}}(t+1) &= \begin{bmatrix} \mathbf{\Phi}(t+1,\mu) & \left[(\mathbf{I}_J \otimes (\mathbf{I}_p - \mathbf{\Theta}))^T \;\; \mathbf{0}_{Jp\times Jp}\right]^T \\ \mathbf{0}_{Jp\times 2Jp} & \mathbf{I}_J \otimes \mathbf{\Theta} \end{bmatrix} \check{\mathbf{z}}(t) + \begin{bmatrix} \boldsymbol{\epsilon}(t+1) \\ \mathbf{0}_{2Jp} \end{bmatrix} \\ &+ \begin{bmatrix} -\mathbf{1}_J \otimes \boldsymbol{\zeta}(t+1) \\ \mathbf{0}_{Jp} \\ \mathbf{1}_J \otimes \boldsymbol{\zeta}(t+1) \end{bmatrix} + \begin{bmatrix} \mathbf{\Phi}(t+1,\mu) \\ \mathbf{0}_{Jp\times 2Jp} \end{bmatrix} (\bar{\boldsymbol{\eta}}_\mu(t-1) + \boldsymbol{\eta}_\mu(t-1)) \qquad (3.25) \\ &:= \mathbf{\Omega}(t+1,\mu)\check{\mathbf{z}}(t) + \boldsymbol{\nu}(t+1) \qquad (3.26)\end{aligned}$$

where for notational convenience $\mathbf{\Omega}(t+1,\mu)$ denotes the new transition matrix and $\boldsymbol{\nu}(t+1)$ encapsulates all three forcing terms. Note that in writing (3.25) we have introduced

$$\bar{\boldsymbol{\eta}}_\mu(t) := \begin{bmatrix} \mu\mathbf{I}_{Jp} \\ \mathbf{0}_{Jp\times Jp} \end{bmatrix} \bar{\boldsymbol{\eta}}(t), \quad \boldsymbol{\eta}_\mu(t) := \begin{bmatrix} \mu(3\mathbf{P}_\alpha - \mathbf{P}_\beta) \\ \mathbf{C} \end{bmatrix} \boldsymbol{\eta}(t) \qquad (3.27)$$

while the structure of the respective covariance matrices $\mathbf{R}_{\bar{\boldsymbol{\eta}}_\mu} := E[\bar{\boldsymbol{\eta}}_\mu(t)\bar{\boldsymbol{\eta}}_\mu^T(t)]$ and $\mathbf{R}_{\boldsymbol{\eta}_\mu} := E[\boldsymbol{\eta}_\mu(t)\boldsymbol{\eta}_\mu^T(t)]$ is given in Appendix C.

By definition of the augmented state $\check{\mathbf{z}}(t)$, the desired covariance matrix $\mathbf{R}_z(t)$ clearly corresponds to the $2Jp \times 2Jp$ upper left submatrix of $\mathbf{R}_{\check{z}}(t) := E[\check{\mathbf{z}}(t)\check{\mathbf{z}}^T(t)]$. Towards obtaining a closed-form expression for $\mathbf{R}_{\check{z}}(t)$, observe that for all $j \in \mathcal{J}$ there exist $p \times p$ unitary matrices $\mathbf{U}_j$ that are arranged in $\mathbf{U} := \text{bdiag}(\mathbf{U}_1, \ldots, \mathbf{U}_J)$ such that $\mathbf{U}_j\mathbf{R}_{h_j}\mathbf{U}_j^T = \mathbf{\Lambda}_j = \text{diag}(\lambda_1^j, \ldots, \lambda_p^j)$, and also $\mathbf{U}\mathbf{R}_h\mathbf{U}^T = \mathbf{\Lambda} = \text{bdiag}(\mathbf{\Lambda}_1, \ldots, \mathbf{\Lambda}_J)$. For the subsequent arguments, it will prove useful to introduce the (invertible) change of variables $\tilde{\mathbf{z}}(t) := \tilde{\mathbf{U}}\check{\mathbf{z}}(t)$ with $\tilde{\mathbf{U}} := \text{bdiag}(\mathbf{U}, \mathbf{I}_{2Jp})$. To proceed, specialize (a2) by assuming that:

(a6) Vectors $\{\mathbf{h}_j(t)\}$ *are spatio-temporally white Gaussian with covariance matrix* $\mathbf{R}_{h_j} \succ \mathbf{0}_p$.

The Gaussianity assumption is instrumental in obtaining closed-form expressions for the regressors' fourth-order moments, which arise in the evaluation of $\mathbf{R}_{\check{z}}(t+1)$ as shown next.

Proposition 3.2 *Under (a4)-(a6) and for $t \geq 0$, the covariance matrix of $\tilde{\mathbf{z}}(t)$ obeys the first-order matrix recursion given by*

$$\begin{aligned}\mathbf{R}_{\tilde{z}}(t+1) = {} & \mathcal{M}(\tilde{\mathbf{\Omega}}(\mu), \mathbf{R}_{\tilde{z}}(t)) + \mathcal{M}(\tilde{\mathbf{\Phi}}(\mu), bdiag(\mathbf{U}, \mathbf{I}_{Jp})(\mathbf{R}_{\bar{\boldsymbol{\eta}}_\mu} + \mathbf{R}_{\boldsymbol{\eta}_\mu}) bdiag(\mathbf{U}^T, \mathbf{I}_{Jp})) \\ & + 4\mu^2 bdiag(\sigma^2_{\epsilon_1}\mathbf{\Lambda}_1, \ldots, \sigma^2_{\epsilon_J}\mathbf{\Lambda}_J, \mathbf{0}_{2Jp\times 2Jp}) \\ & + \tilde{\mathbf{U}}\left(([-1\;0\;1]^T[-1\;0\;1]) \otimes (\mathbf{1}_{J\times J} \otimes \mathbf{R}_\zeta)\right)\tilde{\mathbf{U}}^T \end{aligned} \tag{3.28}$$

with $\tilde{\mathbf{\Omega}}(\mu) := \tilde{\mathbf{U}}E[\mathbf{\Omega}(t,\mu)]\tilde{\mathbf{U}}^T$ *and* $\tilde{\mathbf{\Phi}}(\mu) := \tilde{\mathbf{U}}E[[\mathbf{\Phi}(t+1,\mu)^T\ \mathbf{0}_{2Jp\times Jp}]^T] bdiag(\mathbf{U}^T, \mathbf{I}_{Jp}))$, *while*

$$\begin{aligned}\mathcal{M}(\mathbf{S},\mathbf{T}) := {} & \mathbf{S}\mathbf{T}\mathbf{S}^T + 4\mu^2 bdiag((\mathbf{I}_J \otimes \mathbf{1}_{p\times p}) \circ \mathbf{\Lambda}[\mathbf{T}]_{11}\mathbf{\Lambda}, \mathbf{0}_{2Jp\times 2Jp}) \\ & + 4\mu^2 bdiag(tr(\mathbf{\Lambda}_1[\mathbf{T}]_{11,1})\mathbf{\Lambda}_1, \ldots, tr(\mathbf{\Lambda}_J[\mathbf{T}]_{11,J})\mathbf{\Lambda}_J, \mathbf{0}_{2Jp\times 2Jp}). \end{aligned} \tag{3.29}$$

Proof: In the transformed space, (3.26) becomes $\tilde{\mathbf{z}}(t) = \tilde{\mathbf{\Omega}}(t,\mu)\tilde{\mathbf{z}}(t-1) + \tilde{\boldsymbol{\nu}}(t)$, where $\tilde{\mathbf{\Omega}}(t,\mu) := \tilde{\mathbf{U}}\mathbf{\Omega}(t,\mu)\tilde{\mathbf{U}}^T$ and $\tilde{\boldsymbol{\nu}}(t) := \tilde{\mathbf{U}}\boldsymbol{\nu}(t)$. Using (a4)-(a6), it follows that $E[\tilde{\mathbf{\Omega}}(t,\mu)\tilde{\mathbf{z}}(t-1)\tilde{\boldsymbol{\nu}}^T(t)] = \mathbf{0}_{3Jp\times 3Jp}$. Therefore, $\mathbf{R}_{\tilde{z}}(t) = E[\tilde{\mathbf{\Omega}}(t,\mu)\mathbf{R}_{\tilde{z}}(t-1)\tilde{\mathbf{\Omega}}^T(t,\mu)] + E[\tilde{\boldsymbol{\nu}}(t)\tilde{\boldsymbol{\nu}}^T(t)]$ and we start by showing that the first expectation is $\mathcal{M}(\tilde{\mathbf{\Omega}}(\mu), \mathbf{R}_{\tilde{z}}(t-1))$. Split $\tilde{\mathbf{\Omega}}(t,\mu) = \bar{\mathbf{\Omega}}(\mu) - 2\mu\text{bdiag}(\mathbf{R}_{\tilde{h}}(t), \mathbf{0}_{2Jp\times 2Jp})$ into its deterministic and random components, and drop for simplicity the $t-1$ argument in $\mathbf{R}_{\tilde{z}}(t-1)$ to obtain

$$\begin{aligned}E[\tilde{\mathbf{\Omega}}(t,\mu)\mathbf{R}_{\tilde{z}}\tilde{\mathbf{\Omega}}^T(t,\mu)] = {} & \bar{\mathbf{\Omega}}(\mu)\mathbf{R}_{\tilde{z}}\bar{\mathbf{\Omega}}^T(\mu) \\ & + 4\mu^2 E[\text{bdiag}(\mathbf{R}_{\tilde{h}}(t), \mathbf{0}_{2Jp\times 2Jp})\mathbf{R}_{\tilde{z}}\text{bdiag}(\mathbf{R}_{\tilde{h}}(t), \mathbf{0}_{2Jp\times 2Jp})] \\ & - 2\mu[\bar{\mathbf{\Omega}}(\mu)\mathbf{R}_{\tilde{z}}\text{bdiag}(\mathbf{\Lambda}, \mathbf{0}_{2Jp\times 2Jp}) + (\bar{\mathbf{\Omega}}(\mu)\mathbf{R}_{\tilde{z}}\text{bdiag}(\mathbf{\Lambda}, \mathbf{0}_{2Jp\times 2Jp}))^T]. \end{aligned} \tag{3.30}$$

The second summand in the right hand side of (3.30) has the structure $4\mu^2\text{bdiag}(\mathbf{A}, \mathbf{0}_{2Jp\times 2Jp})$, where $\mathbf{A} \in \mathbb{R}^{Jp\times Jp}$ can be partitioned into $p \times p$ blocks

$$[\mathbf{A}]_{i,j} = \begin{cases} E[\tilde{\mathbf{h}}_j(t)\tilde{\mathbf{h}}_j^T(t)[\mathbf{R}_{\tilde{z}}]_{11,j}\tilde{\mathbf{h}}_j(t)\tilde{\mathbf{h}}_j^T(t)] = 2\mathbf{\Lambda}_j[\mathbf{R}_{\tilde{z}}]_{11,j}\mathbf{\Lambda}_j + \text{tr}(\mathbf{\Lambda}_j[\mathbf{R}_{\tilde{z}}]_{11,j})\mathbf{\Lambda}_j, & i = j \\ E[\tilde{\mathbf{h}}_i(t)\tilde{\mathbf{h}}_i^T(t)[\mathbf{R}_{\tilde{z}}]_{11,i,j}\tilde{\mathbf{h}}_j(t)\tilde{\mathbf{h}}_j^T(t)] = \mathbf{R}_{h_i}[\mathbf{R}_{\tilde{z}}]_{11,i,j}\mathbf{R}_{h_j}, & i \neq j \end{cases}$$

for $i, j = 1, \ldots, J$. To evaluate the regressor's fourth-order moments in the diagonal blocks of $\mathbf{A}$, we have relied on the Gaussianity of $\tilde{\mathbf{h}}_j(t)\ \forall\, j \in \mathcal{J}$, which follows from (a6). The expectations in the non-diagonal blocks follow immediately as regressors are also assumed spatially uncorrelated. Substituting in (3.30) and regrouping terms one obtains

$$\begin{aligned}\tilde{\mathbf{\Omega}}(\mu)\mathbf{R}_{\check{z}}\tilde{\mathbf{\Omega}}^T(\mu) = {} & \bar{\mathbf{\Omega}}(\mu)\mathbf{R}_{\check{z}}\bar{\mathbf{\Omega}}^T(\mu) + 4\mu^2\text{bdiag}(\mathbf{A} - \text{bdiag}([\mathbf{A}]_{1,1}, \ldots, [\mathbf{A}]_{J,J}), \mathbf{0}_{2Jp\times 2Jp}) \\ & + 4\mu^2\text{bdiag}(\mathbf{\Lambda}_1[\mathbf{R}_{\check{z}}]_{11,1}\mathbf{\Lambda}_1, \ldots, \mathbf{\Lambda}_J[\mathbf{R}_{\check{z}}]_{11,J}\mathbf{\Lambda}_J, \mathbf{0}_{2Jp\times Jp}) \\ & - 2\mu[\bar{\mathbf{\Omega}}(\mu)\mathbf{R}_{\check{z}}\text{bdiag}(\mathbf{\Lambda}, \mathbf{0}_{2Jp\times 2Jp}) + (\bar{\mathbf{\Omega}}(\mu)\mathbf{R}_{\check{z}}\text{bdiag}(\mathbf{\Lambda}, \mathbf{0}_{2Jp\times 2Jp}))^T]\end{aligned}$$

which finally yields

$$\begin{aligned}E[\tilde{\mathbf{\Omega}}(t,\mu)\mathbf{R}_{\check{z}}\tilde{\mathbf{\Omega}}^T(t,\mu)] = {} & \tilde{\mathbf{\Omega}}(\mu)\mathbf{R}_{\check{z}}\tilde{\mathbf{\Omega}}^T(\mu) \\ & + 4\mu^2\text{bdiag}(\mathbf{\Lambda}_1[\mathbf{R}_{\check{z}}]_{11,1}\mathbf{\Lambda}_1, \ldots, \mathbf{\Lambda}_J[\mathbf{R}_{\check{z}}]_{11,J}\mathbf{\Lambda}_J, \mathbf{0}_{2Jp\times Jp}) \\ & + 4\mu^2\text{bdiag}(\text{tr}(\mathbf{\Lambda}_1[\mathbf{R}_{\check{z}}]_{11,1})\mathbf{\Lambda}_1, \ldots, \text{tr}(\mathbf{\Lambda}_J[\mathbf{R}_{\check{z}}]_{11,J})\mathbf{\Lambda}_J, \mathbf{0}_{2Jp\times Jp}).\end{aligned} \tag{3.31}$$

Simple manipulations on the second term in the right hand side of (3.31) lead to the desired result [cf. (3.29)]. Back to the remaining covariance $\mathbf{R}_{\tilde{\boldsymbol{\nu}}} := E[\tilde{\boldsymbol{\nu}}(t)\tilde{\boldsymbol{\nu}}^T(t)]$, because the three noise terms within $\tilde{\boldsymbol{\nu}}(t)$ [cf. (3.25)] are pairwise independent and zero-mean, we have that

$$\begin{aligned}\mathbf{R}_{\tilde{\boldsymbol{\nu}}} = {} & E[\tilde{\mathbf{U}}[\mathbf{\Phi}(t+1,\mu)^T\ \mathbf{0}_{2Jp\times Jp}]^T(\mathbf{R}_{\bar{\boldsymbol{\eta}}_\mu} + \mathbf{R}_{\boldsymbol{\eta}_\mu})(\tilde{\mathbf{U}}[\mathbf{\Phi}(t+1,\mu)^T\ \mathbf{0}_{2Jp\times Jp}]^T)^T] \\ & + 4\mu^2\text{bdiag}(\sigma^2_{\epsilon_1}\mathbf{\Lambda}_1, \ldots, \sigma^2_{\epsilon_J}\mathbf{\Lambda}_J, \mathbf{0}_{2Jp\times 2Jp}) + \tilde{\mathbf{U}}\left(([-1\,0\,1]^T[-1\,0\,1]) \otimes (\mathbf{1}_{J\times J} \otimes \mathbf{R}_\zeta)\right)\tilde{\mathbf{U}}^T\end{aligned} \tag{3.32}$$

where the last two terms follow after using (a4)-(a6), and correspond to the covariance matrices of the second and third vectors in the right hand side of (3.25). The structure of $\mathbf{R}_{\bar{\boldsymbol{\eta}}_\mu}$ and $\mathbf{R}_{\boldsymbol{\eta}_\mu}$ is provided in Appendix C. The first expectation in (3.32) can be treated similarly as $E[\tilde{\mathbf{\Omega}}(t,\mu)\mathbf{R}_{\check{z}}(t-1)\tilde{\mathbf{\Omega}}^T(t,\mu)]$ to yield the second summand in the right hand side of (3.28). □

The covariance recursion in Proposition 3.2 (indirectly) characterizes the exact tracking MSE evolution of the D-LMS algorithm, under the white Gaussian setting assumptions and the vector AR(1) model for $\mathbf{s}_0(t)$. With the appropriate simplifications indicated in (a4.1) and (a4.2), (3.28) enables performance evaluation when the parameter vector of interest, $\mathbf{s}_0(t)$, is either time-invariant or adheres to a random-walk model. For example, under (a4.1) the last matrix in the right hand side of (3.28) vanishes because $\mathbf{R}_{\zeta} = \mathbf{0}_{p\times p}$ while the inner structure of $\tilde{\mathbf{\Omega}}(\mu)$ should be adapted to $\mathbf{\Theta} = \mathbf{0}_{p\times p}$.

Starting from Proposition 2, the recipe towards obtaining the performance metrics described in Section 3.4.2 is the following. Given (3.28) and upon inverting the change of variables to yield $\mathbf{R}_{\check{z}}(t) = \tilde{\mathbf{U}}^T\mathbf{R}_{\tilde{z}}(t)\tilde{\mathbf{U}}$, one can readily extract $[\mathbf{R}_z(t)]_{11}$ as the upper-left $Jp \times Jp$ submatrix of $\mathbf{R}_{\check{z}}(t)$. Closed-form evaluation of the MSE(t), EMSE(t) and MSD(t) for all $t \geq 0$ and every sensor $j \in \mathcal{J}$ is now possible by using (3.24) to obtain $[\mathbf{R}_y(t)]_{11}$, and then resorting to the formulae in Section 3.4.2.

The next step is to reformulate (3.28) into a first-order vector recursion which is better suited for stability analysis. Specifically, (3.28) can be vectorized to obtain $\text{vec}[\mathbf{R}_{\tilde{z}}(t+1)] = \text{vec}[\mathcal{M}(\tilde{\mathbf{\Omega}}(\mu), \mathbf{R}_{\tilde{z}}(t))] + \text{vec}[\mathbf{R}_{\tilde{\boldsymbol{\nu}}}]$. As asserted in the following lemma, further simplification is possible by relying on properties of the matrix vectorization operator [33]. It is shown in Appendix D that:

Lemma 3.3 *Under (a4)-(a6) and for $t \geq 0$, the vectorized covariance matrix of $\tilde{\mathbf{z}}(t)$ obeys the first-order vector recursion given by*

$$vec[\mathbf{R}_{\tilde{z}}(t+1)] = \tilde{\mathbf{\Psi}}(\mu) vec[\mathbf{R}_{\tilde{z}}(t)] + vec[\mathbf{R}_{\tilde{\boldsymbol{\nu}}}]. \tag{3.33}$$

The $(3Jp)^2 \times (3Jp)^2$ transition matrix $\tilde{\mathbf{\Psi}}(\mu)$ is

$$\begin{aligned}\tilde{\mathbf{\Psi}}(\mu) &:= \tilde{\mathbf{\Omega}}(\mu) \otimes \tilde{\mathbf{\Omega}}(\mu) + 4\mu^2 \sum_{j=1}^{J} \mathbf{q}_j \mathbf{q}_j^T \\ &\quad + 4\mu^2 (bdiag(\mathbf{\Lambda}, \mathbf{0}_{2Jp\times 2Jp}) \otimes bdiag(\mathbf{\Lambda}, \mathbf{0}_{2Jp\times 2Jp}))\, diag(vec[\mathbf{I}_{3J} \otimes \mathbf{1}_{p\times p}])\end{aligned} \tag{3.34}$$

where $\mathbf{q}_j := vec[diag(\mathbf{b}_{3J,j}) \otimes \mathbf{\Lambda}_j]\ \ \forall\, j \in \mathcal{J}$.

An immediate consequence of Lemma 3.3 is that the D-LMS algorithm is MSE stable if $\lambda_{\max}(\tilde{\mathbf{\Psi}}(\mu)) < 1$. Although deriving explicit bounds on μ for stability appears intractable, the following proposition provides an important existence result.

Proposition 3.3 *Under (a1), (a4)-(a6) the D-LMS algorithm is MSE stable, i.e., $\lim_{t\to\infty}[\mathbf{R}_y(t)]_{11}$ has bounded entries, provided that $\mu > 0$ is chosen sufficiently small.*

Proof: The eigenvalues of $\tilde{\mathbf{\Omega}}(\mu) \otimes \tilde{\mathbf{\Omega}}(\mu)$ are the pairwise products of those of $\tilde{\mathbf{\Omega}}(\mu)$. From (3.34) it is possible to upper-bound $\lambda_{\max}(\tilde{\mathbf{\Psi}}(\mu)) \leq \lambda_{\max}(\tilde{\mathbf{\Omega}}(\mu))^2 + \kappa_1\mu^2$, with κ_1 a finite positive constant. Given the block upper-triangular structure of $E[\mathbf{\Omega}(t,\mu)]$ [cf. (3.25)] which has the same eigenvalues as $\tilde{\mathbf{\Omega}}(\mu)$, for $\kappa_2 \in (0,\infty)$ one obtains that $\lambda_{\max}(\tilde{\mathbf{\Omega}}(\mu)) = \max(\lambda_{\max}(\mathbf{\Phi}(\mu)), \lambda_{\max}(\mathbf{\Theta})) \sim \max(1-\mu\kappa_2, \lambda_{\max}(\mathbf{\Theta}))$, where the scaling of $\lambda_{\max}(\mathbf{\Phi}(\mu))$ follows from the proof of Lemma 3.2. By virtue of (a4), $\lambda_{\max}(\mathbf{\Theta}) \in [0,1)$ and is independent of μ. Hence, $\lambda_{\max}(\tilde{\mathbf{\Omega}}(\mu)) \sim 1-\mu\kappa_2$ for μ small enough so that $\lambda_{\max}(\tilde{\mathbf{\Psi}}(\mu)) \leq 1-\mu[2\kappa_2 - \mu(\kappa_2^2+\kappa_1)]$, which can be made smaller than one for $\mu > 0$ sufficiently small. This readily implies that $\lim_{t\to\infty}\mathbf{R}_{\check{z}}(t)$ has bounded entries, and can be established also for $\lim_{t\to\infty}[\mathbf{R}_y(t)]_{11}$ via the process described after Proposition 3.2. □

While the proof for Proposition 3.3 is still valid for a time-invariant parameter vector, the argument clearly breaks down for the random-walk model because $\lambda_{\max}(\mathbf{\Theta}) = \lambda_{\max}(\mathbf{I}_p) = 1$. In this case, $\mathbf{R}_{\mathbf{s}_0}(t)$ grows unbounded; thus, one would expect that the same happens to the inner state $\mathbf{z}(t)$. However, note that the coupling between $\mathbf{z}(t)$ and $\mathbf{s}_0(t)$ arising in (3.22) disappears under (a4.2). For this reason, it is possible to reproduce all previous results by working just with $\mathbf{z}(t)$ (instead of $\check{\mathbf{z}}(t)$) to finally conclude that Proposition 3.3 holds true for the random-walk model also [31].

Next, we consider an alternative notion of stochastic stability that can be inferred from Proposition 3.3. Specifically, it is possible to show that under the white Gaussian setting assumptions, the error norm $\|\mathbf{y}_1(t)\|$ remains most of the time in a finite interval, i.e., errors are weakly stochastic bounded (WSB) [51], [52, pg. 110]. This WSB stability guarantees that for any $\theta > 0$ there exists a $\delta > 0$ such that $\Pr[\|\mathbf{y}_1(t)\| < \delta] = 1-\theta$ uniformly in t. It is a weak notion of stability, providing an alternative for the analysis of adaptive filters when the presence of, e.g., time-correlated data, renders variance calculations impossible; see also [48], [51]. Nevertheless, it is an important practical notion as it ensures – on a per-realization basis – that there is no probability mass allowing estimation errors escape to infinity. Similar to Proposition 3.3, this property holds for the D-LMS algorithm in the presence of communication noise.

Proposition 3.4 *Under (a1), (a4)-(a6) and if the step-size $\mu > 0$ is chosen sufficiently* *small, then the D-LMS algorithm yields estimation errors which are WSB; i.e.,*

$$\lim_{\delta\to\infty}\sup_{t\geq 0}\Pr[\|\mathbf{y}_1(t)\|\geq\delta]=0. \tag{3.35}$$

Proof: An application of Chebyshev's inequality yields the bound

$$Pr[\|\mathbf{y}_1(t)\|\geq\delta]\leq\frac{E[\|\mathbf{y}_1(t)\|^2]}{\delta^2}=\frac{\mathrm{tr}([\mathbf{R}_y(t)]_{11})}{\delta^2}. \tag{3.36}$$

From Proposition 3.3, $\lim_{t\to\infty}[\mathbf{R}_y(t)]_{11}$ has bounded entries, which also implies that $\sup_{t\geq 0}\mathrm{tr}([\mathbf{R}_y(t)]_{11})<\infty$. Taking the limit as $\delta\to\infty$, while relying on the bound in (3.36) yields the desired result. □

3.5.3 MSE Performance in Steady-State

Under the stability conditions in Proposition 3.3, the steady-state covariance matrix $\mathbf{R}_{\tilde{z}}(\infty):=\lim_{t\to\infty}\mathbf{R}_{\tilde{z}}(t)$ has bounded entries. Lemma 3.3 enables the evaluation of $\mathrm{vec}[\mathbf{R}_{\tilde{z}}(\infty)]$ as a fixed point of (3.33); thus,

$$\mathrm{vec}[\mathbf{R}_{\tilde{z}}(\infty)]=(\mathbf{I}_{(3Jp)^2}-\tilde{\boldsymbol{\Psi}}(\mu))^{-1}\mathrm{vec}[\mathbf{R}_{\tilde{\boldsymbol{\nu}}}]. \tag{3.37}$$

Note that if D-LMS is MSE stable, i.e., $\tilde{\boldsymbol{\Psi}}(\mu)$ is a stable matrix, matrix $(\mathbf{I}_{(3Jp)^2}-\tilde{\boldsymbol{\Psi}}(\mu))^{-1}$ is guaranteed to exist thanks to Gershgorin's circle theorem. Exactly as before, all relevant local and global figures of merit in steady-state can be evaluated provided $[\mathbf{R}_y(\infty)]_{11}$ is available (cf. Section 3.4.2). Just reshape (3.37) to obtain $\mathbf{R}_{\tilde{z}}(\infty)$, undo the change of variables to extract $\mathbf{R}_z(\infty)$ from $\mathbf{R}_{\tilde{z}}(\infty)$, and finally use (3.24).

While MSE stability ensures, e.g., a bounded EMSE(∞), satisfactory tracking of $\mathbf{s}_0(t)$ ultimately requires the error to be small. This will depend on μ and the speed of parameter variation roughly dictated by $\mathrm{tr}(\mathbf{1}_{J\times J}\otimes\mathbf{R}_\zeta)=J\mathrm{tr}(\mathbf{R}_\zeta)$. For simplicity in exposition, consider in the sequel that communication links are ideal so that $\mathbf{R}_{\bar{\boldsymbol{\eta}}_\mu}=\mathbf{R}_{\boldsymbol{\eta}_\mu}=\mathbf{0}_{2Jp\times 2Jp}$ in (3.28). Interestingly, whenever $J\mathrm{tr}(\mathbf{R}_\zeta)$ is comparable to $4\mu^2\sum_{j=1}^{J}\sigma_{\epsilon_j}^2\mathrm{tr}(\boldsymbol{\Lambda}_j)$, there exists an optimal $\mu^\star$ minimizing EMSE(∞); see also the numerical examples in Section 3.6. Because the latter term is $\mathcal{O}(\mu^2)$, $\mathrm{tr}(\mathbf{R}_\zeta)$ should also be small to ensure that EMSE(∞) has an acceptable level. This further implies that D-LMS can track satisfactorily slowly time-varying processes. Inevitable communication-induced delays will affect the D-LMS algorithm, and may further limit the tracking capabilities of the proposed scheme. However, delay analysis falls beyond the scope of the present paper.

The existence of a $\mu^\star$ should not be surprising, given the known results for the centralized LMS algorithm, [43, pg. 367], [52, pg. 123]. Excessive adaptation leads to the same MSE inflation as in the absence of parameter variation, while if μ is too small the tracking ability may be lost and once again an MSE penalty is expected. To gain some insight into this tradeoff for the D-LMS algorithm, recall from Section 3.4.2 and (3.37) that

$$\begin{aligned}\text{EMSE}(\infty) &= \frac{1}{J}\sum_{j=1}^{J}\text{tr}(\mathbf{R}_{h_j}[\mathbf{R}_y(\infty)]_{11,j}) = \frac{1}{J}\sum_{j=1}^{J}\text{tr}(\boldsymbol{\Lambda}_j[\mathbf{R}_{\tilde{z}}(\infty)]_{11,j})\\ &= \frac{1}{J}\sum_{j=1}^{J}\mathbf{q}_j^T\text{vec}[\mathbf{R}_{\tilde{z}}(\infty)] = \frac{1}{J}\sum_{j=1}^{J}\mathbf{q}_j^T(\mathbf{I}_{(3Jp)^2}-\tilde{\boldsymbol{\Psi}}(\mu))^{-1}\text{vec}[\mathbf{R}_{\tilde{\boldsymbol{\nu}}}] \qquad (3.38)\end{aligned}$$

where in obtaining the third equality we used that $\text{tr}(\mathbf{R}^T\mathbf{S}) = \text{vec}[\mathbf{R}]^T\text{vec}[\mathbf{S}]$, and the $\{\mathbf{q}_j\}_{j=1}^J$ were defined as in Lemma 3.3. Now, in the absence of communication noise (cf. (3.28))

$$\mathbf{R}_{\tilde{\boldsymbol{\nu}}} = 4\mu^2\text{bdiag}(\sigma_{\epsilon_1}^2\boldsymbol{\Lambda}_1,\ldots,\sigma_{\epsilon_J}^2\boldsymbol{\Lambda}_J,\mathbf{0}_{2Jp\times 2Jp}) + \tilde{\mathbf{U}}\left(([-1\;0\;1]^T[-1\;0\;1])\otimes(\mathbf{1}_{J\times J}\otimes\mathbf{R}_\zeta)\right)\tilde{\mathbf{U}}^T$$

so that the term due to observation noise is $\mathcal{O}(\mu^2)$, and the second summand due to parameter nonstationarities is $\mathcal{O}(1)$. Roughly, $(\mathbf{I}_{(3Jp)^2}-\tilde{\boldsymbol{\Psi}}(\mu))^{-1} = \mathcal{O}(\mu^{-1})$ and one finds from (3.38) that $\text{EMSE}(\infty) = \mathcal{O}(\mu^{-1})$ for small μ, whereas $\text{EMSE}(\infty) = \mathcal{O}(\mu)$ for moderate- to large values of the step-size approaching the stability bound. This advocates the existence of an optimal step-size $\mu^\star$ minimizing the steady-state EMSE. Unfortunately, deriving an explicit formula for $\mu^\star$ is a formidable task. If needed however, $1-$D minimization can be carried out numerically using, e.g., Newton's method, as the derivatives of the $\text{EMSE}(\infty)$ cost in (3.38) are readily computable in closed form.

If $J\text{tr}(\mathbf{R}_\zeta) \gg 4\mu^2\sum_{j=1}^J\sigma_{\epsilon_j}^2\text{tr}(\boldsymbol{\Lambda}_j)$, then $\mathbf{R}_{\tilde{\boldsymbol{\nu}}} \approx \tilde{\mathbf{U}}\left(([-1\;0\;1]^T[-1\;0\;1])\otimes(\mathbf{1}_{J\times J}\otimes\mathbf{R}_\zeta)\right)\tilde{\mathbf{U}}^T$ in the whole range of stable step-sizes so that $\text{EMSE}(\infty) = \mathcal{O}(\mu^{-1})$, and will not attain a minimum. To achieve the best tracking performance in this scenario, the step-size should be chosen as large as possible while ensuring stability. The other extreme

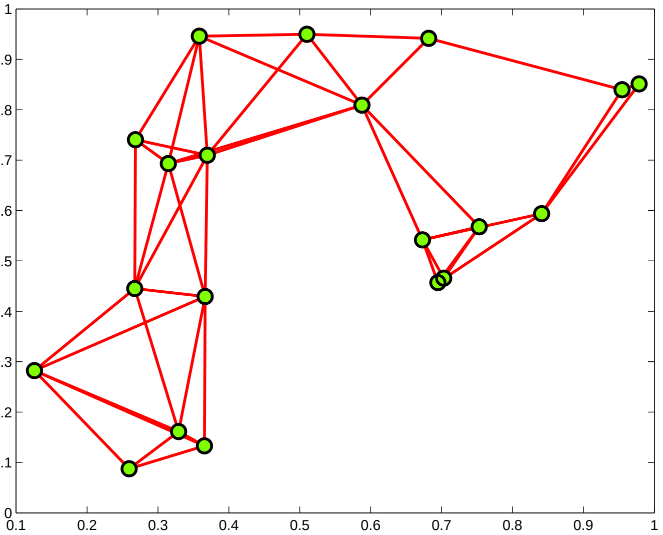

Figure 3.1: An ad hoc WSN with $J = 20$ sensors.

$J\text{tr}(\mathbf{R}_\zeta) \ll 4\mu^2 \sum_{j=1}^{J} \sigma_{\epsilon_j}^2 \text{tr}(\mathbf{\Lambda}_j)$ corresponds to a small degree of nonstationarity, which in the limit $\mathbf{R}_\zeta \rightarrow \mathbf{0}_{p\times p}$ leads to the time-invariant parameter model in (a4.1). Then, $\mathbf{R}_{\tilde{\boldsymbol{\nu}}} \approx 4\mu^2 \text{bdiag}(\sigma_{\epsilon_1}^2 \mathbf{\Lambda}_1, \ldots, \sigma_{\epsilon_J}^2 \mathbf{\Lambda}_J, \mathbf{0}_{2Jp\times 2Jp})$, and as expected $\text{EMSE}(\infty) = \mathcal{O}(\mu)$. The steady-state error can be reduced as much as needed by choosing μ sufficiently small, but this comes at the price of reduced convergence rates.

3.6 Numerical Tests

Here we corroborate the analytical results of Section 3.5 through numerical experiments. Substantiating the comments in Remark 3.4, the usefulness of the analysis is corroborated as the results extend accurately beyond the white Gaussian data setting, allowing for correlated data provided the step-size is small enough. For $J = 20$ sensors, a connected ad hoc WSN is generated as a realization of the random geometric graph model on the unity square, with communication range $r = 0.3$. Hence, sensors are deployed uniformly at random over $[0, 1]^2$ and an edge joining two sensors is included in $\mathcal{E}$ whenever their Euclidean distance does not exceed r; see Fig. 3.1. To model noisy links, additive white Gaussian noise (AWGN) with variance $\sigma_\eta^2 = 10^{-2}$ is added at the receiving end.

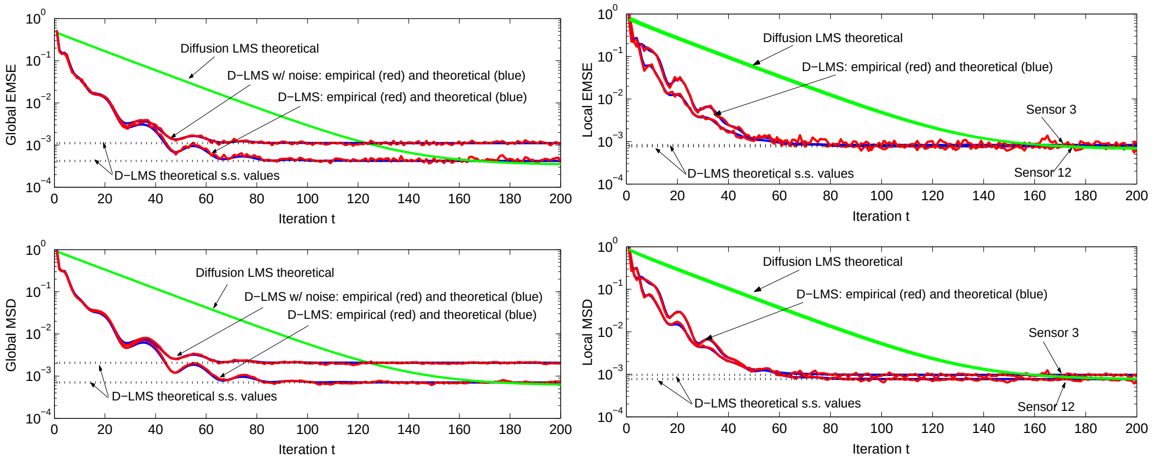

Figure 3.2: (left) Global performance evaluation for a time-invariant parameter; (right) Local performance evaluation for a time-invariant parameter: sensors 3 and 12.

With $p = 4$, observations obey a linear model [cf. (a1)] with sensing WGN of spatial variance profile $\sigma^2_{\epsilon_j} = 10^{-1}\alpha_j$, where $\alpha_j \sim \mathcal{U}[0,1]$ (uniform distribution) and i.i.d.. The regression vectors $\mathbf{h}_j(t) = [h_j(t) \ldots h_j(t-p+1)]^T$ have a shift structure and entries which evolve according to $h_j(t) = (1-\rho)\beta_j h_j(t-1) + \sqrt{\rho}\omega_j(t)$ for all $j \in \mathcal{J}$. We choose $\rho = 5\times10^{-1}$, the $\beta_j \sim \mathcal{U}[0,1]$ i.i.d. in space, and the driving white noise $\omega_j(t) \sim \mathcal{U}[-\sqrt{3}\sigma_{\omega_j}, \sqrt{3}\sigma_{\omega_j}]$ has a spatial variance profile given by $\sigma^2_{\omega_j} = 2\gamma_j$ with $\gamma_j \sim \mathcal{U}[0,1]$ and i.i.d.. The local regressor's covariance matrices $\mathbf{R}_{h_j}$ have symmetric Toeplitz structure, whereby the elements on the i-th diagonal are $[\mathbf{R}_{h_j}]_{i+l,l} = \frac{[(1-\rho)\beta_j]^i \rho\sigma^2_{\omega_j}}{1-[(1-\rho)\beta_j]^2}$ for $i = 0, 1, \ldots, p-1$ and $1 \leq i+l \leq p$ ($i = 0$ corresponds to the main diagonal). Observe that the data is temporally-correlated and non-Gaussian, implying that (a6) does not hold here. Two test cases will be considered with regards to the nature of $\mathbf{s}_0(t)$:

TC1: Large-amplitude slowly time-varying parameters adhering to (a4) with $\mathbf{s}_0 = \mathbf{0}_p$ and $\mathbf{\Theta} = (1-10^{-4})\text{diag}(\theta_1, \ldots, \theta_p)$ with $\theta_i \sim \mathcal{U}[0,1]$ for $i = 1, \ldots, p$. The driving noise is normally distributed with $\mathbf{R}_\zeta = 10^{-4}\mathbf{I}_p$.

TC2: Time-invariant parameters adhering to (a4.1), with $\mathbf{s}_0 = \mathbf{1}_p$.

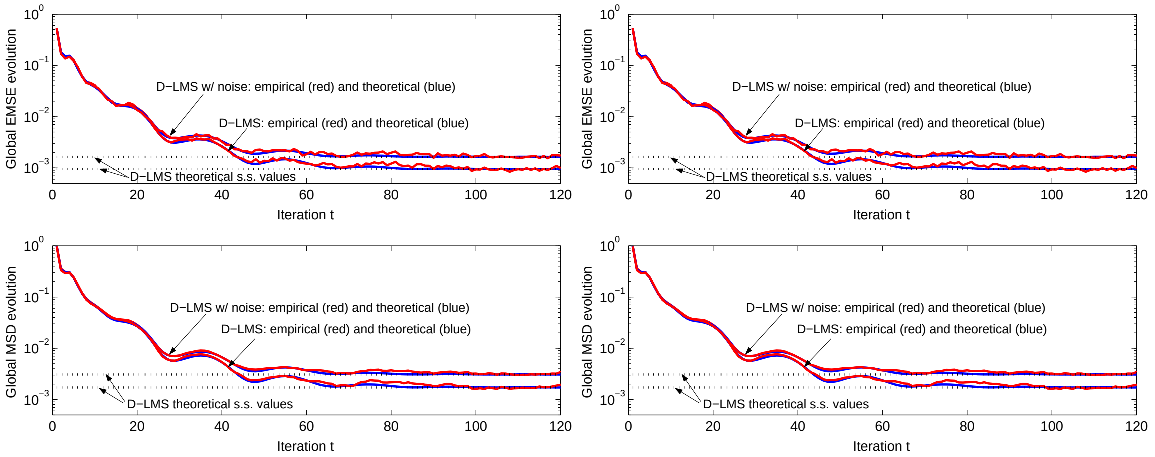

Figure 3.3: (left) Global performance evaluation for a time-varying parameter; (right) Local performance evaluation for a time-varying parameter: sensors 3 and 12.

For all experimental performance curves obtained by running the algorithms, the ensemble averages are approximated via sample averaging 500 runs of the experiment.

First, under TC2 and $\mu = 5 \times 10^{-2}$, $c = 1$ for D-LMS, Fig. 3.2 (left) depicts the network performance through the evolution of EMSE(t) and MSD(t) figures of merit. Both noisy and ideal links are considered, while for the latter case the D-LMS variant in Section 3.3.2 has been used. Even though the simulated data does not adhere to (a6), the empirical curves closely follow the theoretical trajectories evaluated via Proposition 3.2 (and the formulae in Section 3.4.2). The steady-state limiting values found in Section 3.5.3 are also extremely accurate. As intuitively expected and analytically corroborated via the noise-related additive terms in (3.24) and (3.28), the performance penalty due to non-ideal links is also apparent. Theoretical error trajectory curves for the diffusion LMS [28, eqs. (73)-(74)] with Metropolis combining weights are also included. While in this case diffusion LMS has a slight edge on steady-state performance, note that it comes at the price of a much slower convergence rate. Similar overall conclusions can be drawn from the plots in Fig. 3.2 (right), that gauge local performance of two randomly selected representative sensors. Even though the noise levels of both sensors are dissimilar ($\sigma^2_{\epsilon_3} = 7.2 \times 10^{-2}$ and

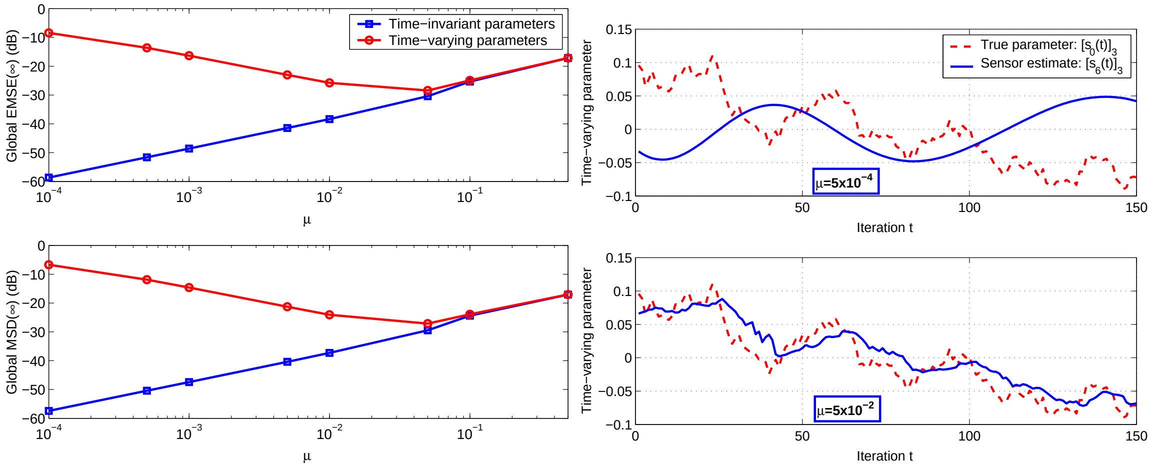

Figure 3.4: (left) Global steady-state EMSE and MSD versus step-size dependencies; (right) Tracking with D-LMS: slow and optimal adaptation levels.

$\sigma^2_{\epsilon_{12}} = 2.3 \times 10^{-2}$), effective percolation of information across the WSN renders the s.s performance of both sensors very simliar. The curves for D-LMS with noisy links have been removed in the interest of clarity.

Turning our attention to the tracking performance of the D-LMS algorithm, Fig. 3.2 is reproduced under TC1 as Fig. 3.3. Once more, it is appealing how well the theoretical findings in Section 3.5.2 agree with the true behavior for all $t \geq 0$. Curves for diffusion LMS are not included as time-varying parameters have not been considered in [28]. To conclude, Fig. 3.4 (left) corroborates the discussion in Section 3.5.3, by showing the theoretically assessed dependence of the steady-state global quantities EMSE(∞) and MSD(∞) on μ, under both TC1 and TC2. While the trend is similar for moderate- to large step-sizes, for small μ the MSE penalty in the tracking setup due to lack of adaptation becomes dominant, and is increasingly severe as $\mu \to 0$. The existence of $\mu^\star \approx 5 \times 10^{-2}$ is also highlighted by Fig. 3.4 (left). From another perspective, Fig. 3.4 (right) illustrates how the adaptation level affects the resulting per-sensor estimates when tracking time-varying parameters with D-LMS. Under TC1 and for $\mu = 5 \times 10^{-4}$ [slow adaptation; see also Fig. 3.4 (left)] and

$\mu = 5 \times 10^{-2}$ (near optimal adaptation), we depict the third entry of the parameter vector $[\mathbf{s}_0(t)]_3$ and the respective estimates from the randomly chosen sixth sensor. Under optimal adaptation the sensor estimate closely follows the true variations, while – as expected – for the smaller step-size D-LMS fails to provide an accurate estimate.

3.7 Appendices

3.7.1 Proof of Equations (3.9)-(3.10)

Starting with [S3], observe that the per sensor decomposable structure of (3.2) is also present in the the augmented Lagrangian. Thus, (3.8) decouples into $\sum_{j=1}^{J} |\mathcal{N}_j|$ quadratic sub-problems

$$\mathbf{z}_j^{j'}(t+1) = \arg\min_{\mathbf{z}_j^{j'}} \left[-[\mathbf{v}_j^{j'}(t) + \mathbf{u}_{j'}^{j}(t)]^T \mathbf{z}_j^{j'} + \frac{c}{2}\left[\|\mathbf{s}_j(t+1) - \mathbf{z}_j^{j'}\|^2 + \|\mathbf{s}_{j'}(t+1) - \mathbf{z}_j^{j'}\|^2 \right] \right] \tag{3.39}$$

which admit the closed-form solutions

$$\mathbf{z}_j^{j'}(t+1) = \frac{1}{2c}[\mathbf{v}_j^{j'}(t) + \mathbf{u}_{j'}^{j}(t)]^T + \frac{1}{2}\left[\mathbf{s}_j(t+1) + \mathbf{s}_{j'}(t+1)\right], \quad j \in \mathcal{J}, \;\; j' \in \mathcal{N}_j. \tag{3.40}$$

Using (3.40) to eliminate $\mathbf{z}_j^{j'}(t)$ and $\mathbf{z}_{j'}^{j}(t)$ from (3.5) and (3.6) respectively, a simple induction argument establishes that if the initial Lagrange multipliers obey $\mathbf{v}_j^{j'}(-1) = -\mathbf{u}_{j'}^{j}(-1)$, then $\mathbf{v}_j^{j'}(t) = -\mathbf{u}_{j'}^{j}(t)$ for all $t \geq 0$ where $j \in \mathcal{J}$ and $j' \in \mathcal{N}_j$. The set $\boldsymbol{u}$ of multipliers has been shown redundant, and (3.40) readily simplifies to

$$\mathbf{z}_j^{j'}(t+1) = \frac{1}{2}\left[\mathbf{s}_j(t+1) + \mathbf{s}_{j'}(t+1)\right], \quad j \in \mathcal{J}, \;\; j' \in \mathcal{N}_j. \tag{3.41}$$

The symmetry in (3.41) implies that $\mathbf{z}_j^{j'}(t) = \mathbf{z}_{j'}^{j}(t)$ for all $t \geq 0$. Upon substituting (3.41) in (3.5), the validity of (3.9) follows readily.

Next, observe that the optimization (3.7) in [S2] can be split into J sub-problems

$$\begin{aligned}\mathbf{s}_j(t+1) = \arg\min_{\mathbf{s}_j} \Bigg[& E\left[(x_j(t+1) - \mathbf{h}_j^T(t+1)\mathbf{s}_j)^2\right] + \sum_{j' \in \mathcal{N}_j} [\mathbf{v}_j^{j'}(t) + \mathbf{u}_j^{j'}(t)]^T \mathbf{s}_j \\ & + \frac{c}{2} \sum_{j' \in \mathcal{N}_j} \left[\|\mathbf{s}_j - \mathbf{z}_j^{j'}(t)\|^2 + \|\mathbf{s}_j - \mathbf{z}_{j'}^{j}(t)\|^2 \right] \Bigg].\end{aligned} \tag{3.42}$$

To arrive at (3.10), use the identities: i) $\mathbf{u}_j^{j'}(t) = -\mathbf{v}_{j'}^{j}(t)$ to eliminate $\mathbf{u}_j^{j'}(t)$ from (3.42); and ii) $\mathbf{z}_j^{j'}(t) = \mathbf{z}_{j'}^{j}(t)$ to recognize that the two quadratic terms in the last summand of (3.42) are identical, while $\mathbf{z}_j^{j'}(t)$ can be eliminated using (3.41). □

3.7.2 Proof of Lemma 3.1

Introduce first the $Jp \times 1$ communication noise supervectors $\boldsymbol{\eta}_\alpha(t) := [(\boldsymbol{\eta}_1^\alpha(t))^T \dots (\boldsymbol{\eta}_J^\alpha(t))^T]^T$ and $\boldsymbol{\eta}_\beta(t) := [(\boldsymbol{\eta}_1^\beta(t))^T \dots (\boldsymbol{\eta}_J^\beta(t))^T]^T$, where for $j \in \mathcal{J}$

$$\boldsymbol{\eta}_j^\alpha(t) := \frac{c}{2} \sum_{j' \in \mathcal{N}_j} \boldsymbol{\eta}_j^{j'}(t), \quad \boldsymbol{\eta}_j^\beta(t) := \frac{c}{2} \sum_{j' \in \mathcal{N}_j} \boldsymbol{\eta}_{j'}^{j}(t). \tag{3.43}$$

In order to relate these vectors with $\boldsymbol{\eta}(t)$ in (3.19), we introduce two $Jp \times (\sum_{j=1}^J |\mathcal{N}_j|)p$ matrices $\mathbf{P}_\alpha := [\mathbf{p}_1 \dots \mathbf{p}_J]^T$ and $\mathbf{P}_\beta := [\mathbf{p}_1' \dots \mathbf{p}_J']^T$. The $(\sum_{j=1}^J |\mathcal{N}_j|)p \times p$ submatrices $\mathbf{p}_j$, $\mathbf{p}_j'$ are given by $\mathbf{p}_j := [(\mathbf{p}_{j,1})^T \dots (\mathbf{p}_{j,J})^T]^T$ and $\mathbf{p}_j' := [(\mathbf{p}_{j,1}')^T \dots (\mathbf{p}_{j,J}')^T]^T$, with $\mathbf{p}_{j,r}, \mathbf{p}_{j',r}$ defined for $r = 1, \dots, J$ as

$$\mathbf{p}_{j,r}^T := \begin{cases} \frac{c}{2}\mathbf{b}_{|\mathcal{N}_r|,r(j)}^T \otimes \mathbf{I}_p & \text{if } j \in \mathcal{N}_r \\ \mathbf{0}_{p \times |\mathcal{N}_r|p} & \text{if } j \notin \mathcal{N}_r \end{cases}$$

$$(\mathbf{p}_{j,r}')^T := \begin{cases} \frac{c}{2}\mathbf{1}_{1 \times |\mathcal{N}_r|} \otimes \mathbf{I}_p & \text{if } r = j \\ \mathbf{0}_{p \times |\mathcal{N}_r|p} & \text{if } r \neq j \end{cases}.$$

Note that $r(j) \in \{1, \dots, |\mathcal{N}_r|\}$ denotes the order in which $\boldsymbol{\eta}_j^r(t)$ appears in $\{\boldsymbol{\eta}_{j'}^r(t)\}_{j' \in \mathcal{N}_r}$ [cf. (3.19)]. It is straightforward to verify that $\boldsymbol{\eta}_\alpha(t) = \mathbf{P}_\alpha \boldsymbol{\eta}(t)$ and $\boldsymbol{\eta}_\beta(t) = \mathbf{P}_\beta \boldsymbol{\eta}(t)$.

The proof entails two steps, the first one being summarized in the following lemma:

Lemma 3.4 *Under (a1) and for $t \geq 0$, the global state $\mathbf{y}(t)$ evolves according to*

$$\mathbf{y}(t+1) = \boldsymbol{\Upsilon}(t+1,\mu)\mathbf{y}(t) + \begin{bmatrix} \boldsymbol{\epsilon}(t+1) \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} \mathbf{1}_J \otimes (\mathbf{s}_0(t+1) - \mathbf{s}_0(t)) \\ \mathbf{0} \end{bmatrix}$$
$$+ \begin{bmatrix} \mu\mathbf{I}_{Jp} \\ \mathbf{0} \end{bmatrix} \bar{\boldsymbol{\eta}}(t) + \begin{bmatrix} \mu 3\mathbf{I}_{Jp} \\ -\mathbf{I}_{Jp} \end{bmatrix} \boldsymbol{\eta}_\alpha(t) - \begin{bmatrix} \mu\mathbf{I}_{Jp} \\ -\mathbf{I}_{Jp} \end{bmatrix} \boldsymbol{\eta}_\beta(t), \tag{3.44}$$

where the $2Jp \times 2Jp$ transition matrix $\boldsymbol{\Upsilon}(t,\mu)$ consists of the $Jp \times Jp$ blocks $[\boldsymbol{\Upsilon}(t,\mu)]_{11} = \mathbf{I}_{Jp} - 2\mu(\mathbf{R}_h(t) + \mathbf{L}_c)$, $[\boldsymbol{\Upsilon}(t,\mu)]_{12} = -\mu\mathbf{I}_{Jp}$, $[\boldsymbol{\Upsilon}(t,\mu)]_{21} = \mathbf{L}_c$ and $[\boldsymbol{\Upsilon}(t,\mu)]_{22} = \mathbf{I}_{Jp}$. The initial condition $\mathbf{y}(0)$ should be selected as $\mathbf{y}(0) = bdiag(\mathbf{I}_{Jp}, \mathbf{L}_c)\mathbf{y}'(0)$, where $\mathbf{y}'(0)$ is any vector in $\mathbb{R}^{2Jp}$.

Proof: After summing $\mathbf{v}_j^{j'}(t) - \mathbf{v}_{j'}^{j}(t)$ over $j' \in \mathcal{N}_j$, it follows from (3.13) that for all $j \in \mathcal{J}$

$$\begin{aligned}
\mathbf{y}_{2,j}(t+1) &:= \sum_{j' \in \mathcal{N}_j} (\mathbf{v}_j^{j'}(t) - \mathbf{v}_{j'}^{j}(t)) \\
&= \mathbf{y}_{2,j}(t) + c \sum_{j' \in \mathcal{N}_j} (\mathbf{s}_j(t) - \mathbf{s}_{j'}(t)) - \frac{c}{2} \sum_{j' \in \mathcal{N}_j} \left(\boldsymbol{\eta}_j^{j'}(t) - \boldsymbol{\eta}_{j'}^{j}(t) \right) \qquad (3.45) \\
&= \mathbf{y}_{2,j}(t) + c \sum_{j' \in \mathcal{N}_j} (\mathbf{y}_{1,j}(t) - \mathbf{y}_{1,j'}(t)) - \boldsymbol{\eta}_j^{\alpha}(t) + \boldsymbol{\eta}_j^{\beta}(t), \qquad (3.46)
\end{aligned}$$

where the last equality was obtained after adding and subtracting $c|\mathcal{N}_j|\mathbf{s}_0(t)$ from the right hand side of (3.45), and relying on the definitions in (4.32). Next, starting from (3.14) and upon: i) using (a1) to eliminate $e_j(t+1) = -\mathbf{h}_j^T(t+1)\mathbf{y}_{1,j}(t) + \epsilon_j(t+1)$ from (3.14); ii) subtracting $\mathbf{s}_0(t+1) + \mathbf{s}_0(t)$ from both sides of (3.14); iii) replacing the sums of noise vectors with the quantities defined in (3.18) and (4.32); and iv) recognizing $\mathbf{y}_{2,j}(t+1)$ in the right hand side of (3.14) and substituting it with (3.46), one arrives at

$$\begin{aligned}
\mathbf{y}_{1,j}(t+1) = \mathbf{y}_{1,j}(t) + \mu &\left[-2\mathbf{h}_j(t+1)\mathbf{h}_j^T(t+1)\mathbf{y}_{1,j}(t) - \mathbf{y}_{2,j}(t) - 2c \sum_{j' \in \mathcal{N}_j} (\mathbf{y}_{1,j}(t) - \mathbf{y}_{1,j'}(t)) \right] \\
&+ 3\mu\boldsymbol{\eta}_j^{\alpha}(t) - \mu\boldsymbol{\eta}_j^{\beta}(t) + \mu\bar{\boldsymbol{\eta}}_j(t) + 2\mu\mathbf{h}_j(t+1)\epsilon_j(t+1) - (\mathbf{s}_0(t+1) - \mathbf{s}_0(t)). \qquad (3.47)
\end{aligned}$$

Again, the term $2c\sum_{j' \in \mathcal{N}_j}(\mathbf{y}_{1,j}(t) - \mathbf{y}_{1,j'}(t))$ inside the square brackets is obtained in error-form after adding and subtracting $2c|\mathcal{N}_j|\mathbf{s}_0(t)$.

What remains to be shown is that after stacking the recursions (3.47) and (3.46) for $j = 1, \ldots, J$ to form the one for $\mathbf{y}(t+1)$, we can obtain the compact representation in (3.44). Consider first the forcing terms in (3.44). Stacking the channel noise terms from (3.47) and (3.46), readily yields the last three terms in (3.44). Likewise, independently stacking the terms $2\mu\mathbf{h}_j(t+1)\epsilon_j(t+1)$ for $j = 1, \ldots, J$ and $\mathbf{s}_0(t+1) - \mathbf{s}_0(t)$ yields the second and third terms in the right hand side of (3.44), which are due to the observation noise and parameter velocity, respectively. These terms are not present in (3.46), which explains the zero vectors at the lower part of the second and third terms in (3.44).

To specify the structure of the transition matrix $\boldsymbol{\Upsilon}(t,\mu)$, note that the first term on the right hand side of (3.46) explains why $[\boldsymbol{\Upsilon}(t,\mu)]_{22} = \mathbf{I}_{Jp}$. Similarly, the second term inside the square brackets in (3.47) explains why $[\boldsymbol{\Upsilon}(t,\mu)]_{12} = -\mu\mathbf{I}_{Jp}$. Next, it follows readily that upon stacking the terms $c\sum_{j'\in\mathcal{N}_j}(\mathbf{y}_{1,j}(t) - \mathbf{y}_{1,j'}(t))$, which correspond to a scaled Laplacian-based combination of $p \times 1$ vectors, one obtains $c(\mathbf{L}\otimes\mathbf{I}_p)\mathbf{y}_1(t) = \mathbf{L}_c\mathbf{y}_1(t)$. This justifies why $[\boldsymbol{\Upsilon}(t,\mu)]_{21} = \mathbf{L}_c$. Using similar arguments and recalling that $\mathbf{R}_h(t) :=$ bdiag$(\mathbf{h}_1(t)\mathbf{h}_1^T(t),\ldots,\mathbf{h}_J(t)\mathbf{h}_J^T(t))$, we establish that $[\boldsymbol{\Upsilon}(t,\mu)]_{11} = \mathbf{I}_{Jp} - 2\mu(\mathbf{R}_h(t) + \mathbf{L}_c)$.

Although the vectors $\{\mathbf{y}_{1,j}(0)\}_{j=1}^J$ are decoupled so that $\mathbf{y}_1(0)$ can be chosen arbitrarily, this is not the case for $\{\mathbf{y}_{2,j}(0)\}_{j=1}^J$ which are coupled and satisfy

$$\sum_{j=1}^{J}\mathbf{y}_{2,j}(t) = \sum_{j=1}^{J}\sum_{j'\in\mathcal{N}_j}\left(\mathbf{v}_j^{j'}(t-1) - \mathbf{v}_{j'}^{j}(t-1)\right) = \mathbf{0}_p, \quad \forall\, t \geq 0. \tag{3.48}$$

The coupling across $\{\mathbf{y}_{2,j}(t)\}_{j=1}^J$ dictates that $\mathbf{y}_2(0)$ should be chosen in compliance with (3.48), so that the system (3.44) is equivalent to D-LMS for all $t \geq 0$. Let $\mathbf{y}_2(0) = \mathbf{L}_c\mathbf{y}_2'(0)$, where $\mathbf{y}_2'(0)$ is any vector in $\mathbb{R}^{Jp}$. Then, $\mathbf{y}_2(0)$ satisfies the conservation law (3.48) as (recall that $\mathbf{1}_J = \text{nullspace}(\mathbf{L})$)

$$\sum_{j=1}^{J}\mathbf{y}_{2,j}(0) = (\mathbf{1}_J^T\otimes\mathbf{I}_p)\mathbf{y}_2(0) = (\mathbf{1}_J^T\otimes\mathbf{I}_p)c(\mathbf{L}\otimes\mathbf{I}_p)\mathbf{y}_2'(0) = c((\mathbf{1}_J^T\mathbf{L})\otimes\mathbf{I}_p)\mathbf{y}_2'(0) = \mathbf{0}_p. \tag{3.49}$$

In conclusion, for arbitrary $\mathbf{y}'(0) \in \mathbb{R}^{2Jp}$ the recursion (3.44) should be initialized as $\mathbf{y}(0) =$ bdiag$(\mathbf{I}_{Jp},\mathbf{L}_c)\mathbf{y}'(0)$, and the proof of Lemma 3.4 is completed.

The second step of the proof involves establishing the equivalence between the dynamical systems in (3.44) and (3.20) for all $t \geq 0$, when the inner state is arbitrarily initialized as $\mathbf{z}(0) = \mathbf{y}'(0)$. We will argue by induction. For $t = 0$, it follows from (3.21) that $\mathbf{z}(1) = \boldsymbol{\Phi}(1,\mu)\mathbf{y}'(0) + [\boldsymbol{\epsilon}^T(1)\ \mathbf{0}^T]^T - [\mathbf{1}_J^T\otimes(\mathbf{s}_0(1) - \mathbf{s}_0(0))^T\ \mathbf{0}^T]^T$. Upon substituting $\mathbf{z}(1)$ into (3.20), we find

$$\begin{aligned}\mathbf{y}(1) = \text{bdiag}(\mathbf{I}_{Jp},\mathbf{L}_c)\boldsymbol{\Phi}(1,\mu)\mathbf{y}'(0) + \begin{bmatrix}\boldsymbol{\epsilon}(1)\\ \mathbf{0}\end{bmatrix} - \begin{bmatrix}\mathbf{1}_J\otimes(\mathbf{s}_0(1) - \mathbf{s}_0(0))\\ \mathbf{0}\end{bmatrix}\\ + \begin{bmatrix}\mu\mathbf{I}_{Jp}\\ \mathbf{0}\end{bmatrix}\bar{\boldsymbol{\eta}}(0) + \begin{bmatrix}\mu(3\mathbf{P}_\alpha - \mathbf{P}_\beta)\\ \mathbf{P}_\beta - \mathbf{P}_\alpha\end{bmatrix}\boldsymbol{\eta}(0).\end{aligned} \tag{3.50}$$

Note that: i) bdiag$(\mathbf{I}_{Jp},\mathbf{L}_c)\boldsymbol{\Phi}(t,\mu) = \boldsymbol{\Upsilon}(t,\mu)$bdiag$(\mathbf{I}_{Jp},\mathbf{L}_c)$ for all $t \geq 1$; ii) $\mathbf{y}(0) =$ bdiag$(\mathbf{I}_{Jp},\mathbf{L}_c)\mathbf{y}'(0)$ for the system in Lemma 3.4; and iii) $\boldsymbol{\eta}_\alpha(t) = \mathbf{P}_\alpha\boldsymbol{\eta}(t)$, while $\boldsymbol{\eta}_\beta(t) =$

$\mathbf{P}_\beta \boldsymbol{\eta}(t)$. Thus, the right hand side of (3.50) is equal to the right hand side of (3.44) for $t = 0$.

Suppose next that (3.20) and (3.21) hold true for $\mathbf{y}(t)$ and $\mathbf{z}(t)$. The same will be shown for $\mathbf{y}(t{+}1)$ and $\mathbf{z}(t{+}1)$. To this end, replace $\mathbf{y}(t)$ with the right hand side of (3.20) evaluated at time instant t, into (3.44) to obtain

$$\begin{aligned}
\mathbf{y}(t+1) &= \boldsymbol{\Upsilon}(t+1,\mu)\mathrm{bdiag}(\mathbf{I}_{Jp},\mathbf{L}_c)\mathbf{z}(t) + \boldsymbol{\Upsilon}(t+1,\mu)\begin{bmatrix}\mu\mathbf{I}_{Jp}\\ \mathbf{0}\end{bmatrix}\bar{\boldsymbol{\eta}}(t-1)\\
&+ \boldsymbol{\Upsilon}(t+1,\mu)\begin{bmatrix}\mu(3\mathbf{P}_\alpha-\mathbf{P}_\beta)\\ \mathbf{P}_\beta-\mathbf{P}_\alpha\end{bmatrix}\boldsymbol{\eta}(t-1)\\
&+ \begin{bmatrix}\boldsymbol{\epsilon}(t+1)\\ \mathbf{0}\end{bmatrix} - \begin{bmatrix}\mathbf{1}_J\otimes(\mathbf{s}_0(t+1)-\mathbf{s}_0(t))\\ \mathbf{0}\end{bmatrix}\\
&+ \begin{bmatrix}\mu\mathbf{I}_{Jp}\\ \mathbf{0}\end{bmatrix}\bar{\boldsymbol{\eta}}(t) + \begin{bmatrix}\mu3\mathbf{I}_{Jp}\\ -\mathbf{I}_{Jp}\end{bmatrix}\boldsymbol{\eta}_\alpha(t) - \begin{bmatrix}\mu\mathbf{I}_{Jp}\\ -\mathbf{I}_{Jp}\end{bmatrix}\boldsymbol{\eta}_\beta(t)\\
&= \mathrm{bdiag}(\mathbf{I}_{Jp},\mathbf{L}_c)\Bigg(\boldsymbol{\Phi}(t+1,\mu)\mathbf{z}(t) + \begin{bmatrix}\boldsymbol{\epsilon}(t+1)\\ \mathbf{0}\end{bmatrix} - \begin{bmatrix}\mathbf{1}_J\otimes(\mathbf{s}_0(t+1)-\mathbf{s}_0(t))\\ \mathbf{0}\end{bmatrix}\\
&+\boldsymbol{\Phi}(t+1,\mu)\begin{bmatrix}\mu\mathbf{I}_{Jp}\\ \mathbf{0}\end{bmatrix}\bar{\boldsymbol{\eta}}(t-1) + \boldsymbol{\Phi}(t+1,\mu)\begin{bmatrix}\mu(3\mathbf{P}_\alpha-\mathbf{P}_\beta)\\ \mathbf{C}\end{bmatrix}\boldsymbol{\eta}(t-1)\Bigg)\\
&+ \begin{bmatrix}\mu\mathbf{I}_{Jp}\\ \mathbf{0}\end{bmatrix}\bar{\boldsymbol{\eta}}(t) + \begin{bmatrix}\mu(3\mathbf{P}_\alpha-\mathbf{P}_\beta)\\ \mathbf{P}_\beta-\mathbf{P}_\alpha\end{bmatrix}\boldsymbol{\eta}(t). \qquad (3.51)
\end{aligned}$$

In obtaining the last equality in (3.51), we used: i) $\mathrm{bdiag}(\mathbf{I}_{Jp},\mathbf{L}_c)\boldsymbol{\Phi}(t,\mu) = \boldsymbol{\Upsilon}(t,\mu)\mathrm{bdiag}(\mathbf{I}_{Jp},\mathbf{L}_c)$; ii) the relationship between $\boldsymbol{\eta}_\alpha(t), \boldsymbol{\eta}_\beta(t)$ and $\boldsymbol{\eta}(t)$; and iii) the existence of a matrix $\mathbf{C}$ such that $\mathbf{L}_c\mathbf{C} = \mathbf{P}_\beta - \mathbf{P}_\alpha$. This made possible to extract the common factor $\mathrm{bdiag}(\mathbf{I}_{Jp},\mathbf{L}_c)$ and deduce from (3.51) that $\mathbf{y}(t+1)$ is given by (3.20), while $\mathbf{z}(t+1)$ is provided by (3.21).

In order to complete the proof, we must show the existence of matrix $\mathbf{C}$. To this end, via a simple evaluation one can check that[2] $\mathrm{nullspace}(\mathbf{L}_c) \subseteq \mathrm{nullspace}(\mathbf{P}_\beta^T - \mathbf{P}_\alpha^T)$, and since $\mathbf{L}_c$ is symmetric, we have $\mathrm{nullspace}(\mathbf{L}_c) \perp \mathrm{range}(\mathbf{L}_c)$. As $\mathrm{nullspace}(\mathbf{P}_\beta^T - \mathbf{P}_\alpha^T) \perp \mathrm{range}(\mathbf{P}_\beta - \mathbf{P}_\alpha)$,

[2] It is straightforward to completely characterize $\mathrm{null}(\mathbf{L}_c)$ from the properties of the Laplacian matrix $\mathbf{L}$ and the connectivity assumption for the WSN.

it follows that $\mathrm{range}(\mathbf{P}_\beta - \mathbf{P}_\alpha) \subseteq \mathrm{range}(\mathbf{L}_c)$, which further implies that we can find $\mathbf{C}$ such that $\mathbf{L}_c\mathbf{C} = \mathbf{P}_\beta - \mathbf{P}_\alpha$. □

3.7.3 Structure of Matrices $\mathbf{R}_{\bar{\boldsymbol{\eta}}_\mu}$ and $\mathbf{R}_{\boldsymbol{\eta}_\mu}$

From (3.27) we have

$$\mathbf{R}_{\bar{\boldsymbol{\eta}}_\mu} = \begin{bmatrix} \mu\mathbf{I}_{Jp} \\ \mathbf{0}_{Jp\times Jp} \end{bmatrix} \mathbf{R}_{\bar{\boldsymbol{\eta}}} \begin{bmatrix} \mu\mathbf{I}_{Jp} \\ \mathbf{0}_{Jp\times Jp} \end{bmatrix}^T, \quad \mathbf{R}_{\boldsymbol{\eta}_\mu} = \begin{bmatrix} \mu(3\mathbf{P}_\alpha - \mathbf{P}_\beta) \\ \mathbf{C} \end{bmatrix} \mathbf{R}_{\boldsymbol{\eta}} \begin{bmatrix} \mu(3\mathbf{P}_\alpha - \mathbf{P}_\beta) \\ \mathbf{C} \end{bmatrix}^T \tag{3.52}$$

so that it suffices to focus on the structure of $\mathbf{R}_{\bar{\boldsymbol{\eta}}}$ and $\mathbf{R}_{\boldsymbol{\eta}}$. From the definition in (3.18) and recalling that communication noise vectors are assumed uncorrelated in space, it follows that

$$\mathbf{R}_{\bar{\boldsymbol{\eta}}} = \mathrm{bdiag}\left(\sum_{j'\in\mathcal{N}_1\setminus\{1\}} \mathbf{R}_{\boldsymbol{\eta}_{1,j'}}, \ldots, \sum_{j'\in\mathcal{N}_J\setminus\{J\}} \mathbf{R}_{\boldsymbol{\eta}_{J,j'}} \right).$$

In the same way it follows from (3.19) that $\mathbf{R}_{\boldsymbol{\eta}}$ is a block diagonal matrix with a total of $\sum_{j=1}^J |\mathcal{N}_j|$ diagonal blocks of size $p \times p$, namely

$$\mathbf{R}_{\boldsymbol{\eta}} = \mathrm{bdiag}\left(\{\mathbf{R}_{\boldsymbol{\eta}_{j',1}}\}_{j'\in\mathcal{N}_1}, \ldots, \{\mathbf{R}_{\boldsymbol{\eta}_{j',J}}\}_{j'\in\mathcal{N}_J} \right).$$

Note also that the blocks $\mathbf{R}_{\boldsymbol{\eta}_{j,j}} = \mathbf{0}_{p\times p}$ for all $j \in \mathcal{J}$.

3.7.4 Proof of Lemma 3.3

We separately treat each of the three summands in $\mathcal{M}(\tilde{\boldsymbol{\Omega}}(\mu), \mathbf{R}_{\tilde{z}}(t))$ [cf. (3.29)] and finally add the results. The algebraic property $\mathrm{vec}[\mathbf{RST}] = (\mathbf{T}^T \otimes \mathbf{R})\mathrm{vec}[\mathbf{S}]$, allows writing the first term as $(\tilde{\boldsymbol{\Omega}}(\mu) \otimes \tilde{\boldsymbol{\Omega}}(\mu))\mathrm{vec}[\mathbf{R}_{\tilde{z}}(t)]$. The second term can be rewritten as $4\mu^2\mathrm{bdiag}(\boldsymbol{\Lambda}, \mathbf{0}_{2Jp\times 2Jp})[(\mathbf{I}_{3J} \otimes \mathbf{1}_{p\times p}) \circ \mathbf{R}_{\tilde{z}}(t)]\mathrm{bdiag}(\boldsymbol{\Lambda}, \mathbf{0}_{2Jp\times 2Jp})$, which is vectorized upon using the property for $\mathrm{vec}[\mathbf{RST}]$ followed by $\mathrm{vec}[\mathbf{R} \circ \mathbf{S}] = \mathrm{diag}(\mathrm{vec}[\mathbf{R}])\mathrm{vec}[\mathbf{S}]$, to yield

$$4\mu^2(\mathrm{bdiag}(\boldsymbol{\Lambda}, \mathbf{0}_{2Jp\times 2Jp}) \otimes \mathrm{bdiag}(\boldsymbol{\Lambda}, \mathbf{0}_{2Jp\times 2Jp}))\mathrm{diag}(\mathrm{vec}[\mathbf{I}_{3J} \otimes \mathbf{1}_{p\times p}])\mathrm{vec}[\mathbf{R}_{\tilde{z}}(t)].$$

The third term is a diagonal matrix, which can be decomposed as (up to a constant factor $4\mu^2$)

$$\begin{aligned}
&\mathrm{bdiag}(\mathrm{tr}(\boldsymbol{\Lambda}_1[\mathbf{R}_{\tilde{z}}(t)]_{11,1})\boldsymbol{\Lambda}_1,\ldots,\mathrm{tr}(\boldsymbol{\Lambda}_J[\mathbf{R}_{\tilde{z}}(t)]_{11,J})\boldsymbol{\Lambda}_J,\mathbf{0}_{2Jp\times 2Jp})\\
&\qquad = \sum_{j=1}^{J}\mathrm{tr}(\boldsymbol{\Lambda}_j[\mathbf{R}_{\tilde{z}}(t)]_{11,j})[\mathrm{diag}(\mathbf{b}_{3J,j})\otimes\boldsymbol{\Lambda}_j]\\
&\qquad = \sum_{j=1}^{J}\mathrm{tr}([\mathrm{diag}(\mathbf{b}_{3J,j})\otimes\boldsymbol{\Lambda}_j]\mathbf{R}_{\tilde{z}}(t))[\mathrm{diag}(\mathbf{b}_{3J,j})\otimes\boldsymbol{\Lambda}_j]\\
&\qquad = \sum_{j=1}^{J}\left(\mathrm{vec}[\mathrm{diag}(\mathbf{b}_{3J,j})\otimes\boldsymbol{\Lambda}_j]^T\mathrm{vec}[\mathbf{R}_{\tilde{z}}(t)]\right)[\mathrm{diag}(\mathbf{b}_{3J,j})\otimes\boldsymbol{\Lambda}_j],
\end{aligned} \tag{3.53}$$

where the last equality follows from the identity $\mathrm{tr}(\mathbf{R}^T\mathbf{S}) = \mathrm{vec}[\mathbf{R}]^T\mathrm{vec}[\mathbf{S}]$. Upon scaling and vectorizing (3.53), while letting $\mathbf{q}_j := \mathrm{vec}[\mathrm{diag}(\mathbf{b}_{3J,j})\otimes\boldsymbol{\Lambda}_j]$, one obtains $4\mu^2\sum_{j=1}^{J}\mathbf{q}_j\mathbf{q}_j^T\mathrm{vec}[\mathbf{R}_{\tilde{z}}(t)]$. The result follows readily after summing the three vectorized terms and taking $\mathrm{vec}[\mathbf{R}_{\tilde{z}}(t)]$ as common factor. □

Chapter 4

Distributed RLS for In-Network Adaptive Estimation

4.1 Introduction

Motivated by WSN-based estimation/tracking applications in which a data model is not available and fast convergence rates are at premium, fully distributed (D-) RLS type of algorithms are developed in this chapter. They perform consensus-based, in-network, adaptive least squares estimation in general ad hoc WSNs that are challenged by additive communication noise. The algorithms are derived by optimizing the convex exponentially-weighted least squares (EWLS) cost using distributed optimization techniques, namely the alternating-direction method of multipliers (AD-MoM) [7, p. 253] and the alternating minimization algorithm (AMA) [59]. The exponential weighting effected through a forgetting factor endows D-RLS with tracking capabilities. This is desirable in a constantly changing environment, within which WSNs are envisioned to operate. Whenever the use of powerful channel codes renders inter-sensor links virtually noise-free, both the AD-MoM and AMA-based D-RLS algorithms can be streamlined to lower communication overhead, yet higher convergence rates with respect to existing approaches in [8, 48].

For sensor observations that are linearly related to the time-invariant parameter of interest, a steady-state mean-square error (MSE) performance analysis is conducted for the

AMA-based D-RLS algorithm (Section 4.5). By relying on simplifying approximations and independence assumptions, exact closed-form expressions are derived for all the relevant MSE performance figures of merit. Stability is established in both the mean and MSE-sense. Numerical tests showcase the merits of the novel distributed estimation algorithms and corroborate the theoretical findings of this chapter.

4.2 Problem Statement

Consider a WSN with sensors $\{1,\ldots,J\} := \mathcal{J}$. Only single-hop communications are allowed, i.e., sensor j can communicate only with the sensors in its neighborhood $\mathcal{N}_j \subseteq \mathcal{J}$, having cardinality $|\mathcal{N}_j|$. Assuming that inter-sensor links are symmetric, the WSN is modeled as an undirected connected graph. Global connectivity information is captured by the symmetric adjacency matrix $\mathbf{E} \in \mathbb{R}^{J\times J}$, where $[\mathbf{E}]_{ij} = 1$ if $i \in \mathcal{N}_j$ and $[\mathbf{E}]_{ij} = 0$ otherwise. The graph Laplacian $\mathbf{L} \in \mathbb{R}^{J\times J}$ will be useful in the sequel, where $\mathbf{L} := \mathbf{D} - \mathbf{E}$, and $\mathbf{D} := \text{diag}(|\mathcal{N}_1|,\ldots,|\mathcal{N}_J|)$. Different from [8, 63] and [44], the present network model accounts explicitly for non-ideal sensor-to-sensor links. Specifically, signals received at sensor j from sensor i at discrete-time instant t are corrupted by a zero-mean additive noise vector $\boldsymbol{\eta}_j^i(t)$, assumed temporally and spatially uncorrelated.

The WSN is deployed to estimate a real signal vector $\mathbf{s}_0(t) \in \mathbb{R}^{p\times 1}$ in a distributed fashion and subject to the single-hop communication constraints, by resorting to the LS criterion [43, p. 658]. Per time instant $t = 0, 1, \ldots,$ each sensor acquires a regression vector $\mathbf{h}_j(t) \in \mathbb{R}^{p\times 1}$ and a scalar observation $x_j(t)$, both assumed zero-mean without loss of generality. A similar data setting was considered in [8] and [44]. Given new data sequentially acquired, a pertinent approach is to consider the EWLSE [8, 43, 44]

$$\hat{\mathbf{s}}_{\text{ewls}}(t) := \arg\min_{\mathbf{s}} \sum_{\tau=0}^{t}\sum_{j=1}^{J} \lambda^{t-\tau}\left[x_j(\tau) - \mathbf{h}_j^T(\tau)\mathbf{s}\right]^2 + \lambda^t \mathbf{s}^T \boldsymbol{\Phi}_0 \mathbf{s} \tag{4.1}$$

where $\lambda \in (0, 1]$ is a forgetting factor, while the positive definite matrix $\boldsymbol{\Phi}_0$ is included for regularization. Note that in forming the EWLSE at time t, the entire history of data $\{x_j(\tau), \mathbf{h}_j(\tau)\}_{\tau=0}^t$, $\forall\, j \in \mathcal{J}$ is incorporated in the online estimation process. Whenever $\lambda < 1$, past data are exponentially discarded thus enabling tracking of nonstationary processes.

Remark 4.1 (Fusion center versus ad hoc WSN operation) If one can afford constructing/maintaining a cyclic path across sensors; or, having sensors continuously communicate their new data to a central unit (fusion center), then the I-RLS algorithm in [44] can find the centralized EWLSE benchmark. However, in-network (or diffusion) estimators may consume less power relative to I-RLS while exhibiting improved resilience to sensor failures-a feature particularly critical as the WSN size increases.

Next, we describe an application setup for distributed adaptive linear LS estimation, which naturally gives rise to the aforementioned data setting and highlights the importance of the problem addressed.

4.2.1 Distributed Power Spectrum Estimation

Consider an ad hoc WSN deployed e.g., for collaborative habitat monitoring, whereby sensors observe a narrowband source to determine its spectral peaks. Such information enables the WSN to disclose hidden periodicities due to a physical phenomenon controlled by e.g., a seismic source. Let $\theta(t)$ denote the source of interest, which can be modeled as an autoregressive (AR) process [55, p. 106]

$$\theta(t) = -\sum_{\tau=1}^{p} \alpha_\tau \theta(t-\tau) + w(t) \tag{4.2}$$

where p is the order of the AR process, $\{\alpha_\tau\}$ the AR coefficients, and $w(t)$ denotes white noise. The source propagates to sensor j via a multi-path channel modeled as an FIR filter $C_j(z) = \sum_{l=0}^{L_j-1} c_{j,l} z^{-l}$, of *unknown* order L_j and tap coefficients $\{c_{j,l}\}$. In the presence of additive sensing noise $\bar{\epsilon}_j(t)$, the observation at sensor j is: $x_j(t) = \sum_{l=0}^{L_j-1} c_{j,l}\theta(t-l) + \bar{\epsilon}_j(t)$. Since $x_j(t)$ is an ARMA process, it can be written as [55]

$$x_j(t) = -\sum_{\tau=1}^{p} \alpha_\tau x_j(t-\tau) + \sum_{\tau'=1}^{m} \beta_{\tau'} \bar{\xi}_j(t-\tau'), \quad j \in \mathcal{J} \tag{4.3}$$

where the moving average (MA) coefficients $\{\beta_{\tau'}\}$ and the variance of the white noise process $\bar{\xi}_j(t)$ depend on $\{c_{j,l}\}$, $\{\alpha_\tau\}$, and the variance of the noise terms $w(t)$ and $\bar{\epsilon}_j(t)$. For the purpose of determining spectral peaks, the MA term in (4.3) can be treated as observation noise, i.e., $\epsilon_j(t) := \sum_{\tau'=1}^{m} \beta_{\tau'} \bar{\xi}_j(t-\tau')$. This is important since sensors do not have to know the source-sensor channel coefficients as well as the noise variances. The spectral content of the source can be obtained provided sensors estimate the coefficients $\{\alpha_\tau\}$, so let $\mathbf{s}_0 := [\alpha_1 \dots \alpha_p]^T$. Note from (4.3) that the regressor vectors here are $\mathbf{h}_j(t) = [-x_j(t-1) \dots - x_j(t-p)]^T$, directly from the sensor data $\{x_j(t)\}$ without the need of training/estimation.

Remark 4.2 (Spatial diversity via sensor collaboration) The source-sensor channels may introduce deep fades at the frequencies occupied by the source. Thus, having each sensor operating on its own may lead to faulty assessments. The necessary spatial diversity to effect improved spectral estimates, can only be achieved through sensor collaboration as in the D-RLS algorithm described next.

4.3 Distributed RLS Algorithm

In this section, we first construct the D-RLS algorithm, and then provide further insights regarding its implementation and associated communication overhead and computational complexity. The approach followed consists of two main steps: (i) reformulate (4.1) into an equivalent separable minimization problem that is amenable to distributed implementation; and (ii) rely on the AD-MoM [7, p. 253] to split (4.1) into simpler optimization subtasks that can be carried out locally at each sensor.

To decompose the cost function in (4.1), in which summands are coupled through the global variable $\mathbf{s}$, we introduce auxiliary variables $\{\mathbf{s}_j\}_{j=1}^J$ that represent local estimates of $\mathbf{s}_0$ per sensor j. These local estimates are utilized to form the convex *constrained* minimization problem:

$$\{\hat{\mathbf{s}}_j(t)\}_{j=1}^J := \arg \min_{\{\mathbf{s}_j\}_{j=1}^J} \sum_{\tau=0}^{t} \sum_{j=1}^{J} \lambda^{t-\tau}[x_j(\tau) - \mathbf{h}_j^T(\tau)\mathbf{s}_j]^2 + J^{-1}\lambda^t \sum_{j=1}^{J} \mathbf{s}_j^T \mathbf{\Phi}_0 \mathbf{s}_j,$$
$$\text{s. t.} \;\; \mathbf{s}_j = \mathbf{s}_{j'}, \;\; j \in \mathcal{J}, \;\; j' \in \mathcal{N}_j. \qquad (4.4)$$

From the connectivity of the WSN, (4.1) and (4.4) are equivalent in the sense that $\hat{\mathbf{s}}_j(t) = \hat{\mathbf{s}}_{\text{ewls}}(t)$, $\forall\, j \in \mathcal{J}$ and $t \geq 0$; see also [49].

4.3.1 Algorithm Construction

In order to tackle (4.4) in a distributed fashion, we resort to AD-MoM to obtain an adaptive algorithm that: i) allows recursive estimation of a time-invariant parameter $\mathbf{s}_0$; and ii) can track a time-varying process $\mathbf{s}_0(t)$. To facilitate application of AD-MoM, consider the auxiliary variables $\{\bar{\mathbf{z}}_j^{j'}, \tilde{\mathbf{z}}_j^{j'}\}_{j' \in \mathcal{N}_j}$ for $j \in \mathcal{J}$, and replace the constraints in (4.4) with the equivalent ones

$$\mathbf{s}_j = \bar{\mathbf{z}}_j^{j'}, \ \ \mathbf{s}_{j'} = \tilde{\mathbf{z}}_j^{j'}, \text{ and } \bar{\mathbf{z}}_j^{j'} = \tilde{\mathbf{z}}_j^{j'}, \ \ j \in \mathcal{J}, \ \ j' \in \mathcal{N}_j, \ \ j \neq j'. \tag{4.5}$$

Variables $\{\bar{\mathbf{z}}_j^{j'}, \tilde{\mathbf{z}}_j^{j'}\}$ are only used to derive the local recursions but will be eventually eliminated. Next, associate Lagrange multipliers $\mathbf{v}_j^{j'}$ and $\boldsymbol{\mu}_j^{j'}$ with the first two constraints in (4.5), and form the quadratically augmented Lagrangian

$$\begin{aligned}\mathcal{L}_a\left[\boldsymbol{s}, \boldsymbol{z}, \mathbf{v}, \boldsymbol{\mu}\right] = & \sum_{j=1}^{J}\sum_{\tau=0}^{t} \lambda^{t-\tau}[x_j(\tau) - \mathbf{h}_j^T(\tau)\mathbf{s}_j]^2 + J^{-1}\lambda^t \sum_{j=1}^{J} \mathbf{s}_j^T \boldsymbol{\Phi}_0 \mathbf{s}_j \\ & + \sum_{j=1}^{J}\sum_{j' \in \mathcal{N}_j} \left[\left(\mathbf{v}_j^{j'}\right)^T (\mathbf{s}_j - \bar{\mathbf{z}}_j^{j'}) + (\boldsymbol{\mu}_j^{j'})^T (\mathbf{s}_{j'} - \tilde{\mathbf{z}}_j^{j'}) \right] \\ & + \frac{c}{2}\sum_{j=1}^{J}\sum_{j' \in \mathcal{N}_j} \left[\|\mathbf{s}_j - \bar{\mathbf{z}}_j^{j'}\|^2 + \|\mathbf{s}_{j'} - \tilde{\mathbf{z}}_j^{j'}\|^2 \right]\end{aligned} \tag{4.6}$$

where c is positive penalty coefficient; and $\boldsymbol{s} := \{\mathbf{s}_j\}_{j=1}^J$, $\boldsymbol{z} := \{\bar{\mathbf{z}}_j^{j'}, \tilde{\mathbf{z}}_j^{j'}\}_{j \in \mathcal{J}}^{j' \in \mathcal{N}_j}$ and $[\mathbf{v}, \boldsymbol{\mu}] := \{\mathbf{v}_j^{j'}, \boldsymbol{\mu}_j^{j'}\}_{j \in \mathcal{J}}^{j' \in \mathcal{N}_j}$. Observe that the remaining constraints in (4.5), namely $\boldsymbol{z} \in C_z := \{\boldsymbol{z} : \bar{\mathbf{z}}_j^{j'} = \tilde{\mathbf{z}}_j^{j'}, \ j \in \mathcal{J}, \ j' \in \mathcal{N}_j, \ j \neq j'\}$, have not been dualized. Now, let $k = 0, 1, \ldots$ denote the iteration index for the recursive algorithm to be constructed in order to minimize (4.4) at time instant $t+1$. The first step in the AD-MoM updates the multipliers using the gradient ascent iterations

$$\mathbf{v}_j^{j'}(t+1;k) = \mathbf{v}_j^{j'}(t+1;k-1) + c[\mathbf{s}_j(t+1;k) - \bar{\mathbf{z}}_j^{j'}(t+1;k)] \tag{4.7}$$

$$\boldsymbol{\mu}_j^{j'}(t+1;k) = \boldsymbol{\mu}_j^{j'}(t+1;k-1) + c[\mathbf{s}_{j'}(t+1;k) - \tilde{\mathbf{z}}_j^{j'}(t+1;k)], \ \ j \in \mathcal{J}, \ \ j' \in \mathcal{N}_j. \tag{4.8}$$

The second step entails recursions that are obtained after minimizing (4.6) with respect to $\boldsymbol{s}$, assuming that all other variables $\boldsymbol{z}(t+1;k) := \{\bar{\mathbf{z}}_j^{j'}(t+1;k), \tilde{\mathbf{z}}_j^{j'}(t+1;k)\}_{j \in \mathcal{J}}^{j' \in \mathcal{N}_j}$ and

$[\mathbf{v}(t{+}1;k),\boldsymbol{\mu}(t{+}1;k)] := \{[\mathbf{v}_j^{j'}(t{+}1;k),\boldsymbol{\mu}_j^{j'}(t{+}1;k)]\}_{j\in\mathcal{J}}^{j'\in\mathcal{N}_j}$ are fixed. The separable structure of (4.6) with respect to $\mathbf{s}_j$ leads to the J separate minimization subproblems

$$\begin{aligned}\mathbf{s}_j(t+1;k+1) = \arg\min_{\mathbf{s}_j} \Bigg[&\sum_{\tau=0}^{t+1}\lambda^{t+1-\tau}[x_j(\tau)-\mathbf{h}_j^T(\tau)\mathbf{s}_j]^2 + J^{-1}\lambda^{t+1}\mathbf{s}_j^T\mathbf{\Phi}_0\mathbf{s}_j\\ &+\sum_{j'\in\mathcal{N}_j}\left[\mathbf{v}_j^{j'}(t+1;k)+\boldsymbol{\mu}_{j'}^{j}(t+1;k)\right]^T\mathbf{s}_j\\ &+\frac{c}{2}\sum_{j'\in\mathcal{N}_j}\left(\|\mathbf{s}_j-\bar{\mathbf{z}}_j^{j'}(t+1;k)\|^2+\|\mathbf{s}_j-\tilde{\mathbf{z}}_{j'}^{j}(t+1;k)\|^2\right)\Bigg]\end{aligned} \tag{4.9}$$

which are quadratic and whose optimal solution is available in closed form.

The third step involves updating $\{\bar{\mathbf{z}}_j^{j'}(t+1;k),\tilde{\mathbf{z}}_j^{j'}(t+1;k)\}$. The related recursions are obtained after minimizing $\mathcal{L}_a\left[\boldsymbol{s}(t+1;k+1),\boldsymbol{z},\mathbf{v}(t+1;k),\boldsymbol{\mu}(t+1;k)\right]$ with respect to $\boldsymbol{z}$ subject to $\boldsymbol{z}\in C_z$, while treating $\boldsymbol{s}(t+1;k+1) := \{\mathbf{s}_j(t+1;k+1)\}_{j\in\mathcal{J}}$ and $[\mathbf{v}(t+1;k),\boldsymbol{\mu}(t+1;k)] := \{[\mathbf{v}_j^{j'}(t+1;k),\ \boldsymbol{\mu}_j^{j'}(t+1;k)]\}_{j\in\mathcal{J}}^{j'\in\mathcal{N}_j}$ as fixed. Then, given the separable structure of the Lagrangian in (4.6) with respect to $\bar{\mathbf{z}}_j^{j'}$, it follows after letting $\bar{\mathbf{z}}_j^{j'}=\tilde{\mathbf{z}}_j^{j'}$ and retaining only the $\bar{\mathbf{z}}_j^{j'}$-dependent terms in (4.6) that

$$\begin{aligned}\bar{\mathbf{z}}_j^{j'}(t+1;k+1) := \arg\min_{\bar{\mathbf{z}}_j^{j'}}\Bigg[&-\left[\mathbf{v}_j^{j'}(t+1;k)+\boldsymbol{\mu}_j^{j'}(t+1;k)\right]^T\bar{\mathbf{z}}_j^{j'}\\ &+\frac{c}{2}\left(\|\mathbf{s}_j(t+1;k+1)-\bar{\mathbf{z}}_j^{j'}\|^2+\|\mathbf{s}_{j'}(t+1;k+1)-\bar{\mathbf{z}}_j^{j'}\|^2\right)\Bigg]\end{aligned}$$

which being linear-quadratic accepts the closed-form solution

$$\bar{\mathbf{z}}_j^{j'}(t+1;k+1) = \frac{1}{2c}\left[\mathbf{v}_j^{j'}(t+1;k)+\boldsymbol{\mu}_j^{j'}(t+1;k)\right]+\frac{1}{2}\left[\mathbf{s}_j(t+1;k+1)+\mathbf{s}_{j'}(t+1;k+1)\right]. \tag{4.10}$$

Clearly, $\bar{\mathbf{z}}_j^{j'}(t+1;k+1)=\tilde{\mathbf{z}}_j^{j'}(t+1;k+1)$. Substituting (4.10) into (4.7) and (4.8), it follows by induction that if the Lagrange multipliers are initialized such that $\mathbf{v}_j^{j'}(t+1;0)=-\boldsymbol{\mu}_j^{j'}(t+1;0)$, then $\mathbf{v}_j^{j'}(t+1;k)=-\boldsymbol{\mu}_j^{j'}(t+1;k)$ for all t and k, while

$$\mathbf{v}_j^{j'}(t+1;k)=\mathbf{v}_j^{j'}(t+1;k-1)+\frac{c}{2}\left[\mathbf{s}_j(t+1;k)-\mathbf{s}_{j'}(t+1;k)\right],\ \ j\in\mathcal{J},\ \ j'\in\mathcal{N}_j. \tag{4.11}$$

Notice that sensor j has to store and update only $\{\mathbf{v}_j^{j'}(t+1;k)\}_{j'\in\mathcal{N}_j}$ since $\boldsymbol{\mu}_j^{j'}$ turned out to be redundant. To obtain a recursion for $\mathbf{s}_j(t+1;k+1)$: i) substitute (4.10) into (4.9);

ii) use the identity $\boldsymbol{\mu}_{j'}^{j}(t+1;k) = -\mathbf{v}_{j'}^{j}(t+1;k)$ to eliminate $\boldsymbol{\mu}_{j'}^{j}$ from (4.9); and iii) apply first-order optimality conditions to the resulting quadratic cost. Then, $\mathbf{s}_j(t+1;k+1)$ can be obtained recursively as

$$\begin{aligned}\mathbf{s}_j(t+1;k+1) = \boldsymbol{\Phi}_j^{-1}(t+1)\boldsymbol{\psi}_j(t+1) + \frac{c}{2}\boldsymbol{\Phi}_j^{-1}(t+1)\sum_{j'\in\mathcal{N}_j}\left[\mathbf{s}_j(t+1;k)+\mathbf{s}_{j'}(t+1;k)\right] \\ - \frac{1}{2}\boldsymbol{\Phi}_j^{-1}(t+1)\sum_{j'\in\mathcal{N}_j}\left[\mathbf{v}_j^{j'}(t+1;k)-\mathbf{v}_{j'}^{j}(t+1;k)\right] \qquad (4.12)\end{aligned}$$

where

$$\boldsymbol{\Phi}_j(t+1) := \sum_{\tau=0}^{t+1}\lambda^{t+1-\tau}\mathbf{h}_j(\tau)\mathbf{h}_j^T(\tau) + J^{-1}\lambda^{t+1}\boldsymbol{\Phi}_0 + c|\mathcal{N}_j|\mathbf{I}_p \qquad (4.13)$$

$$\boldsymbol{\psi}_j(t+1) := \sum_{\tau=0}^{t+1}\lambda^{t+1-\tau}\mathbf{h}_j(\tau)x_j(\tau) = \lambda\boldsymbol{\psi}_j(t) + \mathbf{h}_j(t+1)x_j(t+1). \qquad (4.14)$$

Recursions (4.11) and (4.12) constitute the D-RLS algorithm, whereby all sensors $j \in \mathcal{J}$ keep track of their local estimate $\mathbf{s}_j(t+1;k+1)$ and their multipliers $\{\mathbf{v}_j^{j'}(t+1;k)\}_{j'\in\mathcal{N}_j}$, which can be arbitrarily initialized. For $\lambda = 1$, matrix $\boldsymbol{\Phi}_j^{-1}(t+1)$ can be also recursively obtained from $\boldsymbol{\Phi}_j^{-1}(t)$ with complexity $\mathcal{O}(p^2)$ using the matrix inversion lemma; i.e.,

$$\boldsymbol{\Phi}_j^{-1}(t+1) = \boldsymbol{\Phi}_j^{-1}(t) - \frac{\boldsymbol{\Phi}_j^{-1}(t)\mathbf{h}_j(t+1)\mathbf{h}_j^T(t+1)\boldsymbol{\Phi}_j^{-1}(t)}{1+\mathbf{h}_j^T(t+1)\boldsymbol{\Phi}_j^{-1}(t)\mathbf{h}_j(t+1)}. \qquad (4.15)$$

Interestingly, the first term in $\mathbf{s}_j(t+1;k+1)$, namely $\boldsymbol{\Phi}_j(t+1)^{-1}\boldsymbol{\psi}_j(t+1)$, is a regularized version of the local EWLSE per sensor j at time instant $t+1$. The regularization is imposed by the scaled identity matrix term in $\boldsymbol{\Phi}_j(t+1)$. Contrary to the classical RLS, see e.g., [43], it allows one to set $\boldsymbol{\Phi}_0 = \mathbf{0}$ without compromising the invertibility of $\boldsymbol{\Phi}_j(t)$. The remaining terms in (4.12) are responsible for fusing information from the neighborhood of sensor j, refining in that way the local estimate provided by $\boldsymbol{\Phi}_j^{-1}(t+1)\boldsymbol{\psi}_j(t+1)$. As promised, the variables $\{\bar{\mathbf{z}}_j^{j'}, \tilde{\mathbf{z}}_j^{j'}\}$ have been completely eliminated from the D-RLS recursions in (4.11)-(4.12).

In order to solve (4.4) at time instant $t+1$, all sensors run local consensus recursions. During the $(k+1)$-st consensus iteration, sensor j receives the local estimates $\mathbf{s}_{j'}(t+1;k)$ from its neighbors $j' \in \mathcal{N}_j$ and updates its multipliers $\mathbf{v}_j^{j'}(t+1;k)$ via (4.11). Then, sensor j receives the multipliers $\{\mathbf{v}_j^{j'}(t+1;k)\}_{j'\in\mathcal{N}_j}$ and uses them along with $\{\mathbf{s}_{j'}(t+1;k)\}_{j'\in\mathcal{N}_j}$

to evaluate $\mathbf{s}_j(t+1;k+1)$ via (4.12). This way, recursions (4.11)-(4.12) minimize (4.4) asymptotically. Specifically, it follows that:

Proposition 4.1 *For arbitrarily initialized* $\{\mathbf{v}_j^{j'}(t;0)\}_{j\in\mathcal{J}}^{j'\in\mathcal{N}_j}$, $\mathbf{s}_j(t;0)$ *and any* $c>0$*; the local estimates* $\mathbf{s}_j(t;k)$ *generated by (4.12) reach consensus as* $k\to\infty$*; i.e.,*

$$\lim_{k\to\infty}\mathbf{s}_j(t;k)=\hat{\mathbf{s}}_{ewls}(t),\ \textit{for all}\ j\in\mathcal{J}.$$

Proof: See Appendix 4.6.1.

Thus, D-RLS recursions are able to attain the EWLSE at each time instant t as long as the number of consensus iterations grows. For a time-invariant setup, running many consensus iterations, i.e., $k\gg 1$ would not be a problem, though this is not the case when the sensors track a time-varying process $\mathbf{s}_0(t)$. One way to enable D-RLS operation in nonstationary settings, is to apply one consensus iteration per time instant t. In this case, $k=t$ and recursions (4.11)-(4.12) simplify to

$$\mathbf{v}_j^{j'}(t)=\mathbf{v}_j^{j'}(t-1)+\frac{c}{2}[\mathbf{s}_j(t)-(\mathbf{s}_{j'}(t)+\boldsymbol{\eta}_j^{j'}(t))], \tag{4.16}$$

$$\begin{aligned}\mathbf{s}_j(t+1)=&\ \boldsymbol{\Phi}_j^{-1}(t+1)\boldsymbol{\psi}_j(t+1)+\frac{c}{2}\boldsymbol{\Phi}_j^{-1}(t+1)\sum_{j'\in\mathcal{N}_j}\left[\mathbf{s}_j(t)+(\mathbf{s}_{j'}(t)+\boldsymbol{\eta}_j^{j'}(t))\right]\\ &-\frac{1}{2}\boldsymbol{\Phi}_j^{-1}(t+1)\sum_{j'\in\mathcal{N}_j}\left[\mathbf{v}_j^{j'}(t)-(\mathbf{v}_{j'}^{j}(t)+\bar{\boldsymbol{\eta}}_j^{j'}(t))\right]\end{aligned} \tag{4.17}$$

where $\boldsymbol{\eta}_j^{j'}(t)$ and $\bar{\boldsymbol{\eta}}_j^{j'}(t)$ denote the additive communication noise present in the reception of $\mathbf{s}_j(t)$ and $\mathbf{v}_{j'}^{j}(t)$ at sensor j, respectively. In detail, during time instant $t+1$ sensor j receives the local estimates $\{\mathbf{s}_{j'}(t)+\boldsymbol{\eta}_j^{j'}(t)\}_{j'\in\mathcal{N}_j}$ and plugs them into (4.16) to evaluate $\mathbf{v}_j^{j'}(t)$ for $j'\in\mathcal{N}_j$. Then, it receives $\mathbf{v}_{j'}^{j}(t)+\bar{\boldsymbol{\eta}}_j^{j'}(t)$ from its neighbors $j'\in\mathcal{N}_j$, which are used together with $\{\mathbf{s}_{j'}(t)+\boldsymbol{\eta}_j^{j'}(t)\}_{j'\in\mathcal{N}_j}$ and the new observation data within $\boldsymbol{\Phi}_j^{-1}(t+1)\boldsymbol{\psi}_j(t+1)$ to obtain $\mathbf{s}_j(t+1)$ via (4.17). Recursions (4.16)-(4.17) constitute a single-time (ST-) scale version of D-RLS, abbreviated as STD-RLS and tabulated as Algorithm 4. Note also that there is no need for a common penalty coefficient c across sensors; that is, each sensor can use its own local penalty coefficient $c_j>0$ allowing increased flexibility to attain potentially higher convergence rates.

Algorithm 4 : STD-RLS

```
  Arbitrarily initialize $\{\mathbf{s}_j(0)\}_{j=1}^{J}$ and $\{\mathbf{v}_j^{j'}(-1)\}_{j\in\mathcal{J}}^{j'\in\mathcal{N}_j}$.
  for $t = 0, 1, \ldots$ do
    All $j \in \mathcal{J}$: transmit $\mathbf{s}_j(t)$ to neighbors in $\mathcal{N}_j$.
    All $j \in \mathcal{J}$: update $\{\mathbf{v}_j^{j'}(t)\}_{j'\in\mathcal{N}_j}$ using (4.16).
    All $j \in \mathcal{J}$: transmit $\mathbf{v}_j^{j'}(t)$ to each $j' \in \mathcal{N}_j$.
    All $j \in \mathcal{J}$: update $\mathbf{s}_j(t+1)$ using (4.17).
  end for
```

Remark 4.3 (Comparison with a bridge sensor-based D-RLS algorithm) A similar consensus-based RLS algorithm was put forth in [46]. To enable task parallelization via AD-MoM while ensuring that estimates agree across the whole WSN, the approach in [46] judiciously reformulates (4.1) by relying on a bridge sensor subset. Such approach is related to the one followed in Chapter 2 when developing the D-LMS algorithm, and builds on the framework originally proposed in [45, 49]. Not only setting-up – but readjusting the bridge sensor set, e.g., when sensors inevitably fail in battery-limited WSN deployments – requires additional coordination among sensors with an associated communication overhead. Compared to [46], the approach followed here does not require such a bridge sensor set, and in this sense, it offers a fully distributed, robust, and resource efficient RLS-type algorithm for use in ad hoc WSNs.

4.3.2 Communication and Computational Costs

Next, we analyze the communication and computational costs associated with D-RLS, and compare them with those incurred by existing approaches. Per D-RLS iteration each sensor transmits $p(|\mathcal{N}_j| + 1)$ scalars corresponding to the multipliers $\{\mathbf{v}_j^{j'}\}_{j'\in\mathcal{N}_j}$, and the local estimate $\mathbf{s}_j$. In diffusion RLS [8], each sensor transmits $2p+1$ scalars per iteration. However, when considering the reception cost it can be seen that while in D-RLS each sensor receives $2|\mathcal{N}_j|p$ scalars per recursion, in diffusion RLS the number of received scalars increases to $|\mathcal{N}_j|(2p + 1)$ per iteration. Even though the transmission cost is arguably greater than the one related to reception, it will be corroborated via numerical examples that the higher transmission cost in D-RLS pays off in improved convergence rates and robustness in the presence of communication noise.

The communication cost for I-RLS in [44] is $\mathcal{O}(p^2)$, since each sensor has to transmit to its successor in the Hamiltonian cycle a $p \times p$ covariance matrix; similar complexity is incurred by the scheme in [63]. A low communication cost I-RLS is also proposed in [44] in which each sensor within the cycle transmits and receives p scalars per iteration, though the challenges related to I-RLS remain as the WSN scales.

In comparison to the D-LMS algorithm in Chapter 3, D-RLS incurs the same communication cost. This is the case since the communication steps in both schemes are identical (cf. Algorithm 2 in Chapter 3 and Algorithm 4).

Next, we focus on the computational complexity involved in implementing (4.16)-(4.17). Updating the multipliers incurs complexity in the order of $\mathcal{O}(|\mathcal{N}_j|p)$. In determining $\mathbf{s}_j(t+1)$, the dominating cost arises from calculating $\mathbf{\Phi}_j^{-1}(t+1)$. Recall that when $\lambda = 1$, matrix $\mathbf{\Phi}_j^{-1}(t+1)$ can be computed recursively with a complexity of $\mathcal{O}(p^2)$. If $\lambda < 1$, then the complexity for determining $\mathbf{\Phi}_j^{-1}(t+1)$ is $\mathcal{O}(p^3)$. In diffusion RLS, the computational complexity is also dominated by the cost of recursively updating the inverse of the regression covariance matrix, and is of order $\mathcal{O}(|\mathcal{N}_j|p^2)$. Thus, for $\lambda = 1$ the computational complexity per iteration is smaller than the one in diffusion RLS. For $\lambda < 1$, the way D-RLS and diffusion RLS compare in terms of computational complexity depends on the relative size of $\{|\mathcal{N}_j|\}_{j=1}^J$ and p. Specifically, if $p < |\mathcal{N}_j|$ (e.g., in localization applications where $p \leq 3$), then D-RLS incurs smaller complexity. While, if $p > |\mathcal{N}_j|$ diffusion RLS is less complex computationally.

Finally, we point out that the D-LMS algorithm in Chapter 3 incurs lower complexity when compared to D-RLS. This is also the case for their centralized counterparts; see, e.g., [43, 52]. While the Lagrange multiplier recursions are identical in both schemes, the local estimate updates for D-LMS in (3.14) incur a complexity in the order of $\mathcal{O}(p)$. As already mentioned, the corresponding updates for D-RLS can attain $\mathcal{O}(p^2)$ in the (most favorable) infinite memory case ($\lambda = 1$).

Remark 4.4 (Communication noise resilience) Despite the fact that the computa-

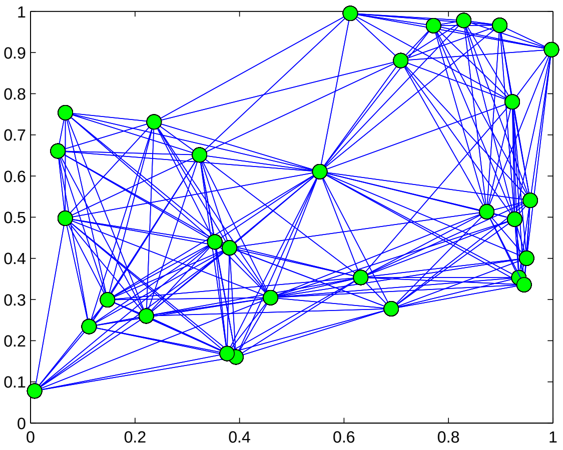

Figure 4.1: An ad hoc WSN with $J = 30$ sensors.

tional complexity of (ST)D-RLS depends on the estimation setting, the novel algorithm enjoys communication noise resilience, not present in existing alternatives. This property makes (ST)D-RLS a viable candidate for estimation/tracking in WSNs. It is also an expected feature since both algorithms rely on the AD-MoM, which exhibits robustness in the presence of communication noise [49].

4.3.3 Numerical Tests

Here we test the novel D-RLS and STD-RLS algorithms in the spectral application setting described in Section 4.2.1, conducting several performance comparisons with: i) I-RLS [44]; ii) diffusion RLS using Metropolis weights [8]; iii) local (L-) RLS, whereby each sensor runs an independent RLS algorithm solely based on its own data (no inter-sensor communications); and iv) D-LMS with step-size $\mu = 10^{-2}$ (cf. Algorithm 2). For $J = 30$ sensors, an ad-hoc WSN is generated by using the random geometric graph model in $[0, 1]^2$, with communication range $r = 0.6$; see Fig. 4.1. For the examples with noisy links, additive white Gaussian noise (AWGN) with variance $\sigma_\eta^2 = 10^{-1}$ is added at the receiving end. The source $\theta(t)$ is an AR(4) process with coefficients $\mathbf{s}_0 = [-0.31, 1.14, 0.28, 0.22]^T$ and driving noise variance $\sigma_w^2 = 10^{-2}$, which yields a spectrum with a single peak at $\omega = \pi/2$. The

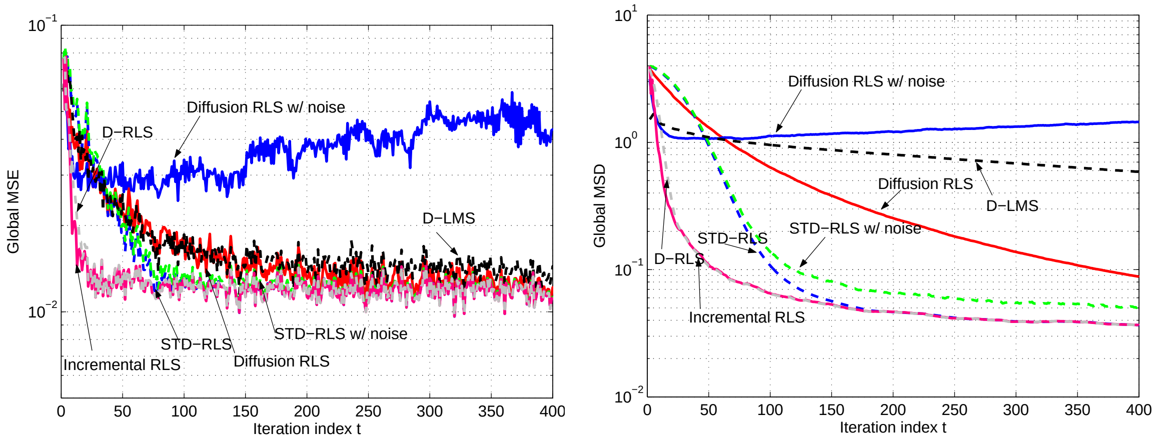

Figure 4.2: Global network performance in a distributed power spectrum estimation task: (left) MSE (learning curve); (right) MSD.

source-sensor channels have order $L_j = 2$, and the channels to the sensors 3, 7, 15 and 27 vanish at the frequency $\omega = \pi/2$. The observation AWGN has a spatial variance profile $\sigma^2_{\bar{\epsilon}_j} = \alpha_j \times 10^{-4}$, where the coefficients $\alpha_j \sim \mathcal{U}[0, 1]$ (uniformly distributed) are i.i.d. across sensors.

Thirty consensus steps are ran per acquired observation in D-RLS, in order to ensure a fair comparison with I-RLS in terms of processing delay. The delay is due to the estimation cycle over all $J = 30$ sensors, that should be completed before new information can be incorporated. With $\lambda = 1$ and $c_j = 7/|\mathcal{N}_j|$ in both D-RLS and STD-RLS, Fig. 4.2 (left) compares the global MSE evolution (learning curve) obtained as $J^{-1}\sum_{j=1}^{J} E[(x_j(t) - \mathbf{h}_j^T(t)\mathbf{s}_j(t-1))^2]$, whereas the expectation is approximated by averaging over 250 realizations of the experiment. Similar curves are shown in Fig. 4.2 (right), in this case for the global mean-square deviation (MSD) metric given by $J^{-1}\sum_{j=1}^{J} E[\|\mathbf{s}_j(t) - \mathbf{s}_0\|^2]$. I-RLS and D-RLS behave similarly providing a performance benchmark, while D-LMS – a first-order method – converges much slower than all distributed RLS schemes. STD-RLS outperforms diffusion RLS in terms of convergence rate, and most importantly, it does not suffer from the catastrophic

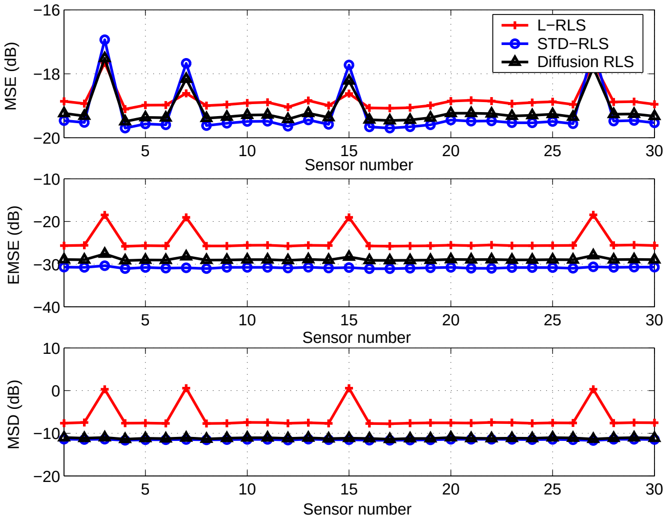

Figure 4.3: Local (per-sensor) performance in a distributed power spectrum estimation task.

noise accumulation exhibited by diffusion RLS when the links are not ideal.

With regards to local performance in steady-state, for $\lambda = 0.9$ we illustrate in Fig. 4.3 the figures of merit which are customary in the adaptive literature [8, 43, 48]: i) MSE $E[(x_j(t) - \mathbf{h}_j^T(t)\mathbf{s}_j(t-1))^2]$; ii) excess-MSE (EMSE) $E[(\mathbf{h}_j^T(t)(\mathbf{s}_j(t-1) - \mathbf{s}_0))^2]$; and MSD $E[\|\mathbf{s}_j(t) - \mathbf{s}_0\|^2]$. With reference to Remark 4.2, it is apparent that a scheme devoid of sensor collaboration such as L-LMS, fails to obtain satisfactory estimates at the sensors affected by the channel fades. On the other hand, STD-RLS exploits the available spatial diversity to attain improved estimation performance, see, e.g., the local MSD curves.

Next, we illustrate the capabilities of STD-RLS when it comes to tracking a time varying parameter $\mathbf{s}_0(t)$. For $p = 6$ and for the same WSN setup, we simulate a large amplitude slowly time-varying process $\mathbf{s}_0(t) = (1 - \rho_1)\mathbf{s}_0(t-1) + \boldsymbol{\nu}(t)$ with $\rho_1 = 0.9$ and $\boldsymbol{\nu}(t) \sim \mathcal{N}(\mathbf{0}, 10^{-2}\mathbf{I}_6)$ (multivariate normal distribution). A linear model is adopted for the sensor observations, i.e., $x_j(t) = \mathbf{h}_j^T(t)\mathbf{s}_0(t) + \zeta_j(t)$ with $\zeta_j(t) \sim \mathcal{N}(0, 10^{-4})$ for all $j \in \mathcal{J}$. Regressors are temporally correlated, as $\mathbf{h}_j(t) = [h_j(t) \ldots h_j(t-5)]^T$ with entries which evolve according to $h_j(t) = (1 - \rho_2)\beta_j h_j(t-1) + \sqrt{\rho_2}\nu_j(t)$. We choose $\rho_2 = 0.7$, the $\beta_j \sim \mathcal{U}[0, 1]$ are i.i.d. in space, and the driving white noise $\nu_j(t) \sim \mathcal{U}[-\sqrt{3}\sigma_{\nu_j}, \sqrt{3}\sigma_{\nu_j}]$ has a spatial variance profile given by $\sigma_{\nu_j}^2 = 10^{-1}\gamma_j$, with $\gamma_j \sim \mathcal{U}[0, 1]$ and i.i.d. For $\lambda = 0.5$ and $c_j = 7/|\mathcal{N}_j|$, Fig. 4.4

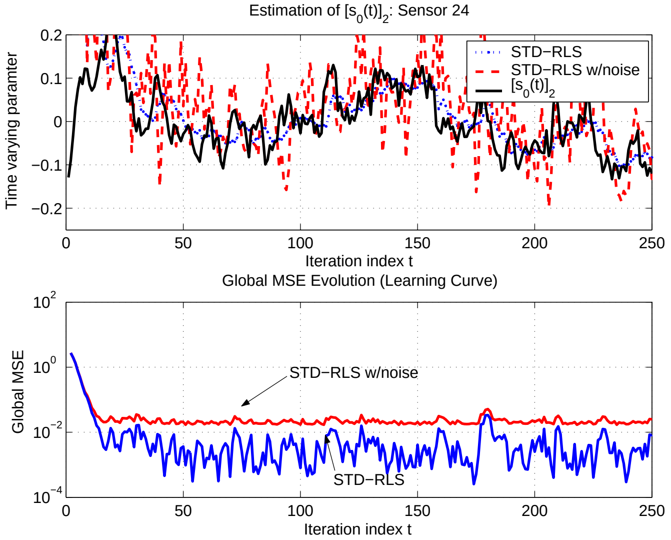

Figure 4.4: Tracking with STD-RLS.

depicts the second entry of $\mathbf{s}_0(t)$ as well as the corresponding estimate for a representative sensor closely tracking the true variations. In the presence of communication noise, the larger estimate fluctuations are a direct manifestation of the (expected) increased MSE, as evidenced by the learning curves in Fig. 4.4.

The scheme in [63] has not been included in the numerical comparisons because a complete data model is not available for the power spectrum estimation problem. Specifically, the variance of the aggregate observation noise term $\epsilon_j(t)$ is unknown (cf. Section 4.2.1). Further, the algorithm in [63] is incapable of tracking $\mathbf{s}_0(t)$ due to its diminishing step-size.

4.4 Reduced Complexity Distributed RLS Algorithms

This section deals with a pair of reduced complexity variants to the (ST)D-RLS algorithm. In the first case, communication and computational savings can be effected under the assumption of ideal inter-sensor links, i.e., when $\boldsymbol{\eta}_j^{j'}(t) = \bar{\boldsymbol{\eta}}_j^{j'}(t) = \mathbf{0}$ in (4.16)-(4.17). A second algorithm is developed by minimizing (4.4) using the alternating minimization algorithm (AMA) in [59], instead of the AD-MoM described in Section 4.3.1. The improved AMA-based D-RLS algorithm yields local estimate updates that incur computational complexity in the order of $\mathcal{O}(p^2)$, *for all* values of the forgetting factor $\lambda \in (0,1]$.

4.4.1 D-RLS Algorithm with Ideal Links

When communication noise is not present as in the scenarios considered in [8,44,63], D-RLS can be modified such that its corresponding communication complexity becomes lower than the one incurred by diffusion RLS. Specifically, note that if the multipliers $\mathbf{v}_j^{j'}$ are initialized such that $\mathbf{v}_j^{j'}(t;0) = -\mathbf{v}_{j'}^{j}(t;0)$, then in the absence of noise $\mathbf{v}_j^{j'}(t;k) = -\mathbf{v}_{j'}^{j}(t;k)$ for all k and t [cf. (4.7)]. Similarly, for the STD-RLS it follows that if $\mathbf{v}_j^{j'}(0) = -\mathbf{v}_{j'}^{j}(0)$ and noise is not present, then $\mathbf{v}_j^{j'}(t) = -\mathbf{v}_j^{j'}(t)$ for all t. Taking into account this equality, and setting $\bar{\boldsymbol{\eta}}_j^{j'}(t) = \boldsymbol{\eta}_j^{j'}(t) = \mathbf{0}$, the recursion for $\mathbf{s}_j(t+1)$ in STD-RLS is rewritten as [(4.12) can be reformulated in the same way]

$$\mathbf{s}_j(t+1) = \mathbf{\Phi}_j^{-1}(t+1)\boldsymbol{\psi}_j(t+1) + \frac{c}{2}\mathbf{\Phi}_j^{-1}(t+1)\sum_{j'\in\mathcal{N}_j}\left[\mathbf{s}_j(t)+\mathbf{s}_{j'}(t)\right] - \mathbf{\Phi}_j^{-1}(t+1)\sum_{j'\in\mathcal{N}_j}\mathbf{v}_j^{j'}(t). \tag{4.18}$$

The last summand in (4.18) incorporates only local multipliers stored at sensor j. Thus, each sensor does not exchange multipliers with its neighbors to update $\mathbf{s}_j(t+1)$. When using the modified STD-RLS comprising recursions (4.16) and (4.18), each sensor transmits p scalars and receives $|\mathcal{N}_j|p$ scalars per iteration. The communication overhead is smaller than the one associated with diffusion RLS (cf. the discussion in Section 4.3.2). However, as in diffusion RLS the low transmission cost is counterbalanced by the lack of resilience in the presence of communication noise; see also Remark 3.3 in Chapter 3.

Because $\mathbf{s}_j(t+1)$ in (4.18) is a function of $\mathbf{p}_j(t) := \sum_{j'\in\mathcal{N}_j}\mathbf{v}_j^{j'}(t)$, there is no need to separately update all multipliers $\{\mathbf{v}_j^{j'}\}_{j'\in\mathcal{N}_j}$ as in (4.16). It suffices to keep track of a single local vector $\mathbf{p}_j(t)$ instead, whose updating rule follows immediately from (4.16). The final recursions per sensor $j \in \mathcal{J}$ are:

$$\mathbf{p}_j(t) = \mathbf{p}_j(t-1) + \frac{c}{2}\sum_{j' \in \mathcal{N}_j} (\mathbf{s}_j(t) - \mathbf{s}_{j'}(t)) \tag{4.19}$$

$$\mathbf{s}_j(t+1) = \mathbf{\Phi}_j^{-1}(t+1)\boldsymbol{\psi}_j(t+1) + \frac{c}{2}\mathbf{\Phi}_j^{-1}(t+1)\sum_{j' \in \mathcal{N}_j} \left[\mathbf{s}_j(t) + \mathbf{s}_{j'}(t)\right] - \mathbf{\Phi}_j^{-1}(t+1)\mathbf{p}_j(t). \tag{4.20}$$

Interestingly, (4.19)-(4.20) require each sensor to store and update only $2p$ scalars, regardless of the WSN topology and corresponding neighborhood sizes. Recalling that $\mathbf{\Phi}_j^{-1}(t)$ and $\boldsymbol{\psi}_j(t)$ need to be updated separately, the total memory requirement per sensor adds up to $p^2 + 3p$ scalars. Recursions (4.19)-(4.20) implement Algorithm 4 exactly when links are ideal; hence, the aforementioned savings are achieved without any performance penalty.

4.4.2 The Alternating Minimization Algorithm

With reference to (4.13), suppose for a moment that $c = 0$ so that

$$\mathbf{\Phi}_j(t+1) = \lambda\mathbf{\Phi}_j(t) + \mathbf{h}_j(t+1)\mathbf{h}_j^T(t+1). \tag{4.21}$$

It is then possible to efficiently update $\mathbf{\Phi}_j^{-1}(t+1)$ via (4.15) using the matrix inversion lemma. As argued in Section 4.3.1, (4.21) does not hold for the general case $c > 0$, unless $\lambda = 1$. This is due to the – from this perspective undesirable – regularization term $c|\mathcal{N}_j|\mathbf{I}_p$ in (4.13), a direct consequence of the quadratic penalty in the augmented Lagrangian. Unfortunately, the penalty coefficient cannot be set to zero because the D-RLS algorithm breaks down. For instance, when the initial Lagrange multipliers are null and $c = 0$, STD-RLS boils down to L-LRS and consensus cannot be enforced. Motivated to further reduce the complexity in updating $\mathbf{s}_j$, an improved version of the D-RLS algorithm is developed in this section such that (4.21) holds true for all values of $\lambda \in (0, 1]$.

Going back to the starting point of this chapter, consider minimizing (4.4) with the consensus constraints equivalently represented as (4.5). Introduce the ordinary Lagrangian function

$$\mathcal{L}\left[\boldsymbol{s},\boldsymbol{z},\mathbf{v},\boldsymbol{\mu}\right] = \sum_{j=1}^{J}\sum_{\tau=0}^{t}\lambda^{t-\tau}[x_j(\tau)-\mathbf{h}_j^T(\tau)\mathbf{s}_j]^2 + J^{-1}\lambda^t\sum_{j=1}^{J}\mathbf{s}_j^T\mathbf{\Phi}_0\mathbf{s}_j$$
$$+\sum_{j=1}^{J}\sum_{j'\in\mathcal{N}_j}\left[\left(\mathbf{v}_j^{j'}\right)^T(\mathbf{s}_j-\bar{\mathbf{z}}_j^{j'})+(\boldsymbol{\mu}_j^{j'})^T(\mathbf{s}_{j'}-\tilde{\mathbf{z}}_j^{j'})\right] \qquad (4.22)$$

which does not include quadratic terms penalizing the constraint violations. Much related to the AD-MoM is the alternating minimization algorithm (AMA), which was proposed in [59] to tackle a class of separable convex optimization problems. The AMA solver also entails an iterative procedure comprising three steps per iteration $k = 0, 1, 2, \ldots$

[S1] Multiplier updates:

$$\mathbf{v}_j^{j'}(t+1;k) = \mathbf{v}_j^{j'}(t+1;k-1) + c[\mathbf{s}_j(t+1;k) - \bar{\mathbf{z}}_j^{j'}(t+1;k)], \quad j \in \mathcal{J},\ j' \in \mathcal{N}_j$$
$$\boldsymbol{\mu}_j^{j'}(t+1;k) = \boldsymbol{\mu}_j^{j'}(t+1;k-1) + c[\mathbf{s}_{j'}(t+1;k) - \tilde{\mathbf{z}}_j^{j'}(t+1;k)], \quad j \in \mathcal{J},\ j' \in \mathcal{N}_j.$$

[S2] Local estimate updates:

$$\boldsymbol{s}(t+1,k+1) = \arg\min_{\boldsymbol{s}}\mathcal{L}\left[\boldsymbol{s},\boldsymbol{z}(t+1,k),\mathbf{v}(t+1,k),\boldsymbol{\mu}(t+1,k)\right]. \qquad (4.23)$$

[S3] Auxiliary variable updates:

$$\boldsymbol{z}(t+1,k+1) = \arg\min_{\boldsymbol{z}\in C_z}\mathcal{L}_a\left[\boldsymbol{s}(t+1,k+1),\boldsymbol{z},\mathbf{v}(t+1,k),\boldsymbol{\mu}(t+1,k)\right]. \qquad (4.24)$$

Steps [S1] and [S3] coincide with the respective ones in AD-MoM. The only difference is with regards to the local estimate updates in [S2], where in AMA the new iterates are obtained by minimizing the ordinary Lagrangian with respect to $\boldsymbol{s}$. For the sake of the aforementioned minimization, all other variables are considered fixed taking their most up to date values $\{\boldsymbol{z}(t+1,k),\mathbf{v}(t+1,k),\boldsymbol{\mu}(t+1,k)\}$. For the AD-MoM instead, the minimized quantity was the augmented Lagrangian. The AMA was motivated in [59] to tackle separable problems that are strictly convex in $\boldsymbol{s}$, but not necessarily with respect to $\boldsymbol{z}$. Under this assumption, [S2] still yields a unique minimizer and the AMA is useful for those cases in which the Lagrangian is much simpler to optimize than the augmented Lagrangian.

Because of the regularization matrix $\mathbf{\Phi}_0 \succ \mathbf{0}$, the EWLS cost in (4.4) is indeed strictly convex for all $t > 0$. From the results in Section 4.3.1, still

$$\bar{\mathbf{z}}_j^{j'}(t+1,k+1) = \tilde{\mathbf{z}}_j^{j'}(t+1,k+1) = \frac{1}{2}\left[\mathbf{s}_j(t+1,k+1) + \mathbf{s}_{j'}(t+1,k+1)\right], \quad j \in \mathcal{J},\, j' \in \mathcal{N}_j$$

while $\mathbf{v}_j^{j'}(t+1;k) = -\boldsymbol{\mu}_j^{j'}(t+1;k)$ for all $k > -1$, and $\mathbf{v}_j^{j'}(t+1;k)$ is given by (4.11). Moving on to [S2], from the separable structure of (4.22) the minimization (4.23) can be split into J subproblems

$$\mathbf{s}_j(t+1,k+1) = \arg\min_{\mathbf{s}_j}\left[\sum_{\tau=0}^{t+1}\lambda^{t+1-\tau}[x_j(\tau) - \mathbf{h}_j^T(\tau)\mathbf{s}_j]^2 + J^{-1}\lambda^{t+1}\mathbf{s}_j^T\mathbf{\Phi}_0\mathbf{s}_j \right.$$
$$\left. + \sum_{j'\in\mathcal{N}_j}\left[\mathbf{v}_j^{j'}(t+1,k) - \mathbf{v}_{j'}^{j}(t+1,k)\right]^T\mathbf{s}_j\right].$$

Since the local subproblems correspond to unconstrained quadratic minimization, they admit closed-form solutions

$$\mathbf{s}_j(t+1,k+1) = \mathbf{\Phi}_j^{-1}(t+1)\boldsymbol{\psi}_j(t+1) - \frac{1}{2}\mathbf{\Phi}_j^{-1}(t+1)\sum_{j'\in\mathcal{N}_j}\left[\mathbf{v}_j^{j'}(t+1,k) - \mathbf{v}_{j'}^{j}(t+1,k)\right] \quad (4.25)$$

where $\boldsymbol{\psi}_j(t+1)$ is given by (4.14), and $\mathbf{\Phi}_j(t+1)$ satisfies the desired first-order recursion (4.21). Recursions (4.11) and (4.25) constitute the improved AMA-based D-RLS algorithm, whereby all sensors $j \in \mathcal{J}$ keep track of their local estimate $\mathbf{s}_j(t+1;k+1)$ and their multipliers $\{\mathbf{v}_j^{j'}(t+1;k)\}_{j'\in\mathcal{N}_j}$, which can be arbitrarily initialized. For all values of the forgetting factor λ, the matrix $\mathbf{\Phi}_j^{-1}(t+1)$ is updated according to (4.15) with complexity $\mathcal{O}(p^2)$. It is recommended to initialize the matrix recursion as $\mathbf{\Phi}_j^{-1}(0) = J\mathbf{\Phi}_0^{-1} := \delta\mathbf{I}_p$, where $\delta > 0$ is chosen sufficiently large [43]. Not surprisingly, by direct application of the convergence results in [59, Proposition 3], it follows that:

Proposition 4.2 *For arbitrarily initialized* $\{\mathbf{v}_j^{j'}(t;0)\}_{j\in\mathcal{J}}^{j'\in\mathcal{N}_j}$, $\mathbf{s}_j(t;0)$ *and* $c \in (0, c_u)$*; the local estimates* $\mathbf{s}_j(t;k)$ *generated by (4.25) reach consensus as* $k \to \infty$*; i.e.,*

$$\lim_{k\to\infty}\mathbf{s}_j(t;k) = \hat{\mathbf{s}}_{ewls}(t), \textit{ for all } j \in \mathcal{J}.$$

Remark 4.5 (On the selection of the penalty coefficient) While the AMA-based D-RLS algorithm is less complex computationally than its AD-MoM counterpart in Section

Algorithm 5 : AMA-based STD-RLS

```
Arbitrarily initialize $\{\mathbf{s}_j(0)\}_{j=1}^J$ and $\{\mathbf{v}_j^{j'}(-1)\}_{j\in\mathcal{J}}^{j'\in\mathcal{N}_j}$.
for $t = 0, 1, \ldots$ do
    All $j \in \mathcal{J}$: transmit $\mathbf{s}_j(t)$ to neighbors in $\mathcal{N}_j$.
    All $j \in \mathcal{J}$: update $\{\mathbf{v}_j^{j'}(t)\}_{j'\in\mathcal{N}_j}$ using (4.26).
    All $j \in \mathcal{J}$: transmit $\mathbf{v}_j^{j'}(t)$ to each $j' \in \mathcal{N}_j$.
    All $j \in \mathcal{J}$: update $\mathbf{s}_j(t+1)$ using (4.27).
end for
```

4.3.1, convergence to the centralized estimator is only guaranteed if $c \in (0, c_u)$. The upper bound c_u is proportional to the modulus of the strictly convex cost function in (4.4), and inversely proportional to the norm of the matrix $\mathbf{A}$ given in Appendix 4.6.1; further details are in [59, Section 4]. On the other hand, the AD-MoM-based D-RLS algorithm will attain the EWLSE for any $c > 0$ (cf. Proposition 4.1), and it does not require tuning the extra parameter δ, since it is applicable when $\mathbf{\Phi}_0 = \mathbf{0}$.

By running a single consensus iteration per acquired observation $x_j(t)$, i.e., letting $k = t$ in recursions (4.11)-(4.25), one arrives at a single time scale D-RLS algorithm which is suitable for operation in nonstationary WSN environments. Accounting also for additive communication noise that corrupts the exchanges of multipliers and local estimates, the per sensor tasks comprising the novel AMA-based STD-RLS algorithm are given by

$$\mathbf{v}_j^{j'}(t) = \mathbf{v}_j^{j'}(t-1) + \frac{c}{2}\left[\mathbf{s}_j(t) - (\mathbf{s}_{j'}(t) + \boldsymbol{\eta}_j^{j'}(t))\right], \quad j' \in \mathcal{N}_j \tag{4.26}$$

$$\mathbf{s}_j(t+1) = \mathbf{\Phi}_j^{-1}(t+1)\boldsymbol{\psi}_j(t+1) - \frac{1}{2}\mathbf{\Phi}_j^{-1}(t+1)\sum_{j'\in\mathcal{N}_j}\left[\mathbf{v}_j^{j'}(t) - (\mathbf{v}_{j'}^{j}(t) + \bar{\boldsymbol{\eta}}_j^{j'}(t))\right] \tag{4.27}$$

and are tabulated as Algorithm 5. When powerful error control codes render inter-sensor links virtually ideal, the simplifications in Section 4.4.1 are applicable to the AMA-based (ST)D-RLS also.

4.5 Stability and Steady-State Performance Analysis

In this section, stability and steady-state performance analyses are conducted for the STD-RLS algorithm developed in Section 4.4.2. The techniques presented here can be utilized with minimal modifications to analyze the AD-MoM-based D-RLS too.

Performance evaluation of the D-RLS algorithm is much more involved than that of D-LMS. The challenges are well documented for the classical (centralized) LMS and RLS filters [43, 52], and results for the latter are less common and typically involve simplifying approximations. What is more, the distributed setting studied in this thesis introduces unique challenges in the analysis. These include space-time sensor data and multiple sources of additive noise, a consequence of imperfect sensors and communication links. When dealing with D-RLS, the main complications are rooted in the stochastic matrix $\mathbf{\Phi}_j^{-1}(t)$ present in the local estimate updates (4.27). Recalling that

$$\mathbf{\Phi}_j(t) := \sum_{\tau=0}^{t} \lambda^{t-\tau}\mathbf{h}_j(\tau)\mathbf{h}_j^T(\tau) + J^{-1}\lambda^t\mathbf{\Phi}_0 \tag{4.28}$$

it is apparent that $\mathbf{\Phi}_j^{-1}(t)$ depends upon the *whole history* of local regression vectors $\{\mathbf{h}_j(\tau)\}_{\tau=0}^{t}$. Even obtaining $\mathbf{\Phi}_j^{-1}(t)$'s distribution or computing its expected value is a formidable task in general, due to the matrix inversion operation. It is for these reasons that some simplifying approximations will be adopted in the sequel, to carry out the analysis that otherwise becomes intractable.

In order to proceed, some modeling assumptions are introduced which delineate the scope of the ensuing stability and performance results. For all $j \in \mathcal{J}$, it is assumed that:

(a1) *Sensor observations obey* $x_j(t) = \mathbf{h}_j^T(t)\mathbf{s}_0 + \epsilon_j(t)$, *where the zero-mean white noise* $\{\epsilon_j(t)\}$ *has variance* $\sigma_{\epsilon_j}^2$;

(a2) Vectors $\{\mathbf{h}_j(t)\}$ *are spatio-temporally white with covariance matrix* $\mathbf{R}_{h_j} \succ \mathbf{0}_{p\times p}$; *and*

(a3) Vectors $\{\mathbf{h}_j(t)\}$, $\{\epsilon_j(t)\}$, $\{\boldsymbol{\eta}_j^{j'}(t)\}_{j'\in\mathcal{N}_j}$ *and* $\{\bar{\boldsymbol{\eta}}_j^{j'}(t)\}_{j'\in\mathcal{N}_j}$ *are independent.*

Assumptions (a1)-(a3) comprise the widely adopted *independence setting*, for sensor observations that are linearly related to the time-invariant parameter of interest; see e.g., [52, pg.

110], [43, pg. 448]. Clearly, (a2) can be violated in, e.g., FIR filtering of signals (regressors) with a shift structure as in the distributed power spectrum estimation problem described in Section 4.2.1. Nevertheless, the steady-state performance results extend accurately to the pragmatic setup that involves time-correlated sensor data; see also the numerical tests in Section 4.5.3.

Neglecting the regularization term in (4.28) that vanishes exponentially as $t \to \infty$, the matrix $\mathbf{\Phi}_j(t)$ is obtained as an exponentially weighted moving average (EWMA). The EWMA can be seen as an average modulated by a sliding window of equivalent length $1/(1-\lambda)$, which clearly grows as $\lambda \to 1$. This observation in conjunction with (a2) and the strong law of large numbers, justifies the approximation

$$\mathbf{\Phi}_j(t) \approx E[\mathbf{\Phi}_j(t)] = \frac{\mathbf{R}_{h_j}}{1-\lambda}, \quad 0 \ll \lambda < 1 \text{ and } t \to \infty. \tag{4.29}$$

The mean of $\mathbf{\Phi}_j^{-1}(t)$, on the other hand, is considerably harder to evaluate. To overcome this challenge, we will invoke the following approximation [8, 43]

$$E[\mathbf{\Phi}_j^{-1}(t)] \approx E[\mathbf{\Phi}_j(t)]^{-1} \approx (1-\lambda)\mathbf{R}_{h_j}^{-1}, \quad 0 \ll \lambda < 1 \text{ and } t \to \infty. \tag{4.30}$$

It is clearly a crude approximation at first sight, because $E\left[X^{-1}\right] \neq E[X]^{-1}$ for any random variable X. However, experimental evidence suggests that the approximation is sufficiently accurate for all practical purposes, when the forgetting factor approaches unity [43, p. 319].

Our approach to steady-sate performance analysis relies on an 'averaged' error-form system representation of D-RLS, where $\mathbf{\Phi}_j^{-1}(t)$ in (4.27) is replaced by the approximation $(1-\lambda)\mathbf{R}_{h_j}^{-1}$. It is then straightforward to establish that local estimates obtained via the AMA-based D-RLS are asymptotically unbiased, the subject dealt with next.

4.5.1 Mean Stability

Aiming at representing the approximate 'averaged' D-RLS system as a first-order difference equation, introduce the local estimation errors $\{\mathbf{y}_{1,j}(t) := \mathbf{s}_j(t) - \mathbf{s}_0(t)\}_{j=1}^J$ and multiplier-based quantities $\{\mathbf{y}_{2,j}(t) := \frac{1}{2}\sum_{j' \in \mathcal{N}_j}(\mathbf{v}_j^{j'}(t-1) - \mathbf{v}_{j'}^{j}(t-1))\}_{j=1}^J$. It turns out that a convenient global state capturing the spatio-temporal dynamics of D-RLS in (4.26)-(4.27) can be defined as $\mathbf{y}(t) := [\mathbf{y}_1^T(t)\, \mathbf{y}_2^T(t)]^T = [\mathbf{y}_{1,1}^T(t) \dots \mathbf{y}_{1,J}^T(t)\, \mathbf{y}_{2,1}^T(t) \dots \mathbf{y}_{2,J}^T(t)]^T$. To concisely capture the effects of both observation and communication noise on the estimation errors across the WSN, define the $Jp \times 1$ noise vectors $\boldsymbol{\epsilon}(t) := \sum_{\tau=0}^{t} \lambda^{t-\tau}[\mathbf{h}_1^T(\tau)\epsilon_1(\tau) \dots \mathbf{h}_J^T(\tau)\epsilon_J(\tau)]^T$ and $\bar{\boldsymbol{\eta}}(t) := [\bar{\boldsymbol{\eta}}_1^T(t) \dots \bar{\boldsymbol{\eta}}_J^T(t)]^T$, where vectors $\{\bar{\boldsymbol{\eta}}_j(t)\}_{j \in \mathcal{J}}$ are given by

$$\bar{\boldsymbol{\eta}}_j(t) := \frac{1}{2} \sum_{j' \in \mathcal{N}_j} \bar{\boldsymbol{\eta}}_j^{j'}(t). \tag{4.31}$$

Their respective covariance matrices are easily computable under (a2)-(a3). For instance, $\mathbf{R}_{\boldsymbol{\epsilon}}(t) := E[\boldsymbol{\epsilon}(t)\boldsymbol{\epsilon}^T(t)] = \frac{1-\lambda^{2(t+1)}}{1-\lambda^2}\text{bdiag}(\mathbf{R}_{h_1}\sigma_{\epsilon_1}^2, \dots, \mathbf{R}_{h_J}\sigma_{\epsilon_J}^2)$ while the structure of $\mathbf{R}_{\bar{\boldsymbol{\eta}}} := E[\bar{\boldsymbol{\eta}}(t)\bar{\boldsymbol{\eta}}^T(t)]$ is given in Appendix 4.6.3. Two additional $Jp \times 1$ communication noise supervectors are needed, namely $\boldsymbol{\eta}_\alpha(t) := \left[(\boldsymbol{\eta}_1^\alpha(t))^T \dots (\boldsymbol{\eta}_J^\alpha(t))^T\right]^T$ and $\boldsymbol{\eta}_\beta(t) := \left[(\boldsymbol{\eta}_1^\beta(t))^T \dots (\boldsymbol{\eta}_J^\beta(t))^T\right]^T$, where for $j \in \mathcal{J}$

$$\boldsymbol{\eta}_j^\alpha(t) := \frac{c}{4} \sum_{j' \in \mathcal{N}_j} \boldsymbol{\eta}_j^{j'}(t), \quad \boldsymbol{\eta}_j^\beta(t) := \frac{c}{4} \sum_{j' \in \mathcal{N}_j} \boldsymbol{\eta}_{j'}^{j}(t). \tag{4.32}$$

Finally, let $(c/2)\mathbf{L} \otimes \mathbf{I}_p \in \mathbb{R}^{Jp \times Jp}$ be a matrix capturing the WSN connectivity pattern through the (scaled) graph Laplacian matrix $\mathbf{L}$, and define $\mathbf{R}_h^{-1} := \text{bdiag}(\mathbf{R}_{h_1}^{-1}, \dots, \mathbf{R}_{h_J}^{-1})$. Based on these definitions, it is possible to establish the following important lemma.

Lemma 4.1 *Let (a1) and (a2) hold. Then for $t \geq t_0$ with t_0 sufficiently large while $0 \ll \lambda < 1$, the global state $\mathbf{y}(t)$ approximately evolves according to*

$$\mathbf{y}(t+1) = bdiag((1-\lambda)\mathbf{R}_h^{-1}, \mathbf{I}_{Jp}) \left\{ \boldsymbol{\Upsilon}\mathbf{y}(t) + \begin{bmatrix} \mathbf{I}_{Jp} \\ \mathbf{0} \end{bmatrix} \boldsymbol{\epsilon}(t+1) + \begin{bmatrix} \mathbf{I}_{Jp} \\ \mathbf{0} \end{bmatrix} \bar{\boldsymbol{\eta}}(t) \right.$$
$$\left. + \begin{bmatrix} \mathbf{I}_{Jp} \\ -\mathbf{I}_{Jp} \end{bmatrix} \boldsymbol{\eta}_\alpha(t) - \begin{bmatrix} \mathbf{I}_{Jp} \\ -\mathbf{I}_{Jp} \end{bmatrix} \boldsymbol{\eta}_\beta(t) \right\} \tag{4.33}$$

where the $2Jp \times 2Jp$ matrix $\boldsymbol{\Upsilon}$ consists of the $Jp \times Jp$ blocks $[\boldsymbol{\Upsilon}]_{11} = -[\boldsymbol{\Upsilon}]_{21} = -\mathbf{L}_c$ and $[\boldsymbol{\Upsilon}]_{12} = -[\boldsymbol{\Upsilon}]_{22} = -\mathbf{I}_{Jp}$. The initial condition $\mathbf{y}(t_0)$ should be selected as $\mathbf{y}(t_0) = bdiag(\mathbf{I}_{Jp}, \mathbf{L}_c)\mathbf{y}'(t_0)$, where $\mathbf{y}'(t_0)$ is any vector in $\mathbb{R}^{2Jp}$.

Proof: After replacing $\boldsymbol{\Phi}_j^{-1}(t)$ in (4.27) with the approximation (4.30) for its expected value, the arguments in Appendix 3.7.2 apply directly to yield the desired result. In the interest of brevity, the details of the proof are omitted. □

Based on Lemma 4.1, it follows that D-RLS achieves consensus in the mean sense on the parameter $\mathbf{s}_0$. The proof of the following mean stability result is deferred to Appendix 4.6.2.

Proposition 4.3 *Under (a1)-(a3) and for $0 \ll \lambda < 1$, the D-RLS algorithm achieves consensus in the mean, i.e.,*

$$\lim_{t \to \infty} E[\mathbf{y}_{1,j}(t)] = \mathbf{0}_p, \quad \forall\, j \in \mathcal{J}$$

provided the penalty coefficient is chosen such that

$$0 < c < \frac{4}{(1-\lambda)\lambda_{\max}(\mathbf{R}_h^{-1}(\mathbf{L} \otimes \mathbf{I}_p))}. \tag{4.34}$$

Before concluding this section, a comment is due on the sufficient condition (4.34). When performing distributed estimation under $0 \ll \lambda < 1$, the condition is actually not restrictive at all since a $1-\lambda$ factor is present in the denominator. When λ is close to one, any practical choice of $c > 0$ will result in asymptotically unbiased sensor estimates.

4.5.2 MSE Performance in Steady-State

In order to assess the steady-state MSE performance of the D-RLS algorithm, we will evaluate the figures of merit introduced in Section 3.4.2 of Chapter 3, when investigating the performance of D-LMS. The limiting values of both the local (per sensor) and global (network) MSE, excess mean-square error (EMSE) and mean-square deviation (MSD) will be assessed. To this end, it suffices to derive a closed-form expression for the global estimation error covariance matrix $\mathbf{R}_{y_1}(t) := E[\mathbf{y}_1(t)\mathbf{y}_1^T(t)]$, as already argued in Chapter 3.

The next result provides an equivalent representation of the approximate D-RLS global recursion (4.33), that is more suitable for the recursive evaluation of $\mathbf{R}_{y_1}(t)$. First, introduce the $p(\sum_{j=1}^{J} |\mathcal{N}_j|) \times 1$ vector

$$\boldsymbol{\eta}(t) := \left[\{(\boldsymbol{\eta}_{j'}^1(t))^T\}_{j' \in \mathcal{N}_1} \dots \{(\boldsymbol{\eta}_{j'}^J(t))^T\}_{j' \in \mathcal{N}_J}\right]^T \tag{4.35}$$

which comprises the receiver noise terms corrupting transmissions of local estimates across the whole network at time instant t, and define $\mathbf{R}_{\boldsymbol{\eta}} := E[\boldsymbol{\eta}(t)\boldsymbol{\eta}^T(t)]$. For notational convenience, let $\mathbf{R}_{h,\lambda}^{-1} := (1-\lambda)\mathbf{R}_h^{-1}$.

Lemma 4.2 *Under the assumptions stated in Lemma 4.1, the global state* $\mathbf{y}(t)$ *in (4.33) can be equivalently written as*

$$\mathbf{y}(t+1) = bdiag(\mathbf{I}_{Jp}, \mathbf{L}_c)\mathbf{z}(t+1) + \begin{bmatrix} \mathbf{R}_{h,\lambda}^{-1} \\ \mathbf{0}_{Jp\times Jp} \end{bmatrix} \bar{\boldsymbol{\eta}}(t) + \begin{bmatrix} \mathbf{R}_{h,\lambda}^{-1}(\mathbf{P}_\alpha - \mathbf{P}_\beta) \\ \mathbf{P}_\beta - \mathbf{P}_\alpha \end{bmatrix} \boldsymbol{\eta}(t) \qquad (4.36)$$

where the inner state $\mathbf{z}(t) := [\mathbf{z}_1^T(t)\ \mathbf{z}_2^T(t)]^T$ *is arbitrarily initialized at time instant* t_0 *and updated according to*

$$\mathbf{z}(t+1) = \boldsymbol{\Psi}\mathbf{z}(t) + \boldsymbol{\Psi} \begin{bmatrix} \mathbf{R}_{h,\lambda}^{-1}(\mathbf{P}_\alpha - \mathbf{P}_\beta) \\ \mathbf{C} \end{bmatrix} \boldsymbol{\eta}(t-1) + \boldsymbol{\Psi} \begin{bmatrix} \mathbf{R}_{h,\lambda}^{-1} \\ \mathbf{0} \end{bmatrix} \bar{\boldsymbol{\eta}}(t-1) + \begin{bmatrix} \mathbf{R}_{h,\lambda}^{-1} \\ \mathbf{0} \end{bmatrix} \boldsymbol{\epsilon}(t+1) \qquad (4.37)$$

and the $2Jp \times 2Jp$ *transition matrix* $\boldsymbol{\Psi}$ *consists of the blocks* $[\boldsymbol{\Psi}]_{11} = [\boldsymbol{\Psi}]_{12} = -\mathbf{R}_{h,\lambda}^{-1}\mathbf{L}_c$ *and* $[\boldsymbol{\Psi}]_{21} = [\boldsymbol{\Psi}]_{22} = \mathbf{L}_c\mathbf{L}_c^\dagger$*. The matrix* $\mathbf{C}$ *is chosen such that* $\mathbf{L}_c\mathbf{C} = \mathbf{P}_\beta - \mathbf{P}_\alpha$*, where the structure of the time-invariant matrices* $\mathbf{P}_\alpha$ *and* $\mathbf{P}_\beta$ *is given in Appendix 4.6.3.*

Proof: The equivalence follows by induction, replicating the steps in Appendix 3.7.2. □

Focusing now on the calculation of $\mathbf{R}_{y_1}(t)$, observe from the upper $Jp\times 1$ block of $\mathbf{y}(t+1)$ in (4.36) that $\mathbf{y}_1(t+1) = \mathbf{z}_1(t+1) + \mathbf{R}_{h,\lambda}^{-1}[\bar{\boldsymbol{\eta}}(t) + (\mathbf{P}_\alpha - \mathbf{P}_\beta)\boldsymbol{\eta}(t)]$. Under (a3), $\mathbf{z}_1(t+1)$ is independent of the zero-mean $\{\bar{\boldsymbol{\eta}}(t), \boldsymbol{\eta}(t)\}$; hence,

$$\mathbf{R}_{y_1}(t) = \mathbf{R}_{z_1}(t) + \mathbf{R}_{h,\lambda}^{-1}\left[\mathbf{R}_{\bar{\boldsymbol{\eta}}} + (\mathbf{P}_\alpha - \mathbf{P}_\beta)\mathbf{R}_{\boldsymbol{\eta}}(\mathbf{P}_\alpha - \mathbf{P}_\beta)^T\right]\mathbf{R}_{h,\lambda}^{-1} \qquad (4.38)$$

based on which we obtain $\mathbf{R}_z(t) := E[\mathbf{z}(t)\mathbf{z}^T(t)]$. We wish to extract its upper-left $Jp \times Jp$ matrix block $[\mathbf{R}_z(t)]_{11} = \mathbf{R}_{z_1}(t)$. To this end, define the vectors

$$\bar{\boldsymbol{\eta}}_\lambda(t) := \begin{bmatrix} \mathbf{R}_{h,\lambda}^{-1} \\ \mathbf{0}_{Jp\times Jp} \end{bmatrix} \bar{\boldsymbol{\eta}}(t), \quad \boldsymbol{\eta}_\lambda(t) := \begin{bmatrix} \mathbf{R}_{h,\lambda}^{-1}(\mathbf{P}_\alpha - \mathbf{P}_\beta) \\ \mathbf{C} \end{bmatrix} \boldsymbol{\eta}(t) \qquad (4.39)$$

whose respective covariance matrices $\mathbf{R}_{\bar{\boldsymbol{\eta}}_\lambda} := E[\bar{\boldsymbol{\eta}}_\lambda(t)\bar{\boldsymbol{\eta}}_\lambda^T(t)]$ and $\mathbf{R}_{\boldsymbol{\eta}_\lambda} := E[\boldsymbol{\eta}_\lambda(t)\boldsymbol{\eta}_\lambda^T(t)]$ have a structure detailed in Appendix 4.6.3. Also recall that $\boldsymbol{\epsilon}(t)$ depends on the entire history of regressors up to time instant t. Starting from (4.37) and capitalizing on the independence setting assumptions (a2)-(a3), it is straightforward to obtain a first-order matrix recursion to update $\mathbf{R}_z(t)$ as

$$\begin{aligned}\mathbf{R}_z(t) =&\boldsymbol{\Psi}\mathbf{R}_z(t-1)\boldsymbol{\Psi}^T + \boldsymbol{\Psi}\mathbf{R}_{\bar{\boldsymbol{\eta}}_\lambda}\boldsymbol{\Psi}^T + \boldsymbol{\Psi}\mathbf{R}_{\boldsymbol{\eta}_\lambda}\boldsymbol{\Psi}^T + \begin{bmatrix}\mathbf{R}_{h,\lambda}^{-1}\\ \mathbf{0}_{Jp}\end{bmatrix}\mathbf{R}_{\boldsymbol{\epsilon}}(t)\begin{bmatrix}\mathbf{R}_{h,\lambda}^{-1}\\ \mathbf{0}_{Jp}\end{bmatrix}^T\\ &+\boldsymbol{\Psi}\mathbf{R}_{z\boldsymbol{\epsilon}}(t)\begin{bmatrix}\mathbf{R}_{h,\lambda}^{-1}\\ \mathbf{0}_{Jp}\end{bmatrix} + \left(\boldsymbol{\Psi}\mathbf{R}_{z\boldsymbol{\epsilon}}(t)\begin{bmatrix}\mathbf{R}_{h,\lambda}^{-1}\\ \mathbf{0}_{Jp}\end{bmatrix}\right)^T \qquad (4.40)\\ =&\boldsymbol{\Psi}\mathbf{R}_z(t-1)\boldsymbol{\Psi}^T + \mathbf{R}_{\boldsymbol{\nu}}(t) \qquad (4.41)\end{aligned}$$

where the cross-correlation matrix $\mathbf{R}_{z\boldsymbol{\epsilon}}(t) := E[\mathbf{z}(t-1)\boldsymbol{\epsilon}^T(t)]$ is recursively updated as

$$\mathbf{R}_{z\boldsymbol{\epsilon}}(t) = \lambda\boldsymbol{\Psi}\mathbf{R}_{z\boldsymbol{\epsilon}}(t-1) + \lambda\begin{bmatrix}\mathbf{R}_{h,\lambda}^{-1}\\ \mathbf{0}_{Jp}\end{bmatrix}\mathbf{R}_{\boldsymbol{\epsilon}}(t-1). \qquad (4.42)$$

For notational simplicity in what follows, $\mathbf{R}_{\boldsymbol{\nu}}(t)$ in (4.41) denotes all the covariance forcing terms in the right hand side of (4.40). The fundamental result of this section pertains to MSE stability of the D-RLS algorithm, and provides a checkable sufficient condition under which the global error covariance matrix $\mathbf{R}_{y_1}(t)$ has bounded entries as $t \to \infty$.

Proposition 4.4 *Under (a1)-(a3) and for* $0 \ll \lambda < 1$*, the D-RLS algorithm is MSE stable, i.e.,* $\lim_{t\to\infty}\mathbf{R}_{y_1}(t)$ *has bounded entries, provided that* $c > 0$ *is chosen so that* $\boldsymbol{\Psi}$ *is a stable matrix.*

Proof: First observe that because $\lambda \in (0,1)$, it holds that

$$\begin{aligned}\lim_{t\to\infty}\mathbf{R}_{\boldsymbol{\epsilon}}(t) &= \lim_{t\to\infty}\left(\frac{1-\lambda^{2(t+1)}}{1-\lambda^2}\right)\text{bdiag}(\mathbf{R}_{h_1}\sigma_{\epsilon_1}^2,\ldots,\mathbf{R}_{h_J}\sigma_{\epsilon_J}^2)\\ &= \left(\frac{1}{1-\lambda^2}\right)\text{bdiag}(\mathbf{R}_{h_1}\sigma_{\epsilon_1}^2,\ldots,\mathbf{R}_{h_J}\sigma_{\epsilon_J}^2) =: \mathbf{R}_{\boldsymbol{\epsilon}}(\infty) \qquad (4.43)\end{aligned}$$

If $c > 0$ is selected such that $\boldsymbol{\Psi}$ is a stable matrix, then clearly $\lambda\boldsymbol{\Psi}$ is also stable and hence the matrix recursion (4.42) converges to the bounded limit

$$\mathbf{R}_{z\boldsymbol{\epsilon}}(\infty) = (\mathbf{I}_{2Jp} - \lambda\boldsymbol{\Psi})^{-1}\begin{bmatrix}\lambda\mathbf{R}_{h,\lambda}^{-1}\\ \mathbf{0}_{Jp}\end{bmatrix}\mathbf{R}_{\boldsymbol{\epsilon}}(\infty). \qquad (4.44)$$

Based on the previous arguments, it follows that the forcing matrix $\mathbf{R}_{\boldsymbol{\nu}}(t)$ in (4.40) will also attain a bounded limit as $t \to \infty$, denoted as $\mathbf{R}_{\boldsymbol{\nu}}(\infty)$. Next, we show that $\lim_{t\to\infty}\mathbf{R}_z(t)$ has bounded entries by studying its equivalent vectorized dynamical system. Upon vectorizing (4.41), it follows that

$$\begin{aligned} vec[\mathbf{R}_z(t)] =& vec[\mathbf{\Psi}\mathbf{R}_z(t-1)\mathbf{\Psi}^T] + vec[\mathbf{R}_{\boldsymbol{\nu}}(t)] \\ =& \left(\mathbf{\Psi} \otimes \mathbf{\Psi}\right) vec[\mathbf{R}_z(t-1)] + vec[\mathbf{R}_{\boldsymbol{\nu}}(t)] \end{aligned}$$

where in obtaining the last equality we used the property $vec[\mathbf{RST}] = \left(\mathbf{T}^T \otimes \mathbf{R}\right) vec[\mathbf{S}]$. Because the eigenvalues of $\mathbf{\Psi} \otimes \mathbf{\Psi}$ are the pairwise products of those of $\mathbf{\Psi}$, stability of $\mathbf{\Psi}$ implies stability of the Kronecker product. As a result, the vectorized recursion will converge to the limit

$$vec[\mathbf{R}_z(\infty)] = \left(\mathbf{I}_{(2Jp)^2} - \mathbf{\Psi} \otimes \mathbf{\Psi}\right)^{-1} vec[\mathbf{R}_{\boldsymbol{\nu}}(\infty)] \tag{4.45}$$

which of course implies that $\lim_{t\to\infty} \mathbf{R}_z(t) = \mathbf{R}_z(\infty)$ has bounded entries. From (4.38), the same holds true for $\mathbf{R}_{y_1}(t)$, and the proof is completed. □

Proposition 4.4 asserts that the AMA-based D-RLS algorithm is stable in the mean-square sense, even when the WSN links are challenged by additive noise. It also provides theoretical support for the comments in Remark 4.4, and the results of the numerical experiments in Section 4.3.3. Noise resilience however, is not a feature of the diffusion RLS algorithm in [8], which under non-ideal links yields local sensor estimates with unbounded variance (asymptotically in time).

As a byproduct, the proof of Proposition 4.4 also provides part of the recipe towards evaluating the steady-state MSE performance of the D-RLS algorithm. In detail, by plugging (4.43) and (4.44) into (4.40) one obtains the steady-state covariance matrix $\mathbf{R}_{\boldsymbol{\nu}}(\infty)$. It is then possible to evaluate $\mathbf{R}_z(\infty)$, by reshaping the vectorized identity (4.45). Matrix $\mathbf{R}_{z_1}(\infty)$ can be extracted from the upper-left $Jp \times Jp$ matrix block of $\mathbf{R}_z(\infty)$, and the desired global error covariance matrix $\mathbf{R}_{y_1}(\infty)$ becomes available via (4.38). Closed-form evaluation of the MSE(∞), EMSE(∞) and MSD(∞) for every sensor $j \in \mathcal{J}$ is now possible given $\mathbf{R}_{y_1}(\infty)$, by resorting to the formulae in Section 3.4.2.

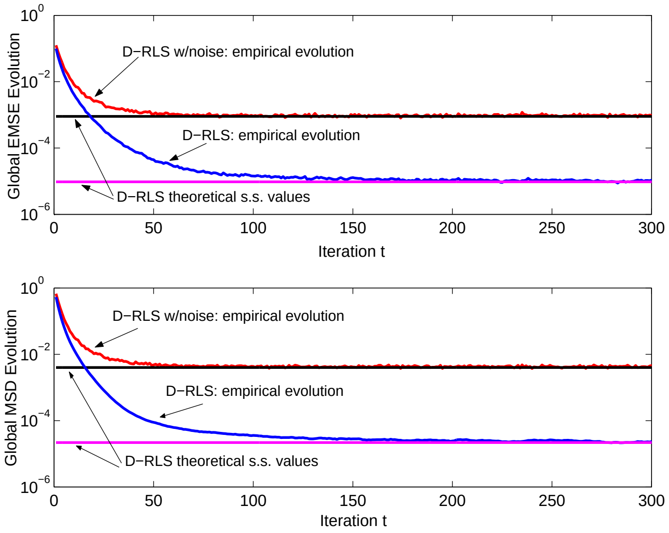

Figure 4.5: Global steady-state performance evaluation. D-RLS is ran with ideal links and when communication noise with variance $\sigma_\eta^2 = 10^{-1}$ is present.

4.5.3 Numerical Tests

Here we validate the analytical results of Section 4.5.2 through numerical experiments. Even though based on simplifying assumptions and approximations, the usefulness of the analysis is corroborated since the predicted steady-state MSE figures of merit accurately match the empirical D-RLS limiting values. For $J = 15$ sensors, a connected ad hoc WSN is generated as a realization of the random geometric graph model on the unity square, with communication range $r = 0.3$. For non-ideal links, additive white Gaussian noise (AWGN) with variance $\sigma_\eta^2 = 10^{-1}$ is added at the receiving end.

With $p = 4$ and $\mathbf{s}_0 = \mathbf{1}_p$, observations obey a linear model [cf. (a1)] with sensing WGN of spatial variance profile $\sigma_{\epsilon_j}^2 = 10^{-3}\alpha_j$, where $\alpha_j \sim \mathcal{U}[0,1]$ (uniform distribution) and i.i.d.. The regression vectors $\mathbf{h}_j(t) = [h_j(t)\dots h_j(t-p+1)]^T$ have a shift structure and entries which evolve according to $h_j(t) = (1-\rho)\beta_j h_j(t-1) + \sqrt{\rho}\omega_j(t)$ for all $j \in \mathcal{J}$. We choose $\rho = 5 \times 10^{-1}$, the $\beta_j \sim \mathcal{U}[0,1]$ i.i.d. in space, and the driving white noise $\omega_j(t) \sim \mathcal{U}[-\sqrt{3}\sigma_{\omega_j}, \sqrt{3}\sigma_{\omega_j}]$ with spatial variance profile given by $\sigma_{\omega_j}^2 = 2\gamma_j$ with $\gamma_j \sim \mathcal{U}[0,1]$ and i.i.d.. Observe that the data is temporally-correlated, implying that (a2) does not hold

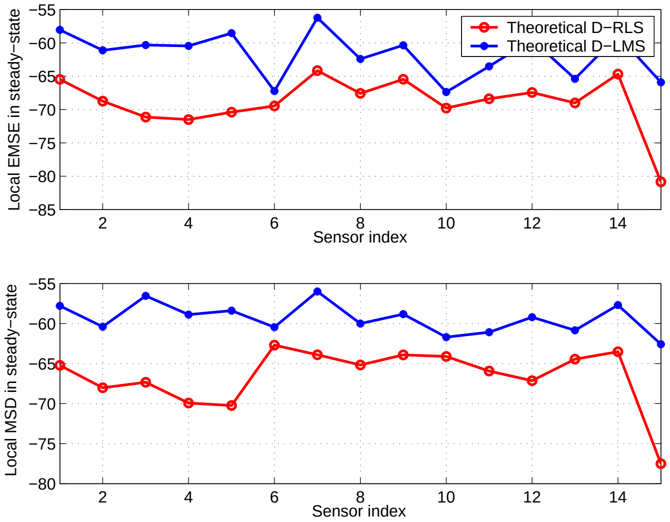

Figure 4.6: Local steady-state performance evaluation. D-RLS is compared to the D-LMS algorithm in Chapter 3.

here.

For all experimental performance curves obtained by running the algorithms, the ensemble averages are approximated by sample averaging 200 runs of the experiment.

First, with $\lambda = 0.99$, $c = 0.1$ and $\delta = 100$ for the AMA-based D-RLS algorithm, Fig. 4.5 depicts the network performance through the evolution of the EMSE(t) and MSD(t) figures of merit. Both noisy and ideal links are considered, while for the latter the D-RLS variant in Section 4.4.1 has been used. The steady-state limiting values found in Section 4.5.2 are extremely accurate, even though the simulated data does not adhere to (a2), and the results are based on simplifying approximations. As intuitively expected and analytically corroborated via the noise-related additive terms in (4.38) and (4.40), the performance penalty due to non-ideal links is also apparent.

We also utilize the analytical results developed throughout this thesis to contrast the per sensor performance of D-RLS and the D-LMS algorithm in Chapter 3. In particular, the parameters chosen for D-LMS are $\mu = 5 \times 10^{-3}$ and $c = 1$. Fig. 4.6 shows the values of the EMSE$_j(\infty)$ and MSD$_j(\infty)$ for all $j \in \mathcal{J}$. As expected, the second-order D-RLS scheme attains improved steady-state performance uniformly across all sensors in the simulated

WSN. In this particular simulated test, gains as high as 10dB in estimation error can be achieved at the price of increasing computational burden per sensor.

4.6 Appendices

4.6.1 Proof of Proposition 4.1

From the equivalence between the optimization problems (4.1) and (4.4), it suffices to show that for all t the iterates $\mathbf{s}_j(t,k)$ generated by (4.12) converge to $\hat{\mathbf{s}}_j(t)$ in (4.4). To this end, observe that (4.4) with its equivalent set of constraints in (4.5) can be written as [7, eq. 4.77, p.255]

$$\min_{\mathbf{x},\mathbf{z}} G_1(\mathbf{x}) + G_2(\mathbf{z}) \tag{4.46}$$

$$\text{s. t. } \mathbf{x} \in C_1,\ \mathbf{z} \in C_2,\ \mathbf{A}\mathbf{x} = \mathbf{z}$$

with the identifications

$$\begin{aligned}
\mathbf{x} &= \left[\mathbf{s}_1^T \ldots \mathbf{s}_J^T\right]^T, \quad C_1 = \mathbb{R}^{Jp}\\
\mathbf{z} &= \left[\bar{\mathbf{z}}^T\ \tilde{\mathbf{z}}^T\right]^T, \quad C_2 = C_z\\
G_1(\mathbf{x}) &= \sum_{\tau=0}^{t}\sum_{j=1}^{J}\lambda^{t-\tau}[x_j(\tau) - \mathbf{h}_j^T(\tau)\mathbf{s}_j]^2 + J^{-1}\lambda^t\sum_{j=1}^{J}\mathbf{s}_j^T\mathbf{\Phi}_0\mathbf{s}_j, \quad G_2(\mathbf{z}) = 0
\end{aligned}$$

where

$$\bar{\mathbf{z}} := \left[(\bar{\mathbf{z}}_1^{i_1(1)})^T \ldots (\bar{\mathbf{z}}_1^{i_1(|\mathcal{N}_1|)})^T \ldots (\bar{\mathbf{z}}_J^{i_J(1)})^T \ldots (\bar{\mathbf{z}}_J^{i_J(|\mathcal{N}_J|)})^T\right]^T$$

and likewise for $\tilde{\mathbf{z}}$. The integer valued functions $i_j : \{1,\ldots,|\mathcal{N}_j|\} \to \mathcal{J}$ are such that $i_j(j')$ is the index of the j'th neighbor of node j. The linear constraint matrix is $\mathbf{A} = [\mathbf{A}_1^T\ \mathbf{A}_2^T\ \mathbf{A}_3^T]^T$, with matrix blocks $\mathbf{A}_1 := \mathbf{I}_{Jp}$ and

$$\mathbf{A}_2 := \begin{bmatrix}\mathbf{A}_{21}\\ \vdots \\ \mathbf{A}_{2J}\end{bmatrix}, \quad \mathbf{A}_{2j} := \left(\mathbf{1}_{|\mathcal{N}_j|}\mathbf{b}_{J,j}^T\right)\otimes\mathbf{I}_p,\ j\in\mathcal{J}$$

$$\mathbf{A}_3 := \begin{bmatrix}\mathbf{A}_{31}\\ \vdots \\ \mathbf{A}_{3J}\end{bmatrix}, \quad \mathbf{A}_{3j} := \begin{bmatrix}\mathbf{b}_{J,i_j(1)}^T\\ \vdots \\ \mathbf{b}_{J,i_j(|\mathcal{N}_j|)}^T\end{bmatrix}\otimes\mathbf{I}_p,\ j\in\mathcal{J}$$

Thus, the three steps used to construct the D-RLS algorithm correspond to the AD-MoM [7, eqs. 4.79-4.81, p.255].

With the identifications above, it is straightforward to check that G_1 and G_2 are convex functions, C_1 and C_2 are nonempty polyhedral sets and $\mathbf{A}^T\mathbf{A}$ is invertible (because $\mathbf{A}$ is full column rank). By virtue of [7, Proposition 4.2, p.256], for every value of $c > 0$ and time instant t the iterates generated by (4.12) converge to the optimal solution $\hat{\mathbf{s}}_j(t) = \hat{\mathbf{s}}_{\text{ewls}}(t)$, $j \in \mathcal{J}$ of problem (4.4). □

4.6.2 Proof of Proposition 4.3

Based on the independence setting assumptions (a1)-(a3) and since the data is zero-mean, one obtains after taking expectations on (4.33) that $E[\mathbf{y}(t)] = \text{bdiag}((1-\lambda)\mathbf{R}_h^{-1}, \mathbf{I}_{Jp})\boldsymbol{\Upsilon}E[\mathbf{y}(t-1)]$. The following lemma characterizes the spectrum of the transition matrix $\boldsymbol{\Omega} := \text{bdiag}((1-\lambda)\mathbf{R}_h^{-1}, \mathbf{I}_{Jp})\boldsymbol{\Upsilon}$

Lemma 4.3 *Regardless of the value of $c > 0$, matrix $\boldsymbol{\Omega} := bdiag((1-\lambda)\mathbf{R}_h^{-1}, \mathbf{I}_{Jp})\boldsymbol{\Upsilon} \in \mathbb{R}^{2Jp\times 2Jp}$ has p eigenvalues equal to one. Further, the left eigenvectors associated with the unity eigenvalue have the structure $\mathbf{v}_i^T = \left[\mathbf{0}_{1\times Jp}\ \mathbf{q}_i^T\right]$ where $\mathbf{q}_i \in nullspace(\mathbf{L}_c)$ and $i = 1, \ldots, p$. The remaining eigenvalues are equal to zero, or else have modulus strictly smaller than one provided c satisfies the bound 4.34.*

Proof: Recall the structure of matrix $\boldsymbol{\Omega}$ given in Lemma 4.1. A vector $\mathbf{v}_i^T = \left[\mathbf{v}_{1,i}^T\ \mathbf{v}_{2,i}^T\right]$ with $\{\mathbf{v}_{j,i}\}_{j=1}^2 \in \mathbb{R}^{Jp\times 1}$ is a left eigenvector of $\boldsymbol{\Omega}$ associated to the eigenvalue one, if an only if it solves the following linear system of equations

$$-\mathbf{v}_{1,i}^T(1-\lambda)\mathbf{R}_h^{-1}\mathbf{L}_c + \mathbf{v}_{2,i}^T\mathbf{L}_c = \mathbf{v}_{1,i}^T$$
$$-\mathbf{v}_{1,i}^T(1-\lambda)\mathbf{R}_h^{-1} + \mathbf{v}_{2,i}^T = \mathbf{v}_{2,i}^T$$

The second equation can only be satisfied for $\mathbf{v}_{1,i} = \mathbf{0}_{Jp}$, and upon substituting this value in the first equation one obtains that $\mathbf{v}_{2,i} \in \text{nullspace}(\mathbf{L}_c) = \text{nullspace}(\mathbf{L}\otimes\mathbf{I}_p)$ for all values of $c > 0$. Under the assumption of a connected ad hoc WSN, then $\text{nullspace}(\mathbf{L}) = \text{span}(\mathbf{1}_J)$ and hence $\text{nullspace}(\mathbf{L}\otimes\mathbf{I}_p)$ is a p-dimensional subspace.

Following steps similar to those in [49, Appendix H], it is possible to express the eigenvalues of $\boldsymbol{\Omega}$ that are different from one as the roots of a second-order polynomial. Such polynomial does not have an independent term, so that some eigenvalues are zero. With respect to the rest of the eigenvalues, it is possible to show that their magnitude is upper bounded by $\lambda_{\max}(\mathbf{I}_{Jp} - (1-\lambda)\mathbf{R}_h^{-1}\mathbf{L}_c)$. Note that

$$\mathbf{R}_h^{1/2}\left[(1-\lambda)\mathbf{R}_h^{-1}\mathbf{L}_c\right]\mathbf{R}_h^{-1/2} = \frac{c}{2}(1-\lambda)\mathbf{R}_h^{-1/2}(\mathbf{L}\otimes\mathbf{I}_p)\mathbf{R}_h^{-1/2} \tag{4.47}$$

has the same eigenvalues as $(1-\lambda)\mathbf{R}_h^{-1}\mathbf{L}_c$ because these are invariant under similarity transformations. Focusing on the right hand side of (4.47), from Sylvester's law of inertia [19, p. 403] it follows that all eigenvalues of $(1-\lambda)\mathbf{R}_h^{-1}\mathbf{L}_c$ are real and nonnegative. Hence, it is possible to select $c > 0$ such that $\lambda_{\max}(\mathbf{I}_{Jp} - (1-\lambda)\mathbf{R}_h^{-1}\mathbf{L}_c) < 1$, or equivalently $|1-(1-\lambda)\lambda_{\max}(\mathbf{R}_h^{-1}\mathbf{L}_c)| < 1$ which is the same as condition 4.34. □

Back to establishing the mean stability result, let $\{\mathbf{u}_i\}$ and $\{\mathbf{v}_i^T\}$ respectively denote the collection of p right and left eigenvectors of $\mathbf{\Omega}$ associated with the eigenvalue one. By virtue of Lemma 4.3 we have that $\lim_{t\to\infty}\mathbf{\Omega}^t = \sum_{i=1}^p \mathbf{u}_i\mathbf{v}_i^T$ and consequently

$$\begin{aligned}\lim_{t\to\infty} E[\mathbf{y}(t)] &= \left(\sum_{i=1}^p \mathbf{u}_i\mathbf{v}_i^T\right)\mathbf{y}(t_0)\\ &= \left(\sum_{i=1}^p \mathbf{u}_i\mathbf{v}_i^T\right)\mathrm{bdiag}(\mathbf{I}_{Jp},\mathbf{L}_c)\mathbf{y}'(t_0)\\ &= \left(\sum_{i=1}^p \mathbf{u}_i\left[\mathbf{0}_{1\times Jp}\ \ \mathbf{q}_i^T\mathbf{L}_c\right]\right)\mathbf{y}'(t_0) = \mathbf{0}_{2Jp}.\end{aligned}$$

In obtaining the second equality, we used the structure for $\mathbf{y}(t_0)$ that is given in Lemma 4.1. The last equality follows from the fact that $\mathbf{q}_i \in \mathrm{nullspace}(\mathbf{L}_c)$ as per Lemma 4.3. □

4.6.3 Structure of Matrices $\mathbf{P}_\alpha$, $\mathbf{P}_\beta$, $\mathbf{R}_{\bar{\boldsymbol{\eta}}_\lambda}$ and $\mathbf{R}_{\boldsymbol{\eta}_\lambda}$

In order to relate the noise vectors $\boldsymbol{\eta}_\alpha(t)$ and $\boldsymbol{\eta}_\beta(t)$ present in (4.33) with $\boldsymbol{\eta}(t)$ in (4.35), we introduce two $Jp \times (\sum_{j=1}^J |\mathcal{N}_j|)p$ matrices $\mathbf{P}_\alpha := [\mathbf{p}_1 \dots \mathbf{p}_J]^T$ and $\mathbf{P}_\beta := [\mathbf{p}'_1 \dots \mathbf{p}'_J]^T$. The $(\sum_{j=1}^J |\mathcal{N}_j|)p \times p$ submatrices $\mathbf{p}_j$, $\mathbf{p}'_j$ are given by $\mathbf{p}_j := [(\mathbf{p}_{j,1})^T \dots (\mathbf{p}_{j,J})^T]^T$ and

$\mathbf{p}'_j := [(\mathbf{p}'_{j,1})^T \dots (\mathbf{p}'_{j,J})^T]^T$, with $\mathbf{p}_{j,r}, \mathbf{p}_{j',r}$ defined for $r = 1, \dots, J$ as

$$\mathbf{p}_{j,r}^T := \begin{cases} \frac{c}{4}\mathbf{b}^T_{|\mathcal{N}_r|,r(j)} \otimes \mathbf{I}_p & \text{if } j \in \mathcal{N}_r \\ \mathbf{0}_{p\times|\mathcal{N}_r|p} & \text{if } j \notin \mathcal{N}_r \end{cases}$$

$$(\mathbf{p}'_{j,r})^T := \begin{cases} \frac{c}{4}\mathbf{1}_{1\times|\mathcal{N}_r|} \otimes \mathbf{I}_p & \text{if } r = j \\ \mathbf{0}_{p\times|\mathcal{N}_r|p} & \text{if } r \neq j \end{cases} .$$

Note that $r(j) \in \{1, \dots, |\mathcal{N}_r|\}$ denotes the order in which $\boldsymbol{\eta}_j^r(t)$ appears in $\{\boldsymbol{\eta}_{j'}^r(t)\}_{j'\in\mathcal{N}_r}$ [cf. (4.35)]. It is straightforward to verify that $\boldsymbol{\eta}_\alpha(t) = \mathbf{P}_\alpha\boldsymbol{\eta}(t)$ and $\boldsymbol{\eta}_\beta(t) = \mathbf{P}_\beta\boldsymbol{\eta}(t)$.

Now we focus on the structure of the noise covariance matrices $\mathbf{R}_{\bar{\boldsymbol{\eta}}_\lambda}$ and $\mathbf{R}_{\boldsymbol{\eta}_\lambda}$. From (4.39) we have

$$\mathbf{R}_{\bar{\boldsymbol{\eta}}_\lambda} = \begin{bmatrix} \mathbf{R}_{h,\lambda}^{-1} \\ \mathbf{0}_{Jp\times Jp} \end{bmatrix} \mathbf{R}_{\bar{\boldsymbol{\eta}}} \begin{bmatrix} \mathbf{R}_{h,\lambda}^{-1} \\ \mathbf{0}_{Jp\times Jp} \end{bmatrix}^T$$

$$\mathbf{R}_{\boldsymbol{\eta}_\lambda} = \begin{bmatrix} \mathbf{R}_{h,\lambda}^{-1}(\mathbf{P}_\alpha - \mathbf{P}_\beta) \\ \mathbf{C} \end{bmatrix} \mathbf{R}_{\boldsymbol{\eta}} \begin{bmatrix} \mathbf{R}_{h,\lambda}^{-1}(\mathbf{P}_\alpha - \mathbf{P}_\beta) \\ \mathbf{C} \end{bmatrix}^T$$

so that it suffices to focus on the structure of $\mathbf{R}_{\bar{\boldsymbol{\eta}}}$ and $\mathbf{R}_{\boldsymbol{\eta}}$. From the definition in (4.31) and recalling that communication noise vectors are assumed uncorrelated in space, it follows that

$$\mathbf{R}_{\bar{\boldsymbol{\eta}}} = \frac{1}{4}\text{bdiag}\left(\sum_{j'\in\mathcal{N}_1\setminus\{1\}} \mathbf{R}_{\boldsymbol{\eta}_{1,j'}}, \dots, \sum_{j'\in\mathcal{N}_J\setminus\{J\}} \mathbf{R}_{\boldsymbol{\eta}_{J,j'}}\right).$$

The structure of $\mathbf{R}_{\boldsymbol{\eta}}$ is given in Appendix 3.7.3.

Chapter 5

Conclusions and Future Work

In this dissertation we dealt with distributed adaptive estimation using ad hoc WSNs. In this final chapter we provide a summary of the main results in the thesis, and point out possible directions for future research.

5.1 Thesis Summary

In Chapter 2, a distributed LMS-type of adaptive algorithm is developed for WSN based tracking applications. Inter-sensor communications are constrained to single-hop neighboring sensors and are challenged by the effects of additive receiver noise. The desired LMS estimator is posed as a convex optimization problem, which is reformulated into an equivalent constrained form whose structure lends itself naturally to decentralized implementation. This favorable structure is exploited by resorting to the AD-MoM, while using stochastic approximation tools in the process to finally arrive at simple recursions. The resulting in-network processing per sensor was interpreted as a local-LMS adaptation rule superimposed to the output of a tunable PI regulator, which drives the local estimate to consensus as dictated by a network-wide information enriched reference. Numerical examples illustrated that D-LMS outperforms comparable adaptive schemes, and has the potential of tracking nonstationary processes.

The challenging problem of algorithm stability in a stochastic sense has been also addressed. The distributed stability results obtained parallel those available for the centralized setting. For observations adhering to a linear model, stationary ergodic regressors, and a fixed step-size below a positive threshold, D-LMS provably incurs local estimation errors satisfying the WSB property even in the presence of additive inter-sensor communication noise. In the absence of noise, D-LMS estimates were shown almost surely exponentially convergent to the true parameter of interest. With regards to performance analysis, a stochastic trajectory locking result was established which shows that for small step-sizes, the D-LMS estimation error trajectories closely follow the ones of its time-invariant ‘averaged’ system mate. An ‘averaged’ system estimation error covariance matrix was obtained in closed form, that provided a means of accurately approximating the actual D-LMS estimation MSE as corroborated by numerical simulations.

To complement the D-LMS estimation framework in Chapter 2, a fully-distributed variant to the algorithm therein is developed in Chapter 3. The improved D-LMS algorithm does not require a subset of bridge sensors, hence all nodes perform identical tasks in the process of consenting on the network-wide estimates adaptively. Different from existing alternatives, the novel scheme does not require a Hamiltonian cycle and can tackle linear regression problems in which a statistical data model is not available. When communication noise is not an issue, a cost-effective variant of D-LMS can be used which circumvents communicating Lagrange multipliers yet incurs no performance penalty.

A detailed MSE tracking performance analysis was conducted for D-LMS, when the parameter fluctuations adhere to a stable first-order AR model. By deriving an exact recursion for the global error covariance matrix under the white Gaussian setting assumptions, the network-wide and per-sensor performance metrics became available for any t, and in particular as $t \to \infty$. D-LMS was shown stable in the mean and MSE-sense in the presence of additive receiver noise, provided $\mu > 0$ is sufficiently small. As a corollary, the resulting local estimation errors satisfy the WSB property and hence remain within a finite interval with overwhelming probability. The tracking analysis led to the conclusion that – different from the time-invariant case whereby one should decrease μ to reduce the steady-state error – for a slowly time-varying parameter there exists an optimal $\mu^{\star}$. While a vanishing step-size renders D-LMS incapable of adapting to the underlying variations, a large one amplifies both observation and communication noise. Numerical simulations demonstrated that these analytical findings carry over to more pragmatic setups, including temporally correlated (non-)Gaussian sensor data as considered in the analysis presented in Chapter 2.

Chapters 2 and 3 dealt with distributed least mean-square estimation and tracking for ad hoc WSNs. Simplicity is a fundamental asset of the D-LMS algorithms proposed. Small step-sizes may be required to attain sufficiently small steady-state estimation errors, hence leading to reduction in the speed of convergence. This motivated the D-RLS algorithms of Chapter 4, which are specially well suited for WSN applications in which fast convergence rates are at a premium, yet increased computational burdens per sensor can be afforded.

In Chapter 4, distributed RLS-like algorithms are developed for adaptive estimation/tracking using WSNs in which sensors communicate via single-hop noisy links. The approach adopted involves: (i) reformulating in a separable way the exponentially weighed least-squares cost involved in the classical RLS algorithm; and (ii) applying either the AD-MoM or the AMA schemes to minimize this separable cost in a distributed fashion. The resulting algorithms entail only local computational tasks across sensors that simply exchange messages with single-hop neighbors only. In particular, it was shown that the AMA-based D-RLS algorithm incurs reduced computational complexity per sensor when compared to its AD-MoM-based counterpart.

In order to accommodate nonstationary applications, STD-RLS is derived from D-RLS and turns out to afford single consensus recursion per time instant t. Different from existing approaches, (ST)D-RLS exhibits robustness to inter-sensor communication noise at the expense of a slightly higher communication complexity. When noise is less of an issue, (ST)D-RLS can be modified to lower communication overhead relative to adaptive diffusion estimation schemes. Numerical examples demonstrate the noise robustness of (ST)D-RLS, as well as its convergence and tracking capabilities in comparison with existing alternatives.

5.2 Future research

The results in this thesis motivate a number of future research topics. Next, we outline two of them that we are currently pursuing. The first one is a direct extension to the performance analysis carried out for D-RLS in Chapter 4. The second on pertains to distributed (possibly adaptive) estimation of *sparse* signals via ℓ_1-norm penalized optimization methods.

5.2.1 D-RLS Performance Analysis Extensions

In Chapter 4, a steady-state MSE performance analysis was carried out for the D-RLS algorithm when $\lambda < 1$. For the infinite memory case in which $\lambda = 1$, numerical simulations indicate that D-RLS provides mean-square sense-consistent estimates, even in the presence of communication noise. This is a well-known result in classical (centralized) least-squares estimation, and establishes a further connection between the classical RLS and the distributed counterparts proposed in this thesis. By formally establishing this property, D-RLS becomes an even more appealing alternative for distributed parameter estimation in stationary environments. While the approximations used in Chapter 4 are no longer valid when $\lambda = 1$, for Gaussian independent and identically distributed regressors, the matrix $\mathbf{\Phi}^{-1}(t)$ is Wishart distributed with known moments. Under these assumptions, consistency analysis is a subject of ongoing investigation.

To carry out the analysis in Chapter 4, the parameter sought was assumed to be time-invariant. By relying on the approximation $\mathbf{\Phi}^{-1}(t) \approx (1-\lambda)\mathbf{R}_h^{-1}$ to construct an 'averaged' system closely related to D-RLS, it is possible to mimic the techniques in Chapter 3 to deal with the slowly time-varying AR(1) parameter model

$$\begin{cases} \mathbf{s}_0(t) & = & \mathbf{s}_0 + \breve{\mathbf{s}}(t) \\ \breve{\mathbf{s}}(t) & = & \mathbf{\Theta}\breve{\mathbf{s}}(t-1) + \boldsymbol{\zeta}(t) \end{cases}.$$

Quantifying the steady-state MSE tracking performance of D-RLS enables a fair comparison with D-LMS when it comes to nonstationary operation, and provides an objective means of selecting the most appropriate algorithm for operation in dynamic ad hoc WSNs.

5.2.2 Distributed Lasso for Estimation and Tracking of Sparse Signals

In statistics, the least absolute shrinkage and selection operator (lasso) [58]

$$\hat{\mathbf{s}}_{\text{lasso}} = \arg \min_{\mathbf{s} \in \mathbb{R}^p} \sum_{j=1}^{J} \|\mathbf{x}_j - \mathbf{H}_j \mathbf{s}\|_2^2 + \lambda \|\mathbf{s}\|_1 \tag{5.1}$$

has well-documented merits over the workhorse LS estimates in the context of *sparse* linear regression. By augmenting the LS cost with the scaled ℓ_1-norm of the unknown vector $\mathbf{s}$, the lasso estimate $\hat{\mathbf{s}}_{\text{lasso}}$ can improve prediction accuracy when it is a priori known that many entries of $\mathbf{s}$ are zero. It is also attractive in terms of interpretation – especially for the case in which p is very large – since only a few nonzero entries are retained in $\hat{\mathbf{s}}_{\text{lasso}}$ which correspond to the strongest predictors. The amount of shrinkage is controlled by the parameter λ. In the signal processing literature, the lasso is referred to as *basis pursuit* [13].

Many efficient algorithms have been developed to solve (5.1) for all values of $\lambda > 0$, when all the regression data $\{\mathbf{x}_j, \mathbf{H}_j\}$ is centrally available; see, e.g., [14, 22, 58]. However, collecting all data in a central location may be impossible in many applications of interest. In the context of statistical linear regression, this limitation may arise due to, e.g., privacy reasons as in medical records of patients in a network of health institutions. In distributed estimation of sparse signals, data is inherently collected by distributed wireless sensors or cognitive radios [5], and transmissions to a central processing unit encounter the challenges described in Chapter 1. All in all, distributed algorithms that can solve (5.1) based on single-hop exchanges of intermediate sparse estimates are well motivated but so far relatively unexplored. The AD-MoM-based framework utilized throughout this dissertation can be applied to this end, and resource-efficient algorithms result that are provably convergent to $\hat{\mathbf{s}}_{\text{lasso}}$. Similar to the D-RLS algorithm in Chapter 4, it is also of interest to address the problem of distributed tracking of time-varying sparse signals. Here, the least-squares cost in (5.1) is replaced by its EWLS counterpart, a problem which has already been considered in a centralized setting [1].

Bibliography

[1] D. Angelosante and G. B. Giannakis, "RLS-weighted Lasso for adaptive estimation of sparse signals," in *Proc. of the Intl. Conf. on Acoustics, Speech and Signal Proc.*, Taipei, Taiwan, pp. 3245 – 3248, Apr. 2009.

[2] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation.* Wiley-Interscience, 2001.

[3] S. Barbarossa, G. Scutari, and T. Battisti, "Cooperative sensing for cognitive radio using decentralized projection algorithms," in *Proc. of the Wrkshp. on Signal Proc. Adv. in Wireless Communications*, Perugia, Italy, pp. 116 – 120, June 2009.

[4] S. Barbarossa, G. Scutari, and T. Battisti, "Distributed signal subspace projection algorithms with maximum convergence rate for sensor networks with topological constraints," in *Proc. of the Intl. Conf. on Acoustics, Speech and Signal Proc.*, Taipei, Taiwan, pp. 2893 – 2896, Apr. 2009.

[5] J.-A. Bazerque and G. B. Giannakis, "Distributed spectrum sensing for cognitive radios by exploiting sparsity," in *Proc. of 42nd Asilomar Conf. On Signals, Systems and Computers*, Pacific Grove, CA, pp. 1588–1592, Oct. 2008.

[6] D. P. Bertsekas, *Nonlinear Programming.* Athena-Scientific, second ed., 1999.

[7] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods.* Athena-Scientific, second ed., 1999.

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

[9] F. S. Cattivelli, C. G. Lopes, and A. H. Sayed, "Diffusion strategies for distributed Kalman filtering: Formulation and performance analysis," in *Proc. of the IAPR Workshop on Cognitive Information Processing*, Santorini, Greece, June 2008.

[10] F. S. Cattivelli and A. H. Sayed, “Diffusion LMS algorithms with information exchange,” in *Proc. of 42nd Asilomar Conf. On Signals, Systems and Computers*, Pacific Grove, CA, pp. 251–255, Nov. 2008.

[11] F. S. Cattivelli and A. H. Sayed, “Diffusion mechanisms for fixed-point distributed Kalman smoothing,” in *Proc. of EUSIPCO*, Lausanne, Switzerland, Aug. 2008.

[12] F. S. Cattivelli and A. H. Sayed, “Multi-level diffusion adaptive networks,” in *Proc. of the Intl. Conf. on Acoustics, Speech and Signal Proc.*, Taipei, Taiwan, pp. 2789–2792, Apr. 2009.

[13] S. S. Chen, D. L. Donoho, and M. A. Saunders, “Atomic decomposition by basis pursuit,” *SIAM Journal on Scientific Computing*, vol. 20, pp. 33–61, 1998.

[14] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, “Least angle regression,” *Annals of Statistics*, vol. 2, pp. 407–499, 2004.

[15] P. A. Forero, A. Cano, and G. B. Giannakis, “Consensus-based distributed expectation-maximization algorithm for density estimation and classification using wireless sensor networks,” in *Proc. of the Intl. Conf. on Acoustics, Speech and Signal Proc.*, Las Vegas, NV, pp. 1989–1992, Mar./Apr. 2008.

[16] G. F. Franklin, J. D. Powell, and A. Emami-Naeini, *Feedback Control of Dynamic Systems*. Addison-Wesley, second ed., 1991.

[17] D. Gabay and B. Mercier, “A dual algorithm for the solution of nonlinear variational problems via finite-element approximations,” *Comp. Math. Appl.*, vol. 2, pp. 17–40, 1976.

[18] R. Glowinski and A. Marrocco, “Sur l‘approximation par elements finis d‘orde un, et la resolution par penalisation-dualite d‘une classe de problemes de Dirichlet nonlineaires,” *Rev. Francaise d‘Aut. Inf. Rech. Oper.*, vol. 2, pp. 41–76, 1975.

[19] G. H. Golub and C. F. V. Loan, *Matrix Computations*. Johns Hopkins, third ed., 1996.

[20] P. Gupta and P. R. Kumar, “The capacity of wireless networks,” *IEEE Trans. on Information Theory*, vol. 46, pp. 388–404, Mar. 2000.

[21] Y. Hatano, A. K. Das, and M. Mesbahi, “Agreement in presence of noise: pseudogradients on random geometric networks,” in *Proc. of the 44th Conf. on Dec. and Contr.*, Seville, Spain, pp. 6382–6387, Dec. 2005.

[22] H. H. J. Friedman, T. Hastie and R. Tibshirani, “Pathwise coordinate optimization,” *Annals of Applied Statistics*, vol. 1, pp. 302–332, 2007.

[23] A. Jadbabaie and S. Morse, “Coordination of groups of mobile autonomous agents using nearest neighbor rules,” *IEEE Trans. on Automatic Control*, vol. 48, pp. 988–1001, 2003.

[24] O. Jahromi and P. Aarabi, “Distributed spectrum estimation in sensor networks,” in *Proc. of Intl. Conf. on Acoust. Speech and Sig. Proc.*, Montreal, Canada, pp. 849–852, May 2004.

[25] S. Kar and J. M. F. Moura, “Distributed consensus algorithms in sensor networks with imperfect communication: link failures and channel noise,” *IEEE Trans. on Signal Processing*, vol. 57, pp. 355–369, Jan. 2009.

[26] H. J. Kushner and G. G. Yin, *Stochastic Approximation and Recursive Algorithms and Applications*. Springer-Verlag, second ed., 2003.

[27] C. G. Lopes and A. H. Sayed, “Incremental adaptive strategies over distributed networks,” *IEEE Trans. on Signal Processing*, vol. 55, pp. 4064–4077, Aug. 2007.

[28] C. G. Lopes and A. H. Sayed, “Diffusion least-mean squares over adaptive networks: Formulation and performance analysis,” *IEEE Trans. on Signal Processing*, vol. 56, pp. 3122–3136, July 2008.

[29] G. Mateos, I. D. Schizas, and G. B. Giannakis, “Performance analysis of the consensus-based distributed LMS algorithm,” *EURASIP Journal on Advances in Signal Processing.* May 2009 (submitted).

[30] G. Mateos, I. D. Schizas, and G. B. Giannakis, “Consensus-based distributed least-mean square algorithm using wireless sensor networks,” in *Proc. of 45th Allerton Conference*, Monticello, IL, Sept. 2007.

[31] G. Mateos, I. D. Schizas, and G. B. Giannakis, “Closed-from MSE performance of the distributed LMS algorithm,” in *Proc. of 13th Digital Signal Proc. Workshop*, Marco Island, FL, Jan. 2009.

[32] G. Mateos, I. D. Schizas, and G. B. Giannakis, “Distributed recursive least-squares for consensus-based in-network adaptive estimation,” *IEEE Trans. on Signal Processing*, vol. 57, Sept. 2009.

[33] T. Minka, “Old and new matrix algebra useful for statistics,” Dec. 2000.

[34] A. Nedic and D. P. Bertsekas, "Incremental subgradient methods for nondifferentiable optimization," *SIAM Jounral on Optimization*, vol. 12, p. 109138, Jan. 2001.

[35] R. Olfati-Saber, "Distributed Kalman filtering for sensor networks," in *Proc. of the 46th Conf. on Dec. and Contr.*, New Orleans, LA, pp. 5492–5498, Dec. 2007.

[36] 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.

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

[38] R. Olfati-Saber and J. S. Shamma, "Consensus filters for sensor networks and distributed sensor fusion," in *Proc. of the 44th Conf. on Dec., and the Eur. Contr. Conf.*, Seville, Spain, pp. 6698–6703, Dec. 2005.

[39] C. H. Papadimitriou, *Computational Complexity.* Addison-Wesley, 1993.

[40] M. G. Rabbat and R. D. Nowak, "Quantized incremental algorithms for distributed optimization," *IEEE Journal on Sel. Areas In Comm.*, vol. 23, pp. 798–808, 2005.

[41] M. G. Rabbat, R. D. Nowak, and J. A. Bucklew, "Generalized consensus computation in networked systems with erasure links," in *Proc. of the Wrkshp. on Signal Proc. Adv. in Wireless Communications*, New York, NY, pp. 1088– 1092, June 2005.

[42] S. S. Ram, A. Nedic, and V. V. Veeravalli, "Stochastic incremental gradient descent for estimation in sensor networks," in *Proc. of 41st Asilomar Conf. on Signals, Systems, and Computers*, Pacific Grove, CA, pp. 582–586, 2007.

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

[44] A. H. Sayed and C. G. Lopes, "Distributed recursive least-squares over adaptive networks," in *Proc. of 40th Asilomar Conf. On Signals, Systems and Computers*, Pacific Grove, CA, pp. 233–237, Oct./Nov. 2006.

[45] I. D. Schizas, G. B. Giannakis, A. Ribeiro, and S. I. Roumeliotis, "Consensus in ad hoc WSNs with noisy links - Part II: Distributed estimation and smoothing of random signals," *IEEE Trans. on Signal Processing*, vol. 56, pp. 1650–1666, Apr. 2008.

[46] I. D. Schizas, G. Mateos, and G. B. Giannakis, “Consensus-based distributed recursive least-squares estimation using ad hoc wireless sensor networks,” in *Proc. of 41st Asilomar Conf. On Signals, Systems and Computers*, Pacific Grove, CA, pp. 386–390, Nov. 2007.

[47] I. D. Schizas, G. Mateos, and G. B. Giannakis, “Stability analysis of the consensus-based distributed LMS algorithm,” in *Proc. of the Intl. Conf. on Acoustics, Speech and Signal Proc.*, Las Vegas, NV, pp. 3289–3292, Mar./Apr. 2008.

[48] I. D. Schizas, G. Mateos, and G. B. Giannakis, “Distributed LMS for consesus-based in-network adaptive processing,” *IEEE Trans. on Signal Processing*, vol. 57, pp. 2365–2381, June 2009.

[49] I. D. Schizas, A. Ribeiro, and G. B. Giannakis, “Consensus in ad hoc WSNs with noisy links - Part I: Distributed estimation of deterministic signals,” *IEEE Trans. on Signal Processing*, vol. 56, pp. 350–364, Jan. 2008.

[50] V. Solo, “On the stability of slowly time-varying linear systems,” *Math. Control. Signals*, vol. 7, pp. 331–350, 1994.

[51] V. Solo, “The stability of LMS,” *IEEE Trans. on Signal Processing*, vol. 45, pp. 3017–3026, Dec. 1997.

[52] V. Solo and X. Kong, *Adaptive Signal Processing Algorithms: Stability and Performance.* Prentice Hall, 1995.

[53] D. P. Spanos, R. Olfati-Saber, and R. M. Murray, “Dynamic consensus on mobile networks,” in *Proc. of the 16th IFAC World Congress*, Prague, Czech, July 2005.

[54] S. S. Stankovic, M. S. Stankovic, and D. S. Stipanovic, “Decentralized parameter estimation by consensus based stochastic approximation,” in *Proc. of the 46th Conf. on Dec. and Contr.*, New Orleans, LA, pp. 1535–1540, Dec. 2007.

[55] P. Stoica and R. Moses, *Spectral Analysis of Signals.* Prentice Hall, 2005.

[56] B. Sundararaman, U. Buy, and A. D. Kshemkalyani, “Clock synchronization for wireless sensor networks: a survey,” *Ad Hoc Networks*, vol. 3, pp. 281–323, May 2005.

[57] N. Takahashi, I. Yamada, and A. H. Sayed, “Diffusion least-mean squares with adaptive combiners,” in *Proc. of the Intl. Conf. on Acoustics, Speech and Signal Proc.*, Taipei, Taiwan, pp. 2845–2848, Apr. 2009.

[58] R. Tibshirani, “Regression shrinkage and selection via the lasso,” *J. Royal. Statist. Soc. B.*, vol. 58, pp. 267–288, 1996.

[59] P. Tseng, “Applications of a splitting algorithm to decomposition in convex programming and variational inequalities,” *SIAM Journal on Control and Optimization*, vol. 29, pp. 119–138, Jan. 1991.

[60] B. Widrow and S. D. Stearns, *Adaptive Signal Processing.* Prentice Hall, 1985.

[61] J. Wu and H. Li, “A dominating-set-based routing scheme in ad hoc wireless networks,” *Telecommunication Systems Journal*, vol. 3, pp. 63–84, Sept. 2001.

[62] L. Xiao and S. Boyd, “Fast linear iterations for distributed averaging,” *Systems and Control Letters*, vol. 53, pp. 65–78, Sept. 2004.

[63] L. Xiao, S. Boyd, and S. Lall, “A space-time diffusion scheme for peer-to-peer least-squares estimation,” in *Proc. of Intl. Conf. on Info. Proc. in Sensor Networks*, Nashville, TN, pp. 168–176, 2006.

[64] H. Zhu, A. Cano, and G. B. Giannakis, “Distributed demodulation using consensus averaging in wireless sensor networks,” in *Proc. of 42nd Asilomar Conf. On Signals, Systems and Computers*, Pacific Grove, CA, pp. 1170–1174, Oct. 2008.

[65] H. Zhu, I. D. Schizas, and G. B. Giannakis, “Power-efficient dimensionality reduction for distributed channel-aware Kalman tracking using WSNs,” *IEEE Trans. on Signal Processing*, vol. 57, pp. 3193–3207, Aug. 2009.