Diffusion LMS Strategies for Distributed Adaptation in Sensor Networks

Reza Abdolee

Department of Electrical & Computer Engineering
McGill University
Montreal, Canada

July 2014

A thesis submitted to McGill University in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Abstract

Diffusion least-mean squares (DLMS) algorithms can serve as efficient and powerful learning mechanisms for solving distributed estimation and optimization problems over networks in response to streaming data originating from different locations in real-time. Owing to their decentralized processing structures, simplicity of implementation and adaptive learning capabilities, they are particularly well-suited for applications to multi-agent networks where the network energy and radio resources are limited. In this dissertation, we propose new DLMS algorithms with application to sensor networks and examine their performance under realistic environmental conditions. Within this framework, the contributions of the dissertation can be divided into three main parts.

In the first part of the dissertation, we propose novel DLMS strategies to estimate parameters of a physical phenomenon that vary over spatial domain. In sensor networks, there are various applications that involve physical phenomena featuring space-varying parameters in their mathematical models, e.g., localization of distributed sources in dynamic systems or the modeling of diffusion phenomena in inhomogeneous media. One notable complication that arises in the context of space-varying parameters estimation is that the covariance matrices of the regression data over the network can become rank-deficient. This condition influences the learning behavior of the network and causes the estimates to become biased. In our analysis, we elaborate on how the judicious use of network combination matrices can help alleviate this difficulty.

In the second part, we examine the performance of DLMS algorithms when the input regression data at each node over the network are noisy. Under this condition, we show that the estimates obtained by DLMS strategies will be biased. We address this issue by relying on a bias-elimination technique and formulating an optimization problem that utilizes the noise variance information of the regression data. By solving this optimization problem, we arrive at novel DLMS algorithms called bias-compensated DLMS strategies that are capable of obtaining unbiased estimates of the unknown parameters over the network. We also derive a recursive adaptive approach by which each node, besides the standard adaptation layer to solve the desired distributed estimation, runs a second layer estimation to locally find their regression noise variances over time.

Within the last part of the dissertation, we analyze the performance of DLMS strategies over wireless sensor networks under fading conditions. Wireless channel impairments, including path loss and fading, can distort the exchanged data between nodes and, subsequently, degrade the performance of DLMS algorithms and cause instability. To resolve this issue, we propose an extended version of these algorithms that incorporate equalization coefficients in their combination update to reverse the effects of fading and path loss. We also analyze the impact of channel estimation error on the performance of the proposed algorithms and obtain conditions that guarantee the network stability in the mean and mean-square error sense. To further improve the performance of the proposed algorithms, we formulate a convex optimization problem from an upper-bound approximation of the network mean-square deviation (MSD) in order to find the optimal combination weights of the network that lead to lower estimation errors.

Throughout the thesis, all the new algorithms have been fully evaluated under controlled and realistic simulation environment and demonstrated superior performance when compared to benchmark algorithms and approaches from the existing literature.

Sommaire

Les algorithmes de diffusion à erreur quadratique moyenne minimale (DLMS) peuvent servir de mécanismes d'apprentissage efficaces et puissants pour la solution en temps réel de problèmes destimation et d'optimisation distribuée dans les réseaux de capteurs, en réponse aux données en continu émanant d'une pluralité d'endroits. En raison de leur structure de traitement décentralisée, leur facilité de mise en œuvre et leur capacité d'apprentissage adaptif, ils sont particulièrement bien adaptés pour les applications réseaux dans lesquelles de multiples agents échangent des informations par liaisons sans fil, et où l'énergie et les ressources radio des agents sont limitées. Dans cette thèse, nous présentons de nouveaux algorithmes DLMS qui s'appliquent aux réseaux de capteurs et examinons leurs efficacités dans des conditions d'utilisation pratiques avec variabilité temporelle de l'environnement radio. Nos contributions se divisent en trois parties.

Dans la première partie, nous proposons de nouvelles stratégies DLMS pour l'estimation des paramètres d'un phénomène physique, lorsque ceux-ci varient dans le domaine spatial. Dans les réseaux de capteurs, il existe plusieurs applications dans lesquelles les phénomènes physiques en cause font intervenir des paramètres variant dans leur modélisation, comme par exemple la localisation des sources distribuées dans les systèmes dynamiques, ou la modélisation des phénomènes de diffusion dans les médias inhomogènes. Une complication notable se pose toutefois dans le cadre de l'estimation de tels paramètres: en effet, les matrices de covariance de données de régression sur le réseau peuvent présenter une déficience de rang. Cette condition influe sur le comportement d'apprentissage et fait en sorte que les estimateurs peuvent devenir biaisés. Dans notre analyse, nous nous intéresserons à l'utilisation judicieuse des matrices de combinaison de réseau afin de palier à cette difficulté.

Dans la deuxième partie de cette dissertation, nous examinons la performance des algorithmes DLMS lorsque les données de régression á chacun des nœuds du réseau sont bruitées. Nous démontrons qu'en présence de telles erreurs, les estimateurs obtenues par stratégies DLMS seront biaisés et donc peu fiables. Afin d'aborder cette question, nous proposons une technique d'élimination de biais et formulons un problème d'optimisation qui utilise les informations sur la variance du bruit des données de régression de tous les agents. Par la solution de ce problème d'optimisation, nous arrivons á de nouveaux algorithmes DLMS, appelés algorithmes DLMS á compensation de biais, qui permettent d'obtenir une estimation non biaisée des paramètres inconnus dans le réseau. Notre analyse montre que sous certaines conditions, les algorithmes proposés sont stables au sens de la moyenne et de l'erreur quadratique moyenne sens, et que les estimateurs convergent asymptotiquement vers leurs vraies valeurs.

Dans la dernière partie, nous analysons la performance des stratégies DLMS dans les réseaux de capteurs sans fil sous des conditions d'évanouissement. Les imperfections du canal, telles l'affaiblissement de trajet et les évanouissements, peuvent fausser les données échangées entre les nœuds et, conséquemment, dégrader la performance des algorithmes DLMS et causer leur instabilité. Afin de pallier aux imperfections du canal sans fil et ainsi résoudre ce problème, nous proposons une extension des algorithmes DLMS qui intègre des coefficients d'égalisation dans le processus de mise á jour par diffusion. Nous analysons l'impact des erreurs d'estimation sur la performance des algorithmes proposés et obtenons des conditions qui garantissent la stabilité du réseau au sens de la moyenne et de l'erreur quadratique moyenne. Afin d'améliorer la performance des nouveaux algorithmes, nous formulons un problème d'optimisation convexe, á partir d'une approximation par borne supérieure sur la déviation quadratique moyenne (MSD), dans le but de déterminer les matrices de combinaison de réseau optimales, permettant ainsi de minimiser les erreurs d'estimation des nouveaux algorithmes.

Acknowledgments

My deepest gratitude goes to my adviser, Prof. Benoit Champagne, for his continuous support and inspiration throughout my Ph.D. Program. Without his guidance and constructive advice, this thesis would never be created. I would also like to express my sincere appreciation to Prof. Ali H. Sayed for his supervisory cooperation during and after my scholarly visit at University of California, Los Angeles (UCLA). His invaluable advice, immense knowledge and insightful comments greatly helped to improve the technical quality of my work. I am very thankful to Dr. Stephan Saur and Dr. Stephan Ten brink for their support and technical advice during my internship at Bell Labs, Alcatel Lucent, Germany. I also would like to thank the member of my Ph.D. committee, Prof. Ioannis Psaromiligkos and Prof. Bruce Shepherd, for their guidance and valuable feedbacks.

I have thoroughly enjoyed spending time with all fellow Lab-mates at Telecommunication and Signal Processing Laboratory. Here, I wish to give special thanks to: Ali, Arash, Chao, Fang, Jiaxin, Golnaz, Saeed, Siamak and Siavash not only to share the daily ups and downs of the research life but also for their willingness to help. My great appreciation goes to friends from Adaptive System Lab. at UCLA for their hospitality during my visit. I have always enjoyed having stimulating technical discussions with Zaid Towfic, Jianshu Chen, Sheng-Yuan Tu, Zhao Xiaochuan, Shang Kee Ting and Chung-Kai Yu.

I am also forever indebted to my parents for their unconditional love and support. They have given me a sense of freedom and encouraged me to make my own decisions throughout my life. Last but not the least, I would like to thank my loving wife, Vida Vakilian, for her patience, continuous support and encouragement over the course of my doctoral studies.

Here, I acknowledge that this research would not have been possible without: a) a Postgraduate Scholarships-Doctoral Program (PGSD) award from Natural Sciences and Engineering Research Council (NSERC) of Canada, b) financial assistance from Prof. Benoit Champagne via Fonds Québécois de la Recherche sur la Nature et les Technologies (FQRNT), c) a McGill International Doctoral Award (MIDA), a Graduate Research Mobility Award (GRMA), and Graduate Funding Travel Awards (GREAT) from the Department of Electrical and Computer Engineering, McGill University, and d) a Research Internship in Science and Engineering (RISE) award from the German Academic Exchange Service (DAAD). I would like to express my gratitude to all those individuals and agencies for their support.

Preface & Contributions of the Authors

The research presented in this dissertation was carried out at the Department of Electrical and Computer Engineering, McGill University from Sept. 2009 to April 2014. This dissertation is the result of my original work and created under the supervision of Prof. Benoit Champagne from the same department. The research has led to several journal and conference publications for which, as the first author, I proposed the ideas, develop the algorithms, and carried out the analysis and numerical experiments. Prof. Ali H. Sayed from University of California, Los Angeles (UCLA) and Prof. Benoit Champagne have reviewed the articles and provided feedbacks to improve the quality and technical presentation of the works. The related contributions are listed below:

- Journal Papers

  1. R. Abdolee, B. Champagne and A. H. Sayed, “Estimation of space-time varying parameters using a diffusion LMS algorithm”, *IEEE Trans. on Signal Processing*, vol 62, no. 2, pp. 403–418, Jan. 2014.
  2. R. Abdolee, B. Champagne, “Diffusion LMS strategies in sensor networks with noisy input data applications”, submitted to *IEEE/ACM Trans. on Networking.*
  3. R. Abdolee, B. Champagne and A. H. Sayed, “Diffusion adaptation over multi-agent networks with wireless link impairments”, submitted to *IEEE Trans. on Mobile Computing.*

- Conference Papers

  1. R. Abdolee, B. Champagne and A. H. Sayed, “Diffusion LMS localization and tracking algorithm for wireless cellular network”, in *Proc. IEEE International Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, Vancouver, Canada, May 2013, pp. 4598–4602.
  2. R. Abdolee, B. Champagne and A. H. Sayed, “Diffusion LMS strategies for parameter estimation over fading wireless channels”, in *Proc. of IEEE Int. Conf. on Communication (ICC)*, Budapest, Hungary, June 2013, pp. 1926–1930.
  3. R. Abdolee, B. Champagne and A. H. Sayed, “Diffusion LMS for source and process estimation in sensor Networks”, in *Proc. of IEEE Statistical Signal Processing (SSP) Workshop*, Ann Arbor, MI, Aug. 2012. pp. 165–168.

4. R. Abdolee, B. Champagne and A. H. Sayed, “A diffusion LMS Strategy for parameter estimation in noisy regressor applications”, in *Proc. of the 20th European Signal Processing Conf. (EUSIPCO)*, Bucharest, Romania, Aug. 2012, pp. 749–753.

5. R. Abdolee and B. Champagne, “Diffusion LMS algorithms for sensor networks over non-ideal inter-sensor wireless channel”, in *Proc. of IEEE Int. Conf. on Dist. Computer Sensor Sys. (DCOSS)*, Barcelona, Spain, June 2011, pp. 1–6.

6. R. Abdolee and B. Champagne, “Distributed blind adaptive algorithms based on constant modulus for wireless sensor networks”, in *Proc. of 6th Int. Conf. on Wireless and Mobile Communications (ICWMC)*, Valencia, Spain, Sept. 2010, pp. 303–308.

Contents

List of Figures

List of Acronyms

DAA	Distributed Adaptive Algorithms
MSE	Mean-Square Error
MMSE	Minimum Mean-Square Error
MSD	Mean-Square Deviation
EMSE	Excess Mean-Square Error
LMS	Least-Mean Square
RLS	Recursive Least Square
TLS	Total Least Square
RTLS	Recursive Total Least Square
DLMS	Distributed Least-Mean Square
AF	Amplify-and-Forward
WSN	Wireless Sensor Networks
CSI	Channel State Information
QoS	Quality-of-Service
PDE	Partial Differential Equation
RF	Radio Frequency
RHS	Right Hand Side
SDP	Semidefinite Programming
SNR	Signal-to-Noise Ratio
ZF	Zero-Forcing

Chapter 1

Introduction

A wireless sensor network (WSN) consists of an array of sensor nodes distributed over a geographical area to cooperatively monitor physical phenomena through noisy observation processes. The nodes, which we also interchangeably call agents, consist of at least three main components : processing units, sensing devices and a wireless transceiver. In more advance sensor networks, nodes are also equipped with actuators to take action according to the command issued from a control unit. Initially, WSN were developed for military applications, including localization and battlefield surveillance. Nowadays they are becoming ubiquitous in several areas in industrial monitoring and consumer applications, including intelligent transportation, health care, precision agriculture, and smart spaces [1, 2].

By design, inexpensive sensor nodes are assumed to have limited computational capabilities, low battery power, and a low-cost radio transceiver. These constraints entail a number of practical challenges in the realization of sensor networks: a) design of computationally simple algorithms that consume low amount of energy and require a small number of data transmissions to perform a processing task, b) scalability of the algorithms must be considered when the size of the network is intrinsically large due to the nature of an application, c) robustness against frequent node failures, changes in link connectivity status and variation in the network topology, d) efficient data aggregation techniques to cope with traffic congestion in the event when the network is flooded with large number of messages from nodes, and e) robust communication mechanisms to deal with the increased possibility of packet collisions and congestions for nodes operating in closely spaced transmission ranges.

In this dissertation, we address some of these challenges that fall within the context of distributed adaptive signal processing and parameter estimation in sensor networks. In what follows, we first provide an introduction to the two well-known signal processing approaches for parameter estimation and optimization in sensor networks and discuss their positions with respect to the above issues. We then review the prior studies in the field, and state the detail objectives and contributions of the dissertation. Finally, we present the organization of the dissertation and introduce notations used in the mathematical development.

1.1 Distributed Algorithms

From data traffic point of view, there are two different approaches that can be implemented to perform a signal processing task over sensor networks, namely, centralized and distributed techniques. In the former, nodes send their measurements to a central unit known as a fusion center for further processing and storage, whereas in the latter, the measured data are locally exchanged and processed within the network. In a centralized approach, transmitting the measured data to the fusion center may cause network congestion and results in a waste of communication resources and power. In this approach, any malfunction in the fusion center can lead to a drastic network breakdown. In addition, the fusion center requires relatively high computation power to process the large volume of collected data. In comparison, in a distributed signal processing approach, the network computational load is divided between nodes and no centralized infrastructure is required. In addition, since in distributed approaches, data are exchanged locally (e.g., through single or multi-hop hop data transmission technique), a communication bottleneck may not be created over the network [3]. The single or multi-hop data transmission also reduces the network energy consumption because the power loss of wireless transmissions increases super-linearly with respect to the propagation distance. These advantages encourage the use of distributed signal processing approaches for various applications in sensor networks.

Over the past few years, there has been extensive research on distributed signal processing, as it holds the promise of overcoming the issues of bandwidth scarcity and limited energy budget in dense sensor networks. Within this framework, in-network distributed adaptive signal processing is emerging as a key enabling technology to support the implementation of flexible cooperative learning and information processing schemes across a set of geographically distributed nodes, with sensing, computing and communications capabilities. Distributed *adaptive* algorithms (DAA) are particularly useful for the solution of optimization problems and parameter estimation over networks, where the underlying signal statistics are unknown or time-varying [4]. Clearly, adaptivity helps the network to track variations of the desired signal parameters as new measurements become available. More importantly, as a result of distributed adaptive processing, a sensor network becomes robust against changes in the environment conditions and network topology. In this thesis, we study and develop distributed adaptive strategies, of diffusion type, for monitoring time-varying physical phenomena in sensor networks under real-world constraints and changes in environmental conditions.

1.2 Related Works

DAA can be classified as the stochastic version of decentralized optimization that are well-investigated in mathematics and computer science [5]. These algorithms are useful for solving estimation problems in real-time over multi-agent networks where the measurement data are random and keep streaming over time. In communication and signal processing fields, the topic of distributed adaptation is a growing research area and a promising technique to handle network scalability and sensor limited energy issues. Recently, DAA have been used to model several instances of organized and complex behavior encountered in nature, such as bird flight formations [6], fish schooling [7], bee swarming, and bacteria motility [8]. In addition, studies have revealed that these algorithms are useful in solving various optimization problems in communication, including, data detection and estimation, localization, and link rate control [9–14].

There are several classes of DAA algorithms for optimization and parameter estimation over networks, including incremental methods [5, 15–17], consensus methods [18–23, 23–26], and diffusion methods [27–33]. Initially, the incremental methods have been proposed to solve distributed least squares problems [5]. Later, a general class of these algorithms has been investigated for in-network processing to reduce the required energy and bandwidth by the nodes over networks [16]. Recently, incremental adaptive algorithms based on least mean square (LMS) and recursive least squares (RLS) techniques have been reported for real-time parameter estimation and tracking over networks [17, 34, 35]. Although the incremental adaptive algorithms function well for low-energy profile networks, they require setting a cyclic path over the nodes, which is generally an NP-hard problem; these techniques are also sensitive to link failure.

Diffusion adaptive strategies have been introduced in [27,36,37] to reduce the risk of network break-down due to link failure and to improve the scalability of the algorithms in large size networks. In diffusion strategies, nodes communicate with their immediate neighbors within a single hop distance, which relaxes the requirement of setting a Hamiltonian loop over the network. Consensus techniques disseminate the data through the network similar to diffusion strategies. However, they require doubly-stochastic combination matrices and, when used in the context of adaptation with constant step-sizes, can lead to unstable behavior at the network level even if all individual nodes are able to solve the inference task in a stable manner [30]. In this dissertation, we will focus on diffusion strategies because they have been shown to be more robust and lead to a stable behavior regardless of the underlying topology, even when some of the underlying nodes are unstable [30]. Moreover, diffusion strategies have a stabilizing effect and endow networks with real-time adaptation and learning abilities.

In previous studies, diffusion strategies were concerned with estimating a parameter vector that is assumed to be invariant over the spatial domain. For this type of applications, it is assumed that the nodes sense physical phenomena featuring parameters that are identical over the network. There are, however, various important applications in sensor networks where the underlying system parameters vary over space, such as in monitoring fluid flow in underground porous media [38], tracking population dispersal in ecology [39], and in the modeling of diffusion phenomena in inhomogeneous media [40]. In these applications, the space-variant parameters being estimated correspond to spatially discretized versions of the space-dependent coefficients of partial differential equations (PDEs) describing the phenomena of interest. The estimation of spatially-varying parameters in PDEs has been addressed in many previous studies, including [41–43]. However, in these works and similar references on the topic, the solutions typically rely on the use of a central processing unit and less attention is paid to distributed solutions over a network of interconnected processing nodes. Within the first part of this thesis, we consider this problem and propose diffusion LMS (DLMS) for the estimation of space-variant parameter vectors in an adaptive and distributed manner.

The second issue that arises in parameter estimation applications over sensor networks is that the system input data, called regression data in this context, are contaminated with measurements noise. Under this condition, if a distributed estimation algorithm is implemented to estimate the underlying system parameter vector without considering the effect of the measurement noise, the estimate will be biased and unreliable. Bias removal and compensation have been extensively investigated for the stand-alone LMS filter in earlier studies [44–50]. For networking applications, a consensus-based total least squares (TLS) algorithm based on convex semidefinite programming has been developed that requires costly eigendecompostion process at each node in every iteration [51]. A distributed recursive total least squares (RTLS) algorithm has also been developed which relies on the bias compensation principle and requires knowledge about regressor noise variances over the network [52]. For this problem, the RLS class of distributed adaptive algorithms may seem attractive because of their distributed structure and fast convergence speed in networking applications. However, the high computational complexity, numerical instability, and slower tracking ability may be hinderance for distributed dynamic systems and low-energy budget network applications.

In this dissertation, motivated by low computational complexity, and satisfactory tracking ability of the LMS, we propose bias-compensated DLMS algorithms that exploits the spatial diversity of the data and attains an unbiased optimal estimate of the underlying system parameters over the network. It is worth noting that the problem we considered here is different from the one investigated in [53], where the authors studied DLMS algorithms for distributed parameter estimation over network with imperfect information exchange. This is because, they assumed that at each node only the received regression data from neighbors are corrupted with communication noise, and the regression data of the node itself are error-free. They have also provided no mechanism for removing the bias from the estimate. We, in this thesis, assume that all regression data are noisy and propose a solution to remove the bias from the network estimates.

Diffusion strategies have been developed based on the existence of ideal communication channels between sensors, meaning each node over the network will receive error-free information from all its neighbors [27–30, 33]. This assumption, however, may not hold in practice where communications between nodes are prone to communication noise and fading. Several works have examined the effect of noisy communication links on the performance of these strategies [53–58]. In these works, it was assumed that the links between nodes are always active, regardless of the instantaneous link noise value, and hence the network topology remains invariant over time (i.e., the network topology will be static).

This assumption will be also restrictive in many existing and emerging applications in wireless communication systems and sensor networks. For example, in mobile networks where agents are allowed to change their position over time, the signal-to-noise ratio (SNR) of the communication links between nodes will vary due to the various channel impairments, including path loss, multi-path fading and shadowing. Consequently, the set of nodes with which each agent can communicate (called neighborhood set), which is a function of the link SNR, will also change as the agents move, and the network topology is therefore intrinsically dynamic. A time-varying network topology can also be created as a consequence of energy drain in nodes, leading to a sudden link-failure, or due to deployment or activation of substitute nodes over the network, i.e., creation of new links. It is therefore essential to investigate the performance of diffusion strategies over networks with time-varying (dynamic) topology and characterize the effects of link activity (especially link failure) and fading on their convergence and stability. This is the third issue that will be investigated in this dissertation.

The performance of DLMS strategies is significantly affected by the network combination matrices, which are used to combine the exchanged information between nodes [59]. There are several well-known combination rules in the literature, especially in the context of consensus-based iterations such as the maximum-degree rule and the Metropolis rule [60–62]. While these schemes focus on improving the convergence behavior of the algorithms, they ignore the variations in noise profile and link status across the network, which can result in substantial performance degradation [63]. Some earlier works consider the variation in measurement noise profile over the network to obtain the optimal combination weights [33]. In their development, they have relied on the formulation and solution of a nonlinear and non-convex optimization problem that is pursued numerically. In [64], the authors have formulated this problem as a convex optimization and incorporated the measurement noise profile into the design of the combination weights. Authors in [53] have studied a more general scenario, where in addition to measurement noise, they also consider communication noise, in the exchange of information between nodes, in their problem formation. In particular, they have formulated an optimization problem that take into the account the node energy profile and the communication noise to obtain the optimal combination weights over the network. In the design of optimal combination weights in the aforementioned works, it was assumed that the links between nodes are always active, regardless of the nodes mobility, communication noise and the fading coefficients values.

In this dissertation, we consider a more general case to design the optimal combination weights. In particular, in our problem formulation, we consider a wireless sensor network, where the links are affected by fading and path loss in addition to communication noise, and as a consequence the link connectivity status and the network topology vary over time.

1.3 Objectives and Contributions

The main objective of this research is the development and investigation of new DLMS algorithms for the monitoring of physical phenomena over sensor networks under practical constraints. We consider limiting aspects or constraints, including, variations of physical parameters over space, distortion of data with measurement noise as well as communication constraints such as fading, path loss and link noise. Ultimately, at the end of this research, we will be able to answer questions regarding the feasibility of implementation of DLMS strategies over practical sensor networks, and comment on the convergence behavior and steady-state performance of the newly proposed algorithms, where both the network topology and wireless channels vary over time. The research work presented in this dissertation consists of three main parts, as detailed below.

In the first part, we propose novel DLMS strategies to enable the estimation and tracking of parameters that may vary over both *space* and *time*. Our approach starts by introducing a linear regression model to characterize space-time varying phenomena over networks. This model is derived by discretizing a representative second-order partial PDE, which can be useful in characterizing many dynamic systems with spatially-varying properties. We then introduce a set of basis functions, e.g., shifted Chebyshev polynomials, to represent the space-varying parameters of the underlying phenomena in terms of a finite set of space-invariant expansion coefficients. Building on this representation, we develop a diffusion LMS strategy that cooperatively estimates, interpolates, and tracks the model parameters over the network. We analyze the convergence and stability of the developed algorithm, and derive closed-form expressions to characterize the learning and convergence behavior of the nodes in the mean-square-error (MSE) sense. In our analysis, we find that in the context of space-time varying models, the covariance matrices of the regression data at the various nodes can become rank deficient. This property influences the learning behavior of the network and causes the estimates to become biased. We elaborate on how the judicious use of stochastic combination matrices can help alleviate this difficulty. The analysis of

DLMS algorithms in *space-invariant* parameter scenarios with rank-deficient covariance matrices is treated as a special case in our analysis.

In the second part of the dissertation, we examine the ability of DLMS strategies over sensor networks to correct for the measurement noise through cooperation. In particular, we first show that the parameter estimates produced by conventional DLMS strategies are biased when the regression data are contaminated with additive noise. We then formulate bias-compensated DLMS strategies that exploit both the spatial diversity of the data and the regressor noise variance information to attain an unbiased estimate via an adaptive diffusion process. The development of the proposed algorithms rely on a bias-elimination technique that assumes prior knowledge about the regression noise variances over the network. We then relax the known variance assumption by incorporating a recursive approach into the algorithm to estimate the variances in real-time. The analysis results show that if the step-sizes are within a given range, the algorithms will be stable in the mean and mean-square sense and the estimated parameters will converge to their true values.

The third part of this dissertation has two contributions. The first contribution is to extend the application of DLMS strategies to wireless sensor networks, where nodes exchange information over fading channels. To this end, we first examine the performance of DLMS algorithms under the effects of wireless channel impairments, including, fading and path-loss in addition to link noise. To counter the effect of channel impairments, we incorporate the equalization coefficients into the diffusion update of DLMS algorithms. Our analysis shows that if each node knows the channel state information (CSI) of its neighboring nodes, the effects of fading and path-loss can be mitigated. We observe that as a consequence of fading, despite using equalization coefficients, the links connectivity status may change due to variation in instantaneous signal-to-noise ratio (SNR), and therefore, the network topology will also vary over time. In networks with dynamic topology, the static combination rules [27, 33] fail to provide satisfactory results and may cause instability. To resolve this issue, we introduce a combination strategy that initializes the network weighting matrices using well-known combination rules and update their values according to link connectivity over time. In the modified algorithms, it is assumed that the nodes do not have access to CSI of their neighbors and thus equalization coefficients are computed from pilot-assisted estimated channel coefficients. This, in turn, introduces estimation error that adversely affects the performance of the algorithms. We comment on the performance limit of the DLMS algorithms with estimated channel coefficients and show how the mean and mean-square stability of the network are affected by channel estimation error and link communication noise. Our analysis reveals that under certain conditions, the modified DLMS algorithms are asymptotically unbiased and converge in the mean and mean-square error sense providing that each node over the network knows CSI of its neighbors. Moreover, we find that when the CSI are estimated through pilot-assisted techniques, the algorithms still converge in the mean and mean-square error sense. while the parameter estimates become biased in this case, we show that if the wireless channels are slowly-varying such that more pilot data can be used for the estimation of fading coefficients, the bias can made sufficiently small.

The second contribution in this part is to find the optimal combination weights that minimize the network MSD. To find the optimal combination weights, we introduce an approximation for the upper-bound of the network MSD and then use it to formulate a convex optimization problem that can be solved in closed-form. The obtained solution provides the desired performance improvement. Nevertheless, it requires knowledge of second order statistics of the network data (e.g., input signal correlation matrix, measurement noise variance and communication noise variances), which may not be available in practice. To overcome this limitation, we introduce an adaptive recursive relation to seek the optimal combination weights by relying on instantaneous approximations of the second order statistics of the network data. In this way, besides the standard adaptation layer to solve the desired distributed estimation problem, each node also runs a second adaptation layer to adjust its combination weights in real-time. Since the proposed adaptive scheme does not require second order signal statistics of the nodes data, it will be particularly useful in sensor network applications with time-varying wireless channel and changing network topologies, where such information are normally unavailable.

1.4 Thesis Organization and Notations

The dissertation is organized as follows:

In Chapter 2, we review two well-known classes of distributed adaptation strategies, namely, consensus and diffusion, and show how these strategies can be derived as special cases of the same formalism. We then carry out a mean-square performance analysis in a unified manner to compare their convergence and stability performance.

In Chapter 3, we introduce a space-varying linear regression model which is motivated from a physical phenomenon characterized by a PDE, and formulate an optimization problem to find the unknown parameters of the introduced model. We then derive a DLMS algorithm that solves this problem in a distributed and adaptive manner. Lastly, we analyze the performance of the proposed algorithm, and present the results of computer experiments.

We begin Chapter 4 by explaining the effects of input measurement noise on the performance of DLMS strategies in parameter estimation over networks, where it is shown that the estimates of DLMS estimates are biased under this condition. Next, we develop the proposed bias-compensated DLMS algorithms by relying on a bias-elimination technique, and present a recursive estimation strategy that locally estimates the regressor noise variances of the nodes. We then analyze the stability and convergence behavior of the developed algorithms, and present the results of our computer experiments to support the analytical findings.

In Chapter 5, we develop the received signal model for the data exchange over wireless sensor networks and explain how links between nodes fail as a consequence of deep fading or low SNR values. Next, we review the standard DLMS algorithms and introduce an extension of DLMS strategies for distributed estimation over wireless networks. The extended method uses equalization coefficients to alleviate the adverse effects of the channel impairments. To obtain the fading coefficients, we use a pilot-assisted LS channel estimation technique. We then show how to compute the weighting matrix when the network topology changes over time to avoid network instability. We analyze the converge behavior of the proposed algorithms, and find conditions under which the network is stable in the mean and mean-square error sense.

In Chapter 6, we briefly review the MSE analysis of DLMS strategies over wireless networks. We then study the effects of combination weights on the performance of DLMS strategies in this type of networks. To find the optimal values of the left-stochastic combination matrix over time, we formulate a convex optimization problem using an upper bound of network MSD. We finally present the simulation of computer experiments to support the theoretical findings.

Finally, in Chapter 7, we conclude the thesis and suggest some future research directions in the field of DAA.

Notations: The following notations are used throughout this dissertation. Matrices are represented by upper-case and vectors by lower-case letters. Boldface fonts are reserved for random variables and normal fonts are used for deterministic quantities. Superscript $(\cdot)^T$ denotes transposition for real-valued vectors and matrices while $(\cdot)^*$ denotes conjugate transposition for complex-valued vectors and matrices. The symbol $\mathbb{E}[\cdot]$ is the expectation operator, $\mathrm{Tr}(\cdot)$ represents the trace of its matrix argument and $\mathrm{diag}\{\cdot\}$ extracts the diagonal entries of a matrix, or constructs a diagonal matrix from a vector. I_M represents the identity matrix of order M (subscript M is omitted when the order can be understood from the context). A set of vectors are stacked into a column vector by $\mathrm{col}\{\cdot\}$. The $\mathrm{vec}(\cdot)$ operator vectorizes a matrix by stacking its columns on top of each other and $\mathrm{bvec}(\cdot)$ is the block-vectorization operator [27]. The operator $\mathrm{rank}(\cdot)$ computes the rank of its matrix argument. The $\mathrm{bvec}(\cdot)$ operator vectorizes a block matrix by first vectorizing each of its blocks and then stacking the resulting vectors on top of each other into a column. The symbol $\det(\cdot)$ denotes the determinant operator. The symbol $\otimes$ denotes the standard Kronecker product, and the symbol $\otimes_b$ represents the *block Kronecker product* [65].

Chapter 2

Background Study

In this chapter, we review two classes of distributed adaptive strategies, namely, consensus and diffusion algorithms and provide a systematic way for their derivations and analysis. The presented material in this chapter serves as a basis for understanding the context and development of subsequent chapters. The flow of the presentation is as follows. We first demonstrate how consensus and diffusion algorithms can be derived by following a similar procedure in minimization of a mean-square error function. We then briefly review the mean and mean-square convergence analysis of these algorithms in a unified manner and compare their performance and stability. At the end of this chapter, we give reasons as to why the focus of this dissertation is mainly on diffusion (DLMS) strategies.

2.1 Introduction

The consensus strategy has been developed to enforce agreement among cooperating agents that are distributed over a spatial domain [66]. In recent years, average consensus strategies, including gossip algorithms, have been studied in many areas, such as control systems [23,62,67–70], formation and schooling over multi-agent networks [71–73], distributed optimization [18,25], and also distributed estimation [20,21,24,26,57,74–76]. Initially, the development of consensus strategy relied on the use of two time-scales process [19,77]: one time-scale for data collection across the agents and another time-scale for averaging process over the collected data. The two time-scale implementations hinder the ability to perform real-time recursive estimation and adaptation when measurement data keep streaming in. In this work, we study the consensus implementations that operate in a single time-scale and is therefore suitable for real-time data processing. Such implementations appear in several recent works, including [24, 26, 57, 78], and are largely motivated by the procedure developed earlier in [18, 79] for the solution of distributed optimization problems.

DLMS strategies, the second class of distributed adaptive algorithms, were introduced for the solution of estimation problems in multi-agent networks [4, 9, 27, 33, 35, 37, 80, 81]. These algorithms were developed as single time-scale distributed adaptive schemes that can respond in real-time to the continuous streaming of data at the agents [82]. Since their introduction, diffusion strategies were applied to model various forms of complex behaviors encountered in nature [6,8,83,84]. They were also adopted to solve distributed optimization problems in [29, 85, 86] and their performance have been studied under varied conditions in [28, 87, 88] as well. Diffusion strategies inherently rely on a single time-scale processing structure and are therefore naturally amenable to real-time and recursive implementations.

2.2 Network Signal Model

Consider a network consisting of N nodes that are distributed over a spatial domain. Node ℓ is said to be a neighbor of node k if it can communicate with node k. We denote the set of all such neighbors, i.e., the neighborhood of node k, by $\mathcal{N}_k$. The objective of the nodes in the network is to estimate an unknown parameter vector $w^o \in \mathbb{C}^{M\times 1}$ in a distributed manner through an on-line learning process. At every time instant, i, each node k observes a scalar random process $\boldsymbol{d}_k(i)$ and a vector random process $\boldsymbol{u}_{k,i} \in \mathbb{C}^{1\times M}$, which are related to w^o via the following linear regression model [89]:

$$\boldsymbol{d}_k(i) = \boldsymbol{u}_{k,i} w^o + \boldsymbol{v}_k(i) \tag{2.1}$$

where $\boldsymbol{v}_k(i)$ is the measurement noise.

Assumption 2.1. *The random variables in model (2.1) are assumed to satisfy the following conditions*[1]*:*

a) The regression data $\{\boldsymbol{u}_{k,i}\}$ are zero-mean, i.i.d. in time and independent over space with covariance matrices $R_{u,k} = \mathbb{E}[\boldsymbol{u}_{k,i}^* \boldsymbol{u}_{k,i}] > 0$.

[1] These are the standard assumptions that normally are used in distributed adaptive literature for mathematical tractability. We also adopt these assumptions in the subsequent chapters in this dissertation.

b) The noise $\{\boldsymbol{v}_k(i)\}$ are zero-mean, i.i.d. in time and independent over space with variances $\sigma_{v,k}^2$.

c) The regression data $\boldsymbol{u}_{k,i}$ and the noise $\boldsymbol{v}_m(n)$ are mutually independent for all k, m, i and n.

■

The models of the form (2.1) are useful in capturing many situations of interest in practice, such as in estimating the parameters of some underlying physical phenomenon, estimating or equalizing a communications channel, tracking a moving target by a collection of nodes, or estimating the location of a nutrient source or predator in biological networks [6, 84, 89, 90].

The nodes estimate w^o by seeking to minimize the following global cost function:

$$J^{\text{glob}}(w) = \sum_{k=1}^{N} \mathbb{E}|\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}w|^2 \tag{2.2}$$

In what follows, we show that how the global objective function (2.2) can be written in terms of N local objective functions whose minimization in parallel over the network are equivalent to minimizing (2.2).

2.3 Distributed Estimation

In this section, we review the derivation of the most frequent special case of DLMS algorithms, i.e., the adapt-then-combine (ATC) diffusion, in multi-agent adaptive networks. The derivation is based on the completion-of-squares argument, followed by a stochastic approximation step and an incremental approximation step [29, 33]. For comparison purposes, we also explain how the single time-scale consensus strategy can be obtained using the same procedure.

To begin with, we express the global cost (2.2) as:

$$J^{\text{glob}}(w) = \sum_{k=1}^{N} J_k(w) \tag{2.3}$$

where

$$J_k(w) = \mathbb{E}|\boldsymbol{d}_k(i) + \boldsymbol{u}_{k,i}w|^2 \tag{2.4}$$

The derivation of ATC diffusion strategies, to optimize $J^{\text{glob}}(w)$ in a distributed manner, is based on two steps optimization procedure [29, 33]. First, using a completion-of-squares argument, we approximate the global cost function (2.4) by an alternative cost that is amenable to distributed optimization. Then, each node will optimize the alternative cost via a combination of a steepest-descents and an incremental approximation step.

We note that each individual cost $J_k(w)$ given by (2.4) can be factored via a completion-of-squares argument and written in the form [33]:

$$J_k(w) = \|w - w^o\|^2_{R_{u,k}} + \text{mmse}_k \tag{2.5}$$

where the notation $\|x\|^2_\Sigma$ denotes the weighted square quantity $x^*\Sigma x$ for any semidefinite matrix $\Sigma \geq 0$, $R_{u,k} = \mathbb{E}[\boldsymbol{u}^*_{k,i}\boldsymbol{u}_{k,i}]$ is the covariance matrix of the regression data at node k, and mmse_k is an additional MMSE term that is independent of w. Using (2.5), we can replace the cost function (2.3) by the following equivalent global cost function that is amenable to a distributed minimization form:

$$J^{\text{glob}'}(w) = J_k(w) + \sum_{\ell \neq k} \left(\|w - w^o\|^2_{R_{u,\ell}} + \text{mmse}_\ell \right) \tag{2.6}$$

The second term on the right-hand side of (2.6), implies that how by incorporating the quadratic parts, the individual cost $J_k(w)$ can be corrected to the global cost $J^{\text{glob}}(w)$. However, the minimizer w^o that appears in the quadratic parts is not known since the nodes wish to determine its value. Likewise, not all weighting matrices $R_{u,\ell}$ are available to node k; only those from its neighbors can be assumed to be available. In spite of this missing information, expression (2.6) motivates us to introduce a new localized cost function at node k that is close enough to the desired $J^{\text{glob}}(w)$ and which can be minimized through local cooperation. We denote this localized cost function at node k by $J_k^{\text{dist}}(w)$; it is obtained from (2.6) by limiting the summation on the right-hand side of (2.6) to the neighbors of node k, namely,

$$J_k^{\text{dist}}(w) = J_k(w) + \sum_{\ell \in \mathcal{N}_k \backslash \{k\}} \|w - w^o\|^2_{R_{u,\ell}} \tag{2.7}$$

We note that the terms mmse_ℓ are ignored in obtaining (2.7), since they are independent of w and have no effects in finding the minimizer w^o. The cost functions $J_k(w)$ and $J_k^{\text{dist}}(w)$ are both associated with node k. The difference between them is that the expression for the latter is closer to the global cost function (2.6) that we want to optimize. The covariance matrices $R_{u,\ell}$ that appear in (2.7) may not be available in practice. Usually, nodes can only observe realizations $\boldsymbol{u}_{\ell,i}$ of regression data arising from distributions whose covariance matrices are the unknown $R_{u,\ell}$. One way to address this issue is to replace each of the weighted norms $\|w - w^o\|^2_{R_{u,\ell}}$ by a scaled multiple of the form

$$\|w - w^o\|^2_{R_{u,\ell}} \approx b_{\ell,k}\|w - w^o\|^2 \tag{2.8}$$

where $b_{\ell,k}$ is some nonnegative coefficient. This substitution amounts to having each node k approximates the moment $R_{u,\ell}$ from its neighbors by multiples of the identity matrix, i.e.,

$$R_{u,\ell} = b_{\ell,k}\ I_M \tag{2.9}$$

Approximation (2.8) is reasonable because using the Rayleigh-Ritz characterization of eigenvalues [91], it holds that

$$\lambda_{\min}(R_{u,\ell})\,\|w - w^o\|^2 \leq \|w - w^o\|^2_{R_{u,\ell}} \leq \lambda_{\max}(R_{u,\ell})\,\|w - w^o\|^2 \tag{2.10}$$

As the derivation will show, the scalars $b_{\ell,k}$ in (2.8) will end up being embedded into another set of coefficients $a_{\ell,k}$ that will be selected by the designer. Later, we will explain how to select these new coefficients to achieve the desired objective. In this way, we replace (2.7) by:

$$J_k^{\text{dist}'}(w) = J_k(w) + \sum_{\ell \in \mathcal{N}_k \backslash \{k\}} b_{\ell,k}\|w - w^o\|^2 \tag{2.11}$$

With the exception of the minimizer w^o, this alternative cost at node k relies solely on information that is available from its neighborhood. Now, each node k can apply a steepest-descent iteration to minimize its localized cost $J_k^{\text{dist}'}(w)$, i.e.,

$$\begin{aligned} \boldsymbol{w}_{k,i} &= \boldsymbol{w}_{k,i-1} - \mu_k[\nabla_w J_k^{\text{dist}'}(w)]^* && (2.12)\\ &= \boldsymbol{w}_{k,i-1} + \mu_k(r_{du,k} - R_{u,k}\boldsymbol{w}_{k,i-1}) - \mu_k \sum_{\ell \in \mathcal{N}_k \backslash \{k\}} b_{\ell,k}(\boldsymbol{w}_{k,i-1} - w^o) && (2.13) \end{aligned}$$

where ∇_w denotes the gradient vector of its argument. The step-size parameters μ_k can be constant or time-variant. Constant step-sizes allow the resulting strategies to learn and adapt continuously, while time-variant step-sizes that decay to zero turn off the learning abilities of the networks with time. An adaptive implementation of (2.13) can be obtained by replacing $\{r_{du,k}, R_{u,k}\}$ by instantaneous approximations:

$$r_{du,k} \approx \boldsymbol{d}_k(i)\boldsymbol{u}^*_{k,i}, \qquad R_{u,k} \approx \boldsymbol{u}^*_{k,i}\boldsymbol{u}_{k,i} \tag{2.14}$$

Doing so leads to the following recursion:

$$\boldsymbol{w}_{k,i} = \boldsymbol{w}_{k,i-1} + \mu_k \boldsymbol{u}^*_{k,i}(\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}\boldsymbol{w}_{k,i-1}) - \mu_k \sum_{\ell \in \mathcal{N}_k \backslash \{k\}} b_{\ell,k}(\boldsymbol{w}_{k,i-1} - w^o) \tag{2.15}$$

According to (2.15), the update from $\boldsymbol{w}_{k,i-1}$ to $\boldsymbol{w}_{k,i}$ now involves adding two correction terms to $\boldsymbol{w}_{k,i-1}$. However, the last correction term still depends on the unknown minimizer w^o. We can now use incremental-type arguments to replace w^o in (2.15) by suitable approximations for it. As it will be shown different replacements for w^o lead to different learning strategies (such as consensus and diffusion strategies) and these replacements will affect the operation of the network in a fundamental way [30].

2.3.1 Consensus Strategy

We note that each of the nodes in the network performs steps similar to (2.15). As such, each node ℓ will have a readily available approximation for w^o, i.e., the local estimate $\boldsymbol{w}_{\ell,i-1}$. Therefore, the first possible substitution for w^o in (2.15) is $\boldsymbol{w}_{\ell,i-1}$. In that case, recursion (2.15) becomes

$$\boldsymbol{w}_{k,i} = \boldsymbol{w}_{k,i-1} - \mu_k \sum_{\ell \in \mathcal{N}_k \backslash \{k\}} b_{\ell,k}(\boldsymbol{w}_{k,i-1} - \boldsymbol{w}_{\ell,i-1}) + \mu_k \boldsymbol{u}^*_{k,i}(\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}\boldsymbol{w}_{k,i-1}) \tag{2.16}$$

Recursion (2.16) is in the form of the well-known consensus strategy [24, 26].

It should be noted that in most other works on consensus implementations, especially in the context of distributed optimization problems [24, 26, 79, 92], the step-sizes μ_k that are used in (2.16) depend on the time-index i and are required to satisfy

$$\sum_{i=1}^{\infty} \mu_k(i) = \infty, \qquad \sum_{i=1}^{\infty} \mu_k^2(i) < \infty \tag{2.17}$$

In other words, for each node k, the step-size sequence $\mu_k(i)$ is required to vanish as $i \to \infty$. Under such conditions, the consensus strategies allow the nodes to reach agreement. Here, instead, in the representations (2.16), we consider constant step-sizes because we are interested to examine the adaptation and tracking abilities of the adaptive networks. We can rewrite the recursion (2.16) in a more compact form by combining the first two terms on the right-hand side of (2.16) and introducing:

$$a_{\ell,k} = \begin{cases} 1 - \sum_{j \in \mathcal{N}_k \backslash \{k\}} \mu_k b_{j,k}, & \ell = k \\ \mu_k b_{\ell,k}, & \ell \in \mathcal{N}_k \backslash \{k\} \\ 0, & \text{otherwise} \end{cases} \tag{2.18}$$

In this way, recursion (2.16) can be rewritten equivalently as Algorithm 2.1, presented below [78, 79].

Algorithm 2.1 : Consensus Strategy

$$\boldsymbol{w}_{k,i} = \sum_{\ell \in \mathcal{N}_k} a_{\ell,k} \boldsymbol{w}_{\ell,i-1} + \mu_k \boldsymbol{u}_{k,i}^* (\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i} \boldsymbol{w}_{k,i-1}) \tag{2.19}$$

The coefficient $a_{\ell,k}$ denotes the weight that node k assigns to the estimate $\boldsymbol{w}_{\ell,i-1}$ received from its neighbor ℓ. From (2.18), it is clear that the weights $a_{\ell,k}$ are nonnegative for $\ell \neq k$ and that $a_{k,k}$ is nonnegative for sufficiently small step-sizes. If we collect the assumed nonnegative weights $a_{\ell,k}$ into an $N \times N$ matrix $A = [a_{\ell,k}]$, which we call the network combination matrix, then it follows from (2.18) that this matrix satisfies the following properties:

$$a_{\ell,k} \geq 0, \qquad A^T \mathbb{1}_N = \mathbb{1}_N \qquad \text{and} \quad a_{\ell,k} = 0 \ \text{ if } \ \ell \notin \mathcal{N}_k \tag{2.20}$$

where $\mathbb{1}_N$ is a vector of size N with all entries equal to one. That is, the weights on the links arriving at node k add up to one, which is equivalent to saying that the matrix A is left-stochastic [93]. Moreover, if node ℓ is not a neighbor of node k, then the corresponding weight $a_{\ell,k}$ is zero.

2.3.2 Diffusion Strategies

The second possible substitution for w^o in (2.15) will lead to two forms of DLMS strategies, namely, adapt-then-combine (ATC) and combine-then-adapt (CTA) diffusion strategies. We begin by noting that there are two correction terms on the right-hand side of (2.15) and these terms can be added one at a time. To derive the ATC diffusion, we rewrite (2.15) in a mathematical form that consists of two steps and use an intermediate auxiliary variable to connect these steps, i.e.:

$$\boldsymbol{\psi}_{k,i} = \boldsymbol{w}_{k,i-1} + \mu_k \boldsymbol{u}^*_{k,i}(\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}\boldsymbol{w}_{k,i-1}) \tag{2.21}$$

$$\boldsymbol{w}_{k,i} = \boldsymbol{\psi}_{k,i} - \mu_k \sum_{\ell \in \mathcal{N}_k \setminus \{k\}} b_{\ell,k}(\boldsymbol{w}_{k,i-1} - w^o) \tag{2.22}$$

The first update (2.21) can be carried out by all nodes independent of the knowledge of w^o. However, the unknown minimizer w^o still appears in (2.22). Now, rather than replacing w^o by $\boldsymbol{w}_{\ell,i-1}$, as was the case with the consensus strategy, it would appear to be more advantageous to replace w^o by the improved estimate $\boldsymbol{\psi}_{\ell,i}$ obtained via the update (2.22). Indeed, for each node ℓ, the intermediate value $\boldsymbol{\psi}_{\ell,i}$ is generally a better estimate for w^o than $\boldsymbol{w}_{\ell,i-1}$ since it is obtained by incorporating information from its recent data $\{\boldsymbol{d}_\ell(i), \boldsymbol{u}_{\ell,i}\}$ as per (2.21). In the same vein, we can also replace $\boldsymbol{w}_{k,i-1}$ in (2.22) by $\boldsymbol{\psi}_{k,i}$. This second substitution is similar to incremental-type approaches of optimization, which have been widely studied in the literature [5]. With these replacements, recursion (2.22) becomes

$$\boldsymbol{w}_{k,i} = \boldsymbol{\psi}_{k,i} - \mu_k \sum_{\ell \in \mathcal{N}_k \setminus \{k\}} b_{\ell,k}(\boldsymbol{\psi}_{k,i} - \boldsymbol{\psi}_{\ell,i}) \tag{2.23}$$

If we again introduce the same coefficients $a_{\ell,k}$ from (2.18), we arrive at the following alternative compact form, known as the adapt-then-combine (ATC) diffusion strategy [33], presented below as Algorithm 2.2.

In Algorithm 2.2, the first step (2.24) involves local adaptation, where node k uses its own data $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$ to update its weight estimate from $\boldsymbol{w}_{k,i-1}$ to an intermediate value $\boldsymbol{\psi}_{k,i}$. The second step (2.25) is a combination step where the intermediate estimates $\boldsymbol{\psi}_{\ell,i}$ from the neighborhood of node k are combined through the weights $a_{\ell,k}$ to obtain the updated weight estimate $\boldsymbol{w}_{k,i}$.

By reversing the order of (2.21)-(2.22), and following a similar procedure as in the ATC

Algorithm 2.2 : ATC Diffusion Algorithm

$$\boldsymbol{\psi}_{k,i} = \boldsymbol{w}_{k,i-1} + \mu_k \boldsymbol{u}^*_{k,i}(\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}\boldsymbol{w}_{k,i-1}) \tag{2.24}$$

$$\boldsymbol{w}_{k,i} = \sum_{\ell \in \mathcal{N}_k} a_{\ell,k}\boldsymbol{\psi}_{\ell,i-1} \tag{2.25}$$

diffusion [33], we arrive at the other variant of DLMS algorithms, known as CTA diffusion strategy, presented below as Algorithm 2.3.

Algorithm 2.3 : CTA Diffusion Algorithm

$$\boldsymbol{\psi}_{k,i-1} = \sum_{\ell \in \mathcal{N}_k} a_{\ell,k}\boldsymbol{w}_{\ell,i-1} \tag{2.26}$$

$$\boldsymbol{w}_{k,i} = \boldsymbol{\psi}_{k,i-1} + \mu_k \boldsymbol{u}^*_{k,i}(\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}\psi_{k,i-1}) \tag{2.27}$$

In this algorithm, the first step is a combination step, by which the existing estimates $\{\boldsymbol{w}_{\ell,i-1}\}$ from the neighbors of node k are combined through the weights $\{a_{\ell,k}\}$. The second step (2.27) is a local adaptation step, where node k uses its own data $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$ to update its weight estimate from the intermediate value $\boldsymbol{\psi}_{k,i-1}$ to $\boldsymbol{w}_{k,i}$. Thus, comparing the ATC and CTA strategies, we note that the order of the combination and adaptation steps are reversed.

More general diffusion strategies can be derived if each agent k, in addition to the estimators $\boldsymbol{\psi}_{\ell,i}$ in ATC or $\boldsymbol{w}_{\ell,i-1}$ in CTA, receives the data $\{\boldsymbol{d}_\ell(i), \boldsymbol{u}_{\ell,i}\}$ from its neighbors $\ell \in \mathcal{N}_k \backslash k$. These generalizations, in addition to the matrix A, require a second combination matrix C with nonnegative entries which is right-stochastic. The generalized ATC strategy is presented in Algorithm 2.4 for comparison.
where $c_{\ell,k}$ are the entries of the right-stochastic matrix C, satisfying:

$$c_{\ell,k} \geq 0, \qquad C\mathbb{1}_N = \mathbb{1}_N \qquad \text{and} \quad c_{\ell,k} = 0 \;\text{ if } \;\; \ell \notin \mathcal{N}_k \tag{2.30}$$

We now note that the ATC diffusion Algorithm 2.2 can be obtained from Algorithm 2.4 by choosing $C = I$.

Algorithm 2.4 : Generalized ATC Diffusion Algorithm

$$\boldsymbol{\psi}_{k,i} = \boldsymbol{w}_{k,i-1} + \mu_k \sum_{\ell \in \mathcal{N}_k} c_{\ell,k} \boldsymbol{u}^*_{\ell,i}(\boldsymbol{d}_\ell(i) - \boldsymbol{u}_{\ell,i}\boldsymbol{w}_{k,i-1}) \tag{2.28}$$

$$\boldsymbol{w}_{k,i} = \sum_{\ell \in \mathcal{N}_k} a_{\ell,k} \boldsymbol{\psi}_{\ell,i-1} \tag{2.29}$$

2.3.3 Diffusion versus Consensus

In this part and in the following section, we only compare the ATC variant of DLMS strategies, Algorithm 2.2, with the consensus strategy, Algorithm 2.1, to highlight the main differences in operation and learning behavior[1].

Considering the actual implementation of ATC diffusion, i.e., Algorithm 2.2, we first note that the combination step (2.25) is able to incorporate additional information into their processing steps without being more complex than the consensus strategy. Second, it can be also seen that these two strategies require sharing the same amount of data, as can be ascertained by comparing the actual implementations. The key fact to note is that the diffusion implementations first generate an intermediate state, which is subsequently used in the final update. As it will be shown, this ordering of the calculations has a critical influence on the performance of the algorithms.

For comparison purposes and performance evaluation of these algorithms , it is of interest to rewrite the ATC diffusion and consensus strategies in a single update as given below:

$$\text{Diffusion,} \quad \boldsymbol{w}_{k,i} = \sum_{\ell \in \mathcal{N}_k} a_{\ell,k} \boldsymbol{w}_{\ell,i-1} + \sum_{\ell \in \mathcal{N}_k} \mu_\ell a_{\ell,k} \boldsymbol{u}^*_{\ell,i}(\boldsymbol{d}_\ell(i) - \boldsymbol{u}_{\ell,i}\boldsymbol{w}_{\ell,i-1}) \tag{2.31}$$

$$\text{Consensus,} \quad \boldsymbol{w}_{k,i} = \sum_{\ell \in \mathcal{N}_k} a_{\ell,k} \boldsymbol{w}_{\ell,i-1} + \mu_k \boldsymbol{u}^*_{k,i}(\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}\boldsymbol{w}_{k,i-1}) \tag{2.32}$$

Note that the first terms on the right hand side of these recursions are the same. For the second terms, only variable $\boldsymbol{w}_{k,i-1}$ appears in the consensus strategy (2.32), while the diffusion strategies incorporate the estimates $\boldsymbol{w}_{\ell,i-1}$ from the neighborhood of node k into

[1] For the sake of fairness, we use the special version of ATC with $C = I$. The ATC with a non-diagonal right stochastic matrix C outperforms the one with $C = I$ [33].

the update of $\boldsymbol{w}_{k,i}$. Moreover, in contrast to the consensus, the ATC diffusion strategy further incorporates the influence of the data $\{\boldsymbol{d}_\ell(i), \boldsymbol{u}_{\ell,i}\}$ from the neighborhood of node k into the update of $\boldsymbol{w}_{k,i}$. These facts have important implications on the evolution of the weight-error vectors in the consensus and diffusion networks.

2.4 Performance Analysis

The mean and mean-square performance of DLMS algorithms have been extensively studied in detail in [29,33,82]. In this section, we briefly review the performance analysis of diffusion algorithms and make comparison with that of the consensus algorithm to highlight their differences.

We define the local weight-error vectors as

$$\tilde{\boldsymbol{w}}_{k,i} = w^o - \boldsymbol{w}_{k,i} \tag{2.33}$$

and form the global weight-error vector by staking up the local error vectors on top of each other, as expressed by

$$\tilde{\boldsymbol{w}}_i = \text{col}\{\tilde{\boldsymbol{w}}_{1,i}, \tilde{\boldsymbol{w}}_{2,i}, \dots, \tilde{\boldsymbol{w}}_{N,i}\} \tag{2.34}$$

We also define the block vector and matrices:

$$\boldsymbol{g}_i \triangleq \text{col}\Big\{\boldsymbol{u}_{1,i}^*\boldsymbol{v}_1(i), \boldsymbol{u}_{2,i}^*\boldsymbol{v}_2(i), \cdots, \boldsymbol{u}_{N,i}^*\boldsymbol{v}_N(i)\Big\} \tag{2.35}$$

$$\boldsymbol{\mathcal{R}}_i \triangleq \text{diag}\Big\{\boldsymbol{u}_{1,i}^*\boldsymbol{u}_{1,i}, \boldsymbol{u}_{2,i}^*\boldsymbol{u}_{2,i}, \cdots, \boldsymbol{u}_{N,i}^*\boldsymbol{u}_{N,i}\Big\} \tag{2.36}$$

$$\mathcal{M} \triangleq \text{diag}\Big\{\mu_1 I_M, \mu_2 I_M, \cdots, \mu_N I_M\Big\} \tag{2.37}$$

Starting from (2.32), and (2.31) and using model (2.1), the global error vector $\boldsymbol{w}_i$ for the diffusion and consensus strategies can be found to evolve in the following way:

$$\text{Diffusion,} \quad \tilde{\boldsymbol{w}}_i = \mathcal{A}^T(I - \mathcal{M}\boldsymbol{\mathcal{R}}_i)\tilde{\boldsymbol{w}}_{i-1} - \mathcal{A}^T\mathcal{M}\boldsymbol{g}_i \tag{2.38}$$

$$\text{Consensus,} \quad \tilde{\boldsymbol{w}}_i = (\mathcal{A}^T - \mathcal{M}\boldsymbol{\mathcal{R}}_i)\tilde{\boldsymbol{w}}_{i-1} - \mathcal{M}\boldsymbol{g}_i \tag{2.39}$$

where $\mathcal{A} = A \otimes I_M$. Recursions (2.39) and (2.38) can be described by a more general recursion of the form:

$$\tilde{\boldsymbol{w}}_i = \boldsymbol{\mathcal{B}}_i \tilde{\boldsymbol{w}}_{i-1} - \mathcal{M}\boldsymbol{y}_i \tag{2.40}$$

where the quantities $\boldsymbol{\mathcal{B}}_i$ and $\boldsymbol{y}_i$ for the diffusing strategy are

$$\boldsymbol{\mathcal{B}}_i = \mathcal{A}^T(I - \mathcal{M}\boldsymbol{\mathcal{R}}_i) \tag{2.41}$$

$$\boldsymbol{y}_i = \mathcal{A}^T\mathcal{M}\boldsymbol{g}_i \tag{2.42}$$

whereas for the consensus strategy, these quantities take the form

$$\boldsymbol{\mathcal{B}}_i = \mathcal{A}^T - \mathcal{M}\boldsymbol{\mathcal{R}}_i \tag{2.43}$$

$$\boldsymbol{y}_i = \mathcal{M}\boldsymbol{g}_i \tag{2.44}$$

The matrix $\boldsymbol{\mathcal{B}}_i$ controls the evolution dynamics of the network error vector $\tilde{\boldsymbol{w}}_i$ as per (2.38). As it will be shown the differences in $\boldsymbol{\mathcal{B}}_i$ in (2.41) and (2.43) substantially, impact the performance of diffusion and consensus strategies.

2.4.1 Mean Convergence and Stability

Below, we review the mean convergence and stability analysis of diffusion and consensus strategies and briefly compare their performance. To obtain a recursion for the evolution of the network mean error vector, we take the expectation of both sides of (2.40) under Assumption 2.1, which leads to

$$\mathbb{E}[\tilde{\boldsymbol{w}}_i] = \mathcal{B}\,\mathbb{E}[\boldsymbol{w}_{i-1}] \tag{2.45}$$

For the diffusion strategy, we have

$$\mathcal{B} \triangleq \mathbb{E}[\boldsymbol{\mathcal{B}}_i] = \mathcal{A}^T(I - \mathcal{M}\mathcal{R}) \tag{2.46}$$

To obtain (2.45), we used the fact that $\mathbb{E}[\mathcal{A}_2{}^T\mathcal{M}\boldsymbol{g}_i] = 0$; because $\boldsymbol{v}_{k,i}$ is independent of $\boldsymbol{u}_{k,i}$ and $\mathbb{E}[\boldsymbol{v}_k(i)] = 0$. According to (2.45), $\lim_{i\to\infty}\mathbb{E}[\tilde{\boldsymbol{w}}_i] \to 0$ if $\mathcal{B}$ is stable, i.e.,

$$\rho(\mathcal{B}) < 1 \tag{2.47}$$

where $\rho(B)$ denotes the spectral radius of its matrix argument. Because $\rho(\mathcal{A}) = 1$, for diffusion algorithms, we have:

$$\rho(\mathcal{B}) \leq \rho(I - \mathcal{M}\mathcal{R}) \tag{2.48}$$

Since $\mathcal{R} > 0$, it follows from here that choosing the step-sizes according to

$$0 < \mu_k < \frac{2}{\lambda_{\max}(R_{u,k})}, \quad \text{for} \quad k = 1, 2, \cdots, N \tag{2.49}$$

will guarantee $\rho(\mathcal{B}) < 1$ [63, 82]. This condition also guarantees the mean-stability of the non-cooperative network when each node operates individually to estimate w^o (see the analysis of the stand-alone LMS filter [89]).

For consensus strategy, we have

$$\mathcal{B} \triangleq \mathbb{E}[\boldsymbol{\mathcal{B}}_i] = \mathcal{A}^T - \mathcal{M}\mathcal{R} \tag{2.50}$$

If matrix A is symetric, the necessary and sufficient condition for mean convergence and stability is [30]:

$$0 < \mu_k < \frac{1 + \lambda_{\min}(A)}{\lambda_{\max}(R_{u,k})}, \quad \text{for} \quad k = 1, 2, \cdots, N \tag{2.51}$$

Since A is a left-stochastic matrix, its spectral radius is equal to one [93]. In addition, because A is assumed to be symmetric, all its eigenvalues are real and hence its maximum eigenvalue is equal to one, i.e., $\lambda_{\max}(A) = \rho(A) = 1$. This implies that the upper bound in (2.51) is less than the upper bound in (2.49) so that diffusion networks are stable over a wider range of step-sizes. The upper bound in (2.51) can even be zero because $\lambda_{\min}(A)$ can be equal to -1.

An other interesting observation that follow from (2.51) is that if we connect a collection of nodes, that are behaving in a stable manner on their own, through a topology and then apply consensus to solve the same estimation problem through cooperation, then the network may end up being unstable and the estimation task may fail [30]. However, this scenario cannot happen for diffusion strategy as the stability range of diffusion, i.e, step-size range (2.49), does not depend on the choice of matrix A.

2.4.2 Mean-Square Stability

To study the mean-square performance of consensus and diffusion strategies, we need to compute a variance relation, which shows the evolution of mean-square error over the network [33, 89]. To determine this relation, we compute the weighted squared norm of both sides of equation (2.40) and take expectations under Assumption 2.1:

$$\mathbb{E}\|\tilde{\boldsymbol{w}}_i\|^2_{\Sigma} =\mathbb{E}\Big[\|\tilde{\boldsymbol{w}}_{i-1}\|^2_{\boldsymbol{\Sigma}'}\Big] + \mathbb{E}[\boldsymbol{y}_i^{*}\Sigma\boldsymbol{y}_i] \tag{2.52}$$

where $\Sigma \geq 0$ is a weighting matrix that we are free to choose, and

$$\boldsymbol{\Sigma}' = \boldsymbol{\mathcal{B}}_i{}^{*}\Sigma\boldsymbol{\mathcal{B}}_i \tag{2.53}$$

It follows from Assumption 2.1 that $\tilde{\boldsymbol{w}}_{i-1}$ and $\boldsymbol{\mathcal{R}}_i$ are statistically independent so that

$$\mathbb{E}\Big[\|\tilde{\boldsymbol{w}}_{i-1}\|^2_{\boldsymbol{\Sigma}'}\Big] = \mathbb{E}\|\tilde{\boldsymbol{w}}_{i-1}\|^2_{\mathbb{E}[\boldsymbol{\Sigma}']} \tag{2.54}$$

Substituting this expression into (2.52), we arrive at:

$$\mathbb{E}\|\tilde{\boldsymbol{w}}_i\|^2_{\Sigma} = \mathbb{E}\|\tilde{\boldsymbol{w}}_{i-1}\|^2_{\Sigma'} + \mathrm{Tr}[\Sigma\mathcal{Y}] \tag{2.55}$$

where we define $\Sigma' = \mathbb{E}[\boldsymbol{\mathcal{B}}_{\boldsymbol{i}}^{*}\Sigma\boldsymbol{\mathcal{B}}_i]$ and $\mathcal{Y} \triangleq \mathbb{E}[\boldsymbol{y}_i\boldsymbol{y}_i^{*}]$. For diffusion strategy, matrix $\mathcal{Y}$ is given by:

$$\mathcal{Y} = \mathcal{A}^T\mathcal{M}\mathcal{G}\mathcal{M}\mathcal{A} \tag{2.56}$$

with

$$\mathcal{G} = \mathrm{diag}\Big\{\sigma^2_{v,1}R_{u,1}, \sigma^2_{v,2}R_{u,2}, \cdots, \sigma^2_{v,N}R_{u,N}\Big\} \tag{2.57}$$

For consensus strategy matrix $\mathcal{Y}$ takes the form:

$$\mathcal{Y} = \mathcal{M}\mathcal{G}\mathcal{M} \tag{2.58}$$

Now, let U, V and X denote arbitrary matrices with size $NM \times NM$. The following relations hold [93, 94]:

$$\mathrm{vec}(U\Sigma V) = (V^T \otimes U)\mathrm{vec}(\Sigma) \tag{2.59}$$

$$\mathrm{Tr}(\Sigma X) = [\mathrm{vec}(X^T)]^T \mathrm{vec}(\Sigma) \tag{2.60}$$

Introducing $\sigma = \mathrm{vec}(\Sigma)$ and $\sigma' = \mathrm{vec}(\Sigma')$ and using (2.59), we can write

$$\sigma' = \mathcal{F}\sigma \tag{2.61}$$

for some matrix $\mathcal{F}$ to be determined (see below). Using properties (2.59) and (2.60), the variance relation in (2.55) can be rewritten more compactly as:

$$\mathbb{E}\|\tilde{\boldsymbol{w}}_i\|^2_\sigma = \mathbb{E}\|\tilde{\boldsymbol{w}}_{i-1}\|^2_{\mathcal{F}\sigma} + \gamma^T\sigma \tag{2.62}$$

where we are using the notation $\|x\|^2_\sigma$ as a short form for $\|x\|^2_\Sigma$, and

$$\gamma = \mathrm{vec}(\mathcal{Y}^T) \tag{2.63}$$

$$\mathcal{F} = \mathbb{E}[\boldsymbol{\mathcal{B}}_i^T \otimes \boldsymbol{\mathcal{B}}_i^*] \tag{2.64}$$

From relation (2.62), we then arrive at:

$$\mathbb{E}\|\tilde{\boldsymbol{w}}_i\|^2_\sigma = \mathbb{E}\|\tilde{\boldsymbol{w}}_{-1}\|^2_{\mathcal{F}^{i+1}\sigma} + \gamma^T \sum_{j=0}^{i} \mathcal{F}^j \sigma \tag{2.65}$$

From this expression, it can be verified that the mean-square stability[1] of diffusion and consensus strategies depends on the stability of matrix $\mathcal{F}$, i.e., $\lim_{i\to\infty}\mathbb{E}\|\tilde{\boldsymbol{w}}_i\|^2_\sigma$ converges if $\rho(\mathcal{F}) < 1$. A simpler condition for mean-square stability can be obtained by assuming sufficiently small step-sizes since in this case we can verify that the matrix $\mathcal{F}$ in (2.65) can be approximated as follows [82]:

$$\mathcal{F} \approx \mathcal{B}^T \otimes \mathcal{B}^* \tag{2.66}$$

[1] We say that an adaptive algorithm is mean-square stable if $\lim_{i\to\infty}\mathbb{E}\|\tilde{\boldsymbol{w}}_i\|^2_\sigma \to a_o$, where a_o is a positive real number.

Now noting that [93]

$$\rho(\mathcal{F}) \approx \rho(\mathcal{B}^T)\rho(\mathcal{B}^*) = [\rho(\mathcal{B})]^2 \tag{2.67}$$

we deduce that $\rho(\mathcal{F}) < 1$ if $\rho(\mathcal{B}) < 1$. For the ATC diffusion strategy, the stability of $\mathcal{B}$ is guaranteed if condition (2.49) on the step-sizes holds, and this does not depend on the choice of coefficient matrix A. Therefore, the condition (2.49) is sufficient to guarantee stability in the mean and mean-square sense for the ATC diffusion algorithm. In contrast, in consensus strategies, the stability of $\mathcal{B}$ depends on matrix A. For the case of a symmetric matrix A, it has been shown in (2.51) that the stability range of $\mathcal{B}$ is affected by the minimum eigenvalue of A. Therefore, for some choices of A, the consensus strategies may become unstable in the mean-square error sense.

2.4.3 Mean-Square Convergence Rate

Let us assume that both diffusion and consensus strategies are stable in the mean and mean-square sense, i.e., matrix $\mathcal{B}$ is stable in both cases. Under this condition, as time-index i progresses, the mean-square error (2.65), in average, will decay toward its steady-state at a rate r that is governed by $\rho(\mathcal{F})$ or equivalently by $(\rho(\mathcal{B}))^2$ according to (2.67). We note that the smaller the value of $\rho(\mathcal{B})$ is, the faster the rate of convergence of $\lim_{i\to\infty} \mathbb{E}\|\tilde{\boldsymbol{w}}_i\|^2$ will be. Under certain simplifying assumptions, it can be shown that the spectral radius of $\mathcal{B}$ in diffusion strategies is smaller than the consensus one. Therefore, diffusion strategies not only stabilize the network learning, but also converge with faster rate compared with their consensus counterparts [95].

2.4.4 Mean-Square Transient Behavior

We use (2.65) to obtain an expression for the mean-square behavior of the algorithm in transient-state. In this expression, if for some $i > 0$, we substitute $\boldsymbol{w}_{k,-1} = 0$, $\forall k \in \{1, \cdots, N\}$, we obtain:

$$\|\tilde{\boldsymbol{w}}_i\|^2_\sigma = \|w^o\|^2_{\mathcal{F}^{i+1}\sigma} + \gamma^T \sum_{j=0}^{i} \mathcal{F}^j \sigma \tag{2.68}$$

Writing this recursion for $i-1$, and subtract it from (2.68) leads to:

$$\|\tilde{\boldsymbol{w}}_i\|^2_\sigma = \|\tilde{\boldsymbol{w}}_{i-1}\|^2_\sigma - \|w^o\|^2_{\mathcal{F}^i(I-\mathcal{F})\sigma} + \gamma^T\mathcal{F}^i\sigma \tag{2.69}$$

By definition, the instantaneous MSD and excess mean-square error (EMSE) at each node k are respectively defined as:

$$\eta_k(i) = \mathbb{E}\|\tilde{\boldsymbol{w}}_{k,i}\|^2_I, \quad \zeta_k(i) = \mathbb{E}\|\tilde{\boldsymbol{w}}_{k,i}\|^2_{R_{u,k}} \tag{2.70}$$

These metrics can be also retrieved from the network error vector $\tilde{\boldsymbol{w}}_i$ by writing:

$$\eta_k(i) = \mathbb{E}\|\tilde{\boldsymbol{w}}_i\|^2_{\{\text{diag}(e_k)\otimes I\}} \tag{2.71}$$

$$\zeta_k(i) = \mathbb{E}\|\tilde{\boldsymbol{w}}_i\|^2_{\{\text{diag}(e_k)\otimes R_{u,k}\}} \tag{2.72}$$

where e_k is a basis vector in $\mathbb{R}^N$ with entry one at position k. Therefore, in relation (2.69), if we replace σ with

$$\sigma_{\text{msd}_k} = \text{vec}\Big(\text{diag}\{e_k\}\otimes I_M\Big) \tag{2.73}$$

$$\sigma_{\text{emse}_k} = \text{vec}\Big(\text{diag}\{e_k\}\otimes R_{u,k}\Big) \tag{2.74}$$

and use $\boldsymbol{w}_{k,-1} = 0$, we will arrive at the following two recursions for the evolution of MSD and EMSE over time:

$$\eta_k(i) = \eta_k(i-1) - \|w^o\|_{\mathcal{F}^i(I-\mathcal{F})\sigma_{\text{msd}_k}} + \gamma^T\mathcal{F}^i\sigma_{\text{msd}_k} \tag{2.75}$$

$$\zeta_k(i) = \zeta_k(i-1) - \|w^o\|_{\mathcal{F}^{i-1}(I-\mathcal{F})\sigma_{\text{emse}_k}} + \gamma^T\mathcal{F}^{i-1}\sigma_{\text{emse}_k} \tag{2.76}$$

The MSD and EMSE learning behavior of the network can be characterized either by averaging the nodes transient behavior, or equivalently by substituting

$$\sigma_{\text{msd}} = \frac{1}{N}\text{vec}(I_{MN}) \tag{2.77}$$

$$\sigma_{\text{emse}} = \frac{1}{N}\text{vec}\Big(\text{diag}\{R_{u,1},\cdots,R_{u,N}\}\Big) \tag{2.78}$$

in recursion (2.69), respectively.

2.4.5 Steady-State Mean-Square Performance

To obtain the steady-state MSE over the network, we let i goes to infinity and use expression (2.62) to write:

$$\lim_{i\to\infty} \mathbb{E}\|\tilde{\boldsymbol{w}}_i\|^2_{(I-\mathcal{F})\sigma} = [\text{vec}(\mathcal{Y}^T)]^T\sigma \tag{2.79}$$

By definition, the steady-state MSD and EMSE at each node k are respectively computed as:

$$\eta_k = \lim_{i\to\infty} \eta_k(i), \quad \zeta_k = \lim_{i\to\infty} \zeta_k(i) \tag{2.80}$$

Alternatively these performance metrics can be obtained from relation (2.71) and (2.72) by letting $i \to \infty$. Now by equating relations (2.79) and (2.71) for $i \to \infty$, we arrive at:

$$\eta_k = [\text{vec}(\mathcal{Y}^T)]^T\,(I-\mathcal{F})^{-1}\text{vec}(\text{diag}(e_k)\otimes I_M) \tag{2.81}$$

which requires $(I-\mathcal{F})$ to be invertible. This condition is satisfied if $\mathcal{F}$ is stable (i.e., $\rho(\mathcal{F}) < 1$).

In the same manner, from (2.79) and (2.72), we compute the steady-state EMSE of node k:

$$\zeta_k = [\text{vec}(\mathcal{Y}^T)]^T\,(I-\mathcal{F})^{-1}\text{vec}(\text{diag}(e_k)\otimes R_{u,k}) \tag{2.82}$$

The network steady-state MSD and EMSE are defined as the average of the steady-state MSD and EMSE over all nodes, i.e.,

$$\eta = \frac{1}{N}\sum_{k=1}^{N}\eta_k, \quad \zeta = \frac{1}{N}\sum_{k=1}^{N}\zeta_k \tag{2.83}$$

Note that expressions (2.75), (2.76), (2.81), (2.82), and (2.83) can be used to characterize the mean-square performance of both diffusion and consensus strategies by properly choosing the corresponding matrices $\mathcal{B}$ and $\mathcal{Y}$. In general, expressions (2.81), and (2.82) may not provide an explicit way to show that the diffusion networks outperform the consensus ones in the mean-square error sense. However, when matrix A is symmetric, using

(2.65), it can be established analytically that the diffusion networks achieve smaller MSD than the consensus networks [95].

2.5 Summary

In this chapter, we presented the derivation procedure of both diffusion and consensus strategies of distributed adaptation in multi-agent networks. We explained the differences between these two strategies in terms of their operations and compared their stability in the mean and mean-square error sense. It was shown that if step-sizes are sufficiently small so that conditions (2.49) and (2.51) hold, then the diffusion and consensus networks will be stable in the mean and mean-square sense. Under these conditions, the networks achieves steady-state operation as $i \to \infty$. We also presented a systematic way to derive expressions for characterizing the MSD and EMSE of individual nodes over the network in transient and steady-state regimes.

The principal conclusion from this chapter is that the stability of diffusion networks is independent of the choice of combination matrix A, whereas the stability of consensus networks highly depends on the choice of this matrix. This indicates that if we connect a collection of nodes that are behaving in a stable manner in their individual estimation task, through a certain network topology and then apply the consensus algorithm to solve the same estimation problem through cooperation, the network may end up being unstable. We observe that, unlike the consensus strategies, the stability of the individual nodes ensures the stability of diffusion networks irrespective of the network topology. Moreover, if both strategies are assumed to be stable in the mean and mean-square error sense, the diffusion networks generally converge faster and reach lower MSD value. For these reasons, in the reminder of this thesis, we solely focus on DLMS to solve estimation tasks over sensor networks.

Chapter 3

Space-Varying Parameter Estimation Using DLMS Algorithms

In this chapter, we study the problem of distributed adaptive estimation over sensor networks where nodes cooperate to estimate physical parameters that vary over spatial domain[1]. Systems with space-varying parameters are prevalent in many applications in sensor networks, such as process monitoring and the study of diffusion phenomena in inhomogeneous media. We begin our derivation by introducing a generic regression model that characterizes the behavior of systems with spatially-varying parameters over sampled space and time. To enable the implementation of distributed optimization strategies for the introduced model, we use a set of basis functions to represent the space-varying parameters with global (i.e. space-invariant) parameters. Under this representation, we propose a DLMS algorithm that estimates and tracks the global parameters, and retrieves the underlying space-varying parameters over the network from successive time measurements. We analyze the stability and convergence of the proposed algorithm as well as its learning behavior and steady-state performance.

[1] Part of the work presented in this chapter was published in:

- R. Abdolee, B. Champagne and A. H. Sayed, "Estimation of space-time varying parameters using a diffusion LMS algorithm", *IEEE Trans. on Signal Processing*, vol 62, no. 2, pp. 403–418, Jan. 2014.
- R. Abdolee, B. Champagne and A. H. Sayed, "Diffusion LMS for source and process estimation in sensor Networks", in *Proc. IEEE Statistical Signal Processing (SSP) Workshop*, Ann Arbor, MI, Aug. 2012. pp. 165–168 .

3.1 Introduction

In previous studies on diffusion algorithms for adaptation over networks, including least-mean-squares (LMS) or recursive least squares (RLS) types, the parameters being estimated are often assumed to be *space-invariant* [28–30, 33, 37, 82]. In other words, all agents are assumed to sense and measure data that arise from an underlying physical model that is represented by fixed parameters over the spatial domain. Some studies considered particular applications of diffusion strategies to data that arise from potentially different models [95, 96]. However, the proposed techniques in these works are not immediately applicable to scenarios where the estimation parameters vary over space across the network. This situation is encountered in many applications, including the monitoring of fluid flow in underground porous media [38], the tracking of population dispersal in ecology [39], the localization of distributed sources in dynamic systems [97] and the modeling of diffusion phenomena in inhomogeneous media [40]. In these applications, the space-variant parameters being estimated usually result from discretization of the coefficients of an underlying partial differential equation through spatial sampling.

The estimation of spatially-varying parameters has been addressed in several previous studies, including [41–43, 98, 99]. In these works and other similar references on the topic, the solutions typically rely on the use of a central processing (fusion) unit and less attention is paid to distributed solutions. In this chapter, we develop a DLMS algorithm of the diffusion type to enable the estimation and tracking of parameters that may vary over both *space* and *time*. Our approach starts by introducing a linear regression model to characterize space-time varying phenomena over networks. This model is derived by discretizing a representative second-order partial differential equation (PDE), which can be useful in characterizing many dynamic systems with spatially-varying properties. We then introduce a set of basis functions, e.g., shifted Chebyshev polynomials, to represent the space-varying parameters of the underlying phenomena in terms of a finite set of space-invariant expansion coefficients. Building on this representation, we develop a diffusion LMS strategy that cooperatively estimates, interpolates, and tracks the model parameters over the network. We analyze the convergence and stability of the developed algorithm, and derive closed-form expressions to characterize the learning and convergence behavior of the nodes in the mean-square-error sense.

We find that in the estimation of the space-varying parameters using distributed approaches, the covariance matrix of the regression data at each node becomes rank-deficient, which entails complications and hinders the ability of distributed approaches to find optimal estimates of the unknown parameters over the network. Our analysis reveals that the proposed algorithm can overcome this difficulty to a large extent by benefiting from the network stochastic matrices that are used to combine exchanged information between nodes. We provide computer experiments to illustrate and support the theoretical findings.

3.2 Modeling and Problem Formulation

In this section, we motivate a linear regression model that can be used to describe dynamic systems with spatially varying properties. We derive the model from a representative second-order one-dimensional PDE that is used to characterize the evolution of the pressure distribution in inhomogeneous media and features a diffusion coefficient as well as an input source, both of which vary over space. Extension and generalization of the proposed approach, in modeling space-varying phenomena, to PDEs of higher order or defined over two-dimensional space are generally straightforward (see, Section 3.5.3).

The PDE under consideration is expressed as [40, 100]:

$$\frac{\partial f(x,t)}{\partial t} = \frac{\partial}{\partial x}\left(\theta(x)\frac{\partial f(x,t)}{\partial x}\right) + q(x,t) \tag{3.1}$$

where $(x,t) \in [0,L] \times [0,T]$ denote the space and time variables with upper limits $L \in \mathbb{R}^+$ and $T \in \mathbb{R}^+$, respectively, $f(x,t) : \mathbb{R}^2 \rightarrow \mathbb{R}$, represents the system distribution (e.g., pressure or temperature) under study, $\theta(x)\colon \mathbb{R} \rightarrow \mathbb{R}$ is the space-varying diffusion coefficient and $q(x,t)\colon \mathbb{R}^2 \rightarrow \mathbb{R}$ is the input distribution that includes sources and sinks. PDE (3.1) is assumed to satisfy the Dirichlet boundary conditions[1], $f(0,t) = f(L,t) = 0$ for all $t \in [0,T]$. The distribution of the system at $t = 0$ is given by $f(x,0) = y(x)$ for $x \in [0,\ L]$. It is convenient to rewrite (3.1) as:

$$\frac{\partial f(x,t)}{\partial t} = \theta(x)\frac{\partial^2 f(x,t)}{\partial x^2} + \frac{\partial \theta(x)}{\partial x}\frac{\partial f(x,t)}{\partial x} + q(x,t) \tag{3.2}$$

and employ the finite difference method (FDM) to discretize the PDE over the time and

[1] Generalization of the boundary conditions to nonzero values is possible as well.

space domains [101]. For N and P given positive integers, let $\Delta x = L/(N+1)$ and $x_k = k\Delta x$ for $k \in \{0, 1, 2, \ldots, N+1\}$, and similarly, let $\Delta t = T/P$ and $t_i = i\Delta t$ for $i \in \{0, 1, 2, \ldots, P\}$. We further introduce the sampled values of the pressure distribution $f_k(i) \triangleq f(x_k, t_i)$, input $q_k(i) \triangleq q(x_k, t_i)$, and space-varying coefficient $\theta_k \triangleq \theta(x_k)$. It can be verified that applying FDM to (3.2), yields the following recursion:

$$f_k(i) = u_{k,i} h_k^o + \Delta t\, q_k(i-1), \quad k \in \{1, 2, \ldots, N\} \tag{3.3}$$

where the vectors $h_k^o \in \mathbb{R}^{3\times 1}$ and $u_{k,i} \in \mathbb{R}^{1\times 3}$ are defined as

$$h_k^o \triangleq [h_{1,k}^o, h_{2,k}^o, h_{3,k}^o]^T \tag{3.4}$$

$$u_{k,i} \triangleq [f_{k-1}(i-1),\ f_k(i-1),\ f_{k+1}(i-1)] \tag{3.5}$$

the entries $h_{m,k}^o \in \mathbb{R}$ are:

$$h_{1,k}^o = \frac{\nu}{4}(\theta_{k-1} + 4\theta_k - \theta_{k+1}) \tag{3.6}$$

$$h_{2,k}^o = 1 - 2\nu\,\theta_k \tag{3.7}$$

$$h_{3,k}^o = \frac{\nu}{4}(-\theta_{k-1} + 4\theta_k + \theta_{k+1}) \tag{3.8}$$

and $\nu = \Delta t/\Delta x^2$. Note that relation (3.3) is defined for $k \in \{1, 2, \cdots, N\}$, i.e., no data sampling is required to be taken at $x = \{0, L\}$ because $f_0(i)$ and $f_{N+1}(i)$ respectively correspond to the known boundary conditions $f(0,t)$ and $f(L,t)$.

For monitoring purposes (e.g., estimation of $\theta(x)$), sensor nodes collect noisy measurement samples of $f(x,t)$ across the network. We denote these scalar measurement samples by

$$\boldsymbol{z}_k(i) = f_k(i) + \boldsymbol{n}_k(i) \tag{3.9}$$

where $\boldsymbol{n}_k(i) \in \mathbb{R}$ is random noise term. Substituting (3.3) into (3.9) leads to

$$\boldsymbol{d}_k(i) = u_{k,i} h_k^o + \boldsymbol{n}_k(i) \tag{3.10}$$

where

$$\boldsymbol{d}_k(i) \triangleq \boldsymbol{z}_k(i) - \Delta t\, q_k(i-1) \tag{3.11}$$

The space-dependent model (3.10) can be generalized to accommodate higher order PDE's, or to describe systems with more than one spatial dimension. In the generalized form, we assume that $u_{k,i}$ is random due to the possibility of sampling errors, and therefore represent it using boldface notation $\boldsymbol{u}_{k,i}$. We also let h_k^o and $\boldsymbol{u}_{k,i}$ be M-dimensional vectors. In addition, we denote the noise more generally by the symbol $\boldsymbol{v}_k(i)$ to account for different sources of errors, including the measurement noise shown in (3.9) and modeling errors. Considering this generalization, the space-varying regression model that we shall consider is of the form:

$$\boldsymbol{d}_k(i) = \boldsymbol{u}_{k,i} h_k^o + \boldsymbol{v}_k(i) \tag{3.12}$$

where $\boldsymbol{d}_k(i) \in \mathbb{R}, \boldsymbol{u}_{k,i} \in \mathbb{R}^{1\times M}, h_k^o \in \mathbb{R}^{M\times 1}$ and $\boldsymbol{v}_k(i) \in \mathbb{R}$. In this work, we study networks that monitor phenomena characterized by regression models of the form (3.12), where the objective is to estimate the space-varying parameter vectors h_k^o for $k \in \{1, 2, \cdots, N\}$. In particular, we seek a distributed solution in the form of an adaptive algorithm with a diffusion mode of cooperation to enable the nodes to estimate and track these parameters over both space and time. The available information for estimation of the $\{h_k^o\}$ are the measurement samples, $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$, collected at the N spatial position x_k, which we take to represent N nodes.

Several studies, e.g., [41–43], solved space-varying parameter estimation problems using *non-adaptive centralized* techniques. In centralized optimization, the space-varying parameters $\{h_k^o\}$ are found by minimizing the following global cost function over the variables $\{h_k\}$:

$$J(h_1, \ldots, h_N) \triangleq \sum_{k=1}^{N} J_k(h_k) \tag{3.13}$$

where

$$J_k(h_k) \triangleq \mathbb{E}|\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i} h_k|^2 \tag{3.14}$$

To find h_k^o using distributed mechanisms, however, preliminary steps are required to transform the global cost (3.13) into a suitable form convenient for decentralized optimization [33]. Observe from (3.6)-(3.8) that collaborative processing is beneficial in this case because the vectors h_k^o of neighboring nodes are related to each other through the space-dependent function $\theta(x)$.

Remark 3.1. *Note that if nodes individually estimate their own space-varying parameters by minimizing $J_k(h_k)$, then at each time instant, they will need to transmit their estimates* *to a fusion center for interpolation, in order to determine the value of the model parameters over regions of space where no measurements were collected. Using the proposed distributed algorithm in Section 3.3.2, it will be possible to update the estimates and interpolate the results in a fully distributed manner. Cooperation also helps the nodes refine their estimates and perform more accurate interpolation.* ■

3.3 Adaptive Distributed Optimization

In distributed optimization over networked systems, nodes achieve their common objective through collaboration. Such an objective may be defined as finding a global parameter vector that minimizes a given cost function that encompasses the entire set of nodes. For the problem stated in this study, the unknown parameters in (3.13) are node-dependent. However, as we explained in Section 3.2, these space-varying parameters are related through a well-defined function, e.g., $\theta(x)$ over the spatial domain. In the continuous space domain, the entries of each h_k^o, i.e., $\{h_{1,k}^o, \cdots, h_{M,k}^o\}$ can be interpreted as samples of M unknown space-varying parameter functions $\{h_1^o(x), \cdots, h_M^o(x)\}$ at location $x = x_k$, as illustrated in Fig. 3.1.

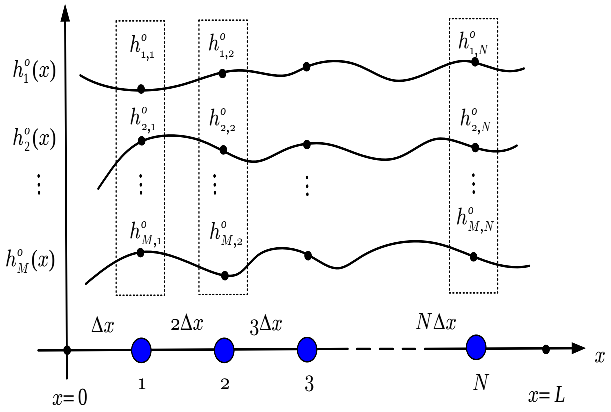


Fig. 3.1 An example of the space-varying parameter estimation problem over a one-dimensional network topology. The larger circles on the x-axis represent the node locations at $x = x_k$. These nodes collect samples $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$ to estimate the space-varying parameters $\{h_k^o\}$. For simplicity in defining the vectors b_k in (3.20), for this example, we assume that the node positions x_k are uniformly spaced, however, generalization to non-uniform spacing is straightforward.

We can now use the well-established theory of interpolation to find a set of linear expansion coefficients, common to all the nodes, in order to estimate space-varying parameters using distributed optimization. Specifically, we assume that the m-th unknown space-varying parameter function, $h_m^o(x)$ can be expressed as a unique linear combination of some N_b space basis functions, i.e.,

$$h_m^o(x) = W_{m,1}b_1(x) + W_{m,2}b_2(x) + \cdots + W_{m,N_b}b_{N_b}(x) \tag{3.15}$$

where $\{W_{m,n}\}$ are the unique expansion coefficients and $\{b_n(x)\}$ are the basis functions. In the application examples treated in Section 3.5, we adopt shifted Chebyshev polynomials as basis functions, which are generated using the following expressions [102]:

$$b_1(x) = 1, \qquad b_2(x) = 2x - 1 \tag{3.16}$$

$$b_{n+1}(x) = 2(2x-1)b_n(x) - b_{n-1}(x), \quad 2 < n < N_b \tag{3.17}$$

The choice of a suitable set of basis functions $\{b_n(x)\}_{n=1}^{N_b}$ is application-specific and guided by multiple considerations such as representation efficiency, low computational complexity, interpolation capability, and other desirable properties, such as orthogonality. Using Chebyshev polynomials as basis functions yields good results in terms of the above criteria and helps avoid the Runge's phenomenon at the endpoints of the space interval [102].

The sampled version of the m-th space-varying parameter $h_m^o(x)$ in (3.15), at $x = x_k = k\Delta x$, can be written as:

$$h_{m,k}^o = W_m^T b_k \tag{3.18}$$

where

$$W_m \triangleq [W_{m,1}, W_{m,2}, \cdots, W_{m,N_b}]^T \tag{3.19}$$

$$b_k \triangleq [b_{1,k}, \cdots, b_{N_b,k}]^T \tag{3.20}$$

and each entry $b_{n,k}$ is obtained by sampling the corresponding basis function at the same location, i.e.,

$$b_{n,k} \triangleq b_n(x_k) = b_n(k\Delta x) \tag{3.21}$$

Collecting the sampled version of all M functions $h_m^o(x)$ for $m \in \{1, \cdots, M\}$ into a column vector as

$$h_k^o = [h_{1,k}^o, h_{2,k}^o, \cdots, h_{M,k}^o]^T \tag{3.22}$$

and using (3.18), we arrive at:

$$h_k^o = W^o b_k \tag{3.23}$$

where

$$W^o \triangleq \begin{bmatrix} W_{1,1}^o & W_{1,2}^o & \dots & W_{1,N_b}^o \\ W_{2,1}^o & W_{2,2}^o & \dots & W_{2,N_b}^o \\ \vdots & \vdots & \dots & \vdots \\ W_{M,1}^o & W_{M,2}^o & \dots & W_{M,N_b}^o \end{bmatrix} \tag{3.24}$$

Remark 3.2. *Several other interpolation techniques can be used to obtain the basis functions $b_n(x)$, such as the so-called kriging method [103]. The latter is a data-based weighting approach, rather than a distance-based interpolation. It is applicable in scenarios where the unknown random field to be interpolated, in our case h_k^o, is wide-sense stationary; accordingly, it requires knowledge about the means and covariances of the random field over space, as employed in [104]. If these covariances are not available, then the variogram models, describing the degree of spatial dependence of the random field, are used to generate substitutes for these covariances [105]. However, a-priori knowledge about the parameters of variogram models, including nugget, sill, and range, are required to obtain the spatial covariances. In this work, since neither the means and covariances nor the parameters of the variogram models of the random fields are available, we focus on interpolation techniques that rely on distance information rather than the statistics of the random field to be interpolated.* ■

Returning to equation (3.23), it is convenient to rearrange W^o into an $MN_b \times 1$ column vector w^o by stacking up the columns of W^{oT}, i.e., $w^o = \text{vec}(W^{oT})$, and defining the block diagonal matrix $B_k \in \mathbb{R}^{M \times MN_b}$ as:

$$B_k \triangleq I_M \otimes b_k^T \tag{3.25}$$

Then, relation (3.23) can be rewritten in terms of the unique parameter vector w^o as:

$$h_k^o = B_k w^o \tag{3.26}$$

so that substituting h_k^o from (3.26) into (3.12) yields:

$$\boldsymbol{d}_k(i) = \boldsymbol{u}_{k,i} B_k w^o + \boldsymbol{v}_k(i) \tag{3.27}$$

Subsequently, the global cost function (3.13) becomes:

$$J(w) = \sum_{k=1}^{N} \mathbb{E}|\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i} B_k w|^2 \tag{3.28}$$

In the following, we elaborate on how the parameter vector w^o and, hence, the $\{h_k^o\}$ can be estimated from the data $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$ using centralized and distributed adaptive optimization.

3.3.1 Centralized Adaptive Solution

We begin by stating the assumed statistical conditions on the data (3.27) over the network.

Assumption 3.1. *Statistical assumption of the network data model (3.27)*

a) $\boldsymbol{d}_k(i)$ *and* $\boldsymbol{u}_{k,i}$ *are zero-mean, jointly wide-sense stationary random processes with second-order moments:*

$$r_{du,k} = \mathbb{E}[\boldsymbol{d}_k(i)\boldsymbol{u}_{k,i}^T] \in \mathbb{R}^{M\times 1} \tag{3.29}$$

$$R_{u,k} = \mathbb{E}[\boldsymbol{u}_{k,i}^T \boldsymbol{u}_{k,i}] \in \mathbb{R}^{M\times M} \tag{3.30}$$

b) The regression data $\{\boldsymbol{u}_{k,i}\}$ *are i.i.d. over time, independent over space, and their covariance matrices,* $R_{u,k}$*, are positive definite for all* k*.*

c) The noise processes $\{\boldsymbol{v}_k(i)\}$ *are zero-mean, i.i.d. over time, and independent over space with variances* $\{\sigma_{v,k}^2\}$*.*

d) The regression data $\boldsymbol{u}_{k,i}$ and the noise $\boldsymbol{v}_m(n)$ are mutually independent for all k, m, i and n.

■

The optimal parameter w^o that minimizes (3.28) can be found by setting the gradient vector of $J(w)$ to zero. This yields the following normal equations:

$$\left(\sum_{k=1}^{N} \bar{R}_{u,k}\right) w^o = \sum_{k=1}^{N} \bar{r}_{du,k} \tag{3.31}$$

where $\{\bar{R}_{u,k}, \bar{r}_{du,k}\}$ denote the second-order moments of $\boldsymbol{u}_{k,i}B_k$ and $\boldsymbol{d}_k(i)$:

$$\bar{R}_{u,k} \triangleq B_k^T R_{u,k} B_k, \quad \bar{r}_{du,k} \triangleq B_k^T r_{du,k} \tag{3.32}$$

It is clear from (3.31) that when $\sum_{k=1}^{N} \bar{R}_{u,k} > 0$, then w^o can be determined uniquely. If, on the other hand, $\sum_{k=1}^{N} \bar{R}_{u,k}$ is singular, then we can use its pseudo-inverse to recover the minimum-norm solution of (3.31). Once the global solution is estimated, we can retrieve the space-varying parameter vectors h_k^o by substituting w^o into (3.26).

Alternatively the solution w^o of (3.31) can be sought iteratively by using the following steepest descent recursion:

$$\boldsymbol{w}_i^{(c)} = \boldsymbol{w}_{i-1}^{(c)} + \mu \sum_{k=1}^{N} \left(\bar{r}_{du,k} - \bar{R}_{u,k} \boldsymbol{w}_{i-1}^{(c)}\right) \tag{3.33}$$

where $\mu > 0$ is a step-size parameter and $\boldsymbol{w}_i^{(c)}$ is the estimate of w^o at the i-th iteration. Recursion (3.33) requires the centralized processor to have knowledge of the covariance matrices, $\bar{R}_{u,k}$, and cross covariance vectors, $\bar{r}_{du,k}$, across all nodes. In practice, these moments are unknown in advance, and we therefore use their instantaneous approximations in (3.33). This substitution leads to the centralized LMS strategy (3.34)–(3.35) for space-varying parameter estimation over networks.

In this algorithm, at any given time instant i, each node transmits its data $\{\boldsymbol{u}_{k,i}, \boldsymbol{d}_k(i)\}$ to the central processing unit to update $\boldsymbol{w}_{i-1}^{(c)}$. Subsequently, the algorithm obtains an estimate for the space-varying parameters, $\boldsymbol{h}_{k,i}$, by using the updated estimate $\boldsymbol{w}_i^{(c)}$, and the basis function matrix at location k, (i.e., B_k). This latter step can also be used as an interpolation mechanism to estimate the space-varying parameters at locations other

Algorithm 3.1 : Centralized LMS for space-varying parameter estimation

$$\boldsymbol{w}_i^{(c)} = \boldsymbol{w}_{i-1}^{(c)} + \mu \sum_{k=1}^{N} B_k^T \boldsymbol{u}_{k,i}^T \big(\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i} B_k \boldsymbol{w}_{i-1}^{(c)}\big) \tag{3.34}$$

$$\boldsymbol{h}_{k,i} = B_k \boldsymbol{w}_i^{(c)}, \quad k \in \{1, 2, \cdots, N\} \tag{3.35}$$

than the pre-determined locations $\{x_k\}$, by using the corresponding matrix $B(x)$ for some desired location x.

3.3.2 Adaptive Diffusion Strategy

There are different distributed optimization techniques that can be applied to (3.28) in order to estimate w^o and consequently obtain the optimal space-varying parameters h_k^o. Let $\mathcal{N}_k$ denote the index set of the neighbors of node k, i.e., the nodes with which node k can share information (including k itself). One possible optimization strategy is to decouple the global cost (3.28) and write it as a set of constrained optimization problems with local variables w_k, [106], i.e.,

$$\begin{aligned} &\min_{w_k} \sum_{\ell \in \mathcal{N}_k} c_{\ell,k} \mathbb{E} |\boldsymbol{d}_\ell(i) - \boldsymbol{u}_{\ell,i} B_k w_k|^2 \\ &\text{subject to} \quad w_k = w \end{aligned} \tag{3.36}$$

where $c_{\ell,k}$ are nonnegative entries of a right-stochastic matrix $C \in \mathbb{R}^{N \times N}$ satisfying:

$$c_{\ell,k} = 0 \text{ if } \ell \notin \mathcal{N}_k \quad \text{and} \quad C\mathbb{1} = \mathbb{1} \tag{3.37}$$

The optimization problem (3.36) can be solved using, for example, the alternating directions method of multipliers (ADMM) [79, 106]. In the algorithm derived using this method, the Lagrangian multipliers associated with the constraints need to be updated at every iteration during the optimization process. To this end, information about the network topology is required to establish a hierarchial communication structure between nodes. In addition, the constraints imposed by (3.36) require all agents to agree on an exact solution; this requirement degrades the learning and tracking abilities of the nodes over the network.

When some nodes observe relevant data, it is advantageous for them to be able to respond quickly to the data without being critically constrained by perfect agreement at that stage. Doing so, would enable information to diffuse more rapidly across the network.

A technique that does not suffer from these difficulties and endows networks with adaptation and learning abilities in real-time is the diffusion strategy [27, 29, 33, 82, 107]. In this technique, minimizing the global cost (3.28) motivates solving the following unconstrained local optimization problem for $k \in \{1, \cdots, N\}$ [33]:

$$\min_{w} \left(\sum_{\ell \in \mathcal{N}_k} c_{\ell,k} \mathbb{E} |\boldsymbol{d}_\ell(i) - \boldsymbol{u}_{\ell,i} B_k w|^2 + \sum_{\ell \in \mathcal{N}_k \backslash \{k\}} p_{\ell,k} \|w - w^o\|^2 \right) \tag{3.38}$$

where $\{p_{\ell,k}\}$ are nonnegative scaling parameters. Following the arguments in [29, 33, 82], the minimization of (3.38) leads to a general form of the diffusion strategy described by (3.39)–(3.42), which can be specialized to several simpler yet useful forms.

Algorithm 3.2 : Diffusion LMS for space-varying parameter estimation

$$\boldsymbol{\phi}_{k,i-1} = \sum_{\ell \in \mathcal{N}_k} a^{(1)}_{\ell,k} \boldsymbol{w}_{\ell,i-1} \tag{3.39}$$

$$\boldsymbol{\psi}_{k,i} = \boldsymbol{\phi}_{k,i-1} + \mu_k \sum_{\ell \in \mathcal{N}_k} c_{\ell,k} B_\ell^T \boldsymbol{u}_{\ell,i}^T \big(\boldsymbol{d}_\ell(i) - \boldsymbol{u}_{\ell,i} B_\ell \boldsymbol{\phi}_{k,i-1}\big) \tag{3.40}$$

$$\boldsymbol{w}_{k,i} = \sum_{\ell \in \mathcal{N}_k} a^{(2)}_{\ell,k} \boldsymbol{\psi}_{\ell,i} \tag{3.41}$$

$$\boldsymbol{h}_{k,i} = B_k \boldsymbol{w}_{k,i} \tag{3.42}$$

In this algorithm, $\mu_k > 0$ is the step-size at node k, $\{\boldsymbol{w}_{k,i}, \boldsymbol{\psi}_{k,i}, \boldsymbol{\phi}_{k,i-1}\}$ are intermediate estimates of w^o, $\boldsymbol{h}_{k,i}$ is an intermediate estimate of h^o_k, and $\{a^{(1)}_{\ell,k}, a^{(2)}_{\ell,k}\}$ are nonnegative entries of left-stochastic matrices $A_1, A_2 \in \mathbb{R}^{N \times N}$ that satisfy:

$$a^{(1)}_{\ell,k} = a^{(2)}_{\ell,k} = 0 \quad \text{if}\, \ell \notin \mathcal{N}_k \tag{3.43}$$

$$A_1^T \mathbb{1} = \mathbb{1} \quad A_2^T \mathbb{1} = \mathbb{1} \tag{3.44}$$

Each node k in the first combination step fuses $\{\boldsymbol{w}_{\ell,i-1}\}_{\ell \in \mathcal{N}_k}$ in a convex manner to generate $\boldsymbol{\phi}_{k,i-1}$. In the following step, named adaptation, each node k uses its own data and that of neighboring nodes, i.e., $\Big\{\boldsymbol{u}_{\ell,i}, \boldsymbol{d}_\ell(i)\Big\}_{\ell \in \mathcal{N}_k}$ to adaptively update $\boldsymbol{\phi}_{k,i-1}$ to an intermediate estimate $\boldsymbol{\psi}_{k,i}$. In the third step, which is also a combination, the intermediate estimates $\{\boldsymbol{\psi}_{\ell,i}\}_{\ell\in\mathcal{N}_k}$ are fused to further align the global parameter estimate at node k to that of its neighbors. Subsequently, the desired space-varying parameter $\boldsymbol{h}_{k,i}$ is obtained from $\boldsymbol{w}_{k,i}$. Note that each step in the algorithm runs concurrently over the network.

Remark 3.3. *The main difference between Algorithm 3.2 and the previously developed diffusion LMS strategies in, e.g., [27, 33, 82] is in the transformed domain regression data* $\boldsymbol{u}_{\ell,i}B_\ell$ *in (3.40) which now have singular covariance matrices. Moreover, there is an additional interpolation step (3.42).* ■

Remark 3.4. *The proposed diffusion LMS algorithm estimates* NM *spatially dependent variables* $\{h_k^o\}$ *using* N_bM *global invariant coefficients in* w^o. *From the computational complexity and energy efficiency point of view, it seems this is advantageous when the number of nodes,* N, *is greater than the number of basis functions* N_b. *However, even if this is not the case, using the estimated* N_bM *global invariant coefficients, the algorithm not only can estimate the space-varying parameters at the locations of the* N *nodes, but can also estimate the space-varying parameters at locations where no measurements are available. Therefore, even when* $N < N_b$, *the algorithm is still useful as it can perform interpolation.* ■

There are different choices for the combination matrices $\{A_1, A_2, C\}$. For example, the choice $A_1 = A_2 = C = I$ reduces the above diffusion algorithm to the non-cooperative case where each node runs an individual LMS filter without coordination with its neighbors. Selecting $C = I$ simplifies the adaptation step (3.40) to the case where node k uses only its own data $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$ to perform local adaptation. Choosing $A_1 = I$ and $A_2 = A$, for some left-stochastic matrix A, removes the first combination step and the algorithm reduces to an adaptation step followed by combination (this variant of the algorithm has the Adapt-then-Combine or ATC diffusion structure) [33, 82]. Likewise, choosing $A_1 = A$ and $A_2 = I$ removes the second combination step and the algorithm reduces to a combination step followed by adaptation (this variant has the Combine-then-Adapt (CTA) structure of diffusion [33, 82]). Often in practice, either the ATC or CTA version of the algorithm is used with C set to $C = I$. Nevertheless for generality, we shall study the performance of Algorithm 3.2 for arbitrary matrices $\{A_1, A_2, C\}$ with C right-stochastic and $\{A_1, A_2\}$ left-stochastic. The results can then be specialized to various situations of interest, including

ATC, CTA, and the non-cooperative case. The combination matrices $\{A_1, A_2, C\}$ are normally obtained using some well-known available combination rules such as the Metropolis or uniform combination rules [19,27,33]. These matrices can also be treated as free variables in the optimization procedure and used to further enhance the performance of the diffusion strategies. Depending on the network topology and the quality of the communication links between nodes, the optimized values of the combination matrices differ from one case to another [53, 54, 63, 82].

3.4 Performance Analysis

In this section, we analyze the performance of the diffusion strategy (3.39)-(3.42) in the mean and mean-square sense and derive expressions to characterize the network mean-square deviation (MSD) and excess mean-square error (EMSE). In the analysis, we need to consider the fact that the covariance matrices $\{\bar{R}_{u,k}\}_{k=1}^{N}$ defined in (3.32) are now rank-deficient since we have $N_b > 1$. We explain in the sequel the ramifications that follow from this rank-deficiency.

3.4.1 Mean Convergence

We introduce the local weight-error vectors

$$\tilde{\boldsymbol{w}}_{k,i} \triangleq w^o - \boldsymbol{w}_{k,i} \tag{3.45}$$

$$\tilde{\boldsymbol{\psi}}_{k,i} \triangleq w^o - \boldsymbol{\psi}_{k,i} \tag{3.46}$$

$$\tilde{\boldsymbol{\phi}}_{k,i} \triangleq w^o - \boldsymbol{\phi}_{k,i} \tag{3.47}$$

and define the network error vectors:

$$\tilde{\boldsymbol{\phi}}_{i} \triangleq \text{col}\{\tilde{\boldsymbol{\phi}}_{1,i}, \ldots, \tilde{\boldsymbol{\phi}}_{N,i}\} \tag{3.48}$$

$$\tilde{\boldsymbol{\psi}}_{i} \triangleq \text{col}\{\tilde{\boldsymbol{\psi}}_{1,i}, \ldots, \tilde{\boldsymbol{\psi}}_{N,i}\} \tag{3.49}$$

$$\tilde{\boldsymbol{w}}_{i} \triangleq \text{col}\{\tilde{\boldsymbol{w}}_{1,i}, \ldots, \tilde{\boldsymbol{w}}_{N,i}\} \tag{3.50}$$

We collect the estimates from across the network into the block vector:

$$\boldsymbol{w}_i \triangleq \text{col}\{\boldsymbol{w}_{1,i}, \ldots, \boldsymbol{w}_{N,i}\} \tag{3.51}$$

and introduce the following extended combination matrices:

$$\mathcal{A}_1 \triangleq A_1 \otimes I_{MN_b} \tag{3.52}$$

$$\mathcal{A}_2 \triangleq A_2 \otimes I_{MN_b} \tag{3.53}$$

$$\mathcal{C} \triangleq C \otimes I_{MN_b} \tag{3.54}$$

We further define the block diagonal matrices and vectors:

$$\boldsymbol{\mathcal{R}}_i \triangleq \text{diag}\left\{ \sum_{\ell \in \mathcal{N}_k} c_{\ell,k} B_\ell^T \boldsymbol{u}_{\ell,i}^T \boldsymbol{u}_{\ell,i} B_\ell : k = 1, \cdots, N \right\} \tag{3.55}$$

$$\mathcal{M} \triangleq \text{diag}\left\{ \mu_1 I_{MN_b}, \ldots, \mu_N I_{MN_b} \right\} \tag{3.56}$$

$$\boldsymbol{t}_i \triangleq \text{col}\left\{ \sum_{\ell \in \mathcal{N}_k} c_{\ell,k} B_\ell^T \boldsymbol{u}_{\ell,i}^T \boldsymbol{d}_\ell(i) : k = 1, \cdots, N \right\} \tag{3.57}$$

$$\boldsymbol{g}_i \triangleq \mathcal{C}^T \, \text{col}\left\{ B_1^T \boldsymbol{u}_{1,i}^T \boldsymbol{v}_1(i), \cdots, B_N^T \boldsymbol{u}_{N,i}^T \boldsymbol{v}_N(i) \right\} \tag{3.58}$$

and introduce the expected values of $\boldsymbol{\mathcal{R}}_i$ and $\boldsymbol{t}_i$:

$$\mathcal{R} \triangleq \mathbb{E}[\boldsymbol{\mathcal{R}}_i] = \text{diag}\left\{ R_1, \cdots, R_N \right\} \tag{3.59}$$

$$r \triangleq \mathbb{E}[\boldsymbol{t}_i] = \text{col}\left\{ r_1, \cdots, r_N \right\} \tag{3.60}$$

where

$$R_k \triangleq \sum_{\ell \in \mathcal{N}_k} c_{\ell,k} \, \bar{R}_{u,\ell} \tag{3.61}$$

$$r_k \triangleq \sum_{\ell \in \mathcal{N}_k} c_{\ell,k} \, \bar{r}_{du,\ell} \tag{3.62}$$

We also introduce an indicator matrix operator, denoted by $\text{Ind}(\cdot)$, such that for any real-valued matrix X with (k,j)-th entry $X_{k,j}$, the corresponding entry of $Y = \text{Ind}(X)$ is:

$$Y_{k,j} = \begin{cases} 1, & \text{if } X_{k,j} > 0 \\ 0, & \text{otherwise} \end{cases} \tag{3.63}$$

Now from (3.39)–(3.41), we obtain:

$$\boldsymbol{w}_i = \boldsymbol{\mathcal{B}}_i \boldsymbol{w}_{i-1} + \mathcal{A}_2^T \mathcal{M} \boldsymbol{t}_i \tag{3.64}$$

where

$$\boldsymbol{\mathcal{B}}_i \triangleq \mathcal{A}_2^T (I - \mathcal{M}\boldsymbol{\mathcal{R}}_i) \mathcal{A}_1^T \tag{3.65}$$

In turn, making use of (3.27) in (3.64), we can verify that the network error vector follows the recursion

$$\tilde{\boldsymbol{w}}_i = \boldsymbol{\mathcal{B}}_i \tilde{\boldsymbol{w}}_{i-1} - \mathcal{A}_2^T \mathcal{M} \boldsymbol{g}_i \tag{3.66}$$

By taking the expectation of both sides of (3.66) and using Assumption 3.1, we arrive at:

$$\mathbb{E}[\tilde{\boldsymbol{w}}_i] = \mathcal{B}\, \mathbb{E}[\tilde{\boldsymbol{w}}_{i-1}] \tag{3.67}$$

where in this relation:

$$\mathcal{B} \triangleq \mathbb{E}[\boldsymbol{\mathcal{B}}_i] = \mathcal{A}_2^T (I - \mathcal{M}\mathcal{R}) \mathcal{A}_1^T \tag{3.68}$$

To obtain (3.67), we used the fact that the expectation of the second term in (3.66), i.e., $\mathbb{E}[\mathcal{A}_2^T \mathcal{M} \boldsymbol{g}_i]$, is zero because $\boldsymbol{v}_k(i)$ is independent of $\boldsymbol{u}_{k,i}$ and $\mathbb{E}[\boldsymbol{v}_k(i)] = 0$. The rank-deficient matrices $\{\bar{R}_{u,k}\}$ appear inside $\mathcal{R}$ in (3.68). We now verify that despite having rank-deficient matrix $\mathcal{R}$, recursion (3.67) still guarantees a bounded mean error vector in steady-state.

To proceed, we introduce the eigendecomposition of each diagonal block R_k and write:

$$R_k = Q_k \Lambda_k Q_k^T \tag{3.69}$$

where $Q_k = [q_{k,1}, \cdots, q_{k,MN_b}]$ is a unitary matrix with column eigenvectors $q_{k,j}$ and $\Lambda_k =$

$\text{diag}\{\lambda_k(1), \cdots, \lambda_k(MN_b)\}$ is a diagonal matrix with eigenvalues $\lambda_k(j) \geq 0$. For this decomposition, we assume that the eigenvalues of R_k are arranged in descending order, i.e, $\lambda_{\max}(R_k) \triangleq \lambda_k(1) \geq \lambda_k(2) \geq \cdots \geq \lambda_k(MN_b)$, and the rank of R_k is $L_k \leq MN_b$. If we define $\mathcal{Q} \triangleq \text{diag}\{Q_1, \ldots, Q_N\}$ and $\Lambda \triangleq \text{diag}\{\Lambda_1, \cdots, \Lambda_N\}$, then the network covariance matrix, $\mathcal{R}$, given by (3.59) can be expressed as:

$$\mathcal{R} = \mathcal{Q}\Lambda\mathcal{Q}^T \tag{3.70}$$

We now note that the mean estimate vector, $\mathbb{E}[\tilde{\boldsymbol{w}}_i]$, expressed by (3.67) will be asymptotically unbiased if the spectral radius of $\mathcal{B}$, denoted by $\rho(\mathcal{B})$, is strictly less than one. Let us examine under what conditions this requirement is satisfied. Since A_1 and A_2 are left-stochastic matrices and $\mathcal{R}$ is block-diagonal, we have from [82] that:

$$\rho(\mathcal{B}) = \rho\Big(\mathcal{A}_2^T(I - \mathcal{M}\mathcal{R})\mathcal{A}_1^T\Big) \leq \rho\Big(I - \mathcal{M}\mathcal{R}\Big) \tag{3.71}$$

Therefore, if $\mathcal{R}$ is positive-definite, then choosing $\mu_k < 2/\lambda_{\max}(R_k)$ ensures convergence of the algorithm in the mean so that $\mathbb{E}[\tilde{\boldsymbol{w}}_i] \to 0$ as $i \to \infty$. However, when $\mathcal{R}$ is singular, it may hold that $\rho(\mathcal{B}) = 1$, in which case choosing the step-sizes according to the above bound guarantees the boundedness of the mean error, $\mathbb{E}[\tilde{\boldsymbol{w}}_i]$, but not necessarily that it converges to zero. The following result clarifies these observations.

Theorem 3.1. *If the step-sizes are chosen to satisfy*

$$0 < \mu_k < \frac{2}{\lambda_{\max}(R_k)} \tag{3.72}$$

then, under Assumption 3.1, the diffusion algorithm is stable in the mean in the following sense: (a) If $\rho(\mathcal{B}) < 1$, then $\mathbb{E}[\tilde{\boldsymbol{w}}_i]$ converges to zero and (b) if $\rho(\mathcal{B}) = 1$ then

$$\lim_{i\to\infty} \Big\|\mathbb{E}[\tilde{\boldsymbol{w}}_i]\Big\|_{b,\infty} \leq \|I - \text{Ind}(\Lambda)\|_{b,\infty} \Big\|\mathbb{E}[\tilde{\boldsymbol{w}}_{-1}]\Big\|_{b,\infty} \tag{3.73}$$

where $\|\cdot\|_{b,\infty}$ stands for the block-maximum norm, as defined in [63, 82].

Proof. See Appendix A.1. □

In what follows, we examine recursion (3.64) and derive an expression for the asymptotic value of $\mathbb{E}[\boldsymbol{w}_i]$—see (3.88) further ahead. Before do so, we first comment on a special case of interest, namely, result (3.75) below.

Special case: Consider a network with $A_1 = A_2 = I$ and an arbitrary right stochastic matrix C satisfying (3.37). Using (3.27) and (3.61)-(3.62), it can be verified that the following linear system of equations holds at each node k:

$$R_k w^o = r_k \tag{3.74}$$

We show in Appendix A.2 that under condition (3.72) the mean estimate of the diffusion LMS algorithm at each node k will converge to:

$$\lim_{i\to\infty} \mathbb{E}[\boldsymbol{w}_{k,i}] = R_k^{\dagger} r_k + \sum_{n=L_k+1}^{MN_b} q_{k,n} q_{k,n}^T \mathbb{E}[\boldsymbol{w}_{k,-1}] \tag{3.75}$$

where $R_k^{\dagger}$ represents the pseudo-inverse of R_k, and $\boldsymbol{w}_{k,-1}$ is the node initial value. This result is consistent with the mean estimate of the stand-alone LMS filter with rank-deficient input data (which corresponds to the situation $A_1 = A_2 = C = I$) [108]. Note that $R_k^{\dagger} r_k$ in (3.75) corresponds to the minimum-norm solution of $R_k w = r_k$. Therefore, the second term on the right hand side of (3.75) is the deviation of the node estimate from this minimum-norm solution. The presence of this term after convergence is due to the zero eigenvalues of R_k. If R_k were full-rank so that $L_k = MN_b$, then this term would disappear and the node estimate will converge, in the mean, to its optimal value, w^o. We point out that even though the matrices $\bar{R}_{u,\ell}$ are rank deficient since $N_b > 1$, it is still possible for the matrices R_k to be full rank owing to the linear combination operation in (3.61). This illustrates one of the benefits of employing the right-stochastic matrix C. However, if despite using C, R_k still remains rank-deficient, the second term on the right-hand side of (3.75) can be annihilated by proper node initialization (e.g., by setting $\mathbb{E}[\boldsymbol{w}_{k,-1}] = 0$). By doing so, the mean estimate of each node will then approach the unique minimum-norm solution, $R_k^{\dagger} r_k$.

General case: Let us now find the mean estimate of the network for arbitrary left-stochastic matrices A_1 and A_2. Considering definitions (3.59)-(3.60) and relation (3.74) and noting that $\mathcal{A}_1^T(\mathbb{1} \otimes w^o) = \mathcal{A}_2^T(\mathbb{1} \otimes w^o) = (\mathbb{1} \otimes w^o)$, it can be verified that $(\mathbb{1} \otimes w^o)$

satisfies the following linear system of equations:

$$(I - \mathcal{B})(\mathbb{1} \otimes w^o) = \mathcal{A}_2^T \mathcal{M} r \tag{3.76}$$

This is a useful intermediate result that will be applied in our argument.

Next, if we iterate recursion (3.64) and apply the expectation operator, we then obtain

$$\mathbb{E}[\boldsymbol{w}_i] = \mathcal{B}^{i+1}\mathbb{E}[\boldsymbol{w}_{-1}] + \sum_{j=0}^{i} \mathcal{B}^j \mathcal{A}_2^T \mathcal{M} r \tag{3.77}$$

The mean estimate of the network can be found by computing the limit of this expression for $i \to \infty$. To find the limit of the first term on the right hand side of (3.77), we evaluate $\lim_{i\to\infty} \mathcal{B}^i$ and find conditions under which it converges. For this purpose, we introduce the Jordan decomposition of matrix $\mathcal{B}$ as [93]:

$$\mathcal{B} = \mathcal{Z}\Gamma\mathcal{Z}^{-1} \tag{3.78}$$

where $\mathcal{Z}$ is an invertible matrix, and Γ is a block diagonal matrix of the form

$$\Gamma = \text{diag}\Big\{\Gamma_1, \Gamma_2, \cdots, \Gamma_s\Big\} \tag{3.79}$$

where the l-th Jordan block, $\Gamma_l \in \mathbb{C}^{m_l \times m_l}$, can be expressed as:

$$\Gamma_l = \gamma_l I_{m_l} + N_{m_l} \tag{3.80}$$

In this relation, N_{m_l} is some nilpotent matrix of size $m_l \times m_l$. Using decomposition (3.78), we can express $\mathcal{B}^i$ as

$$\mathcal{B}^i = \mathcal{Z}\Gamma^i\mathcal{Z}^{-1} \tag{3.81}$$

Since Γ is block diagonal, we have

$$\Gamma^i = \text{diag}\Big\{\Gamma_1^i, \Gamma_2^i, \cdots, \Gamma_s^i\Big\} \tag{3.82}$$

From this relation, it is deduced that $\lim_{i\to\infty} \mathcal{B}^i$ exists if $\lim_{i\to\infty} \Gamma_l^i$ exists for all $l \in$

$\{1, \cdots, s\}$. Using (3.80), we can write [93]:

$$\lim_{i\to\infty} \Gamma_l^i = \lim_{i\to\infty} \gamma_l^{i-m_l} \left(\gamma_l^{m_l} I_{m_l} + \sum_{p=1}^{m_l-1} \binom{i}{p} \gamma_l^{m_l-p} N_{m_l}^p \right) \tag{3.83}$$

When $i \to \infty$, $\gamma_l^{i-m_l}$ becomes the dominant factor in this expression. Note that under condition (3.72), we have $\rho(\mathcal{B}) \leq 1$ which in turn implies that the magnitude of the eigenvalues of $\mathcal{B}$ are bounded as $0 \leq |\gamma_n| \leq 1$. Without loss of generality, we assume that the eigenvalues of $\mathcal{B}$ are arranged as $|\gamma_1| \leq \cdots \leq |\gamma_L| < |\gamma_{L+1}| = \cdots = |\gamma_s| = 1$. Now we examine the limit (3.83) for every $|\gamma_l|$ in this range. Clearly for $|\gamma_l| < 1$, the limit is zero (an obvious conclusion since in this case Γ_l is a stable matrix). For $|\gamma_l| = 1$, the limit is the identity matrix if $\gamma_l = 1$ and $m_l = 1$. However, the limit does not exist for unit magnitude complex eigenvalues and eigenvalues with value -1, even when $m_l = 1$. Motivated by these observations, we introduce the following definition.

Definition: We refer to matrix $\mathcal{B}$ as *power convergent* if (a) its eigenvalues γ_n satisfy $0 \leq |\gamma_n| \leq 1$, (b) its unit magnitude eigenvalues are all equal to one, and (c) its Jordan blocks associated with $\gamma_n = 1$ are all of size 1×1. ■

Example 1: Assume $N_b = 1$, $B_k = I_M$, and uniform step-sizes and covariance matrices across the agents, i.e., $\mu_k \equiv \mu$, $R_{u,k} \equiv R_u$ for all k. Assume further that C is doubly-stochastic (i.e., $C^T\mathbb{1} = \mathbb{1} = C\mathbb{1}$). Then, in this case, the matrix $\mathcal{B}$ can be written as the Kronecker product $\mathcal{B} = A_2^T A_1^T \otimes (I_M - \mu R_u)$. For strongly-connected networks where A_1A_2 is a primitive matrix, it follows from the Perron-Frobenius Theorem [91] that A_1A_2 has a single unit-magnitude eigenvalue at one, while all other eigenvalues have magnitude less than one. We conclude in this case, from the properties of Kronecker products and under condition (3.72), that $\mathcal{B}$ is a power-convergent matrix. ■

Example 2: Assume $M = 2$, $N = 3$, $N_b = 1$, $B_k = I_M$, and uniform step-sizes and covariance matrices across the agents again. Let $A_2 = I = C$ and select

$$A_1 = A = \begin{bmatrix} 1/2 & 0 & 0 \\ 1/2 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \tag{3.84}$$

which is not primitive. Let further $R_u = \text{diag}\{\beta, 0\}$ denoe a singular covariance matrix. Then, it can be verified in this case the corresponding matrix $\mathcal{B}$ will have an eigenvalue with value -1 and is not power convergent. ■

Returning to the above definition and assuming $\mathcal{B}$ is power convergent, then this means that the Jordan decomposition (3.78) can be rewritten as:

$$\mathcal{B} = \underbrace{[\mathcal{Z}_1 \; \mathcal{Z}_2]}_{\mathcal{Z}} \underbrace{\begin{bmatrix} J & 0 \\ 0 & I \end{bmatrix}}_{\Gamma} \underbrace{\begin{bmatrix} \bar{\mathcal{Z}}_1 \\ \bar{\mathcal{Z}}_2 \end{bmatrix}}_{\mathcal{Z}^{-1}} \tag{3.85}$$

where J is a Jordan matrix with all eigenvalues strictly inside the unit circle, and the identity matrix inside Γ accounts for the eigenvalues with value one. In (3.85) we are further partitioning $\mathcal{Z}$ and $\mathcal{Z}^{-1}$ in accordance with the size of J. Using (3.85), it is straightforward to verify that

$$\lim_{i\to\infty} \mathcal{B}^{i+1} = \mathcal{Z}_2 \bar{\mathcal{Z}}_2 \tag{3.86}$$

and if we multiply both sides of (3.76) from the left by $\bar{\mathcal{Z}}_2$, it also follows that

$$\bar{\mathcal{Z}}_2 \mathcal{A}_2^T \mathcal{M} r = 0 \tag{3.87}$$

Using these relations, we can now establish the following result, which describes the limiting behavior of the weight vector estimate.

Theorem 3.2. *If the step-sizes $\{\mu_1, \cdots, \mu_N\}$ satisfy (3.72) and matrix $\mathcal{B}$ is power convergent, then the mean estimate of the network given by (3.77) asymptotically converges to:*

$$\lim_{i\to\infty} \mathbb{E}[\boldsymbol{w}_i] = (\mathcal{Z}_2 \bar{\mathcal{Z}}_2)\, \mathbb{E}[\boldsymbol{w}_{-1}] + (I - \mathcal{B})^{-} \mathcal{A}_2^T \mathcal{M} r \tag{3.88}$$

where the notation X^{-} denotes a (reflexive) generalized inverse for the matrix X. In this case, the generalized inverse for $I - \mathcal{B}$ is given by

$$(I - \mathcal{B})^{-} = \mathcal{Z}_1 (I - J)^{-1} \bar{\mathcal{Z}}_1 \tag{3.89}$$

in terms of the factors $\{\mathcal{Z}_1, \bar{\mathcal{Z}}_1, J\}$ *defined in (3.85).*

Proof. See Appendix A.3. □

We also argue in Appendix A.3 that the quantity on the right-hand side of (3.88) is invariant under basis transformations for the Jordan factors $\{\mathcal{Z}_1, \bar{\mathcal{Z}}_1, \mathcal{Z}_2, \bar{\mathcal{Z}}_2\}$. It can be verified that if $A_1 = A_2 = I$ then $\mathcal{B}$ will be symmetric and the result (3.88) will reduce to (3.75). Now note that the first term on the right hand side of (3.88) is due to the zero eigenvalues of $I - \mathcal{B}$. From this expression, we observe that different initialization values generally lead to different estimates. However, if we set $\mathbb{E}[\boldsymbol{w}_{-1}] = 0$, the algorithm converges to:

$$\lim_{i\to\infty} \mathbb{E}[\boldsymbol{w}_i] = (I - \mathcal{B})^{-}\mathcal{A}_2^T\mathcal{M}r \tag{3.90}$$

In other words, the diffusion LMS algorithm will converge on average to a generalized inverse solution of the linear system of equations defined by (3.76).

When matrix $\mathcal{B}$ is stable so that $\rho(\mathcal{B}) < 1$ then the factorization (3.85) reduces to the form $\mathcal{B} = \mathcal{Z}_1 J \bar{\mathcal{Z}}_1$ and $I - \mathcal{B}$ will be full-rank. In that case, the first term on the right hand side of (3.88) will be zero and the generalized inverse will coincide with the actual matrix inverse so that (3.88) becomes

$$\lim_{i\to\infty} \mathbb{E}[\boldsymbol{w}_i] = (I - \mathcal{B})^{-1}\mathcal{A}_2^T\mathcal{M}r \tag{3.91}$$

Comparing (3.91) with (3.76), we conclude that:

$$\lim_{i\to\infty} \mathbb{E}[\boldsymbol{w}_i] = \mathbb{1} \otimes w^o \tag{3.92}$$

which implies that the mean estimate of each node will be w^o. This result is in agreement with the previously developed mean-convergence analysis of diffusion LMS when the regression data have full rank covariance matrices [82].

3.4.2 Mean-Square Error Convergence

We now examine the mean-square stability of the error recursion (3.66) in the rank-deficient scenario. We begin by deriving an error variance relation as in [89,109]. To find this relation, we form the weighted square "norm" of (3.66), and compute its expectation to obtain:

$$\mathbb{E}\|\tilde{\boldsymbol{w}}_i\|^2_{\Sigma} = \mathbb{E}\Big(\|\tilde{\boldsymbol{w}}_{i-1}\|^2_{\boldsymbol{\Sigma}'}\Big) + \mathbb{E}[\boldsymbol{g}_i^T\mathcal{M}\mathcal{A}_2\Sigma\mathcal{A}_2^T\mathcal{M}\boldsymbol{g}_i] \tag{3.93}$$

where $\|x\|^2_{\Sigma} = x^T\Sigma x$ and $\Sigma \geq 0$ is an arbitrary weighting matrix of compatible dimension that we are free to choose. In this expression,

$$\boldsymbol{\Sigma}' = \mathcal{A}_1(I - \mathcal{M}\boldsymbol{\mathcal{R}}_i)^T\mathcal{A}_2\Sigma\mathcal{A}_2^T(I - \mathcal{M}\boldsymbol{\mathcal{R}}_i)\mathcal{A}_1^T \tag{3.94}$$

Under the temporal and spatial independence conditions on the regression data from Assumption 3.1, we can write:

$$\mathbb{E}\Big(\|\tilde{\boldsymbol{w}}_{i-1}\|^2_{\boldsymbol{\Sigma}'}\Big) = \mathbb{E}\|\tilde{\boldsymbol{w}}_{i-1}\|^2_{\mathbb{E}[\boldsymbol{\Sigma}']} \tag{3.95}$$

so that (3.93) becomes:

$$\mathbb{E}\|\tilde{\boldsymbol{w}}_i\|^2_{\Sigma} = \mathbb{E}\|\tilde{\boldsymbol{w}}_{i-1}\|^2_{\Sigma'} + \mathrm{Tr}[\Sigma\mathcal{A}_2^T\mathcal{M}\mathcal{G}\mathcal{M}\mathcal{A}_2] \tag{3.96}$$

where $\mathcal{G} \triangleq \mathbb{E}[\boldsymbol{g}_i\boldsymbol{g}_i^T]$ is given by

$$\mathcal{G} = \mathcal{C}^T\mathrm{diag}\Big\{\sigma^2_{v,1}\bar{R}_{u,1}, \ldots, \sigma^2_{v,N}\bar{R}_{u,N}\Big\}\mathcal{C} \tag{3.97}$$

and

$$\Sigma' \triangleq \mathbb{E}[\boldsymbol{\Sigma}'] = \mathcal{B}^T\Sigma\mathcal{B} + O(\mathcal{M}^2) \approx \mathcal{B}^T\Sigma\mathcal{B} \tag{3.98}$$

We shall employ (3.98) under the assumption of sufficiently small step-sizes where terms that depend on higher-order powers of the step-sizes are ignored. We next introduce

$$\mathcal{Y} \triangleq \mathcal{A}_2^T\mathcal{M}\mathcal{G}\mathcal{M}\mathcal{A}_2 \tag{3.99}$$

and use (6.11) to write:

$$\mathbb{E}\|\tilde{\boldsymbol{w}}_i\|^2_{\Sigma} = \mathbb{E}\|\tilde{\boldsymbol{w}}_{i-1}\|^2_{\Sigma'} + \mathrm{Tr}(\Sigma\mathcal{Y}) \tag{3.100}$$

From (3.100), we arrive at

$$\mathbb{E}\|\tilde{\boldsymbol{w}}_i\|^2_{\Sigma} = \mathbb{E}\|\tilde{\boldsymbol{w}}_{-1}\|^2_{(\mathcal{B}^T)^{i+1}\Sigma\mathcal{B}^{i+1}} + \sum_{j=0}^{i}\mathrm{Tr}\Big((\mathcal{B}^T)^j\Sigma\mathcal{B}^j\mathcal{Y}\Big) \tag{3.101}$$

To prove the convergence and stability of the algorithm in the mean-square sense, we examine the convergence of the terms on the right hand side of (3.101).

In a manner similar to (3.87), it is shown in Appendix A.4 that the following property holds:

$$\bar{\mathcal{Z}}_2\mathcal{Y} = 0, \qquad \mathcal{Y}\bar{\mathcal{Z}}_2^T = 0 \tag{3.102}$$

Exploiting this result, we can arrive at the following statement, which establishes that relation (3.101) converges as $i \to \infty$ and determines its limiting value.

Theorem 3.3. *Assume the step-sizes are sufficiently small and satisfy (3.72). Assume also that $\mathcal{B}$ is power convergent. Under these conditions, relation (3.101) converges to*

$$\lim_{i\to\infty}\mathbb{E}\|\tilde{\boldsymbol{w}}_i\|^2_{\Sigma} = \mathbb{E}\|\tilde{\boldsymbol{w}}_{-1}\|^2_{(\mathcal{Z}_2\bar{\mathcal{Z}}_2)^T\Sigma\mathcal{Z}_2\bar{\mathcal{Z}}_2} + \Big(\mathrm{vec}(\mathcal{Y})\Big)^T(I-\mathcal{F})^{-1}\mathrm{vec}(\Sigma) \tag{3.103}$$

where

$$\mathcal{F} \triangleq \Big((\mathcal{Z}_1\otimes\mathcal{Z}_1)(J\otimes J)(\bar{\mathcal{Z}}_1\otimes\bar{\mathcal{Z}}_1)\Big)^T \tag{3.104}$$

and factors $\{\mathcal{Z}_1, \bar{\mathcal{Z}}_1, J\}$ are defined in (3.85).

Proof. See Appendix A.4. □

In a manner similar to the proof at the end of Appendix A.3, the term on the right hand side of (3.103) is invariant under basis transformations on the factors $\{\mathcal{Z}_1, \bar{\mathcal{Z}}_1, \mathcal{Z}_2, \bar{\mathcal{Z}}_2\}$. Note that the first term on the right hand side of (3.103) is the network penalty due to rank-deficiency. When the node covariance matrices are full rank, then choosing step-sizes according to (3.72) leads to $\rho(\mathcal{B}) < 1$. When this holds, then $\mathcal{B} = \mathcal{Z}_1 J\bar{\mathcal{Z}}_1$. In this case, the first term on the right hand side of (3.103) will be zero, and $\mathcal{F} = (\mathcal{B}\otimes\mathcal{B})^T$. In this case, we obtain:

$$\lim_{i\to\infty}\mathbb{E}\|\tilde{\boldsymbol{w}}_i\|^2_{\Sigma} = \Big(\mathrm{vec}(\mathcal{Y})\Big)^T(I-\mathcal{F})^{-1}\mathrm{vec}(\Sigma) \tag{3.105}$$

which is in agreement with the mean-square analysis of diffusion LMS strategies for regression data with full rank covariance matrices given in [33, 82].

3.4.3 Learning Curves

For each k, the MSD and EMSE measures are defined as:

$$\eta_k = \lim_{i\to\infty} \mathbb{E}\|\tilde{\boldsymbol{h}}_{k,i}\|^2 = \lim_{i\to\infty} \mathbb{E}\|\tilde{\boldsymbol{w}}_{k,i}\|^2_{B_k^T B_k} \tag{3.106}$$

$$\zeta_k = \lim_{i\to\infty} \mathbb{E}\|\boldsymbol{u}_{k,i}\tilde{\boldsymbol{h}}_{k,i-1}\|^2 = \lim_{i\to\infty} \mathbb{E}\|\tilde{\boldsymbol{w}}_{k,i-1}\|^2_{\bar{R}_{u,k}} \tag{3.107}$$

where $\tilde{\boldsymbol{h}}_{k,i} = h_k^o - \boldsymbol{h}_{k,i}$. These parameters can be computed from the network error vector (3.103) through proper selection of the weighting matrix Σ as follows:

$$\eta_k = \lim_{i\to\infty} \mathbb{E}\|\tilde{\boldsymbol{w}}_i\|^2_{\Sigma_{\text{msd}_k}}, \quad \zeta_k = \lim_{i\to\infty} \mathbb{E}\|\tilde{\boldsymbol{w}}_{i-1}\|^2_{\Sigma_{\text{emse}_k}}, \tag{3.108}$$

where

$$\Sigma_{\text{msd}_k} = \text{diag}(e_k) \otimes (B_k^T B_k),\ \Sigma_{\text{emse}_k} = \text{diag}(e_k) \otimes \bar{R}_{u,k} \tag{3.109}$$

and $\{e_k\}_{k=1}^N$ denote the vectors of a canonical basis set in N dimensional space. The network MSD and EMSE measures are defined as

$$\eta_{\text{net}} = \frac{1}{N}\sum_{k=1}^{N} \eta_k, \qquad \zeta_{\text{net}} = \frac{1}{N}\sum_{k=1}^{N} \zeta_k \tag{3.110}$$

We can also define MSD and EMSE measures over time as

$$\eta_k(i) \;=\; \mathbb{E}\|\tilde{\boldsymbol{h}}_{k,i}\|^2 = \mathbb{E}\|\tilde{\boldsymbol{w}}_i\|^2_{\Sigma_{\text{msd}_k}} \tag{3.111}$$

$$\zeta_k(i) \;=\; \mathbb{E}\|\boldsymbol{u}_{k,i}\tilde{\boldsymbol{h}}_{k,i-1}\|^2 = \mathbb{E}\|\tilde{\boldsymbol{w}}_{i-1}\|^2_{\Sigma_{\text{emse}_k}} \tag{3.112}$$

Using (3.101), it can be verified that these measures evolve according to the following dynamics:

$$\eta_k(i) = \eta_k(i-1) - \|w^o\|_{\mathcal{H}^i(I-\mathcal{H})\sigma_{\text{msd}_k}} + \alpha^T\mathcal{H}^i\sigma_{\text{msd}_k} \tag{3.113}$$

$$\zeta_k(i) = \zeta_k(i-1) - \|w^o\|_{\mathcal{H}^{i-1}(I-\mathcal{H})\sigma_{\text{emse}_k}} + \alpha^T \mathcal{H}^i \sigma_{\text{emse}_k} \tag{3.114}$$

where

$$\begin{aligned} \mathcal{H} &= (\mathcal{B} \otimes \mathcal{B})^T && (3.115) \\ \alpha &= \text{vec}(\mathcal{Y}) && (3.116) \\ \sigma_{\text{msd}_k} &= \text{vec}(\Sigma_{\text{msd}_k}) && (3.117) \\ \sigma_{\text{emse}_k} &= \text{vec}(\Sigma_{\text{emse}_k}) && (3.118) \end{aligned}$$

To obtain (3.113) and (3.114), we set $\mathbb{E}[\boldsymbol{w}_{k,-1}] = 0$ for all k.

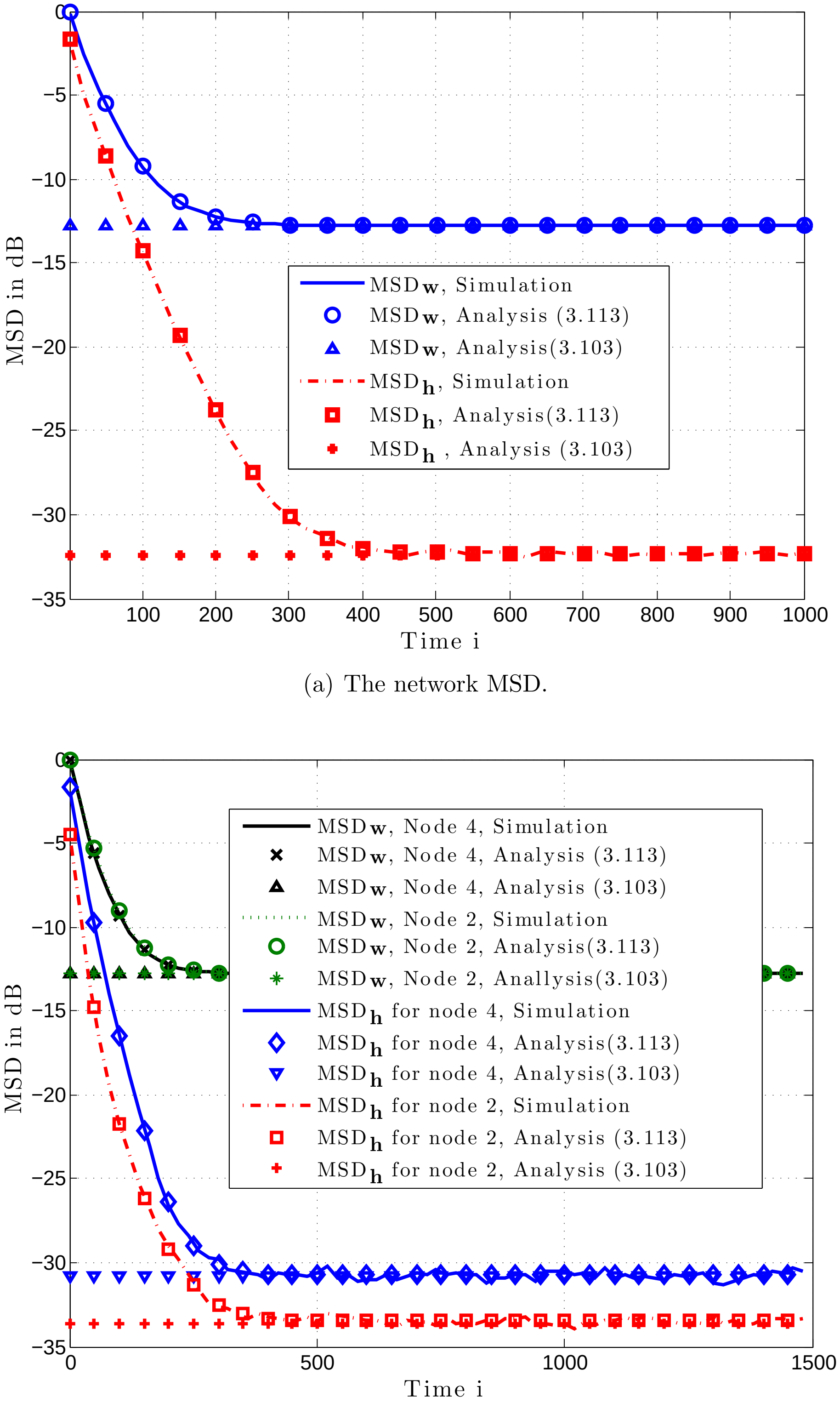
3.5 Computer Experiments

In this section, we examine the performance of the diffusion strategy (3.39)-(3.42) and compare the simulation results with the analytical findings. In addition, we present a simulation example that shows the application of the proposed algorithm in the estimation of space-varying parameters for a physical phenomenon modeled by a PDE system over two spatial dimensions.

3.5.1 Performance of the Distributed Solution

We consider a one-dimensional network topology, illustrated by Fig. 3.1, with $L = 1$ and equally spaced nodes along the x direction. We choose A_1 as the identity matrix, and compute A_2 and C based on the uniform combination and Metropolis rules [33, 82], respectively. We choose $M = 2$ and $N_b = 5$ and generate the unknown global parameter w^o randomly for each experiment. We obtain B_k using the shifted Chebyshev polynomials given by (3.17) and compute the space varying parameters h_k^o according to (3.26). The measurement data $\boldsymbol{d}_k(i)$, $k \in \{1, 2, \cdots, N\}$ are generated using the regression model (3.12). The SNR for each node k is computed as $\text{SNR}_k = \mathbb{E}\|\boldsymbol{u}_{k,i} h_k^o\|^2 / \sigma_{v,k}^2$. The noise and the entries of the regression data are white Gaussian and satisfy Assumption 3.1. The noise variances, $\{\sigma_{v,k}^2\}$, and the trace of the covariance matrices, $\{\text{Tr}(R_{u,k})\}$, are uniformly distributed between $[0.05, 0.1]$ and $[1, 5]$, respectively.

Figure 3.2 illustrates the simulation results for a network with $N = 4$ nodes. For this experiment, we set $\mu_k = 0.01$ for all k and initialize each node at zero. In the legend of the



(a) The network MSD.

(b) The MSD at some individual nodes.

Fig. 3.2 The network MSD learning curve for $N = 4$.

figure, we use the subscript h to denote the MSD for $\tilde{\boldsymbol{h}}_{k,i}$ and the subscript w to refer to the MSD of $\tilde{\boldsymbol{w}}_{k,i}$. The simulation curves are obtained by averaging over 300 independent runs. It can be seen that the simulated and theoretical results match well in all cases. In the figure, we use expression (3.103) to assess the steady-state values and expression (3.113) to generate the theoretical learning curves.

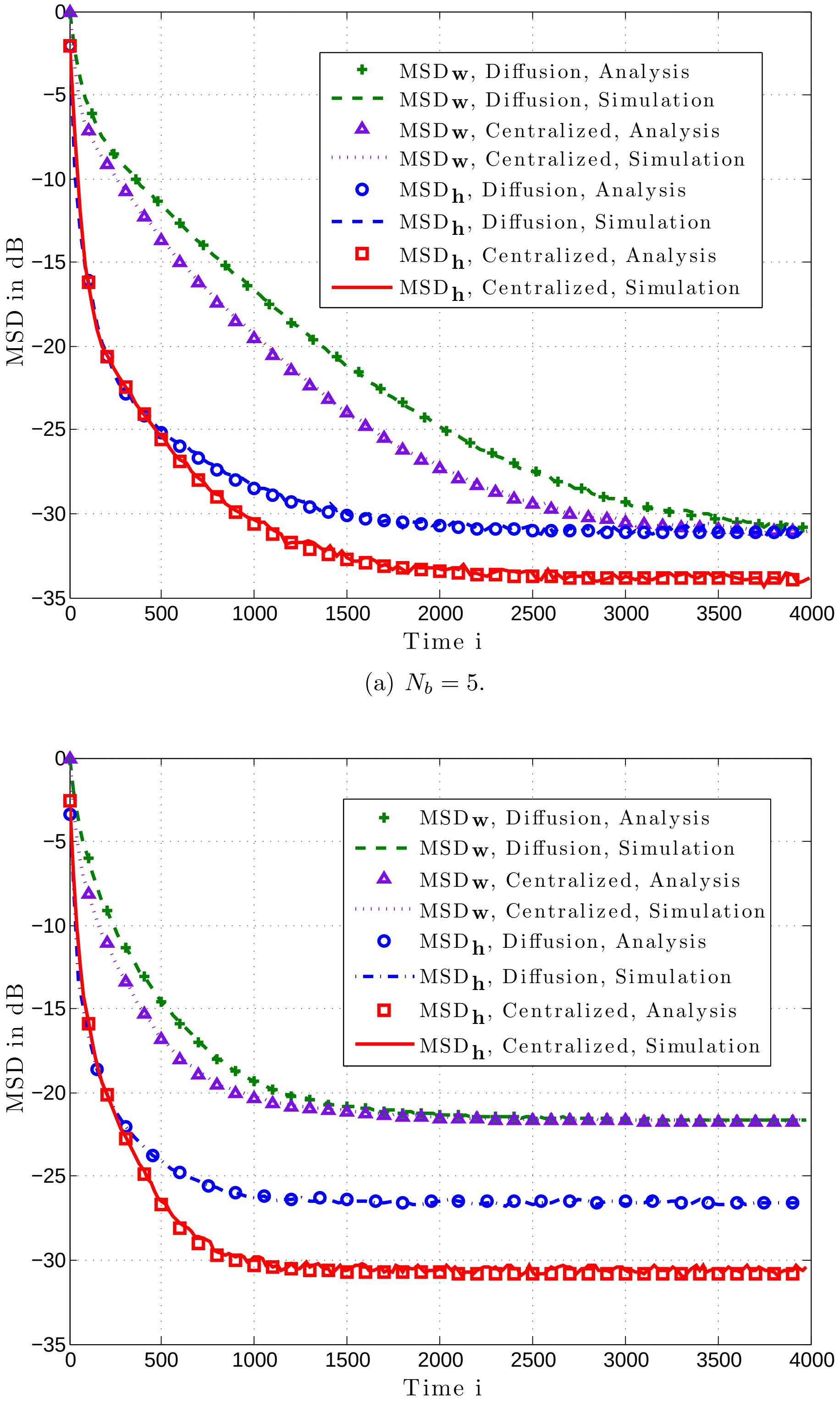
Two important points in Fig. 3.2 need to be highlighted. First, note from Fig. 3.2(a) that the network MSD for $\tilde{\boldsymbol{w}}_{k,i}$ is larger than that for $\tilde{\boldsymbol{h}}_{k,i}$. This is because

$$\mathbb{E}\|\tilde{\boldsymbol{h}}_{k,i}\|^2 = \mathbb{E}\|\tilde{\boldsymbol{w}}_{k,i}\|^2_{B_k^T B_k} \tag{3.119}$$

so that the MSD of $\tilde{\boldsymbol{h}}_{k,i}$ is a weighted version of the MSD of $\tilde{\boldsymbol{w}}_{k,i}$. In this experiment, the weighting leads to a lower estimation error. Second, note from Fig. 3.2(b) that while the MSD values of $\tilde{\boldsymbol{w}}_{k,i}$ are largely independent of the node index, the same is not true for the MSD values of $\tilde{\boldsymbol{h}}_{k,i}$. In previous studies on diffusion LMS strategies, it has been shown that, for strongly-connected networks, the network nodes approach a uniform MSD performance level [107]. The result in Fig. 3.2(b) supports this conclusion where it is seen that the MSD of $\tilde{\boldsymbol{w}}_{k,i}$ for nodes 2 and 4 converge to the same MSD level. However, note that the MSD of $\tilde{\boldsymbol{h}}_{k,i}$ is different for nodes 2 and 4. This difference in behavior is due to the difference in weighting across nodes from (3.119).

3.5.2 Comparison with Centralized Solution

We next compare the performance of the diffusion strategy (3.39)-(3.42) with the centralized solution (3.34)–(3.35). We consider a network with $N = 10$ nodes with the topology illustrated by Fig. 3.1. In this experiment, we set $\mu_k = 0.02$ for all k, while the other network parameters are obtained following the same construction described for Fig. 3.2. As the results in Fig. 3.3 indicate, the diffusion and centralized LMS solutions tend to the same MSD performance level in the w domain. This conclusion is consistent with prior studies on the performance of diffusion strategies in the full-rank case over strongly-connected networks [107]. However, discrepancies in performance are seen between the distributed and centralized implementations in the h domain, and the discrepancy tends to become larger for larger values of N. This is because, in moving from the w domain to the h domain, the inherent aggregation of information that is performed by the centralized solution leads to enhanced estimates for the h_k^o variables. For example, if the estimates



(a) $N_b = 5$.

(b) $N_b = 10$.

Fig. 3.3 The network MSD learning curve for $N = 10$.

$\boldsymbol{w}_{k,i}$ which are generated by the distributed solution are averaged prior to computing the $\boldsymbol{h}_{k,i}$, then it can be observed that the MSD values of $\tilde{\boldsymbol{h}}_{k,i}$ for both the centralized and the distributed solution will be similar.

In these experiments, we also observe that if we increase the number of basis functions, N_b, then both the centralized and diffusion algorithms will converge faster but their steady-state MSD performance will degrade. Therefore, in choosing the number of basis functions, N_b, there is a trade off between convergence speed and MSD performance.

3.5.3 Example: Two-Dimensional Process Estimation

In this example, we consider a two-dimensional network with 13×13 nodes that are equally spaced over the unit square $(x, y) \in [0, 1] \times [0, 1]$ with $\Delta x = \Delta y = 1/12$ (see Fig. 3.4(a)). This network monitors a physical process $f(x, y)$ described by the Poisson PDE:

$$\frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2} = h(x,y) \tag{3.120}$$

where $h(x, y) : [0,\ 1]^2 \rightarrow \mathbb{R}$ is an unknown input function. The PDE satisfies the following boundary conditions:

$$f(x,0) = f(0,y) = f(x,1) = f(1,y) = 0$$

For this problem, the objective is to estimate $h(x, y)$, given noisy measurements collected by $N = N_x \times N_y = 11 \times 11$ nodes corresponding to the *interior points* of the network. To discretize the PDE, we employ the finite difference method (FDM) with uniform spacing of Δx and Δy. We define $x_{k_1} \triangleq k_1 \Delta x$, $y_{k_2} \triangleq k_2 \Delta y$ and introduce the sampled values $f_{k_1,k_2} \triangleq f(x_{k_1}, y_{k_2})$ and $h^o_{k_1,k_2} \triangleq h(x_{k_1}, y_{k_2})$. We use the central difference scheme [101] to approximate the second order partial derivatives:

$$\frac{\partial^2 f(x,y,t)}{\partial x^2} \approx \frac{1}{\Delta x^2}[f_{k_1+1,k_2} - 2f_{k_1,k_2} + f_{k_1-1,k_2}] \tag{3.121}$$

$$\frac{\partial^2 f(x,y,t)}{\partial y^2} \approx \frac{1}{\Delta y^2}[f_{k_1,k_2+1} - 2f_{k_1,k_2} + f_{k_1,k_2-1}] \tag{3.122}$$

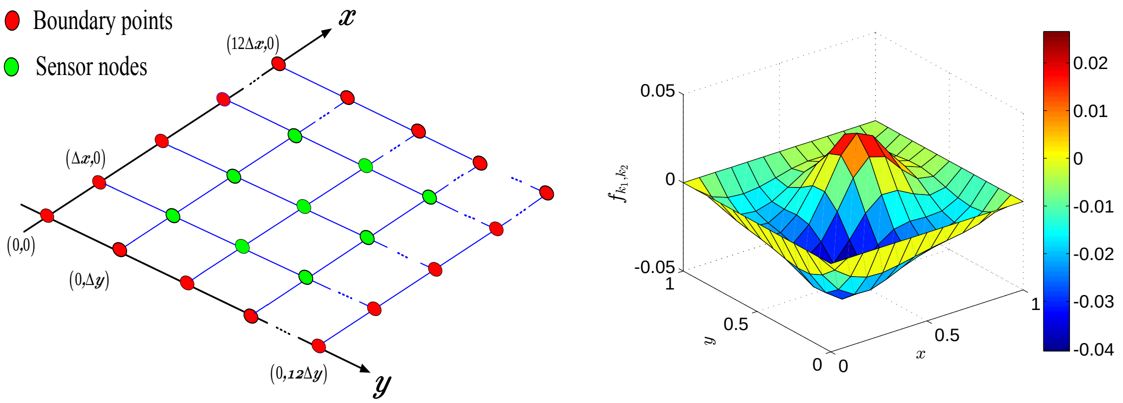
This leads to the following discretized input function:

$$h^o_{k_1,k_2} = \frac{1}{\Delta x^2}\Big(f_{k_1+1,k_2} + f_{k_1,k_2+1} + f_{k_1-1,k_2} \\ + f_{k_1,k_2-1} - 4f_{k_1,k_2}\Big) \tag{3.123}$$

For this example, the unknown input process is

$$h^o_{k_1,k_2} = e^{-\kappa\Big((k_1-4)^2+(k_2-4)^2\Big)} - 5e^{-\kappa\Big((k_1-8)^2+(k_2-8)^2\Big)} + 1 \tag{3.124}$$

where $\kappa = (N_x - 1)^2/4$.



(a) Network topology.

(b) f_{k_1,k_2} over the space.

Fig. 3.4 Spatial distribution of $f(x,y)$ over the network grid $\{(x_{k_1}, y_{k_2})\}$.

To obtain f_{k_1,k_2}, we solve (3.120) using the Jaccobi over-relaxation method [79]. Figure 3.4(b) illustrates the values of f_{k_1,k_2} over the spatial domain. For the estimation of h_{k_1,k_2}, the given information are the noisy measurement samples $\boldsymbol{z}_{k_1,k_2}(i) = f_{k_1,k_2} + \boldsymbol{n}_{k_1,k_2}(i)$. In this relation, the noise process $\boldsymbol{n}_{k_1,k_2}(i)$ is zero mean, temporally white and independent over space. For this network, the two dimensional reference signal is the distorted version of $h^o_{k_1,k_2}$ which is represented by $\boldsymbol{d}_{k_1,k_2}(i)$. The reference signal is obtained from (3.123) with $\{f_{k_1,k_2}\}$ replaced by their noisy measured samples $\boldsymbol{z}_{k_1,k_2}(i)$, i.e.,

$$\boldsymbol{d}_{k_1,k_2}(i) = \frac{1}{\Delta x^2}\Big(\boldsymbol{z}_{k_1+1,k_2}(i) + \boldsymbol{z}_{k_1,k_2+1}(i) + \boldsymbol{z}_{k_1-1,k_2}(i) + \boldsymbol{z}_{k_1,k_2-1}(i) - 4\boldsymbol{z}_{k_1,k_2}(i)\Big) \tag{3.125}$$

According to (3.125), the linear regression model for this problem takes the following form:

$$\boldsymbol{d}_{k_1,k_2}(i) = \boldsymbol{u}_{k_1,k_2}(i) h^o_{k_1,k_2} + \boldsymbol{v}_{k_1,k_2}(i) \tag{3.126}$$

where $\boldsymbol{u}_{k_1,k_2}(i) = 1$. Therefore, in this example, we are led to a linear model (3.126) with *deterministic* as opposed to random regression data. Although we only studied the case of random regression data, this example is meant to illustrate that the diffusion strategy can still be applied to models involving deterministic data in a manner similar to [9, 37].

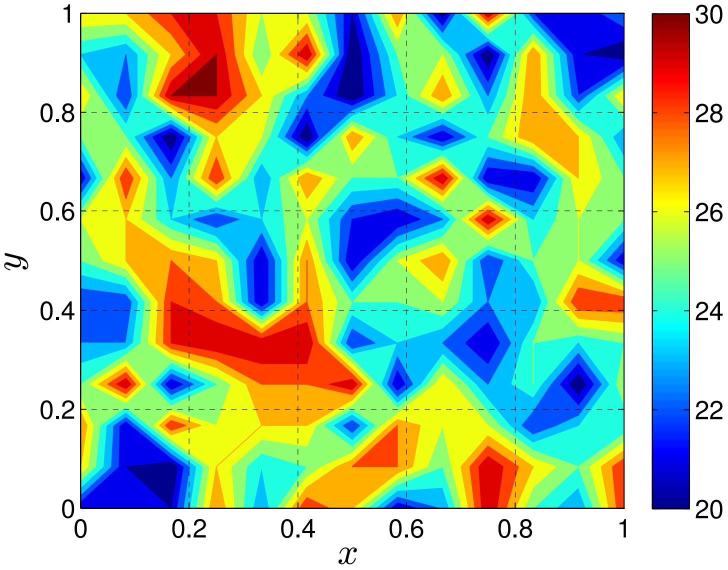


Fig. 3.5 Spatial distribution of SNR over the network.

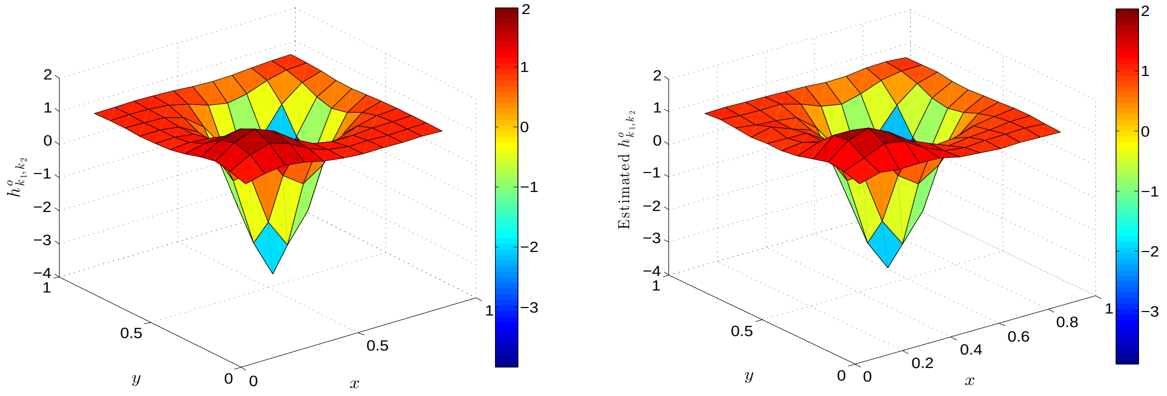
To represent $h^o_{k_1,k_2}$ as a space-invariant parameter vector, we use two-dimensional shifted Chebyshev basis functions [110]. Using this representation, $h^o_{k_1,k_2}$ can be expressed as:

$$h^o_{k_1,k_2} = \sum_{n=1}^{N_b} w^o_n \, p_{n,k_1,k_2} \tag{3.127}$$

where each element of the two-dimensional basis set is given by:

$$p_{n,k_1,k_2} = b_{n_1,k_1} b_{n_2,k_2} \tag{3.128}$$

where $\{b_{n_1,k_1}\}$ and $\{b_{n_2,k_2}\}$ are the one-dimensional shifted Chebyshev polynomials in the x and y directions, respectively–recall (3.21).



(a) True parameters.

(b) Estimated parameters.

Fig. 3.6 True and estimated $h^o_{k_1,k_2}$ by diffusion LMS.

In the network, each interior node communicates with its four immediate neighbors. We use $A_1 = I$ and compute C and A_2 by using the Metropolis and relative degree rules [27, 33, 82]. All nodes are initialized at zero and $\mu_k = 0.01$ for all k. The signal-to-noise ratio (SNR) of the network is uniformly distributed in the range $[20, 30]$dB and is shown in Fig. 3.5.

Figures 3.6(a) and 3.6(b) show three dimensional views of the true and estimated input process using the proposed diffusion LMS algorithm after 3000 iterations. Figure 3.7 illustrates the MSD of the estimated source, i.e., $\lim_{i\to\infty}\mathbb{E}\|h^o_{k_1,k_2} - \boldsymbol{h}_{k_1,k_2}(i)\|^2$.

3.6 Summary

By combining interpolation and distributed adaptive optimization, we proposed a diffusion LMS strategy for estimation and tracking of space-time varying parameters over networks. The proposed algorithm can find the space-varying parameters not only at the node locations but also at spaces where no measurement is collected. We showed that if the network experiences data with rank-deficient covariance matrices, the non-cooperative LMS algorithm will converge to different solutions at different nodes. In contrast, the diffusion LMS algorithm is able to alleviate the rank-deficiency problem through its use of combination

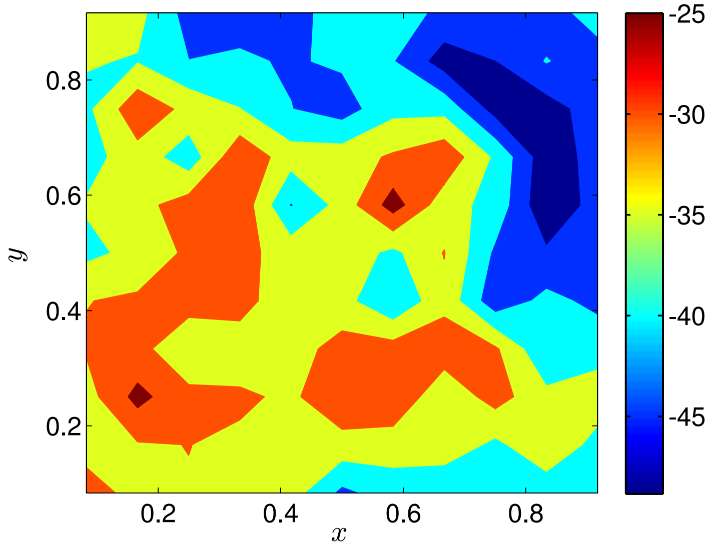


Fig. 3.7 Network steady-state MSD performance in dB.

matrices especially since, as shown by (3.71), $\rho(\mathcal{B}) \leq \rho(I - \mathcal{M}\mathcal{R})$, where $I - \mathcal{M}\mathcal{R}$ is the coefficient matrix that governs the dynamics of the non-cooperative solution. Nevertheless, if these mechanisms fail to mitigate the deleterious effect of the rank-deficient data, then the algorithm converges to a solution space where the error is bounded. We analyzed the performance of the algorithm in transient and steady-state regimes, and gave conditions under which the algorithm is stable in the mean and mean-square sense.

In the next chapter, we study the performance of DLMS algorithms in sensor networks where the input regression data of the underlying physical phenomenon is unavailable beforehand and each agent use its noisy measured version to retrieve the unknown parameter of interest.

Chapter 4

Bias-Compensated DLMS Algorithms

In this chapter, we investigate the performance of DLMS algorithms for parameter estimation over sensor networks where the measured regression data at each node are corrupted by additive noise[1]. We show that the estimates produced by standard DLMS algorithms will be biased if the regression noises over the network are overlooked in the estimation process. To resolve this problem, we propose a bias-elimination technique, in which we add a correction term to the mean-square error function of the network to be optimized, and develop new DLMS algorithms that can mitigate the effect of regression noise and obtain an unbiased estimates of the unknown parameters over the network. In our development, we first assume that the variances of the regression noises are known *a-priori*. Later, we relax this assumption by estimating these variances in real-time. We analyze the stability and convergence of the proposed algorithms and derive closed-form expressions to characterize their mean-square error performance in transient and steady-state regimes. We further provide computer experiment results that show the efficiency of the proposed algorithms and verify the analytical findings.

[1] Part of the work presented in this chapter has led to the following manuscript and conference paper:

- R. Abdolee, B. Champagne, "Diffusion LMS strategies in sensor networks with noisy input data applications", submitted to *IEEE/ACM Trans. on Networking*", Feb. 2014.
- R. Abdolee, B. Champagne and A. H. Sayed, "A diffusion LMS Strategy for parameter estimation in noisy regressor applications", in *Proc. of the 20th European Signal Processing Conf. (EUSIPCO)*, Bucharest, Romania, Aug. 2012, pp. 749–753.

4.1 Introduction

One of the critical issues encountered in distributed parameter estimation over sensor networks is the distortion of the collected regression data by noise, which occurs when the local copy of the underlying system input signal at each node is corrupted by various sources of impairments such as measurement or quantization noise. This problem has been extensively investigated for the case of single-node processing devices [44–50, 111–117]. These studies have shown that if the deleterious effect of the input noise is not taken into account, the parameter estimates so obtained will be inaccurate and biased. Various practical solutions have been suggested to mitigate the effect of the input measurement noise or to remove the bias from the resulting estimates [44–50, 114–117]. These solutions, however, may no longer leads to optimal results in sensor networks with decentralized processing structure where the data measurement and parameter estimation are performed at multiple processing nodes in parallel and with cooperation.

For networking applications, a distributed total-least-squares (DTLS) algorithm has been proposed that is developed using semidefinite relaxation and convex semidefinite programming [51]. This algorithm mitigates the effect of white input noise by running a local TLS algorithm at each sensor node and exchanging the locally estimated parameters between the nodes for further refinement. The DTLS algorithm computes the eigendecomposition of an augmented covariance matrix at every iteration for all nodes in the network, and is therefore mainly suitable for applications involving nodes with powerful processing abilities. In a follow up paper, the authors proposed a low-complexity DTLS algorithm [118] that uses an inverse power iteration technique to reduce the computational complexity of the DTLS while demanding lower communication power. In the class of distributed *adaptive* algorithms, a bias-compensated diffusion-based recursive least-squares (RLS) algorithm has been developed in [52] that can obtain unbiased estimates of the unknown system parameters over sensor networks, where the regression data are distorted by colored noise. While this algorithm offers fast convergence speed, its high computational complexity and numerical instability may be a hindrance in some applications.

In contrast, the DLMS algorithms are characterized by low complexity and numerical stability. Motivated by these features, in this chapter, we investigate the performance of standard DLMS algorithms [27,33,37] over sensor networks where the input regression data are corrupted by additive white noise. To overcome the limitations of these algorithms, as exposed by our analysis for this scenario, we then propose an alternative problem formulation that leads to a novel class of DLMS algorithms, which we call bias-compensated diffusion strategies.

More specifically, we first show that in the presence of noisy input data, the parameter estimates produced by standard DLMS algorithms are biased. We then reformulate this estimation problem in terms of an alternative cost function and develop bias-compensated DLMS strategies that can produce unbiased estimates of the system parameters. The development of these algorithms relies on a bias-elimination strategy that assumes prior knowledge about the regression noise variances over the network. We then relax the known variance assumption by incorporating a recursive approach into the algorithm to estimate the variances in real-time. We analyze the stability and convergence of the proposed algorithms and derive closed-form expressions to characterize their MSD and EMSE. The analysis results show that if the step-sizes are within a given range, the algorithms will be stable in the mean and mean-square sense and the estimated parameters will converge to their true values.

It is worth noting that the problem addressed in this chapter is different from the one investigated in [53] for which the authors have studied DLMS algorithms under the imperfect information exchange. In their work, it was assumed that the regression data as well as the estimated parameters which are exchanged between nodes are corrupted with communication noise. Under such conditions, the regression data of the nodes themselves were assumed to be error-free. The authors have then proposed a recursive estimation approach to minimize the noise impact by appropriately computing the entries of the left-stochastic combination matrix of the network in real-time. Although this technique can reduce the bias caused by the communication noise, it cannot remove the effect of measurement noise in the regression data and therefore their estimates may still remain biased.

4.2 Problem Statement

We consider a collection of N nodes that are distributed over a geographical area to monitor a physical phenomenon characterized by parameter vector $w^o \in \mathbb{C}^{M\times 1}$. As illustrated in Fig. 4.1, at discrete-time i, each node k collects noisy samples of the system input and output, respectively, denoted by $\boldsymbol{z}_{k,i} \in \mathbb{C}^{1\times M}$ and $\boldsymbol{d}_k(i) \in \mathbb{C}$. These measurement samples satisfy the following relations:

$$\boldsymbol{z}_{k,i} = \boldsymbol{u}_{k,i} + \boldsymbol{n}_{k,i} \tag{4.1}$$

$$\boldsymbol{d}_k(i) = \boldsymbol{u}_{k,i} w^o + \boldsymbol{v}_k(i) \tag{4.2}$$

where $\boldsymbol{u}_{k,i} \in \mathbb{C}^{1\times M}$, $\boldsymbol{n}_{k,i} \in \mathbb{C}^{1\times M}$, and $\boldsymbol{v}_k(i) \in \mathbb{C}$, respectively, denote the regression data vector, the input measurement noise vector, and the output measurement noise.

Assumption 4.1. *The random variables in data model (4.1)-(4.2) are assumed to satisfy the following conditions:*

a) The regression data vectors are independent and identically distributed (i.i.d.) over time and independent over space, with zero-mean and positive definite covariance matrix $R_{u,k}$.

b) The regression noise vectors $\boldsymbol{n}_{k,i}$ are Gaussian, i.i.d. over time and independent over space, with zero-mean and covariance matrix $R_{n,k} = \mathbb{E}[\boldsymbol{n}_{k,i}^* \boldsymbol{n}_{k,i}] = \sigma_{n,k}^2 I$.

c) The output noise samples $\boldsymbol{v}_k(i)$ are i.i.d. over time and independent over space, with zero-mean and variance $\sigma_{v,k}^2$.

d) The random processes $\boldsymbol{u}_{k,i}$, $\boldsymbol{n}_{\ell,j}$ and $\boldsymbol{v}_p(m)$ are independent for all $k, \ell, p, i, j,$ and m.

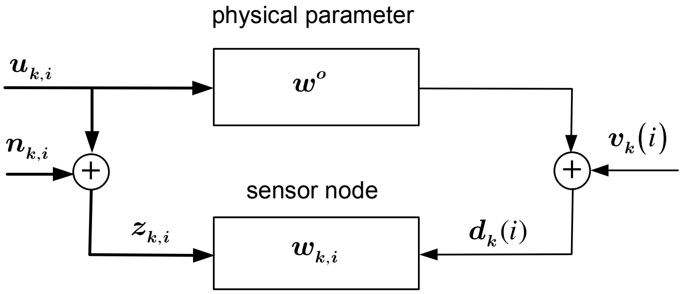


Fig. 4.1 Measurement model for node k.

We use relation (4.1) to model the disturbance in the regressors at each node, and investigate the effect of the noise process $\boldsymbol{n}_{k,i}$ on the distributed estimation of w^o. To better understand the effect of this noise, we first evaluate the performance of a centralized estimation solution under this condition and then explain how the resulting effect carries over to distributed approaches.

In centralized estimation, nodes transmit their measurement data $\{\boldsymbol{z}_{k,i}, \boldsymbol{d}_k(i)\}_{k=1}^N$ to a central processing unit. In the absence of measurement noise, i.e. when $\boldsymbol{n}_{k,i} = 0_M$, the central processor estimates the unknown parameter vector w^o by minimizing the following mean-square error function [89]:

$$J_u(w) = \sum_{k=1}^{N} \mathbb{E}|\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}w|^2 \tag{4.3}$$

Let us introduce $r_{du,k} \triangleq \mathbb{E}[\boldsymbol{u}_{k,i}^* \boldsymbol{d}_k(i)]$ and denote the sums of covariance matrices and cross-covariance vectors over the set of nodes by:

$$R_u = \sum_{k=1}^{N} R_{u,k}, \quad r_{du} = \sum_{k=1}^{N} r_{du,k} \tag{4.4}$$

It can be verified that under Assumption 4.1 the solution of (4.3) will be:

$$w^o = R_u^{-1} r_{du} \tag{4.5}$$

Let us now examine the recovery of w^o for the noisy regression system described by (4.1) and (4.2). Since the regression noise $\boldsymbol{n}_{k,i}$ is independent of $\boldsymbol{u}_{k,i}$ and $\boldsymbol{d}_k(i)$, we have

$$R_{z,k} \triangleq \mathbb{E}[\boldsymbol{z}_{k,i}^* \boldsymbol{z}_{k,i}] = R_{u,k} + \sigma_{n,k}^2 I \tag{4.6}$$

$$r_{dz,k} \triangleq \mathbb{E}[\boldsymbol{z}_{k,i}^* \boldsymbol{d}_k(i)] = r_{du,k} \tag{4.7}$$

Considering these relations and now minimizing the global mean-square error (MSE) function

$$J_z(w) = \sum_{k=1}^{N} \mathbb{E}|\boldsymbol{d}_k(i) - \boldsymbol{z}_{k,i}w|^2 \tag{4.8}$$

with $\boldsymbol{u}_{k,i}$ in (4.3) replaced by $\boldsymbol{z}_{k,i}$ in (4.8), we arrive at the biased solution

$$w^b = \left(R_u + \sigma_n^2 I\right)^{-1} r_{du} \tag{4.9}$$

where

$$\sigma_n^2 = \sum_{k=1}^{N} \sigma_{n,k}^2 \tag{4.10}$$

Let us define the bias implicit in solution (4.9) as $b = w^o - w^b$. To evaluate b, we may use the following identity, which holds for square matrices X_1 and X_2 provided that X_1 and $X_1 + X_2$ are both invertible [119]:

$$(X_1 + X_2)^{-1} = X_1^{-1} - (I + X_1^{-1}X_2)^{-1}X_1^{-1}X_2X_1^{-1} \tag{4.11}$$

Here R_u and $(R_u + \sigma_n^2\, I)$ are invertible, and therefore, we obtain:

$$\Big(R_u + \sigma_n^2\, I\Big)^{-1} = R_u^{-1} - \sigma_n^2(I + \sigma_n^2 R_u^{-1})^{-1}R_u^{-2} \tag{4.12}$$

Considering this expression and relation (4.9), the bias resulting from the MMSE estimation at the fusion center can be expressed as:

$$b = \sigma_n^2(I + \sigma_n^2 R_u^{-1})^{-1}R_u^{-1}w^o \tag{4.13}$$

In noisy regression applications, the estimates generated by DLMS algorithms, which are developed based on the global cost (4.8), will approach (4.9). As we showed in (4.13), this solution is biased and deviates from the optimal estimate by b. Consequently, the effect of measurement noise on the regression data will carry over to distributed algorithms. This will become more explicit in the mean convergence analysis of DLMS algorithms in this chapter.

In the next section, we explain how by forming a suitable objective function, the bias can be compensated in both centralized and distributed LMS implementations.

4.3 Bias-Compensated Adaptive LMS Algorithms

In our development, we initially assume that the regression noise variances, $\{\sigma_{n,k}^2\}_{k=1}^N$, are known *a-priori*. We later remove this assumption by estimating these variances in real-time.

In networks with centralized signal processing structure, one way to obtain the unbiased optimal solution (4.5) is to search for a global cost function whose gradient vector is identical to that of cost (4.3). It can be verified that the following global cost function satisfies this requirement:

$$J(w) = \left(\sum_{k=1}^{N} \mathbb{E}|\boldsymbol{d}_k(i) - \boldsymbol{z}_{k,i}w|^2\right) - \left(\sum_{k=1}^{N} \sigma_{n,k}^2 \|w\|^2\right) \tag{4.14}$$

The derivation of distributed algorithms will be made easier if we can decouple the network global cost function and write it as sum of local cost functions that are formed using the local data. The global cost (4.14) already has such a desired form. For this to become more explicit, we express (4.14) as:

$$J(w) = \sum_{k=1}^{N} J_k(w) \tag{4.15}$$

where $J_k(w)$, the cost function associated with node k, is given by:

$$J_k(w) = \mathbb{E}|\boldsymbol{d}_k(i) - \boldsymbol{z}_{k,i}w|^2 - \sigma_{n,k}^2 \|w\|^2 \tag{4.16}$$

Remark 4.1. *Under Assumption 4.1, the Hessian matrix of (4.16) is positive definite, i.e., $\nabla_w^2 J_k(w) > 0$, hence, $J(w)$ is strongly convex [120].*

Below, we first comment on the centralized LMS algorithm that solves (4.14). We then elaborate on how to develop unbiased distributed counterparts.

4.3.1 Bias-Compensated Centralized LMS Algorithm

To minimize (4.15) iteratively, a centralized steepest descent algorithm [89] can be implemented as:

$$\boldsymbol{w}_i = \boldsymbol{w}_{i-1} - \mu \left[\sum_{k=1}^{N} \nabla J_k(\boldsymbol{w}_{i-1})\right]^* \tag{4.17}$$

where $\mu > 0$ is the step-size, and $\nabla J_k(w)$ is a row vector representing the gradient of J_k with respect to the vector w. Computing the gradient vectors from (4.16) leads to:

$$\boldsymbol{w}_i = \boldsymbol{w}_{i-1} + \mu \sum_{k=1}^{N} \left(r_{dz,k} - R_{z,k}\boldsymbol{w}_{i-1} + \sigma_{n,k}^2 \boldsymbol{w}_{i-1} \right) \tag{4.18}$$

In practice, the moments $R_{z,k}$ and $r_{dz,k}$ are usually unavailable. We therefore replace these moments by their instantaneous approximations $\boldsymbol{z}_{k,i}^{*}\boldsymbol{z}_{k,i}$ and $\boldsymbol{z}_{k,i}^{*}\boldsymbol{d}_k(i)$, respectively, and obtain the bias-compensated centralized LMS algorithm:

$$\boldsymbol{w}_i = \boldsymbol{w}_{i-1} + \mu \sum_{k=1}^{N} \left(\boldsymbol{z}_{k,i}^{*}[\boldsymbol{d}_k(i) - \boldsymbol{z}_{k,i}\boldsymbol{w}_{i-1} + \sigma_{n,k}^{2}\boldsymbol{w}_{i-1}] \right) \tag{4.19}$$

It will be explained in Subsection 4.3.3 that the known variance assumption can be relaxed by incorporating a real-time adaptive estimation approach within the algorithm.

4.3.2 Bias-Compensated Diffusion LMS Strategies

There are different distributed optimization techniques that can be applied on (4.14) to find w^o [27, 33, 79]. We concentrate on diffusion strategies [27, 33] because they endow the network with real-time adaptation and learning abilities. In particular, diffusion optimization strategies lead to distributed algorithms that can estimate the optimal parameter vector w^o and track its changes over time [27, 29, 33, 82]. Here, we briefly explain how diffusion LMS algorithms can be developed for parameter estimation in systems with noisy regression data. One main step [27, 29, 33, 82] in the development of these algorithms is to reformulate the global cost (4.14) and represent it as a group of local optimization problems of the form:

$$\min_{w} \left\{ \sum_{\ell \in \mathcal{N}_k} c_{\ell,k} \left(\mathbb{E}|\boldsymbol{d}_\ell(i) - \boldsymbol{z}_{\ell,i}w|^2 - \sigma_{n,\ell}^{2}\|w\|^2 \right) + \sum_{\ell \in \mathcal{N}_k \backslash \{k\}} b_{\ell,k}\|w - w^o\|^2 \right\} \tag{4.20}$$

The nonnegative scalars $\{c_{\ell,k}\}$ are the entries of a right-stochastic matrix $C \in \mathbb{R}^{N \times N}$ which as before satisfy (2.30). The scalars $\{b_{\ell,k}\}$ are scaling coefficients that end up being incorporated into the combination coefficients $\{a_{\ell,k}\}$ that appear in the final statement (4.22) of the algorithm below. The first term in the objective function (4.20) is the modified mean-squared function incorporating the noise variances of neighboring nodes $\ell \in \mathcal{N}_k$. This part of the objective is based on the same strategy as in the above centralized objective function for bias removal. The second term in (4.20) is in fact a constraint that forces the estimate of the node k to be aligned with the true parameter vector w^o. Since w^o is not known initially, it will be alternatively substituted by an appropriate vector during the optimization process.

One can use the cost function (4.20) and follow similar arguments to those used in Chapter 2 to arrive at the bias-compensated ATC DLMS strategy (Algorithm 4.1) .

Algorithm 4.1 : ATC Bias-Compensated Diffusion LMS

$$\boldsymbol{\psi}_{k,i} = \boldsymbol{w}_{k,i-1} - \mu_k \sum_{\ell \in \mathcal{N}_k} c_{\ell,k} \big[\widehat{\nabla J_\ell}(\boldsymbol{w}_{k,i-1})\big]^* \tag{4.21}$$

$$\boldsymbol{w}_{k,i} = \sum_{\ell \in \mathcal{N}_k} a_{\ell,k} \boldsymbol{\psi}_{\ell,i} \tag{4.22}$$

In this algorithm, $\mu_k > 0$ is the step-size at node k, the vectors $\boldsymbol{\psi}_k$ and $\boldsymbol{w}_{k,i}$ are the intermediate estimates of w^o at node k, and the stochastic gradient vector is computed as:

$$\Big[\widehat{\nabla J_\ell}(\boldsymbol{w}_{k,i-1})\Big]^* = -\bigg[\boldsymbol{z}^*_{\ell,i}\Big(\boldsymbol{d}_\ell(i) - \boldsymbol{z}_{\ell,i}\boldsymbol{w}_{k,i-1}\Big) + \sigma^2_{n,\ell}\boldsymbol{w}_{k,i-1}\bigg] \tag{4.23}$$

Moreover, the nonnegative coefficients $a_{\ell,k}$ are the elements of a left-stochastic matrix $A \in \mathbb{R}^{N\times N}$ satisfying (2.20) as before. To run the algorithm, we only need to select the coefficients $\{c_{\ell,k}, a_{\ell,k}\}$, which can be computed based on any combination rules that satisfy (2.30) and (2.20). Some of these combination rules are presented in [33, 82]. For example, one choice to compute the entries of matrix A is:

$$a_{\ell,k} = \frac{\sigma^{-2}_{n,\ell}}{\sum_{\ell\in\mathcal{N}_k}\sigma^{-2}_{n,\ell}} \quad \text{and} \quad a_{k,k} = 1 - \sum_{\ell\in\mathcal{N}_k\backslash k} a_{\ell,k} \tag{4.24}$$

This rule implies that the entry $a_{\ell,k}$ is inversely proportional to the regressor noise variance of node ℓ. Other left-stochastic choices for A are possible, including those that take into account both the noise variances and the degree of connectivity of the nodes [31].

As explained in Chapter 2, by reversing the order of the adaptation and combination steps in Algorithm 4.1, we can obtain the following CTA diffusion strategy. As we will show in the analysis, the proposed ATC and CTA bias-compensated DLMS, in average, will converge to the unbiased solution w^o even when the regression data are corrupted by noise. In comparison, the estimate of the previous DLMS strategies such as one proposed in [33] will be biased under such condition.

Algorithm 4.2 : CTA Bias-Compensated Diffusion LMS

$$\boldsymbol{\psi}_{k,i-1} = \sum_{\ell \in \mathcal{N}_k} a_{\ell,k} \boldsymbol{w}_{\ell,i-1} \tag{4.25}$$

$$\boldsymbol{w}_{k,i} = \boldsymbol{\psi}_{k,i-1} - \mu_k \sum_{\ell \in \mathcal{N}_k} c_{\ell,k} \left[\widehat{\nabla J_\ell}(\boldsymbol{\psi}_{k,i-1}) \right]^* \tag{4.26}$$

4.3.3 Regression Noise Variance Estimation

In the proposed algorithms, each node still needs to have the regression noise variances, $\{\sigma^2_{n,\ell}\}_{k=1}^{\mathcal{N}_k}$, to evaluate the stochastic gradient vector, $\widehat{\nabla J_\ell}$. In practice, such information is rarely available and normally obtained through estimation. A review of previous works reveals that the regression noise variances can be either estimated off-line [52], or in real-time when the unknown parameter vector, w^o, is being estimated [121, 122]. For example, in the context of speech analysis, they can be estimated off-line during silent periods in between words and sentences [52]. In some other applications, these variances are estimated during the operation of the algorithm using the second-order moments of the regression data and the system output signal [121, 122]. In what follows we propose an adaptive recursive approach to estimate the regression noise variances without using the second order moments of the data.

The variance of the regression noise at each node is classified as local information and, hence, it can be estimated from the node's local data. When the regression data at node k is not corrupted by measurement noise (i.e., $\boldsymbol{z}_{k,i} = \boldsymbol{u}_{k,i}$), and when the node operates independent of all other nodes to estimate w^o by minimizing $\mathbb{E}|\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i} w|^2$, the minimum attainable MSE can be expressed as [89]:

$$J_{\min} \triangleq \sigma^2_{d,k} - r^*_{du,k} R^{-1}_{u,k} r_{du,k}. \tag{4.27}$$

Under noisy regression scenarios where node k operates independently to minimize the cost (4.16), the minimum achievable cost will still be (4.27). To verify this, we note from Remark 4.1 that since $J_k(w)$ is positive definite and, hence, strongly convex, its unique minimizer under Assumption 2.1 will be w^o. Therefore, substituting w^o into (4.16) will give its minimum, i.e.:

$$\begin{aligned}\min_{w} J_k(w) &= \mathbb{E}|\boldsymbol{d}_k(i) - \boldsymbol{z}_{k,i}w^o|^2 - \sigma^2_{n,k}\|w^o\|^2 \\ &= \sigma^2_{d,k} - r^*_{du,k}R^{-1}_{u,k}r_{du,k} \qquad (4.28)\\ &= J_{\min}. \qquad (4.29)\end{aligned}$$

We use this result to estimate the regression noise variance $\sigma^2_{n,k}$ at each node k.

Now, let us introduce

$$\boldsymbol{e}_k(i) \triangleq \boldsymbol{d}_k(i) - \boldsymbol{z}_{k,i}\boldsymbol{w}_{k,i-1} \qquad (4.30)$$

where $\boldsymbol{w}_{k,i-1}$ is the weight estimate from ATC diffusion (which would be replaced by $\boldsymbol{\psi}_{k,i-1}$ for CTA diffusion). Considering $J_k(\boldsymbol{w}_{k,i-1})$, for sufficiently small step-sizes and in the limit when the weight estimate is close enough to w^o, it holds that:

$$\mathbb{E}|\boldsymbol{e}_k(i)|^2 - \sigma^2_{n,k}\|w^o\|^2 \approx J_{\min}. \qquad (4.31)$$

From (4.2) and (4.27), it can be verified that $J_{\min} = \sigma^2_{v,k}$, and hence from (4.31), we can write:

$$\mathbb{E}|\boldsymbol{e}_k(i)|^2 \approx \sigma^2_{v,k} + \sigma^2_{n,k}\|w^o\|^2. \qquad (4.32)$$

In this relation, $\sigma^2_{v,k}$, can be ignored if $\sigma^2_{n,k}\|w^o\|^2 \gg \sigma^2_{v,k}$. Under such circumstances, if we assume $\|w^o\|^2 \neq 0$, which is true for systems with at least one non-zero coefficient, then the variance of the regression noise can be obtained by:

$$\sigma^2_{n,k} \approx \frac{\mathbb{E}|\boldsymbol{e}_k(i)|^2}{\|w^o\|^2}. \qquad (4.33)$$

Since, in (4.33), $\mathbb{E}|\boldsymbol{e}_k(i)|^2$ and the unknown parameter, w^o, are initially unavailable, we can estimate $\sigma^2_{n,k}$ using the following relations as the latest estimates of these quantities become available, i.e.,

$$\boldsymbol{f}_k(i) = \alpha\boldsymbol{f}_k(i-1) + (1-\alpha)|\boldsymbol{e}_k(i)|^2 \qquad (4.34)$$

$$\boldsymbol{\sigma}^2_{n,k}(i) = \frac{\boldsymbol{f}_k(i)}{\|\boldsymbol{w}_{k,i}\|^2} \qquad (4.35)$$

where $0 \ll \alpha < 1$ is a smoothing factor with nominal values in the range of $[0.95, 0.99]$.

Assumption 4.2. *The regression noise variance, $\sigma_{n,k}^2$, and the output measurement noise, $\sigma_{v,k}^2$, satisfy the following inequality*

$$\sigma_{n,k}^2 \|w^o\|^2 \gg \sigma_{v,k}^2 \tag{4.36}$$

Under this assumption, the regressor noise variance at each node k can be adaptively estimated via (4.34) and (4.35) using the data samples $\boldsymbol{e}_k(i)$ and $\boldsymbol{w}_{k,i-1}$ supplied from the bias-compensated LMS iterations.

4.4 Performance Analysis

In this section, we analyze the convergence and stability performance of the ATC and CTA bias-compensated diffusion LMS algorithms by viewing them as special cases of a more general diffusion algorithm of the form:

$$\boldsymbol{\phi}_{k,i-1} = \sum_{\ell \in \mathcal{N}_k} a_{\ell,k}^{(1)} \boldsymbol{w}_{\ell,i-1} \tag{4.37}$$

$$\boldsymbol{\psi}_{k,i} = \boldsymbol{\phi}_{k,i-1} - \mu_k \sum_{\ell \in \mathcal{N}_k} c_{\ell,k} \left[\widehat{\nabla_{\phi} J_\ell}(\boldsymbol{\phi}_{k,i-1}) \right]^* \tag{4.38}$$

$$\boldsymbol{w}_{k,i} = \sum_{\ell \in \mathcal{N}_k} a_{\ell,k}^{(2)} \boldsymbol{\psi}_{\ell,i} \tag{4.39}$$

where $\{a_{\ell,k}^{(1)}\}$ and $\{a_{\ell,k}^{(2)}\}$ are non-negative real coefficients corresponding to the (ℓ, k)-th entries of left-stochastic matrices A_1 and A_2, respectively, which have the same properties as A. Different choices for A_1 and A_2 corresponds to different operation modes. For instance, $A_1 = I$ and $A_2 = A$ correspond to ATC whereas $A_1 = A$ and $A_2 = I$ generate CTA. For mathematical tractability, in our analysis, we assume that the variances of the regression noises, i.e., $\sigma_{n,k}^2$, over the network are known *a-priori*.

We define the local weight-error vectors $\{\tilde{\boldsymbol{w}}_{k,i}, \tilde{\boldsymbol{\psi}}_{k,i}, \tilde{\boldsymbol{\phi}}_i\}$, and form the global weight-error vectors $\{\tilde{\boldsymbol{w}}_i, \tilde{\boldsymbol{\psi}}_i, \tilde{\boldsymbol{\phi}}_i\}$ similar to previous chapter. Note that now the local error vectors are of size $M \times 1$ and the global error vectors are of size $NM \times 1$. We also define the block variables:

$$\boldsymbol{g}_i = \mathcal{C}^T \text{col}\{\boldsymbol{z}^*_{\ell,i}\boldsymbol{v}_1(i), \ldots, \boldsymbol{z}^*_{N,i}\boldsymbol{v}_N(i)\} \tag{4.40}$$

$$\boldsymbol{\mathcal{R}}_i = \text{diag}\Big\{\sum_{\ell\in\mathcal{N}_k} c_{\ell,k}\,(\boldsymbol{z}^*_{\ell,i}\boldsymbol{z}_{\ell,i} - \sigma^2_{n,\ell}I), k = 1, \cdots, N\Big\} \tag{4.41}$$

$$\boldsymbol{\mathcal{P}}_i = \text{diag}\Big\{\sum_{\ell\in\mathcal{N}_k} c_{\ell,k}\,(\boldsymbol{z}^*_{\ell,i}\boldsymbol{n}_{\ell,i} - \sigma^2_{n,\ell}I), k = 1, \cdots, N\Big\} \tag{4.42}$$

$$\mathcal{M} = \text{diag}\Big\{\mu_1 I_M, \cdots, \mu_N I_M\Big\} \tag{4.43}$$

and introduce the following extended combination matrices:

$$\mathcal{A}_1 = A_1 \otimes I_M, \quad \mathcal{A}_2 = A_2 \otimes I_M, \quad \mathcal{C} = C \otimes I_M \tag{4.44}$$

Using these definitions and update equations (4.37)-(4.39), it can be verified that the following relations hold:

$$\begin{aligned}
\tilde{\boldsymbol{\phi}}_{i-1} &= \mathcal{A}_1^T \tilde{\boldsymbol{w}}_{i-1} \\
\tilde{\boldsymbol{\psi}}_i &= \tilde{\boldsymbol{\phi}}_{i-1} - \mathcal{M}(\boldsymbol{g}_i - \boldsymbol{\mathcal{P}}_i\omega^o + \boldsymbol{\mathcal{R}}_i\tilde{\boldsymbol{\phi}}_{i-1}) \\
\tilde{\boldsymbol{w}}_i &= \mathcal{A}_2^T \tilde{\boldsymbol{\psi}}_i
\end{aligned} \tag{4.45}$$

where $\omega^o = \mathbb{1} \otimes w^o$. From the set of equations given in (4.45), it is deduced that the network error vector $\tilde{\boldsymbol{w}}_i$ evolves with time according to the recursion:

$$\tilde{\boldsymbol{w}}_i = \boldsymbol{\mathcal{B}}_i\tilde{\boldsymbol{w}}_{i-1} - \mathcal{A}_2^T\mathcal{M}\boldsymbol{g}_i + \mathcal{A}_2^T\mathcal{M}\boldsymbol{\mathcal{P}}_i\omega^o \tag{4.46}$$

where the time-varying matrix $\boldsymbol{\mathcal{B}}_i$ is defined as:

$$\boldsymbol{\mathcal{B}}_i = \mathcal{A}_2^T(I - \mathcal{M}\boldsymbol{\mathcal{R}}_i)\mathcal{A}_1^T \tag{4.47}$$

4.4.1 Mean Convergence and Stability

Tacking the expectation of both sides of (4.46) and considering Assumption 4.1, we arrive at:

$$\mathbb{E}[\tilde{\boldsymbol{w}}_i] = \mathcal{B}\,\mathbb{E}[\boldsymbol{w}_{i-1}] \tag{4.48}$$

where in this relation:

$$\mathcal{B} \triangleq \mathbb{E}[\boldsymbol{\mathcal{B}}_i] = \mathcal{A}_2^T(I - \mathcal{M}\mathcal{R})\mathcal{A}_1{}^T \tag{4.49}$$

$$\mathcal{R} \triangleq \mathbb{E}[\boldsymbol{\mathcal{R}}_i] = \mathrm{diag}\Big\{\sum_{\ell\in\mathcal{N}_k} c_{\ell,k}\, R_{u,\ell}, k = 1, \cdots, N\Big\} \tag{4.50}$$

To obtain (4.48), we used the fact that $\mathbb{E}[\mathcal{A}_2{}^T\mathcal{M}\boldsymbol{g}_i] = 0$ because $\boldsymbol{v}_{k,i}$ is independent of $\boldsymbol{z}_{k,i}$ and $\mathbb{E}[\boldsymbol{v}_k(i)] = 0$. Moreover, we have $\mathbb{E}[\boldsymbol{\mathcal{P}}_i] = 0$ because $\mathbb{E}[\boldsymbol{z}^*_{\ell,i}\boldsymbol{n}_{\ell,i}] = \sigma^2_{n,\ell}I$. According to (4.48), $\lim_{i\to\infty}\mathbb{E}\|\tilde{\boldsymbol{w}}_i\| \to 0$ if $\mathcal{B}$ is stable (i.e., when $\rho(\mathcal{B}) < 1$). In fact, because $\rho(\mathcal{A}_1) = \rho(\mathcal{A}_2) = 1$ and $\mathcal{R} > 0$ choosing the step-sizes according to:

$$0 < \mu_k < \frac{2}{\rho\Big(\sum_{\ell\in\mathcal{N}_k} c_{\ell,k}R_{u,\ell}\Big)} \tag{4.51}$$

guarantees $\rho(\mathcal{B}) < 1$. We omit the proof, because a similar argument is made in references [82] and [63]. We summarize the mean-convergence results of the proposed bias-compensated diffusion LMS in the following.

Theorem 4.1. *Consider an adaptive network that operates using diffusion Algorithms 4.1 or 4.2 with the space-time data (4.1) and (4.2) under Assumption 4.1. In this network, if we assume that the regressors noise variances are perfectly estimated, the mean error vector evolves with time according to (4.48). Furthermore, Algorithms 4.1 and 4.2 will be asymptotically unbiased and stable provided that the step-size at each node k satisfies (4.51).*

Remark 4.2. *In networks with noisy regression data (4.1), the estimates generated by the previous diffusion LMS strategies such as the ones proposed in [33, 82] are biased, i.e,* $\mathbb{E}[\tilde{\boldsymbol{w}}_i] \neq 0$ *as $i \to \infty$. This can be readily shown if we remove $\sigma^2_{n,k}$ from (4.41) and (4.42).*

In this scenario, the algorithm will be mean stable if

$$0 < \mu_k < \frac{2}{\rho\Big(\sum_{\ell\in\mathcal{N}_k} c_{\ell,k}\Big(R_{u,\ell}+\sigma_{n,\ell}^2 I_M\Big)\Big)} \tag{4.52}$$

Then, for sufficiently small step-sizes, satisfying (4.52), it can be verified that the estimate of the standard diffusion LMS deviates from the network optimal solution ω^o by:

$$\lim_{i\to\infty}\mathbb{E}[\tilde{\boldsymbol{w}}_i] = (I_{NM}-\mathcal{B}')^{-1}\mathcal{A}_2^T\mathcal{M}\mathcal{P}'\omega^o \tag{4.53}$$

where

$$\mathcal{B}' \triangleq \mathcal{A}_2^T(I_{NM}-\mathcal{M}\mathcal{R}')\mathcal{A}_1^T \tag{4.54}$$

$$\mathcal{R}' \triangleq \text{diag}\Big\{\sum_{\ell\in\mathcal{N}_k} c_{\ell,k}\Big(R_{u,\ell}+\sigma_{n,\ell}^2 I_M\Big), k=1,\cdots,N\Big\} \tag{4.55}$$

$$\mathcal{P}' \triangleq \text{diag}\Big\{\sum_{\ell\in\mathcal{N}_k} c_{\ell,k}\,\sigma_{n,\ell}^2 I_M,\ k=1,\cdots,N\Big\} \tag{4.56}$$

As it is clear from (4.53), the bias is created by the regression noise $\{\boldsymbol{n}_{k,i}\}$ only, whereas the noise $\{\boldsymbol{v}_k(i)\}$ has no effect on generating the bias.

4.4.2 Mean-Square Stability and Performance

To study the mean-square performance, we first follow the energy conservation arguments of [33, 89] and determine a variance relation. The relation can be obtained by computing the weighted squared norm of both sides of equation (4.46) and taking expectations under Assumption 4.1:

$$\mathbb{E}\|\tilde{\boldsymbol{w}}_i\|_\Sigma^2 = \mathbb{E}\Big(\|\tilde{\boldsymbol{w}}_{i-1}\|_{\boldsymbol{\Sigma}'}^2\Big) + \mathbb{E}[\boldsymbol{g}_i^*\mathcal{M}\mathcal{A}_2\Sigma\mathcal{A}_2^T\mathcal{M}\boldsymbol{g}_i] + \mathbb{E}[\omega^{o*}\boldsymbol{\mathcal{P}}_i^*\mathcal{M}\mathcal{A}_2\Sigma\mathcal{A}_2^T\mathcal{M}\boldsymbol{\mathcal{P}}_i\omega^o] \tag{4.57}$$

where $\|x\|_\Sigma^2 = x^*\Sigma x$ and $\Sigma \geq 0$ is a weighting matrix that we are free to choose. For relation (4.57), we have:

$$\boldsymbol{\Sigma}' = \boldsymbol{\mathcal{B}}_i{}^*\Sigma\boldsymbol{\mathcal{B}}_i \tag{4.58}$$

It follows from Assumption 4.1 that $\tilde{\boldsymbol{w}}_{i-1}$ and $\boldsymbol{\mathcal{R}}_i$ are independent of each other so that

$$\mathbb{E}\Big(\|\tilde{\boldsymbol{w}}_{i-1}\|^2_{\boldsymbol{\Sigma}'}\Big) = \mathbb{E}\|\tilde{\boldsymbol{w}}_{i-1}\|^2_{\mathbb{E}[\boldsymbol{\Sigma}']} \tag{4.59}$$

Substituting this expression into (4.57), we arrive at:

$$\mathbb{E}\|\tilde{\boldsymbol{w}}_i\|^2_{\Sigma} = \mathbb{E}\|\tilde{\boldsymbol{w}}_{i-1}\|^2_{\Sigma'} + \mathrm{Tr}[\Sigma\mathcal{A}_2^T\mathcal{M}\mathcal{G}\mathcal{M}\mathcal{A}_2] + \mathrm{Tr}[\Sigma\mathcal{A}_2^T\mathcal{M}\Pi\mathcal{M}\mathcal{A}_2] \tag{4.60}$$

where $\Sigma' = \mathbb{E}[\boldsymbol{\mathcal{B}}_{\boldsymbol{i}}^*\Sigma\boldsymbol{\mathcal{B}}_i]$. In equation (4.60) $\mathcal{G} = \mathbb{E}[\boldsymbol{g}_i\boldsymbol{g}_i^*]$, which using (4.40) is given by:

$$\mathcal{G} = \mathcal{C}^T\mathrm{diag}\Big\{\sigma^2_{v,1}(R_{u,1}+\sigma^2_{n,1}I),\ldots,\sigma^2_{v,N}(R_{u,N}+\sigma^2_{n,N}I)\Big\}\mathcal{C} \tag{4.61}$$

In relation (4.60), $\Pi = \mathbb{E}[\boldsymbol{\mathcal{P}}_i\omega^o\omega^{o*}\boldsymbol{\mathcal{P}}_i^*]$ and its (k,j)-th block is computed as (see Appendix A.5):

$$\Pi_{k,j} = \sum_{\ell} c_{\ell,k}c_{\ell,j}\Big\{\sigma^2_{n,\ell}\|w^o\|^2\Big(R_{u,\ell}+\sigma^2_{n,\ell}I\Big) + (\beta-1)\sigma^4_{n,\ell}w^ow^{o*}\Big\} \tag{4.62}$$

where $\beta = 2$ for real-valued data and $\beta = 1$ for complex-valued data.

If we introduce $\sigma = \mathrm{bvec}(\Sigma)$ and $\sigma' = \mathrm{bvec}(\Sigma')$ then we can write $\sigma' = \mathcal{F}\sigma$ for some matrix $\mathcal{F}$ to be determined (see below) and the variance relation in (4.60) can be rewritten more compactly as:

$$\mathbb{E}\|\tilde{\boldsymbol{w}}_i\|^2_{\sigma} = \mathbb{E}\|\tilde{\boldsymbol{w}}_{i-1}\|^2_{\mathcal{F}\sigma} + \gamma^T\sigma \tag{4.63}$$

where we are using the notation $\|x\|^2_{\sigma}$ as a short form for $\|x\|^2_{\Sigma}$, and where

$$\gamma = \mathrm{bvec}(\mathcal{A}_2^T\mathcal{M}\mathcal{G}^T\mathcal{M}\mathcal{A}_2 + \mathcal{A}_2^T\mathcal{M}\Pi^T\mathcal{M}\mathcal{A}_2) \tag{4.64}$$

The operator bvec$(\cdot)$ operator vectorizes a block matrix by first vectorizing each of its blocks and then stacking the resulting vectors on top of each other into a column [65]. To compute $\mathcal{F}$, we consider the two following cases:

Small Step-Sizes: For this case, we multiply the terms in Σ' to get:

$$\Sigma' = \mathcal{A}_1\left(\mathcal{A}_2\Sigma\mathcal{A}_2^T - \mathcal{R}\mathcal{M}\mathcal{A}_2\Sigma\mathcal{A}_2^T - \mathcal{A}_2\Sigma\mathcal{A}_2^T\mathcal{M}\mathcal{R}\right)\mathcal{A}_1{}^T + \mathbb{E}[\mathcal{A}_1\boldsymbol{\mathcal{R}}_i^*\mathcal{M}\mathcal{A}_2\Sigma\mathcal{A}_2^T\mathcal{M}\boldsymbol{\mathcal{R}}_i\mathcal{A}_1{}^T] \tag{4.65}$$

Under the small step size condition, the last term which depends on $\{\mu_k^2\}$ can be neglected. As a result, we obtain $\text{bvec}(\Sigma') = \mathcal{F}\sigma$ where

$$\mathcal{F} = (\mathcal{A}_1 \otimes_b \mathcal{A}_1)(I - I \otimes_b \mathcal{R}\mathcal{M} - \mathcal{R}^T\mathcal{M} \otimes_b I)(\mathcal{A}_2 \otimes_b \mathcal{A}_2) \tag{4.66}$$

Gaussian Regressor: If the regression data, $\{\boldsymbol{u}_{k,i}\}$, are zero mean circular complex-valued Gaussian random vectors, then according to Appendix A.6, we obtain:

$$\mathcal{F} = (\mathcal{A}_1 \otimes_b \mathcal{A}_1)\Big\{I - I \otimes_b \mathcal{R}\mathcal{M} - \mathcal{R}^T\mathcal{M} \otimes_b I\Big\} \times (\mathcal{A}_2 \otimes_b \mathcal{A}_2) + \Delta\mathcal{F} \tag{4.67}$$

where $\Delta\mathcal{F}$ is given by (A.69). Note that since $\Delta\mathcal{F}$ depends on $\{\mu_k^2\}$, it can be neglected under the small step size condition that reduces expression (4.67) to (4.66). It can be verified that under the small step-size condition a good approximation of $\mathcal{F}$ that requires no assumption on the distribution of the regression data is:

$$\mathcal{F} \approx \mathcal{B}^T \otimes_b \mathcal{B}^* \tag{4.68}$$

Before proceed to characterize the network MSD and EMSE, we briefly examine the stability of the algorithms in the mean-square sense. Using (4.69), we can write:

$$\lim_{i\to\infty} \mathbb{E}|\tilde{\boldsymbol{w}}_i\|_\sigma^2 = \lim_{i\to\infty} \mathbb{E}\|\tilde{\boldsymbol{w}}_{-1}\|_{\mathcal{F}^{i+1}\sigma}^2 + \gamma^T \sum_{j=0}^{\infty} \mathcal{F}^j\sigma \tag{4.69}$$

As it is evident from this expression, the stability of the algorithm in the mean-square sense depends on the stability of $\mathcal{F}$. In the previous works on diffusion LMS algorithms, including [82], it has been shown that $\mathcal{F}$ will be stable if $\mathcal{B}$ is stable. According to our mean-convergence analysis, the stability of $\mathcal{B}$ is guaranteed if (4.51) holds. Therefore, the step-size condition (4.51) is sufficient to guarantee stability in the mean and mean-square sense.

To obtain mean-square error (MSE) steady state expressions of the network, we let i goes to infinity and use expression (4.63) to write:

$$\lim_{i\to\infty} \mathbb{E}\|\tilde{\boldsymbol{w}}_i\|^2_{(I-\mathcal{F})\sigma} = \gamma^T\sigma \tag{4.70}$$

As shown in Chapter 2, from (4.70), we then arrive at the following relations to, respectively, obtain MSD and EMSE of each agent k over the network:

$$\eta_k = \gamma^T\sigma_{\text{msd}_k} \tag{4.71}$$

$$\zeta_k = \gamma^T\sigma_{\text{emse}_k} \tag{4.72}$$

where

$$\sigma_{\text{msd}_k} = (I-\mathcal{F})^{-1}\text{bvec}\Big(\text{diag}(e_k)\otimes I_M\Big) \tag{4.73}$$

$$\sigma_{\text{emse}_k} = (I-\mathcal{F})^{-1}\text{bvec}\Big(\text{diag}(e_k)\otimes R_{u,k}\Big) \tag{4.74}$$

The network MSD, and EMSE are defined as the average of MSD and EMSE over the network.

4.4.3 Mean-Square Transient Behavior

We use (4.69) to obtain an expression for the mean-square behavior of the algorithm in transient-state. In this expression, if we substitute $\boldsymbol{w}_{k,-1} = 0$, $\forall k \in \{1,\cdots,N\}$, we obtain:

$$\|\tilde{\boldsymbol{w}}_i\|^2_\sigma = \|w^o\|^2_{\mathcal{F}^{i+1}\sigma} + \gamma^T\sum_{j=0}^{i}\mathcal{F}^j\sigma \tag{4.75}$$

Writing this recursion for $i-1$, and subtract it from (4.75) leads to:

$$\|\tilde{\boldsymbol{w}}_i\|^2_\sigma = \|\tilde{\boldsymbol{w}}_{i-1}\|^2_\sigma + \|w^o\|^2_{\mathcal{F}^i(I-\mathcal{F})\sigma} + \gamma^T\mathcal{F}^i\sigma \tag{4.76}$$

By replacing σ with $\sigma_{\text{msd}_k} = \text{bvec}(\text{diag}\{e_k\}\otimes I_M)$ and $\sigma_{\text{emse}_k} = \text{bvec}(\text{diag}\{e_k\}\otimes R_{u,k})$ and using $\boldsymbol{w}_{k,-1} = 0$, we arrive at the following two recursions for the evolution of MSD and EMSE over time:

$$\eta_k(i) = \eta_k(i-1) - \|w^o\|_{\mathcal{F}^i(I-\mathcal{F})\sigma_{\text{msd}_k}} + \gamma^T \mathcal{F}^i \sigma_{\text{msd}_k} \tag{4.77}$$

$$\zeta_k(i) = \zeta_k(i-1) - \|w^o\|_{\mathcal{F}^{i-1}(I-\mathcal{F})\sigma_{\text{emse}_k}} + \gamma^T \mathcal{F}^{i-1} \sigma_{\text{emse}_k} \tag{4.78}$$

The MSD and EMSE of the network can be computed either by averaging the nodes transient behavior, or by substituting

$$\sigma_{\text{msd}} = \frac{1}{N}\text{bvec}(I_{MN}) \tag{4.79}$$

$$\sigma_{\text{emse}} = \frac{1}{N}\text{bvec}\Big(\text{diag}\{R_{u,1}, \cdots, R_{u,N}\}\Big) \tag{4.80}$$

in recursion (4.76). We summarize the mean-square analysis results of the algorithms in the following:

Theorem 4.2. *Consider an adaptive network operating under bias-compensated diffusion Algorithm 4.1 or 4.2 with the space-time data (4.1) and (4.2) that satisfy Assumption 4.1. In this network, if we assume that the regressors noise variances are known or perfectly estimated and nodes initialize at zero, then the MSD and EMSE of each node k evolve with time according to (4.77) and (4.78) and the network MSD and EMSE follow recursions:*

$$\eta(i) = \eta(i-1) - \|w^o\|_{\mathcal{F}^i(I-\mathcal{F})\sigma_{\text{msd}}} + \gamma^T \mathcal{F}^i \sigma_{\text{msd}} \tag{4.81}$$

$$\zeta(i) = \zeta(i-1) - \|w^o\|_{\mathcal{F}^{i-1}(I-\mathcal{F})\sigma_{\text{emse}}} + \gamma^T \mathcal{F}^{i-1} \sigma_{\text{emse}} \tag{4.82}$$

where σ_{msd}, and σ_{emse} are defined in (4.79) and (4.80). Moreover, if the step-sizes are chosen to satisfy (4.51), the network will be stable and converges in the mean and mean-square sense and the agents over the network reach the steady-state MSD and EMSE, respectively, characterized by (4.71) and (4.72).

4.5 Simulation Results

In this section, we present computer experiments to illustrate the efficiency of the proposed algorithms and to verify the theoretical findings. We evaluate the algorithm performance for known regressor noise variance and with adaptive noise variance estimation. We consider a connected network with $N = 20$ nodes that are positioned on a unit square area with

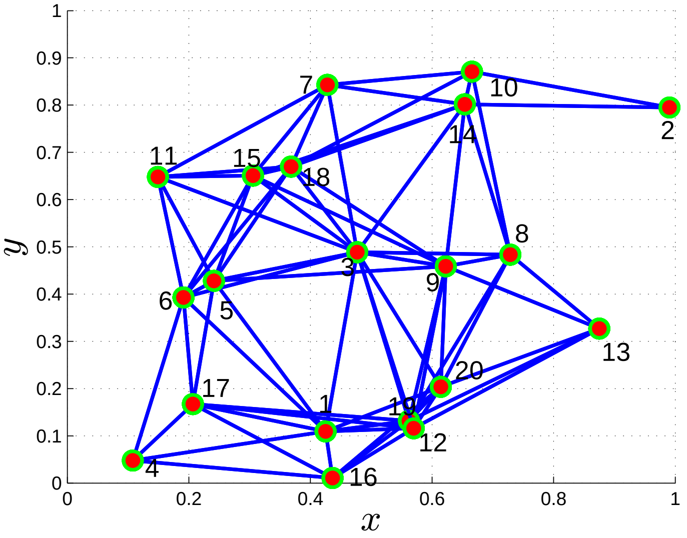


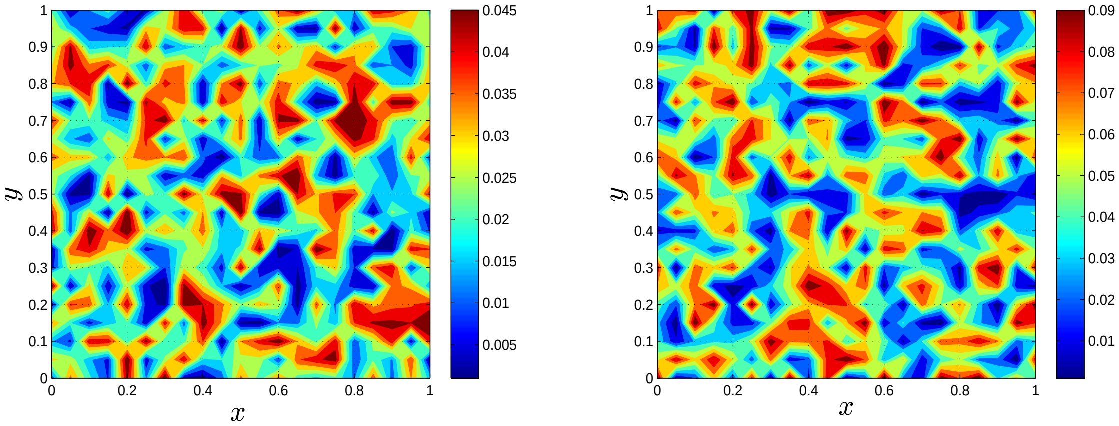
Fig. 4.2 Network topology used in the simulations.

maximum communication distance of 0.4 unit length. The network topology is shown in Fig. 4.2. We choose $A_1 = I$, compute A_2 using the relative-variance rule (4.24) and choose the matrix C according to the metropolis criterion [27,82]. In the plots, we use A_{rel} and C_{met} to refer to this particular choice of A_2 and C. The network data are generated according to model (4.1) and (4.2). The aim is to estimate the system parameter vector $w^o = [1, 1]^T/\sqrt{2}$ over the network using the proposed bias-compensated diffusion algorithms. In all our experiments, the curves from the simulation results are drawn from the average of 500 independent runs.

We choose the step-sizes as $\mu_k = 0.05$, and set $\boldsymbol{w}_{k,-1} = [0, 0]^T$, for all k. We adopt Gaussian distribution to generate $\boldsymbol{v}_k(i)$, $\boldsymbol{n}_{k,i}$ and $\boldsymbol{u}_{k,i}$. The covariance matrices of the regression data and the regression noise are of the form $R_{u,k} = \sigma_{u,k}^2 I_M$, and $\sigma_{n,k}^2 I_M$, respectively. Figs. 4.3(a) and 4.3(b) illustrate the variances of the nodes' regressor noise, $\sigma_{n,k}^2$, and observation noise, $\sigma_{v,k}^2$. The network signal power profile, given by $\text{Tr}(R_{u,k})$ versus node index k, is shown in Fig. 4.4.

Transient MSE Results with Perfect Noise Variance Estimation:

In Fig. 4.5, we demonstrate the network transient behavior in terms of MSD and EMSE for the proposed diffusion LMS algorithm, standard diffusion LMS algorithm [33] and the non-



(a) Distribution of noise variance $\sigma^2_{n,k}$ (b) Distribution of noise variance $\sigma^2_{v,k}$.

Fig. 4.3 Spatial distribution of noise energy profile over the network.

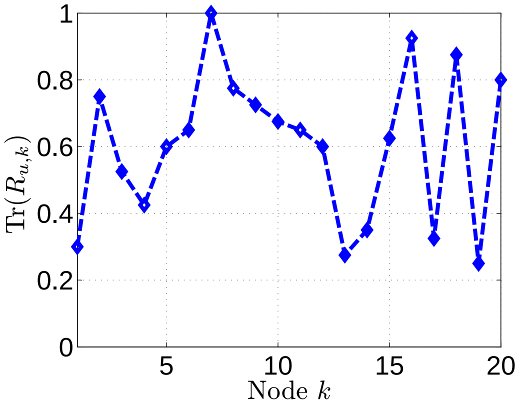


Fig. 4.4 Regressor power Tr$(R_{u,k})$.

cooperative mode of the proposed algorithm. Note that $A_2 = I$ and $C = I$ correspond to the non-cooperative network mode of the proposed algorithm, where each node runs a stand alone bias-compensated LMS. As the results indicate, the performance of the cooperative network with C_{met} and A_{rel} exceeds that of the non-cooperative case by 12 dB. We also observe that the proposed algorithm outperform the standard diffusion LMS [33] by more that 12dB. It is interesting to note that the non-cooperative algorithm outperforms the standard diffusion LMS by about 1dB.

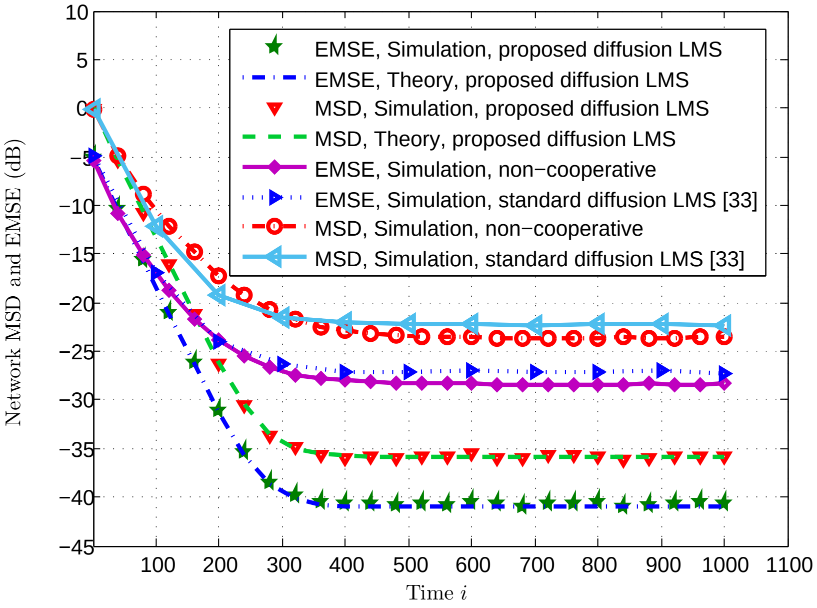


Fig. 4.5 Convergence behavior of the proposed bias-compensated diffusion LMS, standard diffusion LMS and non-cooperative LMS algorithms.

We also present the EMSE and MSD of some randomly chosen nodes in Fig. 4.6. In particular, we plot the EMSE learning curves of nodes 4 and 18 and the MSD learning curves of nodes 5 and 15. We observe that the MSD curves of the chosen nodes are identical. Since the algorithm is unbiased, this implies that these nodes have reached agreement about the unknown network parameter, w^o. As we will show in the steady-state results, all nodes over the network almost reach agreement. We note that, in all scenarios, there is a good agreement between simulations and the analysis results.

Steady-State MSE Results with Perfect Noise Variance Estimation:

The network steady-state MSD and EMSE are shown in Figs. 4.7(a) and 4.7(b). From these figures, we observe that there is a good agreement between simulations and analytical findings. In addition, we consider the case when nodes only exchange their intermediate estimates (i.e., when $C = I$). It is seen that the MSD performance of the algorithm with C_{met} is 1dB superior than that with $C = I$. We also observe that the performance discrepancies between nodes in terms of MSD is less than 0.5dB for cooperative scenarios, while in the non-cooperative scenario it is more than 5dB. This shows agreement in the network in spite of different noise and energy profiles at each node. Note that the fluctuations in EMSE over the network are due to differences in energy level in the nodes' input signals, but this does not preclude the cooperating nodes from reaching a consensus in the estimated parameters.

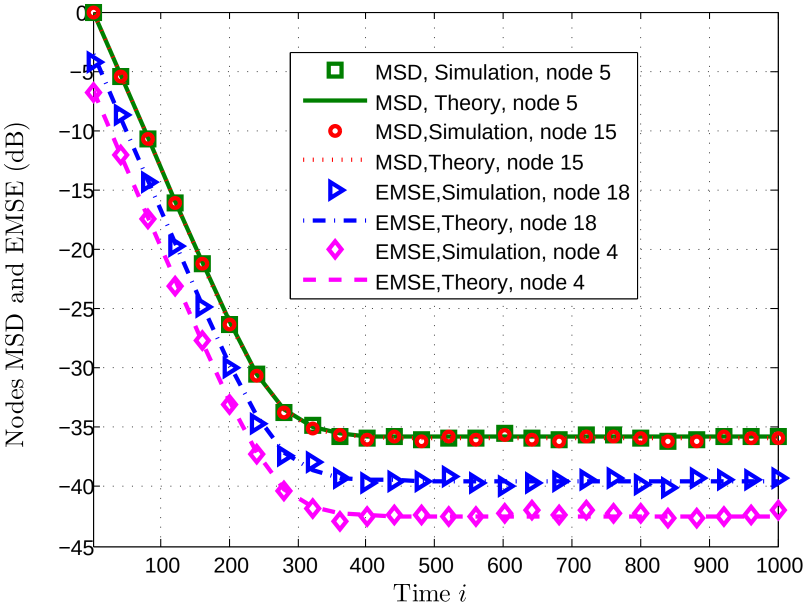
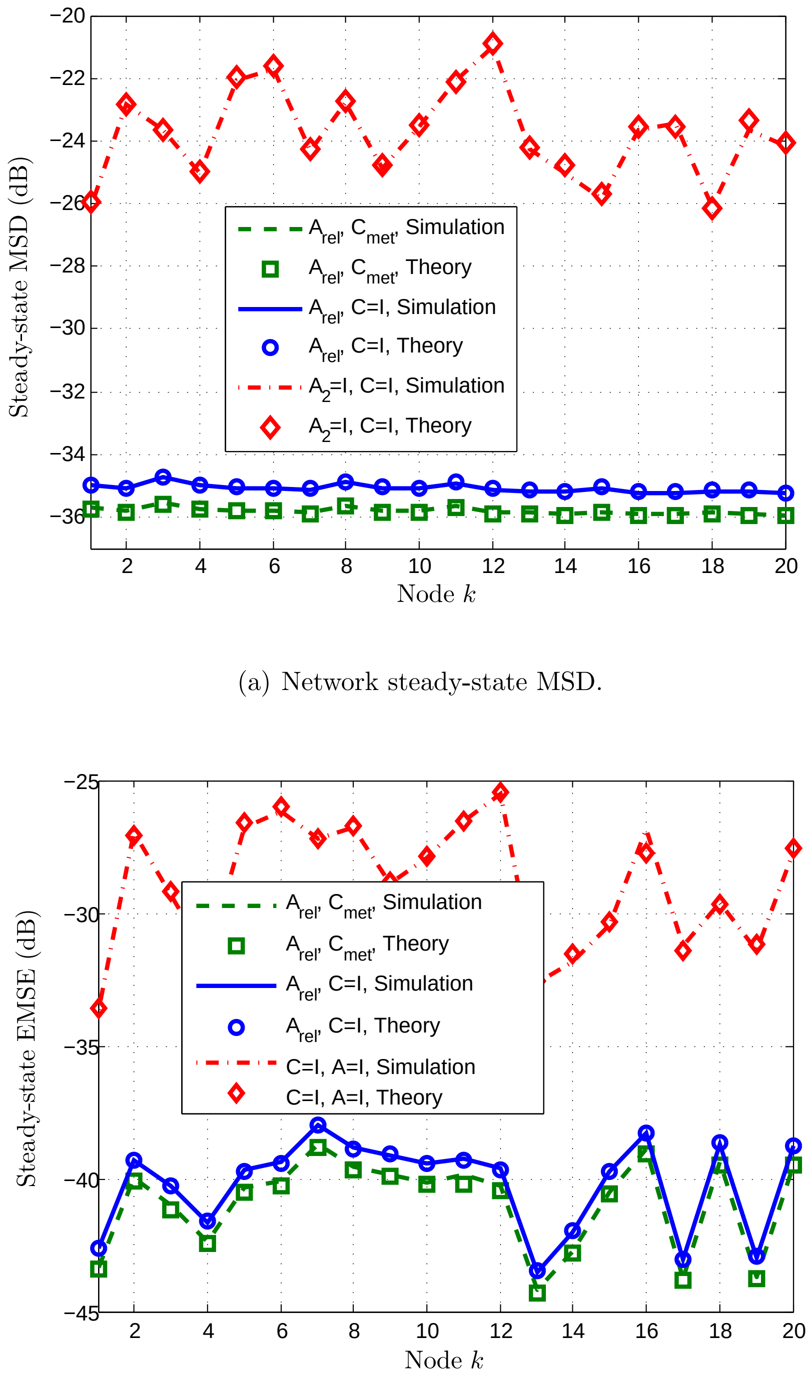


Fig. 4.6 MSD learning curves of nodes 5 and 15 and EMSE learning curves of nodes 4 and 18.

MSE Results with Adaptive Noise Variance Estimation:

We compare the transient and steady-state behavior of the bias-compensated diffusion LMS with known regressor noise variance and adaptive noise variance estimation. For this experiment, we consider the same network topology and noise profile as above. However, the unknown parameter vector to be estimated, in this case, is $w^o = 2\mathbb{1}_5 + 2j\mathbb{1}_5$, where



(a) Network steady-state MSD.

(b) Network steady-state EMSE .

Fig. 4.7 Steady-state MSD and EMSE of the network for different combination matrices.

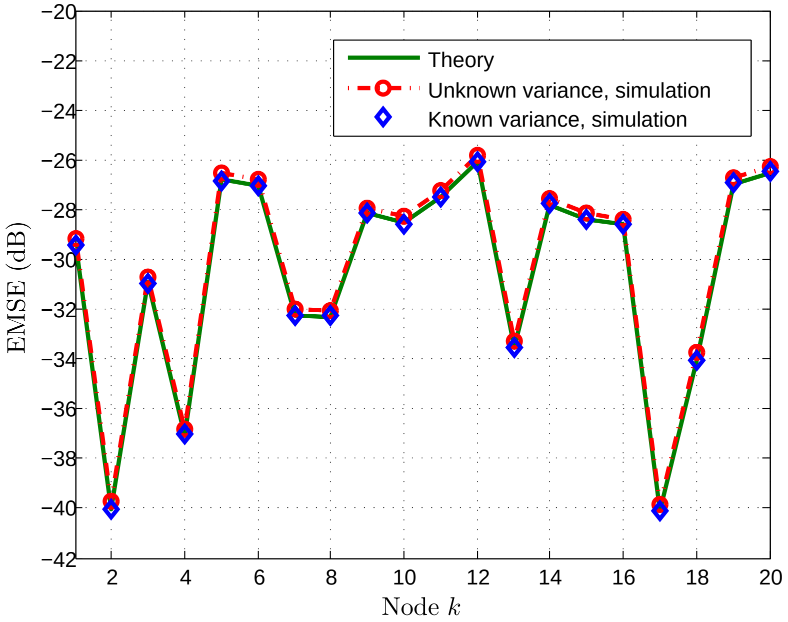


Fig. 4.8 Steady-state network EMSE with known and estimated regressor noise variances.

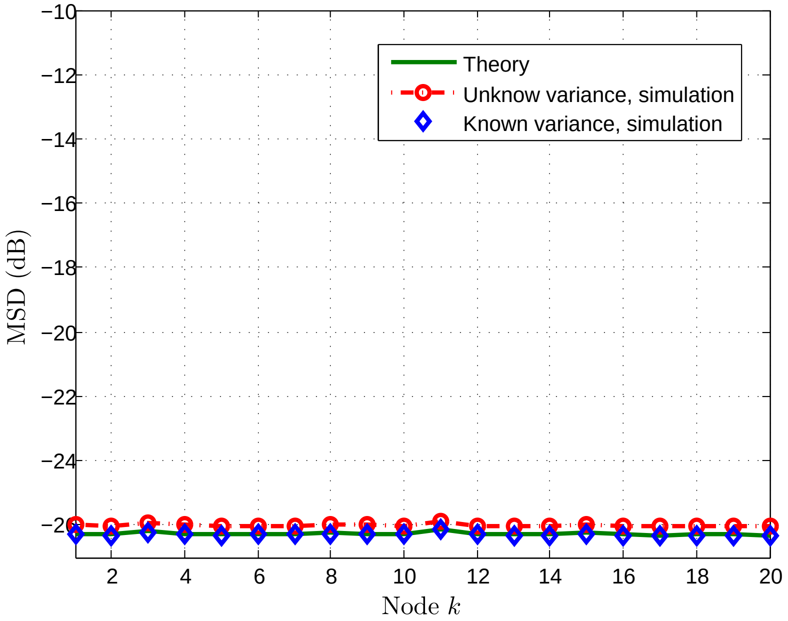


Fig. 4.9 Steady-state network MSD with known and estimated regressor noise variances.

$\mathbb{1}_M$ is a $M \times 1$ column vector with unit entries. The network energy profile is chosen as $\text{Tr}(R_{u,k}) = 20\text{Tr}(\sigma^2_{n,k} I)$. Using these choices, Assumption 4.2 will be satisfied. We set $\alpha = 0.99$ and $\mu_k = 0.01$ for all k.

Figs. 4.8 and 4.9 show the steady-state EMSE and MSD of the network for these two cases. The steady-state values are obtained by averaging over the last 200 samples after initial convergence. We observe that the performance of the proposed bias-compensated LMS algorithm with adaptive noise variance estimation is almost identical to that of the ideal case with known noise variances.

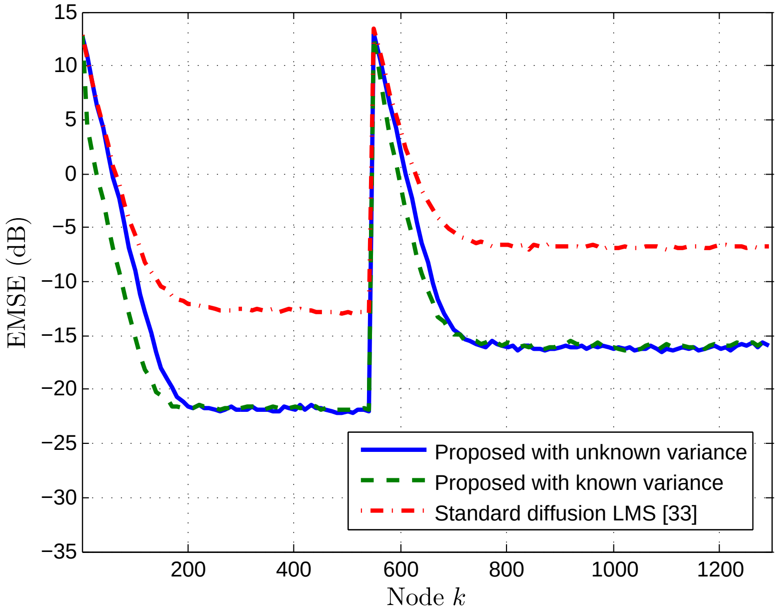


Fig. 4.10 EMSE Tracking performance with known and estimated regressor noise variances.

Fig. 4.10 illustrates the tracking performance of the bias-compensated diffusion LMS algorithm for these two cases for a sudden change in the unknown parameter w^o and compares the results with that of the standard diffusion LMS algorithm given in [33]. The variation in the unknown parameter vector occurs at iteration $i = 550$ when w^o changes to $2\,w^o$. Similar conclusion as in Fig. 4.8 and 4.9 can be made for the proposed algorithms with known and estimated regression noise variances. We also observe that the proposed algorithms outperform the standard diffusion LMS [33] by nearly 10dB in steady-state.

Fig. 4.11 illustrates the results of regression noise variance estimation in the steady state. In this experiment, we observe that for $i \geq 350$, $\mathbb{E}[\boldsymbol{\sigma}^2_{n,k}(i)] \rightarrow \sigma^2_{n,k}$. This indicates that the proposed adaptive estimation strategy for computation of the nodes' regression noise variance over the network works well.

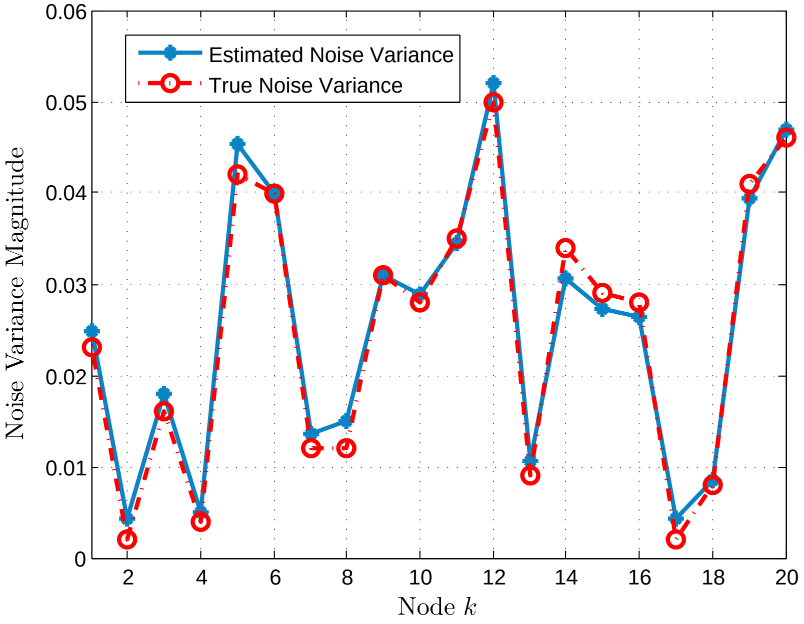


Fig. 4.11 The estimated and true value of the regression noise variance, $\sigma^2_{n,k}$, over the network.

4.6 Summary

We developed bias-compensated DLMS strategies for parameter estimation for sensor networks where the regression data are corrupted with additive noise. The proposed algorithms operate in distributed manner and exchange data with single-hop communication to save energy and communication resources. In the analysis, it has been shown that these algorithms are unbiased and converge in the mean and mean-square error sense for sufficiently small adaptation step-sizes. We carried out computer experiments that confirmed the effectiveness of the algorithms and verified our analytical findings. Further, in the experiments, it was shown that the steady-state performances of the developed algorithms with known and unknown noise variances are almost identical, under the assumed conditions. Nevertheless, the competency of the algorithms with unknown noise variances comes at the expense of higher computational complexity.

In the following chapter, we examine the performance of DLMS algorithms under the effects of wireless channel impairments, including, fading and path-loss in addition to link noise, and propose solution as how to mitigate such impacts.

Chapter 5

DLMS Algorithms over Wireless Sensor Networks

In this chapter, we study the performance of DLMS algorithms for distributed parameter estimation in sensor networks where nodes exchange information over wireless communication links[1]. Wireless channel impairments, such as fading and path-loss adversely affect the exchanged data, and subsequently, can cause instability in these algorithms. For such scenarios, we introduce an extended version of the DLMS algorithms by incorporating equalization coefficients into their structure to reverse the effects of path loss and fading. In the proposed algorithms, since nodes do not have access to the channel state information (CSI) of their neighbors, the equalization coefficients are computed from pilot-aided estimated channel coefficients. We also show that as a consequence of fading and path loss, regardless of using equalization coefficients, the network experiences link failures which render the use of static combination matrices impractical for the data fusion between nodes. The analysis reveals that by properly monitoring the CSI over the network and choosing sufficiently small adaptation step-sizes, the diffusion strategies are able to deliver satisfac-

[1] Part of the work presented in this chapter was published in:

- R. Abdolee, B. Champagne and A. H. Sayed, “Diffusion LMS strategies for parameter estimation over fading wireless channels”, in *Proc. of IEEE International Conference on Communication (ICC)*, Budapest, Hungary, June 2013, pp. 1926–1930.
- R. Abdolee and B. Champagne, “Diffusion LMS algorithms for sensor networks over non-ideal inter-sensor wireless channel”, in *Proc. of IEEE Int. Conf. Dist. Computer Sensor Systems (DCOSS)*, Barcelona, Spain, June 2011, pp. 1–6.

tory performance in the presence of fading and path loss.

5.1 Introduction

DLMS strategies have been widely investigated in networks with static topologies in which the communication links between agents or nodes remain invariant with respect to time [28, 33, 80, 123]. In particular, when the network topology is static, the diffusion strategies have been shown to converge in the mean and mean-square error sense in the slow adaptation regime [29, 33, 82, 124]. Previous studies have also examined the effect of noisy communication links on their performance [53–56]. The main conclusion drawn from these works is that performance degradations occur unless the combination weights used at each node are adjusted to counter the effect of noise over the links.

In the aforementioned works, it has been assumed that the links between neighboring nodes are always active (i.e., the network topology is static), regardless of the value of their instantaneous SNR, which sometimes can be extremely low. This assumption is, however, too restrictive in many existing and emerging applications in wireless communications and sensor network systems. For example, in mobile networks where the agents are allowed to change their position over time, the SNR over the communication links between nodes will vary due to the various channel impairments, including path loss, multi-path fading and shadowing. Consequently, the set of nodes with which each agent can communicate will also change as the agents move, and the network topology is therefore intrinsically dynamic. A time-varying network topology can also be created as a consequence of energy drain in nodes, leading to a sudden link-failure, or due to deployment or activation of substitute nodes over the network, i.e., creation of new links. It is therefore essential to investigate the performance of diffusion strategies over networks with time-varying (dynamic) topology and characterize the effects of link activity (especially link failure) on their convergence and stability.

In this chapter, we extend the application of DLMS strategies from multi-agent networks with ideal communication links to sensor networks with fading wireless channels. Under fading and path loss conditions over wireless links, the neighborhood sets become dynamic, with nodes leaving or entering neighborhoods depending on the quality of the links as defined by the instantaneous SNR conditions. Our analysis will show that if each node knows the channel state information (CSI) of its neighbors, the effects of fading and path-loss can be mitigated by incorporating local equalization coefficients into the diffusion updates. When CSI is not available to the nodes, we explain how the equalization coefficients can be evaluated from a pilot-assisted estimation process along with the main parameter estimation task of the network. We also examine the effect of channel estimation errors on the performance and convergence of the modified algorithms in terms of mean-square-error metric. We establish conditions under which the network is mean-square stable for both known and unknown CSI cases. The analysis reveal that when CSI is known, the modified diffusion algorithms are asymptotically unbiased and converge in the slow adaptation regime. In contrast, the parameter estimates will become biased when the CSI are obtained through pilot-aided channel estimation. Nevertheless, the size of the bias can be made small by increasing the number of pilot symbols or increasing the link SNR.

5.2 Network Signal Model

Consider a set of N sensor nodes that are distributed over a geographical area. At time instant $i \in \{0, 1, \cdots\}$, each node $k \in \{1, 2, \cdots, N\}$ collects measurement data $\{\boldsymbol{d}_k(i)$ and $\boldsymbol{u}_{k,i}\}$ that are related to an unknown parameter vector $w^o \in \mathbb{C}^{M\times 1}$ via the following relation:

$$\boldsymbol{d}_k(i) = \boldsymbol{u}_{k,i} w^o + \boldsymbol{v}_k(i) \tag{5.1}$$

where $\boldsymbol{d}_k(i) \in \mathbb{C}$, $\boldsymbol{u}_{k,i} \in \mathbb{C}^{1\times M}$ and $\boldsymbol{v}_k(i) \in \mathbb{C}$ are, respectively, the scalar measurement, the node's regression vector and the measurement noise. The variables in the linear regression model (5.1) are assumed to satisfy Assumption 2.1.

In this chapter, node ℓ is said to be a neighbor of node k if its distance from node k is less than a preset transmission range r_o, which for simplicity is assumed to remain constant over the given geographical area. The set of all the neighbors of node k, including node k itself, is denoted as before by $\mathcal{N}_k$. Nodes are allowed to communicate with their neighbors only, but due to channel impairments, certain links may fail. Hence, at any given time i, only a subset of the nodes in $\mathcal{N}_k$ can communicate with node k.

The objective of the network is to estimate the unknown parameter vector w^o in a distributed manner when the information exchange between the agents occurs over noisy wireless links that are also subject to fading and path loss. In particular, we assume that the transmit signal $\boldsymbol{\psi}_{\ell,i} \in \mathbb{C}^{M\times 1}$ from node $\ell \in \mathcal{N}_k\backslash\{k\}$ to node k experiences channel distortion of the following form (see Fig. 5.1):

$$\boldsymbol{\psi}_{\ell k,i} = \boldsymbol{h}_{\ell,k}(i)\sqrt{\frac{P_t}{r^{\alpha}_{\ell,k}}}\boldsymbol{\psi}_{\ell,i} + \boldsymbol{v}^{(\psi)}_{\ell k,i} \tag{5.2}$$

where $\boldsymbol{\psi}_{\ell k,i} \in \mathbb{C}^{M\times 1}$ is the distorted estimate received by node k, $\boldsymbol{h}_{\ell,k}(i) \in \mathbb{C}$ denotes the fading channel coefficient over the wireless link between nodes k and ℓ, $P_t \in \mathbb{R}^+$ is the transmit signal power, $r_{\ell,k} = r_{k,\ell} \in \mathbb{R}^+$ is the distance between nodes k and ℓ, $\alpha \in \mathbb{R}^+$ is the path loss exponent and $\boldsymbol{v}^{(\psi)}_{\ell k,i} \in \mathbb{C}^{M\times 1}$ is the additive noise vector with covariance matrix $\sigma^{(\psi)2}_{v,\ell k}\, I_M$. We define $\boldsymbol{\psi}_{kk,i} \triangleq \boldsymbol{\psi}_{k,i}$ to maintain consistency in the notation.

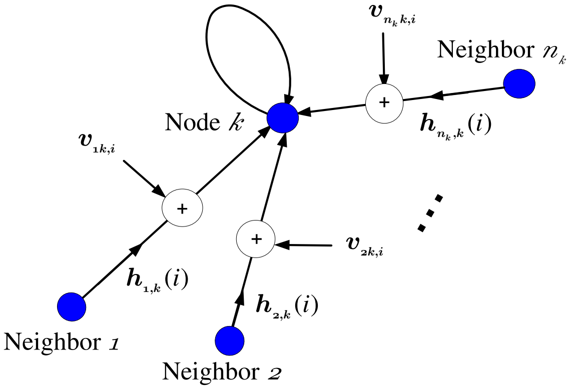

$\boldsymbol{v}_{n_k k,i}$
Neighbor n_k
Node k
$\boldsymbol{h}_{n_k,k}(i)$
$\boldsymbol{v}_{1k,i}$
+
+
+
$\boldsymbol{v}_{2k,i}$
$\boldsymbol{h}_{1,k}(i)$
$\boldsymbol{h}_{2,k}(i)$
Neighbor 1
Neighbor 2

Fig. 5.1 Node k receives distorted data from its $n_k = |\mathcal{N}_k|$ neighbors at time i. The data are affected by channel fading coefficients, $\boldsymbol{h}_{\ell,k}(i)$, and communication noise $\boldsymbol{v}_{\ell k,i}$. As we will explain in sequel, if the SNR between node k and some nodes $\ell_o \in \mathcal{N}_k$ falls below a threshold level, the received data from that node will be considered as noise and discarded.

Assumption 5.1. *The fading channel coefficients and the link noise in (5.2) satisfy the following conditions:*

a) The time-varying channel coefficients $\boldsymbol{h}_{\ell,k}(i)$ follow the Clark's model [125], which are modeled as independent circular Gaussian random variables with zero mean and variance $\sigma^2_{h,\ell k} = 1$.

b) $\{\boldsymbol{h}_{\ell,k}(i)\}$ are independent over space and i.i.d. over time.

c) The noise vectors $\{\boldsymbol{v}_{\ell k,i}^{(\psi)}\}$ *are zero-mean, i.i.d. in time and independent over space.*

d) The channel coefficients, $\boldsymbol{h}_{\ell,k_1}(i_1)$*, the noise vectors,* $\boldsymbol{v}_{\ell k_2,i_2}^{(\psi)}$*, the regression vectors,* $\boldsymbol{u}_{k_3,i_3}$ *and the measurement noise,* $\boldsymbol{v}_{k_4}(i_4)$*, are all mutually independent for all* k_j *and* i_j *with* $j \in \{1,2,3,4\}$*.*

We say a transmission from node ℓ to node k at time i is successful if the signal-to-noise ratio (SNR) between nodes ℓ and k, denoted by $\boldsymbol{\varsigma}_{\ell k}(i)$, exceeds some threshold level $\varsigma_{\ell k}^{o}$. The threshold level is defined as the SNR in the non-fading link scenario and is computed as:

$$\varsigma_{\ell k}^{o} \triangleq \frac{P_t}{\sigma_{v,\ell k}^{2(\psi)} r_o^{\alpha}} \tag{5.3}$$

In fading conditions, the instantaneous SNR is:

$$\boldsymbol{\varsigma}_{\ell k}(i) = \frac{|\boldsymbol{h}_{\ell,k}(i)|^2 P_t}{\sigma_{v,\ell k}^{2(\psi)} r_{\ell,k}^{\alpha}} \tag{5.4}$$

When transmission is successful, we have $\boldsymbol{\varsigma}_{\ell k}(i) \geq \varsigma_{\ell k}^{o}$ which amounts to the condition:

$$|\boldsymbol{h}_{\ell,k}(i)|^2 \geq \nu_{\ell,k} \tag{5.5}$$

where

$$\nu_{\ell,k} = \left(\frac{r_{\ell,k}}{r_o}\right)^{\alpha} \tag{5.6}$$

Since $\boldsymbol{h}_{\ell,k}(i)$ has a circular complex Gaussian distribution, the squared magnitude $|\boldsymbol{h}_{\ell,k}(i)|^2$ is exponentially distributed with parameter $\lambda = 1$ [126]. Considering this, the probability of successful transmission is then given by:

$$p_{\ell,k} = \Pr\Big(|\boldsymbol{h}_{\ell,k}(i)|^2 \geq \nu_{\ell,k}\Big) = e^{-\nu_{\ell,k}} \tag{5.7}$$

This expression shows that the probability of successful transmission decreases as the distance between two nodes increases. As such, the link between neighboring nodes is not guaranteed to be connected all the time, implying that the network topology is time-varying. Under this condition, we redefine the neighborhood set of node k as a time-varying set that consists of nodes $\ell \in \mathcal{N}_k$ for which at time i, $\boldsymbol{\varsigma}_{\ell k}(i)$ exceeds $\varsigma^o_{\ell k}$. In this way, the neighborhood set of each node k becomes random and we, therefore, denote it by $\boldsymbol{\mathcal{N}}_{k,i}$. This implies that $\boldsymbol{\mathcal{N}}_{k,i} \subset \mathcal{N}_k$ for all i.

5.3 Distributed Estimation over Wireless Channels

When the exchange of information between neighboring nodes is subject to noise, some degradation in performance occurs [53–56, 127, 128]; reference [53] shows how the combination weights $a_{\ell,k}$ can be adjusted to counter the effect of noisy links in the presence of additive noise. In this work, we move beyond these earlier analyses and study the performance of these algorithms over fading wireless channels. We also suggest modifications to the update equations to counter the effect of fading.

5.3.1 Diffusion Strategies over Wireless Channels

We begin our derivation from the distributed cost function, given by (2.11). When the communication channels between nodes are wireless, as shown in previous section, the neighborhood set of each node k will be a function of instantaneous SNR and changes over time. This means that each node k can only receive data from nodes $\ell \in \mathcal{N}_k$ whose instantaneous SNR, $\boldsymbol{\varsigma}_{\ell k}(i)$, exceeds the threshold $\varsigma^o_{\ell k}$. Therefore, in the cost function (2.11), we replace $\mathcal{N}_k$ and $b_{\ell,k}$ with $\boldsymbol{\mathcal{N}}_{k,i}$, and $\boldsymbol{b}_{\ell,k}(i)$ and arrive at:

$$J_k^{glob'}(w) = J_k(w) + \mathbb{E}\Big[\sum_{\ell \in \boldsymbol{\mathcal{N}}_{k,i}\backslash\{k\}} \boldsymbol{b}_{\ell,k}(i)\|w - w^o\|^2\Big] \tag{5.8}$$

With the exception of the variable w^o, this alternative cost at node k relies solely on information that is available to node k from its neighborhood. Now, each node k can apply a steepest-descent iteration to minimize its localized cost $J_k^{glob'}(w)$ as follow:

$$\begin{aligned} \boldsymbol{w}_{k,i} &= \boldsymbol{w}_{k,i-1} + \mu_k[\nabla_w J_k^{glob'}(w)]^* && (5.9)\\ &= \boldsymbol{w}_{k,i-1} + \mu_k(r_{du,k} - R_{u,k}\boldsymbol{w}_{k,i-1}) - \mu_k\mathbb{E}\Big[\sum_{\ell \in \boldsymbol{\mathcal{N}}_{k,i}\backslash\{k\}} \boldsymbol{b}_{\ell,k}(i)(\boldsymbol{w}_{k,i-1} - w^o)\Big] && (5.10) \end{aligned}$$

where μ_k is the step-size at node k. An adaptive implementation of (5.10) can be obtained by removing the expectation operator and replacing $\{r_{du,k}, R_{u,k}\}$ by their instantaneous approximations from (2.14). Doing so leads to the following recursion:

$$\boldsymbol{w}_{k,i} = \boldsymbol{w}_{k,i-1} + \mu_k \boldsymbol{u}^*_{k,i}(\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}\boldsymbol{w}_{k,i-1}) - \mu_k \sum_{\ell \in \boldsymbol{\mathcal{N}}_{k,i} \backslash \{k\}} \boldsymbol{b}_{\ell,k}(i)(\boldsymbol{w}_{k,i-1} - w^o) \tag{5.11}$$

Recursion (5.11) indicates that the update from $\boldsymbol{w}_{k,i-1}$ to $\boldsymbol{w}_{k,i}$ now involves adding two correction terms to $\boldsymbol{w}_{k,i-1}$. We can split these two correction terms by writing (5.11) as the following two steps involving an intermediate estimate:

$$\boldsymbol{\psi}_{k,i} = \boldsymbol{w}_{k,i-1} + \mu_k \boldsymbol{u}^*_{k,i}(\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}\boldsymbol{w}_{k,i-1}) \tag{5.12}$$

$$\boldsymbol{w}_{k,i} = \boldsymbol{\psi}_{k,i} - \mu_k \sum_{\ell \in \boldsymbol{\mathcal{N}}_{k,i} \backslash \{k\}} \boldsymbol{b}_{\ell,k}(i)(\boldsymbol{w}_{k,i-1} - w^o) \tag{5.13}$$

The first update (5.12) can be carried out by all nodes independent of knowledge of w^o. However, the unknown w^o still appears in (5.13). In network, with ideal channels, w^o is replaced by $\boldsymbol{\psi}_{\ell,i}$. Under fading condition, since $\boldsymbol{\psi}_{\ell,i}$ is not available at each node k, we, alternatively, use the equalized value of its transmitted version, $\boldsymbol{g}_{\ell,k}(i)\boldsymbol{\psi}_{\ell k,i}$, where the scalar gain $\boldsymbol{g}_{\ell,k}(i)$ is an equalization coefficient to be chosen to counter the effect of fading. Recall that $\boldsymbol{\psi}_{\ell k,i}$ is related to $\boldsymbol{\psi}_{\ell,i}$ via (5.2). We also replace $\boldsymbol{w}_{k,i-1}$ in (5.13) by $\boldsymbol{\psi}_{k,i}$. With these replacements, recursion (5.13) becomes

$$\boldsymbol{w}_{k,i} = \boldsymbol{\psi}_{k,i} - \mu_k \sum_{\ell \in \boldsymbol{\mathcal{N}}_{k,i} \backslash \{k\}} \boldsymbol{b}_{\ell,k}(i)(\boldsymbol{\psi}_{k,i} - \boldsymbol{g}_{\ell,k}(i)\boldsymbol{\psi}_{\ell k,i}) \tag{5.14}$$

By noting that $\boldsymbol{g}_{k,k}(i) = 1$ for all k, and introducing the time-varying coefficients $\boldsymbol{a}_{\ell,k}(i)$ as

$$\boldsymbol{a}_{\ell,k}(i) = \begin{cases} 1 - \sum_{j \in \boldsymbol{\mathcal{N}}_k \backslash \{k\}} \mu_k \boldsymbol{b}_{j,k}, & \ell = k \\ \mu_k \boldsymbol{b}_{\ell,k}(i), & \ell \in \boldsymbol{\mathcal{N}}_{k,i} \backslash \{k\} \\ 0, & \text{otherwise} \end{cases} \tag{5.15}$$

Considering these coefficients, we then arrive at the ATC DLMS strategy, Algorithms 5.1 for parameter estimation over wireless sensor networks.

One way to compute the equalization coefficients, $\boldsymbol{g}_{\ell,k}(i)$ is to employ the following zero-forcing type construction:

$$\boldsymbol{g}_{\ell,k}(i) = \frac{\boldsymbol{h}^*_{\ell,k}(i)}{|\boldsymbol{h}_{\ell,k}(i)|^2} \sqrt{\frac{r^{\alpha}_{\ell,k}}{P_t}} \tag{5.18}$$

Algorithm 5.1 : Diffusion ATC over Wireless Channels

$$\boldsymbol{\psi}_{k,i} = \boldsymbol{w}_{k,i-1} + \mu_k \boldsymbol{u}_{k,i}^* \big[\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}\boldsymbol{w}_{k,i-1}\big] \tag{5.16}$$

$$\boldsymbol{w}_{k,i} = \sum_{\ell \in \boldsymbol{\mathcal{N}}_{k,i}} \boldsymbol{a}_{\ell,k}(i)\boldsymbol{g}_{\ell,k}(i)\boldsymbol{\psi}_{\ell k,i} \tag{5.17}$$

Alternatively, if the noise variances $\sigma_{v,\ell k}^{2(\psi)}$ are known, then one also could use a minimum mean-square-error (MMSE) estimation scheme to obtain the equalization coefficients.

By switching the order of the adaption and combination steps in Algorithm 5.1, we obtain the CTA DLMS strategy listed below. In (5.19), $\boldsymbol{w}_{\ell k,i}$ is the estimate of the global parameter at node ℓ that undergoes similar path loss, fading and noise as $\boldsymbol{\psi}_{\ell k,i}$ described by (5.2).

Algorithm 5.2 : Diffusion CTA over Wireless Channels

$$\boldsymbol{\psi}_{k,i} = \sum_{\ell \in \boldsymbol{\mathcal{N}}_{k,i}} \boldsymbol{a}_{\ell,k}(i)\boldsymbol{g}_{\ell,k}(i)\boldsymbol{w}_{\ell k,i-1} \tag{5.19}$$

$$\boldsymbol{w}_{k,i} = \boldsymbol{\psi}_{k,i} + \mu_k \boldsymbol{u}_{k,i}^* \big[\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}\boldsymbol{\psi}_{k,i}\big] \tag{5.20}$$

The combination coefficients $\boldsymbol{a}_{\ell,k}(i)$ now are random and time-dependent because the neighborhoods are also evolving with time. Moreover, from (5.15) can be verified that they need to satisfy

$$\boldsymbol{a}_{\ell,k}(i) = 0 \ \text{ if } \ \ell \notin \boldsymbol{\mathcal{N}}_{k,i} \quad \text{and} \quad \sum_{\ell \in \boldsymbol{\mathcal{N}}_{k,i}} \boldsymbol{a}_{\ell,k}(i) = 1 \tag{5.21}$$

The randomness of $\boldsymbol{a}_{\ell,k}(i)$ can be further clarified by (5.5). The communication between nodes ℓ and k is successful if (5.5) is satisfied. Otherwise, the link between them fails. When the link fails, the associated combination weight $\boldsymbol{a}_{\ell,k}(i)$ must be set to zero, which in turn implies that other combination coefficients of node k need to be adjusted to preserve their sum to one, i.e., to satisfy (5.21). This suggests that the neighborhood set $\boldsymbol{\mathcal{N}}_{k,i}$ has to be updated whenever the received SNR crosses the threshold in either direction:

$$\boldsymbol{\mathcal{N}}_{k,i} = \left\{ \ell \in \{1, \cdots, N\} \,\middle|\, \boldsymbol{\varsigma}_{\ell k}(i) \geq \varsigma_{\ell k}^{o} \right\} \tag{5.22}$$

Motivated by these considerations, we propose the following dynamic structure to obtain combination weights over time:

$$\boldsymbol{a}_{\ell,k}(i) = \gamma_{\ell,k}\boldsymbol{\mathcal{I}}_{\ell,k}(i), \quad \boldsymbol{a}_{k,k}(i) = 1 - \sum_{\ell \in \boldsymbol{\mathcal{N}}_{k,i}\backslash\{k\}} \boldsymbol{a}_{\ell,k}(i) \tag{5.23}$$

where the $\gamma_{\ell,k}$ are positive fixed combination weights that node k assigns to its neighbors $\ell \in \boldsymbol{\mathcal{N}}_{k,i}$. To assure $\boldsymbol{a}_{k,k}(i) > 0$, these weights need to satisfy:

$$\sum_{\ell \in \boldsymbol{\mathcal{N}}_{k,i}\backslash\{k\}} \gamma_{\ell,k} < 1 \tag{5.24}$$

It can be verified that if each node k obtains the coefficients $\gamma_{\ell,k}$ for the time-invariant neighborhood set $\mathcal{N}_k$ according to well-known left-stochastic matrix combination rules, then the condition (5.24) will be always satisfied. In relation (5.23), $\boldsymbol{\mathcal{I}}_{\ell,k}(i)$ is defined as:

$$\boldsymbol{\mathcal{I}}_{\ell,k}(i) = \begin{cases} 1, & \text{if } \ell \in \boldsymbol{\mathcal{N}}_{k,i} \\ 0, & \text{otherwise} \end{cases} \tag{5.25}$$

When transmission from node ℓ to node k is successful $\boldsymbol{\mathcal{I}}_{\ell,k}(i) = 1$, otherwise, $\boldsymbol{\mathcal{I}}_{\ell,k}(i) = 0$. In this way, the entries $\boldsymbol{a}_{\ell,k}(i)$ satisfy condition (5.21). From (5.22) and (5.25), we see that the indicator operator, $\boldsymbol{\mathcal{I}}_{\ell,k}(i)$, is a random variable with bernoulli distribution for which the probability of success, $p_{\ell,k}$, is given by the exponential function (5.7).

5.3.2 Modeling the Impact of Channel Estimation Errors

In Algorithms 5.1 and 5.2, it is assumed that each node k knows the channel fading coefficients $\boldsymbol{h}_{\ell,k}(i)$, for $\ell \in \boldsymbol{\mathcal{N}}_{k,i}$ needed in (5.18). In practice, this information is usually recovered by means of an estimation step. Consequently, some additional estimation errors will be introduced into the network operation.

There are many ways by which the fading coefficients can be estimated. For example, we may assume that the transmitted data from node ℓ to node k carries two data types, namely, pilot symbols (training data) denoted by $\boldsymbol{s}_\ell(i)$, and data symbols $\boldsymbol{\psi}_{\ell,i}$ or $\boldsymbol{w}_{\ell,i}$. The training data are used for channel estimation and the data symbols are the intermediate estimates of the unknown parameter vector, w^o, which are used to update the network estimate at node k. According to (5.2), the received training data at node k and time i is affected by fading and noise, i.e.,

$$\boldsymbol{y}_{\ell,k}(i) = \boldsymbol{h}_{\ell,k}(i)\sqrt{\frac{P_t}{r^{\alpha}_{\ell,k}}}\boldsymbol{s}_{\ell}(i) + \boldsymbol{v}^{(y)}_{\ell,k}(i) \tag{5.26}$$

where $\boldsymbol{v}^{(y)}_{\ell,k}(i)$ is a zero-mean additive white Gaussian noise with variance $\sigma^{(y)2}_{v,\ell k}$. It is reasonable to assume that $\sigma^{(y)2}_{v,\ell k} = \sigma^{(\psi)2}_{v,\ell k}$.

The length of training symbols depends on the specific application requirements and the time scale variations of the channel. If the channels are quickly time-varying, a transmission block may include one training symbol for each data symbol. On the other hand, when the channels are slowly time-varying, one training symbol might be sufficient for several transmitted blocks. If we use a single training data, the least-squares estimation method gives the following estimate[1]:

$$\hat{\boldsymbol{h}}_{\ell,k}(i) = \sqrt{\frac{r^{\alpha}_{\ell,k}}{P_t}}\frac{\boldsymbol{s}^*_{\ell}(i)}{|\boldsymbol{s}_{\ell}(i)|^2}\boldsymbol{y}_{\ell,k}(i) \tag{5.27}$$

We assume that sensor $k \in \{1, 2, \cdots, N\}$ sends $\boldsymbol{s}_k(i) = 1$ as training symbols. Under this condition, $\hat{\boldsymbol{h}}_{\ell,k}(i) = \sqrt{\frac{r^{\alpha}_{\ell,k}}{P_t}}\boldsymbol{y}_{\ell,k}(i)$. From (5.26), it can be seen that $\boldsymbol{y}_{\ell,k}(i)$ is composed of the sum of two independent circular Gaussian random variables. It follows that $\boldsymbol{y}_{\ell,k}(i)$ has circular Gaussian distribution with zero mean and variance $\sigma^2_{h,\ell k}\frac{P_t}{r^{\alpha}_{\ell,k}} + \sigma^{(\psi)2}_{v,\ell k}$. From (5.27), we therefore conclude that $\hat{\boldsymbol{h}}_{\ell,k}(i)$ has circular Gaussian distribution with zero mean and variance $\sigma^2_{h,\ell k} + \frac{r^{\alpha}_{\ell,k}}{P_t}\sigma^{(\psi)2}_{v,\ell k}$, and $|\hat{\boldsymbol{h}}_{\ell,k}(i)|^2$ has exponential distribution with parameter

$$\lambda_{\ell,k} = \frac{1}{\sigma^2_{h,\ell k} + \frac{r^{\alpha}_{\ell,k}}{P_t}\sigma^{(\psi)2}_{v,\ell k}} \tag{5.28}$$

From here the probability of successful transmission from node ℓ to node k will be

$$p_{\ell,k} = \Pr\left(|\hat{\boldsymbol{h}}_{\ell,k}(i)|^2 \geq \nu_{\ell,k}\right) = e^{-\lambda_{\ell,k}\,\nu_{\ell,k}} \tag{5.29}$$

[1] The minimum number of training symbols to estimate one unknown is one. The estimation accuracy can be enhanced for slowly varying channels by increasing the number of training symbols at the cost of consuming more energy.

Considering the assumed training data and from (5.26) and (5.27), the instantaneous channel estimation error will be

$$\tilde{\boldsymbol{h}}_{\ell,k}(i) = \boldsymbol{h}_{\ell,k}(i) - \hat{\boldsymbol{h}}_{\ell,k}(i) \tag{5.30}$$

$$= -\sqrt{\frac{r_{\ell,k}^{\alpha}}{P_t}}\boldsymbol{v}_{\ell,k}^{(y)}(i) \tag{5.31}$$

Therefore, the variance of the estimation error can be given by:

$$\sigma_{\tilde{\boldsymbol{h}}_{\ell,k}}^2 = \mathbb{E}|\tilde{\boldsymbol{h}}_{\ell,k}(i)|^2 = \frac{r_{\ell,k}^{\alpha}}{P_t}\sigma_{v,\ell k}^{(\psi)2} \tag{5.32}$$

This expression shows that the power of the channel estimation error, $\sigma_{\tilde{\boldsymbol{h}}_{\ell,k}}^2$, decreases if the node transmit power increases or if the distance between node ℓ and k decreases. To reduce the channel estimation error, the alternative solution is to use more pilot data. Since the noise samples are i.i.d. in time, it can be shown that if the wireless channel remains invariant over transmission of n number of pilot data, then the estimation error variance will be scaled by a factor of $1/n$ [129].

We can now express (5.2) in terms of the estimated channels $\hat{\boldsymbol{h}}_{\ell,k}(i)$ and the channel estimation error as

$$\boldsymbol{\psi}_{\ell k,i} = \hat{\boldsymbol{h}}_{\ell,k}(i)\sqrt{\frac{P_t}{r_{\ell,k}^{\alpha}}}\boldsymbol{\psi}_{\ell,i} + \sqrt{\frac{P_t}{r_{\ell,k}^{\alpha}}}\tilde{\boldsymbol{h}}_{\ell,k}(i)\boldsymbol{\psi}_{\ell,i} + \boldsymbol{v}_{\ell k,i}^{(\psi)} \tag{5.33}$$

Under this condition, the equalization coefficients $\hat{\boldsymbol{g}}_{\ell,k}(i)$ are computed using the estimated channels $\hat{\boldsymbol{h}}_{\ell,k}(i)$, according to (5.18). Using these coefficients, the equalized received data at node k become:

$$\hat{\boldsymbol{g}}_{\ell k}(i)\boldsymbol{\psi}_{\ell k,i} = \left(1 + \hat{\boldsymbol{g}}_{\ell,k}(i)\sqrt{\frac{P_t}{r_{\ell,k}^{\alpha}}}\tilde{\boldsymbol{h}}_{\ell k}(i)\right)\boldsymbol{\psi}_{\ell,i} + \hat{\boldsymbol{g}}_{\ell,k}(i)\boldsymbol{v}_{\ell k,i}^{(\psi)} \tag{5.34}$$

Substituting the equalized data into (5.17), we obtain:

$$\boldsymbol{w}_{k,i} = \sum_{\ell\in\boldsymbol{\mathcal{N}}_{k,i}} \boldsymbol{a}_{\ell,k}(i)\boldsymbol{\psi}_{\ell,i} + \sum_{\ell\in\boldsymbol{\mathcal{N}}_{k,i}} \boldsymbol{e}_{\ell,k}(i)\boldsymbol{\psi}_{\ell,i} + \boldsymbol{v}_{k,i}^{(\psi)} \tag{5.35}$$

where

$$\boldsymbol{e}_{\ell,k}(i) = \boldsymbol{a}_{\ell,k}(i)\hat{\boldsymbol{g}}_{\ell,k}(i)\sqrt{\frac{P_t}{r^{\alpha}_{\ell,k}}}\tilde{\boldsymbol{h}}_{\ell k}(i) = -\boldsymbol{a}_{\ell,k}(i)\hat{\boldsymbol{g}}_{\ell,k}(i)\boldsymbol{v}^{(y)}_{\ell k}(i) \tag{5.36}$$

$$\boldsymbol{v}^{(\psi)}_{k,i} = \sum_{\ell\in\boldsymbol{\mathcal{N}}_{k,i}} \boldsymbol{a}_{\ell,k}(i)\hat{\boldsymbol{g}}_{\ell,k}(i)\boldsymbol{v}^{(\psi)}_{\ell k,i} \tag{5.37}$$

There are several important features in the combination step (5.35) that need to be highlighted. First, the combination coefficients used in this step are time varying. These time-varying coefficients, in addition to combining the exchanged information, model the link failure phenomena over the network. Second, $\{\hat{\boldsymbol{g}}_{\ell,k}(i)\}$ account for the effects of the fading channels. Using these variables and the control SNR mechanism introduced above, we can reduce the effects of the link noise. Third, in (5.35), $\{\boldsymbol{e}_{\ell k}(i)\}$ model the channel estimation errors, which allows us to examine the impact of these errors on the diffusion strategies.

5.4 Performance Analysis

In this section, we derive conditions under which the proposed diffusion strategy (5.16)–(5.17) are stable in the mean and mean square sense. We further derive expressions to characterize the MSD and EMSE of the algorithm during the transient phase and in steady-state. We focus on the ATC variant (5.16)–(5.17), because the same conclusions hold for the CTA strategy, represented by (5.19)-(5.20), with minor adjustments.

To derive a recursion for the mean error-vector of the network, we subtract w^o from both sides of (5.16) and (5.35) to obtain:

$$\tilde{\boldsymbol{\psi}}_{k,i} = (I - \mu_k \boldsymbol{u}^*_{k,i}\boldsymbol{u}_{k,i})\tilde{\boldsymbol{w}}_{k,i-1} - \mu_k \boldsymbol{u}^*_{k,i}\boldsymbol{v}_k(i) \tag{5.38}$$

$$\tilde{\boldsymbol{w}}_{k,i} = \sum_{\ell\in\boldsymbol{\mathcal{N}}_{k,i}} \boldsymbol{a}_{\ell,k}(i)\tilde{\boldsymbol{\psi}}_{\ell,i} + \sum_{\ell\in\boldsymbol{\mathcal{N}}_{k,i}} \boldsymbol{e}_{\ell,k}(i)\tilde{\boldsymbol{\psi}}_{\ell,i} + \sum_{\ell\in\boldsymbol{\mathcal{N}}_{k,i}} \boldsymbol{e}_{\ell,k}(i)w^o - \boldsymbol{v}^{(\psi)}_{k,i} \tag{5.39}$$

where $\tilde{\boldsymbol{\psi}}_{k,i}$ and $\tilde{\boldsymbol{w}}_{k,i}$ are the local error vector defined in the previous chapter. We collect the $\{\boldsymbol{a}_{\ell,k}(i)\}$ into a left-stochastic matrix $\boldsymbol{A}_i$ and stack the $\{\boldsymbol{e}_{\ell,k}(i)\}$ into an error matrix E_i. We, respectively, define the extended version of these matrices using Krocecker products as $\boldsymbol{\mathcal{A}}_i \triangleq \boldsymbol{A}_i \otimes I_M$ and $\boldsymbol{\mathcal{E}}_i \triangleq \boldsymbol{E}_i \otimes I_M$. We further introduce variables:

$$\boldsymbol{\mathcal{R}}_i \triangleq \text{diag}\Big\{\boldsymbol{u}_{1,i}^*\boldsymbol{u}_{1,i}, \cdots, \boldsymbol{u}_{N,i}^*\boldsymbol{u}_{N,i}\Big\} \tag{5.40}$$

$$\mathcal{M} \triangleq \text{diag}\Big\{\mu_1 I_M, \cdots, \mu_N I_M\Big\} \tag{5.41}$$

$$\boldsymbol{p}_i \triangleq \text{col}\Big\{\boldsymbol{u}_{1,i}^*\boldsymbol{v}_1(i), \cdots, \boldsymbol{u}_{N,i}^*\boldsymbol{v}_N(i)\Big\} \tag{5.42}$$

$$\boldsymbol{v}_i^{(\psi)} \triangleq \text{col}\Big\{\boldsymbol{v}_{1,i}^{(\psi)}, \cdots, \boldsymbol{v}_{N,i}^{(\psi)}\Big\} \tag{5.43}$$

$$\omega^o \triangleq \mathbb{1}_N \otimes w^o \tag{5.44}$$

where $\mathbb{1}_N$ is a column vector with length N and unit entries. We can now use (5.38) and (5.39) to verify that the following recursion holds for the network error vector:

$$\tilde{\boldsymbol{w}}_i = \boldsymbol{\mathcal{B}}_i\tilde{\boldsymbol{w}}_{i-1} - (\boldsymbol{\mathcal{A}}_i + \boldsymbol{\mathcal{E}}_i)^T\mathcal{M}\boldsymbol{p}_i + \boldsymbol{\mathcal{E}}_i{}^T\omega^o - \boldsymbol{v}_i^{(\psi)} \tag{5.45}$$

where $\tilde{\boldsymbol{w}}_i$ is the global weight error vector defined in (2.34) and

$$\boldsymbol{\mathcal{B}}_i = (\boldsymbol{\mathcal{A}}_i + \boldsymbol{\mathcal{E}}_i)^T(I - \mathcal{M}\boldsymbol{\mathcal{R}}_i) \tag{5.46}$$

5.4.1 Mean Convergence

Taking the expectation of (5.45) under Assumptions 2.1 and 5.1, we arrive at

$$\mathbb{E}[\tilde{\boldsymbol{w}}_i] = \mathcal{B}\,\mathbb{E}[\tilde{\boldsymbol{w}}_{i-1}] + \mathcal{E}^T\omega^o \tag{5.47}$$

where

$$\mathcal{B} \triangleq \mathbb{E}[\boldsymbol{\mathcal{B}}_i] = (\mathcal{A} + \mathcal{E})^T(I - \mathcal{M}\mathcal{R}) \tag{5.48}$$

$$\mathcal{A} \triangleq \mathbb{E}[\boldsymbol{\mathcal{A}}_i] = A \otimes I_M \tag{5.49}$$

$$\mathcal{E} \triangleq \mathbb{E}[\boldsymbol{\mathcal{E}}_i] = E \otimes I_M \tag{5.50}$$

$$\mathcal{R} \triangleq \mathbb{E}[\boldsymbol{\mathcal{R}}_i] = \text{diag}\Big\{R_{u,1}, \ldots, R_{u,N}\Big\} \tag{5.51}$$

To obtain (5.47), we use the fact that $\boldsymbol{v}_{k,i}$ is independent of $\boldsymbol{u}_{k,i}$ and $\mathbb{E}[\boldsymbol{v}_{k,i}] = 0$. Moreover, we have $\mathbb{E}[\boldsymbol{v}_i^{(\psi)}] = 0$ because $\hat{\boldsymbol{g}}_{\ell,k}(i)$ is independent of $\boldsymbol{v}_{\ell k,i}^{(\psi)}$ and $\mathbb{E}[\boldsymbol{v}_{\ell k,i}^{(\psi)}] = 0$. Considering the time-varying left-stochastic matrix $\boldsymbol{A}_i$ in (5.23), the (ℓ, k)-th element of A is computed as:

$$a_{\ell,k} = \begin{cases} \gamma_{\ell,k} p_{\ell,k}, & \text{if } \ell \in \mathcal{N}_k \backslash \{k\} \\ 1 - \sum_{\ell \in \mathcal{N}_k \backslash \{k\}} \gamma_{\ell,k} p_{\ell,k}, & \text{if } \ell = k \end{cases} \tag{5.52}$$

Observe that $A^T \mathbb{1} = \mathbb{1}$. The (ℓ, k)-th entries of matrix E for $\ell = k$ is zero and for $\ell \neq k$ can be computed as:

$$\begin{aligned} e_{\ell,k} &= -\mathbb{E}\Big[\boldsymbol{a}_{\ell,k}(i)\hat{\boldsymbol{g}}_{\ell,k}(i)\boldsymbol{v}^{(y)}_{\ell,k}(i)\Big] \\ &\overset{(i)}{=} -\gamma_{\ell,k}\,\mathbb{E}\Big[\boldsymbol{\mathcal{I}}_{\ell,k}(i)\hat{\boldsymbol{g}}_{\ell,k}(i)\boldsymbol{v}^{(y)}_{\ell,k}(i)\Big] \\ &\overset{(ii)}{=} -\gamma_{\ell,k}\,\mathbb{E}\Big[\hat{\boldsymbol{g}}_{\ell,k}(i)\boldsymbol{v}^{(y)}_{\ell,k}(i)\Big| |\hat{\boldsymbol{h}}_{\ell,k}(i)|^2 \geq \nu_{\ell,k}\Big] \\ &\overset{(iii)}{=} -\gamma_{\ell,k}\,\mathbb{E}\left[\left(\frac{\sqrt{\frac{r^\alpha}{P_t}}\boldsymbol{h}^*_{\ell,k}(i)\boldsymbol{v}^{(y)}_{\ell,k}(i) + \frac{r^\alpha}{P_t}|\boldsymbol{v}^{(y)}_{\ell,k}(i)|^2}{|\boldsymbol{h}_{\ell,k}(i) + \sqrt{\frac{r^\alpha}{P_t}}\boldsymbol{v}^{(y)}_{\ell,k}(i)|^2}\right)\middle| \left(\left|\boldsymbol{h}_{\ell,k}(i) + \sqrt{\frac{r^\alpha}{P_t}}\boldsymbol{v}^{(y)}_{\ell,k}(i)\right|^2 \geq \nu_{\ell,k}\right)\right] \end{aligned} \tag{5.53}$$

The equality in step (ii) follows from the fact that $\hat{\boldsymbol{g}}_{\ell,k}(i)$ is defined for $\ell \in \mathcal{N}_k \backslash \{k\}$ when $|\hat{\boldsymbol{h}}_{\ell,k}(i)|^2 \geq \nu_{\ell,k}$ for which $\boldsymbol{\mathcal{I}}_{\ell,k}(i) = 1$. We obtain (iii) by expressing $\hat{\boldsymbol{g}}_{\ell,k}(i)$ in terms of $\boldsymbol{h}_{\ell,k}(i)$ and $\boldsymbol{v}^{(y)}_{\ell,k}(i)$. Expression (5.53) indicates that $e_{\ell,k}$ is bounded.

Remark 5.1. *From the right hand side of (5.53), it can be verified that the value of the expectation is independent of time since the estimation error, $\boldsymbol{v}^{(y)}_{\ell,k}(i)$, and the channel coefficients, $\boldsymbol{h}_{\ell,k}(i)$, are assumed to be i.i.d. over time with fixed probability density functions.*

According to (5.47), if $\mathcal{B}$ is stable, then the network mean error vector converges to

$$b \triangleq \lim_{i \to \infty} \mathbb{E}[\tilde{\boldsymbol{w}}_i] = (I - \mathcal{B})^{-1}\mathcal{E}^T \omega^o \tag{5.54}$$

If $\hat{\boldsymbol{h}}_{\ell,k}(i) = \boldsymbol{h}_{\ell,k}(i)$ then $\mathcal{E} = 0$ and $\lim_{i \to \infty} \mathbb{E}[\tilde{\boldsymbol{w}}_i] = 0$, i.e., the algorithm will be asymptotically unbiased.

Let us now find conditions under which $\mathcal{B}$ is stable, i.e., conditions under which the spectral radius of $\mathcal{B}$, denoted by $\rho(\mathcal{B})$, is strictly less than one. We use the properties of the block-infinity norm $\|x\|_{b,\infty}$ [63] to establish the following relations:

$$\begin{aligned}\rho(\mathcal{B}) &\leq \|\mathcal{B}\|_{b,\infty}\\ &\leq \|(\mathcal{A}+\mathcal{E})^T\|_{b,\infty}\,\|(I-\mathcal{M}\mathcal{R})\|_{b,\infty}\\ &\leq \Big(\|\mathcal{A}^T\|_{b,\infty}+\|\mathcal{E}^T\|_{b,\infty}\Big)\,\|(I-\mathcal{M}\mathcal{R})\|_{b,\infty}\\ &= \Big(1+\|\mathcal{E}^T\|_{b,\infty}\Big)\|(I-\mathcal{M}\mathcal{R})\|_{b,\infty} \qquad (5.55)\end{aligned}$$

where in the last equality we used the fact that $\|\mathcal{A}^T\|_{b,\infty} = 1$ since A is left-stochastic. According to (5.55), $\rho(\mathcal{B})$ is bounded by one if

$$\|(I-\mathcal{M}\mathcal{R})\|_{b,\infty} < \frac{1}{1+\|\mathcal{E}\|_{b,\infty}} \qquad (5.56)$$

Since $I-\mathcal{M}\mathcal{R}$ is block diagonal and Hermitian, we have $\|(I-\mathcal{M}\mathcal{R})\|_{b,\infty} = \rho(I-\mathcal{M}\mathcal{R})$ [82]. The spectral radius of $I-\mathcal{M}\mathcal{R}$ will be less than $1/(1+\|\mathcal{E}\|_{b,\infty})$ if the absolute maximum eigenvalue of each of its blocks is strictly less than $1/(1+\|\mathcal{E}\|_{b,\infty})$. This condition is satisfied if at each node k the step-size μ_k is chosen as:

$$\frac{1-\frac{1}{1+\|\mathcal{E}\|_{b,\infty}}}{\lambda_{\max}(R_{u,k})} < \mu_k < \frac{1+\frac{1}{1+\|\mathcal{E}\|_{b,\infty}}}{\lambda_{\max}(R_{u,k})} \qquad (5.57)$$

where $\lambda_{\max}(\cdot)$ denotes the maximum eigenvalue of its matrix argument. This relation reveals that the mean-stability range of the algorithm reduces as the channel estimation error over the network increases. If the channel estimation error approaches zero[1], then $\|\mathcal{E}\|_{b,\infty} = 0$ and the stability condition reduces to $0 < \mu_k < \frac{2}{\lambda_{\max}(R_{u,k})}$, which is the mean stability range of diffusion LMS over ideal communication links [82]. A similar analysis can be carried out for the CTA diffusion strategy (5.19)-(5.20). We summarize the main result of the mean-convergence analysis of the developed DLMS algorithm in the following:

Theorem 5.1. *Consider diffusion strategies (5.16)–(5.17) or (5.19)–(5.20) with the space-time data (5.1) and (5.2) satisfying Assumption 2.1 and 5.1, respectively. Furthermore, assume that the channel coefficients are estimated using (5.27) with training symbols* $\boldsymbol{s}_k(i) =$

[1] The channel estimation error can be reduced by transmitting more number of pilot symbols and increasing the SNR during pilot transmission.

1. *Then this algorithms will be stable in the mean sense and its mean error vector converges to (5.54) if the step-sizes are chosen according to (5.57).*

5.4.2 Steady-State Mean-Square Performance

To study the mean-square performance of the algorithm, we need to determine the network variance relation [27,53,89]. This relation can be obtained by equating the weighted squared norms of both sides of (5.45), and taking expectations under Assumptions 2.1 and 5.1:

$$\begin{aligned}\mathbb{E}\|\tilde{\boldsymbol{w}}_i\|^2_{\Sigma} = \mathbb{E}\|\tilde{\boldsymbol{w}}_{i-1}\|^2_{\boldsymbol{\Sigma}_i{}'} + \mathbb{E}\Big[\omega^{o*}\,\boldsymbol{\mathcal{E}}_i^{*T}\Sigma\boldsymbol{\mathcal{E}}_i^T\,\omega^o\Big] + \mathbb{E}\Big[\boldsymbol{p}_i^*\mathcal{M}^T(\boldsymbol{\mathcal{A}}_i + \boldsymbol{\mathcal{E}}_i^{*T})\Sigma(\boldsymbol{\mathcal{A}}_i + \boldsymbol{\mathcal{E}}_i)^T\mathcal{M}\boldsymbol{p}_i\Big] \\ + 2\mathrm{Re}\Big\{\mathbb{E}[\omega^{o*}\boldsymbol{\mathcal{E}}_i^{*T}\Sigma\boldsymbol{\mathcal{B}}_i\tilde{\boldsymbol{w}}_{i-1}]\Big\} + \mathbb{E}[\boldsymbol{v}_i^{(\psi)*}\Sigma\boldsymbol{v}_i^{(\psi)}] \qquad (5.58)\end{aligned}$$

where for a vector x and a weighting matrix $\Sigma \geq 0$ with compatible dimensions $\|x\|^2_{\Sigma} = x^*\Sigma x$, and

$$\boldsymbol{\Sigma}_i' = \boldsymbol{\mathcal{B}}_i{}^*\Sigma\boldsymbol{\mathcal{B}}_i \qquad (5.59)$$

Under the independence assumption between $\tilde{\boldsymbol{w}}_{i-1}$ and $\boldsymbol{\mathcal{R}}_i$, it holds that

$$\mathbb{E}\Big(\|\tilde{\boldsymbol{w}}_{i-1}\|^2_{\boldsymbol{\Sigma}_i'}\Big) = \mathbb{E}\|\tilde{\boldsymbol{w}}_{i-1}\|^2_{\mathbb{E}[\boldsymbol{\Sigma}_i']} \qquad (5.60)$$

Using this equality in (5.58), we arrive at:

$$\begin{aligned}\mathbb{E}\|\tilde{\boldsymbol{w}}_i\|^2_{\Sigma} = \mathbb{E}\|\tilde{\boldsymbol{w}}_{i-1}\|^2_{\Sigma'} + \mathrm{Tr}(\mathbb{E}\Big[\boldsymbol{\mathcal{E}}_i^T\omega^{o*}\,\omega^o\boldsymbol{\mathcal{E}}_i^{*T}\Sigma]\Big) \\ + \mathrm{Tr}\Big(\mathbb{E}[(\boldsymbol{\mathcal{A}}_i + \boldsymbol{\mathcal{E}}_i)^T\mathcal{M}\boldsymbol{p}_i\boldsymbol{p}_i^*\mathcal{M}(\boldsymbol{\mathcal{A}}_i + \boldsymbol{\mathcal{E}}_i^{*T})\Sigma]\Big) \\ + 2\mathrm{Re}\Big\{\mathrm{Tr}(\mathbb{E}[\boldsymbol{\mathcal{B}}_i\tilde{\boldsymbol{w}}_{i-1}\omega^{o*}\boldsymbol{\mathcal{E}}_i^{*T}\Sigma])\Big\} + \mathrm{Tr}\Big(\mathbb{E}[\boldsymbol{v}_i^{(\psi)}\boldsymbol{v}_i^{(\psi)*}\Sigma]\Big) \qquad (5.61)\end{aligned}$$

where $\Sigma' = \mathbb{E}[\boldsymbol{\Sigma}_i']$. To compute (5.61), we introduce:

$$\mathcal{P} \triangleq \mathbb{E}[\boldsymbol{p}_i \boldsymbol{p}_i^*] = \text{diag}\Big\{\sigma_{v,1}^2 R_{u,1}, \cdots, \sigma_{v,N}^2 R_{u,N}\Big\} \tag{5.62}$$

$$\mathcal{R}_v \triangleq \text{diag}\Big\{R_{v,1} \cdots, R_{v,N}\Big\} \tag{5.63}$$

$$R_{v,k} \triangleq \mathbb{E}[\boldsymbol{v}_{k,i}^{(\psi)} \boldsymbol{v}_{k,i}^{(\psi)*}] = \sum_{\ell \in \mathcal{N}_k \backslash k} \mathbb{E}\Big[\boldsymbol{a}_{\ell,k}^2(i)\, |\hat{\boldsymbol{g}}_{\ell,k}(i)|^2\Big] R_{v,\ell k}^{(\psi)} \tag{5.64}$$

We show in Appendix A.7 how to compute the expectation term multiplying the $R_{v,\ell k}^{(\psi)}$ in (5.64).

To proceed, we assume Σ is partitioned into block entries of size $M \times M$ and let $\sigma = \text{bvec}(\Sigma)$ denote the vector that is obtained from the block vectorization of Σ. We shall write $\|\tilde{\boldsymbol{w}}_i\|_\Sigma^2$ and $\|\tilde{\boldsymbol{w}}_i\|_\sigma^2$ interchangeably to denote the same weighted square norm [27]. We now consider block matrices U and $V \in \mathbb{C}^{MN \times MN}$ with block size $M \times M$. Then the following relations hold [27]:

$$\text{bvec}(U \Sigma V) = (V^T \otimes_b U)\text{bvec}(\Sigma) \tag{5.65}$$

$$\text{Tr}(\Sigma X) = \Bigg(\text{bvec}\Big(X^T\Big)\Bigg)^T \text{bvec}(\Sigma) \tag{5.66}$$

Using properties (5.65) and (5.66), and considering real-valued Σ, the variance relation in (5.61) leads in steady-state to:

$$\lim_{i \to \infty} \mathbb{E}\|\tilde{\boldsymbol{w}}_i\|_\sigma^2 = \lim_{i \to \infty} \mathbb{E}\|\tilde{\boldsymbol{w}}_{i-1}\|_{\mathcal{F}\sigma}^2 + \gamma^T \sigma \tag{5.67}$$

where

$$\mathcal{F} = \mathbb{E}[\boldsymbol{\mathcal{B}}_i^T \otimes_b \boldsymbol{\mathcal{B}}_i^*] \tag{5.68}$$

and

$$\begin{aligned}\gamma = \lim_{i \to \infty} \Bigg\{&\mathbb{E}\Big[\boldsymbol{\mathcal{E}}_i^T \otimes_b \boldsymbol{\mathcal{E}}_i^*\Big] \text{bvec}\Big((\omega^o \omega^{o*})^T\Big) + \mathbb{E}\Big[(\boldsymbol{\mathcal{A}}_i + \boldsymbol{\mathcal{E}}_i)^T \otimes_b (\boldsymbol{\mathcal{A}}_i + \boldsymbol{\mathcal{E}}_i^{*T})^T\Big] \text{bvec}(\mathcal{M}\mathcal{P}^T\mathcal{M}) \\ &+ 2\text{Re}\{\mathbb{E}\Big[\boldsymbol{\mathcal{B}}_i \otimes_b \boldsymbol{\mathcal{E}}_i^*] \text{bvec}((b\, \omega^{o*})^T\}\Bigg\} + \text{bvec}(R_v^T)\end{aligned} \tag{5.69}$$

Considering (5.46) and (5.68) matrix $\mathcal{F}$ can be written as:

$$\begin{aligned}\mathcal{F} &= \mathbb{E}\Big\{\Big[(I-\mathcal{M}\boldsymbol{\mathcal{R}}_i)^T(\boldsymbol{\mathcal{A}}_i+\boldsymbol{\mathcal{E}}_i)\Big]\otimes_b\Big[(\boldsymbol{\mathcal{A}}_i+\boldsymbol{\mathcal{E}}_i^{*T})(I-\mathcal{M}\boldsymbol{\mathcal{R}}_i)\Big]\Big\}\\ &= \mathbb{E}\Big\{\Big[(I-\mathcal{M}\boldsymbol{\mathcal{R}}_i)^T\otimes_b(I-\mathcal{M}\boldsymbol{\mathcal{R}}_i)\Big]\Big[(\boldsymbol{\mathcal{A}}_i+\boldsymbol{\mathcal{E}}_i)\otimes_b(\boldsymbol{\mathcal{A}}_i+\boldsymbol{\mathcal{E}}_i^{*T})\Big]\Big\} \qquad (5.70)\end{aligned}$$

Since the entries of matrix $\boldsymbol{\mathcal{R}}_i$, i.e., the regression data $\boldsymbol{u}_{k,i}$, are independent of the entries of matrices $\boldsymbol{\mathcal{A}}_i$ and $\boldsymbol{\mathcal{E}}_i$, i.e., $\boldsymbol{a}_{\ell,k}(i)$ and $\boldsymbol{e}_{\ell,k}(i)$, matrix $\mathcal{F}$ in (5.70) can be written more compactly as:

$$\mathcal{F} = \bar{\mathcal{F}}\mathcal{D} \qquad (5.71)$$

where

$$\bar{\mathcal{F}} \triangleq \mathbb{E}\left[(I-\mathcal{M}\boldsymbol{\mathcal{R}}_i)^T\otimes_b(I-\mathcal{M}\boldsymbol{\mathcal{R}}_i)\right] \qquad (5.72)$$

$$\mathcal{D} \triangleq \mathbb{E}[\boldsymbol{\mathcal{D}}_i] = \mathbb{E}\left[(\boldsymbol{\mathcal{A}}_i+\boldsymbol{\mathcal{E}}_i)\otimes_b(\boldsymbol{\mathcal{A}}_i+\boldsymbol{\mathcal{E}}_i^{*T})\right] \qquad (5.73)$$

We can find an expression for $\bar{\mathcal{F}}$ if we assume that the regression data $\boldsymbol{u}_{k,i}$ are circular Gaussian–see equation (5.74) and Appendix A.8, where e_k is a basis vector in $\mathbb{R}^N$ with entry one at position k, $r_k = \text{vec}(R_{u,k})$, $\beta = 2$ for t real-valued data and $\beta = 1$ for complex-valued data.

$$\begin{aligned}\bar{\mathcal{F}} =&(I-\mathcal{M}\mathcal{R})^T\otimes_b(I-\mathcal{M}\mathcal{R}) + \left\{\sum_{k=1}^{N}\left[\text{diag}\Big((\text{vec}(\text{diag}(e_k))\Big)\right]\otimes\left[(\beta-1)(R_{k,u}^T\otimes R_{k,u})+r_k r_k^*\right]\right\}\\ &\times(\mathcal{M}\otimes_b\mathcal{M}) \qquad (5.74)\end{aligned}$$

A simplified expression can be found to compute $\bar{\mathcal{F}}$ without using the Gaussian assumption on the regression data provided that the following condition holds.

Assumption 5.2. *The channel estimation errors over the network are small enough such that the adaptation step-sizes in (5.57) can be chosen sufficiently small.*

In cases where the distribution of the regression data is unknown, under Assumption 5.2, the contributing terms depending on μ_k^2 can be neglected and as a result $\bar{\mathcal{F}}$ can approximated by

$$\bar{\mathcal{F}} \approx \left[(I - \mathcal{M}\mathcal{R})^T \otimes_b (I - \mathcal{M}\mathcal{R})\right] \tag{5.75}$$

In Appendix A.9, we show how to obtain the matrix $\mathcal{D}$ in (5.73) needed for computing $\mathcal{F}$ in (5.71). To evaluate γ, we use the following relations:

$$\mathbb{E}\Big[\boldsymbol{\mathcal{E}}_i^T \otimes_b \boldsymbol{\mathcal{E}}_i^*\Big] = \mathbb{E}\Big[\Big(\boldsymbol{E}_i^T \otimes \boldsymbol{E}_i^*\Big)\Big] \otimes I_{M^2} \tag{5.76}$$

$$\begin{aligned}\mathbb{E}\Big[(\boldsymbol{\mathcal{A}}_i + \boldsymbol{\mathcal{E}}_i)^T \otimes_b (\boldsymbol{\mathcal{A}}_i^T + \boldsymbol{\mathcal{E}}_i^*)\Big] = \Bigg(&\mathbb{E}\Big[(\boldsymbol{A}_i \otimes \boldsymbol{A}_i)^T\Big] + \mathbb{E}\Big[(\boldsymbol{A}_i^T \otimes \boldsymbol{E}_i^*)\Big] \\ &+ \mathbb{E}\Big[(\boldsymbol{E}_i \otimes \boldsymbol{A}_i)^T\Big] + \mathbb{E}\Big[\boldsymbol{E}_i^T \otimes \boldsymbol{E}_i^*\Big]\Bigg) \otimes I_{M^2}\end{aligned} \tag{5.77}$$

$$\mathbb{E}\Big[\boldsymbol{\mathcal{B}}_i \otimes_b \boldsymbol{\mathcal{E}}_i^*\Big] = \Bigg\{\Big(\mathbb{E}\Big[(\boldsymbol{A}_i \otimes \boldsymbol{E}_i)^*\Big] + \mathbb{E}\Big[\boldsymbol{E}_i^T \otimes \boldsymbol{E}_i^*\Big]\Big) \otimes I_{M^2}\Bigg\}\Bigg\{(I_{MN} - \mathcal{M}\mathcal{R}) \otimes_b I_{MN}\Bigg\} \tag{5.78}$$

The detail computation of each term on the righthand sides of (5.76)-(5.78) can be found in Appendix A.9.

To obtain mean-square error (MSE) steady state expressions for the network, we let i go to infinity and use expression (5.67) to write:

$$\lim_{i\to\infty} \mathbb{E}\|\tilde{\boldsymbol{w}}_i\|^2_{(I-\mathcal{F})\sigma} = \gamma^T\sigma \tag{5.79}$$

Since we are free to choose Σ and hence σ, we choose $(I - \mathcal{F})\sigma = \text{bvec}(\Omega)$, where Ω is another arbitrary positive semidefinite matrix. Doing so, we arrive at:

$$\lim_{i\to\infty} \mathbb{E}\|\tilde{\boldsymbol{w}}_i\|^2_{\Omega} = \gamma^T(I - \mathcal{F})^{-1}\text{bvec}(\Omega) \tag{5.80}$$

Each sub-vector of $\tilde{\boldsymbol{w}}_i$ corresponds to the estimation error at a particular node, e.g., $\tilde{\boldsymbol{w}}_{k,i}$ is the estimation error at node k. Using (5.80), the MSD at node k can be computed by choosing $\Omega = \{\text{diag}(e_k) \otimes I\}$, where e_k is a basis vector in $\mathbb{R}^N$ with entry one at position

k. Therefore, the MSD at node k can be obtained as:

$$\begin{aligned}\mathrm{MSD}_k &= \lim_{i\to\infty} \mathbb{E}\|\tilde{\boldsymbol{w}}_{k,i}\|^2 = \lim_{i\to\infty} \mathbb{E}\|\tilde{\boldsymbol{w}}_i\|^2_{\{\mathrm{diag}(e_k)\otimes I\}} \\ &= \gamma^T (I-\mathcal{F})^{-1} \mathrm{bvec}(\mathrm{diag}(e_k)\otimes I_M)\end{aligned} \tag{5.81}$$

The network MSD is defined as:

$$\mathrm{MSD} = \lim_{i\to\infty} \frac{1}{N}\sum_{k=1}^{N} \mathbb{E}\|\tilde{\boldsymbol{w}}_{k,i}\|^2 \tag{5.82}$$

where it can be computed from (5.80) by using $\Omega = \frac{1}{N} I_{MN}$. This leads to:

$$\begin{aligned}\mathrm{MSD} &= \lim_{i\to\infty} \frac{1}{N} \mathbb{E}\|\tilde{\boldsymbol{w}}_i\|^2 \\ &= \frac{1}{N}\gamma^T (I-\mathcal{F})^{-1} \mathrm{bvec}(I_{MN})\end{aligned} \tag{5.83}$$

In (5.81) and (5.83), it is assumed that $(I-\mathcal{F})$ is invertible. In what follows, we find conditions under which this assumption is satisfied. Using the properties of the Kronecker product and the sub-multiplicative property of norms, we can write:

$$\rho(\mathcal{F}) \leq \|\bar{\mathcal{F}}\mathcal{D}\|_{b,\infty} \leq \|\bar{\mathcal{F}}\|_{b,\infty}\|\mathcal{D}\|_{b,\infty} \tag{5.84}$$

We next show that $\bar{\mathcal{F}}$ from (5.75) is a block diagonal Hermitian matrix with block size $NM^2 \times NM^2$. To this end, we note that $I - \mathcal{M}\mathcal{R}$ is a block diagonal matrix with block size $M \times M$ and then use (5.75) to obtain:

$$\bar{\mathcal{F}} = \mathrm{diag}\Big\{ (I-\mu_1 R_{u,1})^T \otimes (I-\mathcal{M}\mathcal{R}), \cdots, (I-\mu_N R_{u,N})^T \otimes (I-\mathcal{M}\mathcal{R}) \Big\} \tag{5.85}$$

Moreover, $\bar{\mathcal{F}}$ is Hermitian because considering $\mathcal{R} = \mathcal{R}^*$, $\mathcal{M} = \mathcal{M}^T$, $\mathcal{R}\mathcal{M} = \mathcal{M}\mathcal{R}$, we will have

$$\begin{aligned}\bar{\mathcal{F}}^* &= \Big((I-\mathcal{M}\mathcal{R})^T\Big)^* \otimes_b (I-\mathcal{M}\mathcal{R})^* \\ &= (I-\mathcal{M}\mathcal{R})^T \otimes_b (I-\mathcal{M}\mathcal{R}) = \bar{\mathcal{F}}\end{aligned} \tag{5.86}$$

Now we can use the following lemma to bound the spectral radius of matrix $\mathcal{F}$ in (5.84).

Lemma 1. *Consider an $N \times N$ block diagonal Hermitian matrix $Y = \text{diag}\{Y_1, Y_2, \cdots, Y_N\}$, where each block Y_k is of size $M \times M$ and Hermitian. Then it holds that [82]:*

$$\|Y\|_{b,\infty} = \max_{1 \leq k \leq N} \rho(Y_k) = \rho(Y) \tag{5.87}$$

According to this lemma, since $\bar{\mathcal{F}}$ is block diagonal Hermitian, we can substitute its block maximum norm on the right hand side of relation (5.84) with its spectral radius and obtain:

$$\begin{aligned} \rho(\mathcal{F}) &\leq \rho\bigg((I - \mathcal{M}\mathcal{R})^T \otimes_b (I - \mathcal{M}\mathcal{R})\bigg) \, \|\mathcal{D}\|_{b,\infty} \\ &= \rho^2(I - \mathcal{M}\mathcal{R}) \, \|\mathcal{D}\|_{b,\infty} \end{aligned} \tag{5.88}$$

We then deduce that $\rho(\mathcal{F}) < 1$ if:

$$0 < \rho(I - \mathcal{M}\mathcal{R}) < \frac{1}{\sqrt{\|\mathcal{D}\|_{b,\infty}}} \tag{5.89}$$

Since $I - \mathcal{M}\mathcal{R}$ is a block-diagonal matrix, this condition will be satisfied for small step-sizes that also satisfy:

$$\frac{1 - \frac{1}{\sqrt{\|\mathcal{D}\|_{b,\infty}}}}{\lambda_{\max}(R_{u,k})} < \mu_k < \frac{1 + \frac{1}{\sqrt{\|\mathcal{D}\|_{b,\infty}}}}{\lambda_{\max}(R_{u,k})} \tag{5.90}$$

If the channel estimation error is small, then $\|\mathcal{E}\|_{b,\infty} \approx 0$ and $\mathcal{D} \approx \mathcal{A} \otimes_b \mathcal{A}$. Subsequently, $\|\mathcal{D}\|_{b,\infty} \approx 1$ and this mean-square stability condition reduces to $0 < \mu_k < \frac{2}{\lambda_{\max}(R_{u,k})}$ which is the mean-squares stability range of diffusion LMS over ideal communication links [82].

5.4.3 Mean-Square Transient Behavior

In this part, we derive expressions to characterize the mean-square convergence behavior of the diffusion algorithms over wireless networks with fading channels and noisy communication links. To derive these expressions, it is assumed that each node knows the CSI of its neighbors, and hence $\boldsymbol{E}_i = 0$ for all i. We then use (5.58) and consider

$\boldsymbol{w}_{k,-1} = 0,\ \forall k \in \{1, \cdots, N\}$ to arrive at:

$$\|\tilde{\boldsymbol{w}}_i\|^2_\sigma = \|w^o\|^2_{\hat{\mathcal{F}}^{i+1}\sigma} + \bar{\gamma}^T \sum_{j=0}^{i} \hat{\mathcal{F}}^j \sigma \tag{5.91}$$

where

$$\bar{\gamma} \triangleq \mathbb{E}\Big[\boldsymbol{\mathcal{A}}_i^T \otimes_b \boldsymbol{\mathcal{A}}_i\Big] \text{bvec}(\mathcal{M}\mathcal{P}^T\mathcal{M}) + \text{bvec}(\bar{R}_v^T) \tag{5.92}$$

$$\bar{\mathcal{R}}_v \triangleq \text{diag}\Big\{\bar{R}_{v,1} \cdots, \bar{R}_{v,N}\Big\} \tag{5.93}$$

$$\bar{R}_{v,k} \triangleq \mathbb{E}[\boldsymbol{v}^{(\psi)}_{k,i}\boldsymbol{v}^{(\psi)*}_{k,i}] = \sum_{\ell \in \mathcal{N}_k \backslash k} \mathbb{E}\Big[\boldsymbol{a}^2_{\ell,k}(i)\, |\boldsymbol{g}_{\ell,k}(i)|^2\Big] R^{(\psi)}_{v,\ell k} \tag{5.94}$$

Under this condition, and since $\boldsymbol{E}_i = 0$, $\hat{\mathcal{F}}$ can be expressed as:

$$\hat{\mathcal{F}} \approx \Big((I - \mathcal{M}\mathcal{R})^T \otimes_b (I - \mathcal{M}\mathcal{R})^*\Big)\, \mathbb{E}\Big[\boldsymbol{\mathcal{A}}_i^T \otimes_b \boldsymbol{\mathcal{A}}_i\Big] \tag{5.95}$$

Writing this relation for $i-1$ and computing $\|\tilde{\boldsymbol{w}}_i\|^2_\sigma - \|\tilde{\boldsymbol{w}}_{i-1}\|^2_\sigma$ leads to:

$$\|\tilde{\boldsymbol{w}}_i\|^2_\sigma = \|\tilde{\boldsymbol{w}}_{i-1}\|^2_\sigma + \|w^o\|^2_{\hat{\mathcal{F}}^i(I-\hat{\mathcal{F}})\sigma} + \bar{\gamma}^T \hat{\mathcal{F}}^i \sigma \tag{5.96}$$

By replacing σ with $\sigma_{\text{msd}_k} = \text{diag}(e_k) \otimes I_M$ and $\sigma_{\text{emse}_k} = \text{diag}(e_k) \otimes R_{u,k}$, we arrive at two recursions for the evolution of the MSD and EMSE over time:

$$\eta_k(i) = \eta_k(i-1) - \|w^o\|_{\hat{\mathcal{F}}^i(I-\hat{\mathcal{F}})\sigma_{\text{msd}_k}} + \bar{\gamma}^T \hat{\mathcal{F}}^i \sigma_{\text{msd}_k} \tag{5.97}$$

$$\zeta_k(i) = \zeta_k(i-1) - \|w^o\|_{\hat{\mathcal{F}}^i(I-\hat{\mathcal{F}})\sigma_{\text{emse}_k}} + \bar{\gamma}^T \hat{\mathcal{F}}^i \sigma_{\text{emse}_k} \tag{5.98}$$

We can find the learning curves of the network MSD and EMSE by averaging the nodes learning curves (5.97) and (5.98), respectively.

5.5 Simulation Results

In this section, we present computer experiments to illustrate the performance of the ATC diffusion strategy (5.16)–(5.17) in the estimation of the unknown parameter vector $w^o =$

$2[1+j1,\ -1+j1]^T$ over time-varying wireless channels. We consider a network with $N = 10$ nodes, which are randomly spread over a unit square area $(x, y) \in [0,\ 1]\times[0,\ 1]$, as shown in Fig. 5.2. We choose the transmit power of $P_t = 1$, nominal transmission range of $r_o = 0.4$ and the path-loss exponents $\alpha = 3.2$.

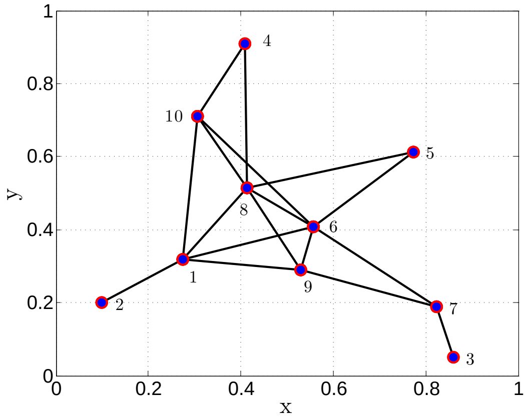


Fig. 5.2 This graph shows the topology of the wireless network at the initial phase. In this phase two nodes are connected if their distance is less than their transmission range, $r_o = 0.4$. In a fading scenario, this topology changes over time, meaning that each node may not communicate with all its neighbors all the time. A node may connect to or disconnect from its neighbors depending on the value of the indicator function.

For each node $k \in \{1, 2, \cdots, N\}$, we set $\mu_k = 0.01$ and $\boldsymbol{w}_{k,-1} = 0$. We adopt zero-mean Gaussian random distributions to generate $\boldsymbol{v}_k(i)$, $\boldsymbol{v}^{(\psi)}_{\ell k,i}$ and $\boldsymbol{u}_{k,i}$. The distribution of the communication noise power over the spatial domain is illustrated in Fig. 6.3. The regression data $\boldsymbol{u}_{k,i}$ have covariance matrices of the form $R_{u,k} = \sigma^2_{u,k} I_M$. The trace of the regression data, $\mathrm{Tr}(R_{u,k})$, and the variances of measurement noise, $\sigma^2_{v,k}$, are illustrated in Fig. 5.4. The distribution of the communication noise power over the spatial domain is illustrated in Fig. 5.3. The exchanged data between nodes experience distortion characterized by (5.2). At time i, the link between nodes ℓ and k fails with probability $1 - p_{\ell,k}$. We obtain $\gamma_{\ell,k}$ using the relative-degree combination rule [33, 82], i.e.,

$$\gamma_{\ell,k} = \begin{cases} \frac{|\mathcal{N}_\ell|}{\sum_{m\in\mathcal{N}_k} |\mathcal{N}_m|}, & \text{if } \ell \in \mathcal{N}_k \\ 0, & \text{otherwise} \end{cases} \tag{5.99}$$

and update $\boldsymbol{A}_i$ it at each time i according to the introduced combination rule (5.23).

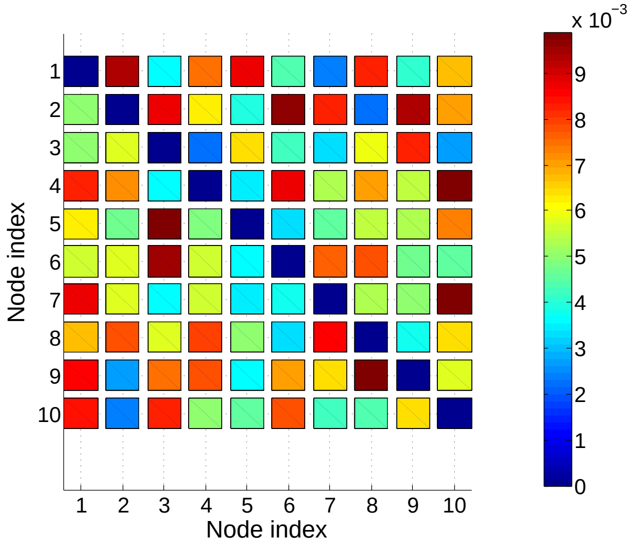


Fig. 5.3 Communication noise power over the network

Figures 5.5 and 5.6 show the network MSD in transient and steady-sate regimes, where the simulation curves are obtained from the average of 500 independent runs. In these figures, we compare the performance of the algorithm over wireless channels for different CSI cases at the receiving nodes. In particular, we first examine the performance of the algorithm with perfect CSI, where each node k knows the CSI of all its neighbors. We then consider scenarios where nodes do not have access to the CSI of their neighbors and obtain this information using one and two samples pilot data. For comparison purposes, we also illustrate the performance of ATC diffusion over ideal communication links in which the communication links between nodes are error-free, i.e., where for each node k, $\psi_{\ell k,i} = \psi_{\ell,i}$ for all i.

The best performance in these experiments belongs to the diffusion strategy that runs over network with ideal communication links. As expected, the diffusion strategy with perfect CSI knowledge outperforms diffusion strategy with channel estimation using one or two samples pilot data, respectively, by 7dB and 5dB. In particular, the steady-sate mean-square performance of the algorithm improves almost by 3dB for an additional sample of pilot data used for channel estimation. Therefore, if the wireless channels are slowly-varying, by using a larger number of pilot data, it is possible to approach the performance of the diffusion strategy algorithm with perfect CSI.

In Fig. 5.7, we compare the performance of diffusion strategies for different ranges of

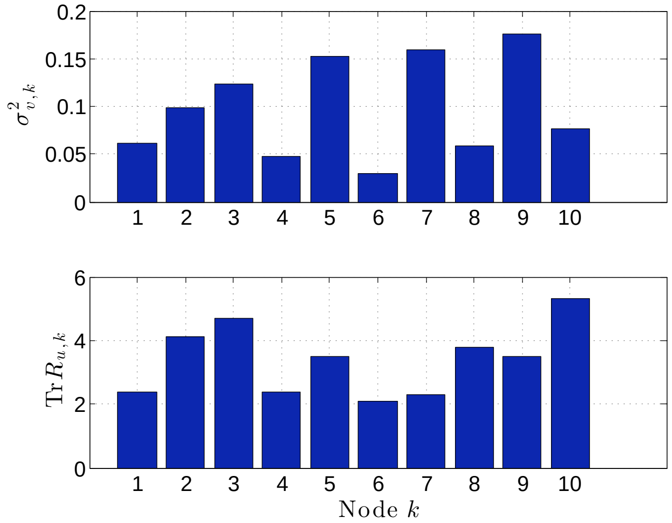


Fig. 5.4 Network energy profile

SNR over the network. We also make some comparisons between the cooperative and non-cooperative networks where in the latter case the network runs a stand-alone LMS filter at each node, which is equivalent to running the diffusion strategy with $\boldsymbol{A}_i = I$. In Fig. 5.7, the SNR index $n \in \{1, 2, \cdots, 7\}$ over the x-axis refers to the n-th network SNR distribution, as obtained by uniformly scaling up the initial SNR distribution over the network by 5dB for each increment in the integer n, as represented by $\mathrm{SNR}_n = \mathrm{SNR}_{\mathrm{ini}} + 5n\,(\mathrm{dB})$, where $\mathrm{SNR}_{\mathrm{ini}}$ are the SNR of the connected nodes illustrated in Fig. 5.2, and are obtained from uniformly distributed random variables in the range between [5 10]dB.

As shown in Fig. 5.7, the performance of non-cooperative adaptation and DLMS with ideal communication links remains invariant with changes in the SNR values. This is expected since the performance of the DLMS in these cases is not affected by the communication noise, $\boldsymbol{v}^{(\psi)}_{\ell k,i}$ and $\boldsymbol{v}^{(y)}_{\ell k,i}$. In comparison, the performance of the modified diffusion strategy over wireless links depends on the CSI. As the knowledge about the network CSI increases, the performance improves. From this result, we observe that at low SNR the performance discrepancies between diffusion with perfect CSI and diffusion with channel estimation is larger compared to high SNR scenarios. This difference in performance can be reduced by using more pilot data to estimate the channel coefficients in each time slot. In addition, at very low SNR, we see that the non-cooperative case outperforms the modified diffusion strategy. This result suggests that in wireless networks with high levels of communication noise at all nodes (e.g., when the nodes transmit power is very low), to maintain a satisfactory performance level the network must switch to the non-cooperative mode. It also can be concluded that if the transmit power of some nodes is below some threshold value, these nodes should go to a sleep mode in order to avoid error propagation over the network.

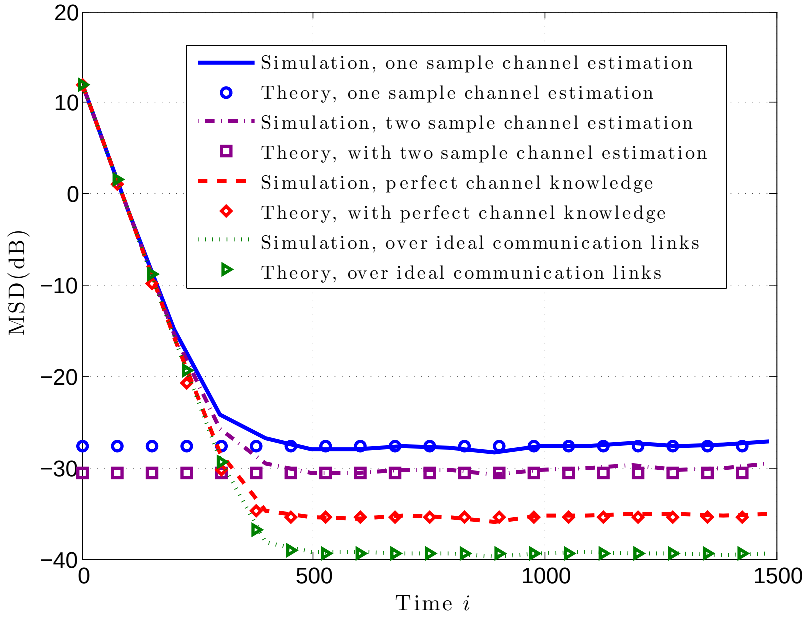


Fig. 5.5 Learning curves of the network in terms of MSD and EMSE.

5.6 Summary

We studied and examined the performance of DLMS algorithms over wireless networks, where the communication links between nodes are impaired by fading, path-loss and noise. We proposed new DLMS algorithms that are able to operate over time-varying wireless channels and under link-failure conditions. We introduced a simple strategy to update the network combination matrix over time when the network topology changes due to link failure or due to creation of new links. Our analytical finding indicate that the modified DLMS algorithms remain stable and converge in the mean and mean-square sense by proper CSI monitoring over the network provided that the network optimization step-sizes are sufficiently small. The analysis also reveals that the performance of the proposed DLMS

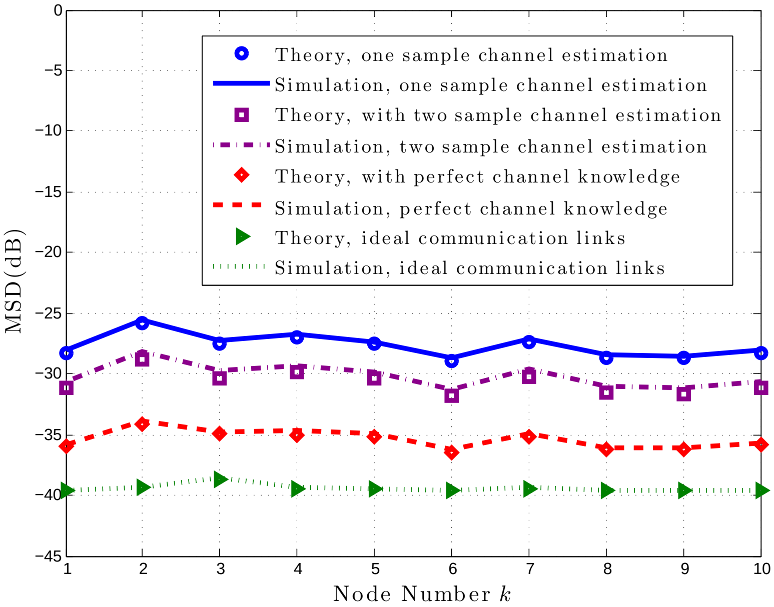


Fig. 5.6 Steady-state MSD over the network.

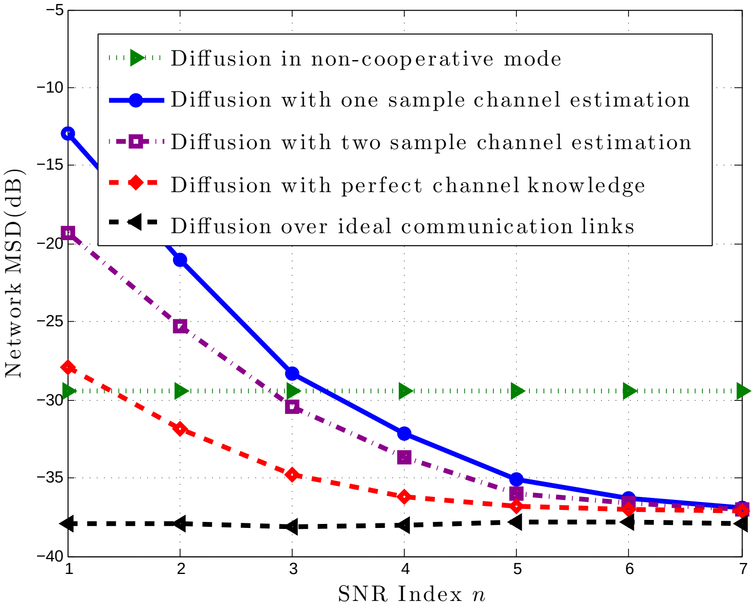


Fig. 5.7 The network performance comparison with non-cooperative diffusion LMS and with diffusion LMS over ideal communication links.

algorithms significantly depends on the network CSI and communication noise power over the network. We carried out computer experiments that show the effectiveness of the modified DLMS algorithms and verify the analytical findings.

As we have shown in this chapter, the performance of DLMS algorithms over ideal communication channel is far superior than that of DLMS over fading wireless channels. In the following chapter, in order to improve to performance of the proposed DLMS algorithms over fading channels and reduce the performance gap between these two scenarios, we will find the optimal value of the network combination matrix and propose an adaptive scheme to obtain their entries in real-time.

Chapter 6

Optimal Combination Weights over Wireless Sensor Networks

In this chapter, we study the performance of the DLMS algorithms developed in the previous chapter for different choices of left-stochastic combination matrices. In particular, we consider sensor networks, where the exchanged data between nodes are distorted by wireless channel impairments such as fading, path loss and noise. Under this condition, the selection of the entries of left-stochastic matrix $\boldsymbol{A}_i$ can significantly impact the accuracy of the network estimates. Our computer experiments show that an improper choice of combination weights can drastically deteriorate the performance of DLMS algorithms such that their steady-state MSE will become larger than the non-cooperative counterparts. To address this issue, we formulate a convex optimization problem using an upper bound of the mean-squares deviation (MSD) of network and obtain a closed form solution for the problem that gives the optimal combination weights and leads to substantial performance improvement. We also propose an adaptive scheme to obtain the optimal weights in real-time along the main estimation task of the network.

6.1 Introduction

The performance of adaptive networks highly depends on a left-stochastic matrix of diffusion strategies which dictates the flow of exchange information between nodes. From the network stability standpoint, this matrix needs to satisfy only a few conditions,i.e., its entries must be positive for neighboring nodes, all remaining entries must be set to zero and the entries on each column must add up to one [27]. Therefore, there are many choices of left stochastic matrices over a network for data fusion [27, 64, 68, 70]. However, not all of these choices will lead to a desired performance result unless additional information such as node's energy and noise profile are taken into account for their construction. For instance, if the measurement accuracy of a particular node over the network is low, the corresponding combination weights assigned it should be small to avoid error prorogation [64]. Based on this observation, we can view the entries of a left-stochastic combination matrix as variables of an optimization problem, which its solution can lead to a desired performance result over the network.

Over the past few years, several combination rules have been proposed for the fusion of the exchanged data between agents, especially in the context of consensus-based iterations [60–62], such as the maximum-degree rule and the Metropolis rule. While these schemes focus on improving the convergence behavior of the algorithms, they ignore the variations in noise profile and link status across the network, which can cause substantial performance degradation [63]. Recently, some studies considered the variation in measurement noise profile in obtaining the optimal weights over the network. For instance, the work in [33], defined a nonlinear and non-convex optimization problem which incorporate the variances of the measurement noise and its solution is pursued numerically. In [64], the authors reformulated this problem as a convex optimization and arrived at a closed-form solution to design the combination weights. Authors in [53] studied a more general scenario, where in addition to measurement noise, they also consider link communication noise in their design. In particular, they formulated an optimization problem that uses the node energy profile as well as the communication noise to obtain the optimal weights over the network. In these works, it was assumed that the links between nodes are always active, regardless of the nodes mobility, communication noise power and the fading conditions.

In this chapter, we consider a more general scenario to design the optimal combination weights. In our formulation, we consider a wireless sensor network where the links between agents (sensor nodes) are affected by fading and path loss in addition to communication noise, and as a consequence the link connectivity status and the network topology vary over time. To find the optimal combination weights, we formulate an optimization problem using an upper-bound approximation of the network MSD that can be solved in closed-form and gives the desired performance improvement. The obtained solution, nevertheless, requires the knowledge of the input correlation matrices, and the variances of the measurement and channel noise, which are often unknown in practice. We subsequently introduce an adaptive computation scheme that relies on the instantaneous data approximations to alternatively find the optimal combination weights over the network. In this way, each node besides running the standard adaptation layer to solve the desired distributed estimation, also runs a second adaptation layer to adjust its combination weights over time.

6.2 Mean-Square Performance

We consider the ATC diffusion algorithm given by (5.16)–(5.17) from the previous chapter. Our objective is to design the optimal combination matrix $\boldsymbol{A}_i$ that minimizes the network mean estimation errors. For mathematical tractability, in this chapter, we assume that the channel coefficients are all known *a-priori*. Therefore, the channel estimation error, $\mathbb{E}_i$, will be zero. Under such conditions, the network error vector given by (5.45) takes the form:

$$\tilde{\boldsymbol{w}}_i = \boldsymbol{\mathcal{B}}_i \tilde{\boldsymbol{w}}_{i-1} - \boldsymbol{\mathcal{A}}_i^T \mathcal{M} \boldsymbol{p}_i - \boldsymbol{v}_i^{(\psi)} \tag{6.1}$$

where

$$\boldsymbol{\mathcal{B}}_i = \boldsymbol{\mathcal{A}}_i^T (I - \mathcal{M} \boldsymbol{\mathcal{R}}_i) \tag{6.2}$$

$$\boldsymbol{\mathcal{A}}_i \triangleq \boldsymbol{A}_i \otimes I_M \tag{6.3}$$

$$\boldsymbol{\mathcal{R}}_i \triangleq \text{diag}\Big\{ \boldsymbol{u}_{1,i}^* \boldsymbol{u}_{1,i}, \cdots, \boldsymbol{u}_{N,i}^* \boldsymbol{u}_{N,i} \Big\} \tag{6.4}$$

$$\mathcal{M} \triangleq \text{diag}\Big\{ \mu_1 I_M, \cdots, \mu_N I_M \Big\} \tag{6.5}$$

$$\boldsymbol{p}_i \triangleq \text{col}\Big\{ \boldsymbol{u}_{1,i}^* \boldsymbol{v}_1(i), \cdots, \boldsymbol{u}_{N,i}^* \boldsymbol{v}_N(i) \Big\} \tag{6.6}$$

$$\boldsymbol{v}_i^{(\psi)} \triangleq \text{col}\Big\{ \boldsymbol{v}_{1,i}^{(\psi)}, \cdots, \boldsymbol{v}_{N,i}^{(\psi)} \Big\} \tag{6.7}$$

We now find the variance relation of the network by computing the weighted squared norm of (6.1), and taking expectations of both sides under Assumptions 2.1 and 5.1:

$$\mathbb{E}\|\tilde{\boldsymbol{w}}_i\|_{\Sigma}^2 = \mathbb{E}\|\tilde{\boldsymbol{w}}_{i-1}\|_{\boldsymbol{\Sigma}_i{}'}^2 + \mathbb{E}\Big[\boldsymbol{p}_i^* \mathcal{M}^T \boldsymbol{\mathcal{A}}_i \Sigma \boldsymbol{\mathcal{A}}_i^T \mathcal{M} \boldsymbol{p}_i \Big] + \mathbb{E}[\boldsymbol{v}_i^{(\psi)*} \Sigma \boldsymbol{v}_i^{(\psi)}] \tag{6.8}$$

where

$$\boldsymbol{\Sigma}_i' = \boldsymbol{\mathcal{B}}_i{}^* \Sigma \boldsymbol{\mathcal{B}}_i \tag{6.9}$$

Under the independence assumption between $\boldsymbol{w}_{i-1}$ and $\boldsymbol{\mathcal{R}}_i$, it holds that

$$\mathbb{E}\Big(\|\tilde{\boldsymbol{w}}_{i-1}\|^2_{\boldsymbol{\Sigma}_i'}\Big) = \mathbb{E}\|\tilde{\boldsymbol{w}}_{i-1}\|^2_{\mathbb{E}[\boldsymbol{\Sigma}_i']} \tag{6.10}$$

Substituting this into (3.93), we arrive at:

$$\mathbb{E}\|\tilde{\boldsymbol{w}}_i\|^2_\Sigma = \mathbb{E}\|\tilde{\boldsymbol{w}}_{i-1}\|^2_{\Sigma'} + \mathrm{Tr}\Big(\mathbb{E}[\boldsymbol{\mathcal{A}}_i^T \mathcal{M}\boldsymbol{p}_i\boldsymbol{p}_i^*\mathcal{M}(\boldsymbol{\mathcal{A}}_i)\Sigma]\Big) + \mathrm{Tr}\Big(\mathbb{E}[\boldsymbol{v}_i^{(\psi)}\boldsymbol{v}_i^{(\psi)*}\Sigma]\Big) \tag{6.11}$$

where $\Sigma' = \mathbb{E}[\boldsymbol{\Sigma}_i']$. Considering real-valued Σ the variance relation in (6.11) reads as:

$$\mathbb{E}\|\tilde{\boldsymbol{w}}_i\|^2_\sigma = \mathbb{E}\|\tilde{\boldsymbol{w}}_{i-1}\|^2_{\mathcal{F}\sigma} + \gamma^T\sigma \tag{6.12}$$

where

$$\begin{align}
\sigma &\triangleq \mathrm{bvec}(\Sigma) \tag{6.13}\\
\sigma' &\triangleq \mathrm{bvec}(\Sigma') = \mathcal{F}\sigma \tag{6.14}\\
\mathcal{F} &= \mathbb{E}[\boldsymbol{\mathcal{B}}_i^T \otimes_b \boldsymbol{\mathcal{B}}_i^*] \tag{6.15}\\
\gamma &= \mathbb{E}[(\boldsymbol{\mathcal{A}}_i \otimes_b \boldsymbol{\mathcal{A}}_i)^T]\mathrm{bvec}(\mathcal{M}\mathcal{P}^T\mathcal{M}) + \mathrm{bvec}(R_v^T) \tag{6.16}\\
\mathcal{P} &\triangleq \mathbb{E}[\boldsymbol{p}_i\boldsymbol{p}_i^*] = \mathrm{diag}\Big\{\sigma_{v,1}^2 R_{u,1}, \cdots, \sigma_{v,N}^2 R_{u,N}\Big\} \tag{6.17}\\
\mathcal{R}_v &\triangleq \mathrm{diag}\Big\{R_{v,1} \cdots, R_{v,N}\Big\} \tag{6.18}\\
R_{v,k} &\triangleq \mathbb{E}[\boldsymbol{v}_{k,i}^{(\psi)}\boldsymbol{v}_{k,i}^{(\psi)*}] = \sum_{\ell\in\mathcal{N}_k}^{N} \mathbb{E}\Big[\boldsymbol{a}_{\ell,k}^2(i)\,|\boldsymbol{g}_{\ell,k}(i)|^2\Big] R_{v,\ell k}^{(\psi)} \tag{6.19}
\end{align}$$

From (6.12), it can be deduced that:

$$\mathbb{E}\|\tilde{\boldsymbol{w}}_i\|^2_\sigma = \mathbb{E}\|\tilde{\boldsymbol{w}}_{-1}\|^2_{\mathcal{F}^{i+1}\sigma} + \gamma^T\sum_{j=0}^{i}\mathcal{F}^j\sigma \tag{6.20}$$

Matrix $\mathcal{F}$ can be written as:

$$\mathcal{F} = \mathbb{E}\left\{\left((I - \mathcal{M}\boldsymbol{\mathcal{R}}_i)^T \otimes_b (I - \mathcal{M}\boldsymbol{\mathcal{R}}_i)^*\right)\boldsymbol{\mathcal{D}}_i\right\} \tag{6.21}$$

where

$$\boldsymbol{\mathcal{D}}_i = \boldsymbol{\mathcal{A}}_i \otimes_b \boldsymbol{\mathcal{A}}_i \tag{6.22}$$

For sufficiently small step-sizes, matrix $\mathcal{F}$ can be approximated by:

$$\mathcal{F} \approx \left[(I - \mathcal{M}\mathcal{R})^T \otimes_b (I - \mathcal{M}\mathcal{R})^*\right]\mathcal{D} \tag{6.23}$$

where $\mathcal{D} = \mathbb{E}[\boldsymbol{\mathcal{D}}_i]$. From this relation, it can be verified that the steady-state mean-squares error of the network highly depends on matrix $\boldsymbol{\mathcal{D}}_i$, which is constructed by $\boldsymbol{A}_i$.

6.3 Combination Weights over Fading Channels

Since, according to (5.22), the neighborhood of each node k changes over time, the static combination rules proposed in previous works [29,33,82,124] are not immediately applicable for DLMS over wireless networks. In these networks, the entries of the time-varying left-stochastic matrix $\boldsymbol{A}_i$ can be expressed as:

$$\boldsymbol{a}_{\ell,k}(i) = \gamma_{\ell,k}\boldsymbol{\mathcal{I}}_{\ell,k}(i) \tag{6.24}$$

where $\gamma_{\ell,k}$ are positive fixed combination weights that node k assigns to neighbors $\ell \in \boldsymbol{\mathcal{N}}_{k,i}$, and $\boldsymbol{\mathcal{I}}_{\ell,k}(i)$ is defined in (5.25). To give an idea as how to choose the combination weights over time, we rewrite the proposed combination rule (5.23) presented in Chapter 5:

$$\boldsymbol{a}_{\ell,k}(i) = \begin{cases} \gamma_{\ell,k}\boldsymbol{\mathcal{I}}_{\ell,k}(i), & \text{if } \ell \in \boldsymbol{\mathcal{N}}_{k,i}\backslash\{k\}k \\ 1 - \sum_{\ell \in \boldsymbol{\mathcal{N}}_{k,i}\backslash\{k\}} \boldsymbol{a}_{\ell,k}(i), & \text{if } \ell = k \\ 0, & \text{Otherwise} \end{cases} \tag{6.25}$$

where $\gamma_{\ell,k}$ must satisfy:

$$\sum_{\ell \in \boldsymbol{\mathcal{N}}_{k,i}\backslash\{k\}} \gamma_{\ell,k} < 1 \tag{6.26}$$

It can be verified that if each node k obtains the coefficients $\gamma_{\ell,k}$ according to well-known left-stochastic combination rules for the fixed neighborhood set $\mathcal{N}_k$, then condition (6.26) will be always satisfied.

The stability of the DLMS Algorithms in the mean and mean-square error sense, does not depend on the particular choice of combination matrix made in (6.25). Theoretically, other choices are possible as long as $\mathbb{E}[\boldsymbol{A}_i]$ yields a *left-stochastic* matrix. Intuitively, $\mathbb{E}[\boldsymbol{A}_i]$ will be a left-stochastic matrix if the instantaneous combination matrix $\boldsymbol{A}_i$ satisfies $\sum_{\ell\in\boldsymbol{\mathcal{N}}_{k,i}}\boldsymbol{a}_{\ell,k}(i)=1$ for all i. Mathematically the claim can be stated as follows. If $\forall i\in\{0,1,\ldots\}$ and $k\in\{1,\cdots,N\}$, $\sum_{\ell\in\boldsymbol{\mathcal{N}}_{k,i}}\boldsymbol{a}_{\ell,k}(i)=1$ then $\mathbb{E}\Big[\sum_{\ell\in\boldsymbol{\mathcal{N}}_{k,i}}\boldsymbol{a}_{\ell,k}(i)\Big]=1$. This can be readily proven as follows:

$$\begin{aligned}
\mathbb{E}\Bigg[\sum_{\ell\in\boldsymbol{\mathcal{N}}_{k,i}}\boldsymbol{a}_{\ell,k}(i)\Bigg] &= \mathbb{E}\Bigg[\sum_{\ell\in\boldsymbol{\mathcal{N}}_{k,i}}\boldsymbol{a}_{\ell,k}(i)\boldsymbol{\mathcal{I}}_{\ell,k}(i)\Bigg] \\
&= \lim_{i\to\infty}\frac{1}{i}\sum_{j=1}^{i}\sum_{\ell\in\boldsymbol{\mathcal{N}}_{k,i}}\boldsymbol{a}_{\ell,k}(j)\boldsymbol{\mathcal{I}}_{\ell,k}(j) && (6.27) \\
&= \lim_{i\to\infty}\frac{1}{i}\Bigg[\sum_{\ell\in\boldsymbol{\mathcal{N}}_{k,i}}\boldsymbol{a}_{\ell,k}(1)\boldsymbol{\mathcal{I}}_{\ell,k}(1)+\cdots+\sum_{\ell\in\boldsymbol{\mathcal{N}}_{k,i}}\boldsymbol{a}_{\ell,k}(i)\boldsymbol{\mathcal{I}}_{\ell,k}(i)\Bigg] && (6.28) \\
&= 1 && (6.29)
\end{aligned}$$

Based on this result, the earlier combination rules proposed for DLMS can be used in wireless networks with the link failure phenomenon if their value recalculated at each time i for the new neighborhood $\boldsymbol{\mathcal{N}}_{k,i}$ and satisfy (5.21). For instance, the Metropolis combination rule [130] in networks with time-varying topologies takes the form:

$$\boldsymbol{a}_{\ell,k}(i)=\begin{cases}\dfrac{1}{\max\Big\{|\boldsymbol{\mathcal{N}}_{k,i}|,|\boldsymbol{\mathcal{N}}_{\ell,i}|\Big\}} & \text{if } \ell\in\boldsymbol{\mathcal{N}}_{k,i}\backslash\{k\} \\ 1-\sum_{\ell\in\boldsymbol{\mathcal{N}}_{k,i}\backslash\{k\}}\boldsymbol{a}_{\ell,k}(i) & \text{if } \ell=k \\ 0 & \text{otherwise}\end{cases} \tag{6.30}$$

and the Laplacian combination rule [70] will change to:

$$\boldsymbol{a}_{\ell,k}(i)=\begin{cases}\dfrac{1}{\max\left\{|\boldsymbol{\mathcal{N}}_{k,i}|,k\in\{1,\cdots,N\}\right\}} & \text{if}\,\ell\in\boldsymbol{\mathcal{N}}_{k,i}\backslash\{k\}\\ 1-\sum_{\ell\in\boldsymbol{\mathcal{N}}_{k,i}\backslash\{k\}}\boldsymbol{a}_{\ell,k}(i) & \text{if}\,\ell=k\\ 0 & \text{otherwise}\end{cases}\tag{6.31}$$

In [53], the authors has proposed a relative variance combination rule that is optimal over network with fixed topologies and noisy communication links. Consequently, it cannot immediately be employed for networks with wireless channels and time varying topologies. According to the above results, the modified version of this combination rule that can be used under such conditions will be:

$$\boldsymbol{a}_{\ell,k}(i)=\begin{cases}\frac{\alpha^{-2}_{\ell,k}(i)}{\sum_{m\in\boldsymbol{\mathcal{N}}_{k,i}}\alpha^{-2}_{m,k}(i)}, & \text{if}\,\ell\in\boldsymbol{\mathcal{N}}_{k,i}\\ 0, & \text{otherwise}\end{cases}\tag{6.32}$$

where $\boldsymbol{\alpha}^2_{\ell,k}(i)$ is given by:

$$\boldsymbol{\alpha}^2_{\ell,k}(i)=\begin{cases}\mu^2_\ell\sigma^2_{v,\ell}\text{Tr}(R_{u,\ell})+M\sigma^{2(\psi)}_{v,\ell k}, & \text{if}\,\ell\in\boldsymbol{\mathcal{N}}_{k,i}\backslash\{k\}\\ \mu^2_k\sigma^2_{v,k}\text{Tr}(R_{u,k}), & \text{if}\,\ell=k\end{cases}\tag{6.33}$$

The variation of $\boldsymbol{\alpha}^2_{\ell,k}(i)$ over time is due to changes in the neighborhood set, $\boldsymbol{\mathcal{N}}_{k,i}$, which is caused by link failure over the network. This combination rule can be improved if we exploit the channel state information (CSI) while optimizing the algorithm performance over $\boldsymbol{A}_i$.

6.3.1 Optimal Combination Weights

As explained above, the performance of DLMS algorithms can be severely affected by the network left-stochastic combination matrices. Therefore, these matrices can be considered as free variables of an optimization procedure to minimize the steady-state mean-square estimation errors of the network. Motivated by this idea, in sequel, we formulate an optimization problem to find the optimal value of the left stochastic matrix $\boldsymbol{A}_i$, for each i, that minimizes the upper bound of the network steady-state MSD.

To obtain an upper bound of the network MSD, we use (6.20) to write

$$\begin{aligned}\eta &= \lim_{i\to\infty} \frac{1}{N}\mathbb{E}\|\tilde{\boldsymbol{w}}_i\|^2_{\text{bvec}(I_{MN})} \\ &= \lim_{i\to\infty} \frac{1}{N}\mathbb{E}\|\tilde{\boldsymbol{w}}_{-1}\|^2_{\mathcal{F}^{i+1}\text{bvec}(I_{MN})} + \lim_{i\to\infty}\frac{1}{N}\sum_{j=0}^{i}\text{Tr}\Big(\mathcal{B}^j\Big(\mathbb{E}[\boldsymbol{\mathcal{A}}_i^T\mathcal{M}\mathcal{P}\mathcal{M}\boldsymbol{\mathcal{A}}_i] + \mathcal{R}_v\Big)\mathcal{B}^{*j}\Big) \quad (6.34)\end{aligned}$$

Since matrix $\mathcal{B}$ is stable, $\mathcal{F} \approx \mathcal{B}^T \otimes_b \mathcal{B}^*$ will be stable and as a result:

$$\lim_{i\to\infty} \|\tilde{\boldsymbol{w}}_{-1}\|^2_{\mathcal{F}^{i+1}\text{bvec}(I_{MN})} = 0 \quad (6.35)$$

This holds irrespective to the choice of the left stochastic matrix $\boldsymbol{A}_i$ over time. Therefore, (6.34) reduces to:

$$\eta = \lim_{i\to\infty}\frac{1}{N}\sum_{j=0}^{i}\text{Tr}\Big(\mathcal{B}^j\Big(\mathbb{E}[\boldsymbol{\mathcal{A}}_i^T\mathcal{M}\mathcal{P}\mathcal{M}\boldsymbol{\mathcal{A}}_i] + \mathcal{R}_v\Big)\mathcal{B}^{*j}\Big) \quad (6.36)$$

Let $\|X\|_*$ denote the nuclear norm of a matrix X which is defined as the sum of its singular values:

$$\|X\|_* = \sum_j \sigma_j(X) \quad (6.37)$$

Using the properties of the nuclear norm, we can write:

$$\begin{aligned}\text{Tr}\Big(\mathcal{B}^j(\mathbb{E}(\boldsymbol{\mathcal{A}}_i^T\mathcal{M}\mathcal{P}\mathcal{M}\boldsymbol{\mathcal{A}}_i) + \mathcal{R}_v)\mathcal{B}^{*j}) &\overset{(i)}{=} \Big\|\mathcal{B}^j\Big(\mathbb{E}(\boldsymbol{\mathcal{A}}_i^T\mathcal{M}\mathcal{P}\mathcal{M}\boldsymbol{\mathcal{A}}_i) + \mathcal{R}_v\Big)\mathcal{B}^{*j}\Big\|_* \\ &\overset{(ii)}{\leq} \|\mathcal{B}^j\|_*\|\mathbb{E}(\boldsymbol{\mathcal{A}}_i^T\mathcal{M}\mathcal{P}\mathcal{M}\boldsymbol{\mathcal{A}}_i) + \mathcal{R}_v\|_*\|\mathcal{B}^{*j}\|_* \\ &\overset{(iii)}{\leq} \|\mathcal{B}\|_*^{2j}\ \text{Tr}(\mathbb{E}(\boldsymbol{\mathcal{A}}_i^T\mathcal{M}\mathcal{P}\mathcal{M}\boldsymbol{\mathcal{A}}_i) + \mathcal{R}_v) \\ &\overset{(iv)}{=} c^2\|\mathcal{B}\|_{b,\infty}^{2j}\ \text{Tr}\Big(\mathbb{E}(\boldsymbol{\mathcal{A}}_i^T\mathcal{M}\mathcal{P}\mathcal{M}\boldsymbol{\mathcal{A}}_i) + \mathcal{R}_v\Big) \\ &\overset{(v)}{\leq} c^2\|I - \mathcal{M}\mathcal{R}\|_{b,\infty}^{2j}\ \text{Tr}\Big(\mathbb{E}(\boldsymbol{\mathcal{A}}_i^T\mathcal{M}\mathcal{P}\mathcal{M}\boldsymbol{\mathcal{A}}_i) + \mathcal{R}_v\Big) \\ &\overset{(vi)}{=} c^2\Big[\rho(I - \mathcal{M}\mathcal{R}\Big)\Big]^{2j}\ \text{Tr}\Big(\mathbb{E}(\boldsymbol{\mathcal{A}}_i^T\mathcal{M}\mathcal{P}\mathcal{M}\boldsymbol{\mathcal{A}}_i) + \mathcal{R}_v\Big)\end{aligned} \quad (6.38)$$

where (i) holds because for any Hermitian positive definite matrix X, we have

$$\|X\|_* = \mathrm{Tr}(X) \tag{6.39}$$

Step (ii) is satisfied due to the sub-multiplicative property of nuclear norm, (iii) holds according to (6.39) and since

$$\|X\|_* = \|X^*\|_* \tag{6.40}$$

Step (iv) is satisfied since $\|X\|_*$ and $\|X\|_{b,\infty}$ are submultiplicative nroms and all such norms are equivalent ;therefore, $\|X\|_* = c\|X\|_{b,\infty}$, where c is a positive scaler. Inequality (v) is satisfied because, for ant time instant i, for the left stochastic matrix $\boldsymbol{A}_i$, we have $\rho(\boldsymbol{A}_i) = \rho(\boldsymbol{\mathcal{A}}_i) = \|\boldsymbol{\mathcal{A}}_i\|_{b,\infty} = 1$. Finally (vi) holds because for any block diagonal Hermitian matrices X, we have [82]:

$$\|X\|_{b,\infty} = \rho(X) \tag{6.41}$$

Substituting (vi) in (6.36), we obtain:

$$\begin{aligned}
\eta &\le \lim_{i\to\infty} \frac{c^2}{N} \mathrm{Tr}\Big(\mathbb{E}[\boldsymbol{\mathcal{A}}_i^T \mathcal{M}\mathcal{P}\mathcal{M}\boldsymbol{\mathcal{A}}_i] + \mathcal{R}_v\Big) \sum_{j=0}^{i} \Big[\rho(I - \mathcal{M}\mathcal{R}\Big)\Big]^{2j} \\
&= \frac{c^2}{N} \frac{\mathrm{Tr}\Big(\mathbb{E}[\boldsymbol{\mathcal{A}}_i^T \mathcal{M}\mathcal{P}\mathcal{M}\boldsymbol{\mathcal{A}}_i] + \mathcal{R}_v\Big)}{1 - [\rho(I - \mathcal{M}\mathcal{R}]^2}
\end{aligned} \tag{6.42}$$

Note that to arrive in (6.42), we used the geometric series formula [131] and considered the fact that $\mathbb{E}[\boldsymbol{\mathcal{A}}_i^T \mathcal{M}\mathcal{P}\mathcal{M}\boldsymbol{\mathcal{A}}_i]$ will be equal for all i. This upper bound will be minimized if the numerator is minimized. Thus, we obtain the following optimization problem:

$$\begin{aligned}
&\min_{A} \ \mathrm{Tr}(\mathbb{E}[\boldsymbol{\mathcal{A}}_i^T \mathcal{M}\mathcal{P}\mathcal{M}\boldsymbol{\mathcal{A}}_i] + \mathcal{R}_v) \\
&\qquad \text{subject to: } A\mathbb{1} = \mathbb{1}, \quad a_{\ell,k} \ge 0 \\
&\qquad\qquad a_{\ell,k} = 0 \ \text{ if } \ \ell \notin \mathcal{N}_k
\end{aligned} \tag{6.43}$$

Substituting the value of $\mathcal{R}_v$ from (6.18), we arrive at the identity:

$$\mathrm{Tr}(\mathbb{E}[\boldsymbol{\mathcal{A}}_i^T\mathcal{M}\mathcal{P}\mathcal{M}\boldsymbol{\mathcal{A}}_i]+\mathcal{R}_v)=\sum_{k=1}^{N}\sum_{\ell\in\mathcal{N}_k}\mathbb{E}[\boldsymbol{a}_{\ell,k}^2(i)]\Big\{\mu_\ell^2\sigma_{v,\ell}^2$$

$$\mathrm{Tr}(R_{u,\ell})+\mathbb{E}[|\boldsymbol{g}_{\ell,k}(i)|^2\Big||\boldsymbol{h}_{\ell,k}(i)|^2>\nu_{\ell,k}]\,\mathrm{Tr}(R_{v,\ell k}^{(\psi)})\Big\} \tag{6.44}$$

Considering the left-stochastic combination matrix structure given by (6.24), and assuming mutual independence between the entries of each column in $\boldsymbol{A}_i$, we obtain:

$$\mathbb{E}[\boldsymbol{a}_{\ell,k}^2(i)]=\gamma_{\ell,k}^2\mathbb{E}[\boldsymbol{\mathcal{I}}_{\ell,k}^2(i)]=\gamma_{\ell,k}^2 p_{\ell,k} \tag{6.45}$$

Substituting (6.45) into (6.44) and replacing the resulting the expression into (6.43) and decoupling the latter into N independent constraint minimization problems, we arrive at:

$$\min_{\gamma_{\ell,k}}\sum_{\ell\in\mathcal{N}_k}\gamma_{\ell,k}^2 p_{\ell,k}\Big\{\mu_\ell^2\sigma_{v,\ell}^2\mathrm{Tr}(R_{u,\ell})+|g_{\ell,k}|^2\mathrm{Tr}(R_{v,\ell k}^{(\psi)})\Big\}$$

$$\text{subject to:}\quad \gamma_{\ell,k}\geq 0,\ \sum_{\ell\in\mathcal{N}_k}p_{\ell,k}\gamma_{\ell,k}=1$$

$$\gamma_{\ell,k}=0\ \text{ if }\ \ell\notin\mathcal{N}_k \tag{6.46}$$

where

$$|g_{\ell,k}|^2=\mathbb{E}\Big[|\boldsymbol{g}_{\ell,k}(i)|^2\Big||\boldsymbol{h}_{\ell,k}(i)|^2>\nu_{\ell,k}\Big] \tag{6.47}$$

Note that the cost function (6.46) is convex. Therefore, forming the Lagrangian and applying the Karush-Kuhn-Tucker(KKT) conditions [120], we will obtain:

$$p_{\ell,k}\gamma_{\ell,k}=\begin{cases}\frac{p_{\ell,k}\alpha_{\ell,k}^{-2}}{\sum_{m\in\mathcal{N}_k}p_{m,k}\alpha_{m,k}^{-2}}, & \text{if}\,\ell\in\mathcal{N}_k\\ 0, & \text{otherwise}\end{cases} \tag{6.48}$$

where

$$\alpha_{\ell,k}^2=\begin{cases}\mu_\ell^2\sigma_{v,\ell}^2\mathrm{Tr}(R_{u,\ell})+M|g_{\ell,k}|^2]\sigma_{v,\ell k}^{2(\psi)}, & \text{if}\,\ell\in\mathcal{N}_k\backslash\{k\}\\ \mu_k^2\sigma_{v,k}^2\mathrm{Tr}(R_{u,k}), & \text{if}\,\ell=k\end{cases} \tag{6.49}$$

The combination rule (6.48) gives the entries of A, which is the average value of $\boldsymbol{A}_i$ that minimizes the network MSD. To achieve this minimum MSD using the DLMS algorithm, we need to know the optimal value of $\boldsymbol{A}_i$, not the optimal value of its average. We recall that $\mathbb{E}[\boldsymbol{\mathcal{I}}_{\ell,k}(i)] = p_{\ell,k}$. Therefore, the optimal entries of $\boldsymbol{A}_i$ can be derived from (6.48) as:

$$\gamma_{\ell,k}\boldsymbol{\mathcal{I}}_{\ell,k}(i) = \begin{cases} \frac{\boldsymbol{\mathcal{I}}_{\ell,k}(i)\alpha_{\ell,k}^{-2}}{\sum_{m\in\mathcal{N}_k}\boldsymbol{\mathcal{I}}_{m,k}(i)\alpha_{m,k}^{-2}}, & \text{if } \ell \in \mathcal{N}_k \\ 0, & \text{otherwise} \end{cases} \tag{6.50}$$

which is equivalent to:

$$\boldsymbol{a}_{\ell,k}(i) = \begin{cases} \frac{\alpha_{\ell,k}^{-2}}{\sum_{m\in\boldsymbol{\mathcal{N}}_{k,i}}\alpha_{m,k}^{-2}}, & \text{if } \ell \in \boldsymbol{\mathcal{N}}_{k,i} \\ 0, & \text{otherwise} \end{cases} \tag{6.51}$$

Note that the combination matrix obtained using this method is optimal at every time instant i.

6.3.2 Adaptive Combination Weights

In computing (6.49), we assumed that at each node k, the second order moments, $\sigma_{v,\ell}^2$, $R_{u,\ell}$, $|g_{\ell,k}|^2$ and $\sigma_{v,\ell k}^{2(\psi)}$ for $\ell \in \mathcal{N}_k$, are known *a-priori*. Such information, however, may not be available in practice. To overcome this difficulty, we here propose an adaptive scheme to compute $\boldsymbol{A}_i$ along with the iterations of DLMS in real-time.

Using the equalization coefficients $\boldsymbol{g}_{\ell,k}(i)$ and the identity (5.2), we can write:

$$\boldsymbol{g}_{\ell,k}(i)\boldsymbol{\psi}_{\ell k,i} = \boldsymbol{\psi}_{\ell,i} + \boldsymbol{g}_{\ell,k}(i)\boldsymbol{v}_{\ell k,i}^{(\psi)} \tag{6.52}$$

From (5.1), (5.16) and (6.52), we then obtain the following relation:

$$\mathbb{E}\|\boldsymbol{g}_{\ell,k}(i)\boldsymbol{\psi}_{\ell k,i} - \boldsymbol{w}_{\ell,i-1}\|^2 \approx \mu_\ell^2\sigma_{v,\ell}^2\mathrm{Tr}(R_{u,\ell}) + M\mathbb{E}|\boldsymbol{g}_{\ell,k}(i)|^2\sigma_{v,\ell k}^{2(\psi)} \tag{6.53}$$

where $\ell \in \boldsymbol{\mathcal{N}}_{k,i}\backslash\{k\}$. For node $\ell = k$, since the communication noise is zero (i.e., $\boldsymbol{v}_{kk}^{(\psi)} = \mathbf{0}_M$), we can write:

$$\mathbb{E}\|\boldsymbol{\psi}_{k,i} - \boldsymbol{w}_{k,i-1}\|^2 \approx \mu_k^2\sigma_{v,k}^2\mathrm{Tr}(R_{u,k}) \tag{6.54}$$

Comparing (6.53) and (6.54) with (6.49), we obtain:

$$\alpha^2_{\ell,k}(i) \approx \mathbb{E}\|\boldsymbol{g}_{\ell,k}(i)\boldsymbol{\psi}_{\ell k,i} - \boldsymbol{w}_{\ell,i-1}\|^2 \tag{6.55}$$

Under the slow adaptation regime, the ATC algorithm over wireless network converges in the mean and mean-square sense [13], which implies that all the estimates $\boldsymbol{w}_{k,i}$ tend close to w^o as $i \to \infty$. This allows us to estimate $\mathbb{E}\|\boldsymbol{g}_{\ell,k}(i)\boldsymbol{\psi}_{\ell k,i} - \boldsymbol{w}_{\ell,i-1}\|^2$ at node k by using its instantaneous realizations of $\boldsymbol{g}_{\ell,k}(i)\boldsymbol{\psi}_{\ell k,i} - \boldsymbol{w}_{\ell,i-1}$. However, since in this relation, $\boldsymbol{w}_{\ell,i-1}$ is not available at node k, we alternatively use the latest estimate of the global parameter at this node, i.e., $\boldsymbol{w}_{k,i-1}$ to replace it. In addition, because some links $\ell \in \mathcal{N}_k$ are found to be disconnected from node k at time instant i, we store $\boldsymbol{\alpha}^2_{\ell,k}(i-1)$ to be used at the following iterations. Note that this storing strategy is useful in obtaining $\boldsymbol{A}_i$ for time-varying fading wireless channels because in tracking $\boldsymbol{\alpha}^2_{\ell,k}(i)$ over time their previous values will be useful. By taking these points into account, relation (6.55) will take the form:

$$\boldsymbol{\alpha}^2_{\ell,k}(i) = \mathbb{E}\|\boldsymbol{g}_{\ell,k}(i)\boldsymbol{\psi}_{\ell k,i} - \boldsymbol{w}_{k,i-1}\|^2 \tag{6.56}$$

and the adaptive combination rule for network over fading wireless channels becomes:

$$\boldsymbol{a}_{\ell,k}(i) = \begin{cases} \frac{\hat{\boldsymbol{\alpha}}^{-2}_{\ell,k}(i)}{\sum_{m\in\boldsymbol{\mathcal{N}}_{k,i}} \hat{\boldsymbol{\alpha}}^{-2}_{m,k}(i)}, & \text{if } \ell \in \boldsymbol{\mathcal{N}}_{k,i} \\ 0, & \text{otherwise} \end{cases} \tag{6.57}$$

where

$$\hat{\boldsymbol{\alpha}}^2_{\ell,k}(i) = \begin{cases} (1-\tau)\hat{\boldsymbol{\alpha}}^2_{\ell,k}(i-1) + \tau\|\boldsymbol{g}_{\ell,k}(i)\boldsymbol{\psi}_{\ell k,i} - \boldsymbol{w}_{k,i-1}\|^2, & \text{if } \ell \in \boldsymbol{\mathcal{N}}_{k,i} \\ \hat{\boldsymbol{\alpha}}^2_{\ell,k}(i-1), & \text{if } \ell \notin \boldsymbol{\mathcal{N}}_{k,i} \end{cases} \tag{6.58}$$

where $0 < \tau < 1$ is the learning factor.

6.4 Simulation Results

In this section, we present computer experiments to illustrate the performance of the ATC DLMS algorithm over wireless channels for different choice of combination matrix $\boldsymbol{A}_i$. We consider a network with $N = 10$ nodes which are randomly spread over a unit square area

$(x, y) \in [0,\ 1] \times [0,\ 1]$, as shown in Fig. 6.1. Here, the unknown parameter vector to be estimated is $w^o = [1 + j,\ -1 + j]^T/2$. We choose equal transmit power $P_t = 1$, nominal transmission range $r_o = 0.4$ and the path-loss exponents $\alpha = 2.5$.

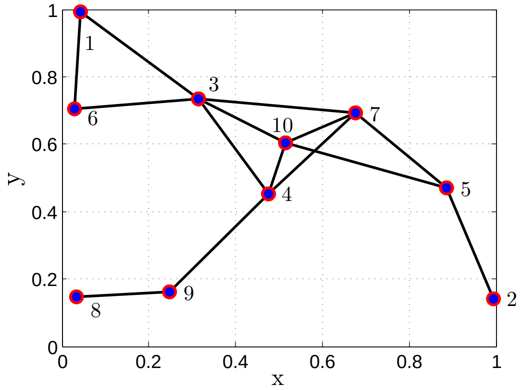


Fig. 6.1 This graph shows the topology of the wireless network at start up time $i = 0$. At this time any two nodes are considered connected if their distance are less than their transmission range, $r_o = 0.4$. In a fading scenario, this topology changes over time, meaning that each node may not communicate with all its neighbors all the time. A node may connect to or disconnect from its neighbors depending on the value of the indicator function (5.25).

In these experiments, the wireless channels between nodes change according to the Clark channel model in which the in-phase and quadrature components, respectively, given by [132]:

$$\boldsymbol{h}^I_{\ell,k}(i) = \frac{1}{N_l}\sum_{n=1}^{N_l} \cos\Big(\boldsymbol{x}_{\ell,k}(n)\, iT_s + \boldsymbol{\phi}_{\ell,k}(n)\Big)$$

$$\boldsymbol{h}^Q_{\ell,k}(i) = \frac{1}{N_l}\sum_{n=1}^{N_l} \sin\Big(\boldsymbol{x}_{\ell,k}(n)\, iT_s + \boldsymbol{\beta}_{\ell,k}(n)\Big)$$

where N_l is the number of multi-paths arriving at the receiving node, T_s is the sampling period, $\boldsymbol{\phi}_{\ell,k}(n)$ and $\boldsymbol{\beta}_{\ell,k}(n)$ are random communication phases between node ℓ and k uniformly distributed over $[0, 2\pi)$, and

$$\boldsymbol{x}_{\ell,k}(n) = 2\pi f^d_{\ell,k} \cos\left\{\frac{(2n-1)\pi + \boldsymbol{\theta}_{\ell,k}}{4N_l}\right\} \tag{6.59}$$

In this expression, $f^d_{\ell,k}$ denotes the doppler frequency shift and $\boldsymbol{\theta}_{\ell,k}$ is a random variable uniformly distributed over $[0, 2\pi)$. For these experiments, we set $N_l = 16$, $T_s = 0.0001$, and $f^d_{\ell,k} = 100Hz$, which corresponds to fast time-varying channels.

For each node $k \in \{1, 2, \cdots, N\}$, we set the step-size to $\mu_k = 0.01$ and the initial weight vectors to $\boldsymbol{w}_{k,-1} = \mathbf{0}_M$. We adopt zero-mean complex circular Gaussian random distribution to generate $\boldsymbol{v}_k(i)$, $\boldsymbol{v}^{(\psi)}_{\ell k,i}$ and $\boldsymbol{u}_{k,i}$. The regression data $\boldsymbol{u}_{k,i}$ have covariance matrices of the form $R_{u,k} = \sigma^2_{u,k} I_M$. The exchanged data between nodes experience distortion characterized by (5.2). The trace of the regression data, $\text{Tr}(R_{u,k})$, and the variance of measurement noise, $\sigma^2_{v,k}$ are illustrated in Fig. 6.2. The distribution of the communication noise power, $\sigma^{2(\psi)}_{v,\ell k}$, over the spatial domain is illustrated in Fig. 6.3.

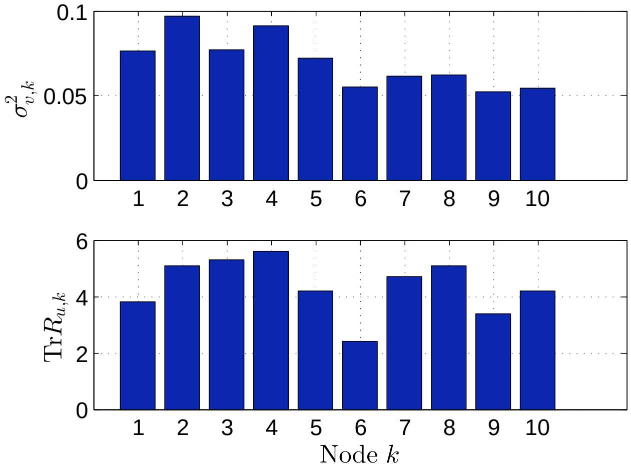


Fig. 6.2 Network energy profile

The combination rules used to compute $\boldsymbol{A}_i$ are: the modified Metropolis rule (6.30), the modified Laplacian scheme (6.31), the modified relative variance, (6.33), the proposed optimal method (6.49), and the proposed adaptive combination rule (6.57).

From Fig. 6.4, we observe that the DLMS Algorithm converges with almost the same rate for all combination rules. As seen from Fig. 6.4 and 6.5, in the steady-state, the performance of the Algorithm with $\boldsymbol{A}_i$ computed using (6.49) and proposed adaptive combination rule (6.57) are superior to DLMS with combination matrices (6.30), (6.31) and (6.33). As the results indicate, in the steady state, the MSD of DLMS with the proposed adaptive combination rule (6.57) is identical to the proposed optimal method (6.49). We note that in time-varying wireless channels, the DLMS algorithm with combination rules

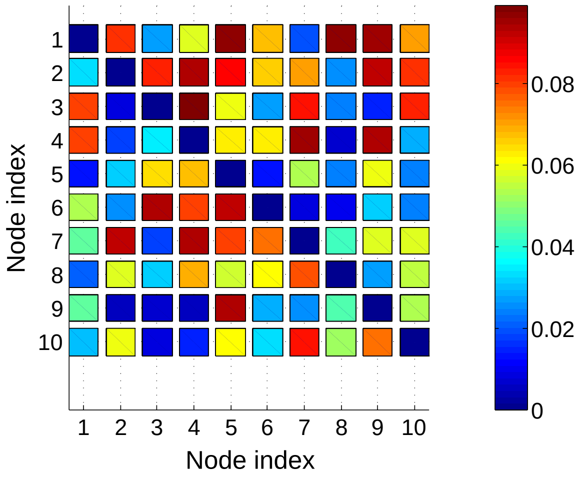


Fig. 6.3 Communication noise power over the network

(6.30), (6.31), (6.33) and (6.49) requires to compute $\boldsymbol{A}_i$ at each time i. In contrast, with the adaptive combination rule (6.57), the algorithm only needs to updates its combination weights along with the main estimation task at each iteration.

6.5 Summary

We showed that the performance of DLMS algorithms over wireless sensor networks mainly depends on the network left-stochastic combination matrices. To improve the performance of the DLMS algorithms over wireless channels, we formulated an optimization problem and solved it to find the optimal combination weights that lead to a lower steady-state MSD over the network. We further developed an adaptive scheme to obtain the optimal combination weights in real-time. The developed adaptive scheme does not require nodes signal statistics (e.g, input signal correlation matrix and measurement and communication noise variances) and it is therefore useful in wireless networks with changing topology and time-varying channel conditions. Moreover, this strategy significantly reduces the computational processing load of each node over the network.

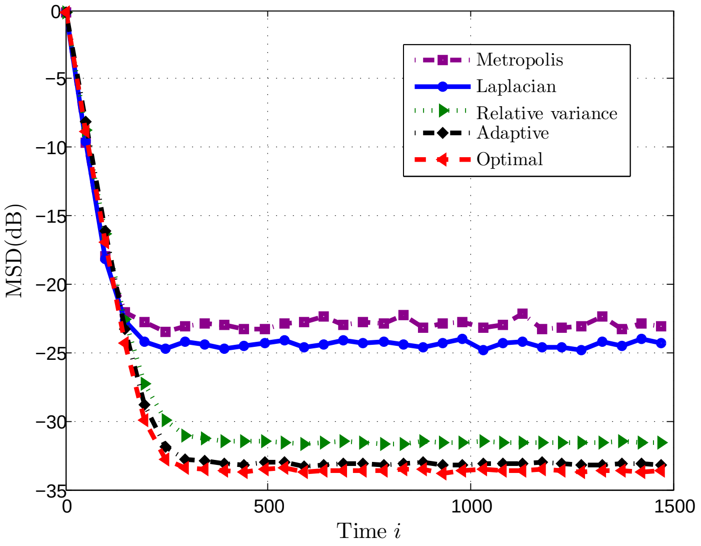


Fig. 6.4 The network MSD learning behavior for different combination matrices.

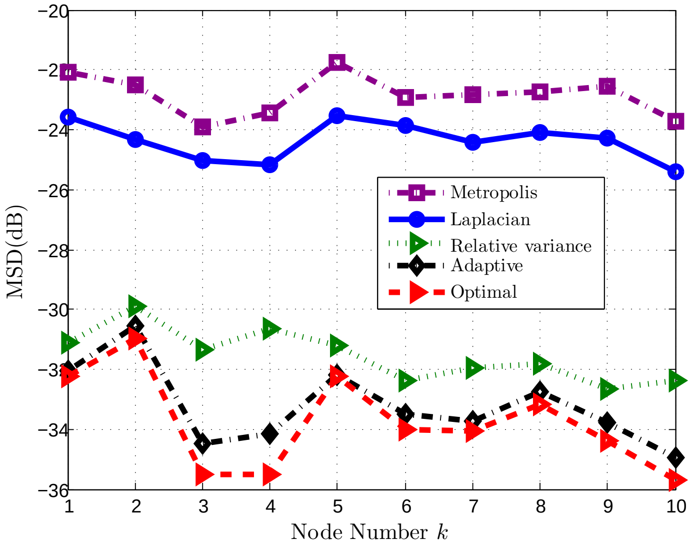


Fig. 6.5 The network steady-states MSD for different nodes and combination matrices.

Chapter 7

Conclusion and Future Works

In this thesis, we have studied the performance of distributed adaptive strategies, mainly DLMS algorithms, and examined their performance over sensor networks under several practical constraints. We consider limiting aspects and practical conditions, including, variations of physical parameters over space, distortion of the regression data with measurement noise as well as communication constraints such as fading, path loss and link noise. Since, these constraint directly impact the functionality of the DLMS algorithms, we developed new form of DLMS strategies that can efficiently operate in such conditions and achieve satisfactory performance results. For all the newly introduced DLMS algorithms, we conducted a detailed performance analysis and validated our theoretical findings through numerical experiments under controlled and realistic network conditions. Below, we summarize the main contributions of the thesis while commenting on the efficiency and performance limits of the newly proposed algorithms. We then present several open problems and research directions in the field that can be pursued in the future.

7.1 Summary and Conclusions

In Chapter 2, we studied the differences between diffusion and consensus strategies in terms of operation, convergence behavior and stability. The main conclusion obtained from this chapter was that the DLMS strategies outperform the consensus ones in multi-agent networks, where the learning and tracking are the main objectives. Motivated by this result, in the reminder of the thesis, we focused on the DLMS strategies and investigated their performance in parameter estimation under practical conditions of network operation and limiting constraints.

In Chapter 3, by combining interpolation and distributed adaptive optimization, we proposed a DLMS strategy for estimation and tracking of space-time varying parameters over sensor networks. The proposed algorithm can estimate the space-varying parameters not only at the sensor nodes but also at locations where no measurement is collected. We showed that if the network experiences data with rank-deficient covariance matrices, the non-cooperative DLMS algorithm will converge to different solutions at different nodes whereas the proposed DLMS algorithm is able to alleviate the rank-deficiency problem in most cases through its use of combination matrices. Nevertheless, if the algorithm fails to mitigate the deleterious effect of the rank-deficient data, then the estimates converges to a solution space where the resulting estimation errors are smaller than that of the non-cooperative DLMS. We analyzed the performance of the proposed algorithm in transient and steady-state regimes, and provided the mean and mean-square error stability conditions.

In Chapter 4, we developed new DLMS strategies for parameter estimation in sensor networks where the regression data are corrupted with additive noise. Under this condition, we first showed that if a DLMS algorithm is implemented to estimate the underlying system parameters without considering the effect of the measurement noise, the estimate will be biased and unreliable. We then resolved this issue by introducing a bias-elimination technique and formulating an optimization problem that utilizes the noise variance information of the regression data. By solving this optimization problem, we arrived at novel DLMS algorithms, called bias-compensated DLMS strategies, that are capable of obtaining unbiased estimates of the unknown parameters over the network. We also derived a recursive adaptive approach by which each node, besides the standard adaptation layer to solve the desired distributed estimation, runs a second layer estimation to locally find their regression noise variances over time. Our analysis showed that in slow adaptation regime, the developed algorithms are stable in the mean and mean-square error sense and the estimates asymptotically converge to their true values. The proposed bias-compensated DLMS algorithms, which operate in a distributed manner and exchange data via single-hop communication, can also help network to save energy and radio resources.

In Chapter 5, we examined the performance of DLMS algorithms over wireless sensor networks, where the communication links between nodes are adversely affected by fading, path-loss and noise. Wireless channel impairments, including path loss and fading, can distort the exchanged data between nodes subsequently, degrade the performance of DLMS algorithms and cause instability. To resolve this issue, we proposed an extended version of these algorithms that incorporate equalization coefficients in their combination update to reverse the effects of fading and path loss. We also introduced a dynamic combination rule to obtain the network weighting matrix, as the network topology varies following the changes in the instantaneous signal-to-noise ratio (SNR) of the links. In cases where channels coefficients are found using least-squares estimation, we analyzed the impact of channel estimation error on the performance of the proposed algorithms and obtained conditions that guarantee the stability of the network in the mean and mean-square error sense. Our analytical finding indicated that the modified DLMS algorithms remain stable and converge in the mean and mean-square sense by proper CSI monitoring over the network, provided that the network optimization step-sizes are sufficiently small.

Our analysis also revealed that the performance of the DLMS algorithms significantly depend on the network CSI and the links SNR, while the performance of the DLMS algorithms generally improve as the knowledge about the network CSI increases. We observed that, the performance discrepancies in DLMS algorithms with perfect and estimated CSI is larger at low SNR. This difference in performance can be reduced by using more pilot data in the estimation of the channel coefficients in each time slot. In addition, at very low SNR, we observed that the non-cooperative DLMS outperform the diffusion strategies. This result suggested that in wireless networks where the radio links experience large noise power, or equivalently when the nodes transmit power are very low, the network must switch to the non-cooperative mode to maintain a satisfactory performance level. This also implies that if the transmit power of some nodes is below a threshold value, they should go to sleep mode to avoid error propagation over the network.

Finally, in Chapter 6, we showed that the performance of DLMS algorithms over multi-agent wireless networks mainly depends on the network left-stochastic combination matrices. To improve the performance of the DLMS algorithms over wireless channels, we formulated a convex optimization problem from an upper-bound approximation of the network MSD in order to find the optimal combination weights, which lead to smaller estimation errors. We further developed an adaptive scheme to obtain the optimal combination weights in real-time. The latter does not require the second order statistics of the nodes' data (e.g; the correlation matrices of the input signals and the variances of the measurement and links noise) and it is, therefore, useful in sensor networks with time-varying wireless channels and changing topology.

7.2 Future Works

There are several potential research topics that can be pursued based on the results obtained in this thesis; they are briefly summarized below:

(1) In Chapter 3, we considered distributed estimation of the parameters of physical phenomena whose discretized PDE can be expressed as a linear regression model. There are, however, other classes of physical phenomena for which the discretized PDE's lead to non-linear regression models. Therefore, the generalization of DLMS strategies to estimate parameters of non-linear regression models is important specially where the corresponding objective function over the network are non-convex. Note that reference [85] has investigated distributed optimization problems over networks where the local cost functions are convex.

(2) In Chapter 4, we assumed that the input regression data at each node over the network are corrupted with zero-mean white Gaussian noise. In our development, to relax the known variance assumption of the regression noise, we exploited the whiteness assumption of the regression noise to estimate their variances over time. Generalization of the proposed adaptive scheme with colored regression noise and unknown covariance matrices is an open problem that needs to be investigated. The analysis of the bias-compensated DLMS algorithms under such conditions can also be considered a significant contribution.

(3) In Chapter 5, we extended the application of DLMS strategies for parameter estimation and optimization wireless sensor networks. We assumed that all agents measure a phenomenon arising from a linear regression model that features an identical parameter vector over spatial domain, or equivalently all agents share a similar minimizer w^o. We showed that under such conditions, the DLMS strategies are useful when the variances of the communication noise over the network is below some threshold level. The important result was that at low SNR, it will be advantageous for agents to work independently on their own to find the global parameter vector. However, there are some applications that nodes may not share the global minimizer and therefore,

require cooperation in order to achieve the network objective. One of this application is Pareto optimization where each node k has its own specific local minimizer w_k^o. Consequently, if each node operate independently, the network global minimizer may not be found. Therefore, the problem can be defined as the performance investigation of DLMS algorithms in solving Pareto optimization problems over WSN at low SNR.

(4) Time-variant estimation problems arise in many applications in sensor networks where the phenomena under the study change over time. One of the effective solution strategies for this class of problems is the DLMS algorithms. The future research direction, in this scenario, can be defined as the search for the optimal adaptation step-sizes of the DLMS algorithms over the network in a distributed manner, concurrently, with the estimation of global parameters over the network. Ideally, these optimal step-sizes will lead to improvement in the tracking speed and estimation accuracy of the DLMS algorithms. In previous studies on DLMS algorithms, the step-sizes are chosen according to some given stability range, where there was a trade off between convergence speed and the steady-state MSE values over the network, i.e.: as the step-sizes get smaller the steady-state MSE values decrease. This is not the case in non-stationary signal environment, because if the step-sizes are chosen too small, the algorithm may not be able to track the changes. In contrast, large step-sizes may lead to unstable behavior. Under such circumstances, there are optimal adaptation step-sizes that lead to low steady-state MSE and agile tracking ability.

Appendix A

Proofs and Derivations

A.1 Mean Error Convergence of Diffusion LMS for Space-Varying Parameters

Based on the rank of $\mathcal{R} = \text{diag}\{R_1, \cdots, R_N\}$, we have two possible cases:

a) $R_k > 0 \; \forall k \in \{1, \cdots, N\}$: As (3.67) implies, $\mathbb{E}[\tilde{\boldsymbol{w}}_i]$ converges to zero if $\rho(\mathcal{B}) < 1$. In [82], it was shown that when $\mathcal{R} > 0$, choosing the step-sizes according to (3.72) guarantees $\rho(\mathcal{B}) < 1$.

b) $\exists k \in \{1, \cdots, N\}$ for which R_k is rank-deficient: For this case, we first show that

$$\left\| \mathcal{B}^{i+1} \right\|_{b,\infty} \leq \left\| \left(I - \mathcal{M}\Lambda \right)^{i+1} \right\|_{b,\infty} \tag{A.1}$$

where $\| \cdot \|_{b,\infty}$ denotes the block-maximum norm for block vectors with block entries of size $MN_b \times 1$ and block matrices with blocks of size $MN_b \times MN_b$. To this end, we note that for the left-stochastic matrices A_1 and A_2, we have $\|\mathcal{A}_1^T\|_{b,\infty} = \|\mathcal{A}_2^T\|_{b,\infty} = 1$ [82], and use the sub-multiplicative property of the block maximum norm [53] to write:

$$\begin{aligned} \left\| \mathcal{B}^{i+1} \right\|_{b,\infty} &\leq \|\mathcal{A}_2^T\|_{b,\infty} \, \|I - \mathcal{M}\mathcal{R}\|_{b,\infty} \, \|\mathcal{A}_1^T\|_{b,\infty} \cdots \|\mathcal{A}_2^T\|_{b,\infty} \, \|I - \mathcal{M}\mathcal{R}\|_{b,\infty} \, \|\mathcal{A}_1^T\|_{b,\infty} \\ &= \left\| I - \mathcal{M}\mathcal{R} \right\|_{b,\infty}^{i+1} \end{aligned} \tag{A.2}$$

If we introduce the (block) eigendecomposition of $\mathcal{R}$ (3.70) into (A.2) and consider the fact that the block-maximum norm is invariant under block-diagonal unitary matrix transformations [63, 82], then inequality (A.2) takes the form:

$$\left\|\mathcal{B}^{i+1}\right\|_{b,\infty} \leq \left\|I - \mathcal{M}\Lambda\right\|_{b,\infty}^{i+1} \tag{A.3}$$

Using the property $\|X\|_{b,\infty} = \rho(X)$ for a block diagonal Hermitian matrix X [82], we obtain:

$$\begin{aligned}\left\|(I - \mathcal{M}\Lambda)^{i+1}\right\|_{b,\infty} =& \rho\Big((I - \mathcal{M}\Lambda)^{i+1}\Big)\\ =& \max_{\substack{1\leq k\leq N\\ 1\leq n\leq MN_b}} \left|\Big(1 - \mu_k\lambda_k(n)\Big)^{i+1}\right|\\ =& \Bigg(\max_{\substack{1\leq k\leq N\\ 1\leq n\leq MN_b}} |1 - \mu_k\lambda_k(n)|\Bigg)^{i+1}\\ =& \Big(\rho(I - \mathcal{M}\Lambda)\Big)^{i+1}\\ =& \left\|I - \mathcal{M}\Lambda\right\|_{b,\infty}^{i+1}\end{aligned} \tag{A.4}$$

Using (A.4) in (A.3), we arrive at (A.1). We now proceed to show the boundedness of the mean error for case (b). We iterate (3.67) to get:

$$\mathbb{E}[\tilde{\boldsymbol{w}}_i] = \mathcal{B}^{i+1}\mathbb{E}[\tilde{\boldsymbol{w}}_{-1}] \tag{A.5}$$

Applying the block maximum norm to (A.5) and using inequality (A.1), we obtain:

$$\lim_{i\to\infty}\left\|\mathbb{E}[\tilde{\boldsymbol{w}}_i]\right\|_{b,\infty} \leq \lim_{i\to\infty}\left\|(I - \mathcal{M}\Lambda)^{i+1}\right\|_{b,\infty}\left\|\mathbb{E}[\tilde{\boldsymbol{w}}_{-1}]\right\|_{b,\infty} \tag{A.6}$$

The value of $\lim_{i\to\infty}\|(I - \mathcal{M}\Lambda)^{i+1}\|_{b,\infty}$ can be computed by evaluating the limits of its diagonal entries. Considering the step-sizes as in (3.72), the diagonal entries are computed as:

$$\lim_{i\to\infty}\Big(1 - \mu_k\lambda_k(n)\Big)^{i+1} = \begin{cases} 1, & \text{if } \lambda_k(n) = 0\\ 0, & \text{otherwise}\end{cases} \tag{A.7}$$

Therefore, (A.6) reads as:

$$\lim_{i\to\infty} \left\| \mathbb{E}[\tilde{\boldsymbol{w}}_i] \right\|_{b,\infty} \leq \|I - \text{Ind}(\Lambda)\|_{b,\infty} \left\| \mathbb{E}[\tilde{\boldsymbol{w}}_{-1}] \right\|_{b,\infty} \tag{A.8}$$

A.2 Mean Behavior When ($A_1 = A_2 = I$)

Setting $A_1 = A_2 = I$ in the diffusion recursions (3.39)-(3.41) and subtracting w^o from both sides of (3.40), we get:

$$\tilde{\boldsymbol{w}}_{k,i} = \tilde{\boldsymbol{w}}_{k,i-1} - \mu_k \sum_{\ell\in\mathcal{N}_k} c_{\ell,k} B_\ell^T \boldsymbol{u}_{\ell,i}^T (\boldsymbol{d}_\ell(i) - \boldsymbol{u}_{\ell,i} B_\ell \boldsymbol{w}_{k,i-1}) \tag{A.9}$$

Under Assumption 3.1 and using $\boldsymbol{d}_\ell(i) = \boldsymbol{u}_{\ell,i} B_\ell w^o + \boldsymbol{v}_\ell(i)$, we obtain:

$$\mathbb{E}[\tilde{\boldsymbol{w}}_{k,i}] = Q_k[I - \mu_k \Lambda_k] Q_k^T \, \mathbb{E}[\tilde{\boldsymbol{w}}_{k,i-1}] \tag{A.10}$$

We define $\boldsymbol{p}_{k,i} \triangleq Q_k^T \tilde{\boldsymbol{w}}_{k,i}$ and start from some initial condition to arrive at

$$\mathbb{E}[\boldsymbol{p}_{k,i}] = [I - \mu_k \Lambda_k] \mathbb{E}[\boldsymbol{p}_{k,i-1}] = [I - \mu_k \Lambda_k]^{i+1} \mathbb{E}[\boldsymbol{p}_{k,-1}]$$

If we choose the step-sizes according to (3.72) then we get:

$$\lim_{i\to\infty} \mathbb{E}[\boldsymbol{p}_{k,i}] = \Big[I - \text{Ind}(\Lambda_k) \Big] \mathbb{E}[\boldsymbol{p}_{k,-1}] \tag{A.11}$$

Equivalently, this can be written as:

$$\lim_{i\to\infty} \mathbb{E}[\tilde{\boldsymbol{w}}_{k,i}] = Q_k \Big[I - \text{Ind}(\Lambda_k) \Big] Q_k^T \, \mathbb{E}[\tilde{\boldsymbol{w}}_{k,-1}] \tag{A.12}$$

This result indicates that the mean error does not grow unbounded. Now from (3.74), we can verify that:

$$Q_k \text{Ind}(\Lambda_k) Q_k^T w^o = R_k^\dagger r_k \tag{A.13}$$

Then, upon substitution of $\tilde{\boldsymbol{w}}_{k,i} = w^o - \boldsymbol{w}_{k,i}$ into (A.12), we obtain:

$$\begin{aligned}\lim_{i\to\infty} \mathbb{E}[\boldsymbol{w}_{k,i}] &= Q_k \mathrm{Ind}(\Lambda_k) Q_k^T w^o + Q_k[I - \mathrm{Ind}(\Lambda_k)]Q_k^T \mathbb{E}[\boldsymbol{w}_{k,-1}] \\ &= R_k^\dagger r_k + \sum_{n=L_k+1}^{MN_b} q_{k,n} q_{k,n}^T \mathbb{E}[\boldsymbol{w}_{k,-1}] \end{aligned} \tag{A.14}$$

A.3 Proof of Theorem 3.2

From (3.86), we readily deduce that

$$\lim_{i\to\infty} \mathcal{B}^{i+1}\mathbb{E}[\boldsymbol{w}_{-1}] = (\mathcal{Z}_2\bar{\mathcal{Z}}_2)\,\mathbb{E}[\boldsymbol{w}_{-1}] \tag{A.15}$$

On the other hand, from (3.85), we have

$$\lim_{i\to\infty} \sum_{j=0}^{i} \mathcal{B}^j \mathcal{A}_2^T \mathcal{M} r = \lim_{i\to\infty} \sum_{j=0}^{i} \left(\mathcal{Z}_1 J^j \bar{\mathcal{Z}}_1 + \mathcal{Z}_2\bar{\mathcal{Z}}_2\right)\mathcal{A}_2^T \mathcal{M} r \tag{A.16}$$

Using (3.87), the term involving $\bar{\mathcal{Z}}_2$ cancels out and the above reduces to

$$\begin{aligned}\lim_{i\to\infty} \sum_{j=0}^{i} \mathcal{B}^j \mathcal{A}_2^T \mathcal{M} r &= \lim_{i\to\infty} \sum_{j=0}^{i} \left(\mathcal{Z}_1 J^j \bar{\mathcal{Z}}_1\right)\mathcal{A}_2^T \mathcal{M} r \\ &= \mathcal{Z}_1 (I - J)^{-1} \bar{\mathcal{Z}}_1 \mathcal{A}_2^T \mathcal{M} r \end{aligned} \tag{A.17}$$

since $\rho(J) < 1$. We now verify that the matrix

$$X^- = \mathcal{Z}_1 (I - J)^{-1} \bar{\mathcal{Z}}_1 \tag{A.18}$$

is a (reflexive) generalized inverse for the matrix $X = (I - \mathcal{B})$. Recall that a (reflexive) generalized inverse for a matrix Y is any matrix Y^- that satisfies the two conditions [133]:

$$YY^-Y = Y \tag{A.19}$$

$$Y^-YY^- = Y^- \tag{A.20}$$

To verify these conditions, we first note from $\mathcal{Z}\mathcal{Z}^{-1} = I$ and $\mathcal{Z}^{-1}\mathcal{Z} = I$ in (3.85) that the following relations hold:

$$\mathcal{Z}_1\bar{\mathcal{Z}}_1 + \mathcal{Z}_2\bar{\mathcal{Z}}_2 = I \tag{A.21}$$

$$\bar{\mathcal{Z}}_1\mathcal{Z}_2 = 0 \tag{A.22}$$

$$\bar{\mathcal{Z}}_2\mathcal{Z}_1 = 0 \tag{A.23}$$

$$\bar{\mathcal{Z}}_1\mathcal{Z}_1 = I \tag{A.24}$$

$$\bar{\mathcal{Z}}_2\mathcal{Z}_2 = I \tag{A.25}$$

We further note that X can be expressed as:

$$X = (I - \mathcal{B}) = \mathcal{Z}_1(I - J)\bar{\mathcal{Z}}_1 \tag{A.26}$$

It is then easy to verify that the matrices $\{X, X^-\}$ satisfy conditions (A.19) and (A.20), as claimed. Therefore, (A.17) can be expressed as:

$$\lim_{i\to\infty} \sum_{j=0}^{i} \mathcal{B}^j \mathcal{A}_2^T \mathcal{M} r = (I - \mathcal{B})^- \mathcal{A}_2^T \mathcal{M} r \tag{A.27}$$

Substituting (A.15) and (A.27) into (3.77) leads to (3.88).

Let us now verify that the right-hand side of (3.88) remains invariant under basis transformations for the Jordan factors $\{\mathcal{Z}_1, \bar{\mathcal{Z}}_1, \mathcal{Z}_2, \bar{\mathcal{Z}}_2\}$. To begin with, the Jordan decomposition (3.85) is not unique. Let us assume, however, that we fix the central term $\text{diag}\{J, I\}$ to remain invariant and allow the Jordan factors $\{\mathcal{Z}_1, \bar{\mathcal{Z}}_1, \mathcal{Z}_2, \bar{\mathcal{Z}}_2\}$ to vary. It follows from (3.85) that

$$\bar{\mathcal{Z}}_2\mathcal{B} = \bar{\mathcal{Z}}_2, \quad \mathcal{B}\mathcal{Z}_2 = \mathcal{Z}_2 \tag{A.28}$$

so that the columns of $\mathcal{Z}_2$ and the rows of $\bar{\mathcal{Z}}_2$ correspond to right and left-eigenvectors of $\mathcal{B}$, respectively, associated with the eigenvalues with value one. If we replace $\mathcal{Z}_2$ by any transformation of the form $\mathcal{Z}_2\mathcal{X}_2$, where $\mathcal{X}_2$ is invertible, then by (A.25), $\bar{\mathcal{Z}}_2$ should be replaced by $\mathcal{X}_2^{-1}\bar{\mathcal{Z}}_2$. This conclusion can also be seen as follows. The new factor $\mathcal{Z}$ is given by

$$\mathcal{Z} \triangleq \begin{bmatrix} \mathcal{Z}_1 & \mathcal{Z}_2\mathcal{X}_2 \end{bmatrix} = \begin{bmatrix} \mathcal{Z}_1 & \mathcal{Z}_2 \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & \mathcal{X}_2 \end{bmatrix} \tag{A.29}$$

and, hence, the new $\mathcal{Z}^{-1}$ becomes

$$\mathcal{Z}^{-1} = \begin{bmatrix} \bar{\mathcal{Z}}_1 \\ \mathcal{X}_2^{-1}\bar{\mathcal{Z}}_2 \end{bmatrix} \tag{A.30}$$

which confirms that $\bar{\mathcal{Z}}_2$ is replaced by $\mathcal{X}_2^{-1}\bar{\mathcal{Z}}_2$. It follows that the product $\mathcal{Z}_2\bar{\mathcal{Z}}_2$ remains invariant under arbitrary invertible transformations $\mathcal{X}_2$.

Moreover, from (3.85) we also have that

$$\bar{\mathcal{Z}}_1\mathcal{B} = J\bar{\mathcal{Z}}_1, \quad \mathcal{B}\mathcal{Z}_1 = \mathcal{Z}_1 J \tag{A.31}$$

Assume we replace $\mathcal{Z}_1$ by any transformation of the form $\mathcal{Z}_1\mathcal{X}_1$, where $\mathcal{X}_1$ is invertible, then by (A.24), $\bar{\mathcal{Z}}_1$ should be replaced by $\mathcal{X}_1^{-1}\bar{\mathcal{Z}}_1$. However, since we want to maintain J invariant, then this implies that the transformation $\mathcal{X}_1$ must also satisfy

$$\mathcal{X}_1^{-1}J\mathcal{X}_1 = J \tag{A.32}$$

It follows that the product $\mathcal{Z}_1(I-J)^{-1}\bar{\mathcal{Z}}_1$ remains invariant under such invertible transformations $\mathcal{X}_1$, since

$$\begin{aligned} \mathcal{Z}_1(I-J)^{-1}\bar{\mathcal{Z}}_1 &= \mathcal{Z}_1\mathcal{X}_1\mathcal{X}_1^{-1}(I-J)^{-1}\mathcal{X}_1\mathcal{X}_1^{-1}\bar{\mathcal{Z}}_1 \\ &= \mathcal{Z}_1\mathcal{X}_1(I-\mathcal{X}_1^{-1}J\mathcal{X}_1)^{-1}\mathcal{X}_1^{-1}\bar{\mathcal{Z}}_1 \\ &= \mathcal{Z}_1\mathcal{X}_1(I-J)^{-1}\mathcal{X}_1^{-1}\bar{\mathcal{Z}}_1 \end{aligned} \tag{A.33}$$

A.4 Proof of Theorem 3.3

We first establish that $\bar{\mathcal{Z}}_2\mathcal{Y}$ and $\mathcal{Y}\bar{\mathcal{Z}}_2^T$ are both equal to zero. Indeed, we start by replacing r in (3.87) by its expression from (3.60) and (3.62) as $r = \mathcal{C}^T\text{col}\{\bar{r}_{du,1}, \cdots, \bar{r}_{du,N}\}$:

$$\bar{\mathcal{Z}}_2\mathcal{A}_2^T\mathcal{M}\mathcal{C}^T\text{col}\{\bar{r}_{du,1}, \cdots, \bar{r}_{du,N}\} = 0 \tag{A.34}$$

By further replacing $\bar{r}_{du,k}$ by their values from (3.32), we obtain:

$$\bar{\mathcal{Z}}_2\mathcal{A}_2^T\mathcal{M}\mathcal{C}^T\text{diag}\{B_1^T, \cdots, B_N^T\}\text{col}\{r_{du,1}, \cdots, r_{du,N}\} = 0 \tag{A.35}$$

This relation must hold regardless of the cross-correlation vectors $\{r_{du,k}\}$. Therefore,

$$\bar{\mathcal{Z}}_2\mathcal{A}_2^T\mathcal{M}\mathcal{C}^T\mathrm{diag}\{B_1^T,\cdots,B_N^T\}=0 \tag{A.36}$$

We now define

$$\mathcal{V}=\mathrm{diag}\{\sigma_{v,1}^2I_{MN_b},\cdots,\sigma_{v,N}^2I_{MN_b}\} \tag{A.37}$$

and rewrite expression (3.99) as

$$\begin{aligned}\mathcal{Y}=\mathcal{A}_2^T\mathcal{M}\mathcal{C}^T&\mathrm{diag}\{B_1^T,\cdots,B_N^T\}\mathrm{diag}\{R_{u,1},\cdots,R_{u,N}\}\\&\times\mathrm{diag}\{B_1,\cdots,B_N\}\,\mathcal{V}\mathcal{C}\mathcal{M}\mathcal{A}_2\end{aligned} \tag{A.38}$$

Multiplying this from the left by $\bar{\mathcal{Z}}_2$ and comparing the result with (A.36), we conclude that

$$\bar{\mathcal{Z}}_2\mathcal{Y}=0 \tag{A.39}$$

Transposing this relation and noting that $\mathcal{Y}$ is symmetric, we obtain:

$$\mathcal{Y}\bar{\mathcal{Z}}_2^T=0 \tag{A.40}$$

Returning to recursion (3.101), we note first from (3.85) that $\mathcal{B}$ can be rewritten as

$$\mathcal{B}=\mathcal{Z}_1J\bar{\mathcal{Z}}_1+\mathcal{Z}_2\bar{\mathcal{Z}}_2 \tag{A.41}$$

Since $\mathcal{B}$ is power convergent, the first term on the right hand side of (3.101) converges to

$$\lim_{i\to\infty}\mathbb{E}\|\tilde{\boldsymbol{w}}_{-1}\|^2_{(\mathcal{B}^T)^{i+1}\Sigma\mathcal{B}^{i+1}}=\mathbb{E}\|\tilde{\boldsymbol{w}}_{-1}\|^2_{(\mathcal{Z}_2\bar{\mathcal{Z}}_2)^T\Sigma\mathcal{Z}_2\bar{\mathcal{Z}}_2} \tag{A.42}$$

Substituting (A.41) into the second term on the right hand side of (3.101) and using (A.39) and (A.40), we arrive at

$$\lim_{i\to\infty}\sum_{j=0}^{i}\mathrm{Tr}\Big((\mathcal{B}^T)^j\Sigma\mathcal{B}^j\mathcal{Y}\Big)=\mathrm{Tr}\Big(\lim_{i\to\infty}\sum_{j=0}^{i}(\mathcal{Z}_1J^j\bar{\mathcal{Z}}_1)^T\Sigma(\mathcal{Z}_1J^j\bar{\mathcal{Z}}_1)\mathcal{Y}\Big) \tag{A.43}$$

If matrices X_1, X_2 and Σ are of compatible dimensions, then the following relations hold [82]:

$$\mathrm{Tr}(X_1X_2) = \Big(\mathrm{vec}(X_2^T)\Big)^T \mathrm{vec}(X_1) \tag{A.44}$$

$$\mathrm{vec}(X_1\Sigma X_2) = (X_2^T \otimes X_1)\mathrm{vec}(\Sigma) \tag{A.45}$$

Using these relations in (A.43), we obtain

$$\mathrm{Tr}\Bigg(\lim_{i\to\infty}\sum_{j=0}^{i}(\mathcal{B}^T)^j\Sigma\mathcal{B}^j\mathcal{Y}\Bigg) = \Bigg(\mathrm{vec}(\mathcal{Y}^T)\Bigg)^T\Bigg(\lim_{i\to\infty}\sum_{j=0}^{i}(\mathcal{Z}_1J^j\bar{\mathcal{Z}}_1)^T\otimes(\mathcal{Z}_1J^j\bar{\mathcal{Z}}_1)^T\Bigg)\mathrm{vec}(\Sigma) \tag{A.46}$$

This is equivalent to:

$$\mathrm{Tr}\Bigg(\lim_{i\to\infty}\sum_{j=0}^{i}(\mathcal{B}^T)^j\Sigma\mathcal{B}^j\mathcal{Y}\Bigg) = \Bigg(\mathrm{vec}(\mathcal{Y})\Bigg)^T\Bigg(\lim_{i\to\infty}\sum_{j=0}^{i}\mathcal{F}^j\Bigg)\mathrm{vec}(\Sigma) \tag{A.47}$$

where

$$\mathcal{F} = \Bigg((\mathcal{Z}_1\otimes\mathcal{Z}_1)(J\otimes J)(\bar{\mathcal{Z}}_1\otimes\bar{\mathcal{Z}}_1)\Bigg)^T \tag{A.48}$$

Since $\rho(J\otimes J) < 1$, the series converges and we obtain:

$$\mathrm{Tr}\Bigg(\lim_{i\to\infty}\sum_{j=0}^{i}(\mathcal{B}^T)^j\Sigma\mathcal{B}^j\mathcal{Y}\Bigg) = \Bigg(\mathrm{vec}(\mathcal{Y})\Bigg)^T(I-\mathcal{F})^{-1}\mathrm{vec}(\Sigma) \tag{A.49}$$

Upon substitution of (A.42) and (A.49) into (3.101), we arrive at (3.103).

A.5 Computation of Π

We rewrite Π as:

$$\Pi = \mathbb{E}[\boldsymbol{\mathcal{P}}_i\Omega\boldsymbol{\mathcal{P}}_i^*] \tag{A.50}$$

where $\Omega = \omega^o\omega^{o*}$. The $[k,j]^{th}$ block of Π can be computed as:

$$\Pi_{k,j} = \mathbb{E}\sum_{\ell}\sum_{m} c_{\ell,k}c_{m,j}(\boldsymbol{z}^*_{\ell,i}\boldsymbol{n}_{\ell,i} - \sigma^2_{n,\ell}I)\Omega_{kj}(\boldsymbol{n}^*_{m,i}\boldsymbol{z}_{m,i} - \sigma^2_{n,m}I) \tag{A.51}$$

We use (4.1) to replace $\boldsymbol{z}_{\ell,i}$ and $\boldsymbol{z}_{m,i}$:

$$\begin{aligned}\Pi_{k,j} = \mathbb{E}\sum_{\ell}\sum_{m} & c_{\ell,k}c_{m,j}(\boldsymbol{u}^*_{\ell,i}\boldsymbol{n}_{\ell,i} + \boldsymbol{n}^*_{\ell,i}\boldsymbol{n}_{\ell,i} - \sigma^2_{n,\ell}I)\Omega_{kj} \\ & \times(\boldsymbol{n}^*_{m,i}\boldsymbol{u}_{m,i} + \boldsymbol{n}^*_{m,i}\boldsymbol{n}_{m,i} - \sigma^2_{n,m}I)\end{aligned} \tag{A.52}$$

This leads to:

$$\Pi_{k,j} = \sum_{\ell}\sum_{m} c_{\ell,k}c_{m,j}\mathbb{E}[\boldsymbol{u}^*_{\ell,i}\boldsymbol{n}_{\ell,i}\Omega_{kj}\boldsymbol{n}^*_{m,i}\boldsymbol{u}_{m,i}] \tag{A.53}$$

$$+\sum_{\ell}\sum_{m} c_{\ell,k}c_{m,j}\mathbb{E}[\boldsymbol{n}^*_{\ell,i}\boldsymbol{n}_{\ell,i}\Omega_{kj}\boldsymbol{n}^*_{m,i}\boldsymbol{n}^*_{m,i}]$$

$$-\sum_{\ell}\sum_{m} c_{\ell,k}c_{m,j}\mathbb{E}[\sigma^2_{n,\ell}I\Omega_{kj}\boldsymbol{n}^*_{m,i}\boldsymbol{n}_{m,i}] \tag{A.54}$$

If we assume the regression $\{\boldsymbol{u}_{k,i}\}$ and the noise $\{\boldsymbol{n}_{k,i}\}$ are zero mean circular Gaussian complex-valued vectors with uncorrelated entries, then:

$$\mathbb{E}[\boldsymbol{u}^*_{\ell,i}\boldsymbol{n}_{\ell,i}\Omega_{kj}\boldsymbol{n}^*_{m,i}\boldsymbol{u}_{m,i}] = \begin{cases} 0 & \ell \neq m \\ \sigma^2_{n,\ell}\mathrm{Tr}(\Omega_{kj})R_{u,\ell} & \ell = m \end{cases} \tag{A.55}$$

$$\mathbb{E}[\boldsymbol{n}^*_{\ell,i}\boldsymbol{n}_{\ell,i}\Omega_{kj}\boldsymbol{n}^*_{m,i}\boldsymbol{n}_{m,i}] = \begin{cases} \sigma^2_{n,\ell}\Omega_{kj}\sigma^2_{n,m} & \ell \neq m \\ \beta\sigma^2_{n,\ell}\Omega_{kj}\sigma^2_{n,m} + \sigma^2_{n,\ell}I\,\mathrm{Tr}(\Omega_{kj}\sigma^2_{n,\ell}I) & \ell = m \end{cases} \tag{A.56}$$

and

$$\mathbb{E}[\sigma^2_{n,\ell}I\Omega_{kj}\boldsymbol{n}^*_{m,i}\boldsymbol{n}_{m,i}] = \sigma^2_{n,\ell}\Omega_{kj}\sigma^2_{n,m} \tag{A.57}$$

We note that $\Omega_{k,j} = w^o_k\, w^{o*}_j$ where $w^o_j = w^o_k,\ \forall\, k,j \in \{1,2,\dots N\}$. Therefore,

$$\Omega_{\ell k} = \Omega_{mn}, \quad \forall \ell,k,m,n \in \{1,2,\cdots,N\} \tag{A.58}$$

and $\text{Tr}(\Omega_{kj}) = \|w^o\|^2$. As a result:

$$\Pi_{k,j} = \sum_{\ell} c_{\ell,k} c_{\ell,j} \left\{ \sigma^2_{n,\ell} \|w^o\|^2 \Big(R_{u,\ell} + \sigma^2_{n,\ell} I \Big) + (\beta - 1)\sigma^4_{n,\ell} w^o w^{o*} \right\} \tag{A.59}$$

A.6 Derivation of (4.67)

Computing the block vectorized version of the first three terms in (4.65) is straightforward. We focus on finding the block vectorization of the fourth term which is:

$$\text{bvec}(\mathbb{E}[\mathcal{A}_1 \boldsymbol{\mathcal{R}}_i Q \boldsymbol{\mathcal{R}}_i \mathcal{A}_1^T]) = (\mathcal{A}_1 \otimes_b \mathcal{A}_1)\text{bvec}(\mathcal{X}) \tag{A.60}$$

where $Q = \mathcal{M}\mathcal{A}_2 \Sigma \mathcal{A}_2^T \mathcal{M}$ and $\mathcal{X} = \mathbb{E}[\boldsymbol{\mathcal{R}}_i Q \boldsymbol{\mathcal{R}}_i]$. The (k,ℓ)-th block of this latter is given by:

$$\mathcal{X}_{k,\ell} = \sum_{m} \sum_{n} c_{m,k} c_{n,\ell} \mathbb{E}\Big[(\boldsymbol{z}^*_{m,i} \boldsymbol{z}_{m,i} - \sigma^2_{n,m} I_M) Q_{k,\ell} (\boldsymbol{z}^*_{n,i} \boldsymbol{z}_{n,i} - \sigma^2_{n,n} I_M) \Big] \tag{A.61}$$

This is equivalent to:

$$\begin{aligned} \mathcal{X}_{k,\ell} = \sum_{m} \sum_{n} c_{m,k} c_{n,\ell} \Big(& \mathbb{E}[\boldsymbol{z}^*_{m,i} \boldsymbol{z}_{m,i} Q_{k,\ell} \boldsymbol{z}^*_{n,i} \boldsymbol{z}_{n,i}] + \mathbb{E}[\sigma^2_{n,m}\, \sigma^2_{n,n}\, Q_{k,\ell}] - \mathbb{E}[\sigma^2_{n,n} \boldsymbol{z}^*_{m,i} \boldsymbol{z}_{m,i} Q_{k,\ell}] \\ & - \mathbb{E}[\sigma^2_{n,m} Q_{k,\ell} \boldsymbol{z}^*_{n,i} \boldsymbol{z}_{n,i}] \Big) \end{aligned} \tag{A.62}$$

When the $\{\boldsymbol{u}_{k,i}\}$ are zero mean circular complex-valued Gaussian random vectors, for any Hermitian matrix Γ of compatible dimensions, it holds that [89]:

$$\mathbb{E}\Big[(\boldsymbol{u}^*_{m,i} \boldsymbol{u}_{m,i}) \Gamma (\boldsymbol{u}^*_{m,i} \boldsymbol{u}_{m,i}) \Big] = \beta R_{u,m} \Gamma R_{u,m} + R_{u,m} \text{Tr}(\Gamma R_{u,m}) \tag{A.63}$$

where $\beta = 1$ for complex regressors and $\beta = 2$ if the regressors are real. From this expression, we deduce that:

$$\mathbb{E}[\boldsymbol{u}^*_{m,i} \boldsymbol{u}_{m,i} \Gamma \boldsymbol{u}^*_{n,i} \boldsymbol{u}_{n,i}] = R_{u,m} \Gamma R_{u,n} + \delta_{mn}(\beta - 1) R_{u,m} \Gamma R_{u,m} + \delta_{mn} R_{u,m} \text{Tr}(\Gamma R_{u,m}) \tag{A.64}$$

where $\delta_{mn} = 1$ if $m = n$ and zero otherwise. By using this relation, we obtain:

$$\begin{aligned}\mathcal{X}_{k,\ell} =& \left(\sum_m \sum_n c_{m,k} c_{n,\ell} R_{u,m} Q_{k,\ell} R_{u,n}\right) + \sum_m c_{m,k} c_{m,\ell} \Big[(\beta - 1) R_{u,m} Q_{k,\ell} R_{u,m} \\ &+ (\sigma^2_{n,m} I_M + R_{u,m}) \text{Tr}[Q_{k,\ell}(\sigma^2_{n,m} I_M + R_{u,m})] + (\beta - 1)\sigma^2_{n,m}\, \sigma^2_{n,m}\, Q_{k,\ell}\Big]\end{aligned} \tag{A.65}$$

Some algebra gives

$$\text{bvec}(\mathcal{X}) = Y(\mathcal{M} \otimes_b \mathcal{M})(\mathcal{A}_2 \otimes_b \mathcal{A}_2)\sigma \tag{A.66}$$

where

$$\begin{aligned}Y =& \Bigg\{ (\mathcal{R}^T \otimes_b \mathcal{R}) + \sum_{m=1}^{N} \Big[\text{diag}\{\text{vec}(C_m \mathbb{1}_{N\times N} C_m)\}\Big] \otimes \Big[(\beta - 1)(R^T_{u,m} \otimes R_{u,m} + \sigma^4_{n,m} I_M \otimes I_M) \\ &+ (r_m + \sigma^2_{n,m} q)(r^*_m + \sigma^2_{n,m} q^T)\Big]\Bigg\}\end{aligned} \tag{A.67}$$

In this expression, $C_m = \text{diag}(e^T_m C)$, e_m is a basis vector in $\mathbb{R}^N$ with entry one at position m, $r_m = \text{vec}(R_{u,m})$ and $\mathbb{1}_{N\times N}$ is the $N \times N$ matrix with unit entries. Substituting (A.66) into (A.60) gives

$$(\mathcal{A}_1 \otimes_b \mathcal{A}_1)\text{bvec}(\mathcal{X}) = \Delta\mathcal{F}\sigma \tag{A.68}$$

where

$$\Delta\mathcal{F} = (\mathcal{A}_1 \otimes_b \mathcal{A}_1) Y (\mathcal{M} \otimes_b \mathcal{M})(\mathcal{A}_2 \otimes_b \mathcal{A}_2) \tag{A.69}$$

Using (A.68) and the block vectorized of the first three terms of (4.65) leads to (4.67).

A.7 Computation of $R_{v,k}$

To obtain $R_{v,k}$ in (5.64), we need to compute the expectation

$$\mathbb{E}\left[\boldsymbol{a}^2_{\ell,k}(i)\, |\hat{\boldsymbol{g}}_{\ell,k}(i)|^2\right] = \mathbb{E}\left[\frac{\boldsymbol{a}^2_{\ell,k}(i)}{\frac{P_t}{r^{\alpha}_{\ell,k}} |\hat{\boldsymbol{h}}_{\ell,k}(i)|^2}\right] \tag{A.70}$$

for $\ell \in \mathcal{N}_k \backslash k$. For the case $\ell = k$, we have $R_{v,\ell k} = 0$ and hence the expectation of $[\boldsymbol{a}^2_{\ell,k}(i)\,|\hat{\boldsymbol{g}}_{\ell,k}(i)|^2]R_{v,\ell k}$ in (5.64) is zero. For $\ell \neq k$, we proceed as follows. Since the joint probability distribution function of the numerator and denominator in (A.70) is unknown, the expectation can be approximated using one of two ways. In the first method, we can resort to computer simulations. In the second method, we can resort to a Taylor series approximation as follows. We introduce the real-valued auxiliary variable $\boldsymbol{x} = \boldsymbol{a}^2_{\ell,k}(i)$. Considering the combination rule (5.23), the expectation of $\boldsymbol{x}$ when $\ell \neq k$ will be:

$$\mathbb{E}[\boldsymbol{x}] = \gamma^2_{\ell k} p_{\ell,k} \tag{A.71}$$

To compute the variance and expectation of the denominator in (A.70), we let the exponential distribution function $f_{\boldsymbol{y}}(y)$ with parameter λ given by (5.28) denote the pdf of $\boldsymbol{y} = |\hat{\boldsymbol{h}}_{\ell,k}(i)|^2$, i.e.,

$$f_{\boldsymbol{y}}(y) = \lambda_{\ell,k} e^{-\lambda_{\ell,k} y}, \text{ for } y \in [0, \infty) \tag{A.72}$$

We also let $f^{(t)}_{\boldsymbol{y}}(y)$ represent the pdf of $\boldsymbol{y}$ for $y \in [\nu_{\ell,k}, \infty)$. It can be verified that $f^{(t)}_{\boldsymbol{y}}(y)$ represents a truncated exponential distribution and is given by:

$$f^{(t)}_{\boldsymbol{y}}(y) = \lambda_{\ell,k} e^{-\lambda_{\ell,k}(y-\nu_{\ell,k})}, \text{ for } y \in [\nu_{\ell,k}, \infty) \tag{A.73}$$

If we now define

$$\boldsymbol{z} = \frac{P_t}{r^{\alpha}_{\ell,k}} \boldsymbol{y} \tag{A.74}$$

Then, the pdf of $\boldsymbol{z}$ can be computed as [126]:

$$f_{\boldsymbol{z}}(z) = \left|\frac{dy}{dz}\right| f^{(t)}_{\boldsymbol{y}}(g^{-1}(z)) \tag{A.75}$$

where

$$\frac{dy}{dz} = \frac{r^{\alpha}_{\ell,k}}{P_t} \text{ and } g^{-1}(z) = \frac{r^{\alpha}_{\ell,k}}{P_t} z \tag{A.76}$$

Therefore,

$$f_{\boldsymbol{z}}(z) = \frac{r_{\ell,k}^{\alpha}}{P_t}\lambda_{\ell,k}e^{-\lambda_{\ell,k}(\frac{r_{\ell,k}^{\alpha}}{P_t}z-\nu_{\ell,k})}, \text{ for } z \in \left[\frac{P_t}{r_{\ell,k}^{\alpha}}\nu_{\ell,k},\, \infty\right) \tag{A.77}$$

Using this distribution the mean and variance of $\boldsymbol{z}$ will be [126]:

$$\mathbb{E}[\boldsymbol{z}] = \frac{P_t}{r_{\ell,k}^{\alpha}}\left(\frac{1}{\lambda_{\ell,k}} + \nu_{\ell,k}\right) \tag{A.78}$$

$$\text{var}(\boldsymbol{z}) = \left(\frac{P_t}{r_{\ell,k}^{\alpha}\lambda_{\ell,k}}\right)^2 \tag{A.79}$$

We can now proceed to approximate the expectation (A.70) by defining

$$f(\boldsymbol{x},\boldsymbol{z}) = \frac{\boldsymbol{x}}{\boldsymbol{z}} \tag{A.80}$$

and employing the second order Taylor series expansion given below:

$$\mathbb{E}[f(\boldsymbol{x},\boldsymbol{z})] \approx \frac{\mathbb{E}[\boldsymbol{x}]}{\mathbb{E}[\boldsymbol{z}]} - \frac{1}{(\mathbb{E}[\boldsymbol{z}])^2}\text{cov}(\boldsymbol{x},\boldsymbol{z}) + \frac{\mathbb{E}[\boldsymbol{x}]}{(\mathbb{E}[\boldsymbol{z}])^3}\text{var}(\boldsymbol{z}) \tag{A.81}$$

Substituting, $\mathbb{E}[\boldsymbol{x}]$, $\mathbb{E}[\boldsymbol{z}]$, $\text{cov}(\boldsymbol{x},\boldsymbol{z})$ and $\text{var}(\boldsymbol{z})$ into (A.81), we then arrive at:

$$\begin{aligned}\mathbb{E}[f(\boldsymbol{x},\boldsymbol{z})] &\approx \mathbb{E}\left[\boldsymbol{a}_{\ell,k}^2(i)\,|\hat{\boldsymbol{g}}_{\ell,k}(i)|^2\right] \\ &\approx \gamma_{\ell k}^2 p_{\ell,k}\left(\frac{1}{\frac{P_t}{r_{\ell,k}^{\alpha}}(\frac{1}{\lambda_{\ell,k}}+\nu_{\ell,k})} - \frac{\nu_{\ell,k}}{\frac{P_t}{r_{\ell,k}^{\alpha}}(\frac{1}{\lambda_{\ell,k}}+\nu_{\ell,k})^2} + \frac{1}{\frac{P_t}{r_{\ell,k}^{\alpha}}\lambda_{\ell,k}^2(\frac{1}{\lambda_{\ell,k}}+\nu_{\ell,k})^3}\right)\end{aligned} \tag{A.82}$$

A.8 Derivation of $\mathcal{F}$ for Gaussian Data

First, we note that when $\boldsymbol{u}_{k,i}$ are zero mean circular complex-valued Gaussian random vectors and i.i.d. over time, then for any Hermitian matrix Γ of compatible dimensions it holds that [89]:

$$\mathbb{E}[\boldsymbol{u}_{k,i}^*\boldsymbol{u}_{k,i}\Gamma\boldsymbol{u}_{k,i}^*\boldsymbol{u}_{k,i}] = \beta(R_{u,k}\Gamma R_{u,k}) + R_{u,k}\text{Tr}(\Gamma R_{u,k}) \tag{A.83}$$

where $\beta = 1$ for complex regressors and $\beta = 2$ when the regressors are real. Using (A.83) and spatial independence of the regression data we have

$$\mathbb{E}[\boldsymbol{u}_{k,i}^* \boldsymbol{u}_{k,i} \Gamma \boldsymbol{u}_{\ell,i}^* \boldsymbol{u}_{\ell,i}] = R_{u,k} \Gamma R_{u,\ell} + \delta_{k\ell}(\beta - 1) R_{u,k} \Gamma R_{u,k} + \delta_{k\ell} R_{u,k} \text{Tr}(\Gamma R_{u,k}) \tag{A.84}$$

where $\delta_{k\ell}$ is the Dirac delta sequence. To compute $\bar{\mathcal{F}}$, we first introduce

$$\boldsymbol{\mathcal{L}}_i = (I - \mathcal{M}\boldsymbol{\mathcal{R}}_i)\mathcal{Q}(I - \mathcal{M}\boldsymbol{\mathcal{R}}_i) \tag{A.85}$$

where $\mathcal{Q}$ is an arbitrary deterministic Hermitian matrix. We now note that

$$\begin{aligned} \text{bvec}\left(\mathbb{E}[\boldsymbol{\mathcal{L}}_i]\right) &\overset{(i)}{=} \mathbb{E}\left[(I - \mathcal{M}\boldsymbol{\mathcal{R}}_i)^T \otimes_b (I - \mathcal{M}\boldsymbol{\mathcal{R}}_i)\right] \text{bvec}(\mathcal{Q}) \\ &\overset{(ii)}{=} \bar{\mathcal{F}} \; \text{bvec}(\mathcal{Q}) \end{aligned} \tag{A.86}$$

where (ii) obtained by comparing the expectation term on the right hand side of (i) with definition (5.72). We proceed by taking expectation of both sides of $(A.85)$, i.e.,

$$\mathbb{E}[\boldsymbol{\mathcal{L}}_i] = \mathcal{Q} - \mathcal{R}\mathcal{M}\mathcal{Q} - \mathcal{Q}\mathcal{M}\mathcal{R} + \mathbb{E}\left[\boldsymbol{\mathcal{R}}_i \mathcal{M}\mathcal{Q}\mathcal{M}\boldsymbol{\mathcal{R}}_i\right] \tag{A.87}$$

To compute the block vectorization of the last term on the right hand side of (A.87), we introduce the block partitioned matrix $\mathcal{Q}' = \mathcal{M}\mathcal{Q}\mathcal{M}$ with blocks $\mathcal{Q}'_{k\ell}$ and use (A.84) to obtain (A.88), where $r_k = \text{vec}(R_{u,k})$.

$$\begin{aligned} \text{bvec}\Big(\mathbb{E}\Big[\boldsymbol{\mathcal{R}}_i \mathcal{Q}' \boldsymbol{\mathcal{R}}_i\Big]\Big) = &\Big\{(\mathcal{R}^T \otimes_b \mathcal{R}) + \sum_{k=1}^{N} \Big[\text{diag}\Big(\text{vec}(\text{diag}(e_k)))\Big] \otimes \Big[(\beta - 1)(R_{k,u}^T \otimes R_{k,u}) + r_k r_k^*\Big]\Big\} \\ &\times (\mathcal{M} \otimes_b \mathcal{M}) \text{bvec}(\mathcal{Q}) \end{aligned} \tag{A.88}$$

Now, using (A.87), we can write:

$$\text{bvec}(\mathbb{E}[\boldsymbol{\mathcal{L}}_i]) = \Big(I - I \otimes_b \mathcal{M}\mathcal{R} - \mathcal{R}^T\mathcal{M} \otimes_b I\Big)\text{bvec}(\mathcal{Q}) + \text{bvec}\Big(\mathbb{E}\Big[\boldsymbol{\mathcal{R}}_i \mathcal{Q}' \boldsymbol{\mathcal{R}}_i\Big]\Big) \tag{A.89}$$

From (A.86), (A.88) and (A.89) and using the fact that the real vector space of Hermitian matrices is isomorphic to $\mathbb{R}^{N^2 \times 1}$, we arrive at (5.74).

A.9 Computation of $\mathcal{D}$

We expand $\mathcal{D} = \mathbb{E}[\boldsymbol{\mathcal{D}}_i]$ in (6.22) as:

$$\mathcal{D} = \Big\{\mathbb{E}[\boldsymbol{A}_i \otimes \boldsymbol{A}_i] + \mathbb{E}[\boldsymbol{A}_i \otimes \boldsymbol{E}_i^{*T}] + \mathbb{E}[\boldsymbol{E}_i \otimes \boldsymbol{A}_i] + \mathbb{E}[\boldsymbol{E}_i \otimes \boldsymbol{E}_i^{*T}]\Big\} \otimes I_{M^2} \tag{A.90}$$

The $(r,\, z)$-th entry of $\mathbb{E}[\boldsymbol{A}_i \otimes \boldsymbol{A}_i]$, denoted by $f_{r,z}$, is:

$$f_{r,z} = \mathbb{E}[\boldsymbol{a}_{\ell,k}(i)\boldsymbol{a}_{m,n}(i)] \tag{A.91}$$

where the relation between (r, z) and $\ell, k)$ is:

$$r = (\ell - 1)N + m, \text{ and } z = (k-1)N + n \tag{A.92}$$

When $k \neq n$, entries $\boldsymbol{a}_{\ell,k}(i)$ and $\boldsymbol{a}_{m,n}(i)$ come from different columns of $\boldsymbol{A}_i$ and are independent. Hence, in this case, we can write:

$$f_{r,z} = \mathbb{E}\Big[\boldsymbol{a}_{\ell,k}(i)\Big]\,\mathbb{E}\Big[\boldsymbol{a}_{m,n}(i)\Big] \tag{A.93}$$

with

$$\mathbb{E}\Big[\boldsymbol{a}_{j,q}(i)\Big] = \begin{cases} 1 - \sum\limits_{r\in\mathcal{N}_q\backslash q} p_{rq}\gamma_{rq}, & \text{if } j = q \\ p_{jq}\gamma_{jq}, & \text{otherwise} \end{cases} \tag{A.94}$$

When $k = n$, the entries $\boldsymbol{a}_{\ell,k}(i)$ and $\boldsymbol{a}_{m,n}(i)$ come from the same column of $\boldsymbol{A}_i$ and may be dependent. In this case, there are four possibilities:
(1) if $\ell = m$ and $\ell \neq k$:

$$f_{r,z} = \gamma_{\ell,k}^2 p_{\ell,k} \tag{A.95}$$

(2) if $\ell = m$ and $\ell = k$:

$$f_{r,z} = \mathbb{E}\Bigg[\Bigg(1 - \sum_{\ell\in\mathcal{N}_k\backslash k} \boldsymbol{a}_{\ell,k}(i)\Bigg)\Bigg(1 - \sum_{\ell\in\mathcal{N}_k\backslash k} \boldsymbol{a}_{\ell,k}(i)\Bigg)\Bigg] \tag{A.96}$$

$$= 1 - 2\sum_{\ell\in\mathcal{N}_k\backslash k} p_{\ell,k}(\gamma_{\ell,k} - \gamma_{\ell,k}^2) - \sum_{\ell\in\mathcal{N}_k\backslash k} p_{\ell,k}^2\gamma_{\ell,k}^2 + \sum_{(\ell\in\mathcal{N}_k\backslash k)}\sum_{(m\in\mathcal{N}_k\backslash k)} p_{\ell,k}p_{m,k}\gamma_{\ell,k}\gamma_{m,k} \tag{A.97}$$

(3) if $\ell \neq m$ and $\ell \neq k$ and $m \neq n$:

$$f_{r,z} = \gamma_{\ell,k}\gamma_{m,n}p_{\ell,k}p_{m,n} \tag{A.98}$$

(4) if $\ell \neq m$ and $\ell = k$ and $m \neq n$:

$$f_{r,z} = \mathbb{E}\Big[\Big(1 - \sum_{j\in\mathcal{N}\backslash k} \boldsymbol{a}_{j,k}(i)\Big)\boldsymbol{a}_{m,n}(i)\Big] = \gamma_{m,n}p_{m,n}\Bigg(1 - \gamma_{m,n} + \sum_{j\in\mathcal{N}_k\backslash\{k,m\}} \gamma_{j,k}p_{j,k}\Bigg) \tag{A.99}$$

The $(r,\, z)$-th entry of $\mathbb{E}[\boldsymbol{A}_i \otimes \boldsymbol{E}_i^{*T}]$, denoted by $x_{r,z}$, can be expressed as:

$$\begin{aligned} x_{r,z} &= -\mathbb{E}\Big[\boldsymbol{a}_{\ell,k}(i)\, \boldsymbol{a}_{m,n}(i)\hat{\boldsymbol{g}}^*_{m,n}(i)\boldsymbol{v}^{(y)*}_{m,n}(i)\Big] \\ &= -\mathbb{E}\left[\boldsymbol{a}_{\ell,k}(i)\boldsymbol{a}_{m,n}(i)\frac{\sqrt{\frac{r^\alpha}{P_t}}\boldsymbol{h}_{m,n}(i)\boldsymbol{v}^{(y)*}_{m,n}(i) + \frac{r^\alpha}{P_t}|\boldsymbol{v}^{(y)}_{m,n}(i)|^2}{\Big|\boldsymbol{h}_{m,n}(i) + \sqrt{\frac{r^\alpha}{P_t}}\boldsymbol{v}^{(y)}_{m,n}(i)\Big|^2}\,\Bigg|\, \Big|\boldsymbol{h}_{m,n}(i) + \sqrt{\frac{r^\alpha}{P_t}}\boldsymbol{v}^{(y)}_{m,n}(i)\Big|^2 \geq \nu_{m,n}\right] \end{aligned} \tag{A.100}$$

Likewise, the entries of $\mathbb{E}[\boldsymbol{E}_i \otimes \boldsymbol{A}_i]$ and $\mathbb{E}[\boldsymbol{E}_i \otimes \boldsymbol{E}_i^{*T}]$ can be expressed in terms of the combination weights, channel coefficients and the estimation error. We can follow the argument presented in Remark 5.1 to show that the right hand side of (A.100) as well as the entries of $\mathbb{E}[\boldsymbol{E}_i \otimes \boldsymbol{A}_i]$ and $\mathbb{E}[\boldsymbol{E}_i \otimes \boldsymbol{E}_i^{*T}]$ are invariant with respect to time and have finite values.

References

[1] D. Estrin, L. Girod, G. Pottie, and M. Srivastava, “Instrumenting the world with wireless sensor networks,” in *Proc. IEEE Int. Conf. on Acoust., Speech, Signal Process.*, May 2001, pp. 2033–2036.

[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless sensor networks: A survey,” *IEEE Commun. Mag.*, vol. 38, pp. 393–422, Mar. 2002.

[3] M. Rabbat and R. Nowak, “Distributed optimization in sensor networks,” in *Proc. Int. Symp. on Inf. Proc. in Sensor Networks (IPSN)*, Berkeley, California, Apr. 2004, pp. 20–27.

[4] A. H. Sayed and C. G. Lopes, “Adaptive processing over distributed networks,” *IEICE Trans. Fund. of Elect. and Comm. Comp. Sci.*, vol. 90, pp. 1504–1510, Aug. 2007.

[5] D. P. Bertsekas, “A new class of incremental gradient methods for least squares problems,” *SIAM J. Optimization*, vol. 7, no. 4, pp. 913–926, 1997.

[6] F. Cattivelli and A. H. Sayed, “Modeling bird flight formations using diffusion adaptation,” *IEEE Trans. on Signal Processing*, vol. 59, no. 5, pp. 2038–2051, May 2011.

[7] S. Y. Tu and A. H. Sayed, “Cooperative prey herding based on diffusion adaptation,” in *Proc. IEEE Int. Conf. on Acoust., Speech, Signal Process.*, Prague, Czech Republic, May 2011, pp. 3752–3755.

[8] J. Chen, X. Zhao, and A. H. Sayed, “Bacterial motility via diffusion adaptation,” in *Proc. 44th Asilomar Conf. on Signals, Systems and Computers*, Pacific Grove, CA, Nov. 2010, pp. 1930–1934.

[9] F. Cattivelli and A. H. Sayed, “Distributed detection over adaptive networks using diffusion adaptation,” *IEEE Trans. on Signal Processing*, vol. 59, no. 5, pp. 1917–1932, May 2011.

[10] A. Bertrand and M. Moonen, “Distributed adaptive node-specific signal estimation in fully connected sensor networkspart I: Sequential node updating,” *IEEE Trans. on Signal Processing*, vol. 58, no. 10, pp. 5277–5291, Nov. 2010.

[11] ——, “Distributed adaptive node-specific signal estimation in fully connected sensor networksPart II: Simultaneous and asynchronous node updating,” *IEEE Trans. on Signal Processing*, vol. 58, no. 10, pp. 5292–5306, Nov. 2010.

[12] R. Abdolee and B. Champagne, “Distributed blind adaptive algorithms based on constant modulus for wireless sensor networks,” in *Proc. IEEE Int. Conf. on Wireless and Mobile Commun.*, Valencia, Spain, Sept. 2010, pp. 303–308.

[13] R. Abdolee, S. Saur, B. Champagne, and A. H. Sayed, “Diffusion LMS localization and tracking algorithm for wireless cellular networks,” in *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Process.*, Vancouver, Canada, May 2013, pp. 4598–4602.

[14] N. Takahashi and I. Yamada, “Link probability control for probabilistic diffusion least-mean squares over resource-constrained networks,” in *Proc. IEEE Int. Conf. on Acoust., Speech, Signal Process.*, Dallas, USA, Mar. 2010, pp. 3518–3521.

[15] A. Nedic and D. P. Bertsekas, “Incremental subgradient methods for nondifferentiable optimization,” *SIAM J. Optim.*, vol. 12, no. 1, pp. 109–138, Jan. 2001.

[16] M. G. Rabbat and R. D. Nowak, “Quantized incremental algorithms for distributed optimization,” *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 4, pp. 798–808, Apr. 2005.

[17] C. G. Lopes and A. H. Sayed, “Incremental adaptive strategies over distributed networks,” *IEEE Trans. on Signal Processing*, vol. 55, no. 8, pp. 4064–4077, Aug. 2007.

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

[19] L. Xiao, S. Boyd, and S. Lall, “A space-time diffusion scheme for peer-to-peer least-squares estimation,” in *Proc. of Int. Symp. on Inf. Process. in Sensor networks (IPSN)*, Nashville, TN, April 2006, pp. 168–176.

[20] T. C. Aysal, M. J. Coates, and M. G. Rabbat, “Distributed average consensus with dithered quantization,” *IEEE Trans. on Signal Processing*, vol. 56, no. 10, pp. 4905–4918, Oct. 2008.

[21] T. C. Aysal, B. N. Oreshkin, and M. J. Coates, “Accelerated distributed average consensus via localized node state prediction,” *IEEE Trans. on Signal Processing*, vol. 57, no. 4, pp. 1563–1576, 2009.

[22] P. Braca, S. Marano, and V. Matta, "Running consensus in wireless sensor networks," in *Proc. Int. Conf. on Information Fusion*, Cologne, Germany, June 2008, pp. 1–6.

[23] S. Sardellitti, M. Giona, and S. Barbarossa, "Fast distributed average consensus algorithms based on advection-diffusion processes," *IEEE Trans. on Signal Processing*, vol. 58, no. 2, pp. 826–842, Feb. 2010.

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

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

[26] S. Kar and J. Moura, "Convergence rate analysis of distributed gossip (linear parameter) estimation: Fundamental limits and tradeoffs," *IEEE J. Selected Topics on Signal Processing*, vol. 5, no. 4, pp. 674–690, Aug. 2011.

[27] 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, no. 7, pp. 3122–3136, July 2008.

[28] S. Chouvardas, K. Slavakis, and S. Theodoridis, "Adaptive robust distributed learning in diffusion sensor networks," *IEEE Trans. on Signal Processing*, vol. 59, no. 10, pp. 4692–4707, Oct. 2011.

[29] J. Chen and A. H. Sayed, "Diffusion adaptation strategies for distributed optimization and learning over networks," *IEEE Trans. on Signal Processing*, vol. 60, no. 8, pp. 4289–4305, Aug. 2012.

[30] S. Y. Tu and A. H. Sayed, "Diffusion strategies outperform consensus strategies for distributed estimation over adaptive networks," *IEEE Trans. on Signal Processing*, vol. 60, no. 12, pp. 6127–6234, Dec. 2012.

[31] X. Zhao and A. H. Sayed, "Performance limits for distributed estimation over LMS adaptive networks," *IEEE Trans. on Signal Processing*, vol. 60, no. 10, pp. 5107–5124, Oct. 2012.

[32] A. H. Sayed, S. Y. Tu, J. Chen, X. Zhao, and Z. Towfic, "Diffusion strategies for adaptation and learning over networks," *IEEE Signal Processing Magazine*, vol. 30, no. 5, pp. 155–171, May 2013.

[33] F. S. Cattivelli and A. H. Sayed, "Diffusion LMS strategies for distributed estimation," *IEEE Trans. on Signal Processing*, vol. 58, no. 3, pp. 1035–1048, Mar. 2010.

[34] L. Li, J. A. Chambers, C. G. Lopes, and A. H. Sayed, "Distributed estimation over an adaptive incremental network based on the affine projection algorithm," *IEEE Trans. on Signal Processing*, vol. 58, no. 1, pp. 151–164, Jan. 2010.

[35] A. H. Sayed and C. G. Lopes, "Distributed recursive least-squares strategies over adaptive networks," in *Proc. Asilomar Conf. on Signals, Syst., Comput.*, Aug. 2006, pp. 233–237.

[36] C. G. Lopes and A. H. Sayed, "Diffusion least-mean squares over adaptive networks," in *Proc. IEEE Int. Conf. on Acoust., Speech, Signal Process.*, vol. 3, Honolulu, Hawaii, Apr. 2007, pp. 917–920.

[37] 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, no. 5, pp. 1865–1877, May 2008.

[38] T. Lee and J. Seinfeld, "Estimation of two-phase petroleum reservoir properties by regularization," *J. of Computational Physics*, vol. 69, no. 2, pp. 397–419, Feb. 1987.

[39] E. Holmes, M. Lewis, J. Banks, and R. Veit, "Partial differential equations in ecology: Spatial interactions and population dynamics," *Ecology*, vol. 75, no. 1, pp. 17–29, Jan. 1994.

[40] N. Van Kampen, "Diffusion in inhomogeneous media," *J. of Physics and Chemistry of Solids*, vol. 49, no. 1, pp. 673–677, June 1988.

[41] C. Chung and C. Kravaris, "Identification of spatially discontinuous parameters in second-order parabolic systems by piecewise regularisation," *Inverse Problems*, vol. 4, pp. 973–994, Feb. 1988.

[42] G. Richter, "Numerical identification of a spatially-varying diffusion coefficient," *Math. Comput.*, vol. 36, no. 154, pp. 375–386, Apr. 1981.

[43] V. Isakov and S. Kindermann, "Identification of the diffusion coefficient in a one-dimensional parabolic equation," *Inverse Problems*, vol. 16, pp. 665–680, June 2000.

[44] S. Sagara and K. Wada, "On-line modified least-squares parameter estimation of linear discrete dynamic systems," *Int. J. of Control*, vol. 25, no. 3, pp. 329–343, Mar. 1977.

[45] W. Zheng and C. Feng, "Unbiased parameter estimation of linear systems in the presence of input and output noise," *Int. J. of Adaptive Control and Sig. Process.*, vol. 3, no. 3, pp. 231–251, Mar. 1989.

[46] C. Davila, "An efficient recursive total least squares algorithm for FIR adaptive filtering," *IEEE Trans. on Signal Processing*, vol. 42, pp. 268–280, 1994.

[47] H. So, "Modified LMS algorithm for unbiased impulse response estimation in nonstationary noise," *Electronics Letters*, vol. 35, no. 10, pp. 791–792, Oct. 1999.

[48] D. Feng, Z. Bao, and X. Zhang, "Modified RLS algorithm for unbiased estimation of FIR system with input and output noise," *Electronics Letters*, vol. 36, no. 3, pp. 273–274, Mar. 2000.

[49] L. Jia, M. Ikenoue, C. Jin, and K. Wada, "On bias compensated least squares method for noisy input-output system identification," in *Proc. of the* 40^{th} *IEEE Conf. on Decision and Control*, Orlando, Florida, Dec. 2001, pp. 3332–3337.

[50] S. Jo and S. Kim, "Consistent normalized least mean square filtering with noisy data matrix," *IEEE Trans. on Signal Processing*, vol. 53, pp. 2112–2123, 2005.

[51] A. Bertrand and M. Moonen, "Consensus-based distributed total least squares estimation in ad hoc wireless sensor networks," *IEEE Trans. on Signal Processing*, pp. 2320–2330, May 2011.

[52] A. Bertrand, M. Moonen, and A. H. Sayed, "Diffusion bias-compensated RLS estimation over adaptive networks," *IEEE Trans. on Signal Processing*, pp. 5212–5224, Nov. 2011.

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

[54] R. Abdolee and B. Champagne, "Diffusion LMS algorithms for sensor networks over non-ideal inter-sensor wireless channels," in *Proc. of Int. Conf. on Dist. Computing in Sensor Systems and Workshops*, Barcelona, Spain, June 2011, pp. 1–6.

[55] A. Khalili, M. Tinati, A. Rastegarnia, and J. Chambers, "Transient analysis of diffusion least-mean squares adaptive networks with noisy channels," *Int. Jnl. of Adaptive Cont. and Signal Proc.*, Feb. 2012.

[56] ——, "Steady-state analysis of diffusion LMS adaptive networks with noisy links," *IEEE Trans. on Signal Processing*, vol. 60, no. 2, pp. 974–979, Feb. 2012.

[57] I. D. Schizas, G. Mateos, and G. B. Giannakis, "Distributed LMS for consensus-based in-network adaptive processing," *IEEE Trans. on Signal Processing*, vol. 57, no. 6, pp. 2365–2382, June 2009.

[58] 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*, pp. 4583–4588, Nov. 2009.

[59] S. Y. Tu and A. H. Sayed, “Optimal combination rules for adaptation and learning over networks,” in *submitted to IEEE CAMSAP Workshop*, San Juan, Puerto Rico, Dec. 2011.

[60] L. Xiao, S. Boyd, and S. Lall, “A scheme for robust distributed sensor fusion based on average consensus,” in *Proc. Fourth International Symposium on Information Processing in Sensor Networks*, 2005, pp. 63–70.

[61] P. Yang, R. A. Freeman, and K. M. Lynch, “Optimal information propagation in sensor networks,” in *Proc. IEEE International Conference onRobotics and Automation (ICRA)*, Orlando, FL, May 2006, pp. 3122–3127.

[62] D. Jakovetic, J. Xavier, and J. M. Moura, “Weight optimization for consensus algorithms with correlated switching topology,” *IEEE Trans. on Signal Processing*, vol. 58, no. 7, pp. 3788–3801, July 2010.

[63] N. Takahashi, I. Yamada, and A. H. Sayed, “Diffusion least-mean squares with adaptive combiners: Formulation and performance analysis,” *IEEE Trans. on Signal Processing*, vol. 58, no. 9, pp. 4795–4810, Sept. 2010.

[64] S.-Y. Tu and A. H. Sayed, “Optimal combination rules for adaptation and learning over networks,” in *Proc. 4-th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, San Juan, PR, Dec. 2011, pp. 317–320.

[65] R. Koning, H. Neudecker, and T. Wansbeek, “Block Kronecker products and the vecb operator,” *Linear Algebra and Its Applications*, vol. 149, pp. 165–184, Apr. 1991.

[66] M. H. DeGroot, “Reaching a consensus,” *Journal of the American Statistical Association*, vol. 69, no. 345, pp. 118–121, 1974.

[67] T. C. Aysal, M. E. Yildiz, A. D. Sarwate, and A. Scaglione, “Broadcast gossip algorithms for consensus,” *IEEE Trans. on Signal Processing*, vol. 57, no. 7, pp. 2748–2761, July 2009.

[68] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, “Randomized gossip algorithms,” *IEEE Trans. on Information Theory*, vol. 52, no. 6, pp. 2508–2530, June 2006.

[69] R. Olfati-Saber, J. A. Fax, and R. M. Murray, “Consensus and cooperation in networked multi-agent systems,” *Proc. of the IEEE*, vol. 95, no. 1, pp. 215–233, Jan. 2007.

[70] L. Xiao and S. Boyd, “Fast linear iterations for distributed averaging,” *Systems & Control Letters*, vol. 53, no. 1, pp. 65–78, 2004.

[71] V. Gazi and K. M. Passino, “Stability analysis of social foraging swarms,” *IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 34, no. 1, pp. 539–557, Feb. 2004.

[72] A. Jadbabaie, J. Lin, and A. S. Morse, “Coordination of groups of mobile autonomous agents using nearest neighbor rules,” *IEEE Trans. on Automatic Control*, vol. 48, no. 6, pp. 988–1001, June 2003.

[73] R. Olfati-Saber, “Flocking for multi-agent dynamic systems: Algorithms and theory,” *IEEE Trans. on Automatic Control*, vol. 51, no. 3, pp. 401–420, Mar. 2006.

[74] B. N. Oreshkin, M. J. Coates, and M. G. Rabbat, “Optimization and analysis of distributed averaging with short node memory,” *IEEE Trans. on Signal Processing*, vol. 58, no. 5, pp. 2850–2865, May 2010.

[75] D. Ustebay, B. N. Oreshkin, M. J. Coates, and M. G. Rabbat, “Greedy gossip with eavesdropping,” *IEEE Trans. on Signal Processing*, vol. 58, no. 7, pp. 3765–3776, July 2010.

[76] 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, no. 1, pp. 350–364, Jan. 2008.

[77] S. Barbarossa and G. Scutari, “Bio-inspired sensor network design,” *IEEE Signal Processing Magazine*, vol. 24, no. 3, pp. 26–35, Mar. 2007.

[78] A. Nedic and A. Ozdaglar, *Cooperative Distributed Multi-Agent Optimization.* Cambridge University Press, 2010.

[79] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation.* Prentice Hall Inc., 1989.

[80] C. G. Lopes and A. H. Sayed, “Diffusion adaptive networks with changing topologies,” in *Proc. IEEE Int. Conf. on Acoust., Speech, Signal Process.*, Las Vegas, Nevada, Apr. 2008, pp. 3285–3288.

[81] F. S. Cattivelli and A. H. Sayed, “Diffusion strategies for distributed Kalman filtering and smoothing,” *IEEE Transactions on Automatic Control*, vol. 55, pp. 2069–2084, Sept. 2010.

[82] A. H. Sayed, “Diffusion adaptation over networks,” in *Academic Press Library in Signal Processing,* R. Chellapa and S. Theodoridis, *Eds.,* pp. 323–454, Academic Press, Elsevier, 2013 (Also available as *arXiv:1205.4220v2)*, May 2012.

[83] J. Li and A. H. Sayed, “Modeling bee swarming behavior through diffusion adaptation with asymmetric information sharing,” *EURASIP J. on Advances in Signal Processing*, vol. 18, DOI:10.1186/1687-6180-2012-18, 2012.

[84] S. Y. Tu and A. H. Sayed, “Mobile adaptive networks,” *IEEE J. Selected Topics on Signal Processing*, vol. 5, pp. 649–664, Aug. 2011.

[85] J. Chen and A. H. Sayed, “Distributed Pareto optimization via diffusion strategies,” *IEEE J. Selected Topics in Signal Processing*, vol. 7, no. 2, pp. 205–220, Apr. 2013.

[86] K. Srivastava and A. Nedic, “Distributed asynchronous constrained stochastic optimization,” *IEEE J. of Selected Topics in Signal Processing*, vol. 5, no. 4, pp. 772–790, Aug. 2011.

[87] R. Abdolee, B. Champagne, and A. H. Sayed, “Diffusion LMS for source and process estimation in sensor networks,” in *Proc. of IEEE Workshop on Statistical Signal Process.*, Ann Arbor, MI, Aug. 2012, pp. 165–168.

[88] P. Di Lorenzo and A. Sayed, “Sparse distributed learning based on diffusion adaptation,” *IEEE Trans. on Signal Processing*, vol. 61, no. 6, pp. 1419–1433, Mar. 2013.

[89] A. H. Sayed, *Adaptive Filters.* Wiley-IEEE Press, 2008.

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

[91] R. A. Horn and C. R. Johnson, *Matrix Analysis.* Cambridge University press, 2003.

[92] S. S. Ram, A. Nedić, and V. V. Veeravalli, “Distributed stochastic subgradient projection algorithms for convex optimization,” *Journal of Optim. Theory and Applications*, vol. 147, no. 3, pp. 516–545, 2010.

[93] C. D. Meyer, *Matrix Analysis and Applied Linear Algebra.* Society for Industrial Applied Mathematics (SIAM), 2001.

[94] G. H. Golub and C. F. Van Loan, *Matrix Computations.* Johns Hopkins University Press, 1996.

[95] S. Y. Tu and A. H. Sayed, “Adaptive decision-making over complex networks,” in *Proc. Asilomar Conf. on Signals, Systems, and Computers*, Pacific Grove, CA, Nov. 2012, pp. 525–530.

[96] P. D. Lorenzo, S. Barbarossa, and A. H. Sayed, “Decentralized resource assignment in cognitive networks based on swarming mechanisms over random graphs,” *IEEE Trans. on Signal Processing*, vol. 60, no. 7, pp. 3755–3769, July 2012.

[97] M. Alpay and M. Shor, “Model-based solution techniques for the source localization problem,” *IEEE Trans. on Control Systems Technology*, vol. 8, no. 6, pp. 895–904, June 2000.

[98] M. Demetriou, “Process estimation and moving source detection in 2-D diffusion processes by scheduling of sensor networks,” in *Proc. American Control Conf.*, New York City, USA, July 2007, pp. 3432–3437.

[99] M. Demetriou and I. Hussein, “Estimation of spatially distributed processes using mobile spatially distributed sensor network,” *SIAM Journal on Control and Optimization*, vol. 48, no. 1, pp. 266–291, Feb. 2009.

[100] R. Mattheij, S. Rienstra, and J. ten Thije Boonkkamp, *Partial Differential Equations: Modeling, Analysis, Computation*. Society for Industrial Applied Mathematics (SIAM), 2005.

[101] J. Thomas, *Numerical Partial Differential Equations: Finite Difference Methods*. Springer Verlag, 1995.

[102] J. C. Mason and D. C. Handscomb, *Chebyshev Polynomials*. Chapman & Hall/CRC, 2003.

[103] E. Isaaks and R. Srivastava, *An Introduction to Applied Geostatistics*. Oxford University Press, 1989.

[104] S.-J. Kim, E. Dall’Anese, and G. B. Giannakis, “Cooperative spectrum sensing for cognitive radios using kriged Kalman filtering,” *IEEE Selected Topics in Signal Process.*, vol. 5, no. 1, pp. 24–36, Jan. 2011.

[105] N. A. Cressie, *Statistics for Spatial Data*. Wiley, 1993.

[106] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, *Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers*. Now Publishers Inc., 2011.

[107] A. H. Sayed, S.-Y. Tu, J. Chen, X. Zhao, and Z. Towfic, “Diffusion strategies for adaptation and learning over networks,” *IEEE Signal Process. Mag.*, vol. 30, pp. 155–171, May 2013.

[108] D. McLernon, M. Lara, and A. Orozco-Lugo, "On the convergence of the LMS algorithm with a rank-deficient input autocorrelation matrix," *Signal Processing*, vol. 89, pp. 2244–2250, Nov. 2009.

[109] T. Al-Naffouri and A. H. Sayed, "Transient analysis of data-normalized adaptive filters," *IEEE Trans. on Signal Processing*, vol. 51, no. 3, pp. 639–652, Mar. 2003.

[110] R. Mukundan, S. Ong, and P. A. Lee, "Image analysis by Tchebichef moments," *IEEE Trans. on Image Processing*, vol. 10, no. 9, pp. 1357–1364, Sept. 2001.

[111] G. Golub and C. Van Loan, "An analysis of the total least squares problem," *SIAM Journal on Numerical Analysis*, pp. 883–893, 1980.

[112] S. Van Huffel and J. Vandewalle, *The Total Least Squares Problem: Computational Aspects and Analysis.* Society for Industrial Mathematics, PA, 1991.

[113] R. DeGroat and E. Dowling, "Data least squares problem and channel equalization." *IEEE Trans. Signal Processing*, vol. 41, no. 1, pp. 407–411, Jan. 1993.

[114] P. Regalia, "An unbiased equation error identifier and reduced-order approximations," *IEEE Trans. Signal Processing*, vol. 42, no. 6, pp. 1397–1412, June 1994.

[115] K. C. Ho and Y. T. Chan, "Bias removal in equation-error adaptive IIR filters," *IEEE Trans. Signal Processing*, vol. 43, no. 1, pp. 51–62, Jan. 1995.

[116] H. Shin and W. Song, "Bias-free adaptive IIR filtering," *IEICE Trans. on Fund. of Elect., Comm. and Compt. Sciences*, vol. 84, pp. 1273–1279, May 2001.

[117] D. Feng, Z. Bao, and L. Jiao, "Total least mean squares algorithm," *IEEE Trans. on Signal Processing*, vol. 46, pp. 2122–2130, 1998.

[118] A. Bertrand and M. Moonen, "Low-complexity distributed total least squares estimation in ad hoc sensor networks," *IEEE Trans. Signal Processing*, vol. 60, no. 8, pp. 4321–4333, Aug. 2012.

[119] K. S. Miller, "On the inverse of the sum of matrices," *Mathematics Magazine*, vol. 54, no. 2, pp. 67–72, 1981.

[120] S. Boyd and L. Vandenberghe, *Convex optimization.* Cambridge University Press, 2004.

[121] W. X. Zheng, "A least-squares based algorithm for fir filtering with noisy data," in *Proc. of the Int. Symp. on Circuits and Systems (ISCAS)*, vol. 4, Bangkok, Thailand, May 2003, pp. IV444–447.

[122] L. Jia, R. Tao, Y. Wang, and K. Wada, "Forward/backward prediction solution for adaptive noisy FIR filtering," *Sci. China: Inf. Sciences*, vol. 52, no. 6, pp. 1007–1014, 2009.

[123] L. Li, J. A. Chambers, C. G. Lopes, and A. H. Sayed, "Distributed estimation over an adaptive incremental network based on the affine projection algorithm," *IEEE Trans. on Signal Processing*, vol. 58, no. 1, pp. 151–164, Jan. 2010.

[124] S. Y. Tu and A. H. Sayed, "Diffusion strategies outperform consensus strategies for distributed estimation over adaptive networks," *IEEE Trans. on Signal Processing*, vol. 60, no. 12, pp. 6217–6234, Dec. 2012.

[125] J. G. Proakis, *Digital Communications.* 4nd edition, McGraw-Hill, New York, 2000.

[126] A. Leon-Garcia, *Probability and Random Processes for Electrical Engineering.* 2nd edition, Addison-Wesley, Reading, MA, 1994.

[127] G. Mateos, I. D. Schizas, and G. B. Giannakis, "Performance analysis of the consensus-based distributed LMS algorithm," *EURASIP J. Adv. Signal Process.*, pp. 1–19, Nov. 2009.

[128] S. Kar and J. M. F. Moura, "Distributed consensus algorithms in sensor networks: Link failures and channel noise," *IEEE Trans. on Signal Processing*, vol. 57, no. 1, pp. 355–369, Jan. 2009.

[129] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory.* Prentice-Hall, 1993.

[130] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *J. of Chemical Physics*, vol. 21, no. 6, pp. 1087–1092, June 1953.

[131] M. K. Jain, *Numerical Methods for Scientific and Engineering Computation.* New Age International, 2003.

[132] J. Proakis and M. Salehi, *Digital Communications.* McGraw-Hill, 1995.

[133] A. Ben-Israel and T. N. Greville, *Generalized Inverses: Theory and Applications.* Springer, 2003.