

# Formulation and Analysis of LMS Adaptive Networks for Distributed Estimation in the Presence of Transmission Errors

Saeed Ghazanfari-Rad, and Fabrice Labeau, *Senior Member, IEEE*

**Abstract**—Wireless sensor network (WSN) technologies and distributed processing are essential to develop ubiquitous sensing in the Internet of Things (IoT) paradigm, wherein sensors pervasively collect data and perform information processing and communication tasks to achieve a common objective. This article presents the formulation and analysis of distributed estimation algorithms based on the diffusion cooperation scheme in the presence of errors due to the unreliable data transfer among nodes. In particular, we highlight the impact of transmission errors on the least-mean squares (LMS) adaptive networks. We develop the closed-form expressions of the steady-state mean-square deviation (MSD) which is helpful to assess the effects of the imperfect information flow on the behavior of diffusion LMS algorithms in terms of the steady-state error. The model is then validated by performing Monte Carlo simulations. It is shown that local and global steady-state MSD values are not necessarily monotonic increasing functions of the error probability. We also assess sufficient conditions that ensure mean and mean-square stability of diffusion LMS strategies in the presence of transmission errors. We examine a practical scenario where errors occur at the medium access control (MAC) level. To overcome the problem of unreliable data exchange, we implement a random pairwise strategy that improves the performance of the estimation algorithm in the presence of high transmission error rates. Moreover, issues such as scalability in the sense of network size and regressor size, convergence rate during the transient phase, spatially correlated observations, as well as the effect of the distribution of the noise variance are studied.

**Index Terms**—Adaptive networks, diffusion LMS algorithm, MAC layer, spatially correlated observations, spatially correlated regressors, distributed estimation, distributed signal processing.

## I. INTRODUCTION

INTERNET of Things (IoT) is envisioned to seamlessly connect billions of sensors and actuator devices in order to create a smart world [1], [2]. Wireless sensor networks (WSNs) are indispensable to the IoT paradigm and their deployment as monitoring and diagnostic systems is receiving significant attention in recent years because of their clear

This work was supported by Hydro-Quebec, the Natural Sciences and Engineering Research Council of Canada and McGill University in the framework of the NSERC/Hydro-Québec/McGill Industrial Research Chair in Interactive Information Infrastructure for the Power Grid.

A preliminary version of this work considering a two-node network has appeared as a conference paper in the *Proceedings of Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, November 2012*.

The authors are with the Department of Electrical and Computer Engineering, McGill University, Montreal, QC H3A 0E9, Canada (e-mail: saeed.ghazanfarirad@mail.mcgill.ca; fabrice.labeau@mcgill.ca).

Copyright (c) 2012 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

advantage of being cost-effective and easy to deploy compared to the traditional wired-based ones. Furthermore, through the implementation of WSNs operating in a collaborative mode, a wide variety of applications foreseen in the IoT, such as smart grid, precision agriculture, intelligent transportation systems, disaster relief management, radar and target detection and tracking, robot networks, vehicular networks, smart cities, etc, would benefit from highly reliable and flexible monitoring and diagnostic systems that rapidly respond to the changing conditions as well as instantaneous faults. Distributed implementation of estimation algorithms are recently forming an active area of research in the context of distributed adaptive filtering. In the conventional estimation systems, the nodes collect measurements and send them to a fusion center for final centralized processing. The central node would also broadcast the information to the individual nodes. Hence, the sensors achieve an estimate that is as accurate as the one that would be obtained if each sensor had access to all the information available across the network. However, such a traditional scheme has the disadvantages of a high communication overhead and power consumption. Furthermore, the centralized implementation is not scalable in terms of the communication bandwidth and computational complexity and lacks robustness in terms of the link failures.

A different approach for information exchange is the distributed in-network processing algorithms. In distributed processing, each node collects noisy observations related to a certain parameter or phenomenon of interest. The nodes would then communicate with their neighbors rather than a fusion center in order to arrive at an estimate of the parameter of interest. Distributed signal processing leads to significant saving in terms of bandwidth and power resources by reducing the communication overhead and the processing load [3]–[6]. Based on the topology of the network, different distributed adaptive algorithms can be implemented [7]–[11]. Several distributed consensus-based algorithms have been proposed in [12]–[14]. Throughout this paper, we study the diffusion mode of cooperation in which each node communicates with all its immediately adjacent neighbors according to the network topology [9], [10], [15], [16]. Due to space constraints, we report here the results obtained for combine-then-adapt (CTA) diffusion algorithm. It is straightforward to extend the analysis to the adapt-then-combine (ATC) strategy. The CTA implementation consists of two steps: first, the local estimate and the ones obtained from the neighbors are linearly combined and in the second step the adaptation is performed. The CTA

algorithm has been first proposed in [9], [17]–[19] and later its modified versions appeared in [11], [20], [21].

There exists previous literature for performance analysis of diffusion algorithms, but most of these studies consider an ideal and error-free transmissions [9]–[11]. There are useful studies that examine the performance of distributed consensus algorithms over networks with imperfect communications [22]–[24]. Some studies have already considered the diffusion algorithms with noisy information exchanges [25], [26]. These studies introduce an additive noise component to model the noisy link in different steps of the diffusion algorithms. The analytical framework that we propose differs from the literature in the way that we model imperfect information exchange. In particular, we incorporate the discrete-event random errors into the formulation which enables the analysis to account for the random transmission errors arising from several factors such as collisions, node failure, link failure and link congestion. To resolve the problem of random link failures, recently, a useful distributed consensus-based algorithm which uses a power control strategy to estimate and control the connectivity of ad-hoc networks was introduced in [27]. Reference [28] proposes algorithms to overcome the problem of channel fading and noise when using diffusion LMS algorithms. The performance comparison and conclusions are indicative of some trends, but are based on simulations. Different from [28], we analytically investigate the performance of diffusion LMS algorithms in the presence of transmission errors. Furthermore, it is assumed in [28] that each agent always receives the information but it includes some distortion and noise component. The way we model such errors is different from [28] as in our problem formulation it is possible that a transmission fails at any iteration. The way we model transmission errors is important and significantly more realistic in several applications such as large-scale and very large-scale dense networks (e.g., IoT systems, biological networks, social networks) where link congestions or other sources of errors prevent successful delivery of information.

Imperfect communications and channel noise were studied in [29] where the authors use the model that we developed and initially published in [30] to represent the link failures along with the additive noise model. Even though the results of [29] are useful, the following shortcomings have to be stressed. The authors of [29] do not address the mean-square stability analysis which is of considerable importance in the context of adaptive filters from theoretical and practical perspectives. One of the contributions of our paper is to provide conditions on step-sizes to ensure stability in the mean-square sense when transmission errors exist. Furthermore, reference [29] does not study how to model and/or quantify the transmission errors in a real network. Evidenced from the simulations, the discussions and results are limited to identical and time-invariant error probabilities ( $p = 0.1$ ) which are not realistic assumptions. Based on such simulations over a very small network with 8 nodes, it is concluded that additive noise components are the main factor in performance degradation of diffusion LMS algorithms. We will conclude that such results are limited to specific experiments and can not be extended to general realistic scenarios and particularly dense networks. Unlike

[29], we describe different sources of errors and we consider practical MAC level transmissions to accurately quantify such errors. We study realistic experiments where the probabilities of failures are non-identical over the links and they might be time variant as well. Furthermore, we show that in a dense network, the errors arising from congested links can seriously degrade the performance of diffusion LMS algorithms. Additionally, reference [29] does not provide a solution to mitigate the problem of transmission errors and noisy links. Finally, the assumption of independent regressors is used in [29], an assumption that is relaxed in our work. It is worth noticing that recently, diffusion strategies were proposed for adaptive estimation over networks where spatial correlation among noise signals is exploited according to a Gaussian Markov random field (GMRF) model [31]. But, it is assumed in [31] that the regressors are spatially independent (see Assumption 2 (independent regressors) of [31]) which is too strong of an assumption in realistic applications. Different from [29] and [31], we remove the independence assumption and in the sequel we examine the spatially correlated regressors.

When dealing with practical and error-prone transmissions, a careful modeling and performance analysis is required to highlight the effects of transmission errors on the behavior of the diffusion algorithms. Since adaptive filters are inherently nonlinear time-varying systems, often theoretical development of a single stand-alone LMS filter is a difficult task and involves a number of assumptions on the observed data. This mathematical hurdle would be more challenging when dealing with the coupling effects arising from the diffusion algorithm and the inherently time-varying transmission errors.

In detail, the contributions of this paper are as follows: 1) The main contribution of our research is to formulate and analyze the performance of LMS adaptive networks for distributed estimation considering transmission errors. We already examined a two-node network [30] with regressor vectors of size  $M \times 1$  which can be considered as a special case of this article that focuses on a network including  $N$  nodes. 2) The mathematical foundation proposed in this article brings the advantage of avoiding the procedure of inverting a matrix of size  $N^2 M^2 \times N^2 M^2$  (as required in [8]–[11]) which might lead to computational problems for large  $N$  and/or  $M$ , for instance, in large-scale networks and future IoT systems. 3) We assess sufficient conditions that ensure mean and mean-square stability of diffusion LMS strategies in the presence of transmission errors. 4) Our proposed analytical framework incorporates spatially correlated observations. Specifically and different from [8]–[11], we do not impose the constraint of spatially independent regressors across the distributed nodes. 5) We investigate the medium access control (MAC) level errors for a practical scenario, wherein a set of nodes employing a backoff procedure, compete with each other to access the channel. We quantify such errors and study their impacts on the performance of diffusion LMS algorithms. 6) We implement a random pairwise strategy that improves the performance of the diffusion estimation algorithm in the presence of high transmission error rates, meanwhile it provides savings in terms of energy and bandwidth by reducing the amount of data exchange among nodes.

The remainder of the paper is organized as follows. In Section II, we formulate the problem of diffusion algorithms over distributed adaptive networks in the presence of transmission errors. In this section, we provide the mean analysis, mean-square analysis and closed-form derivation of the steady-state MSD. We also derive sufficient conditions to ensure the stability of the algorithm in the mean and mean-square sense. In Section III, We quantify transmission errors by studying a case scenario where errors occur at the MAC layer. In order to verify the accuracy of the proposed theoretical framework and observe the impact of transmission errors on the distributed estimation algorithms, we present the simulation results in Section IV. Finally, we conclude the paper in Section V.

## II. ANALYSIS OF DIFFUSION LMS OVER A NETWORK INCLUDING $N$ NODES

### A. Problem Formulation

Consider a distributed network with a set of nodes  $\mathcal{N} = \{1, 2, \dots, N\}$  and a predefined topology including  $L$  links. In a typical wireless deployment, a link  $(k, \ell)$ ,  $k, \ell \in \mathcal{N}$  exists between two nodes if and only if the physical distance between the end nodes is less than the maximum radio transmission range. Let  $\mathcal{L} = \{(k, \ell), k, \ell \in \mathcal{N}\}$  denote the set of all links of the network. Let  $\mathcal{N}_k$  denote the set of nodes in the neighborhood of node  $k$  (i.e., those with which node  $k$  has a link) including node  $k$  itself. The objective of the network is to estimate the unknown parameter vector  $\mathbf{w}^o$  in a distributed manner from measurements of  $N$  nodes. In a diffusion strategy every node  $k$  at each time  $i \geq 0$  employs some mixing coefficients to combine estimates from its neighborhood  $\mathcal{N}_k$  [9]. However, in practice not all of the linked nodes are able to send their estimates to node  $k$  due to the errors at different layers of the communication protocol stack. Consequently, unlike the ideal situation in which no error occurs, each iteration of the adaptive algorithm builds up a different set of mixing coefficients that depends on the error probabilities.

Let  $\mathcal{S}_{k,i} \subseteq \mathcal{N}$  denote the set of neighbors of node  $k$  that successfully transmit their information to node  $k$  at time  $i$  including node  $k$ . We introduce the adaptive filter of node  $k$  in a distributed network based on diffusion LMS with transmission errors as follows:

$$\mathbf{w}_{k,i+1} = \phi_{k,i} + \mu_k e_{k,i} \mathbf{u}_{k,i}, \quad i \geq 0, \quad (1)$$

where

$$\phi_{k,i} = \sum_{\ell \in \mathcal{S}_{k,i}} a_{k,\ell,i} \mathbf{w}_{\ell,i}, \quad (2)$$

$$e_{k,i} = d_{k,i} - \phi_{k,i}^T \mathbf{u}_{k,i}, \quad (3)$$

$$d_{k,i} = v_{k,i} + \mathbf{w}^{oT} \mathbf{u}_{k,i}. \quad (4)$$

In the above expressions,  $\mathbf{w}_{k,i}$  (for  $k = 1, 2, \dots, N$ , and  $i \geq 0$ ) are the  $M$ -dimensional coefficient weight vectors of the adaptive algorithm at node  $k$  and iteration  $i$ ,  $\mu_k$  is the step size at node  $k$ ,  $a_{k,\ell,i}$ ,  $\ell = 1, 2, \dots, N$  are the mixing coefficients at iteration  $i$ ,  $\phi_{k,i}$  are the intermediate variables to obtain the new weight vectors after information exchange among nodes,

$\mathbf{u}_{k,i}$  are the  $M$ -dimensional input vectors,  $e_{k,i}$  are the error signals,  $v_{k,i}$  are the noise signals and  $d_{k,i}$  are the desired signals obtained from the unknown weight vector  $\mathbf{w}^o$  through the linear regression model (4). The superscript  $(\cdot)^T$  denotes transpose operation.

Throughout this article we assume that a failure occurs in information flow from node  $\ell$  to node  $k$  at each iteration  $i$  with probability  $p_{k,\ell}$ . The difficulties that cause such failures or transmission errors may include, but are not limited to: mobility of nodes and time-varying network topology, interference and multipath fading, signal attenuation at the physical (PHY) layer, packet loss at the MAC layer and attacks originated by attackers or intruders. Thus, it is reasonable to assume that in general such errors are independent non-identically distributed. However, our analysis is not restricted to independent errors assumption. For the sake of mathematical tractability, we assume that the error probabilities are time-invariant which is reasonable in stationary or slowly time-varying practical scenarios. However, we verify in Section IV that the proposed analysis is valid even if the error probabilities are time-variant. Furthermore it is reasonable to assume that transmission errors that occur with probabilities  $p_{k,\ell}$  and the measurement errors  $v_{k,i}$  are independent. As an example, a node may observe a low measurement noise variance (for example if it is close to a target in tracking applications) but still experience high transmission error due to the high density of nodes in that area.

We notice that unlike the previous formulation (ideal transmission) [9]–[11], here, the intermediate variable  $\phi_{k,i}$  is constructed by a weighted sum over the set of nodes who have successfully transmitted their local information to node  $k$ . To further demonstrate the new formulation, a different way to express  $\phi_{k,i}$  is described in Appendix A. In order to incorporate the transmission errors in the formulation of diffusion algorithm, we subsequently define some useful notation. Define  $\Lambda = [\lambda_{k\ell}], k, \ell \in \mathcal{N}$  as the  $N \times N$  adjacency matrix representing the network connectivity, i.e., each entry  $\lambda_{k\ell}$  is 1 if there is a link between nodes  $k$  and  $\ell$  and is 0 if there is not. Assume that  $E$  is the number of 1's not located on the main diagonal of  $\Lambda$ , i.e.,  $E = 2L$ . Motivated by the aforementioned discussion and assumptions regarding unreliable transmissions, we define a set of possible events  $\mathcal{E} = \{\varepsilon_1, \dots, \varepsilon_V\}$ , their corresponding probabilities  $\mathcal{P} = \{p_1, \dots, p_V\}$ , where  $V = 2^E$  and the set of combining matrices  $\mathcal{A} = \{\mathcal{A}^{(1)}, \dots, \mathcal{A}^{(v)}\}$ . We also introduce the set  $\mathcal{V} = \{1, \dots, V\}$  whose  $j^{\text{th}}$  element corresponds to the occurrence of event  $e_j, j = 1, \dots, V$ . We note that  $\mathcal{A}^{(j)} = [a_{k,\ell}^{(j)}], j = 1, \dots, V$  is the  $N \times N$  combining matrix that collects the nonnegative mixing coefficients of diffusion update followed by event  $\varepsilon_j$  during information exchange period satisfying

$$\sum_{\ell \in \mathcal{N}_k} a_{k,\ell}^{(j)} = 1, \quad \text{for all } k \in \mathcal{N} \text{ and } j \in \mathcal{V}. \quad (5)$$

At each time iteration  $i$ , we collect the mixing coefficients as  $\mathcal{A}_i = [a_{k,\ell,i}]$ . Followed by the occurrence of event  $\varepsilon_j$ , we have that  $\mathcal{A}_i = \mathcal{A}^{(j)}$ . It follows from the above diffusion algorithm that each entry  $a_{k,\ell}^{(j)}$  of matrix  $\mathcal{A}^{(j)}$  represents the weight

given to node  $\ell$  in order to find the intermediate variable at node  $k$  conditioned that event  $\varepsilon_j$  occurred during information sharing. As an example, suppose that  $\varepsilon_1$  represents the event in which all transmissions fail due to congested links. Under the independent errors assumption, the probability associated to this case is  $p_1 = \prod_{(k,\ell) \in \mathcal{L}} p_{k,\ell}$  and it follows that  $\mathcal{A}^{(1)} = \mathcal{I}_N$ , where  $\mathcal{I}_N$  is the  $N \times N$  identity matrix, i.e., each node establishes the update only according to its local observation.

Regarding the statistics of the measurement data and noise signals, we assume that the regressors  $\mathbf{u}_{k,i}$  are temporally independent identically distributed (i.i.d.) zero-mean white Gaussian random variables with covariance matrices  $\mathcal{R}_{u_k} = \mathbb{E}[\mathbf{u}_k \mathbf{u}_k^T] = \sigma_{u_k}^2 \mathcal{I}_M$ , where  $\mathbb{E}[\cdot]$  denotes the expected-value operator. However, we explore the spatial correlation between nodes by assuming that two locally observed vectors  $\mathbf{u}_k$  and  $\mathbf{u}_\ell$  are correlated Gaussian random vectors with cross-correlation matrix  $\mathcal{R}_{u_k, u_\ell} = \sigma_{u_k \ell}^2 \mathcal{I}_M$ , where  $\sigma_{u_k \ell}^2 = \rho_{k\ell} \sigma_{u_k} \sigma_{u_\ell}$  and  $\rho_{k\ell}$  is the spatial correlation index between nodes  $k$  and  $\ell$ . The noise signal  $v_{k,i}$  is zero mean i.i.d. white Gaussian random variable with variance  $\sigma_{v_k}^2$ . The input vectors  $\mathbf{u}_{k,i}$  and noise signals  $v_{k,i}$  are temporally and spatially independent of each other.

Note that (2) represents a linear combination of the received weight vectors to produce the intermediate variable  $\phi_{k,i}$  at node  $k$  at iteration  $i$ . In general, the combiners may be nonlinear or even time-variant to reflect variations in network topology or to respond efficiently to nonstationary conditions [9]. In the following discussion the mixing coefficients are considered to be time-varying in order to capture the effects of transmission errors. We use the above formulation throughout the forthcoming sections to work out the detailed mean and mean-squared analyses of diffusion estimation algorithms in the presence of transmission errors.

### B. Mean Analysis

In this subsection, we provide the mean analysis which will be used later in order to find the expression for the steady-state mean-square deviation (MSD). Let  $\varepsilon_j(i)$  represent transmission errors during iteration  $i$  which corresponds to  $\varepsilon_j$ . Using (1)-(4) and conditioned that transmission errors during iteration  $i$  correspond to  $\varepsilon_j$ , we can obtain a recursive expression to calculate  $\mathbb{E}[\mathbf{w}_{k,i} | \varepsilon_j(i)]$  as follows

$$\mathbb{E}[\mathbf{w}_{k,i+1} | \varepsilon_j(i)] = a_{k,k}^{(j)} \rho_k \mathbb{E}[\mathbf{w}_{k,i}] + \rho_k \sum_{\substack{\ell \in \mathcal{N} \\ \ell \neq k}} a_{k,\ell}^{(j)} \mathbb{E}[\mathbf{w}_{\ell,i}] + \mathbf{c}_k, \quad (6)$$

where

$$\rho_k = 1 - \mu_k \sigma_{u_k}^2, \quad k \in \mathcal{N}, \quad (7)$$

$$\mathbf{c}_k = \mu_k \sigma_{u_k}^2 \mathbf{w}^o, \quad k \in \mathcal{N}. \quad (8)$$

In order to find  $\mathbb{E}[\mathbf{w}_{k,i+1}]$ , we consider all possibilities according to set  $\mathcal{E}$  and replace (6) in the following equation

$$\mathbb{E}[\mathbf{w}_{k,i+1}] = \sum_{j \in \mathcal{V}} p_j \mathbb{E}[\mathbf{w}_{k,i+1} | \varepsilon_j(i)], \quad (9)$$

which yields

$$\mathbb{E}[\mathbf{w}_{k,i+1}] = a_{k,k} \rho_k \mathbb{E}[\mathbf{w}_{k,i}] + \rho_k \sum_{\substack{\ell \in \mathcal{N} \\ \ell \neq k}} a_{k,\ell} \mathbb{E}[\mathbf{w}_{\ell,i}] + \mathbf{c}_k, \quad (10)$$

where

$$a_{k,k} = \sum_{j \in \mathcal{V}} p_j a_{k,k}^{(j)}, \quad a_{k,\ell} = \sum_{j \in \mathcal{V}} p_j a_{k,\ell}^{(j)}, \quad (11)$$

for all  $k, \ell \in \mathcal{N}, \ell \neq k$ . It follows from (5) that

$$\sum_{\ell \in \mathcal{N}_k} a_{k,\ell} = 1, \quad k \in \mathcal{N}. \quad (12)$$

From Appendix B, we conclude that

$$\mathbb{E}[\mathbf{w}_{k,s}] = \mathbf{w}^o, \quad k \in \mathcal{N}, \quad (13)$$

i.e., the weights converge to the optimal value.

### C. Mean Stability

In the solution procedure of the previous subsection and in particular in using the Cramer's law in Appendix B, we should verify that  $z = 1$  is not a root of the polynomial of order  $N$  obtained from  $\det(\mathcal{E}_n) = 0$ . This polynomial can be written as  $F(z) = \sum_{k=0}^N f_k z^{-k}$ . Moreover, for stability in mean, it is required that all roots of  $F(z)$  lie within the unit circle. Using the structure of  $\mathcal{E}_n$  defined in (61), it is easily verified that  $F(z)$  is the characteristic polynomial of  $\mathcal{E}'_{n,s} = [\rho_i a_{i,j}]_{N \times N}$ ,  $i, j = 1, 2, \dots, N$  and it is immediate that the roots of  $F(z)$  are the eigenvalues of the square matrix  $\mathcal{E}'_{n,s}$ . Let  $\lambda_k$ ,  $k = 1, 2, \dots, N$  denote the eigenvalues of  $\mathcal{E}'_{n,s}$ . We also use  $\rho(\mathcal{E}'_{n,s})$  to denote the spectral radius of  $\mathcal{E}'_{n,s}$ . In the sequel, we find the sufficient condition that guarantees that the maximum absolute eigenvalue  $\max_{1 \leq k \leq N} |\lambda_k|$  or equivalently the spectral radius  $\rho(\mathcal{E}'_{n,s})$  is less than one. This condition is sufficient to place the roots of  $F(z)$  within the unit circle and hence ensure stability in mean. Considering (12), we notice that one interesting feature of the rows of  $\mathcal{E}'_{n,s}$  is that

$$\sum_{\ell \in \mathcal{N}} \rho_k a_{k,\ell} = \rho_k, \quad k = 1, 2, \dots, N. \quad (14)$$

Consider the induced infinity-norm of matrix  $\mathcal{E}'_{n,s}$  defined as

$$\|\mathcal{E}'_{n,s}\|_\infty = \max_{1 \leq k \leq N} \sum_{\ell \in \mathcal{N}} |\rho_k a_{k,\ell}|. \quad (15)$$

It is also known from the characteristics of a matrix norm that

$$\rho(\mathcal{E}'_{n,s}) \leq \|\mathcal{E}'_{n,s}\|. \quad (16)$$

To satisfy  $\max_{1 \leq k \leq N} |\lambda_k| < 1$ , we use (14)-(16) to write

$$|\rho_k| < 1, \quad k = 1, 2, \dots, N, \quad (17)$$

which is equivalent to impose the following lower and upper bounds on the step-sizes

$$0 < \mu_k < \frac{2}{\sigma_{u_k}^2}, \quad k = 1, 2, \dots, N. \quad (18)$$

We conclude that in a network including  $N$  nodes deploying distributed diffusion estimation algorithm (1)-(4) with combining weight matrices satisfying (5), in the presence of multiplicative transmission errors modeled as (57), the sufficient condition for mean stability is provided by (18). It is important to note that (18) coincides with sufficient condition for mean stability when transmissions are assumed to be perfect [11]. In other words, if step sizes satisfy (18), then transmission errors described in Section II-A and modeled according to a Bernoulli process (57) do not diverge diffusion estimation algorithm (1)-(4) in the mean sense.

#### D. Mean-Square Analysis

We aim at finding the closed form expressions for the steady-state MSD. Notice that the steady state MSD value for weight vector at node  $k$  is defined as follows:

$$\text{MSD}_k = \lim_{i \rightarrow \infty} \mathbb{E}[(\mathbf{w}_{k,i} - \mathbf{w}^o)^\top (\mathbf{w}_{k,i} - \mathbf{w}^o)]. \quad (19)$$

It is shown in Appendix C that we can write the following expression for  $\mathbb{E}[\mathbf{w}_{k,i+1}^\top \mathbf{w}_{\ell,i+1} | \varepsilon_j(i)]$ :

$$\begin{aligned} & \mathbb{E}[\mathbf{w}_{k,i+1}^\top \mathbf{w}_{\ell,i+1} | \varepsilon_j(i)] \\ &= \eta_{k\ell} \left\{ \sum_{m \in \mathcal{N}} a_{k,m}^{(j)} a_{\ell,m}^{(j)} \mathbb{E}[\mathbf{w}_{m,i}^\top \mathbf{w}_{m,i}] \right. \\ & \quad + \sum_{m,n \in \mathcal{N}, m \neq n} (a_{k,m}^{(j)} a_{\ell,n}^{(j)} + a_{k,n}^{(j)} a_{\ell,m}^{(j)}) \mathbb{E}[\mathbf{w}_{m,i}^\top \mathbf{w}_{n,i}] \left. \right\} \\ & \quad + \sum_{m \in \mathcal{N}} (a_{k,m}^{(j)} (\epsilon_\ell - \nu_{k\ell}) + a_{\ell,m}^{(j)} (\epsilon_k - \nu_{k\ell})) \mathbf{w}^{o\top} \mathbb{E}[\mathbf{w}_{m,i}] \\ & \quad + \nu_{k\ell} \mathbf{w}^{o\top} \mathbf{w}^o, \end{aligned} \quad (20)$$

where

$$\eta_{k\ell} = 1 - (\mu_k \sigma_{u_k}^2 + \mu_\ell \sigma_{u_\ell}^2) + \mu_k \mu_\ell (\sigma_{u_k}^2 \sigma_{u_\ell}^2 + (M+1) \sigma_{u_{k\ell}}^4), \quad (21)$$

$$\epsilon_k = \mu_k \sigma_{u_k}^2, \quad (22)$$

$$\nu_{k\ell} = \mu_k \mu_\ell (\sigma_{u_k}^2 \sigma_{u_\ell}^2 + (M+1) \sigma_{u_{k\ell}}^4). \quad (23)$$

In order to consider the set of all possible events during information exchange period, we write:

$$\mathbb{E}[\mathbf{w}_{k,i+1}^\top \mathbf{w}_{\ell,i+1}] = \sum_{j \in \mathcal{V}} p_j \mathbb{E}[\mathbf{w}_{k,i+1}^\top \mathbf{w}_{\ell,i+1} | \varepsilon_j(i)]. \quad (24)$$

Using (20) and (24), it follows that:

$$\begin{aligned} & \mathbb{E}[\mathbf{w}_{k,i+1}^\top \mathbf{w}_{\ell,i+1}] \\ &= \sum_{m \in \mathcal{N}} c_{k\ell,mm} \mathbb{E}[\mathbf{w}_{m,i}^\top \mathbf{w}_{m,i}] + \sum_{\substack{m,n \in \mathcal{N} \\ m \neq n}} c_{k\ell,mn} \mathbb{E}[\mathbf{w}_{m,i}^\top \mathbf{w}_{n,i}] \\ & \quad + \sum_{m \in \mathcal{N}} c_{k\ell,om} \mathbf{w}^{o\top} \mathbb{E}[\mathbf{w}_{m,i}] + \nu_{k\ell} \mathbf{w}^{o\top} \mathbf{w}^o, \end{aligned} \quad (25)$$

where

$$c_{k\ell,mm} = \eta_{k\ell} \sum_{j \in \mathcal{V}} p_j a_{k,m}^{(j)} a_{\ell,m}^{(j)}, \quad (26)$$

$$c_{k\ell,mn} = \eta_{k\ell} \sum_{j \in \mathcal{V}} p_j (a_{k,m}^{(j)} a_{\ell,n}^{(j)} + a_{k,n}^{(j)} a_{\ell,m}^{(j)}), \quad (27)$$

$$c_{k\ell,om} = \sum_{j \in \mathcal{V}} p_j [a_{k,m}^{(j)} (\epsilon_\ell - \nu_{k\ell}) + a_{\ell,m}^{(j)} (\epsilon_k - \nu_{k\ell})], \quad (28)$$

for all  $k, \ell, m, n \in \mathcal{N}$ ,  $k \neq \ell$  and  $m \neq n$ . In a similar way, we can write:

$$\begin{aligned} & \mathbb{E}[\mathbf{w}_{k,i+1}^\top \mathbf{w}_{k,i+1}] \\ &= \sum_{m \in \mathcal{N}} c_{kk,mm} \mathbb{E}[\mathbf{w}_{m,i}^\top \mathbf{w}_{m,i}] + \sum_{\substack{m,n \in \mathcal{N} \\ m \neq n}} c_{kk,mn} \mathbb{E}[\mathbf{w}_{m,i}^\top \mathbf{w}_{n,i}] \\ & \quad + \sum_{m \in \mathcal{N}} c_{kk,om} \mathbf{w}^{o\top} \mathbb{E}[\mathbf{w}_{m,i}] + \nu_k \mathbf{w}^{o\top} \mathbf{w}^o + M \mu_k^2 \sigma_{u_k}^2 \sigma_{v_k}^2, \end{aligned} \quad (29)$$

where

$$c_{kk,mm} = \eta_k \sum_{j \in \mathcal{V}} p_j a_{k,m}^{(j)2}, \quad (30)$$

$$c_{kk,mn} = 2\eta_k \sum_{j \in \mathcal{V}} p_j a_{k,m}^{(j)} a_{k,n}^{(j)}, \quad m \neq n, \quad (31)$$

$$c_{kk,om} = 2 \sum_{j \in \mathcal{V}} p_j a_{k,m}^{(j)} (\epsilon_k - \nu_k), \quad (32)$$

$$c_{v_k} = M \mu_k^2 \sigma_{u_k}^2 \sigma_{v_k}^2, \quad (33)$$

and:

$$\eta_k = 1 - 2\mu_k \sigma_{u_k}^2 + \mu_k^2 (M+2) \sigma_{u_k}^4, \quad (34)$$

$$\nu_k = \mu_k^2 (M+2) \sigma_{u_k}^4. \quad (35)$$

Let  $\mathcal{W}_{k\ell}$  denote the one-sided  $z$ -transform of  $\mathbb{E}[\mathbf{w}_{k,i}^\top \mathbf{w}_{\ell,i}]$ . Taking the  $z$ -transform of both sides of (25) and (29) and after some algebra we obtain

$$\begin{aligned} & (1 - z^{-1} c_{kk,kk}) \mathcal{W}_{kk} - z^{-1} \sum_{m \in \mathcal{N}, m \neq k} c_{kk,mm} \mathcal{W}_{mm} \\ & \quad - z^{-1} \sum_{m \in \mathcal{N}, m \neq k} c_{kk,mn} \mathcal{W}_{mm} \\ &= z^{-1} \sum_{m \in \mathcal{N}} c_{kk,om} \mathbf{w}^{o\top} \mathcal{W}_m + \frac{\nu_k \mathbf{w}^{o\top} \mathbf{w}^o + c_{v_k}}{1 - z^{-1}}, \end{aligned} \quad (36)$$

and

$$\begin{aligned} & (1 - z^{-1} c_{k\ell,k\ell}) \mathcal{W}_{k\ell} - z^{-1} \sum_{m \in \mathcal{N}} c_{k\ell,mm} \mathcal{W}_{mm} \\ & \quad - z^{-1} \sum_{\substack{m,n \in \mathcal{N}, m \neq n \\ m \neq k, n \neq \ell}} c_{k\ell,mn} \mathcal{W}_{mm} \\ &= z^{-1} \sum_{m \in \mathcal{N}} c_{k\ell,om} \mathbf{w}^{o\top} \mathcal{W}_m + \frac{\nu_{k\ell} \mathbf{w}^{o\top} \mathbf{w}^o}{1 - z^{-1}}. \end{aligned} \quad (37)$$

Note that (36) and (37) are the expressions at one-node level and two-node level that completely describe the coupling

effects among different nodes in a diffusion estimation algorithm. The total number of equations in a network including  $N$  nodes would then be  $Q = N(N+1)/2$ . In order to build the equations in a compact form, we consider writing all one-node level equations followed by those representing two-node level. Thus, ensuring that the permutation of the set of equations is selected as  $\{11, \dots, NN, 12, \dots, 1N, 23, \dots, 2N, \dots, N-2, N-1, N-2, N, N-1, N\}$ , the system description in  $z$ -domain can be represented as (38) shown at the bottom of the page.

Denoting the  $i^{th}$  column of  $\mathcal{C}$  by  $\mathcal{C}_i$ ,  $i = 1, 2, \dots, Q$ , we write

$$[\mathcal{C}_1 \ \mathcal{C}_2 \ \dots \ \mathcal{C}_Q]_{Q \times Q} [\mathcal{W}]_{Q \times 1} = [\mathcal{D}]_{Q \times 1}. \quad (39)$$

Recall that our objective is to find  $\mathcal{W}_{kk}$ ,  $k = 1, 2, \dots, N$ , i.e., the first  $N$  elements of  $\mathcal{W}$  from the set of equations described in (38). Let  $\mathcal{C}_{\mathcal{D}i}$ ,  $i = 1, 2, \dots, N$  denote the matrix obtained after replacing the  $i^{th}$  column of  $\mathcal{C}$  by  $\mathcal{D}$ . Then, using Cramer's rule we obtain  $\mathcal{W}_{kk}$  as follows:

$$\mathcal{W}_{kk} = \frac{\det(\mathcal{C}_{\mathcal{D}k})}{\det(\mathcal{C})}, \quad k = 1, 2, \dots, N, \quad (40)$$

where  $\det(\cdot)$  denotes the determinant operator.

We can rewrite matrix  $\mathcal{D}$  as follows:

$$\mathcal{D} = \underbrace{\begin{bmatrix} z^{-1} \sum_{m \in \mathcal{N}} c_{11,om} \mathbf{w}^o \mathcal{W}_m + \frac{\nu_1 \mathbf{w}^o \mathcal{W}_m}{1-z^{-1}} \\ z^{-1} \sum_{m \in \mathcal{N}} c_{22,om} \mathbf{w}^o \mathcal{W}_m + \frac{\nu_2 \mathbf{w}^o \mathcal{W}_m}{1-z^{-1}} \\ \vdots \\ z^{-1} \sum_{m \in \mathcal{N}} c_{N-1, N, om} \mathbf{w}^o \mathcal{W}_m + \frac{\nu_{N-1} \mathbf{w}^o \mathcal{W}_m}{1-z^{-1}} \end{bmatrix}}_{[\mathcal{D}_{w^o}]} + \underbrace{\begin{bmatrix} \frac{c_{v1}}{1-z^{-1}} \\ \frac{c_{v2}}{1-z^{-1}} \\ \vdots \\ 0 \end{bmatrix}}_{[\mathcal{D}_v]}. \quad (41)$$

It follows that

$$\det(\mathcal{C}_{\mathcal{D}i}) = \det(\mathcal{C}_1 \ \dots \ \mathcal{D}_{w^o} \ \mathcal{C}_Q) + \det(\mathcal{C}_1 \ \dots \ \mathcal{D}_v \ \mathcal{C}_Q). \quad (42)$$

Using (42) in (40), we obtain the following expression

$$\mathcal{W}_{kk} = \frac{\det(\mathcal{C}_1 \ \dots \ \mathcal{D}_{w^o} \ \mathcal{C}_Q)}{\det(\mathcal{C})} + \frac{\det(\mathcal{C}_1 \ \dots \ \mathcal{D}_v \ \mathcal{C}_Q)}{\det(\mathcal{C})}, \quad (43)$$

for all  $k \in \mathcal{N}$ . To proceed, we define the square matrix  $\mathcal{C}'$  of size  $Q$  as  $\mathcal{C}' = [c_{k\ell, mn}]_{Q \times Q}$  with  $c_{mn}$  obtained from (26),(27),(30) and (31) and arranged as in  $\mathcal{C}$  in (38). To simplify, we define  $c_s$ ,  $c_{w^o, s}$  at the steady-state as follows, as well as  $c_{v_k, s}$ ,  $k = 1, 2, \dots, N$  according to (44) as shown at the bottom of the next page.

$$c_s = \lim_{z \rightarrow 1} \det(\mathcal{C}) = \det(\mathcal{I}_N - \mathcal{C}'), \quad (45)$$

$$c_{w^o, s} = \lim_{z \rightarrow 1} (z-1) \det(\mathcal{C}_1 \ \dots \ \mathcal{D}_{w^o} \ \mathcal{C}_Q) = \mathbf{w}^o \mathcal{W}^o \det(\mathcal{C}). \quad (46)$$

Notice that the derivation of (46) is given in Appendix D. Let  $w_{k,s}$  denote the expectation of the norm of the weight vector corresponding to the  $k^{th}$  node at the steady-state as expressed in (19). Then, using the final value theorem,  $\mathcal{W}_{kk}$  as described in (43) and the steady-state quantities as defined in (44) and (45), we arrive at the following result:

$$\begin{aligned} w_{k,s} &\triangleq \lim_{i \rightarrow \infty} \mathbb{E}[\mathbf{w}_k^T \mathbf{w}_k] \\ &= \lim_{z \rightarrow 1} (z-1) \mathcal{W}_{kk} = \mathbf{w}^o \mathcal{W}^o + \frac{c_{v_k, s}}{c_s}. \end{aligned} \quad (47)$$

Finally, the closed-form expression for local MSD $_k$  is given by

$$\text{MSD}_k = \frac{c_{v_k, s}}{c_s}. \quad (48)$$

We notice that the global network MSD is obtained by averaging over the local MSD's as follows:

$$\text{MSD} = \frac{1}{N} \sum_{k \in \mathcal{N}} \text{MSD}_k. \quad (49)$$

$$\underbrace{\begin{bmatrix} 1 - z^{-1} c_{11,11} & -z^{-1} c_{11,22} & \dots & -z^{-1} c_{11, N-1, N} \\ -z^{-1} c_{22,11} & 1 - z^{-1} c_{22,22} & \dots & -z^{-1} c_{22, N-1, N} \\ \vdots & \vdots & \ddots & \vdots \\ -z^{-1} c_{N-1, N, 11} & -z^{-1} c_{N-1, N, 22} & \dots & 1 - z^{-1} c_{N-1, N, N-1, N} \end{bmatrix}}_{[\mathcal{C}]_{Q \times Q}} \underbrace{\begin{bmatrix} \mathcal{W}_{11} \\ \mathcal{W}_{22} \\ \vdots \\ \mathcal{W}_{N-1, N} \end{bmatrix}}_{[\mathcal{W}]_{Q \times 1}} = \underbrace{\begin{bmatrix} z^{-1} \sum_{m \in \mathcal{N}} c_{11,om} \mathbf{w}^o \mathcal{W}_m + \frac{\nu_1 \mathbf{w}^o \mathcal{W}_m + c_{v1}}{1-z^{-1}} \\ z^{-1} \sum_{m \in \mathcal{N}} c_{22,om} \mathbf{w}^o \mathcal{W}_m + \frac{\nu_2 \mathbf{w}^o \mathcal{W}_m + c_{v2}}{1-z^{-1}} \\ \vdots \\ z^{-1} \sum_{m \in \mathcal{N}} c_{N-1, N, om} \mathbf{w}^o \mathcal{W}_m + \frac{\nu_{n-1} \mathbf{w}^o \mathcal{W}_m}{1-z^{-1}} \end{bmatrix}}_{[\mathcal{D}]_{Q \times 1}}. \quad (38)$$

### E. Mean-square Stability

In this subsection, we discuss the mean-square stability of diffusion LMS algorithms in the presence of transmission errors. In particular, our aim is to answer the following questions. How do transmission errors affect the convergence of diffusion algorithms in the mean-square sense? Is there an explicit sufficient condition to ensure mean-square stability? To address these important issues, we use the same approach as the one presented in Subsection II-C.

We start with arguing that in order to use Cramer's law in (38) around  $|z| \rightarrow 1$ , we must provide condition to prevent any root of  $\det(\mathcal{C}) = 0$  be placed on the unit circle. Furthermore, all of the corresponding roots must lie within the unit circle to guarantee stability in the mean-square sense. We note that these roots are the eigenvalues of  $\mathcal{C}'$ . For convenience, we rewrite matrix  $\mathcal{C}'$  as  $\mathcal{C}' = [\gamma_{i,j}]_{Q \times Q}$  and define  $\eta'_i, i = 1, 2, \dots, Q$  to denote the common factor ( $\eta_k$  in (34) and  $\eta_{k\ell}$  in (21)) of elements in the  $i^{\text{th}}$  row of  $\mathcal{C}'$ . Let us define  $\lambda'_k, k = 1, 2, \dots, Q$  to refer to the eigenvalues of  $\mathcal{C}'$ . Using (12), (26),(27),(30) and (31) and noting the structure of  $\mathcal{C}'$ , we find one important feature of the rows of  $\mathcal{C}'$

$$\sum_{j=1}^Q \gamma_{i,j} = \eta'_i, \quad i = 1, 2, \dots, Q. \quad (50)$$

Using this result and considering the induced infinity-norm of matrix  $\mathcal{C}'$  and the similar principles already discussed in Subsection II-C, we find out that to satisfy  $\max_{1 \leq k \leq Q} |\lambda'_k| < 1$ , it is sufficient to ensure that  $|\eta'_i| < 1, i = 1, 2, \dots, Q$ , or equivalently

$$|\eta_k| < 1, \quad k \in \mathcal{N}, \quad \text{and} \quad |\eta_{k\ell}| < 1, \quad k, \ell \in \mathcal{N}, \quad k \neq \ell. \quad (51)$$

It is worth mentioning that (34) suggests that  $\eta_k$  only depends on the local step size of the individual filter of a single node and the statistics of the filter input. On the other hand, from (21), it is evident that  $\eta_{k\ell}$  accounts for the interaction among node pairs and is a function of step sizes and the statistics of the inputs at two different nodes. The solution of (51) provides useful and practically applicable lower and upper bounds for step sizes which can be written as follows:

$$0 < \mu_k < \frac{2}{(M+2)\sigma_{u_k}^2}, \quad k = 1, 2, \dots, N. \quad (52)$$

The following important result is drawn from the above discussions. In a network with  $N$  nodes using distributed diffusion estimation algorithm (1)-(4) with combining weight matrices satisfying (5), in the presence of multiplicative

transmission errors modeled as (57), the sufficient condition provided by (52) ensures stability in the mean-square sense. An important feature of the condition (52) is that it is not dependent on error probabilities  $p_{k,\ell}$ . This suggests that if step sizes satisfy (52), then transmission errors as described in Section II-A and modeled according to a Bernoulli distribution (57) do not lead to the mean-square divergence of diffusion estimation algorithms (1)-(4).

It is worth mentioning that condition (52) represents a novel bound which can be used even in the case of perfect information exchange. Considering perfect transmissions, it has been argued that sufficiently small step-sizes that satisfy mean stability, will also ensure mean-square stability [11]. Sufficiently small step-sizes may however lead to slow convergence rates and thus high energy requirements at individual nodes. This is undesirable given that in WSNs energy is crucially scarce. Furthermore, in certain applications where minimizing the speed of convergence is more important than achieving a small steady-state error one should not select very small step-sizes during the transient time [32]. Another practical significance of (52) is that the condition is fully distributed, i.e., each node can locally select the step-size according to the estimate of second-order statistics of its regression data.

### III. MODELING TRANSMISSION ERRORS

As previously discussed, there exist various uncertainties in WSNs such as mobility of nodes and time-varying network topology, interference and multipath fading, signal attenuation at the PHY layer, packet loss at the medium access control (MAC), etc. Any of these uncertainties may result in transmission errors during information exchange. Thus, in order to accurately evaluate transmission errors in WSNs, it is required to consider all sources of errors depending on the type of application. Given the scope of this paper, we limit our discussion to the MAC-level errors as one example of how to quantify  $p_{k,\ell}$  throughout the rest of this section.

There is a vast literature on designing efficient channel access mechanisms for WSNs and due to page restrictions, we refer the reader to [33]–[35] for further details. For our purpose it is sufficient to concentrate on the exponential backoff procedure that has been standardized as the basic access mechanism in IEEE 802.11 [36] and the contention phase of the IEEE 802.15.4 [37] which is designed for low rate WSNs. In this mechanism, every node that has a packet to transmit senses the channel and if it is idle for a period called distributed interframe space (DIFS), the node transmits. Otherwise, it waits until the channel is idle for a DIFS and then starts a backoff. The random backoff period is uniformly

---


$$\begin{aligned} c_{v_k,s} &= \lim_{z \rightarrow 1} (z-1) \det(\mathcal{C}_1 \mathcal{C}_2 \cdots \mathcal{D}_v \mathcal{C}_Q) \\ &= \det \begin{pmatrix} 1 - c_{11,11} & -c_{11,22} & \cdots & c_{v_1} & -c_{11,N-1} & N \\ -c_{22,11} & 1 - c_{22,22} & \cdots & c_{v_2} & -c_{22,N-1} & N \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ -c_{N-1} & N,11 & -c_{N-1} & N,22 & \cdots & 0 & 1 - c_{N-1} & N,N-1 & N \end{pmatrix}. \end{aligned} \quad (44)$$

selected between 0 and the contention window. The initial size of the contention window is  $CW$  and is doubled at each retransmission. Let  $R$  denotes the maximum number of retransmissions. Then, the maximum contention window size is  $CW_{\max} = 2^R CW$ . The backoff counter is decremented after each slot time provided that the channel is sensed idle. The transmission starts when the backoff counter is zero. If an acknowledgement (ACK) is received from the destination, the transmission is successful; otherwise, a collision is inferred. Let  $q_{k,\ell}$   $k \in \mathcal{N}$ ,  $\ell \in \mathcal{N}_k$  denote the probability of collision assigned to the transmissions with node  $k$  as destination and node  $\ell$  as source. Furthermore, we assume that each node  $k$  has  $n_k$  neighbors (degree of node  $k$  is  $n_k$ ). Node  $k$  successfully receives a packet from node  $\ell$  if none of its remaining neighbors or itself transmit simultaneously. We assume that all nodes are deploying the same set up for backoff procedure, i.e., the maximum number of retransmissions and the initial window sizes are identical. Consequently, each node transmits a packet with probability  $\tau$ . Thus, the collision probability  $q_{k,\ell}$  can be written as follows:

$$q_{k,\ell} = 1 - (1 - \tau)^{n_k}, \quad k \in \mathcal{N}, \ell \in \mathcal{N}_k. \quad (53)$$

We assume that each node knows how many neighboring nodes it has; thus,  $n_k$  is known. Additionally, we consider the seminal paper of Bianchi that develops a two state Markov chain to evaluate the performance of the exponential backoff algorithm to express  $\tau$  as a function of  $R$ ,  $CW$  and  $q_{k,\ell}$  as follows [38]:

$$\tau = \frac{2(1 - 2q_{k,\ell})}{(1 - 2q_{k,\ell})(CW + 1) + q_{k,\ell}CW(1 - (2q_{k,\ell})^R)}. \quad (54)$$

Solving (53) and (54), we find the probability of collision on each link. In a channel access mechanism based on the exponential backoff procedure as described above, packet collisions are closely related to the transmission errors  $p_{k,\ell}$ . If a packet collides more than the maximum number of retransmissions during the information exchange period, then the packet is discarded and a transmission error occurs. In other words, transmission errors  $p_{k,\ell}$  on all directional links to node  $k$  are identical and equivalent to the packet loss experienced by node  $k$  which is given by

$$p_{k,\ell} = q_{k,\ell}^{R+1}, \quad k \in \mathcal{N}, \ell \in \mathcal{N}_k. \quad (55)$$

#### IV. SIMULATIONS

In the first phase of simulations, we consider identical failure probabilities. To justify the assumption of identical error probabilities, consider as an example a scenario where each node randomly turns into sleep mode for power saving and does not share information with its neighbors. For such source of failure, it is reasonable to assume identical error probabilities in a network with homogeneous nodes. In the second phase of simulations, we consider another practical scenario where errors occur at the MAC level while nodes access the channel using a backoff procedure. In such scenario, we use (53)-(55) to model non-identical error probabilities experienced by each node.

#### A. Uniform Error Probabilities

We consider a small 7-node network where nodes are randomly distributed in a square area with side  $S = 100$  units. There exists a link between any pair of nodes with a distance less than 50 units. The network topology is shown in Fig. 1. For convenience, we denote the error probability by  $p$  since through this simulation, transmission errors are assumed to be identical for all links, i.e.,  $p_{k,\ell} = p$ ,  $(k,\ell) \in \mathcal{L}$ .

Without loss of generality, we apply the relative variance rule which gives more weight to nodes with lower noise variance to obtain the mixing coefficients [11]. Hence, it follows that  $a_{k,\ell,i} = \sigma_{v_\ell}^{-2} / (\sum_{m \in \mathcal{S}_{k,i}} \sigma_{v_m}^{-2})$  for  $\ell \in \mathcal{S}_{k,i}$  and otherwise we have that  $a_{k,\ell,i} = 0$ . Notice that any rule for finding the mixing coefficients (e.g., uniform [39], Laplacian [40], Metropolis [41], maximum degree [15], relative degree variance [11], Hasting [42], etc.) is possible as long as it satisfies the condition  $\sum_{\ell \in \mathcal{N}} a_{k,\ell,i} = 1$ ,  $k = 1, 2, \dots, N$  for all  $i$ , as discussed in Section II-A. Each node has access to the distorted and noisy version of the same unknown vector  $\mathbf{w}^o = \text{col}\{1, 1, \dots, 1\} / \sqrt{M}$ , with  $M = 2$  according to (4) which may represent for example, the location coordinates of a moving object [43]. The  $M$ -dimensional input regressors are assumed to be temporally independent Gaussian, but spatially correlated. The spatial correlation index  $\rho_{k\ell}$  between two nodes  $k$  and  $\ell$  is obtained according to  $\rho_{k\ell} = \rho^{|k-\ell|}$ , where  $\rho$  is a constant that lies between 0 and 1. The measurement noise is assumed to be white and Gaussian. The noise variances are generated randomly from  $[-40, 0)$  dB and shown in Fig. 2 (top). The variances of input regressors are randomly selected over  $(0, 1]$  and depicted in Fig. 2 (bottom). For all nodes we choose identical step-sizes, i.e.,  $\mu_k = \mu$  and we select two distinct sets of step-sizes ( $\mu = 0.01$  and  $\mu = 0.05$ ) to illustrate the trade-off between the convergence rate in transient phase and the error in steady-state. In order to obtain the performance measures, the results are averaged over 500 independent experiments each using 1,000 iterations. A random noise is generated at each run according to the noise profile shown in Fig. 2 (top).

Fig. 3 shows the learning curves in terms of the global MSD for different values of error probability, i.e.,  $p \in \{0, 0.2, 0.5, 0.7, 1\}$  and different values of step-sizes. Both step-sizes are selected according to (52) to ensure stability. As it can be seen from Fig. 3, a large step-size ( $\mu = 0.05$ ) makes the algorithm converge in less than 100 iterations. Contrarily, a small step-size ( $\mu = 0.01$ ) decreases the steady-state error with a price paid in terms of a slower convergence rate. To meet the conflicting requirements of large step-sizes in transient phase and small step-sizes in steady-state, variable step-size diffusion LMS algorithms should be exploited (see [32] for details).

To investigate the effect of the transmission errors on the steady-state MSD, we notice that in the presence of transmission errors, i.e.  $p > 0$ , the global network MSD at the steady-state increases compared with the error-free scenario ( $p = 0$ ). Another important observation is that the global convergence rate might increase when the error probability increases. In other words, transmission errors might prevent

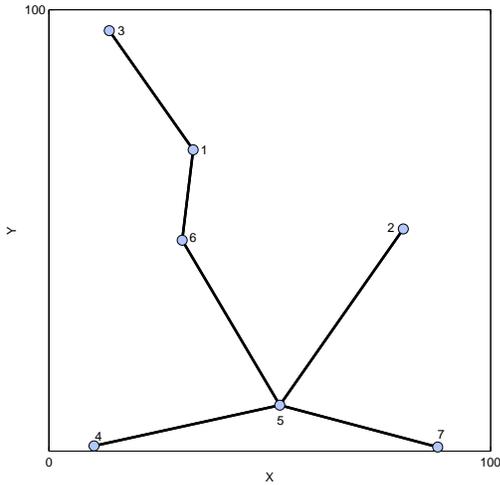


Fig. 1. Network topology for 7-node network. The node index is shown next to each node.

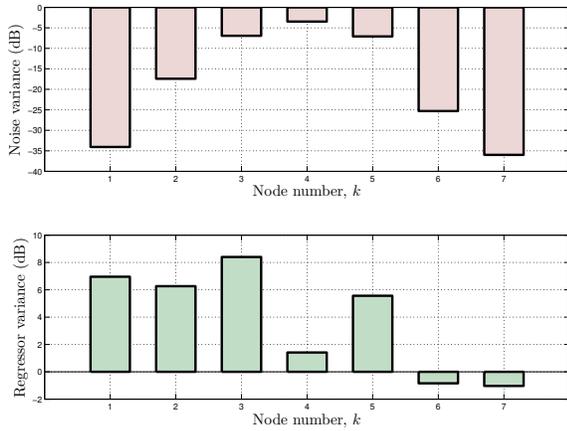


Fig. 2. Noise power profile  $\sigma_{v_k}^2$  (top) and regressor power profile  $\sigma_{u_k}^2$  (bottom) for 7-node network in Fig. 1.

the negative effect of a slow node on a fast converging node and thus improve the global convergence rate. In this network, for example, it might be better for node 7 with a low noise level to discard information received from node 5 with a high noise level rather than giving it some weight according to the relative variance rule. The policy of discarding such worthless received information is equivalent to not receiving the information at all due to transmission errors.

Notice that in general, we need to look at the convergence modes of adaptive algorithms to exploit their transient behavior. In a diffusion mode of cooperation, we may identify the modes of convergence by evaluating the  $N$  eigenvalues corresponding to matrix  $\mathcal{E}'_{n,s} = [\rho_i a_{i,j}]_{N \times N}$ , as well as the  $Q$  eigenvalues of matrix  $\mathcal{C}_{Q \times Q}$ . We plot the convergence modes of the 7-node network in ascending order for different values of error probability, i.e.,  $p \in \{0, 0.2, 0.5, 0.7, 1\}$  in Fig. 4. In Fig. 4, we also illustrate the magnified view of the dominant convergence mode (the eigenvalue with the largest absolute value) and the second largest eigenvalue. We observe that in the presence of transmission errors ( $p \in \{0.2, 0.5, 0.7\}$ ), the first and second dominant modes of convergence are smaller

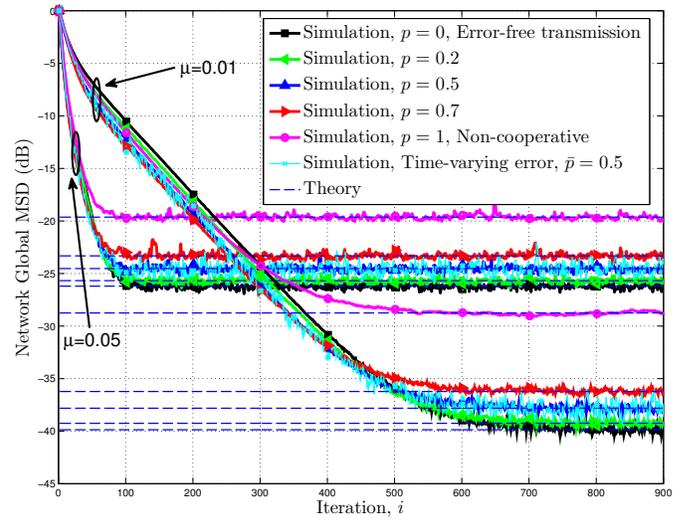


Fig. 3. Learning curve in terms of the global MSD in dB for different values of time-invariant error probability  $p \in \{0.2, 0.5, 0.7\}$  and time-varying errors  $\bar{p} = 0.5$  compared with non-cooperative ( $p = 1$ ) and error-free ( $p = 0$ ) scenarios in 7-node network with two distinct sets of step-sizes ( $\mu = 0.01$  and  $\mu = 0.05$ ). The dashed lines show the theoretical expression (49) for the steady-state MSD.

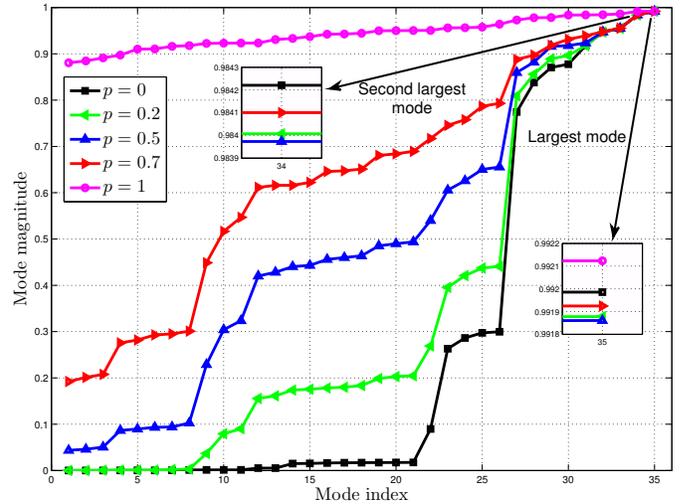


Fig. 4. Modes of convergence in 7-node network with uniform errors for different values of error probability  $p \in \{0, 0.2, 0.5, 0.7, 1\}$ . First and second largest modes are shown in the magnified images.

compared with the error-free scenario ( $p = 0$ ) which makes the algorithm converge faster.

Up to now, we assumed that the error probability  $p_{k,\ell}$  is time-invariant. Note that however, the error patterns are time-variant. The assumption that  $p_{k,\ell}$  is time-invariant, is reasonable in stationary or slowly time-varying scenarios. In such scenarios, the error probability can be assumed to be time-invariant during the whole period required for the algorithm to converge. In most wireless sensor environments with limited sources of energy, fewer number of iterations before convergence is desirable. Although, in scenarios with fewer iterations to converge, it is reasonable to assume a time-invariant  $p_{k,\ell}$ ; our analysis is valid even if the error

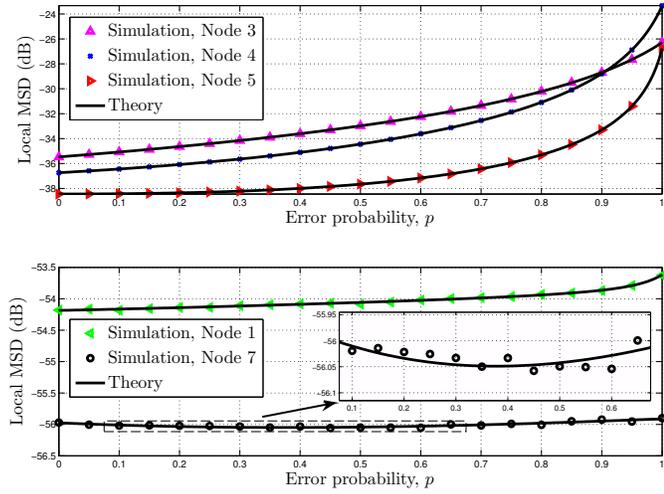


Fig. 5. Local steady-state MSD in dB for nodes 1,3,4,5 and 7 as a function of error probability in 7-node network. The solid lines curves show the theoretical values obtained from expression (48) and the markers represent the simulation results. The magnified image shows the Local steady-state MSD for node 7 around its minimum at  $p = 0.375$ .

probabilities  $p_{k,\ell}$  have instant variations even between two successive iterations. To illustrate, we plot a distinct curve corresponding to time-variant error probabilities in Fig. 3. The error probability  $p_{k,\ell}$  at each time iteration  $i$  is generated uniformly from  $[0, 1]$ . Consequently, a node receives the packet on average in half of the iterations and the average probability of error is  $\bar{p}_{k,\ell} = \bar{p} = 0.5$ . As it can be seen from the figure, the theoretical steady-state MSD match very well with this simulation curve.

To further investigate the impacts of the errors on the performance of the diffusion estimation algorithm, we evaluate the local steady-state MSD associated to each node by averaging over the last 1,000 samples of the individual learning curves. Local steady-state MSD curves as a function of error probability for nodes  $\{1, 3, 4, 5, 7\}$  are illustrated in Fig. 5. We observe that the MSD values obtained from analysis and simulation coincide with each other. It is seen that the MSD curves for a group of nodes  $\{1, 3, 4, 5\}$  are monotonically increasing functions of the error probability; on the contrary, for node  $\{7\}$  there exist non-zero values of probability of error that minimizes the corresponding local MSD. From the magnified image of the local MSD curve corresponding to node 7, we observe that the minimum MSD is  $-56.05$  dB occurring at  $p = 0.375$ . This is due to the fact that the noise variances of neighbors of node 7 ( $\{2, 4, 5\}$ ) are significantly higher than that of  $\{7\}$ . As a result, there exist certain optimum points in terms of the error probability that minimizes the noise amplification effect and consequently minimizes the local steady-state MSD of the nodes with low noise variances. Thus, we conclude that in general the local and global steady-state MSD values are not necessarily monotonically increasing functions of the error probability.

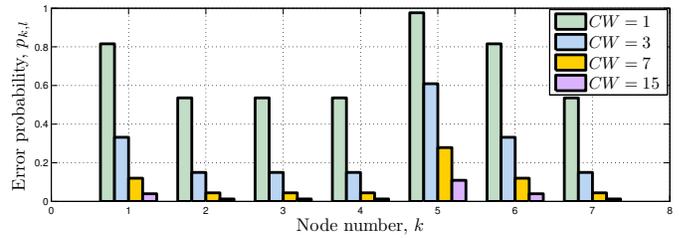


Fig. 6. MAC-collision based error probability  $p_{k,\ell}$  corresponding to backoff parameters  $R = 1$  and  $CW \in \{1, 3, 7, 15\}$  for 7-node network in Fig. 1.

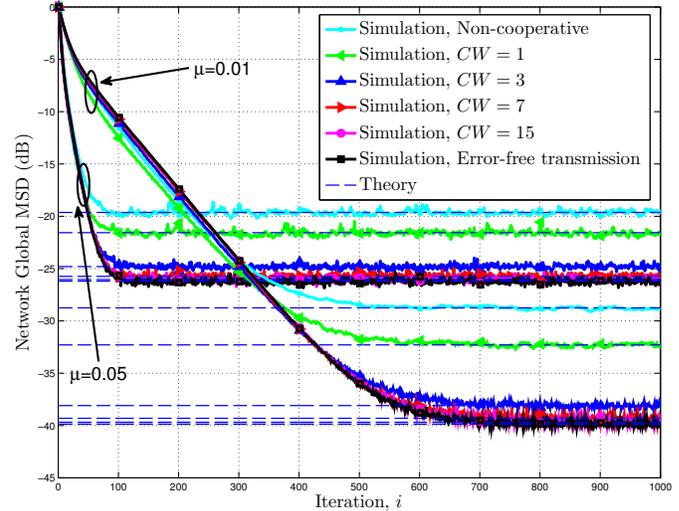


Fig. 7. Learning curve in terms of the global MSD in dB when errors arise from collisions at the MAC level of 7-node network in Fig. 1, for different values of contention window  $CW \in \{1, 3, 7, 15\}$  compared with non-cooperative and error-free scenarios with two distinct sets of step-sizes ( $\mu = 0.01$  and  $\mu = 0.05$ ). The dashed lines show the theoretical expression (49) for the steady-state MSD.

## B. MAC-collision Based Errors

In the second phase of simulations, we concentrate on the MAC-level errors. First, we consider the small-scale network as shown in Fig. 1 with the simulation parameters identical to those used in the previous subsection. To model the transmission errors we perform a discrete-event simulation with one retransmission ( $R = 1$ ) and contention window sizes  $CW \in \{1, 3, 7, 15\}$ . The probability of errors due to collisions at the MAC level for different values of contention window is shown in Fig. 6. As expected, nodes in dense areas with more neighbors (for example, node 5) experience more collisions and higher error rates. As the contention window size increases, the error probability decreases. We now simulate the diffusion estimation algorithm for the corresponding MAC level errors. Learning curves in terms of the global MSD for different values of contention window and two distinct step-sizes are shown in Fig. 7. From Fig. 7, it is seen that the theoretical values match very well with the steady-state values obtained from simulation and as the contention window size increases, the performance of the algorithm tends to the error-free case.

As the last experiment, we consider a medium-scale network including 30 nodes randomly placed in a square region with

side  $S = 100$  units as shown in Fig. 8. Nodes communicate with neighbors located within the range of 25 units. Noise levels are generated randomly from  $[-30 \text{ dB}, -10 \text{ dB}]$  and variances of input regressors are randomly selected from  $(0, 2]$ . We select  $M = 2$  and identical step-sizes:  $\mu = \mu_k = 0.1$ . Similar to the previous experiment, we test the performance of the diffusion LMS algorithm in the presence of MAC level errors. From Fig. 9, we observe that the error probability is considerably higher than the previous case due to the congested links and the high density of the network. In such a dense network, the imperfect data transfer among nodes during information exchange, results in a significant performance degradation in terms of the network global MSD as shown in Fig. 10. From Fig. 10 we observe that for  $CW = 1$ , the steady-state error is significantly higher than the error-free scenario and the network behavior is close to the non-cooperative mode. To resolve this problem, we implement an energy-efficient technique aimed at limiting the number of data transmissions. In this implementation each node communicates solely with one neighbor which is selected randomly during each information exchange period. From Fig. 9, we observe that the error probability for random pairwise strategy is significantly lower compared with standard diffusion algorithm. A small window size for contention at the MAC level is preferable to save energy in WSNs. From Fig. 10, we observe that the performance of the random pairwise strategy is improved compared with the standard diffusion in the presence of high transmission error rates, i.e., when  $CW \in \{1, 3\}$ . The random pairwise strategy thus, increases the energy efficiency by reducing the number of data transmissions; meanwhile it enhances the performance of the distributed estimation algorithm in the presence of high transmission error rates.

## V. CONCLUSION

We formulate the problem of distributed estimation based on the diffusion cooperation scheme over adaptive networks in the presence of transmission errors. We propose a theoretical framework and derive the closed-form expressions for the local and global steady-state MSD's under the assumption of imperfect information flow. Compared to the related work in the context of diffusion algorithms with error-free information exchange, the proposed analysis has less complexity and ensures scalability in terms of the input regressor size and the network size. Additionally, the present study does not impose the independence assumption between the observation vectors which in turn allows us to examine the performance measures of the distributed nodes with spatially correlated regressors. Simulation and analysis verify that a well-designed diffusion estimation algorithm will converge slower, achieving a higher steady-state MSD as a result of transmission errors. More importantly, we conclude that the local and global steady-state MSD values are not necessarily monotonically increasing functions of the error probability. We also derive practically applicable sufficient conditions to assure the stability of diffusion LMS strategies with imperfect information sharing. Finally, we study a practical case scenario where errors occur at the MAC layer and implement a random pairwise technique to mitigate the negative effects of such errors.

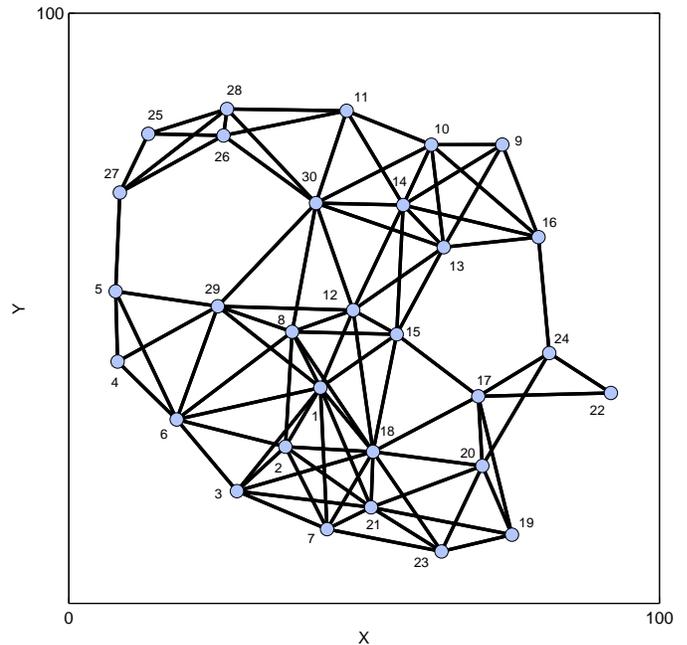


Fig. 8. Network topology for 30-node network. The node index is shown next to each node.

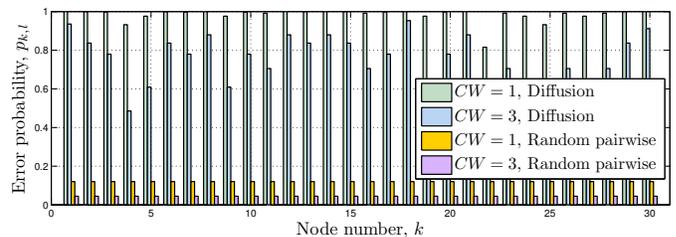


Fig. 9. MAC-collision based error probability  $p_{k, \ell}$  corresponding to backoff parameters  $R = 1$  and  $CW \in \{1, 3\}$  for 30-node network in Fig. 8 when standard diffusion and random pairwise diffusion strategies are implemented.

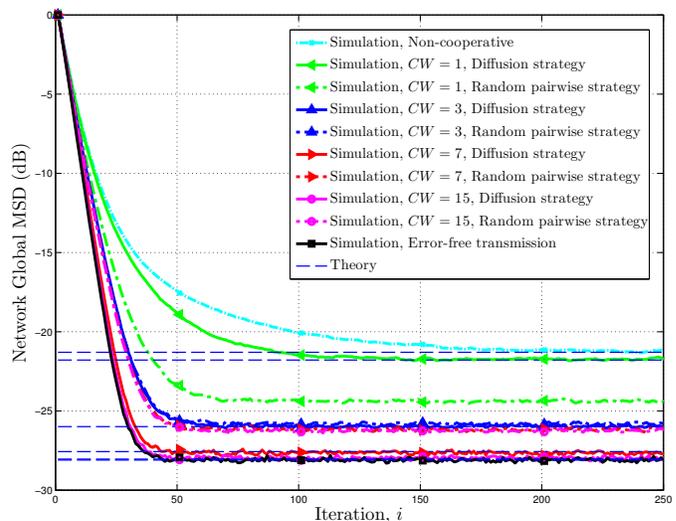


Fig. 10. Learning curve in terms of the global MSD in dB when errors arise from collisions at the MAC level of 30-node network in Fig. 8, for standard diffusion and random pairwise LMS techniques, using different values of contention window  $CW \in \{1, 3, 7, 15\}$  compared with non-cooperative and error-free scenarios in 30-node network.

## APPENDIX A

DIFFERENT REPRESENTATION OF  $\phi_{k,i}$ 

An alternative method to represent  $\phi_{k,i}$  is as follows:

$$\begin{aligned} \phi_{k,i} = & a'_{k,k,i} \mathbf{w}_{k,i} + \sum_{\ell \in \mathcal{N}_k \setminus \{k\}} a'_{k,\ell,i} \mathbf{w}_{\ell,i} \\ & - \sum_{\ell \in \mathcal{N}_k \setminus \{k\}} \delta_{k,\ell,i} a'_{k,\ell,i} \mathbf{w}_{\ell,i} + \sum_{\ell \in \mathcal{S}_{k,i}} a''_{k,\ell,i} \mathbf{w}_{\ell,i}, \end{aligned} \quad (56)$$

where  $\delta_{k,\ell,i}, k = 1, 2, \dots, N, \ell \in \mathcal{N}_k \setminus \{k\}$  is a Bernoulli random variable with parameter  $p_{k,\ell}$ :

$$\delta_{k,\ell,i} := \begin{cases} 1 & \text{with probability } p_{k,\ell} \\ 0 & \text{with probability } 1 - p_{k,\ell} \end{cases}. \quad (57)$$

In (56), the coefficients  $a'_{k,\ell,i}$  represent the mixing coefficients when perfect information exchange is established and  $0 \leq a''_{k,\ell,i} < 1$  are some virtual coefficients added to the expression to ensure a convex combination of weight vectors. The interpretation of the third term on the right hand side of (56) is to eliminate the local weight vectors of those nodes that have not been able to successfully transmit their information to node  $k$ . In the case of error-free transmissions ( $\delta_{k,\ell,i} = 0$ ), the virtual coefficients are set to 0. Otherwise, they are non-zero and less than 1. We call them virtual, because they don't actually exist and the mixing coefficients  $a_{k,\ell,i}$  are calculated over the set of neighbors that successfully transmitted their information, as written in (2).

## APPENDIX B

## DERIVATION OF (13)

Define the one-sided  $z$ -transform of the weight vector  $\mathbf{w}_k$  as  $\mathcal{W}_k = \mathcal{Z}\{\mathbb{E}[\mathbf{w}_k]\}$ . Taking the  $z$ -transform of (10), we arrive at

$$(1 - a_{k,k} \rho_k z^{-1}) \mathcal{W}_k - z^{-1} \sum_{\ell \in \mathcal{N}, \ell \neq k} a_{k,\ell} \mathcal{W}_\ell = \frac{c_k}{1 - z^{-1}}. \quad (58)$$

Thus, we can write the set of equations (59) as shown at the bottom of the next page. For convenience, we define matrix  $\mathcal{E}'_{n,s}$  to collect  $a_{k,\ell}$  and  $\rho_k, k, \ell \in \mathcal{N}$  as  $\mathcal{E}'_{n,s} = [a_{k,\ell} \rho_k]_{N \times N}$ . Using the Cramer's rule we obtain the following expression for  $\mathcal{W}_k$

$$\mathcal{W}_k = \frac{\det(\mathcal{E}_{n,\mathcal{F}_k})}{\det(\mathcal{E}_n)} \mathbf{w}^o, \quad k = 1, 2, \dots, N, \quad (60)$$

where

$$\mathcal{E}_n = \mathcal{I}_N - z^{-1} \mathcal{E}'_{n,s}, \quad (61)$$

and  $\mathcal{E}_{n,\mathcal{F}_i}$  is obtained by replacing the  $i^{\text{th}}$  column of  $\mathcal{E}_n$  by  $\mathcal{F}_n$  defined as

$$\mathcal{F}_n = \frac{1}{1 - z^{-1}} [\mu_1 \sigma_{u_1}^2 \quad \mu_2 \sigma_{u_2}^2 \quad \dots \quad \mu_N \sigma_{u_N}^2]^\top. \quad (62)$$

Denote  $\mathbb{E}[\mathbf{w}_{k,s}]$  as the expectation of the weight vector of the  $k^{\text{th}}$  node at the steady-state. Then, we yield

$$\mathbb{E}[\mathbf{w}_{k,s}] = \frac{\det(\mathcal{E}_{n,s,\mathcal{F}_k})}{\det(\mathcal{E}_{n,s})} \mathbf{w}^o, \quad k = 1, 2, \dots, N, \quad (63)$$

where  $\mathcal{E}_{n,s} = \mathcal{I}_N - \mathcal{E}'_{n,s}$  and  $\mathcal{E}_{n,s,\mathcal{F}_i}$  is obtained by replacing the  $i^{\text{th}}$  column of  $\mathcal{E}_{n,s}$  by the  $N \times 1$  column vector  $\mathcal{F}_{n,s}$  whose  $k^{\text{th}}$  element is  $\mu_k \sigma_{u_k}^2$ . Recall the following property of determinant: if any column of the determinant is replaced by a new column which is a linear combination of all columns, then the value of the determinant is not altered. As a result, for any arbitrary determinant and considering the sum of all columns as a particular linear combination of the columns, we can write

$$\det(\mathcal{C}_1 \mathcal{C}_2 \dots \mathcal{C}_i \dots \mathcal{C}_Q) = \det(\mathcal{C}_1 \mathcal{C}_2 \dots \mathcal{C}'_i \dots \mathcal{C}_Q), \quad (64)$$

where  $\mathcal{C}'_i = \sum_{j=1}^Q \mathcal{C}_j$ , for all  $i = 1, 2, \dots, Q$ . Let us denote each column of  $\mathcal{E}_{n,s}$  by  $\mathcal{E}_j$  and find the sum of the columns of  $\mathcal{E}_{n,s}$  as follows:

$$\begin{aligned} \sum_{j \in \mathcal{N}} \mathcal{E}_j &= [1 - \rho_1 \sum_{\ell \in \mathcal{N}} a_{1,\ell} \quad \dots \quad 1 - \rho_N \sum_{\ell \in \mathcal{N}} a_{N,\ell}]^\top \\ &= [1 - \rho_1 \quad \dots \quad 1 - \rho_N]^\top = \mathcal{F}_{n,s}. \end{aligned} \quad (65)$$

Notice that in the first and second step, we used (12) and (7) respectively. Hence, replacing any column of the  $\mathcal{E}_{n,s}$  by  $\mathcal{F}_{n,s}$  does not alter the value of the determinant of  $\mathcal{E}_{n,s}$ . The expressions in (13) is then derived from (63).

## APPENDIX C

## DERIVATION OF (20)

We begin by replacing (2)-(4) in the weight update rule (1) which gives the following equation:

$$\begin{aligned} \mathbf{w}_{k,i+1} = & a_{k,k,i} \mathbf{w}_{k,i} + \sum_{\ell \in \mathcal{N}_k \setminus \{k\}} a_{k,\ell,i} \mathbf{w}_{\ell,i} + \mu_k v_{k,i} \mathbf{u}_{k,i} \\ & + \mu_k \mathbf{u}_{k,i} \mathbf{u}_{k,i}^\top \mathbf{w}^o - a_{k,k,i} \mu_k \mathbf{u}_{k,i} \mathbf{u}_{k,i}^\top \mathbf{w}_{k,i} \\ & - \sum_{\ell \in \mathcal{N}_k \setminus \{k\}} a_{k,\ell,i} \mu_k \mathbf{u}_{k,i} \mathbf{u}_{k,i}^\top \mathbf{w}_{\ell,i}. \end{aligned} \quad (66)$$

Followed by multiplying recursion (66) with correct indexes and taking the conditional expectation given that event  $\varepsilon_j(i)$  occurs during data transmission at iteration  $i$ , we arrive at (20).

Note that in order to compute expressions of the form  $\mathbb{E}[\mathbf{w}_{m,i}^\top \mathbf{u}_{\ell,i} \mathbf{u}_{h,i} \mathbf{u}_{h,i}^\top \mathbf{w}_{n,i}]$ , we use the independence assumption in the context of adaptive filters, i.e., the statistical correlations between the regressor vectors and the weight vectors is negligible [44], [45]. Also recall that the observation vectors are assumed to be temporally independent identically distributed (i.i.d.) white Gaussian random variables. This enables us to use the Gaussian moment factoring theorem [46]. With these in mind and omitting the time index  $i$  for simplicity, a term of the form  $\mathbb{E}[\mathbf{w}_m^\top \mathbf{u}_\ell \mathbf{u}_h \mathbf{u}_h^\top \mathbf{w}_n]$  can be written as:

$$\begin{aligned}
& \mathbb{E}[\mathbf{w}_m^\top \mathbf{u}_\ell \mathbf{u}_\ell^\top \mathbf{u}_h \mathbf{u}_h^\top \mathbf{w}_n] \\
&= \mathbb{E}\left[\sum_k w_{m,k} u_{\ell,k} \sum_j u_{\ell,j} u_{h,j} \sum_i u_{h,i} w_{n,i}\right] \\
&= \sum_{k,j,i} \mathbb{E}[w_{m,k} u_{\ell,k} u_{\ell,j} u_{h,j} u_{h,i} w_{n,i}] \\
&= \sum_{k,j,i} \mathbb{E}[w_{m,k} w_{n,i}] \mathbb{E}[u_{\ell,k} u_{\ell,j} u_{h,j} u_{h,i}], \quad (67)
\end{aligned}$$

where the notations  $w_{m,k}$ ,  $k = 1, \dots, M$  and  $u_{\ell,k}$ ,  $k = 1, \dots, M$  are used to represent the  $k^{\text{th}}$  elements of vectors  $\mathbf{w}_m$  and  $\mathbf{u}_m$  respectively. Notice that in the last step of Equation (67), the independence assumption is used. Using the Gaussian moment factoring theorem, it can be verified that

$$\begin{aligned}
& \mathbb{E}[\mathbf{w}_m^\top \mathbf{u}_\ell \mathbf{u}_\ell^\top \mathbf{u}_h \mathbf{u}_h^\top \mathbf{w}_n] \\
&= \sum_i \mathbb{E}[w_{m,i} w_{n,i}] (\sigma_{u_\ell}^2 \sigma_{u_h}^2 + 2\sigma_{u_{\ell h}}^4) \\
&+ \sum_i \mathbb{E}[w_{m,i} w_{n,i}] (M-1) \sigma_{u_{\ell h}}^4 \\
&= [\sigma_{u_\ell}^2 \sigma_{u_h}^2 + (M+1) \sigma_{u_{\ell h}}^4] \sum_i \mathbb{E}[w_{m,i} w_{n,i}] \\
&= [\sigma_{u_\ell}^2 \sigma_{u_h}^2 + (M+1) \sigma_{u_{\ell h}}^4] \mathbb{E}[\mathbf{w}_m^\top \mathbf{w}_n]. \quad (68)
\end{aligned}$$

#### APPENDIX D DERIVATION OF (47)

Using (41), we rewrite  $\mathcal{D}_{w^o}$  as follows:

$$[\mathcal{D}_{w^o}]_i = z^{-1} \sum_{m \in \mathcal{N}} c_{j,om} \mathbf{w}^{o\top} \mathcal{W}_m + \frac{\nu_j \mathbf{w}^{o\top} \mathbf{w}^o}{1-z^{-1}}, \quad i = 1, \dots, Q, \quad (69)$$

where index  $j$  selects the proper coefficient for each index  $i$  according to the permutation that is used to obtain the set of equations in (38). The key is to prove that

$$\lim_{z \rightarrow 1} (z-1) [\mathcal{D}_{w^o}]_i = \mathbf{w}^{o\top} \mathbf{w}^o \sum_{m \in \mathcal{N}} c_{j,om} + \nu_j, \quad (70)$$

for all  $i = 1, 2, \dots, Q$ . Denote by  $\mathcal{E}_{n,i}$  the  $i^{\text{th}}$  column of  $\mathcal{E}_n$ . Then it holds that  $\mathcal{E}'_n = \sum_{i \in \mathcal{N}} \mathcal{E}_{n,i} = [1 - z^{-1} \rho_j]_j$ . Using  $\rho_j = 1 - \mu_j \sigma_{u_j}^2$  and (62), we obtain:

$$\begin{aligned}
[\mathcal{E}'_n]_j &= 1 - z^{-1} (1 - \mu_j \sigma_{u_j}^2) \\
&= 1 - z^{-1} (1 - (1 - z^{-1}) [\mathcal{F}_n]_j) \\
&= (1 - z^{-1}) (1 + z^{-1} [\mathcal{F}_n]_j). \quad (71)
\end{aligned}$$

We know that

$$\begin{aligned}
\det(\mathcal{E}_n) &= \det(\mathcal{E}_{n,1} \cdots \mathcal{E}'_n \cdots \mathcal{E}_{n,N}) \\
&= (1 - z^{-1}) \det(\mathcal{E}_{n,1} \cdots 1 + z^{-1} \mathcal{F}_n \cdots \mathcal{E}_{n,N}) \\
&= (1 - z^{-1}) [\det(\mathcal{E}_{n,1} \cdots 1 \cdots \mathcal{E}_{n,N}) \\
&+ z^{-1} \det(\mathcal{E}_{n,1} \cdots \mathcal{F}_n \cdots \mathcal{E}_{n,N})] \\
&= (1 - z^{-1}) [\det(\mathcal{E}_{n,1} \cdots 1 \cdots \mathcal{E}_{n,N}) \\
&+ z^{-1} \det(\mathcal{E}_n, \mathcal{F}_n)].
\end{aligned}$$

Therefore (60) can be written as follows:

$$\mathcal{W}_i = \frac{\mathbf{w}^o}{(1 - z^{-1}) \zeta(z) + z^{-1} (1 - z^{-1})}, \quad (72)$$

where

$$\zeta(z) = \frac{\det(\mathcal{E}_{n,1} \cdots 1 \cdots \mathcal{E}_{n,N})}{\det(\mathcal{E}_n, \mathcal{F}_n)}. \quad (73)$$

With these, (69) becomes

$$[\mathcal{D}_{w^o}]_i = \frac{\mathbf{w}^{o\top} \mathbf{w}^o}{1 - z^{-1}} [z^{-1} \sum_{m \in \mathcal{N}} \frac{c_{j,om}}{\zeta(z) + z^{-1}} + \nu_j], \quad (74)$$

and we get

$$\begin{aligned}
& \lim_{z \rightarrow 1} (z-1) [\mathcal{D}_{w^o}]_i \\
&= \lim_{z \rightarrow 1} z \mathbf{w}^{o\top} \mathbf{w}^o [z^{-1} \sum_{m \in \mathcal{N}} \frac{c_{j,om}}{\zeta(z) + z^{-1}} + \nu_j]. \quad (75)
\end{aligned}$$

Note that

$$\det(\mathcal{E}_n, \mathcal{F}_i) = \frac{1}{1 - z^{-1}} \det(\mathcal{E}_{n,1} \cdots [\mu_i \sigma_{u_i}^2]_{i \in \mathcal{N}} \cdots \mathcal{E}_{n,N}).$$

Therefore,  $\zeta(z)$  can be expressed as follows

$$\zeta(z) = \frac{(1 - z^{-1}) \det(\mathcal{E}_{n,1} \cdots 1 \cdots \mathcal{E}_{n,N})}{\det(\mathcal{E}_{n,1} \cdots [\mu_i \sigma_{u_i}^2]_{i \in \mathcal{N}} \cdots \mathcal{E}_{n,N})}, \quad (76)$$

and we arrive at (70) under one of the following conditions: either If we have that  $[\mu_i \sigma_{u_i}^2]_{i \in \mathcal{N}} = \mu \sigma_u^2$  or we have that  $\det(\mathcal{E}_{n,1} \cdots [\mu_i \sigma_{u_i}^2]_{i \in \mathcal{N}} \cdots \mathcal{E}_{n,N}) \neq 0$ . In order to proceed, we use (70) to obtain the following expression for  $c_{w^o,s}$

$$c_{w^o,s} = \mathbf{w}^{o\top} \mathbf{w}^o \det(\mathcal{C}_1 \mathcal{C}_2 \cdots \mathcal{D}_s \mathcal{C}_Q), \quad (77)$$

where  $[\mathcal{D}_s]_i = \sum_{m \in \mathcal{N}} c_{j,om} + \nu_j$ ,  $i = 1, \dots, Q$ . After some algebra and using (28) and (32) to replace  $c_{k\ell,om}$ ,  $k, \ell, m \in \mathcal{N}$

$$\underbrace{\begin{bmatrix} (1 - z^{-1} a_{1,1} \rho_1) \mathcal{I}_M & -z^{-1} a_{1,2} \rho_1 \mathcal{I}_M & \cdots & -z^{-1} a_{1,N} \rho_1 \mathcal{I}_M \\ -z^{-1} a_{2,1} \rho_2 \mathcal{I}_M & (1 - z^{-1} a_{2,2} \rho_2) \mathcal{I}_M & \cdots & -z^{-1} a_{2,N} \rho_2 \mathcal{I}_M \\ \vdots & \vdots & \ddots & \vdots \\ -z^{-1} a_{N,1} \rho_N \mathcal{I}_M & -z^{-1} a_{N,2} \rho_N \mathcal{I}_M & \cdots & (1 - z^{-1} a_{N,N} \rho_N) \mathcal{I}_M \end{bmatrix}}_{[\mathcal{E}]_{NM \times NM}} \underbrace{\begin{bmatrix} \mathcal{W}_1 \\ \mathcal{W}_2 \\ \vdots \\ \mathcal{W}_N \end{bmatrix}}_{[\mathcal{W}]_{NM \times 1}} = \underbrace{\begin{bmatrix} \frac{c_1}{1 - z^{-1}} \\ \frac{c_2}{1 - z^{-1}} \\ \vdots \\ \frac{c_N}{1 - z^{-1}} \end{bmatrix}}_{[\mathcal{F}]_{NM \times 1}}. \quad (79)$$

and then using  $\sum_{\ell \in \mathcal{N}} a_{k,\ell} = 1$ ,  $k = 1, 2, \dots, N$  we obtain  $[\mathcal{D}_s]_i = 1 - \eta_j$ ,  $i = 1, \dots, Q$ . Using (21), (22), (23), (34) and (35) we arrive at

$$2\epsilon_k - \nu_k = 1 - \eta_k, \quad k = 1, 2, \dots, N, \quad (78)$$

$$\epsilon_k + \epsilon_\ell - \nu_{k\ell} = 1 - \eta_{k\ell}, \quad k, \ell \in \mathcal{N}, k \neq \ell. \quad (79)$$

With these, the sum of all columns of  $\mathcal{I}_N - \mathcal{C}'$  according to (45) becomes:

$$\sum_{q=1}^Q \mathcal{C}_q = \left[ 1 - \sum_{k,\ell \in \mathcal{N}} c_{11,k\ell} \dots 1 - \sum_{k,\ell \in \mathcal{N}} c_{N-1,N,k\ell} \right]^T. \quad (80)$$

Using (26), (27), (30) and (31), it can be checked that

$$\begin{aligned} \sum_{q=1}^Q \mathcal{C}_q &= \left[ 1 - \eta_1 \sum_{j \in \mathcal{V}} p_j \left( \sum_{m \in \mathcal{N}} a_{1,m}^{(j)} \right)^2 \dots \right. \\ &\quad \left. 1 - \eta_{N-1} \sum_{j \in \mathcal{V}} p_j \left( \sum_{m \in \mathcal{N}} a_{N-1,m}^{(j)} \sum_{n \in \mathcal{N}} a_{N,n}^{(j)} \right) \right]^T \\ &= \left[ 1 - \eta_1 \dots 1 - \eta_{N-1} \ N \right]^T = \mathcal{D}_s. \end{aligned} \quad (81)$$

Recalling the determinant property as stated in (64), we conclude that replacing any of the columns of  $\mathcal{I}_N - \mathcal{C}'$  by  $\mathcal{D}_s$  does not modify the value of its determinant, i.e.,  $\det(\mathcal{C}_1 \dots \mathcal{D}_s \dots \mathcal{C}_Q) = \det(\mathcal{C})$ . As a result, equality (46) holds.

## REFERENCES

- [1] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Elsevier Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.
- [2] J. Stankovic, "Research directions for the internet of things," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 3–9, Feb. 2014.
- [3] D. Li, K. Wong, Y. H. Hu, and A. Sayeed, "Detection, classification, and tracking of targets," *IEEE Signal Process. Mag.*, vol. 19, no. 2, pp. 17–29, Mar. 2002.
- [4] D. Culler, D. Estrin, and M. Srivastava, "Overview of sensor networks," *IEEE Computer*, vol. 37, no. 8, pp. 41–49, Aug. 2004.
- [5] M. Rabbat and R. Nowak, "Quantized incremental algorithms for distributed optimization," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 4, pp. 798–808, Apr. 2005.
- [6] A. Dimakis, S. Kar, J. Moura, M. Rabbat, and A. Scaglione, "Gossip algorithms for distributed signal processing," *Proceedings of the IEEE*, vol. 98, no. 11, pp. 1847–1864, Nov 2010.
- [7] A. H. Sayed, S.-Y. Tu, J. Chen, X. Zhao, and Z. J. Towfic, "Diffusion strategies for adaptation and learning over networks," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 155–171, 2013.
- [8] C. Lopes and A. H. Sayed, "Incremental adaptive strategies over distributed networks," *IEEE Trans. Signal Process.*, vol. 55, no. 8, pp. 4064–4077, Aug. 2007.
- [9] C. G. Lopes and A. H. Sayed, "Diffusion least-mean squares over adaptive networks: Formulation and performance analysis," *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 3122–3136, Jul. 2008.
- [10] F. Cattivelli, C. Lopes, and A. H. Sayed, "Diffusion recursive least-squares for distributed estimation over adaptive networks," *IEEE Trans. Signal Process.*, vol. 56, no. 5, pp. 1865–1877, May 2008.
- [11] F. Cattivelli and A. H. Sayed, "Diffusion LMS strategies for distributed estimation," *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1035–1048, Mar. 2010.
- [12] A. Jadbabaie, J. Lin, and A. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Trans. Autom. Control*, vol. 48, no. 6, pp. 988–1001, Jun. 2003.
- [13] R. Olfati-Saber and R. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1520–1533, Sep. 2004.
- [14] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Trans. Autom. Control*, vol. 54, no. 1, pp. 48–61, Jan. 2009.
- [15] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *Proc. Int. Conf. Information Processing in Sensor Networks (ISPN)*, vol. 4, Apr. 2005, pp. 63–70.
- [16] —, "A space-time diffusion scheme for peer-to-peer least-squares estimation," in *Proc. Int. Conf. Information Processing in Sensor Networks (ISPN)*, Apr. 2006, pp. 168–176.
- [17] A. H. Sayed and C. G. Lopes, "Distributed processing over adaptive networks," in *Proc. International Symposium on Signal Processing and Its Applications*, Feb. 2007, pp. 1–3.
- [18] C. G. Lopes and A. H. Sayed, "Diffusion least-mean squares over adaptive networks," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 3, Apr. 2007, pp. III–917–III–920.
- [19] —, "Steady-state performance of adaptive diffusion least-mean squares," in *Proc. IEEE Workshop on Statistical Signal Processing (SSP)*, Aug. 2007, pp. 136–140.
- [20] F. S. Cattivelli and A. H. Sayed, "Diffusion LMS algorithms with information exchange," in *Proc. Asilomar Conf. Signals, Syst., Comput.*, Pacific Grove, CA, Oct. 2008, pp. 251–255.
- [21] —, "Diffusion strategies for distributed Kalman filtering and smoothing," *IEEE Trans. Autom. Control*, vol. 55, no. 9, pp. 2069–2084, Sep. 2010.
- [22] S. Kar and J. Moura, "Distributed consensus algorithms in sensor networks with imperfect communication: Link failures and channel noise," *IEEE Trans. Signal Process.*, vol. 57, no. 1, pp. 355–369, Jan. 2009.
- [23] A. Tahbaz-Salehi and A. Jadbabaie, "Consensus over ergodic stationary graph processes," *IEEE Trans. Autom. Control*, vol. 55, no. 1, pp. 225–230, Jan. 2010.
- [24] —, "Consensus over ergodic stationary graph processes," *IEEE Trans. Autom. Control*, vol. 55, no. 1, pp. 225–230, Jan. 2010.
- [25] A. Khalili, M. Tinati, A. Rastegarnia, and J. Chambers, "Steady-state analysis of diffusion LMS adaptive networks with noisy links," *IEEE Trans. Signal Process.*, vol. 60, no. 2, pp. 974–979, Feb. 2012.
- [26] X. Zhao, S.-Y. Tu, and A. H. Sayed, "Diffusion adaptation over networks under imperfect information exchange and non-stationary data," *IEEE Trans. Signal Process.*, vol. 60, no. 7, pp. 3460–3475, Jul. 2012.
- [27] P. Di Lorenzo and S. Barbarossa, "Distributed estimation and control of algebraic connectivity over random graphs," *IEEE Trans. Signal Process.*, vol. 62, no. 21, pp. 5615–5628, Nov. 2014.
- [28] R. Abdolee and B. Champagne, "Diffusion LMS algorithms for sensor networks over non-ideal inter-sensor wireless channels," in *Proc. IEEE Int. Conf. Dist. Comput. Sensor Syst. (DCOSS)*, Jun. 2011, pp. 1–6.
- [29] A. Rastegarnia, W. Bazzi, A. Khalili, and J. Chambers, "Diffusion adaptive networks with imperfect communications: link failure and channel noise," *IET Signal Process.*, vol. 8, no. 1, pp. 59–66, Feb. 2014.
- [30] S. Ghazanfari-Rad and F. Labeau, "Diffusion least-mean squares over distributed networks in the presence of MAC errors," in *Proc. Asilomar Conf. Signals, Syst., Comput.*, Pacific Grove, CA, Nov. 2012, pp. 1787–1791.
- [31] P. Di Lorenzo, "Diffusion adaptation strategies for distributed estimation over gaussian markov random fields," *IEEE Trans. Signal Process.*, vol. 62, no. 21, pp. 5748–5760, Nov. 2014.
- [32] S. Ghazanfari-Rad and F. Labeau, "Optimal variable step-size diffusion LMS algorithms," in *Proc. IEEE Workshop on Statistical Signal Processing (SSP)*, Jun.-Jul. 2014, pp. 504–507.
- [33] I. Demirkol, C. Ersoy, and F. Alagoz, "MAC protocols for wireless sensor networks: a survey," *IEEE Commun. Mag.*, vol. 44, no. 4, pp. 115–121, 2006.
- [34] I. Rhee, A. Warrior, M. Aia, J. Min, and M. Sichituu, "Z-MAC: A hybrid MAC for wireless sensor networks," *IEEE/ACM Trans. Netw.*, vol. 16, no. 3, pp. 511–524, 2008.
- [35] A. Bachir, M. Dohler, T. Watteyne, and K. Leung, "MAC essentials for wireless sensor networks," *IEEE Communications Surveys Tutorials*, vol. 12, no. 2, pp. 222–248, 2010.
- [36] *IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications*. IEEE Computer Society/Local and Metropolitan Area Networks, 1999.
- [37] *IEEE 802.15.4 Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-rate Wireless Personal Area Networks (WPANs)*. IEEE Std. 802.15.4, 2006.

- [38] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 3, pp. 535–547, 2000.
- [39] V. Blondel, J. Hendrickx, A. Olshevsky, and J. Tsitsiklis, "Convergence in multiagent coordination, consensus, and flocking," in *Proc. IEEE Conference on Decision and Control (CDC)*, 2005, pp. 2996–3000.
- [40] D. Scherber and H. Papadopoulos, "Locally constructed algorithms for distributed computations in ad-hoc networks," in *Proc. Int. Conf. Information Processing in Sensor Networks (ISPN)*, 2004, pp. 11–19.
- [41] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Syst. Control Lett.*, vol. 53, no. 1, pp. 65–78, Sep. 2004.
- [42] X. Zhao and A. H. Sayed, "Performance limits for distributed estimation over LMS adaptive networks," *IEEE Trans. on Signal Process.*, vol. 60, no. 10, pp. 5107–5124, Oct. 2012.
- [43] A. H. Sayed, "Diffusion adaptation over networks," in *E-Reference Signal Processing*, R. Chellapa and S. Theodoridis, editors, Elsevier 2013.
- [44] S. Haykin, *Adaptive Filter Theory*. 4th ed. Englewood Cliffs, NJ: Prentice-Hall, 2002.
- [45] A. H. Sayed, *Adaptive Filters*. New York: Wiley, 2008.
- [46] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*. New York: McGraw-Hill, 1991.



**Saeed Ghazanfari-Rad** Biography text here.



**Fabrice Labeau** Biography text here.