Formulation and Steady-state Analysis of LMS Adaptive Networks for Distributed Estimation in the Presence of Transmission Errors

Saeed Ghazanfari-Rad and Fabrice Labeau

Abstract

This article presents the formulation and steady-state analysis of the 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 the diffusion LMS algorithm in terms of the steady-state error. The model is then validated by performing Monte Carlo simulations. It is shown that local and global MSD curves 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. Moreover, issues such as scalability in the sense of network size and regressor size, spatially correlated observations, as well as the effect of the distribution of the noise variance are studied.

While the proposed theoretical framework is general in the sense that it is not confined to a particular source of error during information diffusion, for practical reasons we additionally study a specific scenario where errors occur at the medium access control (MAC) level. We develop a model to quantify the MAC-level transmission errors according to the network topology and system parameters for a set of nodes employing a backoff procedure to access the channel. To overcome the problem of unreliable data exchange, we propose an enhanced combining rule that can be deployed in order to improve the performance of diffusion estimation algorithms by using the knowledge of the properties of the transmission errors.

Index Terms

Adaptive networks, diffusion LMS algorithm, MAC layer, distributed estimation, distributed signal processing.

The authors are with the Department of Electrical and Computer Engineering, McGill University, Montréal, QC H3A 0E9, Canada. E-mail: {saeed.ghazanfarirad@mail.mcgill.ca; fabrice.labeau@mcgill.ca}.

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.

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-Quebec/McGill Industrial Research Chair in Interactive Information Infrastructure for the Power Grid.
I. INTRODUCTION

Deployment of wireless sensor networks (WSNs) as monitoring and diagnostic systems is receiving significant attention in recent years because of their clear 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 such as smart grid, precision agriculture, intelligent transportation systems, disaster relief management, radar and target detection and tracking, etc, would benefit from highly reliable and flexible monitoring and diagnostic system that rapidly respond to the changing conditions as well as instantaneous faults. Thus, 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 [1]–[3]. Based on the topology of the network, different distributed algorithms can be implemented [4]–[7]. 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 [5], [6], [8], [9]. Furthermore, we concentrate on the combine-then-adapt (CTA) diffusion algorithm. 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 [5], [10]–[12] and later its modified versions appeared in [7], [13], [14].

There exists previous literature for performance analysis of diffusion algorithms, but all these studies consider an ideal and error-free transmissions [5]–[7]. Some studies have already considered the diffusion algorithms with noisy information exchanges [15], [16]. Theses 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 the 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.

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. 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 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 \cite{17}. We notice that ideal error-free scenarios as studied previously \cite{4}–\cite{7} can be considered as a special case of the formulation in this paper. Meanwhile 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 \cite{4}–\cite{7}) which might lead to computational problems for large $N$ and/or $M$. Furthermore, we do not impose the constraint of spatially independent regressors across the distributed 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 Section [IV] we aim at improving the performance of diffusion estimation algorithms by introducing a combining policy that accounts for transmission errors. 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 [V]. Finally, we conclude the paper in Section [VI].

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

A. Problem Formulation

Consider a distributed network with a set of nodes $\mathcal{N} = \{1, 2, \ldots, N\}$ and a predefined topology including $L$ links. In a typical wireless deployment, a link $(k, l), k, l \in \mathcal{N}$ exists between two nodes if and only if the physical distance between the end nodes is less than the maximum radio transmission range. Let $\mathcal{L} = \{(k, l)\}, k, l \in \mathcal{N}$ 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 \( N_k \). 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 \( S_{k,i} \subseteq N \) 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:

\[
\mathbf{w}_{k,i+1} = \phi_{k,i} + \mu_k e_{k,i} \mathbf{u}_{k,i}, \quad i \geq 0
\]  

(1)

where

\[
\phi_{k,i} = \sum_{l \in S_{k,i}} a_{k,l,i} \mathbf{w}_{l,i}
\]

(2)

\[
e_{k,i} = d_{k,i} - \phi_{k,i}^T \mathbf{u}_{k,i}
\]

(3)

\[
d_{k,i} = v_{k,i}^o + \mathbf{w}_o^T \mathbf{u}_{k,i}
\]

(4)

where \( \mathbf{w}_{k,i}, k = 1, 2, \ldots, N, \ i \geq 0 \) 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 \( k \), \( a_{k,l,i}, l = 1, 2, \ldots, N \) 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 \( l \) to node \( k \) at iteration \( i \) with probability \( p_{k,l} \). 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. Furthermore it is reasonable to assume that transmission errors that occur with probabilities \( p_{k,l} \) 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. In order to give insights to how to quantify transmission errors, we provide a model in Section \( \text{[IV]} \) to assess \( p_{k,l} \) when concentrating on the MAC-level errors.
We notice that unlike the previous formulation (ideal transmission) [5]–[7], 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, we use a different way to express $\phi_{k,i}$ as follows:

$$
\phi_{k,i} = a_{k,k,i}w_{k,i} + \sum_{l \in \mathcal{N}_k \setminus \{k\}} a_{k,l,i}w_{l,i} - \sum_{l \in \mathcal{N}_k \setminus \{k\}} \delta_{k,l,i} a_{k,l,i}w_{l,i}
$$

(5)

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

$$
\delta_{k,l,i} := \begin{cases} 
1 & \text{with probability } p_{k,l} \\
0 & \text{with probability } 1 - p_{k,l}
\end{cases}
$$

(6)

The interpretation of the last term on the right hand side of (5) is to eliminate the local weight vectors of those nodes that have not been able to successfully transmit their information to node $k$. The compact form of (5) is the expression already stated in (2) which builds $\phi_{k,i}$ using a weighted sum of the local weight vectors over $\mathcal{S}_{k,i}$. In order to incorporate the transmission errors in the formulation of diffusion algorithm, we subsequently define some useful notation. Define $\Lambda = [\lambda_{kl}], k, l \in \mathcal{N}$ as the $N \times N$ adjacency matrix representing the network connectivity, i.e., each entry $\lambda_{kl}$ is 1 if there is a link between nodes $k$ and $l$ 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 $\mathcal{E} = \{e_1, \ldots, e_V\}$, their corresponding probabilities $\mathcal{P} = \{p_1, \ldots, p_V\}$, where $V = 2^E$ and the set of combining matrices $\mathcal{A} = \{\mathcal{A}_1, \ldots, \mathcal{A}_V\}$. We also introduce the set $\mathcal{V} = \{1, \ldots, V\}$ whose $j^{th}$ element corresponds to the occurrence of event $e_j, j = 1, \ldots, V$. We note that $\mathcal{A}_j = [a_{k,l,j}], j = 1, \ldots, V$ is the $N \times N$ combining matrix that collects the nonnegative mixing coefficients of diffusion update followed by event $e_j$ during information exchange period satisfying

$$
\sum_{l \in \mathcal{N}_k} a_{k,l,j} = 1, \quad k \in \mathcal{N}, \quad \text{for all } j.
$$

(7)

It follows from the above diffusion algorithm that each entry $a_{k,l,j}$ of matrix $\mathcal{A}_j$ represents the weight given to node $l$ in order to find the intermediate variable at node $k$ conditioned that event $e_j$ occurred during information sharing. As an example, suppose that $e_1$ represents the event in which all transmissions fail due to congested links. Under the independent errors assumption, the probability associated to this case is $p_1 = \prod_{(k,l) \in \mathcal{L}} p_{k,l}$ and it follows that $\mathcal{A}_1 = \mathcal{I}_N$, where $\mathcal{I}_N$ is the $N \times N$ identity matrix, i.e., for this iteration 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 $u_{k,i}$ are temporally independent identically distributed (i.i.d.) zero-mean white Gaussian random variables with covariance
matrices $\mathcal{R}_{u_k} = E[u_ku_k^T] = \sigma_{u_k}^2 I_M$. However, we explore the spatial correlation between nodes by assuming that two locally observed vectors $u_k$ and $u_l$ are correlated Gaussian random vectors with cross-correlation matrix $\mathcal{R}_{u_k,u_l} = \sigma_{u_ku_l}^2 I_M$, where $\sigma_{u_ku_l}^2 = \rho_{kl}\sigma_{u_k}\sigma_{u_l}$ and $\rho_{kl}$ is the spatial correlation index between nodes $k$ and $l$. 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 $u_{k,i}$ 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 [5]. 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). Using (1)-(4) and conditioned that transmission errors correspond to $e_j$, we can obtain a recursive expression to calculate $E[w_{k,i} | e_j]$ as follows

$$E[w_{k,i+1} | e_j] = a_{k,k,j}E[w_{k,i} | e_j] + \rho_k \sum_{l \in \mathcal{N}} a_{k,l,j}E[w_{l,i} | e_j] + c_k,$$

where

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

$$c_k = \mu_k\sigma_{u_k}^2 w^0, \quad k \in \mathcal{N}.$$  

(9)

In order to find $E[w_{k,i+1}]$, we consider all possibilities according to set $\mathcal{E}$ and replace (8) in the following equation

$$E[w_{k,i+1}] = \sum_{j \in \mathcal{V}} p_j E[w_{k,i+1} | e_j],$$

(11)

which yields

$$E[w_{k,i+1}] = a_{k,k}E[w_{k,i}] + \rho_k \sum_{l \in \mathcal{N}} a_{k,l}E[w_{l,i}] + c_k,$$

(12)

where

$$a_{k,k} = \sum_{j \in \mathcal{V}} p_j a_{k,k,j}, \quad a_{k,l} = \sum_{j \in \mathcal{V}} p_j a_{k,l,j}.$$  

(13)
for all \( k, l \in \mathcal{N}, l \neq k \). It follows from (7) that
\[
\sum_{l \in \mathcal{N}_k} a_{k,l} = 1, \ k \in \mathcal{N}.
\] (14)

From Appendix [A] we conclude that
\[
E[w_{k,s}] = w^0, \ \ k = 1, 2, \ldots, N,
\] (15)
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 [A] 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 (62), it is easily verified that \( F(z) \) is the characteristic polynomial of \( \mathcal{E}'_{n,s} = [\rho_i a_{i,j}]_{N \times N}, \ i, j = 1, 2, \ldots, N \) and it is immediate that the roots of \( F(z) \) are the eigenvalues of the square matrix \( \mathcal{E}'_{n,s} \). Let \( \lambda_k, \ k = 1, 2, \ldots, N \) 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 (14), we notice that one interesting feature of the rows of \( \mathcal{E}'_{n,s} \) is that
\[
\sum_{l \in \mathcal{N}} a_{k,l} = \rho_k, \ \ k = 1, 2, \ldots, N.
\] (16)
Consider the induced infinity-norm of matrix \( \mathcal{E}'_{n,s} \) defined as
\[
\|\mathcal{E}'_{n,s}\|_{\infty} = \max_{1 \leq k \leq N} \sum_{l \in \mathcal{N}} |a_{k,l}|.
\] (17)
It is also known from the characteristics of a matrix norm that
\[
\rho(\mathcal{E}'_{n,s}) \leq \|\mathcal{E}'_{n,s}\|.
\] (18)
To satisfy \( \max_{1 \leq k \leq N} |\lambda_k| < 1 \), we use (16)-(18) to express the following condition:
\[
|\rho_k| < 1, \ \ k = 1, 2, \ldots, N,
\] (19)
which is equivalent to impose the following lower and upper bounds on the step-sizes

\[ 0 < \mu_k < \frac{2}{\sigma^2 u_k}, \quad k = 1, 2, \ldots, N. \] (20)

Consequently, the important result can be stated as follows. In a network including \( N \) nodes deploying distributed diffusion estimation algorithm (1)-(4) with combining weight matrices satisfying (7), in the presence of multiplicative transmission errors modeled as (6), the sufficient condition for mean stability is provided by (20). It is important to note that (6) coincides with sufficient condition for mean stability when transmissions are assumed to be perfect [7]. In other words, transmission errors modeled as (6) 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:

\[
\text{MSD}_k = \lim_{i \to \infty} E[(w_{k,i} - \omega^o)^T(w_{k,i} - \omega^o)].
\] (21)

It is shown in Appendix [B] that we can write the following expression for \( E[w_{k,i+1}^T w_{l,i+1}|e_j] \):

\[
E[w_{k,i+1}^T w_{l,i+1}|e_j] = \eta_{kl} \left\{ \sum_{m \in \mathcal{N}} a_{k,m,j} a_{l,m,j} E[w_{m,i}^T w_{m,i}] + \sum_{m,n \in \mathcal{N}, m \neq n} (a_{k,m,j} a_{l,n,j} + a_{k,n,j} a_{l,m,j}) E[w_{m,i}^T w_{n,i}] \right\}
+ \sum_{m \in \mathcal{N}} (a_{k,m,j}(\epsilon_l - \nu_{kl}) + a_{l,m,j}(\epsilon_k - \nu_{kl})) \omega^o E[w_{m,i}] + \nu_{kl} \omega^o T \omega^o,
\] (22)

where

\[
\eta_{kl} = 1 - (\mu_k \sigma^2 u_k + \mu_l \sigma^2 u_l) + \mu_k \mu_l (\sigma^2 u_k \sigma^2 u_l + (M + 1) \sigma^4 u_k u_l),
\] (23)

\[
\epsilon_k = \mu_k \sigma^2 u_k,
\] (24)

\[
\nu_{kl} = \mu_k \mu_l (\sigma^2 u_k \sigma^2 u_l + (M + 1) \sigma^4 u_k u_l).
\] (25)

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

\[
E[w_{k,i+1}^T w_{l,i+1}] = \sum_{j \in \mathcal{V}} p_j E[w_{k,i+1}^T w_{l,i+1}|e_j].
\] (26)

Using (22) and (26), it follows that:

\[
E[w_{k,i+1}^T w_{l,i+1}] = \sum_{m \in \mathcal{N}} c_{kl,mm} E[w_{m,i}^T w_{m,i}] + \sum_{m,n \in \mathcal{N}, m \neq n} c_{kl,mm} E[w_{m,i}^T w_{n,i}]
\]
\[
+ \sum_{m \in \mathcal{N}} c_{kl,om} \mathbf{w}^o \mathbf{E}[\mathbf{w}_{m,i}] + \nu_{kl} \mathbf{w}^o \mathbf{w}^o,
\]

(27)

where

\[ c_{kl,mm} = \eta_{kl} \sum_{j \in \mathcal{V}} p_j \mathbf{a}_{k,m,j} \mathbf{a}_{l,m,j}, \]

(28)

\[ c_{kl,mn} = \eta_{kl} \sum_{j \in \mathcal{V}} p_j (\mathbf{a}_{k,m,j} \mathbf{a}_{l,n,j} + \mathbf{a}_{l,n,j} \mathbf{a}_{l,m,j}), \]

(29)

\[ c_{kl,om} = \sum_{j \in \mathcal{V}} p_j [\mathbf{a}_{k,m,j} (\epsilon_l - \nu_{kl}) + \mathbf{a}_{l,m,j} (\epsilon_k - \nu_{kl})], \]

(30)

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

\[
\mathbf{E}[\mathbf{w}_{k,i+1}^T \mathbf{w}_{k,i+1}] = \sum_{m \in \mathcal{N}} c_{kk,mm} \mathbf{E}[\mathbf{w}_{m,i}^T \mathbf{w}_{m,i}] + \sum_{m,n \in \mathcal{N}} c_{kk,mn} \mathbf{E}[\mathbf{w}_{m,i}^T \mathbf{w}_{n,i}]
\]

\[ + \sum_{m \in \mathcal{N}} c_{kk,om} \mathbf{w}^o \mathbf{E}[\mathbf{w}_{m,i}] + \nu_k \mathbf{w}^o \mathbf{w}^o + M \mu_k^2 \sigma_{u_k}^2 \sigma_{v_k}^2, \]

(31)

where

\[ c_{kk,mm} = \eta_k \sum_{j \in \mathcal{V}} p_j \mathbf{a}_{k,m,j}^2, \]

(32)

\[ c_{kk,mn} = 2 \eta_k \sum_{j \in \mathcal{V}} p_j \mathbf{a}_{k,m,j} \mathbf{a}_{k,n,j}, \ m \neq n, \]

(33)

\[ c_{kk,om} = 2 \sum_{j \in \mathcal{V}} p_j \mathbf{a}_{k,m,j} (\epsilon_k - \nu_k), \]

(34)

\[ c_{v_k} = M \mu_k^2 \sigma_{u_k}^2 \sigma_{v_k}^2, \]

(35)

and:

\[ \eta_k = 1 - 2 \mu_k \sigma_{u_k}^2 + \mu_k^2 (M + 2) \sigma_{u_k}^4, \]

(36)

\[ \nu_k = \mu_k^2 (M + 2) \sigma_{u_k}^4. \]

(37)

Let \( \mathcal{W}_{kl} \) denote the one-sided \( z \)-transform of \( \mathbf{E}[\mathbf{w}_{k,i}^T \mathbf{w}_{l,i}] \). Taking the \( z \)-transform of both sides of (27) and (31) and after some algebra we obtain

\[
(1 - z^{-1} c_{kk,kk}) \mathbf{W}_{kk} - z^{-1} \sum_{m \in \mathcal{N}, m \neq k} c_{kk,mm} \mathbf{W}_{mm} - z^{-1} \sum_{m \in \mathcal{N}, m \neq k} c_{kk,mm} \mathbf{W}_{mm} = z^{-1} \sum_{m \in \mathcal{N}} c_{kk,om} \mathbf{w}^o \mathbf{W}_{m} + \frac{\nu_k \mathbf{w}^o \mathbf{w}^o + c_{v_k}}{1 - z^{-1}},
\]

(38)
\[(1 - z^{-1}c_{kl,kl})W_{kl} - z^{-1} \sum_{m \in \mathcal{N}} c_{kl,mm}W_{mm} - z^{-1} \sum_{m,n \in \mathcal{N}, m \neq k, n \neq l} c_{kl,mm}W_{mm} - \sum_{m,n \in \mathcal{N}, m \neq k, n \neq l} c_{kl,mm}W_{mm} = z^{-1} \sum_{m \in \mathcal{N}} c_{kl,om}w_m^oT\mathcal{W}_m + \frac{\nu_{kl}w_m^oT\mathcal{W}_m}{1 - z^{-1}}. \quad (39)\]

Note that (38) and (39) 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 \(Q = \frac{N(N+1)}{2}\). 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 \(\{11, \ldots, NN, 12, \ldots, 1N, 23, \ldots, 2N, \ldots, N - 2, N - 1, N - 2, N, N - 1, N\}\), the system description in \(z\)-domain can be represented as follows:

\[
\begin{bmatrix}
1 - z^{-1}c_{11,11} & -z^{-1}c_{11,22} & \cdots & -z^{-1}c_{11,N-1}N \\
-z^{-1}c_{22,11} & 1 - z^{-1}c_{22,22} & \cdots & -z^{-1}c_{22,N-1}N \\
\vdots & \vdots & \ddots & \vdots \\
-z^{-1}c_{N-1,N,11} & -z^{-1}c_{N-1,N,22} & \cdots & 1 - z^{-1}c_{N-1,N,N-1}N \\
\end{bmatrix}
\begin{bmatrix}
W_{11} \\
W_{22} \\
\vdots \\
W_{N-1,N} \\
\end{bmatrix}
= \begin{bmatrix}
\sum_{m \in \mathcal{N}} c_{11,om}w_m^oT\mathcal{W}_m + \frac{\nu_1w_m^oT\mathcal{W}_m + \nu_1}{1 - z^{-1}} \\
\sum_{m \in \mathcal{N}} c_{22,om}w_m^oT\mathcal{W}_m + \frac{\nu_2w_m^oT\mathcal{W}_m + \nu_2}{1 - z^{-1}} \\
\vdots \\
\sum_{m \in \mathcal{N}} c_{N-1,N,om}w_m^oT\mathcal{W}_m + \frac{\nu_{N-1}w_m^oT\mathcal{W}_m + \nu_{N-1}}{1 - z^{-1}} \\
\end{bmatrix} \begin{bmatrix}
\mathcal{C} \\
\mathcal{W} \\
\end{bmatrix}_{Q \times 1}
\]

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

\[
[\mathcal{C}_1 \mathcal{C}_2 \ldots \mathcal{C}_Q]_{Q \times Q} [\mathcal{W}]_{Q \times 1} = [\mathcal{D}]_{Q \times 1}. \quad (41)
\]

Recall that our objective is to find \(\mathcal{W}_{kk}\), \(k = 1, 2, \ldots, N\), i.e. the first \(N\) elements of \(\mathcal{W}\) from the set of equations described in (40). Let \(\mathcal{C}_D_i\), \(i = 1, 2, \ldots, N\) 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:

\[
\mathcal{W}_{kk} = \frac{\text{det}(\mathcal{C}_D_k)}{\text{det}(\mathcal{C})}, \quad k = 1, 2, \ldots, N. \quad (42)
\]
We can rewrite matrix $D$ as follows:

$$
D = \begin{bmatrix}
z^{-1} \sum_{m \in \mathcal{N}} c_{11,om} w_{o,m}^{T} W_{m} + \frac{v_{1} w_{o,m}^{T} w_{o}}{1-z^{-1}} \\
\vdots \\
z^{-1} \sum_{m \in \mathcal{N}} c_{N,om} w_{o,m}^{T} W_{m} + \frac{v_{N-1} w_{o,m}^{T} w_{o}}{1-z^{-1}} \\
\end{bmatrix}
\begin{bmatrix}
[D_{wo}] \\
[D_{v}] 
\end{bmatrix} + \begin{bmatrix}
c_{v_1} \\
\vdots \\
c_{v_N-1,N} \\
\end{bmatrix}
$$

(43)

It follows that

$$
\det(C_{Di}) = |C_1 \cdots D_{wo} C_Q| + |C_1 \cdots D_v C_Q|.
$$

(44)

Using (44) in (42), we obtain the following expression

$$
W_{kk} = \frac{|C_1 \cdots D_{wo} C_Q|}{|C|} + \frac{|C_1 \cdots D_v C_Q|}{|C|},
$$

(45)

for all $k \in \mathcal{N}$. To proceed, we define the square matrix $C'$ of size $Q$ as $C' = [c_{kl,mn}]_{Q \times Q}$ with $c_{kl,mn}$ obtained from (28),(29),(32) and (33) and arranged as in $C$ in (40). To simplify, we define $c_s$, $c_{w^o,s}$ and $c_{v_k,s}$, $k = 1, 2, \ldots, N$ at the steady-state as follows:

$$
c_{v_k,s} = \lim_{z \to 1} (z - 1) \frac{|C_1 \cdots D_v C_Q|}{|C|} = \begin{vmatrix}
1 - c_{11,11} & -c_{11,22} & \cdots & -c_{11,N-1,N} \\
-c_{22,11} & 1 - c_{22,22} & \cdots & -c_{22,N-1,N} \\
\vdots & \vdots & \ddots & \vdots \\
-c_{N-1,N,11} & -c_{N-1,N,22} & \cdots & 1 - c_{N-1,N,N-1,N}
\end{vmatrix},
$$

(46)

$$
c_s = \lim_{z \to 1} \frac{|C|}{|C_N - C'|},
$$

(47)

$$
c_{w^o,s} = \lim_{z \to 1} (z - 1) \frac{|C_1 \cdots D_{wo} C_Q|}{|C|} = \frac{w_{o}^{T} w_{o}}{1-z^{-1}} |C|.
$$

(48)

Notice that the derivation of (48) is given in Appendix C. 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 (21). Then, using the final value theorem, $W_{kk}$ as described in (45) and the steady-state quantities as defined in (46) and (47), we arrive at the following
result:
\[ w_{k,s} \triangleq \lim_{i \to \infty} E[w_k^T w_k] = \lim_{z \to 1} (z - 1)W_{kk} = w^{oT}w^o + \frac{c_{vk,s}}{c_s}. \] (49)

Finally, the closed-form expression for local MSD \( k \) is given by
\[ \text{MSD}_k = \frac{c_{vk,s}}{c_s}. \] (50)

We notice that the global network MSD is obtained by averaging over the local MSD’s as follows:
\[ \text{MSD} = \frac{1}{N} \sum_{k \in \mathcal{N}} \text{MSD}_k. \] (51)

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 (40) around \(|z| \to 1\), we must provide condition to prevent any root of \( \det(C) = 0 \) 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 \( C' \). For convenience, we rewrite matrix \( C' \) as \( C' = [\gamma_{i,j}]_{Q \times Q} \) and define \( \eta'_i, i = 1, 2, \ldots, Q \) to denote the common factor in (36) and \( \eta_{kl} \) in and (23) of elements in the \( i^{th} \) row of \( C' \). Let us define \( \lambda'_k, k = 1, 2, \ldots, Q \) to refer to the eigenvalues of \( C' \). Using (14), (28), (29), (32) and (33) and noting the structure of \( C' \), we find one important feature of the rows of \( C' \)
\[ \sum_{j=1}^{Q} \gamma_{i,j} = \eta'_i, \quad i = 1, 2, \ldots, Q. \] (52)

Using this result and considering the induced infinity-norm of matrix \( C' \) 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 \(|\eta'_i| < 1, i = 1, 2, \ldots, Q\), or equivalently
\[ |\eta_k| < 1, \quad k \in \mathcal{N}, \quad \text{and} \quad |\eta_{kl}| < 1, \quad k, l \in \mathcal{N}, k \neq l. \] (53)

It is worth mentioning that (36) 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 (23), it is evident that \( \eta_{kl} \) 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 (53) provides useful and practically applicable lower and upper bounds for step sizes which can be written as follows:

\[0 < \mu_k < \frac{2}{(M + 2)\sigma^2_{u_k}}, \quad k = 1, 2, \ldots, N.\] (54)

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 (7), in the presence of multiplicative transmission errors modeled as (6), the sufficient condition provided by (54) ensures stability in the mean-square sense. An important feature of the condition (54) is that it is not dependent on error probabilities \(p_{k,l}\). This suggests that transmission errors do not lead to the mean-square divergence of diffusion estimation algorithms.

It is worth mentioning that condition (54) represents a novel bound 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 [7]. Sufficiently small step-sizes may however be a conservative approach which leads 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. All these reveal the importance of the upper bound of step-sizes provided by (54). Another practical significance of (54) is that the condition is fully distributed, i.e, each node can locally select its step-size according to the statistics of the filter input.

**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 \(p_{k,l}\) 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 [18]–[20] 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 [21] and the contention phase of the IEEE 802.15.4 [22] 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 selected between 0 and the contention window. The initial size of the contention
window is $CW$ and is doubled at each retransmission. Let $R$ denote the maximum number of retransmissions.
Then, the maximum contention window size is $CW_{\text{max}} = 2^R CW$. 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 $q_{k,l} \ k \in \mathcal{N}, \ l \in \mathcal{N}_k$ denote the probability of collision assigned to the transmissions with node $k$
as destination and node $l$ 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 $l$ 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 $q_{k,l}$ can be written as follows:

$$q_{k,l} = 1 - (1 - \tau)^{n_k} \quad k \in \mathcal{N}, \ l \in \mathcal{N}_k. \quad (55)$$

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 $p_{k,l}$ as follows [23]:

$$\tau = \frac{2(1 - 2q_{k,l})}{(1 - 2q_{k,l})(CW + 1) + q_{k,l} CW (1 - (2q_{k,l})^R)} \quad (56)$$

Solving (55) and (56), 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 $p_{k,l}$. 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 $p_{k,l}$ on all
directional links to node $k$ are identical and equivalent to the packet loss experienced by node $k$ which is denoted
by $q_k$ for convenience. More precisely, we have

$$p_{k,l} = q_k^{n_k + 1} = q_k, \quad k \in \mathcal{N}, l \in \mathcal{N}_k. \quad (57)$$

IV. PERFORMANCE IMPROVEMENT

Throughout this section, we briefly review different combining rules and then discuss how to use the knowledge
of the properties of transmission errors in order to improve the performance of diffusion algorithms. Different
methods for combining the received information from neighboring nodes can be divided into three groups. The first
group includes the methods that solely rely on the information regarding each node’s degree such as relative degree
uniform \([24]\), Laplacian \([25, 26]\) and Metropolis \([26]\), as well as the maximum degree method \([8]\) which only uses the information regarding the total number of nodes in the network. We notice that in these methods no information regarding the noise \((v_{k,i} \text{ in } (4))\) variances across the network is required. The second group includes the rules that only takes into account the noise levels across the network such as the relative variance method. In the relative variance method, larger weight is assigned to the node with smaller noise variance. Finally, the third group includes the methods that require information regarding both the nodes’s degree and noise levels such as relative degree variance rule \([7]\) and Hasting rule \([27]\). If any of the methods in the last two groups are used, and unlike the first group, there should be mechanisms to estimate the local noise variance and then distribute such information across the network. This extra cost will however result in lower MSD compared to the methods of the first group. The combining rules constructed only in terms of the degrees of the nodes are considerably more efficient in networks where the noise levels experienced by the neighboring nodes are in the same range. In such networks, it is rational to assign more weight to the nodes with larger degrees. In particular, the relative degree policy constructs the mixing coefficients in the following manner:

\[
a_{k,l,i} = \frac{n_l}{\sum_{m \in S_{k,i}} n_m}, \quad l \in S_{k,i}, \quad \text{(relative degree rule)}
\]

where \(n_l\) is the cardinality of \(N_l\). In the presence of transmission errors, it is reasonable to modify the relative degree rule to account for errors. It is evident that such errors affect the connectivity of the network and in particular the degrees of nodes at each iteration of the adaptive estimation algorithm. The most natural idea is to take into account the transmission errors to define the true degree of each node. Particularly, we obtain the effective degree of each node denoted by \(n'_k\) with multiplying its degree when no error occurs by its average probability of successful reception:

\[
n'_k = n_k (1 - q_k), \quad k \in N.
\]

(58)

In general, the average error probability \(q_k\) can be calculated at each node by counting the successfully received information packets over a period. The average error probability when concentrating on the MAC-level errors as described in Section \([III]\) is the average packet loss probability experienced by each node which can be locally estimated over a short period and based on the number of the received ACKs. It can also for instance be computed according to the model we already considered for the backoff procedure in Section \([III]\). The new combining rule which we refer to as the enhanced relative degree method can be expressed as follows:

\[
a_{k,l,i} = \frac{n'_l}{\sum_{m \in S_{k,i}} n'_m}, \quad l \in S_{k,i}, \quad \text{(enhanced relative degree rule)}.
\]
V. 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 (55)-(57) 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., $p_{k,l} = p, (k,l) \in \mathcal{L}$. 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 [7]. Hence, it follows that $a_{k,l,i} = \frac{\sigma_{v,l}^{-2}}{\sum_{m \in S_{k,i}} \sigma_{v,m}^{-2}}$ for $l \in S_{k,i}$ and otherwise we have that $a_{k,l,i} = 0$. Notice that any rule for finding the mixing coefficients is possible as long as it satisfies the condition $\sum_{l \in N} a_{k,l,j} = 1, \ k = 1, 2, \ldots, N$ for all $j$, as discussed in Section II-A. Each node has access to the distorted and noisy version of the same unknown vector $\mathbf{w}^o = \text{col}\{1,1,\ldots,1\}/\sqrt{M}$, with $M = 200$ according to (4). The $M$-dimensional input regressors are assumed to be temporally independent Gaussian, but spatially correlated. The spatial correlation index $\rho_{kl}$

![Fig. 1: Network topology for 7-node network. The node index is shown next to each node.](image-url)
between two nodes $k$ and $l$ is obtained according to $\rho_{kl} = \rho^{|k-l|}$, 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 $[10^{-6}, 10^{-2}]$ 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., $\mu_k = \mu = 0.001$. In order to obtain the performance measures, the results are averaged over 150 independent experiments each using 100,000 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., $p \in \{0, 0.3, 0.8, 0.9, 1\}$. We observe that when the transmissions experience a high error, i.e., $p = 0.9$, the global network MSD at the steady-state increases. Another observation is that the convergence speed of the diffusion estimation algorithm decreases as the
transmission unreliability considerably increases, i.e., \( p = 0.9 \). However, the global convergence rate might increase when the error probability increases. In other words, transmission errors might prevent 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. 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 MSD curves as a function of error probability are illustrated in Figs. 4-6. We observe that
Fig. 6: Local steady-state MSD (dB) for node 7 as a function of error probability in 7-node network. The solid line curves show the theoretical expression (50) and the markers represent the simulation results.

MSD curves obtained from analysis and simulation coincide with each other. It is seen that the MSD curves for a group of nodes \{2, 3, 4, 5\} are monotonically increasing functions of the error probability; on the contrary, for other nodes \{1, 6, 7\} there exist non-zero values of probability of error that minimizes the corresponding local MSD. This is due to the fact that the noise variances of nodes \{2, 3, 4, 5\} are significantly higher than those of \{1, 6, 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 curves are not necessarily monotonically increasing functions of the error probability. Fig. 7 shows the network global MSD curve for two different values of \(\rho\).

Fig. 7: Global steady-state MSD in dB as a function of error probability in 7-node network for different values of spatial correlation index when \(\rho \in \{0, 0.9\}\). The solid line curves show the theoretical expression (51) and the markers represent the simulation results.

Note that in both scenarios, the minimum value of the steady-state network MSD is not obtained at \(p = 0\) which confirms the non-increasing behavior of the global MSD curve. We observe that for both scenarios the minimum MSD is obtained at \(p = 0.06\) and the impact of the spatially correlated observation on the global steady-state MSD is negligible. In Table I we sort the nodes in an ascending order in terms of the local MSD. We observe that while all nodes achieve a lower MSD compared to non-cooperative mode by employing diffusion algorithm, the order of nodes changes depending on which region the network operates in, in terms of the transmission errors. In particular,
for $p = 0.9$, node 5 achieves a better performance compared to node 2. We can get insight into this behavior by noticing that the degree of node 5 is 4 while the degree of node 2 is 1. Consequently, even a small amount of information flow in the case of high error probability namely, $p = 0.9$ significantly improves the performance of node 5. Meanwhile, as the error probability decreases, i.e., $p \to 0$, the degree of a node will not be a dominant factor for local performance improvement and node 2 achieves a lower MSD compared to node 5 due to its lower noise variance.

TABLE I: node number according to the increasing order of local MSD

| Error probability $p$ | node number |
|-----------------------|-------------|
| 1                     | 7 1 6 2 5 3 4 |
| 0.9                   | 7 1 6 5 2 4 3 |
| 0.5                   | 7 6 1 5 2 4 3 |
| 0.1                   | 7 6 1 2 5 4 3 |
| 0                     | 7 2 6 1 5 4 3 |

B. MAC-collision Based Errors

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

In the second phase of simulations, we concentrate on the MAC-level errors. 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. Initially, we perform a discrete-event simulation with $R = 1$ and $CW = 3$ to quantify error probabilities $q_k$ which is then verified using model (55)-(57). Fig. 9 (bottom) shows the profile of the error probability across the network. It is seen that nodes located in dense areas...
experience more errors than those in scattered regions. We now simulate the diffusion estimation algorithm for the corresponding MAC level errors shown at the bottom of Fig. 9. Noise levels are generated randomly from $[10^{-5}, 10^{-2}]$ and depicted in Fig. 9 (top). We assume that the variances of input regressors are identical across nodes, i.e., $\sigma^2_u = \sigma^2_{u_k} = 0.5$. We select $M = 200$ and identical step-sizes: $\mu = \mu_k = 0.01$.

We apply several combining policies that require only the degrees of nodes such as relative degree, Laplacian, Metropolis, maximum degree and enhanced relative degree. Learning curves in terms of the global MSD are shown in Fig. 10. We observe that the diffusion estimation algorithm achieves the best performance in terms of the steady-state global MSD when using the enhanced relative degree rule.
VI. 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 curves 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 introduce the enhanced relative degree to mitigate the negative effects of such errors.

APPENDIX A

DERIVATION OF (15)

Define the one-sided $z$-transform of the weight vector $w_k$ as $W_k = \mathcal{Z}\{E[w_k]\}$. Taking the $z$-transform of (12), we arrive at

$$
(1 - a_{k,k}\rho_k z^{-1})W_k - z^{-1} \sum_{l \in N, l \neq k} a_{k,l}W_l = \frac{c_k}{1 - z^{-1}}. 
$$

Thus, we can write the set of equations (60) as follows:

$$
\begin{bmatrix}
(1 - z^{-1}a_{1,1}\rho_1)I_M & -z^{-1}a_{1,2}\rho_1I_M & \cdots & -z^{-1}a_{1,N}\rho_1I_M \\
-z^{-1}a_{2,1}\rho_2I_M & (1 - z^{-1}a_{2,2}\rho_2)I_M & \cdots & -z^{-1}a_{2,N}\rho_2I_M \\
\vdots & \vdots & \ddots & \vdots \\
-z^{-1}a_{N,1}\rho_NI_M & -z^{-1}a_{N,2}\rho_NI_M & \cdots & (1 - z^{-1}a_{N,N}\rho_N)I_M \\
\end{bmatrix}
\begin{bmatrix}
W_1 \\
W_2 \\
\vdots \\
W_N \\
\end{bmatrix}
= 
\begin{bmatrix}
\frac{c_1}{1 - z^{-1}} \\
\frac{c_2}{1 - z^{-1}} \\
\vdots \\
\frac{c_N}{1 - z^{-1}} \\
\end{bmatrix}.
$$

(60)

For convenience, we define matrix $\mathcal{E}_{n,s}$ to collect $a_{k,l}$ and $\rho_k$, $k, l \in N$ as $\mathcal{E}_{n,s} = [a_{k,l}\rho_k]_{N \times N}$. Using the Cramer’s rule we obtain the following expression for $W_k$

$$
W_k = \frac{\det(\mathcal{E}_{n,s})}{\det(\mathcal{E}_n)} w^o, \quad k = 1, 2, \ldots, N,
$$

(61)
where

\[ E_n = I_N - z^{-1} E'_{n,s}, \]  

(62)

and \( E_{n,F_i} \) is obtained by replacing the \( i^{th} \) column of \( E_n \) by \( F_n \) defined as

\[ F_n = \frac{1}{1 - z^{-1}} \left[ \mu_1 \sigma^2_{u_1}, \mu_2 \sigma^2_{u_2}, \ldots, \mu_N \sigma^2_{u_N} \right]^T. \]  

(63)

Denote \( E[w_k,s] \) as the expectation of the weight vector of the \( k^{th} \) node at the steady-state. Then, we yield

\[ E[w_k,s] = \det(E_{n,F_i}) \cdot \det(E_{n,s})^{-1}, \quad k = 1, 2, \ldots, N, \]  

(64)

where \( E_{n,s} = I_N - E'_{n,s} \) and \( E_{n,s,F_i} \) is obtained by replacing the \( i^{th} \) column of \( E_{n,s} \) by the \( N \times 1 \) column vector \( F_{n,s} \) whose \( k^{th} \) element is \( \mu_k \sigma^2_{u_k} \). 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

\[ \begin{vmatrix} C_1 & C_2 & \cdots & C_i & \cdots & C_Q \end{vmatrix} = \begin{vmatrix} C_1 & C_2 & \cdots & C'_i & \cdots & C_Q \end{vmatrix}, \]  

(65)

where \( C'_i = \sum_{j=1}^Q C_j \), for all \( i = 1, 2, \ldots, Q \). Let us denote each column of \( E_{n,s} \) by \( E_j \) and find the sum of the columns of \( E_{n,s} \) as follows:

\[
\sum_{j \in N} E_j = \left[ 1 - \rho_1 \sum_{l \in N} a_{1,l} \ldots 1 - \rho_N \sum_{l \in N} a_{N,l} \right]^T
\]

\[
= \left[ 1 - \rho_1 \ldots 1 - \rho_N \right]^T = F_{n,s}.
\]  

(66)

Notice that in the first and second step, we used (14) and (9) respectively. Hence, replacing any column of the \( E_{n,s} \) by \( F_{n,s} \) does not alter the value of the determinant of \( E_{n,s} \). The expressions in (15) is then derived from (64).

**APPENDIX B**

**DERIVATION OF (22)**

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

\[
w_{k,i+1} = a_{k,k,i}w_{k,i} + \sum_{l \in N \setminus \{k\}} a_{k,l,i}w_{l,i} + \mu_k v_{k,i}u_{k,i} + \mu_k u_{k,i}^T w^o
\]

\[
- a_{k,k,i}\mu_k u_{k,i}^T w_{k,i} - \sum_{l \in N \setminus \{k\}} a_{k,l,i}\mu_k u_{k,i}^T w_{l,i}.
\]  

(67)
Followed by multiplying recursion (67) with correct indexes and taking the conditional expectation given that event $e_j$ occurs during data transmission, we arrive at (22).

Note that in order to compute expressions of the form $E[w_{m,i}^T u_i^T u_h^T u_h^T w_n]$, 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 [28], [29]. 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 [30]. With these in mind and omitting the time index $i$ for simplicity, a term of the form $E[w_{m,i}^T u_i^T u_h^T u_h^T w_n]$ can be written as:

$$E[w_{m,i}^T u_i^T u_h^T u_h^T w_n] = E\left[ \sum_k w_{m,k} u_{i,k} u_{h,j} u_{h,i} w_{n,i} \right]$$

$$= \sum_{k,j,i} E[w_{m,k} u_{i,k} u_{h,j} u_{h,i} w_{n,i}]$$

$$= \sum_{k,j,i} E[w_{m,k} w_{n,i}] E[u_{i,k} u_{h,j} u_{h,i} w_{n,i}], \quad (68)$$

where the notations $w_{m,k}$, $k = 1, \ldots, M$ and $u_{i,k}$, $k = 1, \ldots, M$ are used to represent the $k^{th}$ elements of vectors $w_m$ and $u_m$ respectively. Notice that in the last step of Equation (68), the independence assumption is used. Using the Gaussian moment factoring theorem, it can be verified that

$$E[w_{m,i}^T u_i^T u_h^T u_h^T w_n] = \sum_i E[w_{m,i} w_{n,i}] \left( \sigma_u^2 \sigma_{u_h}^2 + 2 \sigma_{u_h}^4 \right) + \sum_i E[w_{m,i} w_{n,i}] (M - 1) \sigma_{u_h}^4$$

$$= \left( \sigma_u^2 \sigma_{u_h}^2 + (M + 1) \sigma_{u_h}^4 \right) \sum_i E[w_{m,i} w_{n,i}]$$

$$= \left( \sigma_u^2 \sigma_{u_h}^2 + (M + 1) \sigma_{u_h}^4 \right) E[w_{m,i}^T w_{n,i}], \quad (69)$$

**APPENDIX C**

**DERIVATION OF (49)**

Using (43), we rewrite $D_w$ as follows:

$$[D_w]_i = z^{-1} \sum_{m \in N} c_{j,om} w_o^T W_m + \nu_j w_o^T w_o \frac{w_o}{1 - z^{-1}}, \quad i = 1, \ldots, Q, \quad (70)$$

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 (40). The key is to prove that

$$\lim_{z \to 1} (z - 1) [D_w]_i = w_o^T w_o \sum_{m \in N} c_{j,om} + \nu_j, \quad (71)$$
for all \( i = 1, 2, \ldots, Q \). Denote by \( E_{n,i} \) the \( i^{th} \) column of \( E_n \). Then it holds that \( E'_n = \sum_{i \in N} E_{n,i} = [1 - z^{-1} \rho_j]_j \).

Using \( \rho_j = 1 - \mu_j \sigma^2_{u_j} \) and (63), we obtain:

\[
[E'_n]_j = 1 - z^{-1}(1 - \mu_j \sigma^2_{u_j}) \\
= 1 - z^{-1}(1 - (1 - z^{-1})[F_n]_j) \\
= (1 - z^{-1})(1 + z^{-1}[F_n]_j).
\]

Using (72) we obtain:

\[
[E'_n]_j = 1 - z^{-1} \left(1 - (1 - z^{-1})[F_n]_j\right) \\
= (1 - z^{-1})(1 + z^{-1}[F_n]_j).
\]

We know that

\[
\det(E_n) = \begin{vmatrix}
E_{n,1} & \cdots & E'_n & \cdots & E_{n,N} \\
\end{vmatrix}
\]

\[
= (1 - z^{-1}) \begin{vmatrix}
E_{n,1} & \cdots & 1 + z^{-1}F_n & \cdots & E_{n,N} \\
\end{vmatrix}
\]

\[
= (1 - z^{-1}) \left( \begin{vmatrix}
E_{n,1} & \cdots & 1 & \cdots & E_{n,N} \\
\end{vmatrix} + z^{-1} \begin{vmatrix}
E_{n,1} & \cdots & F_n & \cdots & E_{n,N} \\
\end{vmatrix} \right)
\]

\[
= (1 - z^{-1}) \left( \begin{vmatrix}
E_{n,1} & \cdots & 1 & \cdots & E_{n,N} \\
\end{vmatrix} + z^{-1} \begin{vmatrix}
E_{n,1} & \cdots & E_{n,N} \\
F_n & \cdots & E_{n,N} \\
\end{vmatrix} \right)
\]

Therefore (61) can be written as follows:

\[
W_i = \frac{w^o}{(1 - z^{-1})\zeta(z) + z^{-1}(1 - z^{-1})},
\]

where

\[
\zeta(z) = \frac{\det(E_{n,1} \cdots 1 \cdots E_{n,N})}{\det(E_n, F_i)}.
\]

With these, (70) becomes

\[
[D_{w^o}]_i = \frac{w^{oT}w^o}{1 - z^{-1}} \left[ z^{-1} \sum_{m \in N} \frac{c_{j,om}}{\zeta(z) + z^{-1}} + \nu_j \right],
\]

and we get

\[
\lim_{z \to 1} (z - 1)[D_{w^o}]_i = \lim_{z \to 1} z w^{oT}w^o \left[ z^{-1} \sum_{m \in N} \frac{c_{j,om}}{\zeta(z) + z^{-1}} + \nu_j \right].
\]

Note that

\[
\det(E_n, F_i) = \frac{1}{1 - z^{-1}} \det(E_{n,1} \cdots [\mu_j \sigma^2_{u_j}]_{i \in N} \cdots E_{n,N}).
\]

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

\[
\zeta(z) = \frac{(1 - z^{-1})\det(E_{n,1} \cdots 1 \cdots E_{n,N})}{\det(E_{n,1} \cdots [\mu_j \sigma^2_{u_j}]_{i \in N} \cdots E_{n,N})}.
\]
and we arrive at (71) under one of the following conditions: either if we have that \([\mu_i \sigma_u^2]_{i \in \mathcal{N}} = \mu \sigma_u^2\) or we have that det(\(E_{n,1} \cdots [\mu_i \sigma_u^2]_{i \in \mathcal{N}} \cdots E_{n,N}\)) \(\neq 0\). In order to proceed, we use (71) to obtain the following expression for \(c_{w^n,s}\)

\[
c_{w^n,s} = \mathbf{w}^{o^T} \mathbf{w}^0 \begin{bmatrix} C_1 & C_2 & \cdots & D_s & C_Q \end{bmatrix},
\]

(78)

where \([D_s]_i = \sum_{m \in \mathcal{N}} c_{j,om} + \nu_j, i = 1, \ldots, Q\). After some algebra and using (30) and (34) to replace \(c_{kl,om}, l, m \in \mathcal{N}\) and then using \(\sum_{l \in \mathcal{N}} a_{kl,l} = 1, k = 1, 2, \ldots, N\) we obtain \([D_s]_i = 1 - \epsilon_j, i = 1, \ldots, Q\). Using (23), (24), (25), (36) and (37) we arrive at

\[
2\epsilon_k - \nu_k = 1 - \eta_k, \quad k = 1, 2, \ldots, N,
\]

(79)

\[
\epsilon_k + \epsilon_l - \nu_{kl} = 1 - \eta_{kl}, \quad k, l \in \mathcal{N}, k \neq l.
\]

(80)

With these, the sum of all columns of \(I_N - \mathcal{C}'\) according to (47) becomes:

\[
\sum_{j=1}^{Q} C_j = \begin{bmatrix} 1 - \sum_{k,l \in \mathcal{N}} c_{11,kl} \cdots 1 - \sum_{k,l \in \mathcal{N}} c_{N-1,N,kl} \end{bmatrix}^T.
\]

(81)

Using (28), (29), (32) and (33), it can be checked that

\[
\sum_{j=1}^{Q} C_j = \begin{bmatrix} 1 - \eta_1 \sum_{j \in \mathcal{V}} p_j (\sum_{m \in \mathcal{N}} a_{1,m,j})^2 \cdots \\
1 - \eta_{N-1,N} \sum_{j \in \mathcal{V}} p_j (\sum_{m \in \mathcal{N}} a_{N-1,m,j} \sum_{n \in \mathcal{N}} a_{N,n,j})^T \\
1 - \eta_1 \cdots 1 - \eta_{N-1,N} \end{bmatrix}^T = D_s.
\]

(82)

Recalling the determinant property as stated in (65), we conclude that replacing any of the columns of \(I_N - \mathcal{C}'\) by \(D_s\) does not modify the value of its determinant, i.e., \(|\mathcal{C}_1 \cdots D_s \cdots \mathcal{C}_Q| = |\mathcal{C}|\). As a result, equality (48) holds.

REFERENCES

[1] D. Li, K. Wong, Y. H. Hu, and A. Sayeed, “Detection, classification, and tracking of targets,” IEEE Signal Processing Magazine, vol. 19, no. 2, pp. 17 –29, Mar. 2002.

[2] D. Culler, D. Estrin, and M. Srivastava, “Guest editors’ introduction: Overview of sensor networks,” Computer, vol. 37, no. 8, pp. 41 – 49, Aug. 2004.

[3] M. Rabbat and R. Nowak, “Quantized incremental algorithms for distributed optimization,” IEEE Journal on Selected Areas in Communications, vol. 23, no. 4, pp. 798 – 808, Apr. 2005.

[4] 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.

[5] 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.
[6] 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.

[7] 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.

[8] L. Xiao, S. Boyd, and S. Lall, “A scheme for robust distributed sensor fusion based on average consensus,” in Proc. of Fourth ISPN, vol. 4, Apr. 2005, pp. 63 –70, Los Angeles, UCLA.

[9] ——, “A space-time diffusion scheme for peer-to-peer least-squares estimation,” in Proc. of Fourth ISPN, Apr. 2006, pp. 168 –176, Nashville, TN.

[10] A. H. Sayed and C. G. Lopes, “Distributed processing over adaptive networks,” in 9th International Symposium on Signal Processing and Its Applications, ISSPA 2007., Feb. 2007, pp. 1 –3.

[11] C. G. Lopes and A. H. Sayed, “Diffusion least-mean squares over adaptive networks,” in IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2007., vol. 3, Apr. 2007, pp. III–917 –III–920.

[12] ——, “Steady-state performance of adaptive diffusion least-mean squares,” in IEEE/SP 14th Workshop on Statistical Signal Processing, SSP ‘07., Aug. 2007, pp. 136 –140.

[13] F. S. Cattivelli and A. H. Sayed, “Diffusion LMS algorithms with information exchange,” in 42nd Asilomar Conference on Signals, Systems and Computers, 2008, Oct. 2008, pp. 251 –255.

[14] ——, “Diffusion strategies for distributed kalman filtering and smoothing,” IEEE Transactions on Automatic Control, vol. 55, no. 9, pp. 2069 –2084, Sep. 2010.

[15] 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.

[16] 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.

[17] S. Ghazanfari-Rad and F. Labeau, “Diffusion least-mean squares over distributed networks in the presence of MAC errors,” in Asilomar Conf. Signals, Syst., Comput., Nov. 2012, pp. 1787–1791.

[18] I. Demirkol, C. Ersoy, and F. Alagoz, “MAC protocols for wireless sensor networks: a survey,” IEEE Communications Magazine, vol. 44, no. 4, pp. 115–121, 2006.

[19] I. Rhee, A. Warrier, M. Aia, J. Min, and M. Sichitiu, “Z-MAC: A hybrid MAC for wireless sensor networks,” IEEE/ACM Transactions on Networking., vol. 16, no. 3, pp. 511–524, 2008.

[20] 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.

[21] IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications. IEEE Computer Society/Local and Metropolitan Area Networks, 1999.

[22] 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.

[23] G. Bianchi, “Performance analysis of the IEEE 802.11 distributed coordination function,” IEEE Journal on Selected Areas in Communications., vol. 18, no. 3, pp. 535–547, 2000.

[24] V. Blondel, J. Hendrickx, A. Olshevsky, and J. Tsitsiklis, “Convergence in multiagent coordination, consensus, and flocking,” in 44th IEEE Conference on Decision and Control, European Control Conference., 2005, pp. 2996–3000.

[25] D. Scherber and H. Papadopoulos, “Locally constructed algorithms for distributed computations in ad-hoc networks,” in Third International Symposium on Information Processing in Sensor Networks., 2004, pp. 11–19.

[26] L. Xiao and S. Boyd, “Fast linear iterations for distributed averaging,” Syst. Control Lett., vol. 53, no. 1, pp. 65 –78, Sep. 2004.
[27] 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.

[28] S. Haykin, *Adaptive Filter Theory*. 4th ed. Englewood Cliffs, NJ: Prentice-Hall, 2002.

[29] A. H. Sayed, *Adaptive Filters*. New York: Wiley, 2008.

[30] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*. New York: McGraw-Hill, 1991.