# Steady-State Behavior of General Complex-Valued Diffusion LMS Strategies

Sithan Kanna, *Student Member, IEEE*, and Danilo P. Mandic, *Fellow, IEEE*

***Abstract*—A novel methodology to bound the steady-state mean square performance of the diffusion complex least mean square (D-CLMS) and the diffusion widely linear (augmented) CLMS (D-ACLMS) algorithm is proposed. This is achieved by exploiting the almost identical nature of the steady-state filter weights at all nodes. The proposed approach allows for the consideration of the second-order terms in the recursion for the weight error covariance matrix, without compromising the mathematical tractability of the problem. The closed form expressions for the mean square deviation (MSD) and excess mean square error (EMSE) for both the D-CLMS and D-ACLMS allow for the performance of the algorithms to be quantified as a function of the noncircularity of the input data.**

***Index Terms*—Augmented statistics, diffusion adaptation, distributed optimization, least mean square (LMS), widely linear.**

## I. INTRODUCTION

THE use of sensor networks for distributed learning tasks has gained popularity as inexpensive sensors with processing and communication technologies are becoming increasingly available. Among the strategies for distributed learning and optimization, the diffusion strategy has been shown to provide excellent performance while maintaining the structural simplicity of standard adaptive filtering algorithms [1]. Moreover, the diffusion strategy was demonstrated to approach the performance of a centralized implementation, while having lower communication requirements and exhibiting greater robustness to node failures [2].

Although quantifying the performance of diffusion adaptive networks is not straightforward, as the interactions between multiple connected adaptive filters add to the complexity of the analysis, considerable advances have been made in the area [3], [4]. To make the analysis mathematically tractable, current approaches assume that the step-sizes are small enough so that the second-order terms in the analysis can be neglected [4]–[9]. However, this assumption somewhat compromises the steady-state analysis since important characteristics of the performance can be influenced by the second-order terms.

To this end, we propose a new method to bound the mean square performance of an adaptive network while incorporating the second-order terms in the analyses. The proposed bound is based on what we refer to as the "similarity conjecture", which states that, upon convergence, the steady-state filter weights in the network are almost identical. The similarity conjecture is physically meaningful as it exploits the inherent behavior of the diffusion scheme in which the individual filters are adapting toward the same optimal weight, while diffusing their intermediate estimates with neighboring nodes. For a fully connected network, the bound we propose is exact.

The proposed approach is applied to the analysis of the diffusion complex least mean square (D-CLMS) [3] and its widely linear counterpart, the diffusion augmented CLMS (D-ACLMS) [10], [11]. The closed form expressions for the mean square deviation (MSD) and excess mean square error (EMSE) allow us to quantify the steady-state performance of both the D-CLMS and D-ACLMS as a function of the input data noncircularity; this is not possible with the current methods as the noncircularity of the input data is a second-order effect that is neglected if the small step-size theory is used.

*Notation:* Lowercase letters are used to denote scalars $a$ boldface letters for column vectors $\boldsymbol{a}$ and boldface uppercase letters for matrices $\mathbf{A}$. The symbol $(\cdot)^{\mathsf{T}}$ is the transpose operator while $(\cdot)^{\mathsf{H}}$ designates the Hermitian transpose operator. The symbol $\mathbb{E}[\cdot]$ represents the statistical expectation operator and $\otimes$ denotes the Kronecker product. The symbol $(\cdot)^*$ represents complex conjugation. An $M \times M$ identity matrix is denoted by $\mathbf{I}_M$ while $\mathbf{1}_N$ is an $N \times 1$ vector of ones.

## II. BACKGROUND

Consider a collaborative estimation task that is carried out by several nodes in a network of $N$ nodes. A node $i$ is able to send and receive information from its neighboring nodes in the neighborhood $\mathcal{N}_i$ as shown in Fig. 1.

Each node $i$ receives streaming measurement data $\{d_{i,k}, \boldsymbol{x}_{i,k}\}$ at every time instant $k$, where the desired signal $d_{i,k} \in \mathbb{C}$ and input vector $\boldsymbol{x}_{i,k} \in \mathbb{C}^{M\times 1}$ are related via a widely linear model [10]

$$d_{i,k} = \boldsymbol{h}_{\text{opt}}^{\mathsf{H}}\boldsymbol{x}_{i,k} + \boldsymbol{g}_{\text{opt}}^{\mathsf{H}}\boldsymbol{x}_{i,k}^* + \eta_{i,k}, \tag{1}$$

corrupted with zero-mean white noise $\eta_{i,k}$ with variance $\sigma_\eta^2$. Note that the widely linear model in (1) is a generalization of the standard strictly linear model for which $\boldsymbol{g}_{\text{opt}} = \mathbf{0}$. For the compactness of the analysis, we shall represent both the strictly linear and widely linear models using

$$d_{i,k} = \boldsymbol{w}_{\text{opt}}^{\mathsf{H}}\mathbf{z}_{i,k} + \eta_{i,k}, \tag{2}$$

where $\boldsymbol{w}_{\text{opt}} \in \mathbb{C}^{\overline{M}\times 1}$ and $\mathbf{z}_{i,k} \in \mathbb{C}^{\overline{M}\times 1}$ with $\overline{M} = M$ for strictly linear models and $\overline{M} = 2M$ for widely linear models; see Table I for details.

Manuscript received February 03, 2016; revised March 17, 2016; accepted March 18, 2016. Date of publication March 25, 2016; date of current version April 20, 2016. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Magno T. M. Silva.

The authors are with the Department of Electrical and Electronic Engineering, Imperial College London, SW7 2BT London, U.K. (e-mail: ssk08@imperial.ac.uk; d.mandic@imperial.ac.uk).

Color versions of one or more of the figures in this letter are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LSP.2016.2547219

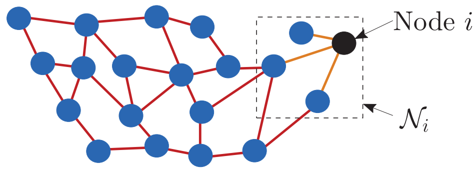


Fig. 1. Distributed network with $N = 20$ nodes.

TABLE I
INPUT AND OPTIMAL WEIGHT VECTORS FOR THE STRICTLY LINEAR AND WIDELY LINEAR MODELS

| Model | $\boldsymbol{w}_{\text{opt}}$ | $\boldsymbol{z}_{i,k}$ | Dimension |
|---|---|---|---|
| Strictly linear | $\boldsymbol{h}_{\text{opt}}$ | $\boldsymbol{x}_{i,k}$ | $M \times 1$ |
| Widely linear | $\begin{bmatrix} \boldsymbol{h}_{\text{opt}} \\ \boldsymbol{g}_{\text{opt}} \end{bmatrix}$ | $\begin{bmatrix} \boldsymbol{x}_{i,k} \\ \boldsymbol{x}^*_{i,k} \end{bmatrix}$ | $2M \times 1$ |

For mathematical tractability, we adopt the usual assumptions that the data $\{d_{i,k}, \mathbf{z}_{i,k}, \eta_{i,k}\}$ are zero-mean and that the following holds [12]:

A1. The input data $\mathbf{z}_{i,k}$ are temporally uncorrelated and independent over space.

A2. The noise process $\eta_{i,k}$ is temporally white and spatially independent with $\sigma^2_\eta = \mathbb{E}\left[|\eta_{i,k}|^2\right]\ \forall i$.

A3. The input vector $\boldsymbol{z}_{i,k}$ and noise process $\eta_{i,k}$ are mutually independent over time and space.

The task of the network is to estimate the unknown parameter vector $\boldsymbol{w}_{\text{opt}}$ from measurement data $\{d_{i,k}, \boldsymbol{x}_{i,k}\}$. At each time instant $k$, the node $i$ estimates $\boldsymbol{w}_{\text{opt}}$ by updating $\boldsymbol{w}_{i,k}$ via the diffusion augmented complex LMS (D-ACLMS) update scheme [10]. The D-ACLMS update is a two-step process whereby at each time instant, the first step is to update the weights using local measurements, while in the second step, the weights from the neighboring nodes are combined. This setting of the algorithm is referred to as the adapt-then-combine (ATC) setting and is given by [12]

$$\text{Adapt: } \boldsymbol{\psi}_{i,k+1} = \boldsymbol{w}_{i,k} + \mu \mathbf{z}_{i,k}(d^*_{i,k} - \mathbf{z}^{\mathsf{H}}_{i,k}\boldsymbol{w}_{i,k}) \tag{3a}$$

$$\text{Combine: } \boldsymbol{w}_{i,k+1} = \sum_{\ell=1}^{N} a_{\ell i}\boldsymbol{\psi}_{\ell,k+1} \tag{3b}$$

where the term $\boldsymbol{\psi}_{i,k+1}$ denotes the intermediate update, $\mu$ is the learning rate that is identical throughout the network, and $a_{\ell i}$ are the weighting coefficients used by node $i$ to scale the weights from its neighbors. The weighting coefficients are positive, $a_{\ell i} \geq 0$, sum up to unity, $\sum_{\ell=1}^{N} a_{\ell i} = 1$, and are zero, $a_{\ell i} = 0$, only if the node $i$ is not connected to node $\ell$ [1]. The weights $a_{\ell i}$ can be combined into an $N \times N$ matrix

$$\mathbf{A} = \begin{bmatrix} a_{11} & \cdots & a_{1N} \\ \vdots & \ddots & \vdots \\ a_{N1} & \cdots & a_{NN} \end{bmatrix}, \quad \mathbf{A} \in \mathbb{R}^{N\times N} \tag{4}$$

where the weights that node $i$ assigns to nodes $\ell \in \mathcal{N}_i$ are in the $i$th column of $\mathbf{A}$. Observe that the matrix $\mathbf{A}$ is left-stochastic since all the columns of the matrix $\mathbf{A}$ sum up to unity. The D-ACLMS algorithm for the network can therefore be expressed as

$$\text{Adapt: } \boldsymbol{\psi}_{k+1} = \boldsymbol{w}_k + \mu \mathbf{Z}_k\left(\boldsymbol{d}^*_k - \mathbf{Z}^{\mathsf{H}}_k \boldsymbol{w}_k\right) \tag{5a}$$

$$\text{Combine: } \boldsymbol{w}_{k+1} = \boldsymbol{\mathcal{A}}^{\mathsf{T}}\boldsymbol{\psi}_{k+1} \tag{5b}$$

where all the filter variables are given by

$$\begin{aligned} \boldsymbol{w}_k &= \mathsf{col}\{\boldsymbol{w}_{1,k}, \ldots, \boldsymbol{w}_{N,k}\}, \quad \boldsymbol{w}_k \in \mathbb{C}^{\overline{M}N\times 1} \\ \boldsymbol{\psi}_k &= \mathsf{col}\{\boldsymbol{\psi}_{1,k}, \ldots, \boldsymbol{\psi}_{N,k}\}, \quad \boldsymbol{\psi}_k \in \mathbb{C}^{\overline{M}N\times 1} \\ \boldsymbol{d}_k &= \mathsf{col}\{d_{1,k}, \ldots, d_{N,k}\}, \quad \boldsymbol{d}_k \in \mathbb{C}^{N\times 1} \\ \mathbf{Z}_k &= \begin{bmatrix} \mathbf{z}_{1,k} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{z}_{2,k} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{z}_{N,k} \end{bmatrix}, \quad \mathbf{Z}_k \in \mathbb{C}^{\overline{M}N\times N} \\ \boldsymbol{\mathcal{A}} &= \mathbf{A} \otimes \mathbf{I}_{\overline{M}}, \quad \boldsymbol{\mathcal{A}} \in \mathbb{R}^{\overline{M}N\times \overline{M}N}. \end{aligned} \tag{6}$$

The $\mathsf{col}\{\cdot\}$ operator creates a column vector of its arguments. This setup applies to both the D-CLMS with $\overline{M} = M$ and D-ACLMS with $\overline{M} = 2M$; see Table I.

## III. Mean Square Analysis of the D-ACLMS

The mean square behavior of the network is analyzed using the weight error covariance matrix [1]

$$\boldsymbol{\mathcal{K}}_k \stackrel{\text{def}}{=} \mathbb{E}\left[\tilde{\boldsymbol{w}}_k \tilde{\boldsymbol{w}}^{\mathsf{H}}_k\right] \tag{7}$$

where $\tilde{\boldsymbol{w}}_k \stackrel{\text{def}}{=} \bar{\boldsymbol{w}}_{\text{opt}} - \boldsymbol{w}_k$ is the network weight error vector and the optimal weight vector for the network is $\bar{\boldsymbol{w}}_{\text{opt}} \stackrel{\text{def}}{=} \mathbf{1}_N \otimes \boldsymbol{w}_{\text{opt}}$ since all the nodes in the network are estimating the same optimal weights $\boldsymbol{w}_{\text{opt}}$.

Subtracting $\bar{\boldsymbol{w}}_{\text{opt}}$ from both sides of (5a) and (5b), using the desired signal model in (2) and recognizing that $\boldsymbol{\mathcal{A}}^{\mathsf{T}}\bar{\boldsymbol{w}}_{\text{opt}} = \bar{\boldsymbol{w}}_{\text{opt}}$ enables the network weight error recursion to be formulated as

$$\tilde{\boldsymbol{w}}_{k+1} = \boldsymbol{\mathcal{A}}^{\mathsf{T}}\left(\tilde{\boldsymbol{w}}_k - \mu\widehat{\boldsymbol{\mathcal{R}}}_k\tilde{\boldsymbol{w}}_k - \mu\boldsymbol{n}_k\right) \tag{8}$$

where $\widehat{\boldsymbol{\mathcal{R}}}_k \stackrel{\text{def}}{=} \mathbf{Z}_k\mathbf{Z}^{\mathsf{H}}_k$ and $\boldsymbol{n}_k \stackrel{\text{def}}{=} \mathbf{Z}_k\boldsymbol{\eta}^*_k$, while $\boldsymbol{\eta}_k = \mathsf{col}\{\eta_{1,k}, \ldots, \eta_{N,k}\}$ is the vector containing measurement noise from all the nodes.

The weight error covariance matrix $\boldsymbol{\mathcal{K}}_k$ is obtained by first post-multiplying both sides of (8) by their Hermitian transpose to give [2]

$$\begin{aligned} \tilde{\boldsymbol{w}}_{k+1}\tilde{\boldsymbol{w}}^{\mathsf{H}}_{k+1} = \boldsymbol{\mathcal{A}}^{\mathsf{T}}\Big(&\tilde{\boldsymbol{w}}_k\tilde{\boldsymbol{w}}^{\mathsf{H}}_k - \mu\tilde{\boldsymbol{w}}_k\tilde{\boldsymbol{w}}^{\mathsf{H}}_k\widehat{\boldsymbol{\mathcal{R}}}_k - \mu\widehat{\boldsymbol{\mathcal{R}}}_k\tilde{\boldsymbol{w}}_k\tilde{\boldsymbol{w}}^{\mathsf{H}}_k \\ &+ \mu^2\,\widehat{\boldsymbol{\mathcal{R}}}_k\tilde{\boldsymbol{w}}_k\tilde{\boldsymbol{w}}^{\mathsf{H}}_k\widehat{\boldsymbol{\mathcal{R}}}_k + \mu^2\boldsymbol{n}_k\boldsymbol{n}^{\mathsf{H}}_k\Big)\boldsymbol{\mathcal{A}}. \end{aligned} \tag{9}$$

Then, upon applying the statistical expectation operator $\mathbb{E}[\cdot]$ to (9), and employing the assumptions A1– A3, we have

$$\begin{aligned} \boldsymbol{\mathcal{K}}_{k+1} = \boldsymbol{\mathcal{A}}^{\mathsf{T}}\Big(&\boldsymbol{\mathcal{K}}_k - \mu\boldsymbol{\mathcal{K}}_k\boldsymbol{\mathcal{R}} - \mu\boldsymbol{\mathcal{R}}\boldsymbol{\mathcal{K}}_k + \mu^2\boldsymbol{\mathcal{R}}\boldsymbol{\mathcal{K}}_k\boldsymbol{\mathcal{R}} \\ &+ \mu^2\boldsymbol{\mathcal{P}}\bar{\boldsymbol{\mathcal{K}}}^{\mathsf{T}}_k\boldsymbol{\mathcal{P}} + \mu^2\boldsymbol{\mathcal{T}}_k\boldsymbol{\mathcal{R}} + \mu^2\sigma^2_\eta\boldsymbol{\mathcal{R}}\Big)\boldsymbol{\mathcal{A}} \end{aligned} \tag{10}$$

where apart from the weight error covariance matrix $\boldsymbol{\mathcal{K}}_k$ and the combination matrix $\boldsymbol{\mathcal{A}}$, all the other matrices are block

[1] The weight error covariance matrix is a block matrix in which the $ij$th block is given by $[\boldsymbol{\mathcal{K}}_k]_{\boldsymbol{ij}} \stackrel{\text{def}}{=} \mathbf{K}_{ij,k} = \mathbb{E}\left[\tilde{\boldsymbol{w}}_{i,k}\tilde{\boldsymbol{w}}^{\mathsf{H}}_{j,k}\right]$.

[2] Cross-terms are ignored.

diagonal. Notice that the network input data statistics are given by both the covariance and pseudocovariance matrices, respectively, denoted by $\boldsymbol{\mathcal{R}} = \mathbb{E}\left[\mathbf{Z}_k\mathbf{Z}_k^{\mathsf{H}}\right]$ and $\boldsymbol{\mathcal{P}} = \mathbb{E}\left[\mathbf{Z}_k\mathbf{Z}_k^{\mathsf{T}}\right]$. It was also necessary to define $\bar{\boldsymbol{\mathcal{K}}}_k$ and $\boldsymbol{\mathcal{T}}_k$, which resulted from analyzing the block-matrix $\mathbb{E}\left[\widehat{\boldsymbol{\mathcal{R}}}_k\tilde{\boldsymbol{w}}_k\tilde{\boldsymbol{w}}_k^{\mathsf{H}}\widehat{\boldsymbol{\mathcal{R}}}_k\right]$ in (9) where the $ij$th block is given by

$$\mathbb{E}\left[\widehat{\boldsymbol{\mathcal{R}}}_k\tilde{\boldsymbol{w}}_k\tilde{\boldsymbol{w}}_k^{\mathsf{H}}\widehat{\boldsymbol{\mathcal{R}}}_k\right]_{\boldsymbol{ij}} = \mathbb{E}\left[\mathbf{z}_{i,k}\mathbf{z}_{i,k}^{\mathsf{H}}\tilde{\boldsymbol{w}}_{i,k}\tilde{\boldsymbol{w}}_{j,k}^{\mathsf{H}}\mathbf{z}_{j,k}\mathbf{z}_{j,k}^{\mathsf{H}}\right]. \quad (11)$$

Using assumption A1, and the well-known Gaussian moment factorizing theorem, the expression in (11) becomes

$$\begin{cases}\mathbf{R}_i\mathbf{K}_{i,k}\mathbf{R}_i + \mathbf{P}_i\mathbf{K}_{i,k}^{\mathsf{T}}\mathbf{P}_i^* + \mathbf{R}_i\mathsf{Tr}\{\mathbf{R}_i\mathbf{K}_{i,k}\}, & i = j\\ \mathbf{R}_i\mathbf{K}_{ij,k}\mathbf{R}_j, & i \neq j,\end{cases}$$

where $\mathbf{R}_i = \mathbb{E}\left[\mathbf{z}_{i,k}\mathbf{z}_{i,k}^{\mathsf{H}}\right]$ and $\mathbf{P}_i = \mathbb{E}\left[\mathbf{z}_{i,k}\mathbf{z}_{i,k}^{\mathsf{T}}\right]$ are, respectively, the covariance and pseudocovariance matrices of the input vector at node $i$. This results in $\bar{\boldsymbol{\mathcal{K}}}_k = \mathsf{bdiag}\left\{\mathbf{K}_{1,k},\ldots,\mathbf{K}_{N,k}\right\}$ and the term $\boldsymbol{\mathcal{T}}_k = \mathsf{bdiag}$ $\{\mathsf{Tr}\left\{\mathbf{R}_1\mathbf{K}_{1,k}\right\}\mathbf{I}_{\overline{M}},\ldots,\mathsf{Tr}\left\{\mathbf{R}_N\mathbf{K}_{N,k}\right\}\mathbf{I}_{\overline{M}}\}$, where the operator $\mathsf{bdiag}\{\cdot\}$ creates a block diagonal matrix.

## IV. Similarity Conjecture

To enable second-order analysis of the MSE, we introduce a "similarity conjecture," which uses the fact that at the steady-state, the weight vectors at each node are almost identical; this can be formalized as

$$\lim_{k\to\infty}\tilde{\boldsymbol{w}}_k = \mathbf{1}_N \otimes \tilde{\boldsymbol{w}}_{ss}, \quad \tilde{\boldsymbol{w}}_{ss} \in \mathbb{C}^{\overline{M}\times 1}. \quad (12)$$

The similarity conjecture is realistic as the "combine step" (5a) reduces the difference between filter weight vectors via averaging and the "adapt step" in (5b) forces the filter weights to evolve toward the optimal value $\boldsymbol{w}_{\text{opt}}$, which is identical throughout the network. A consequence of the similarity conjecture which will be exploited in the analysis is

$$\lim_{k\to\infty}\boldsymbol{\mathcal{K}}_k = \mathbf{1}_N\mathbf{1}_N^{\mathsf{T}} \otimes \mathbf{K}_{ss}, \quad \mathbf{K}_{ss} \in \mathbb{C}^{\overline{M}\times\overline{M}}. \quad (13)$$

*Remark 1:* For a fully connected network (equivalent to a centralized system), the approximations in (12) and (13) are exact not only at the steady-state but at every iteration.

*Proof:* For a fully connected network, the combination matrix becomes $\mathbf{A} = \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^{\mathsf{T}}$, [13] therefore from the combine step in (5b)

$$\tilde{\boldsymbol{w}}_k = \boldsymbol{\mathcal{A}}^{\mathsf{T}}\tilde{\boldsymbol{\psi}}_k = \mathbf{1}_N \otimes \tilde{\boldsymbol{\psi}}_{\text{av},k},$$

where $\tilde{\boldsymbol{\psi}}_{\text{av},k} = \frac{1}{N}\sum_{\ell=1}^N\tilde{\boldsymbol{\psi}}_{\ell,k}$. This implies that $\boldsymbol{\mathcal{K}}_k = \mathbf{1}_N\mathbf{1}_N^{\mathsf{T}} \otimes \mathbb{E}\left[\tilde{\boldsymbol{\psi}}_{\text{av},k}\tilde{\boldsymbol{\psi}}_{\text{av},k}^{\mathsf{H}}\right]$, thereby completing the proof.

*Remark 2:* For all other topologies, the similarity assumption in (13) acts as an upper bound for the mean square performance. This is because (13) can be interpreted as replacing the weight error cross-covariance matrices between the different nodes $\mathbf{K}_{ij,k}$ with the weight error autocovariance matrix $\mathbf{K}_{ii,k}$, which acts as an upper bound to $\mathbf{K}_{ij,k}$.

### A. Single Filter Representation

Finally, incorporating the standard assumption that the network has uniform data statistics, i.e., $\mathbf{R}_i = \mathbf{R}\ \forall i$ and $\mathbf{P}_i = \mathbf{P}\ \forall j$, together with the similarity assumption in (13) enables the terms in the network weight error covariance recursion in (10) to be reduced to $\boldsymbol{\mathcal{R}} = \mathbf{I}_N \otimes \mathbf{R}$, $\boldsymbol{\mathcal{P}} = \mathbf{I}_N \otimes \mathbf{P}$, $\bar{\boldsymbol{\mathcal{K}}}_k = \mathbf{I}_N \otimes \mathbf{K}_k$, and $\boldsymbol{\mathcal{T}}_k = \mathsf{Tr}\left\{\mathbf{R}\mathbf{K}_k\right\}\mathbf{I}_{N\overline{M}}$.

Taking the average of the matrices in the block diagonal entries of (10) as $\mathbf{K}_k \overset{\text{def}}{=} \frac{1}{N}\sum_{i=1}^N\left[\boldsymbol{\mathcal{K}}_k\right]_{\boldsymbol{ii}}$ now gives the evolution of the average weight error covariance matrix in the network in the form

$$\mathbf{K}_{k+1} = \mathbf{K}_k - \mu\mathbf{K}_k\mathbf{R} - \mu\mathbf{R}\mathbf{K}_k + \mu^2\mathbf{R}\mathbf{K}_k\mathbf{R} + \alpha\mu^2\left(\mathbf{P}\mathbf{K}_k^{\mathsf{T}}\mathbf{P}^* + \mathsf{Tr}\left\{\mathbf{R}\mathbf{K}_k\right\}\mathbf{R} + \sigma_\eta^2\mathbf{R}\right), \quad (14)$$

where $\alpha = \mathsf{Tr}\left\{\mathbf{A}^{\mathsf{T}}\mathbf{A}\right\}/N$ is the factor that captures the effect of the diffusion strategy.

*Remark 3:* The factor $\alpha \leq 1$ acts as a contractive term, which reduces the mean square error along the iterations.

*Proof:* Since $a_{\ell i} \geq 0$ and $\sum_\ell a_{\ell i} = 1$, it can be shown that

$$\alpha = \frac{1}{N}\mathsf{Tr}\left\{\mathbf{A}^{\mathsf{T}}\mathbf{A}\right\} = \frac{1}{N}\sum_i\sum_\ell a_{\ell i}^2 \leq \frac{1}{N}\sum_i\left(\sum_\ell a_{\ell i}\right)^2 = 1. \quad (15)$$

It is important to note that employing the similarity assumption and uniform data statistics transforms the network mean square performance recursion in (10) into a representation in (14), which resembles a classical single filter weight error covariance recursion [14]–[16]. Although the diffusion network has been analyzed from a single-node perspective in the past [17], [18], the proposed similarity conjecture simplifies the analysis by giving a means to deal with cross-nodal weight error covariance terms as explained in Remark 2.

### B. Steady-State Mean Square Analysis

Next, the weight error recursion in (14) can be diagonalized using the approximate uncorrelating transform (AUT) [19], which diagonalizes both the covariance and pseudocovariance matrices with a single eigenvector matrix $\bar{\mathbf{Q}}$.

Pre- and post-multiplying (14) by $\bar{\mathbf{Q}}^{\mathsf{H}}$ and $\bar{\mathbf{Q}}$, respectively, gives rise to the recursion for the diagonalized weight error covariance matrix $\tilde{\mathbf{K}}_k \overset{\text{def}}{=} \bar{\mathbf{Q}}^{\mathsf{H}}\mathbf{K}_k\bar{\mathbf{Q}}$ in the form [20]

$$\tilde{\mathbf{K}}_{k+1} = \tilde{\mathbf{K}}_k - \mu\left(\boldsymbol{\Lambda}_{\mathbf{r}}\tilde{\mathbf{K}}_k + \tilde{\mathbf{K}}_k\boldsymbol{\Lambda}_{\mathbf{r}}\right) + \mu^2\boldsymbol{\Lambda}_{\mathbf{r}}\tilde{\mathbf{K}}_k\boldsymbol{\Lambda}_{\mathbf{r}} + \alpha\mu^2\left(\boldsymbol{\Lambda}_{\mathbf{p}}\tilde{\mathbf{K}}_k^{\mathsf{T}}\boldsymbol{\Lambda}_{\mathbf{p}} + \mathsf{Tr}\left\{\boldsymbol{\Lambda}_{\mathbf{r}}\tilde{\mathbf{K}}_k\right\}\boldsymbol{\Lambda}_{\mathbf{r}} + \sigma_\eta^2\boldsymbol{\Lambda}_{\mathbf{r}}\right), \quad (16)$$

where $\boldsymbol{\Lambda}_{\mathbf{r}} \approx \bar{\mathbf{Q}}^{\mathsf{H}}\mathbf{R}\bar{\mathbf{Q}}$ and $\boldsymbol{\Lambda}_{\mathbf{p}} = \bar{\mathbf{Q}}^{\mathsf{H}}\mathbf{P}\bar{\mathbf{Q}}$. Upon combining the diagonal terms of $\tilde{\mathbf{K}}_k$ into a vector $\boldsymbol{\kappa}_k = [\tilde{K}_{k,11}, \tilde{K}_{k,22},\ldots,\tilde{K}_{k,\overline{MM}}]^{\mathsf{T}}$, we arrive at the recursion

$$\boldsymbol{\kappa}_{k+1} = \left[(\mathbf{I} - \mu\boldsymbol{\Lambda}_{\mathbf{r}})^2 + \alpha\mu^2(\boldsymbol{\Lambda}_{\mathbf{p}}^2 + \boldsymbol{\lambda}\boldsymbol{\lambda}^{\mathsf{T}})\right]\boldsymbol{\kappa}_k + \alpha\mu^2\sigma_\eta^2\boldsymbol{\lambda}, \quad (17)$$

where $\boldsymbol{\lambda} = \boldsymbol{\Lambda}_{\mathbf{r}} \mathbf{1}_{\overline{M}}$ contains the eigenvalues of the covariance matrix $\mathbf{R}$. In the steady-state, $\boldsymbol{\kappa}_{k+1} = \boldsymbol{\kappa}_k = \boldsymbol{\kappa}_\infty$, yielding

$$\boldsymbol{\kappa}_\infty = \left[2\boldsymbol{\Lambda}_{\mathbf{r}} - \mu(\boldsymbol{\Lambda}_{\mathbf{r}}^2 + \alpha\boldsymbol{\Lambda}_{\mathbf{p}}^2 + \alpha\boldsymbol{\lambda}\boldsymbol{\lambda}^{\mathsf{T}})\right]^{-1} \alpha\mu\sigma_\eta^2\boldsymbol{\lambda}. \quad (18)$$

The average MSD and excess mean square error (EMSE) for the filters in the network are defined as [1], [21]

$$\mathrm{MSD}_{\mathrm{av}} \stackrel{\mathrm{def}}{=} \mathbf{1}^{\mathsf{T}}\boldsymbol{\kappa}_\infty, \quad \mathrm{EMSE}_{\mathrm{av}} \stackrel{\mathrm{def}}{=} \boldsymbol{\lambda}^{\mathsf{T}}\boldsymbol{\kappa}_\infty. \quad (19)$$

*Remark 4:* The closed form expressions in (19) have much simpler forms compared to the existing results. As stated in Remark 2, the so obtained MSD and EMSE expressions are exact for a fully connected network and represent an upper bound on the performance for other topologies.

## V. Simulations

In the following simulations, the proposed MSD expression in (19), denoted by "New" was benchmarked against the current mean square analysis [1] denoted by "Old" and Monte-Carlo simulations for MSD values for a system identification task in a network of 20 nodes. The system was an finite impulse response (FIR) channel with weight vector $\boldsymbol{h}_{\mathrm{opt}} = [-0.2, 0.8]^{\mathsf{T}}$. Unless stated otherwise, the input $\boldsymbol{x}_k$ was a zero-mean Gaussian process with covariance matrix $\mathbf{R} = \mathbf{I}_M$ and pseudocovariance matrix $\mathbf{P} = \mathbf{0}$ and the step-size was $\mu = 0.1$. The Metropolis rule was chosen as the combination rule where the combination coefficients were given by

$$a_{\ell i} = \begin{cases} 1/\max\{|\mathcal{N}_\ell|, |\mathcal{N}_i|\}, & \text{for } \ell \neq i \\ 1 - \sum_{\ell \neq i} a_{\ell i}, & \text{for } \ell = i. \end{cases} \quad (20)$$

Note that the simulations were repeated with other well-known combination rules like the uniform, Laplacian, relative degree and maximum degree rules [8], and the relationship between the MSD and the level of connectivity, step-sizes and input circularity were identical. We therefore have chosen to present the results with the Metropolis rule.

*Case Study 1: Varying levels of connectivity.* In the first set of simulations, the MSD was evaluated at different levels of connectivity within the network. For each level of connectivity (i.e., number of connections in the network), the performance was averaged over an ensemble of 5000 randomly generated network topologies. A single realization of a 20-node network with 30 connections is shown in Fig. 1. Fig. 2 shows that as discussed in Remark 2, the theoretical MSD proposed in (19) was closest to the actual MSDs obtained from the Monte-Carlo simulation in a fully connected network because it mimics the centralized implementation where the filter weights in all the nodes are identical. For networks with only a few connections, the theoretical MSD from (19) acts as an upper bound for the algorithm performance.

*Case Study 2: Varying the step-sizes.* Fig. 3 shows the MSD values for a fully connected 20-node network with different step-sizes. For small step-sizes, as expected, the existing expression for the MSD matches the simulated values. However, for larger step-sizes, the proposed MSD better models the simulated values.

*Case Study 3: Effects of noncircularity.* In the third set of simulations (as shown in Fig. 4), the input signal was

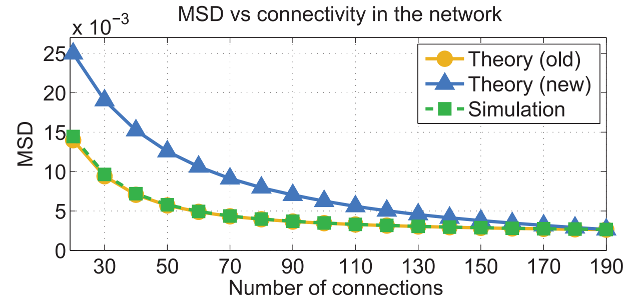


Fig. 2. MSD values for in a 20-node network with varying levels of connectivity.

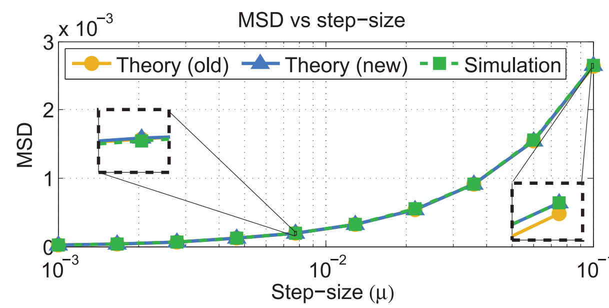


Fig. 3. MSD levels for different step-sizes.

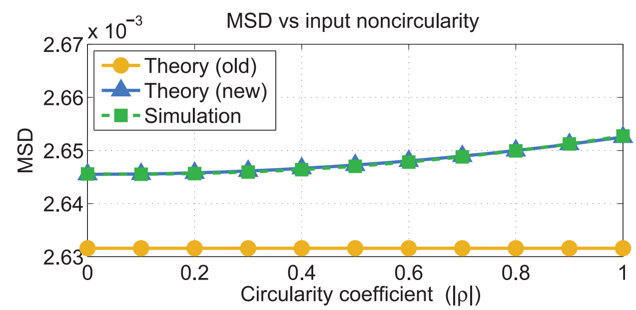


Fig. 4. MSD values for different levels of input noncircularity.

noncircular with varying levels of noncircularity modeled as $\mathbf{P} = \rho\mathbf{I}$. Current mean square analyses [1] use a small step-size assumption to neglect the second-order terms associated with $\mu^2$, and in doing so they do not capture the impact of noncircularity of the input signal on the performance. The proposed expression in (19), however, produced the MSD performance that almost identically matches the values obtained from the Monte-Carlo simulations. As shown in [20], the MSD is directly proportional to the noncircularity of the input.

## VI. Conclusion

We have introduced a novel, compact, and physically meaningful mean square analysis for D-CLMS and D-ACLMS algorithms. This has been achieved based on the so-called "similarity conjecture" which allows for the incorporation of the second-order terms, while maintaining the mathematical tractability of the analysis. The proposed method has then been applied to study the effect of the noncircularity of the input signal on the steady state mean square performance of the D-CLMS and D-ACLMS. Simulations on synthetic data have verified the proposed analysis.

## REFERENCES


[1] A. H. Sayed, "Diffusion adaptation over networks," *Acad. Press Lib. Signal Process.*, vol. 3, pp. 323–454, 2014.

[2] A. Sayed, "Adaptive networks," *Proc. IEEE*, vol. 102, no. 4, pp. 460–497, Apr. 2014.

[3] 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.

[4] F. S. Cattivelli and A. H. Sayed, "Diffusion LMS strategies for distributed estimation," *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1035–1048, Mar. 2010.

[5] J. Chen, C. Richard, and A. Sayed, "Diffusion LMS over multitask networks," *IEEE Trans. Signal Process.*, vol. 63, no. 1, pp. 2733–2748, Jun. 2015.

[6] R. Arablouei, S. Werner, Y.-F. Huang, and K. Dogancay, "Distributed least mean-square estimation with partial diffusion," *IEEE Trans. Signal Process.*, vol. 62, no. 2, pp. 472–484, Jan. 2014.

[7] R. Abdolee and B. Champagne, "Centralized adaptation for parameter estimation over wireless sensor networks," *IEEE Commun. Lett.*, vol. 19, no. 9, pp. 1624–1627, Sep. 2015.

[8] C. Jiang, Y. Chen, and K. Liu, "Distributed adaptive networks: A graphical evolutionary game-theoretic view," *IEEE Trans. Signal Process.*, vol. 61, no. 22, pp. 5675–5688, Nov. 2013.

[9] X. Zhao and A. Sayed, "Performance limits for distributed estimation over LMS adaptive networks," *IEEE Trans. Signal Process.*, vol. 60, no. 10, pp. 5107–5124, Oct. 2012.

[10] Y. Xia, D. P. Mandic, and A. H. Sayed, "An adaptive diffusion augmented CLMS algorithm for distributed filtering of noncircular complex signals," *IEEE Signal Process. Lett.*, vol. 18, no. 11, pp. 659–662, Nov. 2011.

[11] S. Kanna, S. Talebi, and D. Mandic, "Diffusion widely linear adaptive estimation of system frequency in distributed power grids," in *Proc. IEEE Int. Energy Conf. (ENERGYCON)*, May 2014, pp. 772–778.

[12] A. Sayed, S.-Y. Tu, J. Chen, X. Zhao, and Z. Towfic, "Diffusion strategies for adaptation and learning over networks: An examination of distributed strategies and network behavior," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 155–171, May 2013.

[13] V. Blondel, J. Hendrickx, A. Olshevsky, and J. Tsitsiklis, "Convergence in multiagent coordination, consensus, and flocking," in *Proc. 44th IEEE Conf. Decis. Control Eur. Control Conf. (CDC-ECC'05)*, Dec. 2005, pp. 2996–3000.

[14] A. Uncini, *Fundamentals of Adaptive Signal Processing*. New York, NY, USA: Springer, 2015.

[15] S. Douglas and M. Rupp, "Convergence issues in the LMS adaptive filter," in *The DSP Handbook*, V. Madisetti and D. Williams, Eds. Boca Raton, FL, USA: CRC/IEEE Press, 1998, ch. 19, pp. 19.1–19.21.

[16] M. T. M. Silva, V. H. Nascimento, and J. Arenas-García, "A transient analysis for the convex combination of two adaptive filters with transfer of coefficients," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, Mar. 2010, pp. 3842–3845.

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

[18] J. Fernandez-Bes, L. A. Azpicueta-Ruiz, M. T. M. Silva, and J. Arenas-Garcia, "Improved least-squares-based combiners for diffusion networks," in *Proc. 10th Int. Symp. Wireless Commun. Syst. (ISWCS'13)*, Aug. 2013, pp. 1–5.

[19] C. C. Took, S. Douglas, and D. Mandic, "On approximate diagonalization of correlation matrices in widely linear signal processing," *IEEE Trans. Signal Process.*, vol. 60, no. 3, pp. 1469–1473, Mar. 2012.

[20] D. Mandic, S. Kanna, and S. Douglas, "Mean square analysis of the CLMS and ACLMS for non-circular signals: The approximate uncorrelating transform approach," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Apr. 2015, pp. 3531–3535.

[21] D. P. Mandic and V. S. L. Goh, *Complex Valued Adaptive Filters*. Hoboken, NJ, USA: Wiley, 2009, pp. 69–90.