Diffusion Least-Mean Squares With Adaptive Combiners: Formulation and Performance Analysis

Noriyuki Takahashi, *Member, IEEE*, Isao Yamada, *Senior Member, IEEE*, and Ali H. Sayed, *Fellow, IEEE*

***Abstract*—This paper presents an efficient adaptive combination strategy for the distributed estimation problem over diffusion networks in order to improve robustness against the spatial variation of signal and noise statistics over the network. The concept of minimum variance unbiased estimation is used to derive the proposed adaptive combiner in a systematic way. The mean, mean-square, and steady-state performance analyses of the diffusion least-mean squares (LMS) algorithms with adaptive combiners are included and the stability of convex combination rules is proved. Simulation results show i) that the diffusion LMS algorithm with the proposed adaptive combiners outperforms those with existing static combiners and the incremental LMS algorithm, and ii) that the theoretical analysis provides a good approximation of practical performance.**

***Index Terms*—Adaptive filter, adaptive networks, combination, diffusion, distributed algorithm, distributed estimation, energy conservation.**

I. Introduction

DISTRIBUTED or cooperative processing over networks has been emerging as an efficient data processing technology for network applications, such as environment monitoring, disaster relief management, source localization, and other applications [2]–[7]. In contrast to classical centralized techniques, distributed processing utilizes local computations at each node and communications among neighboring nodes to solve problems over the entire network. This useful capability expands the scalability and flexibility of the network and leads to a wide range of applications.

In this paper, we consider the problem of distributed estimation over adaptive networks as advanced in [8]–[14]. The objective of this problem is to estimate adaptively certain parameters of interest in a distributed manner. Specifically, each node of the network is allowed to cooperate with its neighbors in order to improve the accuracy of its local estimate. Such cooperation enables each node to leverage the spatial diversity obtained from the geographical distribution of the nodes as well as the temporal diversity. Evidently, the performance of the resulting adaptive network depends on the mode of cooperation, for example, incremental [9], diffusion [10], or probabilistic diffusion [11]. Although each mode possesses its own advantages, in this paper, we focus on the diffusion mode of cooperation because this mode is more robust to node and link failure [11].

In the diffusion mode of cooperation, the nodes exchange their estimates with neighbors and fuse the collected estimates via linear combinations [10]. Several combination rules, such as the Metropolis [15] and relative-degree [12] rules, have been proposed that are based solely on the network topology, i.e., the combination weights are calculated from the degree of each node, and hence do not reflect the node profile. Therefore, the performance of such rules may deteriorate if, for instance, the signal-to-noise ratio (SNR) at some nodes is appreciably lower than others; because the noisy estimates of such nodes diffuse into the entire network by cooperation among the nodes. Therefore, the design of combination weights plays a key role in the diffusion mode of cooperation.

In [12] and [14], offline optimization of the combination weights was proposed based on the steady-state performance analysis. If the weights are optimized in advance, the performance of algorithms improves. However, such offline optimization requires the knowledge of network statistics, such as regressors and noise profile. Moreover, it might be difficult to solve the optimization problem in a distributed manner because the problem depends on information of the entire network. This difficulty can be overcome by resorting to online learning of the combination weights, say *adaptive combiners*. Initial investigations on adaptive combiners for diffusion algorithms were done in [8] based on the convex combination of two adaptive filters [16]. However, the structure of this adaptive combiner limits the degree of freedom of the weights; indeed, only one scalar coefficient is adaptive.

In this paper, we take a more systematic and more general approach than [8] and formulate the problem of controlling the combination weights as a well defined minimum variance unbiased estimation problem. We then use the problem to propose an adaptive combination rule that learns its combination weights so that the effect of noisy estimates is reduced. The resulting combiner is fully adaptive, i.e., in contrast to the offline optimization approach in [14], no knowledge of the network statistics is required. We also analyze the performance of the proposed adap-

Manuscript received July 16, 2009; accepted May 11, 2010. Date of publication June 01, 2010; date of current version August 11, 2010. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Daniel P. Palomar. This material was based on work supported in part by the National Science Foundation under awards ECS-0601266, ECS-0725441, and CCF-0942936. The work of the N. Takahashi was also supported in part by the Japan Society for the Promotion of Science (JSPS) while visiting the UCLA Adaptive Systems Laboratory. Some preliminary parts of this work appeared in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing* (ICASSP), Taipei, Taiwan, April 2009 [1].

N. Takahashi is with the Global Edge Institute, Tokyo Institute of Technology, Tokyo, 152-8550, Japan (e-mail: takahashi@sp.ss.titech.ac.jp).

I. Yamada is with the Department of Communications and Integrated Systems, Tokyo Institute of Technology, Tokyo 152-8550, Japan (e-mail: isao@sp.ss.titech.ac.jp).

A. H. Sayed is with the Department of Electrical Engineering, University of California, Los Angeles, CA 90095 USA (e-mail: sayed@ee.ucla.edu).

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

Digital Object Identifier 10.1109/TSP.2010.2051429

tive combiner. In the mean-transient analysis, we show that stability in the mean is improved via convex combinations, even if they are random and time-varying. This analysis is done under weaker assumptions than those given in our prior work [1]. On the other hand, in the mean-square performance analysis, we derive the learning behavior and the steady-state performance of the diffusion LMS algorithm with the proposed adaptive combiners. We finally verify via simulations that the theoretical performance curves agree closely with experimental performance curves.

The paper is organized as follows. In the end of this section, we summarize the notation we use throughout the paper. In Section II, a mathematical model for the distributed estimation problem and two types of diffusion strategies are introduced. In Section III, an adaptive combination rule is established by formulating a minimum variance unbiased estimation problem. Sections IV and V analyze the performance of the diffusion LMS algorithm with the proposed adaptive combiners. Also, stability conditions and explicit performance expressions are derived. Section VI studies the proposed combiner and the results of the analysis in numerical simulations. We also verify that the theoretical performance analysis provides a good approximation of experimental performance. Finally, we conclude the paper in Section VII.

A. Notation

Let $\mathbb{R}$ and $\mathbb{C}$ denote the sets of real and complex numbers, respectively. We use boldface letters to denote random variables and normal font for deterministic (non-random) quantities, e.g., $\boldsymbol{\psi}$ and ψ, respectively. Capital letters are used for matrices and small letters for vectors and scalars. All vectors are column vectors except for regression vectors, which are denoted by $u_{k,i}$ throughout. The superscript $(\cdot)^T$ represents the transpose of a matrix or a vector, while $(\cdot)^*$ represents the Hermitian (conjugate) transpose. The superscript $(\cdot)^*$ is also used to represent complex conjugation for scalars. The notation $\mathrm{col}\{\cdots\}$ stands for a vector obtained by stacking the specified vectors. Similarly, we use $\mathrm{diag}\{\cdots\}$ to denote the (block) diagonal matrix consisting of the specified vectors or matrices. The trace of a matrix is denoted by $\mathrm{Tr}(\cdot)$ and expectation is denoted by $\mathrm{E}[\cdot]$. We omit the brackets $[\cdot]$ if there is no possibility of confusion. Other notation will be introduced as necessary.

II. CTA and ATC Diffusion LMS Algorithms

Consider N nodes in a predefined network topology; see Fig. 1. We denote by $\mathcal{N}_k$ the neighborhood of node k including k itself and by n_k the degree of node k, i.e., the cardinality of $\mathcal{N}_k$. At each time $i \geq 0$, each node k has access to a scalar measurement $d_k(i) \in \mathbb{C}$ and a regression *row* vector $u_{k,i} \in \mathbb{C}^{1\times M}$ of length M that are related via

$$d_k(i) = u_{k,i}w^o + v_k(i), \tag{1}$$

where $w^o \in \mathbb{C}^M$ is an unknown column vector we wish to estimate and $v_k(i) \in \mathbb{C}$ accounts for noise and modeling errors. The objective of the distributed estimation problem is to generate an estimate $\psi_{k,i}$ of w^o at each node k and time i in a distributed manner; namely, each node is allowed to cooperate only with its neighbors.

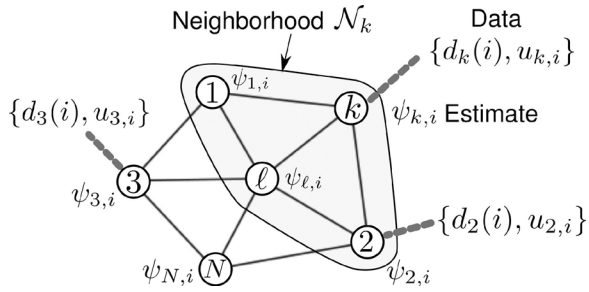


Fig. 1. A distributed network with N nodes. In this instance, the neighborhood of node k is $\mathcal{N}_k = \{1, 2, \ell, k\}$ and the degree of node k is $n_k = 4$. Each node k generates an estimate $\psi_{k,i}$ at each time i in cooperation with its neighbors.

The diffusion strategy we consider is performed in two stages [10], [13], [14]: combination and adaptation. In this strategy, each node k first computes a linear combination of local estimates $\{\psi_{\ell,i-1}\}_{\ell\in\mathcal{N}_k}$ collected from its neighbors, i.e.,

$$\phi_{k,i-1} \stackrel{\triangle}{=} \sum_{\ell\in\mathcal{N}_k} c_{\ell k}(i)\,\psi_{k,i-1}$$

where $\{c_{\ell k}(i)\}$ are possibly time-varying combination weights calculated from information up to time i available at node k. Then, the intermediate estimate $\phi_{k,i-1}$ of node k is used by a local adaptive filter to adapt the local data $\{d_k(i), u_{k,i}\}$ observed at node k. We refer to this strategy as *Combine-then-Adapt* (CTA) diffusion [10]; see Fig. 2(a). For example, using the LMS algorithm as the core adaptive filter leads to the CTA diffusion LMS algorithm [10]:

$$\begin{cases} \phi_{k,i-1} = & \sum_{\ell\in\mathcal{N}_k} c_{\ell k}(i)\,\psi_{\ell,i-1}, \\ \psi_{k,i} = & \phi_{k,i-1} + \mu_k u_{k,i}^*(d_k(i) - u_{k,i}\phi_{k,i-1}) \end{cases} \tag{2}$$

where $\mu_k > 0$ is the stepsize at node k.

It is possible to reverse the order of the two stages, i.e., adaptation followed by combination. We refer to this version as *Adapt-then-Combine* (ATC) diffusion ; see Fig. 2(b). Again, using the LMS algorithm as the core adaptive filter, we get the ATC diffusion LMS algorithm :

$$\begin{cases} \phi_{k,i} = \psi_{k,i-1} + \mu_k u_{k,i}^*(d_k(i) - u_{k,i}\psi_{k,i-1}) \\ \psi_{k,i} = \sum_{\ell\in\mathcal{N}_k} c_{\ell k}(i)\,\phi_{\ell,i}. \end{cases} \tag{3}$$

More general diffusion strategies that exchange not only estimates but also data have been studied in [13], [14].

In the CTA and ATC strategies, the combination weights $\{c_{\ell k}(i)\}$ play an important role. Table I shows examples of static combination rules, which keep $\{c_{\ell k}(i)\}$ constant; see, e.g., [15] and [17]–[22]. These rules calculate the weights based only on the network topology. However, such topology-based rules are sensitive to the spatial variation of signal and noise statistics across the network. Suppose, for example, that the estimates collected from some neighbors are less reliable than others due to low SNR conditions. In such case, we should give less weight to the noisy estimates. Conversely, nodes in low SNR conditions could improve the accuracy of their estimates by putting more weight on the estimates from neighbors in higher SNR conditions. Thus, static combination rules are

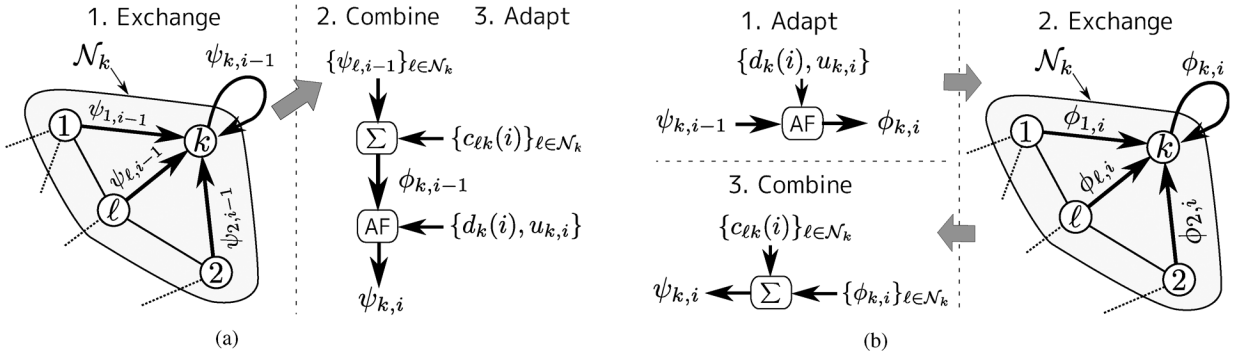


Fig. 2. Combine-then-Adapt (CTA) and Adapt-then-Combine (ATC) diffusion strategies. (a) CTA diffusion. (b) ATC diffusion.

TABLE I
STATIC COMBINATION RULES BASED ON NETWORK TOPOLOGY

Rule	Weights* $c_{\ell k}$ for $\ell \in \mathcal{N}_k$
Uniform (e.g., [18])	$1/n_k$ for all $\ell \in \mathcal{N}_k$
Maximum degree [20]	$\begin{cases} 1/N & \text{for } \ell \neq k \\ 1-(n_k-1)/N & \text{for } \ell = k \end{cases}$
Metropolis [15]	$\begin{cases} 1/\max\{n_k, n_\ell\} & \text{for } \ell \neq k \\ 1-\sum_{m\in\mathcal{N}_k\setminus\{k\}} c_{mk} & \text{for } \ell = k \end{cases}$
Relative degree [12]	$n_\ell / \sum_{m\in\mathcal{N}_k} n_m$ for all $\ell \in \mathcal{N}_k$
No cooperation	$\begin{cases} 0 & \text{if } \ell \neq k \\ 1 & \text{if } \ell = k \end{cases}$

*In all these rules, $c_{\ell k} = 0$ for $\ell \notin \mathcal{N}_k$.

likely to result in performance deterioration. To improve the robustness to such cases, we propose an adaptive combination strategy.

III. ADAPTIVE COMBINERS

We first formulate a problem that determines optimal combination weights. Then, a stochastic gradient type algorithm is derived to approximate the optimal weights, leading to an adaptation rule for the combination weights. In what follows, we restrict the weights $\{c_{\ell k}(i)\}$ to real values and focus on the CTA diffusion. Subsequently, our combiner for the ATC version is derived in a similar manner.

A. Problem Formulation

To begin with, let us assume that for each $k \in \{1,\ldots,N\}$ the local estimates $\{\psi_{k,i}, i \geq 0\}$ are realizations of some random vector $\boldsymbol{\psi}_k$ that satisfies the following condition:

$$\mathrm{E}\boldsymbol{\psi}_k = w^o \quad \text{for all } k \in \{1,\ldots,N\}. \tag{4}$$

We further introduce the notation

$$\boldsymbol{\Psi} \triangleq [\boldsymbol{\psi}_1,\ldots,\boldsymbol{\psi}_N],$$
$$c_k \triangleq \mathrm{col}\{c_{1k}, c_{2k}, \ldots, c_{Nk}\} \in \mathbb{R}^N, \quad k \in \{1,\ldots,N\}$$

where $\boldsymbol{\Psi}$ is an $M \times N$ complex random matrix and each c_k represents the combination weight vector for node k to be optimized. Our approach is to minimize the difference between the linear combination $\boldsymbol{\Psi}c_k$ and the unknown vector w^o, i.e.,

$$\begin{aligned} \underset{c_1,\ldots,c_N\in\mathbb{R}^N}{\text{minimize}} \quad & \sum_{k=1}^{N} \mathrm{E}\|\boldsymbol{\Psi}c_k - w^o\|^2 \\ \text{subject to} \quad & c_{\ell k} = 0 \\ & \text{for all } \ell \notin \mathcal{N}_k, \quad k \in \{1,\ldots,N\} \end{aligned}$$

where $\|\cdot\|$ denotes the Euclidean norm on the space $\mathbb{C}^M$ and the constraints must be satisfied because node k has no direct access to realizations of $\boldsymbol{\psi}_\ell$ for $\ell \notin \mathcal{N}_k$. Obviously, this problem can be decoupled into N subproblems: for each $k \in \{1,\ldots,N\}$,

$$\begin{aligned} \underset{c_k\in\mathbb{R}^N}{\text{minimize}} \quad & J(c_k) \triangleq \mathrm{E}\|\boldsymbol{\Psi}c_k - w^o\|^2 \\ \text{subject to} \quad & c_{\ell k} = 0 \quad \text{for all } \ell \notin \mathcal{N}_k. \end{aligned}$$

Unfortunately, each subproblem cannot be solved directly due to the presence of the unknown quantity w^o. Therefore, we employ the concept of minimum variance unbiased estimation (or best linear unbiased estimation, see, e.g., [23]) to reformulate the problem into a form solvable in a distributed way.

Let us expand the cost function $J(c_k)$ to obtain a bias-variance decomposition of J as follows:

$$\begin{aligned} J(c_k) &= \mathrm{E}\|(\boldsymbol{\Psi}c_k - \mathrm{E}\boldsymbol{\Psi}c_k) + (\mathrm{E}\boldsymbol{\Psi}c_k - w^o)\|^2 \\ &= \underbrace{c_k^T Q_\Psi c_k}_{\text{variance}} + \underbrace{\|\mathrm{E}\boldsymbol{\Psi}c_k - w^o\|^2}_{\text{bias}^2} \end{aligned}$$

where Q_Ψ is an $N \times N$ matrix defined as

$$Q_\Psi \triangleq \mathrm{E}[(\boldsymbol{\Psi} - \mathrm{E}\boldsymbol{\Psi})^*(\boldsymbol{\Psi} - \mathrm{E}\boldsymbol{\Psi})].$$

Moreover, noting that Q_Ψ is Hermitian and using (4), we have

$$\begin{aligned} J(c_k) &= c_k^T \left(\frac{Q_\Psi + Q_\Psi^T}{2}\right) c_k + \|\mathrm{E}\boldsymbol{\Psi}c_k - w^o\|^2 \\ &= c_k^T \mathrm{Re}(Q_\Psi) c_k + \|(\mathbb{1}_N^T c_k - 1)w^o\|^2 \end{aligned}$$

where $\mathrm{Re}(\cdot)$ denotes the real part and $\mathbb{1}_N$ is the vector of length N whose components are all unity. Therefore, if we impose the condition $\mathbb{1}_N^T c_k = 1$, the second term involving the unknown quantity w^o vanishes and we arrive at the following

minimum variance unbiased estimation problem: for each $k \in \{1,\ldots,N\}$,

$$\boxed{\begin{array}{ll}\underset{c_k\in\mathbb{R}^N}{\text{minimize}} & c_k^T \operatorname{Re}(Q_\Psi) c_k \\ \text{subject to} & \mathbb{1}_N^T c_k = 1, \\ & c_{\ell k} = 0 \quad \text{for all } \ell \notin \mathcal{N}_k.\end{array}} \tag{5}$$

Note that the nonnegativity of $\{c_{\ell k}\}$ is not imposed in (5). In what follows, we seek a closed form solution and an adaptive solution to problem (5).

B. Closed Form Solution

The dimension of problem (5) can be reduced from N unknowns to n_k by introducing an auxiliary variable. For each $k \in \{1,\ldots,N\}$, define the $N \times n_k$ matrix P_k:

$$P_k \triangleq [\ell\text{th column of } I_N]_{\ell\in\mathcal{N}_k}. \tag{6}$$

Then, any vector $c_k \in \mathbb{R}^N$ that satisfies the constraints of (5) can be represented as

$$c_k = P_k a_k \tag{7}$$

using some $a_k \in \mathbb{R}^{n_k}$ satisfying $\mathbb{1}_{n_k}^T a_k = 1$, where $\mathbb{1}_{n_k} \triangleq P_k^T \mathbb{1}_N$ is the vector of length n_k whose components are all unity. Therefore, substituting (7) into (5), we obtain the following problem:

$$\begin{array}{ll}\underset{a_k\in\mathbb{R}^{n_k}}{\text{minimize}} & f_k(a_k) \triangleq a_k^T \operatorname{Re}(Q_{\Psi,k}) a_k \\ \text{subject to} & a_k \in V_k \triangleq \{x \in \mathbb{R}^{n_k} \mid \mathbb{1}_{n_k}^T x = 1\}\end{array} \tag{8}$$

where $Q_{\Psi,k}$ is the $n_k \times n_k$ submatrix of Q_Ψ given by

$$Q_{\Psi,k} \triangleq P_k^T Q_\Psi P_k.$$

The constraint V_k of problem (8) is called a *hyperplane* and the solution to (8) is well known to be (see, e.g., [23] or the Appendix):

$$a_k^o \triangleq \frac{\operatorname{Re}(Q_{\Psi,k})^{-1}\mathbb{1}_{n_k}}{\mathbb{1}_{n_k}^T \operatorname{Re}(Q_{\Psi,k})^{-1}\mathbb{1}_{n_k}} \tag{9}$$

provided that $\operatorname{Re}(Q_{\Psi,k})$ is positive definite. (Note that $\operatorname{Re}(Q_{\Psi,k})$ is at least positive semidefinite.) Then, the solution of (5) can be recovered from (7) as $c_k^o = P_k a_k^o$. Since all the components of $Q_{\Psi,k}$ are second order moments of the random vectors $\boldsymbol{\psi}_\ell$, $\ell \in \mathcal{N}_k$, we can approximate a_k^o from local information available at node k, if we collect a number of realizations of $\{\boldsymbol{\psi}_\ell\}_{\ell\in\mathcal{N}_k}$ sufficient to estimate the moment $Q_{\Psi,k}$. However, for the purpose of an adaptive implementation, we introduce a steepest-descent type solution.

C. Steepest-Descent Solution

In order to apply the standard steepest-descent method to (8), we need to eliminate the constraint V_k. We apply a similar technique introduced in [24, Example 5]. Let $\mathcal{P}_{V_k}$ be the *metric projection* from $\mathbb{R}^{n_k}$ onto V_k, which is defined and given by (see the Appendix)

$$\begin{aligned}\mathcal{P}_{V_k}(x) &\triangleq \underset{y\in V_k}{\arg\min} \|y - x\| \\ &= \left(I_{n_k} - \frac{\mathbb{1}_{n_k}\mathbb{1}_{n_k}^T}{n_k}\right) x + \frac{\mathbb{1}_{n_k}}{n_k}\end{aligned} \tag{10}$$

for all $x \in \mathbb{R}^{n_k}$. The transformation $\mathcal{P}_{V_k}$ maps an arbitrary vector $x \in \mathbb{R}^{n_k}$ into a vector $a \in V_k$ satisfying $\mathbb{1}_{n_k}^T a = 1$. Using $\mathcal{P}_{V_k}$, we therefore introduce a second auxiliary variable. Since the projection $\mathcal{P}_{V_k} : \mathbb{R}^{n_k} \to V_k$ is surjective [25], [26], every point $a_k \in V_k$ can be represented as the projection of a point $b_k \in \mathbb{R}^{n_k}$, i.e.,

$$a_k = \mathcal{P}_{V_k}(b_k). \tag{11}$$

Therefore, substituting (11) into (8), we arrive at the following unconstrained problem:

$$\underset{b_k\in\mathbb{R}^{n_k}}{\text{minimize}} \quad \mathcal{P}_{V_k}(b_k)^T \operatorname{Re}(Q_{\Psi,k}) \mathcal{P}_{V_k}(b_k). \tag{12}$$

Since this cost function is quadratic because $\mathcal{P}_{V_k}$ is affine in b_k, the gradient of (12) at b_k can be easily calculated as

$$2\left(I_{n_k} - \frac{\mathbb{1}_{n_k}\mathbb{1}_{n_k}^T}{n_k}\right) \operatorname{Re}(Q_{\Psi,k}) \mathcal{P}_{V_k}(b_k).$$

Hence, applying the standard steepest-descent method to (12) and then recovering c_k using (11) and (7), we obtain the following algorithm:

$$\begin{cases} g_{k,i} = \left(I_{n_k} - \dfrac{\mathbb{1}_{n_k}\mathbb{1}_{n_k}^T}{n_k}\right) \operatorname{Re}(Q_{\Psi,k}) \mathcal{P}_{V_k}(b_{k,i-1}) \\ b_{k,i} = b_{k,i-1} - \nu_k(i) g_{k,i} \\ a_{k,i} = \mathcal{P}_{V_k}(b_{k,i}) \\ c_{k,i} = P_k a_{k,i} \end{cases} \tag{13}$$

where $\nu_k(i) \geq 0$ is a stepsize parameter. This recursion can be simplified if we choose an initial point $b_{k,-1}$ from V_k. Indeed, if $b_{k,-1}$ lies in V_k, i.e., if $\mathbb{1}_k^T b_{k,-1} = 1$, the fact that $\mathbb{1}_{n_k}^T g_{k,i} = 0$ for all $i \geq 0$ implies

$$\mathbb{1}_{n_k}^T b_{k,i} = \mathbb{1}_{n_k}^T b_{k,i-1} = \cdots = \mathbb{1}_{n_k}^T b_{k,-1} = 1$$

or $b_{k,i} \in V_k$ for all $i \geq -1$. Therefore, since $b_{k,i} \in V_k$ is equivalent to $\mathcal{P}_{V_k}(b_{k,i}) = b_{k,i}$, algorithm (13) is simplified to the following:

$$\begin{cases} g_{k,i} = \left(I_{n_k} - \dfrac{\mathbb{1}_{n_k}\mathbb{1}_{n_k}^T}{n_k}\right) \operatorname{Re}(Q_{\Psi,k}) b_{k,i-1} \\ b_{k,i} = b_{k,i-1} - \nu_k(i) g_{k,i} \\ c_{k,i} = P_k b_{k,i} \end{cases} \tag{14}$$

where $b_{k,-1}$ must satisfy $\mathbb{1}_{n_k}^T b_{k,-1} = 1$. For instance, $b_{k,-1} = \mathbb{1}_{n_k}/n_k$.

There is a relation between the update direction $-g_{k,i}$ and the antigradient of f_k at $b_{k,i-1}$, say $-\nabla f_k(b_{k,i-1})$. The direction $-g_{k,i}$ is the component of $-\nabla f_k(b_{k,i-1})$ orthogonal to the

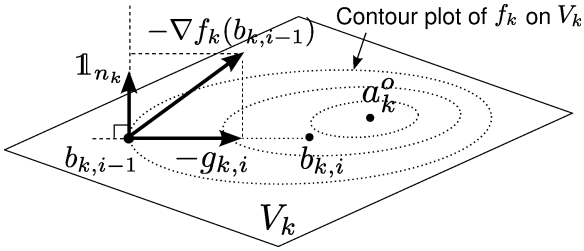


Fig. 3. An interpretation of algorithm (14) from a geometric point of view.

normal vector $\mathbb{1}_{n_k}$ of V_k; see Fig. 3. Namely, the descent direction of f_k is embedded in V_k so that $b_{k,i}$ stays in V_k.

D. Adaptive Solution

We now replace $Q_{\Psi,k}$ by an instantaneous approximation to derive an adaptive version of (14). Since

$$Q_{\Psi,k} = P_k^T Q_\Psi P_k = \mathrm{E}[(\boldsymbol{\Psi}_k - \mathrm{E}\boldsymbol{\Psi}_k)^*(\boldsymbol{\Psi}_k - \mathrm{E}\boldsymbol{\Psi}_k)]$$

where $\boldsymbol{\Psi}_k \overset{\triangle}{=} \boldsymbol{\Psi} P_k = [\boldsymbol{\psi}_\ell]_{\ell\in\mathcal{N}_k}$ is the $M \times n_k$ random matrix, we need to approximate $\boldsymbol{\Psi}_k - \mathrm{E}\boldsymbol{\Psi}_k$. At each time i, we replace this random quantity by its instantaneous variation:

$$\boldsymbol{\Psi}_k - \mathrm{E}\boldsymbol{\Psi}_k \approx \Delta\Psi_{k,i-1} \overset{\triangle}{=} [\psi_{\ell,i-1} - \psi_{\ell,i-2}]_{\ell\in\mathcal{N}_k}.$$

Using this, we consider the following approximation:

$$Q_{\Psi,k} \approx \frac{1}{2}(\Delta\Psi_{k,i-1})^*\Delta\Psi_{k,i-1}. \tag{15}$$

Note that $\Delta\Psi_{k,i-1}$ can be calculated from information available at node k. The instantaneous approximation (15) is justified as follows. Let us now interpret $\Delta\Psi_{k,i-1}$ as a random matrix, i.e., denote by boldface: $\Delta\boldsymbol{\Psi}_{k,i-1} \overset{\triangle}{=} \boldsymbol{\Psi}_{k,i-1} - \boldsymbol{\Psi}_{k,i-2}$, where $\boldsymbol{\Psi}_{k,i-1} \overset{\triangle}{=} [\boldsymbol{\psi}_{\ell,i-1}]_{\ell\in\mathcal{N}_k}$ and $\boldsymbol{\Psi}_{k,i-2} \overset{\triangle}{=} [\boldsymbol{\psi}_{\ell,i-2}]_{\ell\in\mathcal{N}_k}$. Furthermore, assume that $\boldsymbol{\Psi}_{k,i-1}$ and $\boldsymbol{\Psi}_{k,i-2}$ are uncorrelated and have the same distribution as $\boldsymbol{\Psi}_k$. Then, noting that

$$\Delta\boldsymbol{\Psi}_{k,i-1} = (\boldsymbol{\Psi}_{k,i-1} - \mathrm{E}\boldsymbol{\Psi}_k) - (\boldsymbol{\Psi}_{k,i-2} - \mathrm{E}\boldsymbol{\Psi}_k)$$

we have

$$\begin{aligned}
&\mathrm{E}[(\Delta\boldsymbol{\Psi}_{k,i-1})^*\Delta\boldsymbol{\Psi}_{k,i-1}] \\
&\quad = \mathrm{E}[(\boldsymbol{\Psi}_{k,i-1} - \mathrm{E}\boldsymbol{\Psi}_k)^*(\boldsymbol{\Psi}_{k,i-1} - \mathrm{E}\boldsymbol{\Psi}_k)] \\
&\qquad + \mathrm{E}[(\boldsymbol{\Psi}_{k,i-2} - \mathrm{E}\boldsymbol{\Psi}_k)^*(\boldsymbol{\Psi}_{k,i-2} - \mathrm{E}\boldsymbol{\Psi}_k)] \\
&\quad = 2\mathrm{E}[(\boldsymbol{\Psi}_k - \mathrm{E}\boldsymbol{\Psi}_k)^*(\boldsymbol{\Psi}_k - \mathrm{E}\boldsymbol{\Psi}_k)] \\
&\quad = 2Q_{\Psi,k}.
\end{aligned}$$

Thus, removing the expectation in the left-hand side yields the instantaneous approximation (15). Finally, combining (2), (14), and (15), we obtain the CTA diffusion LMS algorithm with adaptive combiners. In a similar fashion, the ATC version of our adaptive combiner is derived. Fig. 4 summarizes the proposed diffusion LMS algorithms, where the scalar constant $1/2$ in (15) merged with the stepsize $\nu_k(i)$.

Diffusion LMS Algorithms with Adaptive Combiners

For every time $i \geq 0$, repeat the following CTA or ATC updates. (In what follows, $\nu_k(i) \geq 0$ is a stepsize for the combiner update while $\mu_k > 0$ is a stepsize for the LMS algorithm.)

CTA

Initialization: At each node k, set $\psi_{k,-1} = \psi_{k,-2} = 0$ and choose $b_{k,-1} \in \mathbb{R}^{n_k}$ so that $\mathbb{1}_{n_k}^T b_{k,-1} = 1$.

1. Combiner update: After exchanging the local estimates $\psi_{k,i-1}$ with neighbors, update $b_{k,i-1}$ and compute the combination weights $c_{k,i} = \mathrm{col}\{c_{1k}(i), \ldots, c_{Nk}(i)\}$ via

$$\begin{cases}
g_{k,i} = \left(I_{n_k} - \dfrac{\mathbb{1}_{n_k}\mathbb{1}_{n_k}^T}{n_k}\right)\mathrm{Re}\{(\Delta\Psi_{k,i-1})^*\Delta\Psi_{k,i-1}\}b_{k,i-1}, \\
b_{k,i} = b_{k,i-1} - \nu_k(i)\, g_{k,i}, \\
c_{k,i} = P_k b_{k,i},
\end{cases} \tag{16}$$

where $\Delta\Psi_{k,i-1} \overset{\triangle}{=} [\psi_{\ell,i-1} - \psi_{\ell,i-2}]_{\ell\in\mathcal{N}_k}$.

2. Combination:

$$\phi_{k,i-1} = \sum_{\ell\in\mathcal{N}_k} c_{\ell k}(i)\, \psi_{k,i-1},$$

where $c_{k,i} = [c_{\ell k}(i)]_{\ell=1}^N$.

3. Adaptation by the LMS algorithm:

$$\psi_{k,i} = \phi_{k,i-1} + \mu_k u_{k,i}^*\{d_k(i) - u_{k,i}\phi_{k,i-1}\}. \tag{17}$$

ATC

Initialization: At each node k, set $\psi_{k,-1} = \phi_{k,-1} = 0$ and choose $b_{k,-1} \in \mathbb{R}^{n_k}$ so that $\mathbb{1}_{n_k}^T b_{k,-1} = 1$.

1. Adaptation by the LMS algorithm:

$$\phi_{k,i} = \psi_{k,i-1} + \mu_k u_{k,i}^*\{d_k(i) - u_{k,i}\psi_{k,i-1}\}.$$

2. Combiner update: After exchanging the local estimates $\phi_{k,i}$ with neighbors, update $b_{k,i-1}$ and compute the combination weights $c_{k,i} = \mathrm{col}\{c_{1k}(i), \ldots, c_{Nk}(i)\}$ via

$$\begin{cases}
g_{k,i} = \left(I_{n_k} - \dfrac{\mathbb{1}_{n_k}\mathbb{1}_{n_k}^T}{n_k}\right)\mathrm{Re}\{(\Delta\Phi_{k,i})^*\Delta\Phi_{k,i}\}b_{k,i-1}, \\
b_{k,i} = b_{k,i-1} - \nu_k(i)\, g_{k,i}, \\
c_{k,i} = P_k b_{k,i},
\end{cases}$$

where $\Delta\Phi_{k,i} \overset{\triangle}{=} [\phi_{\ell,i} - \phi_{\ell,i-1}]_{\ell\in\mathcal{N}_k}$.

3. Combination:

$$\psi_{k,i} = \sum_{\ell\in\mathcal{N}_k} c_{\ell k}(i)\, \phi_{k,i},$$

where $c_{k,i} = [c_{\ell k}(i)]_{\ell=1}^N$.

Fig. 4. CTA and ATC diffusion LMS algorithms with the proposed adaptive combiners.

Remark 3.1: The proposed adaptive combiners can be combined with any adaptive filters and are not limited to the LMS algorithm.

E. Normalized Stepsize Rule

As we will see in the next section, the stability of the CTA diffusion LMS algorithm is ensured in the mean sense if we use convex combination weights for $\{c_{\ell k}(i)\}_{\ell\in\mathcal{N}_k}$. Since algorithm (16) (see Fig. 4) guarantees $\mathbb{1}_N^T c_{k,i}=1$, the weight vector $c_{k,i}$ becomes convex combination if $c_{\ell k}(i)\geq 0$ for all $\ell\in\{1,\ldots,N\}$. A possible choice for $\nu_k(i)$ that enforces $c_{\ell k}(i)\geq 0$ is the following:

$$\nu_k(i)=\alpha\frac{\min\left\{b_{k,i-1}^{(m)}\,|\,1\leq m\leq n_k\right\}}{\|g_{k,i}\|_\infty+\varepsilon} \tag{18}$$

where $\alpha\in(0,1)$ and $\varepsilon>0$ are constants, $\|\cdot\|_\infty$ denotes the maximum norm, and $b_{k,i-1}^{(m)}$ is the mth component of $b_{k,i-1}$. It is easily verified from (18) and the second line of (14) that the weight vector $b_{k,i}$ enforces a convex combination for all $i\geq 0$, provided that the initial vector $b_{k,-1}$ is a convex combination. Hence, $c_{k,i}=P_k b_{k,i}$ is ensured to be a convex combination as well.

Remark 3.2: As used in [8], [16], another possible way to ensure the nonnegativity of each weight is to introduce a parameterization of the form $c_{\ell k}(i)=1/(1+e^{\gamma_{\ell k}(i)})$, where $\gamma_{\ell k}(i)$ is any real number. Such a parameterization is useful to guarantee convex combination of *two* estimates because any weights $\{c_{\ell k}(i),\ c_{kk}(i)\}$ of the form

$$c_{\ell k}(i)=\frac{1}{1+e^{\gamma_{\ell k}(i)}},\quad c_{kk}(i)=1-c_{\ell k}(i),\quad \gamma_{\ell k}(i)\in\mathbb{R}$$

become convex combination. However, if we consider convex combination of more than two estimates, that parameterization is not generally useful because it is as difficult to handle the constraint $\sum_{\ell\in\mathcal{N}_k}1/(1+e^{\gamma_{\ell k}(i)})=1$. Moreover, the cost function f_k in (8) becomes nonlinear due to the introduction of $\{\gamma_{\ell k}(i),\ \ell\in\mathcal{N}_k\}$.

IV. Mean Transient Analysis

In this section, the mean transient analysis of the CTA diffusion LMS algorithm with *any* adaptive combiners, including the proposed combiner in Fig. 4, is presented. The goal of this section is to find sufficient conditions under which every local estimate converges in the mean to the unknown parameter w^o by extending the techniques of [10] to the challenging case where the combination weights are adaptive (i.e., time-variant) as opposed to static. To begin with, let us introduce a mathematical tool for the analysis.

A. Block Maximum Norm

Given a vector $x=\mathrm{col}\{x_1,\ldots,x_N\}\in\mathbb{C}^{MN}$ consisting of N blocks $x_k\in\mathbb{C}^M$, $k\in\{1,\ldots,N\}$, we define the *block maximum norm* $\|\cdot\|_{\mathrm{b}\infty}$ on $\mathbb{C}^{MN}$ by

$$\|x\|_{\mathrm{b}\infty}\triangleq\max_{1\leq k\leq N}\|x_k\|$$

where $\|\cdot\|$ denotes the standard Euclidean norm on $\mathbb{C}^M$ as before. We also define the matrix norm induced from the block maximum norm. Namely, given an $MN\times MN$ matrix A, its induced norm is defined as

$$\|A\|_{\mathrm{b}\infty}\triangleq\max_{\substack{x\in\mathbb{C}^{MN}\\ x\neq 0}}\frac{\|Ax\|_{\mathrm{b}\infty}}{\|x\|_{\mathrm{b}\infty}}.$$

This kind of norms is useful for the analysis of diffusion type algorithms; see for example [27] and [26]. The block maximum norm inherits the unitary invariance of the Euclidean norm $\|\cdot\|$ under block-wise transformation.

Lemma 4.1: Let $Y=\mathrm{diag}\{Y_1,\ldots,Y_N\}$ be an $MN\times MN$ block unitary matrix with $M\times M$ unitary blocks Y_k, $k\in\{1,\ldots,N\}$. Then, the following hold:

a) $\|Yx\|_{\mathrm{b}\infty}=\|x\|_{\mathrm{b}\infty}$ for all $x\in\mathbb{C}^{MN}$;

b) $\|YAY^*\|_{\mathrm{b}\infty}=\|A\|_{\mathrm{b}\infty}$ for all $A\in\mathbb{C}^{MN\times MN}$.

Proof: Property a) immediately follows from the fact that $\|Y_k x_k\|=\|x_k\|$ for every k. On the other hand, property b) immediately follows from a). ■

B. Mean Convergence Analysis

Let us now interpret the data as random variables. Recall that random variables are denoted by boldface letters, e.g., $\boldsymbol{u}_{k,i}$. To construct a data model in terms of global quantities, we introduce the following global quantities across the network:

$$w^{(o)}\triangleq\mathrm{col}\{w^o,\ldots,w^o\},\quad \boldsymbol{d}_i\triangleq\mathrm{col}\{\boldsymbol{d}_1(i),\ldots,\boldsymbol{d}_N(i)\}$$
$$\boldsymbol{v}_i\triangleq\mathrm{col}\{\boldsymbol{v}_1(i),\ldots,\boldsymbol{v}_N(i)\}$$
$$\boldsymbol{U}_i\triangleq\mathrm{diag}\{\boldsymbol{u}_{1,i},\ldots,\boldsymbol{u}_{N,i}\}=\begin{bmatrix}\boldsymbol{u}_{1,i}&&\\&\ddots&\\&&\boldsymbol{u}_{N,i}\end{bmatrix}.$$

Note that $\boldsymbol{U}_i$ is the $N\times NM$ block diagonal matrix with diagonal blocks $\{\boldsymbol{u}_{k,i}:k=1,\ldots,N\}$. Then, data model (1) can be written as

$$\boldsymbol{d}_i=\boldsymbol{U}_i w^{(o)}+\boldsymbol{v}_i,\quad i=0,1,\ldots.$$

Here, we assume the following throughout our analysis:

Assumption 4.2 (Data model):

a) The regressors $\{\boldsymbol{u}_{k,i}\}$ are i.i.d. in time and spatially independent, with $R_{u,k}\triangleq\mathrm{E}[\boldsymbol{u}_{k,i}^*\boldsymbol{u}_{k,i}]$.

b) The noise $\{\boldsymbol{v}_k(i)\}$ is i.i.d. zero-mean in time and spatially independent, with $\sigma_{v,k}^2\triangleq\mathrm{E}|\boldsymbol{v}_k(i)|^2$. In addition, $\boldsymbol{v}_k(i)$ is independent of the regressors $\{\boldsymbol{u}_{k,i}\}$.

Although the independence of regressors might be unrealistic in practice, such assumptions are employed frequently in the analysis of adaptive algorithms and lead to a close agreement between theory and experiments; see, for example, [10], [28], and references therein.

We next derive the state space model of the CTA diffusion LMS algorithm with adaptive combiners. Let $\{\boldsymbol{c}_{\ell k}(i)\}_{\ell=1}^N$ be any real random combination weights of node k at time i that satisfy $\boldsymbol{c}_{\ell k}(i)=0$ for $\ell\notin\mathcal{N}_k$ and $\sum_{\ell\in\mathcal{N}_k}\boldsymbol{c}_{\ell k}(i)=1$. We

collect all the weights into an $N \times N$ random matrix $\boldsymbol{C}_i$ and define the $MN \times MN$ random matrix $\boldsymbol{G}_i$ as follows:

$$\boldsymbol{C}_i \triangleq \begin{bmatrix} \boldsymbol{c}_{11}(i) & \cdots & \boldsymbol{c}_{1N}(i) \\ \vdots & \ddots & \vdots \\ \boldsymbol{c}_{N1}(i) & \cdots & \boldsymbol{c}_{NN}(i) \end{bmatrix}, \quad \boldsymbol{G}_i \triangleq \boldsymbol{C}_i^T \otimes I_M$$

where $\otimes$ represents the Kronecker product. Furthermore, letting

$$\begin{aligned} D &\triangleq \mathrm{diag}\{\mu_1 I_M, \ldots, \mu_N I_M\} \\ \boldsymbol{\psi}_{i-1} &\triangleq \mathrm{col}\{\boldsymbol{\psi}_{1,i-1}, \ldots, \boldsymbol{\psi}_{N,i-1}\} \\ \boldsymbol{\phi}_{i-1} &\triangleq \mathrm{col}\{\boldsymbol{\phi}_{1,i-1}, \ldots, \boldsymbol{\phi}_{N,i-1}\} \end{aligned}$$

we obtain the state space model for the CTA diffusion LMS algorithm with adaptive combiners:

$$\begin{cases} \boldsymbol{\phi}_{i-1} = \boldsymbol{G}_i \boldsymbol{\psi}_{i-1}, \\ \boldsymbol{\psi}_i = \boldsymbol{\phi}_{i-1} + D\boldsymbol{U}_i^*(\boldsymbol{d}_i - \boldsymbol{U}_i \boldsymbol{\phi}_{i-1}). \end{cases} \tag{19}$$

Now, let us rewrite this representation in terms of the following weight errors:

$$\begin{aligned} \tilde{\boldsymbol{\psi}}_{k,i} &\triangleq w^o - \boldsymbol{\psi}_{k,i}, \quad k = 1, \ldots, N \\ \tilde{\boldsymbol{\psi}}_i &\triangleq \mathrm{col}\{\tilde{\boldsymbol{\psi}}_{1,i}, \ldots, \tilde{\boldsymbol{\psi}}_{N,i}\} = w^{(o)} - \boldsymbol{\psi}_i. \end{aligned} \tag{20}$$

Noting that $\boldsymbol{G}_i w^{(o)} = w^{(o)}$ holds because of $\sum_{\ell=1}^N \boldsymbol{c}_{\ell,k}(i) = 1$, and subtracting $w^{(o)}$ from both sides of (19), we find out that the global error $\tilde{\boldsymbol{\psi}}_i$ evolves according to the following recursion:

$$\boxed{\tilde{\boldsymbol{\psi}}_i = (I_{MN} - D\boldsymbol{U}_i^* \boldsymbol{U}_i)\boldsymbol{G}_i \tilde{\boldsymbol{\psi}}_{i-1} - D\boldsymbol{U}_i^* \boldsymbol{v}_i.} \tag{21}$$

In what follows, we evaluate the expectation of both sides. However, the presence of the matrix $\boldsymbol{G}_i$ makes this evaluation difficult. Therefore, let us employ the following assumption to overcome this difficulty:

Assumption 4.3: For each $i \geq 0$, the weight matrix $\boldsymbol{C}_i$ is independent both of $\boldsymbol{U}_i$ and of $\tilde{\boldsymbol{\psi}}_{i-1}$.

Note that, at least for the proposed combiner for the CTA diffusion, the weight matrix $\boldsymbol{C}_i$ is independent of $\boldsymbol{U}_i$ under Assumption 4.2. Also, every static combiner satisfies this assumption obviously. Under Assumptions 4.2 and 4.3, we take the expectation of both sides of (21) to get

$$\mathrm{E}\tilde{\boldsymbol{\psi}}_i = (I_{MN} - DR_u)G_i \mathrm{E}\tilde{\boldsymbol{\psi}}_{i-1}$$

where

$$R_u \triangleq \mathrm{diag}\{R_{u,1}, \ldots, R_{u,N}\}, \quad G_i \triangleq \mathrm{E}[\boldsymbol{G}_i]. \tag{22}$$

Moreover, evaluating the block maximum norm yields the following inequality:

$$\|\mathrm{E}\tilde{\boldsymbol{\psi}}_i\|_{\mathrm{b}\infty} \leq \|B\|_{\mathrm{b}\infty} \cdot \|G_i\|_{\mathrm{b}\infty} \cdot \|\mathrm{E}\tilde{\boldsymbol{\psi}}_{i-1}\|_{\mathrm{b}\infty}$$

where $B \triangleq I_{MN} - DR_u$. Therefore, if

$$\sup_{i \geq 0}\{\|B\|_{\mathrm{b}\infty} \cdot \|G_i\|_{\mathrm{b}\infty}\} < 1 \tag{23}$$

then we can conclude that $\mathrm{E}\tilde{\boldsymbol{\psi}}_i \to 0$. Here, the convergence is ensured not only in the block maximum norm sense but also in any norm because all norms are equivalent in a finite dimensional vector space. One sufficient condition for (23) to hold is $\|B\|_{\mathrm{b}\infty} < 1$ and $\|G_i\|_{\mathrm{b}\infty} \leq 1$ for all $i \geq 0$. Therefore, let us find sufficient conditions on the stepsizes and the combination weights to ensure this condition.

Consider the requirement $\|B\|_{\mathrm{b}\infty} < 1$ first. To evaluate $\|B\|_{\mathrm{b}\infty}$, let us introduce the eigenvalue decomposition $R_{u,k} = T_k \Lambda_k T_k^*$. This immediately gives the eigenvalue decomposition of R_u:

$$R_u = T\Lambda T^* \tag{24}$$

where

$$T \triangleq \mathrm{diag}\{T_1, \ldots, T_N\} \quad \text{and } \Lambda \triangleq \mathrm{diag}\{\Lambda_1, \ldots, \Lambda_N\}.$$

Hence, noting that $T^* D T = D$, we get the eigenvalue decomposition of B:

$$B = T(I_{MN} - D\Lambda)T^*.$$

Since T is a block diagonal unitary matrix, it follows from Lemma 4.1b that

$$\begin{aligned} \|B\|_{\mathrm{b}\infty} &= \|I_{MN} - D\Lambda\|_{\mathrm{b}\infty} = \max_{1 \leq k \leq N} \|I_M - \mu_k \Lambda_k\| \\ &= \max_{1 \leq k \leq N} \max_{1 \leq m \leq M} |1 - \mu_k \lambda_{m,k}| \end{aligned}$$

where $\lambda_{m,k}$ is the mth diagonal of Λ_k. Therefore, we can see that $\|B\|_{\mathrm{b}\infty} < 1$ if, and only if,

$$0 < \mu_k < \frac{2}{\lambda_{\max}(R_{u,k})} \quad \text{for all } k = 1, 2, \ldots, N \tag{25}$$

where $\lambda_{\max}(R_{u,k})$ denotes the largest eigenvalue of $R_{u,k}$. It is well known that each condition for μ_k is a stability condition of the LMS algorithm in the mean [28].

Let us next consider the requirement $\|G_i\|_{\mathrm{b}\infty} \leq 1$ for all $i \geq 0$. By the definition of the block maximum norm and the triangle inequality, we have

$$\begin{aligned} \|G_i\|_{\mathrm{b}\infty} &= \max_{\substack{x \in \mathbb{C}^{MN} \\ x \neq 0}} \frac{\|G_i x\|_{\mathrm{b}\infty}}{\|x\|_{\mathrm{b}\infty}} \\ &= \max_{\substack{x \in \mathbb{C}^{MN} \\ x \neq 0}} \max_{1 \leq k \leq N} \left\| \sum_{\ell \in \mathcal{N}_k} \mathrm{E}[\boldsymbol{c}_{\ell k}(i)] x_\ell \right\| \cdot \frac{1}{\|x\|_{\mathrm{b}\infty}} \\ &\leq \max_{\substack{x \in \mathbb{C}^{MN} \\ x \neq 0}} \max_{1 \leq k \leq N} \sum_{\ell \in \mathcal{N}_k} |\mathrm{E}\boldsymbol{c}_{\ell k}(i)| \cdot \frac{\|x_\ell\|}{\|x\|_{\mathrm{b}\infty}} \\ &\leq \max_{1 \leq k \leq N} \sum_{\ell \in \mathcal{N}_k} |\mathrm{E}\boldsymbol{c}_{\ell k}(i)|. \end{aligned}$$

Therefore, it suffices to find conditions under which $\sum_{\ell \in \mathcal{N}_k} |\mathrm{E}\boldsymbol{c}_{\ell k}(i)| \leq 1$ for all $k \in \{1, \ldots, N\}$. Since $\sum_{\ell \in \mathcal{N}_k} \boldsymbol{c}_{\ell k}(i) = 1$, this condition is equivalent to

$$\sum_{\ell \in \mathcal{N}_k} |\mathrm{E}\boldsymbol{c}_{\ell k}(i)| \leq \sum_{\ell \in \mathcal{N}_k} \mathrm{E}\boldsymbol{c}_{\ell k}(i).$$

However, since the reverse inequality is always true, the condition to be satisfied is the following:

$$\sum_{\ell\in\mathcal{N}_k}|\mathrm{E}\boldsymbol{c}_{\ell k}(i)|=\sum_{\ell\in\mathcal{N}_k}\mathrm{E}\boldsymbol{c}_{\ell k}(i).$$

Obviously, this holds if and only if every $\mathrm{E}\boldsymbol{c}_{\ell k}(i)$ is nonnegative; otherwise we have

$$\sum_{\ell\in\mathcal{N}_k}|\mathrm{E}\boldsymbol{c}_{\ell k}(i)|-\sum_{\ell\in\mathcal{N}_k}\mathrm{E}\boldsymbol{c}_{\ell k}(i)=-2\sum_{\ell\in\mathcal{N}_k:\mathrm{E}\boldsymbol{c}_{\ell k}(i)<0}\mathrm{E}\boldsymbol{c}_{\ell k}(i)>0.$$

Moreover, $\mathrm{E}\boldsymbol{c}_{\ell k}(i)\geq 0$ is ensured if $\boldsymbol{c}_{\ell k}(i)$ is a nonnegative random variable. Hence, combining with $\sum_{\ell\in\mathcal{N}_k}\boldsymbol{c}_{\ell k}(i)=1$, we conclude that $\|G_i\|_{\mathrm{b}\infty}\leq 1$ is ensured if $\{\boldsymbol{c}_{\ell k}(i)\}_{\ell\in\mathcal{N}_k}$ is a convex combination for all $k\in\{1,\ldots,N\}$ and $i\geq 0$.

We summarize the above argument in the following:

Theorem 4.4 (Convergence): Under Assumptions 4.2 and 4.3, if every stepsize μ_k satisfies $0<\mu_k<2/\lambda_{\max}(R_{u,k})$ and if all nodes use possibly random, time-varying convex combinations at every time i, then every local estimate $\boldsymbol{\psi}_{k,i}$ generated by the CTA diffusion LMS algorithm converges in the mean to the unknown parameter w^o, i.e., $\mathrm{E}\boldsymbol{\psi}_{k,i}\to w^o$ as $i\to\infty$ for all $k\in\{1,\ldots,N\}$.

C. Stabilizing Effect

We derived from (23) one condition for the stability of the diffusion LMS algorithm with adaptive combiners. We further observe from (23) that cooperation via convex combination enhances stability in the mean. Since $\|I_{MN}\|_{\mathrm{b}\infty}=1$ and every convex combination satisfies $\|G_i\|_{\mathrm{b}\infty}\leq 1$, we have

$$\|B\|_{\mathrm{b}\infty}\cdot\|G_i\|_{\mathrm{b}\infty}\leq\|B\|_{\mathrm{b}\infty}\cdot\|I_{MN}\|_{\mathrm{b}\infty}.$$

The right hand side, which corresponds to the case of $\boldsymbol{C}_i=I_N$, represents the convergence rate of the diffusion LMS algorithm without cooperation among the nodes. Therefore, every convex combination has a stabilizing effect. The following is a remark on our mean-transient analysis.

Remark 4.5: The convexity of the combination weights is a weaker condition than those given earlier in [1] and [10] to ensure the stability in the mean. In the analyses of [1] and [10], the stability was analyzed using the matrix 2-norm. However, such an approach does not cover stability analysis for *asymmetric* combination weights, such as the relative degree rule in Table I, for which $\boldsymbol{C}_i$ is not symmetric. In contrast, our current analysis successfully covers asymmetric combination weights thanks to the use of the block maximum norm.

V. Mean-Square Transient Analysis

In this section, we analyze the mean-square performance of the CTA diffusion LMS algorithm with adaptive combiners using the energy conservation approach [9], [10], [23], [28], [31]. In [10], the CTA diffusion LMS algorithm with static combiners was analyzed. We generalize this analysis to the case of adaptive combiners. However, the analysis for adaptive combiners is more challenging because of the randomly evolving combination weights. To begin with, let us introduce mathematical tools useful for evaluating vectors and matrices with block structures.

A. Block Operations and Weighted Norm

Given an $MN\times MN$ block matrix

$$\Sigma=\begin{bmatrix}\Sigma_{11}&\cdots&\Sigma_{1N}\\ \vdots&\ddots&\vdots\\ \Sigma_{N1}&\cdots&\Sigma_{NN}\end{bmatrix}$$

with blocks $\Sigma_{\ell k}\in\mathbb{C}^{M\times M}$, the block vectorization operator bvec is defined as follows [29], [30]. After vectorizing each block via the standard vec operator, say $\sigma_{\ell k}=\mathrm{vec}\{\Sigma_{\ell k}\}$, $\mathrm{bvec}\{\Sigma\}$ is obtained by stacking all $\sigma_{\ell k}$'s as follows:

$$\mathrm{bvec}\{\Sigma\}\triangleq\mathrm{col}\{\sigma_{11},\ldots,\sigma_{N1};\ \sigma_{12},\ldots,\sigma_{N2};\ldots;\sigma_{1N},\ldots,\sigma_{NN}\}.$$

Namely, the bvec operator converts a block matrix Σ into a column vector of length M^2N^2 in this fashion. When $\sigma=\mathrm{bvec}\{\Sigma\}$ is given, we also write $\Sigma=\mathrm{bvec}\{\sigma\}$ to recover the original block matrix from the vector σ.

The block Kronecker product of two $MN\times MN$ block matrices A and B, denoted by $A\odot B$, is defined as the block matrix whose ℓkth block is given by the $M^2N\times M^2N$ matrix

$$[A\odot B]_{\ell k}\triangleq\begin{bmatrix}A_{\ell k}\otimes B_{11}&\ldots&A_{\ell k}\otimes B_{1N}\\ \vdots&\ddots&\vdots\\ A_{\ell k}\otimes B_{N1}&\ldots&A_{\ell k}\otimes B_{NN}\end{bmatrix}$$

where $\otimes$ denotes the standard Kronecker product. As a result, $A\odot B$ becomes an $M^2N^2\times M^2N^2$ block matrix. There are useful formulas for these block operations.

Fact 5.1: For any complex matrices A, Σ, and B of appropriate sizes, the following hold:

a) $\mathrm{bvec}\{A\Sigma B\}=(B^T\odot A)\mathrm{bvec}\{\Sigma\}$;

b) $\mathrm{Tr}(A^TB)=\mathrm{bvec}(A)^T\mathrm{bvec}(B)$.

Proof: These properties can be verified in a straightforward manner; see, e.g., [29] and [30]. ■

Finally, we define the weighted norm in a common way. Given a vector $x=\mathrm{col}\{x_1,\ldots,x_N\}\in\mathbb{C}^{MN}$ and an $MN\times MN$ positive definite matrix Σ, the weighted norm is defined as

$$\|x\|_\Sigma\triangleq\sqrt{x^*\Sigma x}.$$

Note that this is a seminorm in general when Σ is positive semidefinite. For convenience, when $\sigma=\mathrm{bvec}\{\Sigma\}$ is given, we also write $\|x\|_\sigma$ to represent $\|x\|_\Sigma$.

B. Performance Measures

We first define the performance measures we analyze. At each node k and time i, the *mean square deviation* (MSD), $\eta_k(i)$, and the *excess mean square error* (EMSE), $\zeta_k(i)$, are defined as:

$$\eta_k(i)\triangleq\mathrm{E}\|\tilde{\boldsymbol{\psi}}_{k,i-1}\|^2\quad\text{and}\ \zeta_k(i)\triangleq\mathrm{E}\,|\boldsymbol{e}_k^a(i)|^2\qquad(26)$$

respectively, where $\boldsymbol{e}_k^a(i)\triangleq\boldsymbol{u}_{k,i}\tilde{\boldsymbol{\psi}}_{k,i-1}$ is called the *a priori error* and the weight error vector $\tilde{\boldsymbol{\psi}}_{k,i-1}$ is defined in (20). Av-

eraging these quantities across the network, we introduce the *network MSD*, $\eta(i)$, and the *network EMSE*, $\zeta(i)$:

$$\eta(i) \triangleq \frac{1}{N}\sum_{k=1}^{N}\eta_k(i) \quad \text{and } \zeta(i) \triangleq \frac{1}{N}\sum_{k=1}^{N}\zeta_k(i). \tag{27}$$

Tracking the above four performances can be dealt with in a unified manner by studying the following weighted norm:

$$\mathrm{E}\|\tilde{\boldsymbol{\psi}}_i\|_{\Sigma}^2,$$

for suitably chosen $MN \times MN$ matrices Σ. For example, choosing $\Sigma = I_{MN}/N$ leads to the network MSD $\eta(i)$, while the network EMSE $\zeta(i)$ is obtained by setting $\Sigma = R_u/N$ under Assumption 4.2a, where R_u is defined in (22). On the other hand, the local MSD $\eta_k(i)$ or the local EMSE $\zeta_k(i)$ can be obtained by selecting

$$\begin{aligned} &\Sigma = \mathrm{diag}\{\mathbf{0}_{(k-1)M}, I_M, \mathbf{0}_{(N-k)M}\} \\ \text{or} \quad &\Sigma = \mathrm{diag}\{\mathbf{0}_{(k-1)M}, R_{u,k}, \mathbf{0}_{(N-k)M}\} \end{aligned}$$

respectively. Thus, we analyze the behavior of the weighted error $\mathrm{E}\|\tilde{\boldsymbol{\psi}}_i\|_{\Sigma}^2$. We first analyze the behavior for arbitrary random combiners and then derive the performance of the proposed combiner from Fig. 4.

C. Learning Behavior for General Adaptive Combiners

Let us start by transforming variables as follows. Using the block unitary matrix T given in (24), we introduce the following change of variables:

$$\begin{aligned} &\overline{\boldsymbol{\psi}}_i \triangleq T^*\tilde{\boldsymbol{\psi}}_i, \quad \bar{\boldsymbol{U}}_i \triangleq \boldsymbol{U}_i T, \\ &\bar{\boldsymbol{G}}_i \triangleq T^*\boldsymbol{G}_i T, \quad \overline{\Sigma} \triangleq T^*\Sigma T. \end{aligned}$$

Note that tracking $\mathrm{E}\|\tilde{\boldsymbol{\psi}}_i\|_{\Sigma}^2$ is done by tracking the transformed weighted error $\mathrm{E}\|\overline{\boldsymbol{\psi}}_i\|_{\overline{\Sigma}}^2$ for an arbitrary $MN \times MN$ weighting matrix $\overline{\Sigma}$ because $\mathrm{E}\|\tilde{\boldsymbol{\psi}}_i\|_{\Sigma}^2 = \mathrm{E}\|\overline{\boldsymbol{\psi}}_i\|_{T^*\Sigma T}^2$. Under this transformation, the matrix $\mathrm{E}[\bar{\boldsymbol{U}}_i^*\bar{\boldsymbol{U}}_i]$ becomes the diagonal matrix Λ given in (24). This will help us calculate moments in what follows.

Premultiplying (21) by T^* and noting that $T^*DT = D$, we get

$$\overline{\boldsymbol{\psi}}_i = (I_{MN} - D\bar{\boldsymbol{U}}_i^*\bar{\boldsymbol{U}}_i)\bar{\boldsymbol{G}}_i\overline{\boldsymbol{\psi}}_{i-1} - D\bar{\boldsymbol{U}}_i^*\boldsymbol{v}_i.$$

Furthermore, evaluating the weighted norms on both sides with an arbitrary weighting matrix $\overline{\Sigma}$ and then taking expectations under Assumption 4.2b, we obtain

$$\mathrm{E}\|\overline{\boldsymbol{\psi}}_i\|_{\overline{\Sigma}}^2 = \mathrm{E}\left(\|\overline{\boldsymbol{\psi}}_{i-1}\|_{\overline{\boldsymbol{\Sigma}}_i}^2\right) + \mathrm{E}[\boldsymbol{v}_i^*\bar{\boldsymbol{U}}_i D\overline{\Sigma}D\bar{\boldsymbol{U}}_i^*\boldsymbol{v}_i] \tag{28}$$

where $\overline{\boldsymbol{\Sigma}}_i$ is the $MN \times MN$ random matrix defined as

$$\overline{\boldsymbol{\Sigma}}_i \triangleq \bar{\boldsymbol{G}}_i^*(I_{MN} - D\bar{\boldsymbol{U}}_i^*\bar{\boldsymbol{U}}_i)^*\overline{\Sigma}(I_{MN} - D\bar{\boldsymbol{U}}_i^*\bar{\boldsymbol{U}}_i)\bar{\boldsymbol{G}}_i.$$

The second term of (28) can be evaluated as follows. Letting

$$\Lambda_v \triangleq \mathrm{diag}\{\sigma_{v,1}^2, \ldots, \sigma_{v,N}^2\}$$

and noting that $\boldsymbol{U}_i$ is block diagonal, we have

$$\begin{aligned} \mathrm{E}[\boldsymbol{v}_i^*\bar{\boldsymbol{U}}_i D\overline{\Sigma}D\bar{\boldsymbol{U}}_i^*\boldsymbol{v}_i] &= \mathrm{Tr}(D\mathrm{E}[\bar{\boldsymbol{U}}_i^*\boldsymbol{v}_i\boldsymbol{v}_i^*\bar{\boldsymbol{U}}_i]D\overline{\Sigma}) \\ &= \mathrm{Tr}(D\mathrm{E}[\bar{\boldsymbol{U}}_i^*\Lambda_v\bar{\boldsymbol{U}}_i]D\overline{\Sigma}) \\ &= \mathrm{Tr}(D(\Lambda_v \otimes I_M)\mathrm{E}[\bar{\boldsymbol{U}}_i^*\bar{\boldsymbol{U}}_i]D\overline{\Sigma}) \\ &= \mathrm{Tr}(D^2\Lambda(\Lambda_v \otimes I_M)\overline{\Sigma}) \end{aligned}$$

where the last equality follows from the fact that D, Λ and $\Lambda_v \otimes I_M$ are all diagonal. Hence, (28) leads to

$$\mathrm{E}\|\overline{\boldsymbol{\psi}}_i\|_{\overline{\Sigma}}^2 = \mathrm{E}\left(\|\overline{\boldsymbol{\psi}}_{i-1}\|_{\overline{\boldsymbol{\Sigma}}_i}^2\right) + \mathrm{Tr}(D^2\Lambda(\Lambda_v \otimes I_M)\overline{\Sigma}).$$

In general, this recursion is mathematically intractable due to the random, time-varying weighting matrix $\overline{\boldsymbol{\Sigma}}_i$ on the right-hand side. However, using Assumption 4.3 again, $\overline{\boldsymbol{\Sigma}}_i$ becomes independent of $\overline{\boldsymbol{\psi}}_{i-1}$. Namely, $\overline{\boldsymbol{\Sigma}}_i$ can be replaced by its expectation $\overline{\Sigma}_i \triangleq \mathrm{E}\overline{\boldsymbol{\Sigma}}_i$, and, hence we have

$$\begin{aligned} \mathrm{E}\|\overline{\boldsymbol{\psi}}_i\|_{\overline{\Sigma}}^2 &= \mathrm{E}\|\overline{\boldsymbol{\psi}}_{i-1}\|_{\overline{\Sigma}_i}^2 + \mathrm{Tr}(D^2\Lambda(\Lambda_v \otimes I_M)\overline{\Sigma}) \\ \overline{\Sigma}_i &= \mathrm{E}[\bar{\boldsymbol{G}}_i^*\{\overline{\Sigma} - \Lambda D\overline{\Sigma} - \overline{\Sigma}D\Lambda \\ &\qquad + \mathrm{E}[\bar{\boldsymbol{U}}_i^*\bar{\boldsymbol{U}}_i D\overline{\Sigma}D\bar{\boldsymbol{U}}_i^*\bar{\boldsymbol{U}}_i]\}\bar{\boldsymbol{G}}_i]. \end{aligned}$$

To proceed further, let us convert $\overline{\Sigma}$ into $\bar{\sigma} \triangleq \mathrm{bvec}\{\overline{\Sigma}\}$ and rewrite this recursion using the equivalent notation $\|\overline{\boldsymbol{\psi}}_i\|_{\bar{\sigma}}$ instead of $\|\overline{\boldsymbol{\psi}}_i\|_{\overline{\Sigma}}$. Using Lemma 5.1 to express $\mathrm{bvec}\{\overline{\Sigma}_i\}$ and $\mathrm{Tr}(D^2\Lambda(\Lambda_v \otimes I_M)\overline{\Sigma})$ with $\bar{\sigma}$, we have

$$\boxed{\mathrm{E}\|\overline{\boldsymbol{\psi}}_i\|_{\bar{\sigma}}^2 = \mathrm{E}\|\overline{\boldsymbol{\psi}}_{i-1}\|_{\bar{F}_i\bar{\sigma}}^2 + b^T\bar{\sigma}} \tag{29}$$

where

$$\begin{aligned} \bar{F}_i &\triangleq \mathrm{E}\left[\bar{\boldsymbol{G}}_i^T \odot \bar{\boldsymbol{G}}_i^*\right][I_{M^2N^2} - (I_{MN} \odot \Lambda D) \\ &\qquad - (\Lambda D \odot I_{MN}) + \mathcal{A}(D \odot D)] \\ \mathcal{A} &\triangleq \mathrm{E}[(\bar{\boldsymbol{U}}_i^*\bar{\boldsymbol{U}}_i)^T \odot \bar{\boldsymbol{U}}_i^*\bar{\boldsymbol{U}}_i] \\ b &\triangleq \mathrm{bvec}\{D^2\Lambda(\Lambda_v \otimes I_M)\}. \end{aligned} \tag{30}$$

Fortunately, in the case where the regressors $\{\boldsymbol{u}_{k,i}\}$ are circular Gaussian, by the Gaussian factorization theorem [28], the fourth order moment $\mathcal{A}$ can be expressed in closed form as follows [10]:

$$\begin{aligned} &\mathcal{A} = \mathrm{diag}\{\mathcal{A}_1, \ldots, \mathcal{A}_N\} \\ &\mathcal{A}_k \triangleq \mathrm{diag}\left\{\mathcal{A}_k^{(1)}, \ldots, \mathcal{A}_k^{(N)}\right\} \\ &\mathcal{A}_k^{(\ell)} \triangleq \begin{cases} \Lambda_\ell \otimes \Lambda_\ell, & \text{if } \ell \neq k \\ \lambda_k\lambda_k^T + \gamma\Lambda_k \otimes \Lambda_k, & \text{if } \ell = k \end{cases} \end{aligned} \tag{31}$$

where $\lambda_k \triangleq \mathrm{vec}\{\Lambda_k\}$ and $\gamma = 1$ for complex regressors and $\gamma = 2$ for real regressors. However, in our analysis, we do not

assume that $\{\boldsymbol{u}_{k,i}\}$ are circular Gaussian but just assume that $\mathcal{A}$ is known.

Now, iterating (29) and using the fact that every node sets the initial estimate to $\boldsymbol{\psi}_{k,-1}=0$, we have

$$\begin{aligned}
\mathrm{E}\|\overline{\boldsymbol{\psi}}_i\|^2_{\bar{\sigma}} &= \mathrm{E}\|\overline{\boldsymbol{\psi}}_{i-1}\|^2_{\bar{F}_i\bar{\sigma}} + b^T\bar{\sigma}\\
\mathrm{E}\|\overline{\boldsymbol{\psi}}_{i-1}\|^2_{\bar{F}_i\bar{\sigma}} &= \mathrm{E}\|\overline{\boldsymbol{\psi}}_{i-2}\|^2_{\bar{F}_{i-1}\bar{F}_i\bar{\sigma}} + b^T\bar{F}_i\bar{\sigma}\\
&\vdots\\
\mathrm{E}\|\overline{\boldsymbol{\psi}}_1\|^2_{\bar{F}_2\bar{F}_3\ldots\bar{F}_i\bar{\sigma}} &= \mathrm{E}\|\overline{\boldsymbol{\psi}}_0\|^2_{\bar{F}_1\bar{F}_2\bar{F}_3\ldots\bar{F}_i\bar{\sigma}} + b^T\bar{F}_2\bar{F}_3\ldots\bar{F}_i\bar{\sigma}\\
\mathrm{E}\|\overline{\boldsymbol{\psi}}_0\|^2_{\bar{F}_1\bar{F}_2\ldots\bar{F}_i\bar{\sigma}} &= \|\overline{w}^{(o)}\|^2_{\bar{F}_0\bar{F}_1\bar{F}_2\ldots\bar{F}_i\bar{\sigma}} + b^T\bar{F}_1\bar{F}_2\ldots\bar{F}_i\bar{\sigma}
\end{aligned}$$

where $\overline{w}^{(o)}=T^*w^{(o)}$. Summing up these equations, we get

$$\begin{aligned}
\mathrm{E}\|\overline{\boldsymbol{\psi}}_i\|^2_{\bar{\sigma}} = \|\overline{w}^{(o)}\|^2_{\bar{F}_0\bar{F}_1\ldots\bar{F}_i\bar{\sigma}}\\
+b^T\left\{I_{M^2N^2}+\sum_{j=1}^{i}\bar{F}_j\bar{F}_{j+1}\ldots\bar{F}_i\right\}\bar{\sigma}.
\end{aligned}$$

Hence, we finally obtain the following recursion for the weighted error:

$$\boxed{\begin{aligned}
\mathrm{E}\|\overline{\boldsymbol{\psi}}_i\|^2_{\bar{\sigma}} &= \mathrm{E}\|\overline{\boldsymbol{\psi}}_{i-1}\|^2_{\bar{\sigma}} + b^T(\bar{\mathcal{G}}_i-\bar{\mathcal{G}}_{i-1})\bar{\sigma}\\
&\qquad -\|\overline{w}^{(o)}\|^2_{(\bar{\mathcal{F}}_{i-1}-\bar{\mathcal{F}}_i)\bar{\sigma}}\\
\bar{\mathcal{F}}_i &\triangleq \begin{cases}\bar{\mathcal{F}}_{i-1}\bar{F}_i, & \text{for } i\geq 0,\\ I_{M^2N^2}, & \text{for } i=-1,\end{cases}\\
\bar{\mathcal{G}}_i &\triangleq \begin{cases}(I_{M^2N^2}+\bar{\mathcal{G}}_{i-1})\bar{F}_i, & \text{for } i\geq 0,\\ \mathbf{0}_{M^2N^2}, & \text{for } i=-1.\end{cases}
\end{aligned}} \tag{32}$$

This is a generalization of the result in [10]. Using the matrices Σ mentioned in the previous subsection and setting $\bar{\sigma}=\mathrm{bvec}\{T^*\Sigma T\}$, we can obtain the performance of *any* adaptive combiner, provided that the matrix $\bar{F}_i$ involving the moment of the combination matrix $\boldsymbol{C}_i$ is known for every i. Unfortunately, it is impossible to derive $\bar{F}_i$ without knowing the update rule of adaptive combiners. Therefore, we continue the analysis for the proposed adaptive combiner. We first summarize the above argument for general combiners.

Theorem 5.2 (Learning Curves for Adaptive Combiners): Suppose that Assumptions 4.2 and 4.3 hold. Let $q\triangleq \mathrm{bvec}\{I_{MN}\}/N$ and $\lambda\triangleq\mathrm{bvec}\{\Lambda\}/N$, where Λ is obtained from the eigenvalue decomposition $R_u=T\Lambda T^*$. Then, the network MSD, $\eta(i)$, and the network EMSE, $\zeta(i)$, evolve according to

$$\begin{aligned}
\eta(i) &= \eta(i-1)+b^T(\bar{\mathcal{G}}_i-\bar{\mathcal{G}}_{i-1})q - \|\overline{w}^{(o)}\|^2_{(\bar{\mathcal{F}}_{i-1}-\bar{\mathcal{F}}_i)q}\\
\zeta(i) &= \zeta(i-1)+b^T(\bar{\mathcal{G}}_i-\bar{\mathcal{G}}_{i-1})\lambda - \|\overline{w}^{(o)}\|^2_{(\bar{\mathcal{F}}_{i-1}-\bar{\mathcal{F}}_i)\lambda}
\end{aligned}$$

with initial conditions $\eta(-1)=\|w^o\|^2$ and $\zeta(-1)=\|\overline{w}^{(o)}\|^2_\Lambda/N$, respectively. Here, b is defined in (30), $\overline{w}^{(o)}=T^*w^{(o)}$, and the matrices $\bar{\mathcal{F}}_i$ and $\bar{\mathcal{G}}_i$ are recursively calculated as in (32) using the matrix $\bar{F}_i$ defined in (30). Similarly, the local MSD and EMSE at node k evolve according to

$$\begin{aligned}
\eta_k(i) &= \eta_k(i-1)+b^T(\bar{\mathcal{G}}_i-\bar{\mathcal{G}}_{i-1})j_{q,k} - \|\overline{w}^{(o)}\|^2_{(\bar{\mathcal{F}}_{i-1}-\bar{\mathcal{F}}_i)j_{q,k}}\\
\zeta_k(i) &= \zeta_k(i-1)+b^T(\bar{\mathcal{G}}_i-\bar{\mathcal{G}}_{i-1})j_{\lambda,k} - \|\overline{w}^{(o)}\|^2_{(\bar{\mathcal{F}}_{i-1}-\bar{\mathcal{F}}_i)j_{\lambda,k}}
\end{aligned}$$

with initial conditions $\eta_k(-1)=\|w^o\|^2$ and $\zeta_k(-1)=\|w^o\|^2_{R_{u,k}}$, respectively, and where the vectors $j_{q,k}$ and $j_{\lambda,k}$ are given by

$$\begin{aligned}
j_{q,k} &\triangleq \mathrm{bvec}\{\mathrm{diag}\{\mathbf{0}_{(k-1)M}, I_M, \mathbf{0}_{(N-k)M}\}\}\\
j_{\lambda,k} &\triangleq \mathrm{bvec}\{\mathrm{diag}\{\mathbf{0}_{(k-1)M}, \Lambda_k, \mathbf{0}_{(N-k)M}\}\}.
\end{aligned}$$

Remark: If the regressors are circular Gaussian, the fourth order moment $\mathcal{A}$ in (30) can be calculated as in (31).

D. Learning Behavior of the Proposed Combiner

Let us now perform the evaluation of $\bar{F}_i$ for the proposed combiner with the normalized stepsize (18). In view of (30), it suffices to evaluate the time-varying moment $\mathrm{E}[\bar{\boldsymbol{G}}_i^T\odot\bar{\boldsymbol{G}}_i^*]$. However, this is challenging because the combination matrix $\boldsymbol{C}_i$ and the local estimates $\boldsymbol{\psi}_{k,i}$ are mutually dependent. However, we tackle this problem by utilizing approximations.

To begin with, let us rewrite (16) and (18) as expressions in terms of $\boldsymbol{c}_{k,i}$ by eliminating $\boldsymbol{b}_{k,i}$. By the definition of P_k (see (6)), it can be verified that

$$\min\left\{\boldsymbol{b}^{(m)}_{k,i-1}\,|\,1\leq m\leq n_k\right\}=\min\left\{\boldsymbol{c}^{(m)}_{k,i-1}\,|\,m\in\mathcal{N}_k\right\}$$

and that $\|\boldsymbol{g}_{k,i}\|_\infty=\|\hat{\boldsymbol{g}}_{k,i}\|_\infty$, where $\hat{\boldsymbol{g}}_{k,i}\triangleq P_k\boldsymbol{g}_{k,i}$. Hence, multiplying (16) from left by P_k and noting that $\Delta\boldsymbol{\Psi}_{k,i-1}=\Delta\boldsymbol{\Psi}_{i-1}P_k$, we see that $\boldsymbol{c}_{k,i}$ evolves according to

$$\begin{cases}
\hat{\boldsymbol{Q}}_{i-1}=\mathrm{Re}\{(\Delta\boldsymbol{\Psi}_{i-1})^*\Delta\boldsymbol{\Psi}_{i-1}\}\\
\hat{\boldsymbol{g}}_{k,i}=P_k\left(I_{n_k}-\dfrac{\mathbb{1}_{n_k}\mathbb{1}_{n_k}^T}{n_k}\right)P_k^T\hat{\boldsymbol{Q}}_{i-1}\boldsymbol{c}_{k,i-1}\\
\hat{\boldsymbol{\nu}}_k(i)=\alpha\dfrac{\min\left\{\boldsymbol{c}^{(m)}_{k,i-1}\,|\,m\in\mathcal{N}_k\right\}}{\|\hat{\boldsymbol{g}}_{k,i}\|_\infty+\varepsilon}\\
\boldsymbol{c}_{k,i}=\boldsymbol{c}_{k,i-1}-\boldsymbol{\nu}_k(i)\hat{\boldsymbol{g}}_{k,i}.
\end{cases} \tag{33}$$

These equations show that the combination weight matrix $\boldsymbol{C}_i$ depends on the quadratic form of $\Delta\boldsymbol{\Psi}_{i-1}$ and, hence, the moment $\mathrm{E}[\bar{\boldsymbol{G}}_i^T\odot\bar{\boldsymbol{G}}_i^*]$ involves fourth order quantities of $\Delta\boldsymbol{\Psi}_{i-1}$, which is difficult to evaluate. To overcome this difficulty, let us assume the following:

Assumption 5.3: The mean vector $\mathrm{E}\boldsymbol{c}_{k,i}$ evolves according to the system that is obtained by replacing the random variables in (33) with their expectations, that is,

$$\begin{cases}
\hat{Q}_{i-1}=\mathrm{Re}\{\mathrm{E}[(\Delta\boldsymbol{\Psi}_{i-1})^*\Delta\boldsymbol{\Psi}_{i-1}]\}\\
\hat{g}_{k,i}=P_k\left(I_{n_k}-\dfrac{\mathbb{1}_{n_k}\mathbb{1}_{n_k}^T}{n_k}\right)P_k^T\hat{Q}_{i-1}\mathrm{E}\boldsymbol{c}_{k,i-1}\\
\hat{\nu}_k(i)=\alpha\dfrac{\min\left\{\mathrm{E}\boldsymbol{c}^{(m)}_{k,i-1}\,|\,m\in\mathcal{N}_k\right\}}{\|\hat{g}_{k,i}\|_\infty+\varepsilon}\\
\mathrm{E}\boldsymbol{c}_{k,i}=\mathrm{E}\boldsymbol{c}_{k,i-1}-\hat{\nu}_k(i)\hat{g}_{k,i}.
\end{cases} \tag{34}$$

Moreover, we use $\mathrm{E}[\bar{\boldsymbol{G}}_i^T\odot\bar{\boldsymbol{G}}_i^*]\approx\mathrm{E}[\bar{\boldsymbol{G}}_i]^T\odot\mathrm{E}[\bar{\boldsymbol{G}}_i]^*$.

Although this assumption is unrealistic, $\mathrm{E}\boldsymbol{c}_{k,i}$ is guaranteed to satisfy at least the constraints $\mathbb{1}_N^T \mathrm{E}\boldsymbol{c}_{k,i} = 1$ and $\mathrm{E}\boldsymbol{c}_{\ell k}(i) = 0$ for $\ell \notin \mathcal{N}_k$. Since $\mathrm{E}[\bar{\boldsymbol{G}}_i]$ can be calculated by

$$\begin{aligned} \mathrm{E}[\bar{\boldsymbol{G}}_i] &= T^*(\mathrm{E}[\boldsymbol{C}_i]^T \otimes I_M)T \\ \mathrm{E}[\boldsymbol{C}_i] &= [\mathrm{E}\boldsymbol{c}_{1,i}, \ldots, \mathrm{E}\boldsymbol{c}_{N,i}] \end{aligned} \tag{35}$$

the remaining task is to calculate the moment $\hat{Q}_{i-1}$. To calculate the ℓkth component of $\hat{Q}_{i-1}$, which is given by

$$\mathrm{E}(\boldsymbol{\psi}_{\ell,i-1} - \boldsymbol{\psi}_{\ell,i-2})^*(\boldsymbol{\psi}_{k,i-1} - \boldsymbol{\psi}_{k,i-2})$$

we employ the following assumption:

Assumption 5.4: For each time i, the *a priori* error $\boldsymbol{e}_k^a(i) = \boldsymbol{u}_{k,i}\tilde{\boldsymbol{\psi}}_{k,i-1}$ is spatially independent and independent of the regressor $\boldsymbol{u}_{k,i}$. Furthermore, we use the approximation $\boldsymbol{\psi}_{k,i-2} \approx \boldsymbol{\phi}_{k,i-2}$.

The independence between $\boldsymbol{e}_k^a(i)$ and $\boldsymbol{u}_{k,i}$ is a common assumption in the analysis of adaptive filters [28], [31]. Now, by using the approximation $\boldsymbol{\psi}_{k,i-2} \approx \boldsymbol{\phi}_{k,i-2}$, it follows from (17) that

$$\begin{aligned} \boldsymbol{\psi}_{k,i-1} &\approx \boldsymbol{\psi}_{k,i-2} + \mu_k\{\boldsymbol{d}_k(i-1) - \boldsymbol{u}_{k,i-1}\boldsymbol{\psi}_{k,i-2}\}\boldsymbol{u}_{k,i-1}^* \\ &= \boldsymbol{\psi}_{k,i-2} + \mu_k\left\{\boldsymbol{e}_k^a(i-1) + \boldsymbol{v}_k(i-1)\right\}\boldsymbol{u}_{k,i-1}^* \end{aligned}$$

where the second equality follows from the data model $\boldsymbol{d}_k(i-1) = \boldsymbol{u}_{k,i-1}w^o + \boldsymbol{v}_k(i-1)$. This gives, under Assumption 5.4, that

$$\mathrm{E}\|\boldsymbol{\psi}_{k,i-1} - \boldsymbol{\psi}_{k,i-2}\|^2 = \mu_k^2\left\{\zeta_k(i-1) + \sigma_{v,k}^2\right\}\mathrm{Tr}(R_{u,k}).$$

Also, the spatial independence of the regressors $\{\boldsymbol{u}_{k,i}\}$ yields

$$\mathrm{E}(\boldsymbol{\psi}_{\ell,i-1} - \boldsymbol{\psi}_{\ell,i-2})^*(\boldsymbol{\psi}_{k,i-1} - \boldsymbol{\psi}_{k,i-2}) = 0$$

for $\ell \neq k$. Therefore, $\hat{Q}_{i-1}$ can be approximated by

$$\begin{aligned} \hat{Q}_{i-1} = \mathrm{diag}\big\{&\mu_1^2\left\{\zeta_1(i-1) + \sigma_{v,1}^2\right\}\mathrm{Tr}(R_{u,1}), \ldots, \\ &\mu_N^2\left\{\zeta_N(i-1) + \sigma_{v,N}^2\right\}\mathrm{Tr}(R_{u,N})\big\}. \end{aligned} \tag{36}$$

Combining this approximation and Theorem 5.2, we can track the mean-square performance of the CTA diffusion LMS with the proposed combiners. The following is the summary of the performance evaluation for the proposed combiner:

Theorem 5.5 (Learning Curves for the Proposed Combiner): Under Assumptions 4.2, 4.3, 5.3 and 5.4, the mean-square performance of the CTA diffusion LMS algorithm with the proposed combiner can be evaluated as follows. For each $k \in \{1, \ldots, N\}$, initialize $\zeta_k(-1)$ as in Theorem 5.2 and set $\mathrm{E}\boldsymbol{c}_{k,-1} = c_{k,-1}$, where $c_{k,-1}$ is the initial combination weight vector. Then, repeat the following for every $i \geq 0$:

1) calculate $\hat{Q}_{i-1}$ via (36);
2) for each k, update $\mathrm{E}\boldsymbol{c}_{k,i-1}$ via (34);
3) calculate $\mathrm{E}[\bar{\boldsymbol{G}}_i]$ via (35) to approximate $\bar{F}_i$ in (30) by

$$\begin{aligned} \bar{F}_i \approx [\mathrm{E}[\bar{\boldsymbol{G}}_i]^T \odot \mathrm{E}[\bar{\boldsymbol{G}}_i]^*][I_{M^2N^2} - (I_{MN} \odot \Lambda D) \\ -(\Lambda D \odot I_{MN}) + \mathcal{A}(D \odot D)]; \end{aligned}$$

4) use $\bar{F}_i$ to update $\bar{\mathcal{F}}_i$ and $\bar{\mathcal{G}}_i$ in Theorem 5.2 and then update every local EMSE $\zeta_k(i-1)$ and the other performances.

E. Steady-State Mean-Square Performance

Let us now evaluate the steady-state MSD and EMSE, which are defined as the limits of $\eta(i)$ and $\zeta(i)$, respectively. One possible way is to calculate $\eta(i)$ and $\zeta(i)$ iteratively according to Theorem 5.5 until they converge. However, this may take a long time and does not clarify the relation between the parameters and the steady-state performance. Thus, we seek closed form expressions for the steady-state performance.

Now, let us return to (29). In view of this, if the matrix $\bar{F}_i$ can be approximated by some constant matrix $\bar{F}$, then we would have

$$\mathrm{E}\|\overline{\boldsymbol{\psi}}_i\|_{\bar{\sigma}}^2 \approx \mathrm{E}\|\overline{\boldsymbol{\psi}}_{i-1}\|_{\bar{F}\bar{\sigma}}^2 + b^T\bar{\sigma}.$$

Evaluating this equation at the limit, we get

$$\mathrm{E}\|\overline{\boldsymbol{\psi}}_\infty\|_{\bar{\sigma}}^2 = \mathrm{E}\|\overline{\boldsymbol{\psi}}_\infty\|_{\bar{F}\bar{\sigma}}^2 + b^T\bar{\sigma}$$

or

$$\mathrm{E}\|\overline{\boldsymbol{\psi}}_\infty\|_{(I_{M^2N^2}-\bar{F})\bar{\sigma}}^2 = b^T\bar{\sigma}.$$

Hence, replacing the free parameter $\bar{\sigma}$ by $(I_{M^2N^2} - \bar{F})^{-1}\bar{\sigma}$, we get

$$\mathrm{E}\|\overline{\boldsymbol{\psi}}_\infty\|_{\bar{\sigma}}^2 = b^T(I_{M^2N^2} - \bar{F})^{-1}\bar{\sigma}. \tag{37}$$

This provides the steady-state mean-square performance by choosing $\bar{\sigma}$ as before. Therefore, let us find a good approximation for $\bar{F}_i$.

Recall that the time-dependent factor of $\bar{F}_i$ is $\mathrm{E}[\bar{\boldsymbol{G}}_i^T \odot \bar{\boldsymbol{G}}_i^*]$. We approximate this by using $\hat{Q}_{i-1}$ defined in (36). In view of (36), if every local EMSE is sufficiently small, then by ignoring $\zeta_k(i-1)$ we have a reasonable approximation. Namely, we assume that

$$\hat{Q}_{i-1} \approx \hat{Q} \triangleq \mathrm{diag}\left\{\mu_1^2\sigma_{v,1}^2\mathrm{Tr}(R_{u,1}), \ldots, \mu_N^2\sigma_{v,N}^2\mathrm{Tr}(R_{u,N})\right\}$$

for sufficiently large i. Moreover, comparing (34) with the original algorithm (14) and the optimal solution (9), we assume that $\mathrm{E}\boldsymbol{c}_{k,i}$ converges to

$$\hat{c}_k^o \triangleq \frac{P_k\hat{Q}_k^{-1}\mathbb{1}_{n_k}}{\mathbb{1}_{n_k}^T\hat{Q}_k^{-1}\mathbb{1}_{n_k}}, \quad \text{where } \hat{Q}_k \triangleq P_k^T\hat{Q}P_k. \tag{38}$$

Finally, letting

$$C^o \triangleq [\hat{c}_1^o, \ldots, \hat{c}_N^o] \quad \text{and } \bar{G}^o \triangleq T^*[(C^o)^T \otimes I_M]T$$

and approximating $\mathrm{E}[\bar{\boldsymbol{G}}_i]^T \odot \mathrm{E}[\bar{\boldsymbol{G}}_i]^*$ by $(\bar{G}^o)^T \odot (\bar{G}^o)^*$, we obtain the following approximation:

$$\begin{aligned} \bar{F}_i \approx \bar{F} \triangleq [(\bar{G}^o)^T \odot (\bar{G}^o)^*][I_{M^2N^2} - (I_{MN} \odot \Lambda D) \\ -(\Lambda D \odot I_{MN}) + \mathcal{A}(D \odot D)]. \end{aligned} \tag{39}$$

The use of this approximation with (37) leads to the following result.

Theorem 5.7 (Steady-State Mean-Square Performance): Suppose that $\mathrm{E}\boldsymbol{c}_{k,i}$ converges to $\hat{c}_k^o$ defined in (38) and the matrix $\bar{F}$ defined in (39) is such that $I_{M^2N^2} - \bar{F}$ is nonsingular. Then, the steady-state network MSD, η, and the steady-state network EMSE, ζ, are approximately given as follows:

$$\eta = (I_{M^2N^2} - \bar{F})^{-1} q, \quad \zeta = (I_{M^2N^2} - \bar{F})^{-1} \lambda$$

where q and λ are defined as in Theorem 5.2. Similarly, the steady-state MSD, η_k, and the steady-state EMSE, ζ_k, of node k are approximately given as follows:

$$\eta_k = (I_{M^2N^2} - \bar{F})^{-1} j_{q,k}, \quad \zeta_k = (I_{M^2N^2} - \bar{F})^{-1} j_{\lambda,k}$$

where $j_{q,k}$ and $j_{\lambda,k}$ are also defined as in Theorem 5.2.

Remark 5.7: Although we focused on the analysis of CTA diffusion, the analysis of ATC version can be done in a similar way. In [14], performance analysis of more general CTA and ATC diffusion LMS algorithms equipped with static combiners has been done.

VI. Simulation Results

In this section, we show the performance of the proposed adaptive combiner and compare our performance analysis with experiments.

A. Performance Comparison

We consider two simulation scenarios, say Examples 1 and 2, over different network topologies. In both examples, we compare the CTA and ATC diffusion LMS algorithms with the following combiners: i) the proposed adaptive combiner, ii) the static combiners shown in Table I, and iii) the adaptive combiner based on convex combination of two adaptive filters (see for details [8]). We also compare these diffusion algorithms with the incremental LMS algorithm, which cycles estimates along a predefined cyclic path (see [9] for details).

In Example 1, we consider a network topology with $N = 15$ nodes shown in Fig. 5. The unknown vector is set to $w^o = \mathbb{1}_5/\sqrt{5} (M = 5)$. Both the regressors and the noise are zero-mean Gaussian, i.i.d. in time and independent in space. Their statistics are shown in Fig. 5. The stepsize of the LMS algorithm used at each node k is set to $\mu_k = 0.01$, except for the incremental LMS algorithm. For the incremental LMS algorithm, the stepsize is set to $\mu_k = 0.01/N$ because the incremental LMS algorithm uses the LMS-type iterations N times for every i [9]. For the proposed combiner, $b_{k,-1}$ is set to $\mathbb{1}_{n_k}/n_k$ for every k and the normalized stepsize rule (18) with $\alpha = 0.8$ and $\varepsilon = 10^{-3}$ is used.

Fig. 6(a) and (b) shows the learning behavior of each algorithm in terms of the network MSD, which is defined in (27). The expectation is calculated by averaging 200 independent experiments. We observe that the proposed algorithm outperforms the other algorithms at the steady-state, although the convergence speed is a little slower than the others. On the other hand, Fig. 6(c) and (d) shows the steady-state MSD at each node, which is obtained by averaging the last 100 samples after a sufficient number of iterations. From Fig. 6, we find out that the ATC diffusion LMS algorithms outperform the CTA versions.

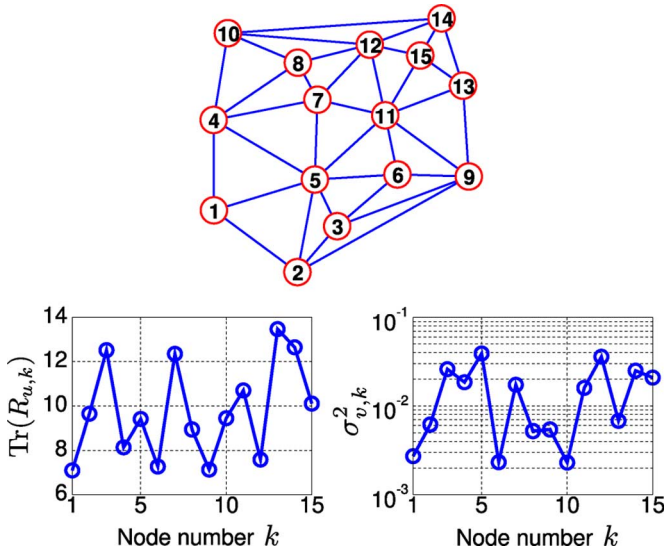


Fig. 5. Network settings for Example 1: topology with $N = 15$ nodes and 35 links (top), regressor statistics (bottom left) and noise variances (bottom right). In the incremental LMS algorithm, we used the following cyclic path: $1 \rightarrow 2 \rightarrow 3 \rightarrow 5 \rightarrow 6 \rightarrow 9 \rightarrow 13 \rightarrow 14 \rightarrow 15 \rightarrow 11 \rightarrow 12 \rightarrow 7 \rightarrow 8 \rightarrow 10 \rightarrow 4 \rightarrow 1$.

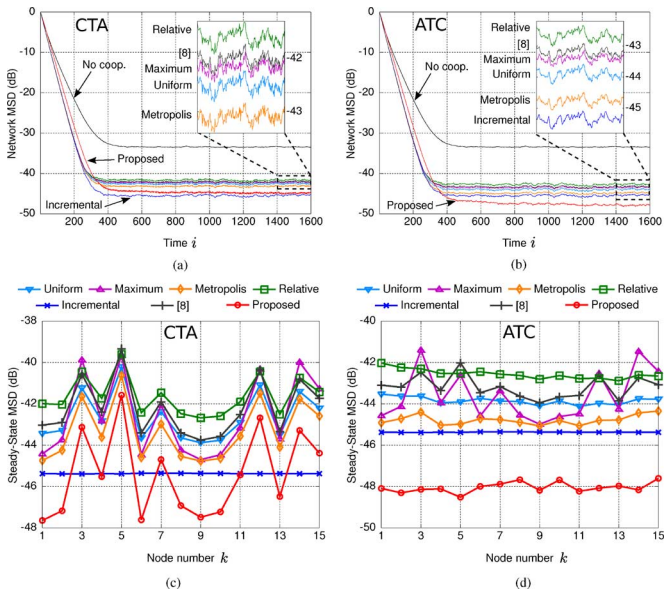


Fig. 6. Performance results for Example 1. (a) and (b): Learning curves of network MSD. (c) and (d): Steady-state MSD. (a) CTA; (b) ATC; (c) CTA; and (d) ATC.

To see why the algorithm is robust, we show in Fig. 7 the relation between the combination weights of the CTA diffusion algorithms at the steady-state and the SNR for the CTA diffusion algorithms. The SNR at node k was calculated by $10\log_{10}(\mathrm{E}|\boldsymbol{u}_{k,i}w^o|^2/\sigma^2_{v,k})$. We omit the results for ATC versions because no significant difference was observed between the weights of CTA and ATC diffusion algorithms. In our adaptive combiner, every neighbor of node 5 successfully puts a small weight on node 5, at which the SNR is relatively lower than the others. The reverse situation is observed for node 10, at which the SNR is relatively high. Such a successful control leads to the robustness of our combiner.

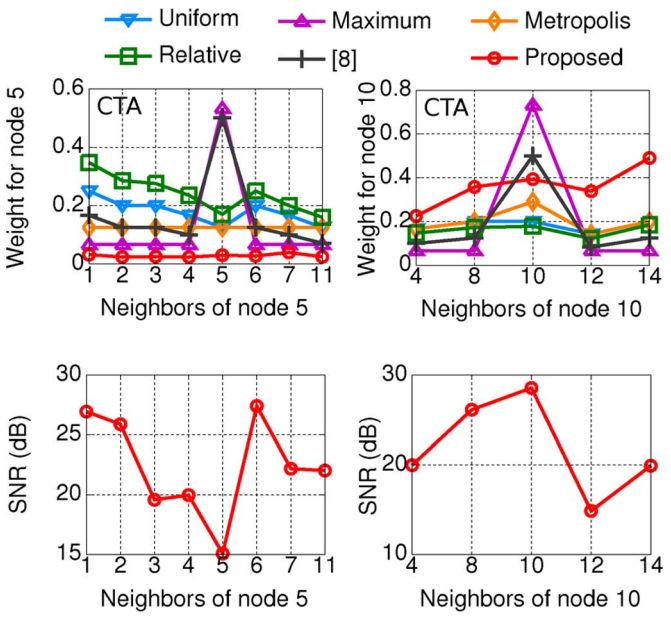


Fig. 7. Combination weights $c_{5k}(i)$ and $c_{10k}(i)$ of the CTA diffusion algorithms at steady-state in Example 1. The result for the ATC versions is omitted because almost the same results were observed.

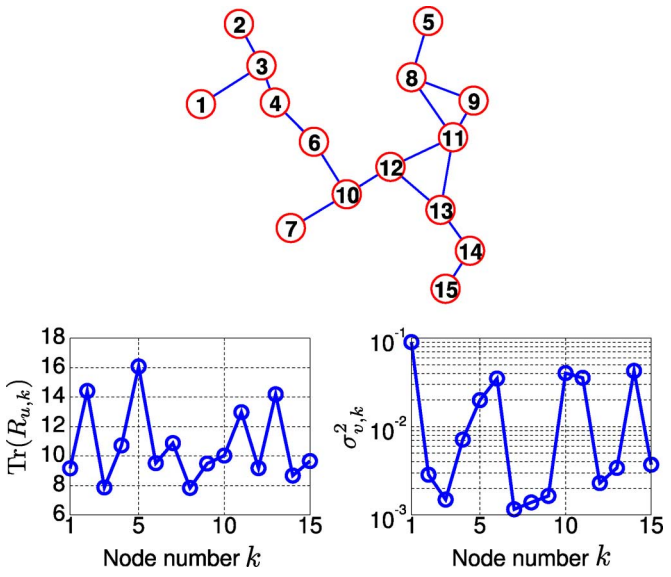


Fig. 8. Network settings for Example 2: topology with $N = 15$ nodes and 16 links (top), regressor statistics (bottom left) and noise variances (bottom right). In this topology, there exists no cycle graph.

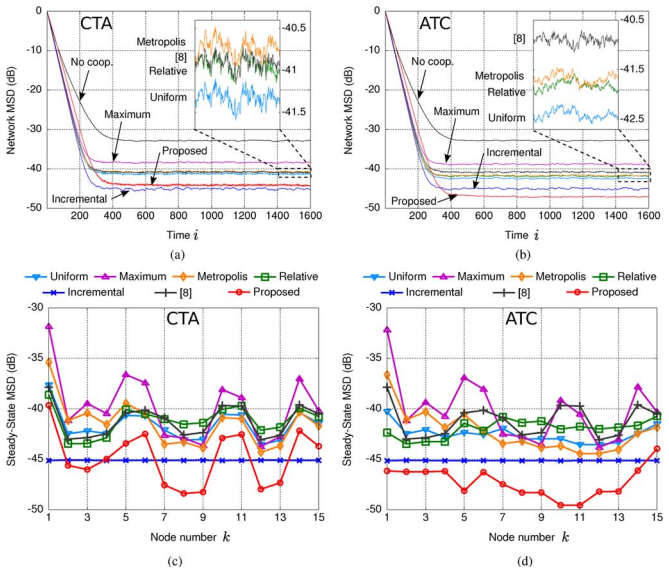


Fig. 9. Performance results for Example 2. (a) and (b): Learning curves of network MSD. (c) and (d): Steady-state MSD. (a) CTA; (b) ATC; (c) CTA; and (d) ATC.

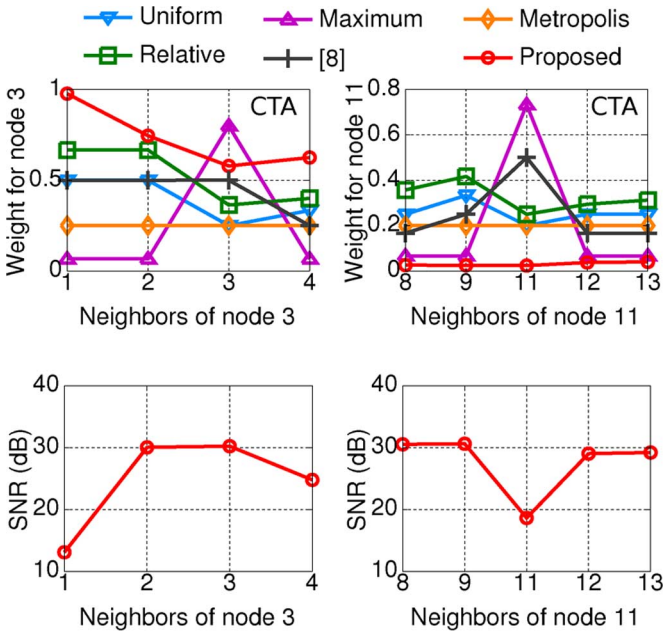


Fig. 10. Combination weights $c_{3k}(i)$ and $c_{11k}(i)$ of the CTA diffusion algorithms at steady-state in Example 2. The result for the ATC versions is omitted because almost the same results were observed.

In Example 2, we show results in a topology where the number of links is less than that of Example 1 (see Fig. 8). There are 16 links in Example 2, while 35 links in Example 1. Since there exists no cyclic path in this topology, the incremental LMS algorithm is not implementable. However, assuming that there exists a cyclic path of the same ordering as Example 1, we show its performance as a benchmark. The other parameters are the same as Example 1.

Figs. 9 and 10 show the MSD performance and the combination weights for nodes 3 and 11, respectively. We see that the performance improvement of the diffusion algorithms except for the proposed algorithm is not as significant as those in Example 1. This is because the number of links is less than that of Example 1, which results in the limitation in cooperation between the nodes. However, our algorithm still achieves good performance in this example thanks to the successful control of the combination weights (see Fig. 10). Therefore, the proposed algorithm is robust against the network topology as well as against the spatial variation of node profile.

B. Performance Analysis: Theory Versus Experiments

Let us now examine the theoretical performance for the CTA diffusion LMS algorithm with the proposed adaptive combiners.

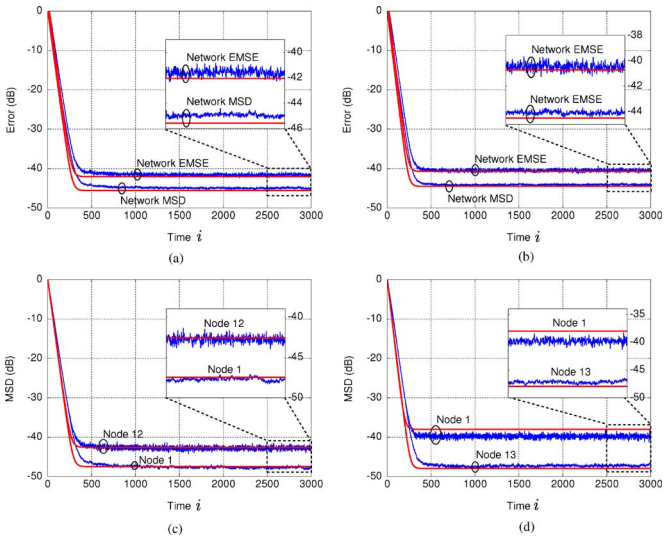


Fig. 11. Learning curves of the CTA diffusion LMS algorithm with the proposed adaptive combiners: theory versus experiments. (a) and (b): Network MSD and EMSE. (c) and (d): Local MSD. The theoretical curves were generated by Theorem 5.5. (a) Example 1; (b) Example 2; (c) Example 1; and (d) Example 2.

We consider the same networks and statistics as Examples 1 and 2 (see Figs. 5 and 8). The unknown vector and the stepsize of LMS algorithm at each node are again set to $w^o = \mathbb{1}_5/\sqrt{5}$ and $\mu_k = 0.01$. For the adaptive combiner, $b_{k,-1}$ is set to $\mathbb{1}_{n_k}/n_k$ for every k and the normalized stepsize rule (18) with $\alpha = 0.8$ and $\varepsilon = 10^{-3}$ is used. All results are obtained by averaging 200 independent experiments.

Fig. 11 compares the learning curves for Examples 1 and 2 obtained by Theorem 5.5 and experiments in terms of the network MSD, network EMSE, and local MSD. We can see a close agreement between Theorem 5.5 and experiments, although the theoretical performance curves are slightly lower than the experimental performance curves. Also, we observe a large difference for node 1 in Example 2.

On the other hand, Fig. 12 compares the steady-state performances of experiments with Theorems 5.5 and 5.6. The steady-state performance in Theorem 5.5 was calculated as a limit of its learning curve, while Theorem 5.5 uses the closed form expressions. We observe that our theory at least captures a trend of the steady-state performance, although a large difference is observed at those nodes that achieve low MSD or EMSE. However, this is because the decibel scale emphasizes errors in small values. In fact, the numerical errors are not so different across the nodes. Therefore, after averaging the performances, we get a close agreement between our theory and experiments in terms of the network performance. Also, compared to results for static combiners [10], we observe larger errors because of the additional assumptions used in our analysis for adaptive combiners.

Finally, we show the comparison of combination weights. Fig. 13 shows the combination weights of nodes 3, 6, 10, and 11 at the steady-state for Examples 1 and 2. We can see that both of Theorems 5.5 and 5.6 agree closely with the results of experiments.

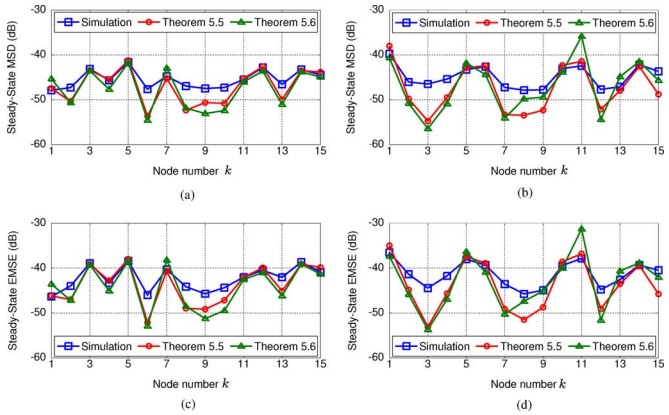


Fig. 12. Steady-state performance of the CTA diffusion LMS algorithm with the proposed adaptive combiners: theory versus experiments. The theoretical values were obtained by Theorems 5.5 and 5.6. Note that difference in small values is emphasized because of the decibel (logarithmic) scale. (a) Example 1; (b) Example 2; (c) Example 1; and (d) Example 2.

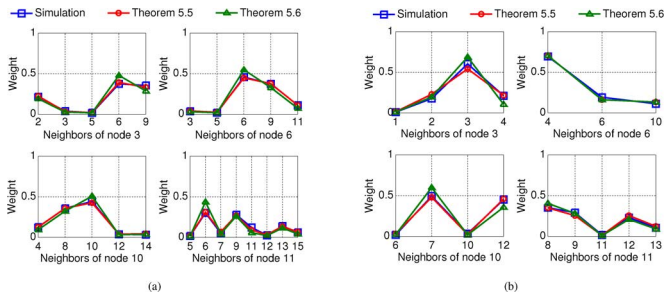


Fig. 13. Combination weights at steady-state: theory versus experiments. (a) Example 1 and (b) Example 2.

VII. Conclusion

We proposed an efficient adaptive combination rule for distributed estimation over diffusion networks in a systematic way by formulating the problem of controlling combination weights as a minimum variance unbiased estimation problem. We verified in numerical simulations that the diffusion least-mean squares (LMS) algorithm with the proposed adaptive combiners possesses robustness against the spatial variation of statistical profile across the network. Although we focused on the LMS algorithm as the adaptive filter module, combinations with other adaptive filters are possible.

The mean-transient and mean-square performance analyses of the diffusion LMS algorithm with the proposed combiner were also done. In particular, we showed that any convex combination rule is stable in the mean even if the weights are random and time-varying, which is remarkably useful for the design of combination strategies.

Appendix
Metric Projection Onto A Hyperplane

The metric projection onto a hyperplane plays an important role in the derivation of the proposed algorithm; see Section III. Here, let us derive the projection in a general form. Let $\mathcal{H}$ be

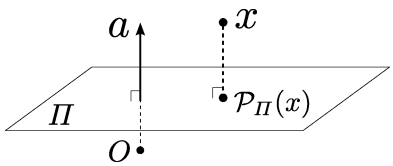


Fig. 14. Hyperplane Π and the metric projection of $x \in \mathcal{H}$ onto Π.

a *real Hilbert space*[1] with inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ and its induced norm $\| \cdot \|_{\mathcal{H}}$. Given a nonzero normal vector $a \in \mathcal{H}$ and a scalar $c \in \mathbb{R}$, a set

$$\Pi \triangleq \{x \in \mathcal{H} \,|\, \langle x, a \rangle = c\}$$

is called a hyperplane (see Fig. 14). It is known that for any $x \in \mathcal{H}$ there exists a unique point $\mathcal{P}_\Pi(x) \in \Pi$ that is closest to x among all points in Π, i.e.,

$$\mathcal{P}_\Pi(x) \triangleq \arg\min_{y \in \Pi} \|y - x\|_{\mathcal{H}}.$$

This $\mathcal{P}_\Pi(x)$ is called the *metric projection of* x *onto* Π [25], [26]. Fortunately, $\mathcal{P}_\Pi$ has a closed form expression. From Fig. 14, we can see that $\mathcal{P}_\Pi(x)$ is of the form

$$\mathcal{P}_\Pi(x) = x - ta \quad \text{for some } t \in \mathbb{R}.$$

Using the condition $\mathcal{P}_\Pi(x) \in \Pi$, i.e., $\langle x - ta, a \rangle_{\mathcal{H}} = c$, we have

$$t = \frac{\langle x, a \rangle_{\mathcal{H}} - c}{\|a\|_{\mathcal{H}}^2}$$

which gives the closed form expression of $\mathcal{P}_\Pi$:

$$\mathcal{P}_\Pi(x) = x - \frac{\langle x, a \rangle_{\mathcal{H}} - c}{\|a\|_{\mathcal{H}}^2} a \quad \text{for all } x \in \mathcal{H}. \tag{40}$$

Equation (10) immediately follows from this formula by setting $\mathcal{H}$ to the n_k-dimensional Euclidean space $\mathbb{R}^{n_k}$ (with inner product $\langle x, y \rangle_{\mathcal{H}} = y^T x$) and $\Pi = V_k$.

The projection $\mathcal{P}_\Pi$ is also available to derive the optimal solution a_k^o in (9). Define the inner product on $\mathbb{R}^{n_k}$ by $\langle x, y \rangle_{\mathrm{Re}(Q_{\Psi,k})} \triangleq x^T \mathrm{Re}(Q_{\Psi,k})\, y$ and induce the weighted norm $\|x\|_{\mathrm{Re}(Q_{\Psi,k})} \triangleq [\langle x, x \rangle_{\mathrm{Re}(Q_{\Psi,k})}]^{(1/2)}$. Then, problem (8) is equivalent to the following problem:

$$\begin{aligned} \underset{a_k \in \mathbb{R}^{n_k}}{\text{minimize}} \quad & \|a_k - 0\|^2_{\mathrm{Re}(Q_{\Psi,k})} \\ \text{subject to} \quad & a_k \in V_k. \end{aligned}$$

Hence, problem (8) is equivalent to finding a point in V_k that is closest to the origin in the sense of the norm $\| \cdot \|_{\mathrm{Re}(Q_{\Psi,k})}$. Namely, the solution is obviously the projection of the origin onto V_k with respect to the norm $\| \cdot \|_{\mathrm{Re}(Q_{\Psi,k})}$. Indeed, noting that V_k can be represented in an equivalent form

$$V_k = \{x \in \mathbb{R}^{n_k} \,|\, \langle \mathrm{Re}(Q_{\Psi,k})^{-1} \mathbb{1}_{n_k}, x \rangle_{\mathrm{Re}(Q_{\Psi,k})} = 1\},$$

[1]A complete inner product space over the real field is called a real Hilbert space [25].

the solution a_k^o is obtained from (40) by letting $x = 0$, $a = \mathrm{Re}(Q_{\Psi,k})^{-1} \mathbb{1}_{n_k}$, $c = 1$, $\langle \cdot, \cdot \rangle_{\mathcal{H}} = \langle \cdot, \cdot \rangle_{\mathrm{Re}(Q_{\Psi,k})}$ and $\| \cdot \|_{\mathcal{H}} = \| \cdot \|_{\mathrm{Re}(Q_{\Psi,k})}$.

REFERENCES

[1] N. Takahashi, I. Yamada, and A. H. Sayed, "Diffusion least-mean squares with adaptive combiners," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, Taipei, Taiwan, Apr. 2009, pp. 2845–2848.

[2] D. Estrin, G. Pottie, and M. Srivastava, "Instrumenting the world with wireless sensor networks," in *Proc. IEEE Int. Conf. Acoustics, Speech, ignal Processing (ICASSP)*, Salt Lake City, UT, May 2001, pp. 2033–2036.

[3] D. Li, K. D. Wong, Y. H. Hu, and A. M. Sayeed, "Detection, classification, and tracking of targets," *IEEE Signal Process. Mag.*, vol. 19, no. 2, pp. 17–29, Mar. 2002.

[4] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Commun. Mag.*, vol. 40, no. 8, pp. 102–114, Aug. 2002.

[5] L. A. Rossi, B. Krishnamachari, and C.-C. J. Kuo, "Distributed parameter estimation for monitoring diffusion phenomena using physical models," in *Proc. IEEE Conf. Sensor Ad Hoc Comm. Networks*, Santa Clara, CA, Oct. 2004, pp. 460–469.

[6] D. Culler, D. Estrin, and M. Srivastava, "Overview of sensor networks," *Computer*, vol. 37, no. 8, pp. 41–49, Aug. 2004.

[7] F. S. Cattivelli and A. H. Sayed, "Self-organization in bird flight formations using diffusion adaptation," presented at the Proc. 3rd Int. Workshop Computational Advances Multi-Sensor Adaptive Processing (CAMSAP), Aruba, Dec. 2009, Dutch Antilles.

[8] A. H. Sayed and C. G. Lopes, "Adaptive processing over distributed networks," *IEICE Trans. Fundam. Electron., Commun. Comput. Sci.*, vol. 90, no. 8, pp. 1504–1510, Aug. 2007.

[9] C. G. Lopes and A. H. Sayed, "Incremental adaptive strategies over distributed networks," *IEEE Trans. Signal Process.*, vol. 55, no. 8, pp. 4064–4077, Aug. 2007.

[10] 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.

[11] C. G. Lopes and A. H. Sayed, "Diffusion adaptive networks with changing topologies," in *Proc. IEEE ICASSP*, Las Vegas, NV, Apr. 2008, pp. 3285–3288.

[12] F. S. Cattivelli, C. G. 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.

[13] F. S. Cattivelli and A. H. Sayed, "Diffusion LMS algorithms with information exchange," in *Proc. 42nd Asilomar Conf. Signals, Systems, Computers*, Pacific Grove, CA, Oct. 2008, pp. 251–255.

[14] 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.

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

[16] J. Arenas-García, A. R. Figueiras-Vidal, and A. H. Sayed, "Mean-square performance of a convex combination of two adaptive filters," *IEEE Trans. Signal Process.*, vol. 54, no. 3, pp. 1078–1090, Mar. 2006.

[17] V. D. Blondel, J. M. Hendrickx, A. Olshevsky, and J. N. Tsitsiklis, "Convergence in multiagent coordination, consensus, and flocking," in *Proc. Joint 44th IEEE Conf. Decision Control Eur. Control Conf. (CDC-ECC)*, Seville, Spain, Dec. 2005, pp. 2996–3000.

[18] K. Kocay and D. L. Kreher, *Graphs, Algorithms, and Optimization*. Boca Raton, FL: Chapman & Hall/CRC Press, 2005.

[19] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *Proc. Int. Symp. Information Processing Sensor Networks (IPSN)*, Los Angeles, CA, Apr. 2005, pp. 63–70.

[20] D. S. Scherber and H. C. Papadopoulos, "Locally constructed algorithms for distributed computations in ad hoc networks," in *Proc. Int. Symp. Information Processing Sensor Networks (IPSN)*, Berkeley, CA, Apr. 2004, pp. 11–19.

[21] W. K. Hastings, "Monte Carlo sampling methods using Markov chains and their applications," *Biometrika*, vol. 57, pp. 97–109, 1970.

[22] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *J. Chem. Phys.*, vol. 21, no. 6, pp. 1087–1092, Jun. 1953.
[23] A. H. Sayed, *Adaptive Filters*. New York: Wiley, 2008.
[24] I. Yamada and N. Ogura, "Adaptive projected subgradient method for asymptotic minimization of sequence of nonnegative convex functions," *Numer. Funct. Anal. Optim.*, vol. 25, no. 7, pp. 593–617, Jan. 2004.
[25] D. G. Luenberger, *Optimization by Vector Space Methods*. New York: Wiley-Interscience, 1969.
[26] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, 1st ed. Singapore: Athena Scientific, Jan. 1997.
[27] N. Takahashi and I. Yamada, "Parallel algorithms for variational inequalities over the cartesian product of the intersections of the fixed point sets of nonexpansive mappings," *J. Approx. Theory*, vol. 153, no. 2, pp. 139–160, Aug. 2008.
[28] A. H. Sayed, *Fundamentals of Adaptive Filtering*. New York: Wiley-IEEE Press, 2003.
[29] D. S. Tracy and R. P. Singh, "A new matrix product and its applications in partitioned matrix differentiation," *Statistica Neerlandica*, vol. 26, no. 4, pp. 143–157, 1972.
[30] R. H. Koning, H. Neudecker, and T. Wansbeek, "Block Kronecker products and the Vecb operator," Economics Dept., Institute of Economics Research, Univ. of Groningen, Groningen, The Netherlands, Tech. Rep. 351, 1990.
[31] T. Y. Al-Naffouri and A. H. Sayed, "Transient analysis of adaptive filters with error nonlinearities," *IEEE Trans. Signal Process.*, vol. 51, no. 3, pp. 653–663, Mar. 2003.

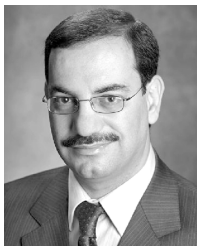
Noriyuki Takahashi (M'09) received the B.E. degree in computer science and the M.E. and Ph.D. degrees in electrical and electronic engineering from the Tokyo Institute of Technology, Tokyo, Japan, in 2005, 2006, and 2009, respectively.

From April 2007 to March 2009, he was a research fellow of the Japan Society for the Promotion of Science (JSPS). He is currently an Assistant Professor in the Global Edge Institute, Tokyo Institute of Technology. His current research interests are in distributed algorithms, mathematical signal processing, and convex analysis.

Dr. Takahashi received the Teshima Memorial Award in 2010 from the Tokyo Institute of Technology.

Isao Yamada (M'96–SM'06) is a Professor in the Department of Communications and Integrated Systems, Tokyo Institute of Technology. His current research interests are in mathematical signal processing, optimization theory, nonlinear inverse problem.

Dr. Yamada has been an Associate Editor for several journals, including the *International Journal on Multidimensional Systems and Signal Processing* (since 1997), the *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* (2001–2005), and the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—PART I: FUNDAMENTAL THEORY AND APPLICATIONS (2006–2007). Currently, he is a member of the IEEE Signal Processing Theory and Methods Technical Committee, as well as an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING. He received Excellent Paper Awards in 1991, 1995, 2006, and 2009 and the Young Researcher Award in 1992 and the Achievement Award in 2009 from the IEICE; the ICF Research Award from the International Communications Foundation in 2004; the DoCoMo Mobile Science Award (Fundamental Science Division) from Mobile Communication Fund in 2005; and the Fujino Prize from Tejima Foundation in 2008.



Ali H. Sayed (F'01) is Professor and Chairman of Electrical Engineering at the University of California, Los Angeles (UCLA), where he directs the Adaptive Systems Laboratory (www.ee.ucla.edu/asl). He has published widely in the areas of adaptive systems, statistical signal processing, estimation theory, adaptive and cognitive networks, and signal processing for communications. He has authored or coauthored several books, including *Adaptive Filters* (Wiley, 2008), *Fundamentals of Adaptive Filtering* (Wiley, 2003), and *Linear Estimation* (Prentice-Hall, 2000).

Dr. Sayed has served on the editorial boards of several journals, including as Editor-in-Chief of the IEEE TRANSACTIONS ON SIGNAL PROCESSING. His work has received several awards, including the 1996 IEEE Donald G. Fink Award, the 2003 Kuwait Prize, the 2005 Terman Award, and the 2002 Best Paper Award and 2005 Young Author Best Paper Award from the IEEE Signal Processing Society. He has served as a 2005 Distinguished Lecturer of the same society and as General Chairman of ICASSP 2008. He is serving as the Vice-President of Publications of the IEEE Signal Processing Society.