IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received March 1, 2019, accepted March 27, 2019, date of publication April 9, 2019, date of current version April 18, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2909775

Diffusion LMS Based on Message Passing Algorithm

AYANO NAKAI-KASAI[1], (Student Member, IEEE), AND KAZUNORI HAYASHI[2], (Member, IEEE)

[1]Graduate School of Informatics, Kyoto University, Kyoto 606-8501, Japan
[2]Graduate School of Engineering, Osaka City University, Osaka 558-8585, Japan

Corresponding author: Ayano Nakai-Kasai (nakai.ayano@sys.i.kyoto-u.ac.jp)

This work was supported in part by the Grants-in-Aid for Scientific Research 18K04148 and 18H03765 through the Ministry of Education, Culture, Sports, Science and Technology of Japan.

ABSTRACT Diffusion least-mean-square (LMS) is an adaptive algorithm that estimates an unknown global vector from its linear measurements obtained at all nodes in a distributed manner when each node in the network needs to track the unknown vector in real-time. The algorithm uses the conventional average consensus protocol in order to combine neighbors' estimates at each node, while another protocol, consensus propagation (CP), is known to achieve faster and exact average consensus when the network has a tree structure. This paper proposes a novel diffusion LMS algorithm using CP, which can be applied for any network by extracting a spanning tree from the original network and can achieve the same solution as the centralized LMS in a fully distributed manner. This paper also proposes an algorithm by using the idea of loopy CP, so that it can be directly applied even when the network is not a tree and shows that its special case results in the diffusion LMS using a novel combination rule. Moreover, we optimize the constants involved in the proposed combination rule in terms of the steady-state mean-square-deviation of the diffusion LMS and show an adaptive implementation of the proposed algorithm. The simulation results demonstrate that the proposed algorithm using CP is beneficial for large-scale networks, and the diffusion LMS with the proposed combination rule achieves better convergence performance than that with the conventional combination rules when the measurement noise power depends on nodes.

INDEX TERMS Average consensus, consensus propagation, diffusion LMS, in-network signal processing.

I. INTRODUCTION

Large-scale communication networks composed of a number of small nodes having abilities of wireless communication, computation, and sensing, such as sensor networks and M2M (machine-to-machine), are gathering much attention in various fields recently. There are two major approaches for collecting and processing measurements in such networks, namely, a centralized fusion-based solution and a distributed in-network signal processing solution. In the centralized approach, local measurements from all nodes are collected at a special node called fusion node or fusion center, and processed at the node in a centralized manner. This approach usually achieves better performance because it can use all measurements in the network at hand. It suffers however from problems that the nodes around the fusion node tend to consume much energy for the relaying and also that the network is vulnerable to the failure of the fusion node. In order to collect and process measurements efficiently and reliably, distributed in-network signal processing has been proposed [1]. In this approach, each node updates the estimate of an unknown parameter using not only its local measurements but also its neighbors' estimates instead of gathering all measurements at a special node.

For the distributed in-network estimation problem, least-mean-square (LMS) based strategies [2]–[5] are effective when each node in the network needs to track an unknown global vector in real-time. In these algorithms, each node iteratively updates its estimate by the LMS algorithm using local measurements and by averaging the estimates of its neighbors obtained via wireless communications, and finally all nodes in the network achieve a common estimate. The incremental LMS [2] is applicable for any connected networks but it suffers from slow convergence because it assumes that each node

The associate editor coordinating the review of this manuscript and approving it for publication was Suzhi Bi.

communicates only with a specific node among its neighbors. In order to achieve faster convergence, the Combine-then-Adapt (CTA) diffusion LMS [3] is derived by modifying the updating rules of [2] so that it allows each node to utilize the estimates at all neighbors. Moreover, the distributed LMS in [4] achieves faster convergence than the CTA diffusion LMS [3] while it requires some special nodes having different capabilities from other nodes, and a special network structure to use the node hierarchy. Furthermore, a modified diffusion LMS named Adapt-then-Combine (ATC) diffusion LMS, which is obtained by just exchanging the order of the updating rules of CTA diffusion [3], has been proposed in [5]. The ATC diffusion LMS can outperform not only the original version [3] but also the distributed LMS [4] in terms of convergence rate. Moreover, the convergence performance of this algorithm approximately agrees with the centralized solution if noise statistics of all nodes are known and sufficiently small step-size parameters are employed [6]–[8]. More recently, some improvements have been proposed such as a doubly-compressed diffusion LMS [9] for communication reduction and an Adapt-Multi-Combine (AMC) diffusion LMS [10] for better convergence performance at the cost of frequent communications. The diffusion LMS for more specific settings has been attracted recently in the context of the multitask learning where each node in the network estimates its own target vector and the neighbor nodes have related targets [11].

The diffusion LMS algorithms employ the conventional average consensus protocol [12] to combine neighbors' estimates at each node, which requires a large number of iterations especially in large-scale networks. It is also known that, even in small-scale networks, the choice of combination weights used in the averaging step has a great impact on the convergence performance of the diffusion LMS. Thus, several combination rules have been proposed in the literature, such as uniform rule [13], maximum degree rule [14], Metropolis rule [15], and relative degree rule [5]. More sophisticated static rules are considered in [5], [16]–[18], which are derived by solving some optimization problems. In particular, a closed-form solution that minimizes the steady-state error has been derived in [16] and [17], which is referred to as a relative-variance rule. Since the relative-variance rule requires network statistics such as noise variance at all nodes, which are not locally available at each node in general, adaptive estimation methods of the parameters have been also proposed in [16] and [17].

In this paper, we propose a novel distributed LMS algorithm [19] by applying consensus propagation (CP) [20], which is an average consensus algorithm based on belief propagation [21]–[24], in order to further improve the convergence rate of the diffusion LMS in [5]. For networks with the tree structure, CP is known to achieve exact average consensus with the minimum number of iterations required for message propagation through the network, i.e., the diameter of the tree. Since networks for in-network signal processing do not necessarily have the tree structure in general, the proposed diffusion LMS algorithm is applied for the spanning tree extracted from the original network using some spanning tree protocols [25], [26]. This paper also proposes a novel combination rule [27] of the diffusion LMS, which is directly applicable to the original networks having cycles. The proposed combination rule uses the idea of loopy CP, which has been also proposed in [20] and can achieve average consensus approximately on the network with some cycles. Loopy CP involves some constants that control the convergence property but they are known to be difficult to optimize in general. We thus select the constants by minimizing steady-state mean-squared-deviation (MSD) of the diffusion LMS as in [16], [17]. We further extend the proposed combination rule to an adaptive version, which can be implemented in a fully distributed manner. Simulation results show that the proposed diffusion LMS using CP can achieve faster convergence than the conventional ATC diffusion LMS for large-scale networks and that the proposed combination rule inspired by loopy CP achieves better convergence performance than the conventional combination rules when the measurement noise power depends on the nodes.

A. NOTATIONS

In the rest of the paper, we use the following notations. Let $\mathbb{C}$ be the set of complex numbers. Superscripts $(\cdot)^{\mathrm{T}}$ and $(\cdot)^{\mathrm{H}}$ denote the transpose and the Hermitian transpose, respectively. $\boldsymbol{I}_M$ is the identity matrix with size $M \times M$. $\mathbf{0}_M$ and $\mathbf{1}_M$ represent the $M \times 1$ vectors whose elements are all 0 and 1, respectively. $\mathrm{E}[\cdot]$ and $\mathrm{Tr}(\cdot)$ stand for the expectation and the trace operators. $\mathrm{vec}(\cdot)$ and $\mathrm{vec}^{-1}(\cdot)$ denote the vectorization and the inverse vectorization operators. $\otimes$, $(\cdot)_p$, and $\lambda_{\max}(\cdot)$ denote Kronecker product, the p-th element of the vector, and the maximum eigenvalue of the matrix, respectively. $\mathrm{diag}\{\cdots\}$ means the diagonal or block diagonal matrix, where its diagonal elements or matrices are composed of the elements or matrices in the braces.

II. PRELIMINARIES

A. DIFFUSION LMS ALGORITHM

Consider a network with N nodes. Each node k in the network can perform single-hop communications with its neighbors and obtains linear measurements of an unknown deterministic vector of interest $\boldsymbol{w}^{\mathrm{o}} \in \mathbb{C}^{M\times 1}$ as [7], [8]:

$$d_k^{(i)} = \boldsymbol{u}_k^{(i)\mathrm{H}} \boldsymbol{w}^{\mathrm{o}} + v_k^{(i)}. \tag{1}$$

Here, $i\,(\geq 0)$ is a time index, $d_k^{(i)} \in \mathbb{C}$ is a scalar measurement at node k at time i, $\boldsymbol{u}_k^{(i)} \in \mathbb{C}^{M\times 1}$ is a random measurement vector with a correlation matrix of $\boldsymbol{R}_{u_k} = \mathrm{E}[\boldsymbol{u}_k^{(i)}\boldsymbol{u}_k^{(i)\mathrm{H}}]$, and $v_k^{(i)} \in \mathbb{C}$ is a zero-mean additive complex white Gaussian noise with variance of σ_k^2. Fig. 1 shows an example of the network with $N = 10$. The stochastic processes $\{d_k^{(i)}, \boldsymbol{u}_k^{(i)}\}$ are assumed to be jointly wide-sense stationary and zero-mean. For simplicity, we assume that all communications between neighbor nodes are perfect, i.e., we do not consider any communication error, while the proposed methods in this paper are applicable to the networks with noisy links [16].

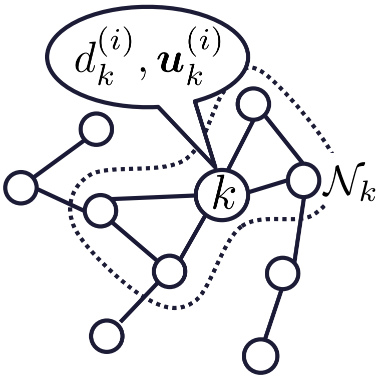

FIGURE 1. An example of a network with 10 nodes. The nodes that are directly connected by edges can communicate and share information with each other. $\mathcal{N}_k$ is the set of neighbors of node k (including node k itself). Each node k obtains the measurement $d_k^{(i)}$ by using the measurement vector $u_k^{(i)}$ at time i.

All nodes in the network estimate $\boldsymbol{w}^{\mathrm{o}}$ by solving the following optimization problem [7], [8]:

$$\hat{\boldsymbol{w}}^{\mathrm{o}} = \arg\min_{\boldsymbol{w}} \sum_{k=1}^{N} \mathrm{E}[|d_k^{(i)} - \boldsymbol{u}_k^{(i)\mathrm{H}}\boldsymbol{w}|^2]. \tag{2}$$

The diffusion LMS algorithm [3]–[5] is the iterative method to solve this global problem in a distributed manner. Specifically, the update rule of the diffusion LMS uses locally available information at each node only, such as its own measurements and its neighbors' estimates, while the direct calculation of (2) requires all nodes' measurements at hand. The ATC version of the diffusion LMS update can be described as [5]

$$\boldsymbol{\psi}_k^{(i)} = \boldsymbol{\phi}_k^{(i-1)} + \mu_k \boldsymbol{u}_k^{(i)}(d_k^{(i)} - \boldsymbol{u}_k^{(i)\mathrm{H}}\boldsymbol{\phi}_k^{(i-1)}), \tag{3}$$
$$\boldsymbol{\phi}_k^{(i)} = \sum_{l\in\mathcal{N}_k} a_{lk}\boldsymbol{\psi}_l^{(i)}, \tag{4}$$

where $\boldsymbol{\psi}_k^{(i)}$ is an immediate estimate obtained by LMS update at node k and time i, $\boldsymbol{\phi}_k^{(i)}$ is a subsequent estimate obtained by the weighted average of its neighbors' immediate estimates with $\boldsymbol{\phi}_k^{(-1)} = 0$, μ_k is the step-size parameter, a_{lk} is a nonnegative combination weight, which is the (l, k) element of an $N \times N$ matrix $\boldsymbol{A}$ that satisfies $\boldsymbol{1}^{\mathrm{T}}\boldsymbol{A} = \boldsymbol{1}^{\mathrm{T}}$, and $\mathcal{N}_k$ is the set of neighbors of node k including k itself (see Fig. 1). Note that an alternative algorithm obtained by replacing the order of the updates (3) and (4) is referred to as the CTA version of the diffusion LMS. In this paper, we focus on ATC because it outperforms CTA under realistic conditions [6], though the following discussion can hold in both versions. Note also that our methods are applicable to more general LMS update in [5] that uses the measurements and the measurement vectors at neighbors without loss of generality.

Equation (4) is related to the conventional consensus protocol using a weighted average of neighbors' values in multi-agent networked systems [12]. In the protocol, all nodes in the network update their state values by exchanging the current values with their neighbors and finally obtain the average of all initial state values, which is called average consensus. An iterative update equation of the conventional discrete time consensus protocol is given by

$$x_k^{(i'+1)} = x_k^{(i')} + \epsilon \sum_{l\in\mathcal{N}_k\setminus k} a'_{lk}(x_l^{(i')} - x_k^{(i')}), \tag{5}$$

where $x_k^{(i')}$ is a state value at node k and time i', ϵ is a positive small number, a'_{lk} is the (l, k) element of the weighted adjacency matrix of the network, and $\mathcal{N}_k \setminus k$ is the set $\mathcal{N}_k$ but without node k. In order to relate the update equation (4) with (5), we substitute $\boldsymbol{\phi}_k^{(i)}$ to $x_k^{(i'+1)}$ and $\boldsymbol{\psi}_k^{(i)}$ to $x_k^{(i')}$. Then, we have

$$\begin{aligned}\boldsymbol{\phi}_k^{(i)} &= \boldsymbol{\psi}_k^{(i)} + \epsilon \sum_{l\in\mathcal{N}_k\setminus k} a'_{lk}(\boldsymbol{\psi}_l^{(i)} - \boldsymbol{\psi}_k^{(i)}) \\ &= (1 - \sum_{l\in\mathcal{N}_k\setminus k} a_{lk})\boldsymbol{\psi}_k^{(i)} + \sum_{l\in\mathcal{N}_k\setminus k} a_{lk}\boldsymbol{\psi}_l^{(i)} \\ &= \sum_{l\in\mathcal{N}_k} a_{lk}\boldsymbol{\psi}_l^{(i)},\end{aligned}$$

where we set $a_{lk} = \epsilon a'_{lk}$. Therefore, the update rule of (4) can be regarded as the conventional average consensus protocol.

Note that the choice of the combination rule has a great impact on the convergence performance of the diffusion LMS [18]. Possible choices of the combination weights a_{lk} in (4) will be uniform rule [13], Metropolis rule [15]

$$a^{\mathrm{met}}{}_{lk} = \begin{cases} \dfrac{1}{\max\{|\mathcal{N}_k|, |\mathcal{N}_l|\}} & \text{if } l \in \mathcal{N}_k \\ 0 & \text{otherwise,} \end{cases} \tag{6}$$

relative degree rule [5], and so on.

The estimation error at node k and time i is given by

$$\tilde{\boldsymbol{w}}_k^{(i)} = \boldsymbol{w}^{\mathrm{o}} - \boldsymbol{\phi}_k^{(i)}, \tag{7}$$

and the steady-state MSD at each node k and the steady-state network MSD are expressed as

$$\mathrm{MSD}_k = \lim_{i\to\infty} \mathrm{E}[\|\tilde{\boldsymbol{w}}_k^{(i)}\|^2], \tag{8}$$
$$\mathrm{MSD}^{\mathrm{nw}} = \frac{1}{N}\sum_{k=1}^{N}\mathrm{MSD}_k, \tag{9}$$

respectively.

AMC diffusion LMS algorithm [10] has been proposed recently and is constructed based on the diffusion LMS. The update rules are described as

$$\boldsymbol{\psi}_k^{(i)} = \boldsymbol{\phi}_k^{(i-1)} + \mu_k \boldsymbol{u}_k^{(i)}(d_k^{(i)} - \boldsymbol{u}_k^{(i)\mathrm{H}}\boldsymbol{\phi}_k^{(i-1)}), \tag{10}$$
$$\boldsymbol{\phi}_k^{(i)[1]} = \sum_{l\in\mathcal{N}_k} a_{lk}^{[1]}\boldsymbol{\psi}_l^{(i)}, \tag{11}$$
$$\boldsymbol{\phi}_k^{(i)[2]} = \sum_{l\in\mathcal{N}_k} a_{lk}^{[2]}\boldsymbol{\phi}_l^{(i)[1]}, \tag{12}$$
$$\vdots$$
$$\boldsymbol{\phi}_k^{(i)} = \sum_{l\in\mathcal{N}_k} a_{lk}^{[J']}\boldsymbol{\phi}_l^{(i)[J'-1]}, \tag{13}$$

where J' denotes the number of iterations of the weighted average, $\boldsymbol{\psi}_k^{(i)[\cdot]}$ is the immediate estimate, and $a_{lk}^{[\cdot]}$ is a combination weight. The LMS update of AMC (10) is the same as that of the diffusion LMS (3), but the AMC diffusion LMS communicates J' times for (11)–(13) to collect immediate estimates of the nodes within J'-hop.

B. CONSENSUS PROPAGATION

CP [20] is the algorithm to achieve the average consensus in the network with a tree structure by using the idea of message passing algorithm [21]–[24]. Assume that each node k has an initial state value $x_k \in \mathbb{C}$ in a network composed of N nodes with a tree structure, the goal of CP is that each node obtains the average of the initial state values $\bar{x} = \frac{1}{N}\sum_{k=1}^{N} x_k$.

CP consists of two types of updates, namely, message update between neighbor nodes and state update at each node, to calculate the average using locally available information only. The updates of CP at the j-th iteration are given as follows:

$$K_{(k\to l)}^{[j]} = 1 + \sum_{u\in\mathcal{N}_k\backslash l,k} K_{(u\to k)}^{[j-1]}, \tag{14}$$

$$\theta_{(k\to l)}^{[j]} = \frac{x_k + \sum_{u\in\mathcal{N}_k\backslash l,k} K_{(u\to k)}^{[j-1]}\theta_{(u\to k)}^{[j-1]}}{1+\sum_{u\in\mathcal{N}_k\backslash l,k} K_{(u\to k)}^{[j-1]}}, \tag{15}$$

$$x_k^{[j]} = \frac{x_k + \sum_{u\in\mathcal{N}_k\backslash k} K_{(u\to k)}^{[j]}\theta_{(u\to k)}^{[j]}}{1+\sum_{u\in\mathcal{N}_k\backslash k} K_{(u\to k)}^{[j]}}, \tag{16}$$

where $K_{(k\to l)}^{[0]} = 0$, $K_{(k\to l)}^{[j]}$ and $\theta_{(k\to l)}^{[j]}$ are the messages sent from node k to l, and $x_k^{[j]}$ is the state value at node k. Figs. 2 and 3 show the updates at node k and its neighbors. By iterating (14) and (15) between all neighbor nodes with the same number as the diameter of the tree, $x_k^{[j]}$ converges to $\bar{x}$ meaning that all nodes obtain $\bar{x}$. Since the diameter is the minimum number of iterations required for the message propagation through the entire network, CP is the fast and efficient algorithm to achieve average consensus.

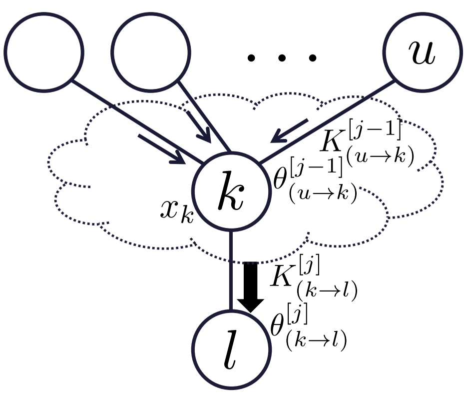

FIGURE 2. Update of message from node k to node l in the j-th iteration. The message is generated from the messages received in the $(j-1)$-th iteration from its neighbors except for node l, and its initial value x_k.

Although the algorithm (14)–(16) usually diverges when the network involves cycles, CP can achieve average

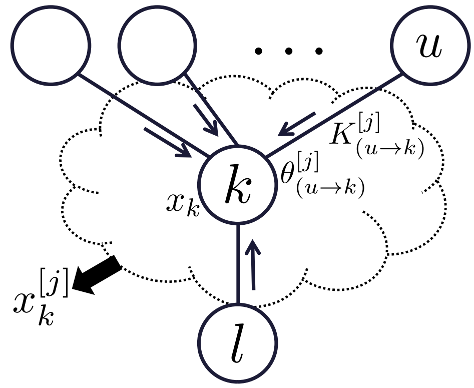

FIGURE 3. State update at node k in the j-th iteration. Node k calculates the current value $x_k^{[j]}$ using the messages received in the j-th iteration from its all neighbors.

consensus approximately even in such networks, if the update of the message $K_{(u\to k)}^{[j]}$ is replaced with

$$K_{(u\to k)}^{[j]} = \frac{1+\sum_{m\in\mathcal{N}_u\backslash k,u} K_{(m\to u)}^{[j-1]}}{1+\frac{1}{\beta_k}\left(1+\sum_{m\in\mathcal{N}_u\backslash k,u} K_{(m\to u)}^{[j-1]}\right)}, \tag{17}$$

where β_k is a positive constant. It is notable that β_k in (17) can be also set as $\beta_{k,u}$ so that it depends not only on k but also on u. However, the convergence behavior has not been fully understood yet because of the complicated message propagation due to cycles. It is known that β_k in (17) plays an important role to ensure the convergence, but, to the best of our knowledge, the optimal value of β_k has not been derived. We call the algorithm using (17), (15), and (16) for the updates as loopy CP.

III. DIFFUSION LMS USING CONSENSUS PROPAGATION

A. PROPOSED ALGORITHM

In the first approach, we firstly extract a spanning tree from the original network which possibly has some cycles using some centralized or distributed spanning tree protocol such as [25] and [26]. For example, we can use the algorithm in [26] to find a minimum diameter spanning tree of any graph $G = \{V, E\}$ with $O(|V|)$ time complexity and $O(|V||E|)$ message complexity. We then apply CP in the averaging step of the diffusion LMS on the extracted tree network. Every time each node obtains a linear measurement, the estimate $\boldsymbol{\psi}_k^{(i)}$ at node k is updated by LMS (3) as in the case with the conventional method. The subsequent estimate $\boldsymbol{\phi}_k^{(i)}$ is obtained as the consensus value achieved by CP on each element of the vector $\boldsymbol{\psi}_k^{(i)}$, i.e., the average of all nodes' estimates at time i. After each node performs the same number of CP updates as the diameter J of the spanning tree, the average is substituted to the new estimate $\boldsymbol{\phi}_k^{(i)}$, and then the algorithm proceeds to the next step.

The proposed updating rules are summarized in Algorithm 1. $K_{p,(k\to l)}^{(i)[j]}$ and $\theta_{p,(k\to l)}^{(i)[j]}$ are the p-th elements of the messages transmitted from node k to node l at the j-th iteration of CP at time i. $x_{p,k}^{(i)[J]}$ is the p-th element of the

Algorithm 1 CP-LMS

Initialization: $\boldsymbol{\phi}_k^{(-1)} = 0$
for each time $i \geq 0$, each node k, and each element p **do**
  $\boldsymbol{\psi}_k^{(i)} = \boldsymbol{\phi}_k^{(i-1)} + \mu_k \boldsymbol{u}_k^{(i)}(d_k^{(i)} - \boldsymbol{u}_k^{(i)\mathrm{H}} \boldsymbol{\phi}_k^{(i-1)})$
  Substitute $x_{p,k}^{(i)[0]} = (\boldsymbol{\psi}_k^{(i)})_p$,
  $\theta_{p,(k\to l)}^{(i)[0]} = 0, K_{p,(k\to l)}^{(i)[0]} = 0$
  for $j = 1$ to J **do**
    $K_{p,(k\to l)}^{(i)[j]} = 1 + \sum_{u\in\mathcal{N}_k \setminus l,k} K_{p,(u\to k)}^{(i)[j-1]}$
    $\theta_{p,(k\to l)}^{(i)[j]} = \frac{x_{p,k}^{(i)[0]} + \sum_{u\in\mathcal{N}_k \setminus l,k} K_{p,(u\to k)}^{(i)[j-1]} \theta_{p,(u\to k)}^{(i)[j-1]}}{1 + \sum_{u\in\mathcal{N}_k \setminus l,k} K_{p,(u\to k)}^{(i)[j-1]}}$
  end for
  $x_{p,k}^{(i)[J]} = \frac{x_{p,k}^{(i)[0]} + \sum_{u\in\mathcal{N}_k \setminus k} K_{p,(u\to k)}^{(i)[J]} \theta_{p,(u\to k)}^{(i)[J]}}{1 + \sum_{u\in\mathcal{N}_k \setminus k} K_{p,(u\to k)}^{(i)[J]}}$
  Substitute $(\boldsymbol{\phi}_k^{(i)})_p = x_{p,k}^{(i)[J]}$
end for

average of all nodes' estimates at time i. We call this novel diffusion LMS algorithm CP-LMS.

B. PERFORMANCE ANALYSIS

In this section, we analyze the mean stability, the mean-square stability, and the transient behavior of the proposed CP-LMS. In order to make the analysis tractable, we introduce the following assumptions as in [5]:

- The noise process $\{v_k^{(i)}\}$ is temporally white and spatially independent.
- The measurement vector process $\{\boldsymbol{u}_k^{(i)}\}$ is temporally white and spatially independent.
- $v_k^{(i)}$ is independent of $\boldsymbol{u}_l^{(j)}$ for all $l \neq k$ and $j \neq i$.
- The step-sizes $\{\mu_k\}$ are sufficiently small.

It should be noted here that the proposed CP-LMS realizes the centralized fusion-based solution (block LMS [6], [28]) in a fully distributed manner if we introduce further assumption:

- All nodes have the same step-size parameter, i.e., $\mu_k = \mu$ for all k,

because all nodes can collect all nodes' estimates at each iteration after J times updates of CP. To be more specific, the CP-LMS update equations can be summarized as

$$\begin{cases} \boldsymbol{\psi}_k^{(i)} = \boldsymbol{\phi}_k^{(i-1)} + \mu \boldsymbol{u}_k^{(i)}\left(d_k^{(i)} - \boldsymbol{u}_k^{(i)H} \boldsymbol{\phi}_k^{(i-1)}\right), \\ \boldsymbol{\phi}_k^{(i)} = \dfrac{1}{N} \sum_{l=1}^{N} \boldsymbol{\psi}_l^{(i)}. \end{cases}$$

Since $\boldsymbol{\phi}_k^{(i)}$ does not depend on k, we replace $\boldsymbol{\phi}_k^{(i)}$ with $\boldsymbol{w}^{(i)}$, then the whole update is given by

$$\begin{aligned} \boldsymbol{w}^{(i)} &= \frac{1}{N} \sum_{k=1}^{N} \left[\boldsymbol{w}^{(i-1)} + \mu \boldsymbol{u}_k^{(i)}\left(d_k^{(i)} - \boldsymbol{u}_k^{(i)H} \boldsymbol{w}^{(i-1)}\right)\right] \\ &= \boldsymbol{w}^{(i-1)} + \frac{\mu}{N} \sum_{k=1}^{N} \boldsymbol{u}_k^{(i)}\left(d_k^{(i)} - \boldsymbol{u}_k^{(i)H} \boldsymbol{w}^{(i-1)}\right), \end{aligned} \quad (18)$$

which corresponds to the centralized block LMS algorithm [6], [28]. This means that the conventional performance analysis for the block LMS is directly applicable to the proposed CP-LMS.

For example, the condition that the estimate error $\tilde{\boldsymbol{w}}_k^{(i)} = \tilde{\boldsymbol{w}}^{(i)}$ converges to $\boldsymbol{0}_M$ as $i \to \infty$ in the mean and mean-square sense is given by [28]

$$0 < \mu < \frac{2}{\lambda_{\max}(\boldsymbol{R})}, \quad (19)$$

where $\boldsymbol{R} = \sum_{k=1}^{N} \boldsymbol{R}_{u_k}$.

Moreover, the transient behavior of the CP-LMS can be also represented by the same expression as the conventional block LMS [28]. Now let $\boldsymbol{r}_{u_k} = \mathrm{vec}(\boldsymbol{R}_{u_k})$ and

$$\begin{aligned} \boldsymbol{F} = \boldsymbol{I}_{M^2} &- \mu'(\boldsymbol{I}_M \otimes \boldsymbol{R}) - \mu'(\boldsymbol{R}^{\mathrm{T}} \otimes \boldsymbol{I}_M) \\ &+ \mu'^2(\boldsymbol{R}^{\mathrm{T}} \otimes \boldsymbol{R}) + \mu'^2 \sum_{k=1}^{N} \boldsymbol{r}_{u_k} \boldsymbol{r}_{u_k}^{\mathrm{H}}, \end{aligned}$$

where $\mu' = \mu/N$, then the theoretical network MSD learning curve of the CP-LMS algorithm is given by

$$\begin{aligned} \eta^{(i)} = \eta^{(i-1)} &+ \mu'^2 \sum_{k=1}^{N} \sigma_k^2 \boldsymbol{r}_{u_k}^{\mathrm{H}} \boldsymbol{F}^i \boldsymbol{q} \\ &- \boldsymbol{w}^{\mathrm{oT}} \left[\mathrm{vec}^{-1}\left(\boldsymbol{F}^i \left[\boldsymbol{I}_{M^2} - \boldsymbol{F}\right] \boldsymbol{q}\right)\right] \boldsymbol{w}^{\mathrm{o}}, \end{aligned} \quad (20)$$

where $\eta^{(i)}$ is the network MSD at time i and $\boldsymbol{q} = \mathrm{vec}(\boldsymbol{I}_M)$. It is clear that $\mathrm{MSD}_k = \mathrm{MSD}^{\mathrm{nw}}$ holds in the case of CP-LMS because all nodes obtain the same estimate $\boldsymbol{w}^{(i)}$ at each step i. Thus, the steady-state MSD is given by

$$\mathrm{MSD}_k = \mathrm{MSD}^{\mathrm{nw}} = \mu'^2 \sum_{l=1}^{N} \sigma_l^2 \boldsymbol{r}_{u_l}^{\mathrm{H}} (\boldsymbol{I}_{M^2} - \boldsymbol{F})^{-1} \boldsymbol{q}. \quad (21)$$

Note that it is one of the important merits of the CP-LMS scheme that the transient behavior can be described by the same expression as that of the centralized solution. Unlike the conventional diffusion LMS, the calculation of the theoretical transient behavior of the centralized solution requires much smaller computational complexity than that of the conventional diffusion LMS.

IV. DIFFUSION LMS BASED ON LOOPY CONSENSUS PROPAGATION

The extraction of the spanning tree in the algorithm in Sect. III-A can ensure the perfect consensus at each iteration of LMS. However, the extraction may also deteriorate the convergence performance because some communication links available in the original network are ignored in the extracted spanning tree. In this section, we consider to apply loopy CP to the diffusion LMS without extracting any spanning tree. We firstly show the general algorithm, and then consider a special case, where the number of iterations of loopy CP updates is limited to one. The special case enables us to select optimal constants β_k in terms of the performance of the diffusion LMS and to analyze the convergence performance of the algorithm.

Algorithm 2 LCP-LMS

Initialization: $\boldsymbol{\phi}_k^{(-1)} = 0$
for each time $i \geq 0$, each node k, and each element p **do**
  $\boldsymbol{\psi}_k^{(i)} = \boldsymbol{\phi}_k^{(i-1)} + \mu_k \boldsymbol{u}_k^{(i)}(d_k^{(i)} - \boldsymbol{u}_k^{(i)\mathrm{H}}\boldsymbol{\phi}_k^{(i-1)})$
  Substitute $x_{p,k}^{(i)[0]} = (\boldsymbol{\psi}_k^{(i)})_p$,
  $\theta_{p,(k\to l)}^{(i)[0]} = 0,\ K_{p,(k\to l)}^{(i)[0]} = 0$
  for $t = 1$ to T **do**
    $K_{p,(k\to l)}^{(i)[t]} = \frac{1+\sum_{u\in\mathcal{N}_k\setminus l,k} K_{p,(u\to k)}^{(i)[t-1]}}{1+\frac{1}{\beta_l}(1+\sum_{u\in\mathcal{N}_k\setminus l,k} K_{p,(u\to k)}^{(i)[t-1]})}$
    $\theta_{p,(k\to l)}^{(i)[t]} = \frac{x_{p,k}^{(i)[0]}+\sum_{u\in\mathcal{N}_k\setminus l,k} K_{p,(u\to k)}^{(i)[t-1]}\theta_{p,(u\to k)}^{(i)[t-1]}}{1+\sum_{u\in\mathcal{N}_k\setminus l,k} K_{p,(u\to k)}^{(i)[t-1]}}$
  end for
  $x_{p,k}^{(i)[T]} = \frac{x_{p,k}^{(i)[0]}+\sum_{u\in\mathcal{N}_k\setminus k} K_{p,(u\to k)}^{(i)[T]}\theta_{p,(u\to k)}^{(i)[T]}}{1+\sum_{u\in\mathcal{N}_k\setminus k} K_{p,(u\to k)}^{(i)[T]}}$
  Substitute $(\boldsymbol{\phi}_k^{(i)})_p = x_{p,k}^{(i)[T]}$
end for

A. PROPOSED ALGORITHM – GENERAL CASE

Here, we describe the modified CP-LMS that can be directly applicable to networks with some cycles. Every time each node obtains a linear measurement, the estimate $\boldsymbol{\psi}_k^{(i)}$ at node k is updated by LMS (3) as in the case with the conventional method. Since it is difficult to know the required number of updates of CP for the case of loopy CP, we set a fixed number of updates. After each node performs CP updates T times, $x_{p,k}^{(i)[T]}$ is assigned to the p-th element of the new estimate $\boldsymbol{\phi}_k^{(i)}$, and then the algorithm proceeds to the next step. Note that $x_{p,k}^{(i)[T]}$ will not be exactly the same as the average of the p-th elements of $\boldsymbol{\psi}_k^{(i)}$ in general for any T.

The proposed updating rules are summarized in Algorithm 2. We call this algorithm LCP-LMS.

B. PROPOSED ALGORITHM – SPECIAL CASE

LCP-LMS in Sect. IV-A has difficulties in analyzing the performance and determining the constants β_k because the behavior of loopy CP has not been fully understood. However, we can analytically evaluate the convergence performance and determine the constants if we consider a special case of $T = 1$.

1) ALGORITHM

First, we describe the algorithm for the special case that employs only the first update of loopy CP. As a result, this method turns out to be the diffusion LMS using a novel combination rule as shown in what follows.

By substituting $(\boldsymbol{\psi}_k^{(i)})_p$ into $x_{p,k}^{(i)[0]}$ in Algorithm 2, we have

$$K_{p,(u\to k)}^{(i)[1]} = \frac{\beta_k}{1+\beta_k}, \tag{22}$$

$$\theta_{p,(u\to k)}^{(i)[1]} = x_{p,u}^{(i)[0]} = (\boldsymbol{\psi}_u^{(i)})_p. \tag{23}$$

We omit the subscript p in $K_{p,(u\to k)}^{(i)[1]}$ as $K_{(u\to k)}^{(i)[1]}$ because $K_{p,(u\to k)}^{(i)[1]}$ does not depend on p, and thus the element-wise update rule in Algorithm 2 can be rewritten as a vector-wise update as

$$K_{(u\to k)}^{(i)[1]} = \frac{\beta_k}{1+\beta_k}, \tag{24}$$

$$\theta_{(u\to k)}^{(i)[1]} = \boldsymbol{\psi}_u^{(i)}. \tag{25}$$

Moreover, deriving $x_k^{(i)[1]}$, namely, $\boldsymbol{\phi}_k^{(i)}$ by using (24) and (25) gives

$$\boldsymbol{\phi}_k^{(i)} = \frac{1+\beta_k}{1+|\mathcal{N}_k|\beta_k}\boldsymbol{\psi}_k^{(i)} + \frac{\beta_k}{1+|\mathcal{N}_k|\beta_k}\sum_{u\in\mathcal{N}_k\setminus k}\boldsymbol{\psi}_u^{(i)}. \tag{26}$$

This can be regarded as a novel combination rule summarized as

$$a_{lk} = \begin{cases} \dfrac{\beta_k}{1+|\mathcal{N}_k|\beta_k} & \text{if } l \in \mathcal{N}_k \text{ and } l \neq k \\ \dfrac{1+\beta_k}{1+|\mathcal{N}_k|\beta_k} & \text{if } k = l \\ 0 & \text{otherwise,} \end{cases} \tag{27}$$

which satisfies $\mathbf{1}^{\mathrm{T}}\boldsymbol{A} = \mathbf{1}^{\mathrm{T}}$.

2) OPTIMAL CONSTANT

As mentioned in Sect. II-B, how to select β_k has been an open issue [20]. In this section, we choose β_k that minimizes the steady-state network MSD of the diffusion LMS algorithm (9). We introduce the following reasonable assumptions in all subsequent discussions as in the case in Sect. III-B.

- The noise process $\{v_k^{(i)}\}$ is temporally white and spatially independent.
- The measurement vector process $\{\boldsymbol{u}_k^{(i)}\}$ is temporally white and spatially independent.
- $v_k^{(i)}$ is independent of $\boldsymbol{u}_l^{(j)}$ for all $l \neq k$ and $j \neq i$.
- The step-sizes $\{\mu_k\}$ are sufficiently small.

It can be shown that the upper bound of the $\mathrm{MSD}^{\mathrm{nw}}$ is proportional to [8], [16], [17]

$$\sum_{k=1}^{N}\sum_{l=1}^{N}\gamma_l^2 a_{lk}^2, \tag{28}$$

where $\gamma_l^2 = \mu_l^2\sigma_l^2\mathrm{Tr}(\boldsymbol{R}_{u_l})$. For the proposed combination rule (27), the optimization problem to determine $\{\beta_k\}$ is written as

$$\{\beta_k^{\mathrm{opt}}\}_{k=1}^N = \arg\min_{\{\beta_k\}_{k=1}^N} \sum_{k=1}^{N}\sum_{l=1}^{N}\gamma_l^2\, a_{lk}^2, \quad \text{s.t. (27).} \tag{29}$$

This optimization problem can be separated into independent N subproblems as

$$\beta_k^{\mathrm{opt}} = \arg\min_{\beta_k} f(\beta_k), \tag{30}$$

where

$$f(\beta_k) = \gamma_k^2\left(\frac{1+\beta_k}{1+|\mathcal{N}_k|\beta_k}\right)^2 + \sum_{l\in\mathcal{N}_k\setminus k}\gamma_l^2\left(\frac{\beta_k}{1+|\mathcal{N}_k|\beta_k}\right)^2,$$

and β_k is positive. The above function $f(\beta_k)$ is differentiable and we have

$$\frac{\partial f}{\partial \beta_k} = \frac{2}{(1+|\mathcal{N}_k|\beta_k)^4}\left\{\left(\sum_{l\in\mathcal{N}_k}\gamma_l^2 - |\mathcal{N}_k|\gamma_k^2\right)|\mathcal{N}_k|\beta_k^2 + \left(\sum_{l\in\mathcal{N}_k}\gamma_l^2 - |\mathcal{N}_k|^2\gamma_k^2\right)\beta_k + (1-|\mathcal{N}_k|)\gamma_k^2\right\}. \quad (31)$$

In line with this equation, the shape of the cost function $f(\beta_k)$ largely depends on $A_k = \sum_{l\in\mathcal{N}_k}\gamma_l^2 - |\mathcal{N}_k|\gamma_k^2 \neq 0$ in the first term of the right side of (31). When $A_k > 0$, the function $f(\beta_k)$ has a global minimum in $\beta_k > 0$ and the optimal value is obtained as

$$\beta_k^{\min} = \frac{(|\mathcal{N}_k|-1)\gamma_k^2}{A_k}. \quad (32)$$

On the other hand, it becomes monotonically decreasing function of $\beta_k > 0$ when $A_k < 0$. Thus, in summary, the optimum β_k is given by

$$\beta_k^{\mathrm{opt}} = \begin{cases} \beta_k^{\min} & \text{if } A_k > 0 \\ +\infty & \text{otherwise,} \end{cases} \quad (33)$$

and the corresponding combination weights are given by

$$a_{lk}^{\mathrm{cp}} = \begin{cases} \dfrac{\beta_k^{\mathrm{opt}}}{1+|\mathcal{N}_k|\beta_k^{\mathrm{opt}}} & \text{if } l\in\mathcal{N}_k \text{ and } l\neq k \\ \dfrac{1+\beta_k^{\mathrm{opt}}}{1+|\mathcal{N}_k|\beta_k^{\mathrm{opt}}} & \text{if } k = l \\ 0 & \text{otherwise.} \end{cases} \quad (34)$$

We name this combination rule as CP rule.

In the existing works [8], [16], [17], the upper bound of the $\mathrm{MSD}^{\mathrm{nw}}$ in (28) has been used to determine $\{a_{lk}\}$ in (4) as

$$\{a_{lk}^{\mathrm{opt}}\}_{k=1}^N = \underset{\{a_{lk}\}_{k=1}^N}{\arg\min} \sum_{l=1}^N \gamma_l^2\, a_{lk}^2, \quad \text{s.t.} \sum_{l=1}^N a_{lk} = 1, \quad a_{lk} = 0 \text{ if } l \notin \mathcal{N}_k. \quad (35)$$

The solution results in the relative-variance rule [17] given by

$$a_{lk}^{\mathrm{rv}} = \begin{cases} \dfrac{[\gamma_l^2]^{-1}}{\sum_{m\in\mathcal{N}_k}[\gamma_m^2]^{-1}} & \text{if } l \in \mathcal{N}_k \\ 0 & \text{otherwise.} \end{cases} \quad (36)$$

3) ADAPTIVE COMBINER

The CP rule (34) and the conventional relative-variance rule (36) require the knowledge of γ_l^2, which depends on locally unavailable network statistics such as the correlation matrices of the measurement vectors and the measurement noise profile. Thus, in [8], [16], and [17], the estimation method of γ_l^2 at each node is proposed as

$$\gamma_{lk}^{2,(i)} = (1-\nu_k)\gamma_{lk}^{2,(i-1)} + \nu_k\|\boldsymbol{\psi}_l^{(i)} - \boldsymbol{\phi}_k^{(i-1)}\|^2, \quad (37)$$

Algorithm 3 Diffusion LMS With Adaptive Relative-Variance Rule [16]

1: Initialization: $\boldsymbol{\phi}_k^{(-1)} = 0$
2: **for** each time $i \geq 0$, each node k, and each neighbor l **do**
3: $\quad \boldsymbol{\psi}_k^{(i)} = \boldsymbol{\phi}_k^{(i-1)} + \mu_k \boldsymbol{u}_k^{(i)}(d_k^{(i)} - \boldsymbol{u}_k^{(i)\mathrm{H}}\boldsymbol{\phi}_k^{(i-1)})$
4: $\quad \gamma_{lk}^{2,(i)} = (1-\nu_k)\gamma_{lk}^{2,(i-1)} + \nu_k\|\boldsymbol{\psi}_l^{(i)} - \boldsymbol{\phi}_k^{(i-1)}\|^2$
5: $\quad a_{lk}^{(i)} = \frac{[\gamma_{lk}^{2,(i)}]^{-1}}{\sum_{m\in\mathcal{N}_k}[\gamma_{mk}^{2,(i)}]^{-1}}$
6: $\quad \boldsymbol{\phi}_k^{(i)} = \sum_{l\in\mathcal{N}_k} a_{lk}^{(i)}\boldsymbol{\psi}_l^{(i)}$
7: **end for**

where ν_k is a forgetting factor ($0 < \nu_k < 1$) and $\gamma_{lk}^{2,(i)}$ is the estimate of γ_l^2 at node k and time i. By using this estimate, the adaptive version of (36) is proposed in [16] as

$$a_{lk}^{\mathrm{rv}(i)} = \begin{cases} \dfrac{[\gamma_{lk}^{2,(i)}]^{-1}}{\sum_{m\in\mathcal{N}_k}[\gamma_{mk}^{2,(i)}]^{-1}} & \text{if } l \in \mathcal{N}_k \\ 0 & \text{otherwise.} \end{cases} \quad (38)$$

In the same manner, the adaptive version of the CP rule is given by

$$\beta_k^{(i)} = \begin{cases} \dfrac{(|\mathcal{N}_k|-1)\gamma_{kk}^{2,(i)}}{A_k^{(i)}} & \text{if } A_k^{(i)} > 0 \\ +\infty & \text{otherwise,} \end{cases} \quad (39)$$

$$a_{lk}^{\mathrm{cp}(i)} = \begin{cases} \dfrac{\beta_k^{(i)}}{1+|\mathcal{N}_k|\beta_k^{(i)}} & \text{if } l\in\mathcal{N}_k \text{ and } l\neq k \\ \dfrac{1+\beta_k^{(i)}}{1+|\mathcal{N}_k|\beta_k^{(i)}} & \text{if } k = l \\ 0 & \text{otherwise,} \end{cases} \quad (40)$$

where $A_k^{(i)} = \sum_{l\in\mathcal{N}_k}\gamma_{lk}^{2,(i)} - |\mathcal{N}_k|\gamma_{kk}^{2,(i)}$.

The algorithms of the diffusion LMS using the conventional adaptive combination rule in (38) and this combination rule as in (40) are summarized in Algorithm 3 and Algorithm 4, respectively. The computational complexity of Algorithm 3 and Algorithm 4 are almost the same because the complexity of the proposed rule (40) becomes comparable to that of the conventional rule (38) by substituting (39) into (40).

4) CONVERGENCE ANALYSIS

In this section, we analyze the mean stability, the mean-square stability, and the transient behavior of the diffusion LMS using the proposed CP rules. Note that the transient behavior of the diffusion LMS using the adaptive combination rules (not only CP rule but also relative-variance rule) remains an open issue. So, the theoretical network MSD learning curve below will not be valid for the adaptive CP rule, while other theoretical results are applicable to the adaptive CP rule as well because it is identical to the static one at the steady-state. Note also that the conventional performance analysis framework for the diffusion LMS is applicable for the

Algorithm 4 Diffusion LMS With Proposed Adaptive CP Rule

1: Initialization: $\boldsymbol{\phi}_k^{(-1)} = 0$
2: **for** each time $i \geq 0$, each node k, and each neighbor l **do**
3: $\quad \boldsymbol{\psi}_k^{(i)} = \boldsymbol{\phi}_k^{(i-1)} + \mu_k \boldsymbol{u}_k^{(i)}(d_k^{(i)} - \boldsymbol{u}_k^{(i)\mathrm{H}}\boldsymbol{\phi}_k^{(i-1)})$
4: $\quad \gamma_{lk}^{2,(i)} = (1-\nu_k)\gamma_{lk}^{2,(i-1)} + \nu_k \|\boldsymbol{\psi}_l^{(i)} - \boldsymbol{\phi}_k^{(i-1)}\|^2$
5: $\quad$ **if** $\sum_{l\in\mathcal{N}_k} \gamma_{lk}^{2,(i)} - \gamma_{kk}^{2,(i)}|\mathcal{N}_k| > 0$ **then**
6: $\quad\quad \beta_k^{(i)} = \frac{(|\mathcal{N}_k|-1)\gamma_{kk}^{2,(i)}}{\sum_{l\in\mathcal{N}_k} \gamma_{lk}^{2,(i)} - \gamma_{kk}^{2,(i)}|\mathcal{N}_k|}$
7: $\quad$ **else**
8: $\quad\quad \beta_k^{(i)} = +\infty$ (large positive constant)
9: $\quad$ **end if**
10: $\quad a_{lk}^{(i)} = \frac{\beta_k^{(i)}}{1+|\mathcal{N}_k|\beta_k^{(i)}}\ (l \in \mathcal{N}_k \setminus k),\quad \frac{1+\beta_k^{(i)}}{1+|\mathcal{N}_k|\beta_k^{(i)}}\ (l = k)$
11: $\quad \boldsymbol{\phi}_k^{(i)} = \sum_{l\in\mathcal{N}_k} a_{lk}^{(i)} \boldsymbol{\psi}_l^{(i)}$
12: **end for**

proposed diffusion LMS using CP rule because CP rule can be regarded as one of the combination rules.

First, the condition that the estimate error $\tilde{\boldsymbol{w}}_k^{(i)}$ converges to $\mathbf{0}_M$ as $i \to \infty$ in the mean and mean-square sense is given by [5], [7]

$$0 < \mu_k < \frac{2}{\lambda_{\max}(\boldsymbol{R}_{u_k})}. \tag{41}$$

Moreover, the transient behavior of the diffusion LMS using CP rule (34) can be represented by the same expression as [5]. Now let

$$\boldsymbol{F}' = (\boldsymbol{I}_{(MN)^2} - \boldsymbol{I}_{MN} \otimes (\mathcal{D}\mathcal{M}) - (\mathcal{D}^{\mathrm{T}}\mathcal{M}) \otimes \boldsymbol{I}_{MN})\,(\mathcal{A} \otimes \mathcal{A}),$$

where $\mathcal{A} = \boldsymbol{A} \otimes \boldsymbol{I}_M$, $\mathcal{D} = \mathrm{diag}\left\{\boldsymbol{R}_{u_1}, \ldots, \boldsymbol{R}_{u_N}\right\}$, and $\mathcal{M} = \mathrm{diag}\{\mu_1 \boldsymbol{I}_M, \ldots, \mu_N \boldsymbol{I}_M\}$, then the theoretical network MSD learning curve of the diffusion LMS using CP rule is given by

$$\eta'^{(i)} = \eta'^{(i-1)} + \frac{1}{N}\boldsymbol{r}'^{\mathrm{T}}\boldsymbol{F}'^{i}\boldsymbol{q}' - \frac{1}{N}\boldsymbol{w}^{\mathrm{T}}\left[\mathrm{vec}^{-1}\left(\boldsymbol{F}'^{i}\left[\boldsymbol{I}_{(MN)^2} - \boldsymbol{F}'\right]\boldsymbol{q}'\right)\right]\boldsymbol{w}, \tag{42}$$

where $\eta'^{(i)}$ is the network MSD at time i, $\boldsymbol{r}' = \mathrm{vec}\left(\mathcal{A}^{\mathrm{T}}\mathcal{M}\mathcal{G}^{\mathrm{T}}\mathcal{M}\mathcal{A}\right)$, $\mathcal{G} = \mathrm{diag}\left\{\sigma_1^2\boldsymbol{R}_{u_1}, \ldots, \sigma_N^2\boldsymbol{R}_{u_N}\right\}$, $\boldsymbol{w} = \mathbf{1}_N \otimes \boldsymbol{w}^{\mathrm{o}}$, and $\boldsymbol{q}' = \mathrm{vec}(\boldsymbol{I}_{MN})$. The steady-state network MSD and the steady-state MSD at each node k are given by

$$\mathrm{MSD}^{\mathrm{nw}} = \frac{1}{N}\boldsymbol{r}'^{\mathrm{T}}(\boldsymbol{I}_{(MN)^2} - \boldsymbol{F}')^{-1}\boldsymbol{q}', \tag{43}$$
$$\mathrm{MSD}_k = \boldsymbol{r}'^{\mathrm{T}}(\boldsymbol{I}_{(MN)^2} - \boldsymbol{F}')^{-1}\mathrm{vec}(\mathcal{J}_k), \tag{44}$$

respectively, where $\mathcal{J}_k$ is $N \times N$ block diagonal matrix with $M \times M$ blocks having all zero block matrices except for the identity matrix on the k-th diagonal block.

V. COMPUTATIONAL COMPLEXITY AND COMMUNICATION COST

We compare the computational complexity and the communication cost of the proposed schemes with those of the conventional methods in this section. Table 1 shows the complexity and the cost of the proposed algorithms and other algorithms in each iteration. Note that, since the LMS update is common for all algorithms including the proposed schemes, the complexities only in the averaging step are evaluated. Here, the communication cost is defined as the number of scalar values that node k has to receive. We can find that the proposed CP-LMS requires higher computational complexity and communication cost than other algorithms because it contains exchanges of two types of messages between nodes. However, the orders are common for all the algorithms with respect to the communication cost which mainly dominates the execution time in applications such as wireless sensor networks.

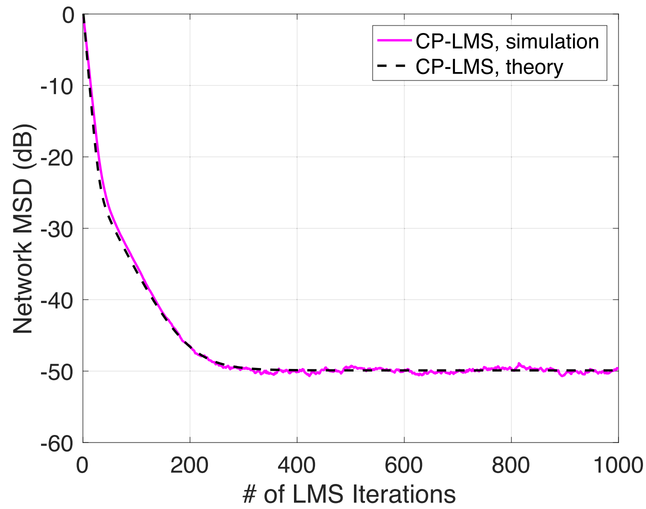

FIGURE 4. Network MSD learning curves of the proposed CP-LMS.

VI. SIMULATION RESULTS

In this section, we compare the network MSD learning curves of the proposed schemes with the theoretical results and those of the conventional schemes via computer simulations. All the simulation results have been obtained by using MATLAB, and we have implemented all algorithms by ourselves without any toolbox. We assume that the measurement vectors $\{\boldsymbol{u}_k^{(i)}\}$ are zero-mean circular Gaussian random vectors with sizes $M = 5$ and have time-correlated shift structures [2]. The specific structure is given by

$$\boldsymbol{u}_k^{(i)} = [u_k(i)\ u_k(i-1)\ \cdots\ u_k(i-M+1)]^{\mathrm{T}}, \tag{45}$$

where

$$u_k(i) = \alpha_k u_k(i-1) + \sqrt{\sigma_{u_k}^2(1-\alpha_k^2)}z_k(i) \quad (i > -\infty). \tag{46}$$

$\alpha_k \in [0, 1)$ and $\sigma_{u_k}^2 \in (0, 1]$ are chosen from uniform distribution, and $z_k(i)$ is a spatially independent white Gaussian process with unit variance. In this case, a trace of the correlation matrix can be obtained as $\mathrm{Tr}(\boldsymbol{R}_{u_k}) = M\sigma_{u_k}^2$. We use the common step-size parameters $\mu_k = \mu$, the common forgetting factors $\nu_k = \nu$, and the common initial values $\gamma_{lk}^{2,(0)} = \gamma^{2,(0)}$, for all k, l. The unknown vector is set to

TABLE 1. Computational complexity and communication cost for the averaging step of the proposed algorithms and other conventional algorithms.

Algorithm	$\times,\div$	$+,-$	Communication cost
diffusion LMS w/ static rules	$M\lvert\mathcal{N}_k\rvert$	$M(\lvert\mathcal{N}_k\rvert-1)$	$M(\lvert\mathcal{N}_k\rvert-1)$
AMC diffusion	$M\lvert\mathcal{N}_k\rvert$	$M(\lvert\mathcal{N}_k\rvert-1)$	$M(\lvert\mathcal{N}_k\rvert-1)$
CP-LMS (proposed)	$M(2\lvert\mathcal{N}_k\rvert-3)(\lvert\mathcal{N}_k\rvert-1)$ $+M(2\lvert\mathcal{N}_k\rvert-1)/J$	$M(\lvert\mathcal{N}_k\rvert-2)$ $\cdot(2\lvert\mathcal{N}_k\rvert-2+1/J)$	$2M(\lvert\mathcal{N}_k\rvert-1)$
diffusion LMS w/ adaptive relative-variance rule	$(3M+5)\lvert\mathcal{N}_k\rvert$	$(3M+2)\lvert\mathcal{N}_k\rvert-M-1$	$M(\lvert\mathcal{N}_k\rvert-1)$
diffusion LMS w/ adaptive CP rule (proposed)	$(2M+4)\lvert\mathcal{N}_k\rvert+3$	$(3M+5)\lvert\mathcal{N}_k\rvert-M+1$	$M(\lvert\mathcal{N}_k\rvert-1)$

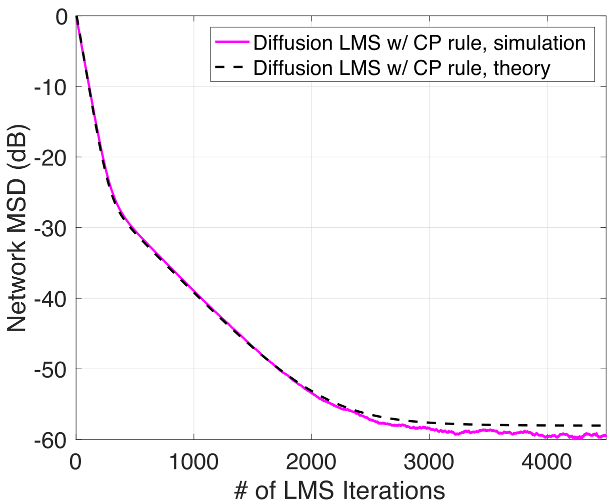

FIGURE 5. Network MSD learning curves of the diffusion LMS with proposed CP rule.

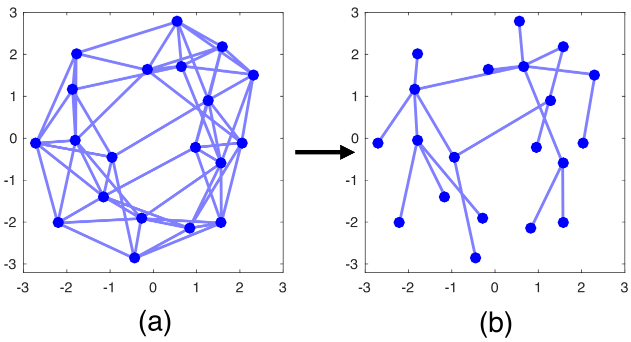

FIGURE 6. (a) Original network topology with $N = 20$. (b) Spanning tree extracted from original network.

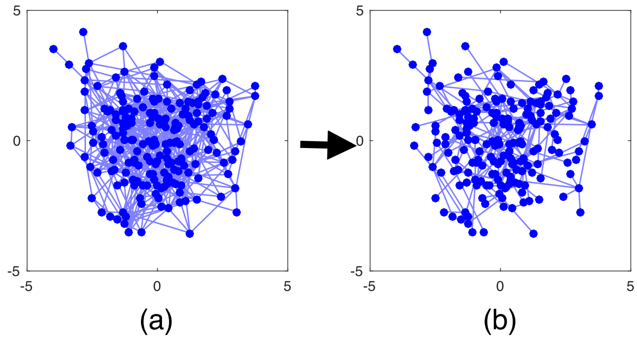

FIGURE 7. (a) Original network topology with $N = 200$. (b) Spanning tree extracted from original network.

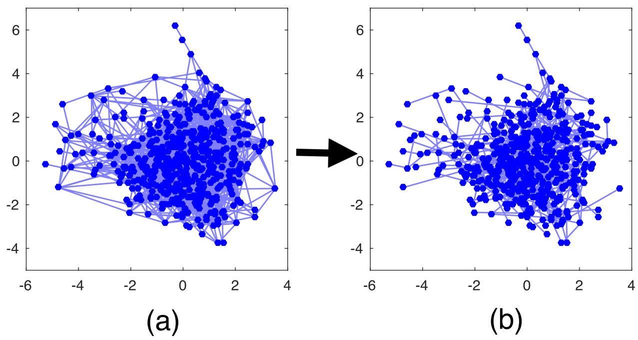

FIGURE 8. (a) Original network topology with $N = 500$. (b) Spanning tree extracted from original network.

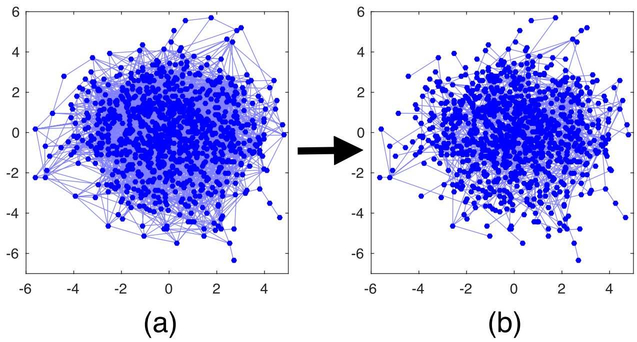

FIGURE 9. (a) Original network topology with $N = 1000$. (b) Spanning tree extracted from original network.

be $\boldsymbol{w}^{\mathrm{o}} = \frac{1}{\sqrt{M}}\mathbf{1}_M$. All simulation results are obtained by averaging over 100 independent trials.

A. COMPARISON WITH THEORETICAL CURVE

In Figs. 4 and 5, we compare the learning curves of the CP-LMS and the diffusion LMS using the static CP rule obtained by simulations with the theoretical curves using (20) and (42), respectively. We use only a small network with $N = 20$ in Fig. 6 due to the computational difficulty of the theoretical values. The simulation of CP-LMS and its theoretical calculation are performed on the spanning tree shown in Fig. 6(b), and that of the diffusion LMS using CP rule and its theoretical calculation are performed on the original network shown in Fig. 6(a). The step-size parameters are $\mu = 0.08$ ($\mu' = 0.004$) in CP-LMS and $\mu = 0.01$ in the diffusion LMS with CP rule, respectively. The measurement noise power is set to be $\sigma_k^2 = 10^{-3}$ at any node. Fig. 4 shows that the simulation result agrees well with the theoretical curve. On the other hand, in Fig. 5, we find a slight disagreement between the simulation and the theoretical results at the steady-state. This will be due to the assumption of independence of the measurement vectors in Sect. IV-B2.

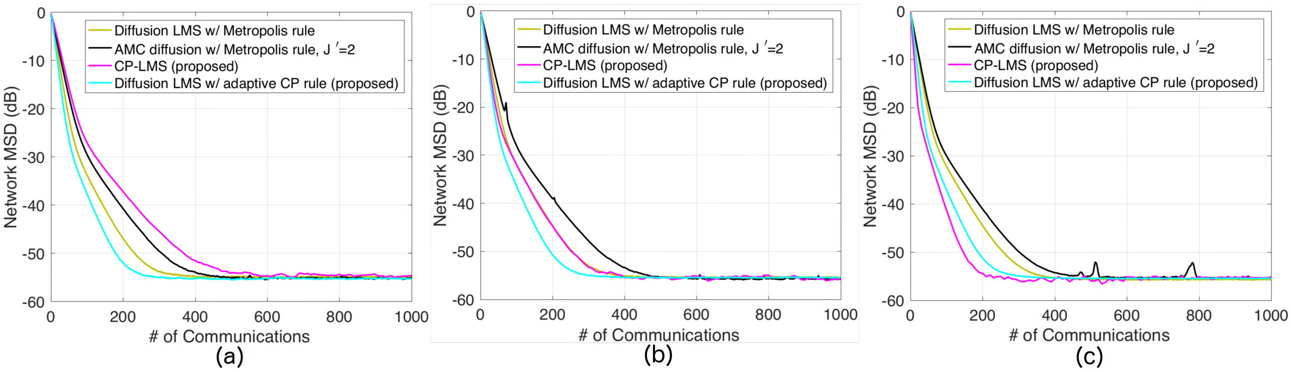

FIGURE 10. Network MSD learning curves of proposed methods and conventional methods versus number of communications. (a) $N = 200$ (b) $N = 500$ (c) $N = 1000$.

B. PROPOSED METHODS V.S. CONVENTIONAL DIFFUSION SCHEMES

Here, we compare the performance of the proposed CP-LMS and the diffusion LMS using the proposed adaptive CP rule with that of the diffusion LMS and the AMC diffusion LMS using the conventional Metropolis rule (6). We have generated random networks with $N = 200, 500$, and 1000 as shown in Figs. 7(a), 8(a), and 9(a) to compare the performance in networks of different sizes. Figs. 7(b), 8(b), and 9(b) show the spanning trees extracted from the original networks so that the diameter of the tree becomes the smallest among all possible choices, i.e., the minimum diameter spanning trees. The diameters of the trees with $N = 200, 500$, and 1000 are $J = 8$, 10, and 10, respectively. Here, we adopt the diffusion LMS algorithms to the original connected networks and the CP-LMS to the minimum diameter spanning trees, because the original networks are preferable for the diffusion LMS due to their larger degrees. The number of iterations of the AMC diffusion LMS and the measurement noise power are set to be $J' = 2$ and $\sigma_k^2 = 10^{-3}$ at any node, respectively.

Fig. 10 shows the network MSD learning curves for the proposed and the conventional methods for the networks with different sizes versus the number of communications. We have controlled the step-size parameters of the algorithms so that the steady-state performance becomes comparable for all algorithms. In the figures, the diffusion LMS using the proposed adaptive CP rule outperforms the conventional methods for all cases. As for CP-LMS, though the conventional methods achieve faster convergence in Fig. 10(a), CP-LMS converges faster as the number of nodes increases and it outperforms the conventional methods and the diffusion LMS using the proposed adaptive CP rule in Fig. 10(c). These results are consistent with the expectation that the proposed CP-LMS would be valid especially for large-scale networks. This is because the consensus protocol employed in the conventional LMS generally achieves slower convergence when the network size is large, while the convergence rate of CP depends only on the diameter of the graph. Note that the reason why the AMC diffusion LMS converges slower than the diffusion LMS is that this paper evaluates the network MSD with respect to the number of communications instead of the number of LMS updates.

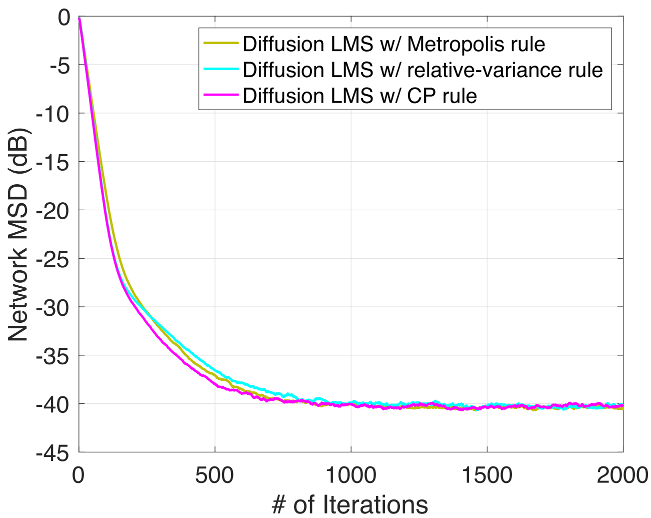

FIGURE 11. Network MSD learning curves of proposed CP rule and conventional static combination rules with $N = 20$ in Fig. 6(a).

C. PROPOSED CP RULE AND ADAPTIVE CP RULE V.S. CONVENTIONAL RULES

Finally, we compare the performance of the proposed diffusion LMS using the static CP rule (34) and the adaptive CP rule (40) with that of conventional diffusion LMS using the static Metropolis rule (6), the static relative-variance rule (36), and the adaptive relative-variance rule (38). We use the network with $N = 20$ shown in Fig. 6(a). The measurement noise power σ_k^2 is changed among nodes in order to confirm the estimation ability of the proposed adaptive CP rule. We set σ_k^2 in proportion to the square of the distance from the origin. This setting can be understood as a model that the power of target signal decays with the square of distance assuming the observation target is located at the origin, while the measurement noise power is uniform for all observation nodes. The initial value and the forgetting factor of the adaptation in Algorithm 3 and 4 are $\gamma^{2,(0)} = 4.5 \times 10^{-2}$ and $\nu = 0.07$. We have controlled the step-size parameters of

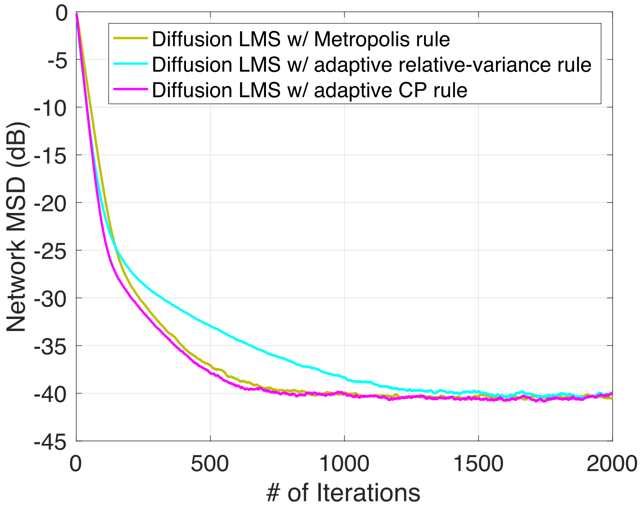

FIGURE 12. **Network MSD learning curves of proposed adaptive CP rule and conventional combination rules with $N = 20$ in Fig. 6(a).**

the algorithms so that the steady-state performance becomes comparable for all methods.

Fig. 11 shows the learning curves of the diffusion LMS using the static CP rule, Metropolis rule, and the static relative-variance rule in terms of the network MSD assuming that true γ_l^2s are known to each node. In the figure, we see that the diffusion LMS using the CP rule converges faster than that using other rules. Fig. 12 shows the learning curves of the diffusion LMS using the proposed adaptive CP rule, Metropolis rule, and the adaptive relative-variance rule in order to verify the influence of weight adaptation. The adaptive CP rule can achieve comparable performance as in Fig. 11, while the performance of the adaptive relative-variance rule is significantly degraded.

VII. CONCLUSION

In this paper, we have proposed novel diffusion LMS algorithms for in-network signal processing based on the idea of the message passing algorithm of CP. By using CP on the spanning tree of the original network, CP-LMS can achieve the same solution as the centralized LMS in a fully distributed manner. LCP-LMS and its special case of CP rule are based on loopy CP, and we have optimized the constants involved in CP rule in terms of the steady-state MSD of the diffusion LMS. Moreover, we have shown that their theoretical learning curves and steady-state MSDs can be obtained using existing frameworks. Also, we have extended the CP rule to an adaptive version. Simulation results show that the proposed CP-LMS can achieve better performance than the conventional diffusion LMS especially in large-scale networks, and that the diffusion LMS with the static and the adaptive CP rules can achieve better performance than that with the conventional combination rules when the measurement noise power depends on the nodes.

Future work includes the extension of the proposed CP rule to more flexible weight control method using asymmetric updates in loopy CP, i.e., using different constant at each node depending on the direction of the messages.

ACKNOWLEDGMENT

This paper was presented in part at the APSIPA ASC 2017 and ICASSP 2018.

REFERENCES

[1] S. Hara, H. Yomo, P. Popovski, and K. Hayashi, "New paradigms in wireless communication systems," *Wireless Pers. Commun.*, vol. 37, nos. 3–4, pp. 233–241, May 2006.

[2] 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.

[3] C. G. Lopes and A. H. Sayed, "Diffusion least-mean squares over adaptive networks: Formulation and performance analysis," *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 3122–3136, Jul. 2008.

[4] I. D. Schizas, G. Mateos, and G. B. Giannakis, "Distributed LMS for consensus-based in-network adaptive processing," *IEEE Trans. Signal Process.*, vol. 57, no. 6, pp. 2365–2382, Jun. 2009.

[5] 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.

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

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

[8] A. H. Sayed, "Diffusion adaptation over networks,"in *Library in Signal Processing*, vol. 3, R. Chellapa and S. Theodoridis, Eds. New York, NY, USA: Academic, 2014, pp. 323–454.

[9] I. E. K. Harrane, R. Flamary, and C. Richard, "On reducing the communication cost of the diffusion LMS algorithm," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 5, no. 1, pp. 100–112, Mar. 2019.

[10] J.-T. Kong, J.-W. Lee, S.-E. Kim, and W.-J. Song, "Diffusion LMS algorithms with multi combination for distributed estimation: Formulation and performance analysis," *Digit. Signal Process.*, vol. 71, pp. 117–130, Dec. 2017.

[11] R. Nassif, C. Richard, A. Ferrari, and A. H. Sayed, "Diffusion LMS for multitask problems with local linear equality constraints," *IEEE Trans. Signal Process.*, vol. 65, no. 19, pp. 4979–4993, Oct. 2017.

[12] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proc. IEEE*, vol. 95, no. 1, pp. 215–233, Jan. 2007.

[13] V. D. Blondel, J. M. Hendrickx, A. Olshevsky, and J. N. Tsitsiklis, "Convergence in multiagent coordination, consensus, and flocking," in *Proc. CDC-ECC*, Dec. 2005, pp. 2996–3000.

[14] D. S. Scherber and H. C. Papadopoulos, "Locally constructed algorithms for distributed computations in ad-hoc networks," in *Proc. IPSN*, Apr. 2004, pp. 11–19.

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

[16] X. Zhao, S.-Y. Tu, and A. H. Sayed, "Diffusion adaptation over networks under imperfect information exchange and non-stationary data," *IEEE Trans. Signal Process.*, vol. 60, no. 7, pp. 3460–3475, Jul. 2012.

[17] S.-Y. Tu and A. H. Sayed, "Optimal combination rules for adaptation and learning over networks," in *Proc. IEEE CAMSAP*, Dec. 2011, pp. 317–320.

[18] N. Takahashi, I. Yamada, and A. H. Sayed, "Diffusion least-mean squares with adaptive combiners: Formulation and performance analysis," *IEEE Trans. Signal Process.*, vol. 58, no. 9, pp. 4795–4810, Sep. 2010.

[19] A. Nakai and K. Hayashi, "Diffusion LMS using consensus propagation," in *Proc. APSIPA ASC*, Kuala Lumpur, Malaysia, Dec. 2017, pp. 943–948.

[20] C. C. Moallemi and B. Van Roy, "Consensus propagation," *IEEE Trans. Inf. Theory*, vol. 52, no. 11, pp. 4753–4766, Nov. 2006.

[21] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo, CA, USA: Morgan Kaufmann, 1988.

[22] F. Krzakala *et al.*, "Spectral redemption in clustering sparse networks," *Proc. Nat. Acad. Sci. USA*, vol. 110, no. 52, pp. 20935–20940, Dec. 2013.

[23] A. Decelle, F. Krzakala, C. Moore, and L. Zdeborová, "Inference and phase transitions in the detection of modules in sparse networks," *Phys. Rev. Lett.*, vol. 107, no. 6, 2011, Art. no. 065701.

[24] A. Coja-Oghlan, E. Mossel, and D. Vilenchik, "A spectral approach to analysing belief propagation for 3-colouring," *Combinatorics, Probab. Comput.*, vol. 18, no. 6, pp. 881–912, Mar. 2009.

[25] J. Herzen, C. Westphal, and P. Thiran, "Scalable routing easy as PIE: A practical isometric embedding protocol," in *Proc. IEEE ICNP*, Oct. 2011, pp. 49–58.

[26] M. Bui, F. Butelle, and C. Lavault, "A distributed algorithm for constructing a minimum diameter spanning tree," *J. Parallel Distrib. Comput.*, vol. 64, no. 5, pp. 571–577, May 2004.

[27] A. Nakai and K. Hayashi, "An adaptive combination rule for diffusion LMS based on consensus propagation," in *Proc. ICASSP*, Apr. 2018, pp. 3839–3843.

[28] F. S. Cattivelli and A. H. Sayed, "Analysis of spatial and incremental LMS processing for distributed estimation," *IEEE Trans. Signal Process.*, vol. 59, no. 4, pp. 1465–1480, Apr. 2011.

AYANO NAKAI-KASAI (S'18) received the B.E. and M.E. degrees from Kyoto University, Kyoto, Japan, in 2016 and 2018, respectively, where she is currently pursuing the Ph.D. degree with the Department of Systems Science, Graduate School of Informatics. Her research interests include signal processing and wireless communication. She received the Young Researchers' Award from the Institute of Electronics, Information and Communication Engineers, in 2018.

KAZUNORI HAYASHI (S'97–M'03) received the B.E., M.E., and Ph.D. degrees in communication engineering from Osaka University, Osaka, Japan, in 1997, 1999, and 2002, respectively. He is currently a Professor with the Graduate School of Engineering, Osaka City University. He was an Assistant Professor (2002–2007), and an Associate Professor (2007–2017) with the Graduate School of Informatics, Kyoto University. His research interests include statistical signal processing for communication systems. He is a member of the IEICE, APSIPA, and ISCIE. He received the ICF Research Award from the KDDI Foundation, in 2008; the IEEE GLOBECOM 2009 Best Paper Award; the IEICE Communications Society Best Paper Award, in 2010; the WPMC'11 Best Paper Award, the Telecommunications Advancement Foundation Award, in 2012; and the IEICE Communications Society Best Tutorial Paper Award, in 2013.