SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

November 2018, Vol. 61 112210:1–112210:15
https://doi.org/10.1007/s11432-018-9461-3

Distributed bias-compensated normalized least-mean squares algorithms with noisy input

Lu FAN, Lijuan JIA*, Ran TAO & Yue WANG

School of Information and Electronics, Beijing Institute of Technology, Beijing 100081*, China*

Received 8 January 2018/Revised 30 March 2018/Accepted 3 May 2018/Published online 18 October 2018

Abstract In this paper, we study the problem of distributed normalized least-mean squares (NLMS) estimation over multi-agent networks, where all nodes collaborate to estimate a common parameter of interest. We consider the situations that all nodes in the network are corrupted by both input and output noise. This yields into biased estimates by the distributed NLMS algorithms. In our analysis, we take all the noise into consideration and prove that the bias is dependent on the input noise variance. Therefore, we propose a bias compensation method to remove the noise-induced bias from the estimated results. In our development, we first assume that the variances of the input noise are known a priori and develop a series of distributed-based bias-compensated NLMS (BCNLMS) methods. Under various practical scenarios, the input noise variance is usually unknown a priori, therefore it is necessary to first estimate for its value before bias removal. Thus, we develop a real-time estimation method for the input noise variance, which overcomes the unknown property of this noise. Moreover, we perform some main analysis results of the proposed distributed BCNLMS algorithms. Furthermore, we illustrate the performance of the proposed distributed bias compensation method via graphical simulation results.

Keywords distributed parameter estimation, normalized least-mean squares, bias-compensated algorithms

Citation Fan L, Jia L J, Tao R, et al. Distributed bias-compensated normalized least-mean squares algorithms with noisy input. Sci China Inf Sci, 2018, 61(11): 112210, https://doi.org/10.1007/s11432-018-9461-3

1 Introduction

One of the critical issues encountered in distributed parameter estimation is that both the input and the output are corrupted by additive noise such as measurement error or quantization noise, where the estimation will be biased using common distributed adaptive filtering algorithms. The problem has been extensively studied for the case of single-node and several solutions have been proposed to mitigate the effect of additive noise or remove the estimation bias [1–4]. However, they may not be suitable for the case of distributed parameter estimation, where all the in-network nodes collaborate to perform data measurement and parameter estimation simultaneously.

Recently, distributed solutions are widely studied where agents are permitted to share information with their immediate neighbors. In [5, 6], some distributed parameter estimation algorithms have been proposed to address the problem of cooperative sensing and signals reconstruction. Several distributed Kalman filtering algorithms [7, 8] have been developed for state estimation problems over the network. Comparing with centralized solution, distributed strategies significantly reduce energy consumption for communication while improving the performance and robustness of the network. Three mainstream distributed strategies have been proposed including incremental [9–12], consensus [13–15] and

* Corresponding author (email: jlj@bit.edu.cn)

diffusion [16–19] strategies. For incremental strategy, we need to determine a cyclic path that covers all in-network nodes. All the agents in the network transmit their information to their immediate neighbors over the cyclic path. After the last agent updates its estimate, it forwards this information to the first agent and the loop process repeats. This multi-step updating mechanism improves the estimation performance. Besides, it reduces the communication cost since that each node only cooperates with its adjacent nodes over the path. However, the incremental approach suffers from several drawbacks that include sensitivity to link failure in the cyclic trajectory construction and the limited cooperation among agents. Also, determination of a closed loop path for all agents in the network is an NP-hard problem. For consensus strategy, each node of the networks exchanges information with all its neighbor nodes and updates the estimation of the unknown parameter simultaneously. Thus, it can be applied to a given network using the existing network topology, which improves the robustness and real-time features of the network. However, the cost of communication consumption is increased. Besides, the asymmetric construction of the consensus strategy causes unstable growth in the network state. Just like the consensus approach, in the diffusion strategy, all nodes collaborate with their adjacent agents to estimate the parameter of interest. Besides, the diffusion approach focuses on a symmetrical structure that improves the robustness and stability of the network. However, it suffers from increased consumption of the communication resources. In this paper, we focus on three distributed strategies, which can significantly improve the performance and robustness of the networks through cooperation among the in-network nodes [20–23].

Several distributed solutions have been developed to address the cooperative estimation problem in the presence of additive noise. In [24], a consensus-based distributed total least squares (DTLS) estimation is developed to mitigate the effect of input noise using an equivalent convex semidefinite program and a semidefinite relaxation technique, which requests the in-network nodes provided with powerful processing capability. A diffusion bias-compensated recursive least squares (BC-RLS) algorithm has been proposed in [25], which can obtain unbiased estimations in the case that both the input and output are corrupted by colored noise. This method attains fast convergence speeds at the expense of large computational power requirements. A diffusion-based least-mean squares (DLMS) method has been proposed in [26] with easy implementation, although it requires priori knowledge of the input noise variance, a value that is usually unavailable.

Adaptive filtering algorithms have been extensively studied for decades, with various applications such as channel estimation, system identification and echo cancellation [27, 28]. Among them, recursive least squares (RLS) algorithm and least-mean squares (LMS) algorithm are two common algorithms. For RLS algorithm, it is suitable for the occasions where the fast convergence speed is required but suffers from complex computation. For LMS algorithm, its low computational cost and strong robustness features allow it to be widely used in many industry applications. However, it is difficult to choose an appropriate step size that ensures stability of the LMS algorithm. Moreover, the LMS algorithm suffers from a gradient noise amplification problem when the input data is large. Hence, the normalized LMS (NLMS) algorithm has been developed to inherit the excellent features and overcome the drawbacks of LMS algorithm [29]. For NLMS algorithm, the adjustment applied to the tap-weight vector is normalized with respect to the squared Euclidean norm of the tap-input vector, which solves the gradient noise amplification problem.

In [30], an incremental bias-compensated RLS (Inc-BCRLS) algorithm is developed, which attains fast convergence speed and gives an unbiased estimate in the case where both the input and output are corrupted by additive noises. Besides, a method for estimating input noise variance is proposed to remove the estimation bias. However, it requires high computation cost. Based on the incremental cooperation strategy, each agent only shares information with its local neighbors. It still consumes an amount of communication resource to transmit estimates and matrices. A novel simplified incremental strategy is developed to reduce the communication cost. However, the simplified Inc-BCRLS method performs not that well with the common Inc-BCRLS algorithm. In this paper, we focus on NLMS algorithm and propose a bias-compensated NLMS (BCNLMS) method for its easy implementation and low computational cost, in the presence of noisy inputs. Besides, we propose a series of distributed BCNLMS algorithms, which have the capability to give unbiased estimates. Comparing to Inc-BCRLS algorithms in [30], our proposed methods reduce communication cost as well as improve the stability and robustness of the network. Moreover, we perform some main analysis results of the proposed distributed BCNLMS algorithms.

The contributions of this article are summarized as follows. We initially demonstrate that the common NLMS algorithm attains biased estimates in the case that both the input and output are corrupted by additive noise. We derive expressions that the bias is dependent on the input noise. Then, we propose a method to remove the noise-induced bias from the estimation results. We first assume to have a priori knowledge of the noise variance so that we develop three distributed BCNLMS algorithms. Later on, we make an assumption that the input noise knowledge is unknown a priori to computation. Therefore, a real-time approach to estimate the input noise variance is proposed. We perform some main analysis results of the proposed distributed BCNLMS algorithms. We derive mean stability conditions for different strategies and closed-form expressions to predict the steady-state mean-square performance over the network. Some simulations are conducted to test the performance and robustness of distributed BCNLMS algorithms .

This paper is organized as follows. We define the problem and modify the cost function in Section 2. In Section 3, we develop and present the proposed distributed BCNLMS algorithms, followed by a detailed discussion of the noise variance estimation method. We perform some main analysis results of the proposed distributed BCNLMS algorithms in Section 4. Several experimental simulations are conducted in Section 5 for testing performance of the proposed algorithms, followed by concluding remarks in Section 6.

Notation. In this paper, we use boldface letters to denote vectors and normal letters are reserved for scalars. Matrices and vectors are represented by uppercase letters and lowercase letters respectively. Superscript $(\cdot)^{\mathrm{T}}$ denotes transposition. The operator $\mathrm{E}[\cdot]$ defines the statistical expectation. All vectors in this study are considered as columnar.

2 Problem statement

Consider a multi-agent network consisting of N nodes distributed in different spatial locations. The objective of the network is to estimate a vector parameter target $\boldsymbol{w}_o \in \mathbb{R}^{M\times 1}$ and all nodes are corrupted by input and output noise. At discrete time $i \in \mathbb{N}$, each node $k \in 1, 2, \ldots, N$ collects the distorted input $\boldsymbol{z}_{k,i} \in \mathbb{R}^{M\times 1}$ and the output data $d_k(i) \in \mathbb{R}$ with the relationship

$$\boldsymbol{z}_{k,i} = \boldsymbol{u}_{k,i} + \boldsymbol{n}_{k,i}, \tag{1}$$

$$d_k(i) = \boldsymbol{u}_{k,i}^{\mathrm{T}} \boldsymbol{w}_o + v_k(i), \tag{2}$$

where $\boldsymbol{u}_{k,i} \in \mathbb{R}^{M\times 1}$, $\boldsymbol{n}_{k,i} \in \mathbb{R}^{M\times 1}$ and $v_k(i) \in \mathbb{R}$ denote the noise-free input, the input noise and the output noise, respectively.

Assumption 1. The following assumptions are made for the random variables in measurement model equation (1) and (2).

(a) Both the input data vector $\boldsymbol{u}_{k,i}$ and the input noise vector $\boldsymbol{n}_{k,i}$ are independent and identically distributed (i.i.d.) over time and space, with zero-mean and covariance matrices $\boldsymbol{R}_{u,k} = \mathrm{E}[\boldsymbol{u}_{k,i}\boldsymbol{u}_{k,i}^{\mathrm{T}}]$, $\boldsymbol{R}_{n,k} = \mathrm{E}[\boldsymbol{n}_{k,i}\boldsymbol{n}_{k,i}^{\mathrm{T}}]$, respectively.

(b) The output noise $v_k(i)$ is i.i.d. over time and space, with zero-mean and noise variance $\sigma_{v,k}^2$.

(c) Random variables $\boldsymbol{u}_{k,i}$, $\boldsymbol{n}_{l,j}$, $v_p(m)$ are independent for all k, i, l, j, p and m.

We first consider the estimation problem in the absence of the input noise. In this case, traditional distributed NLMS method updates the estimation results by minimizing the global cost function below

$$\min \sum_{k=1}^{N} J_{u,k}(\boldsymbol{w}) = \min \sum_{k=1}^{N} \left(\frac{\mathrm{E}|d_k(i) - \boldsymbol{u}_{k,i}^{\mathrm{T}}\boldsymbol{w}|^2}{\boldsymbol{u}_{k,i}^{\mathrm{T}}\boldsymbol{u}_{k,i}} \right), \tag{3}$$

where $J_{u,k}(\boldsymbol{w})$ denotes the cost function of node k with noise-free input and the global cost function is the sum of all local cost functions.

For node k, it degenerates into local optimization problem expressed as

$$\widehat{\boldsymbol{w}}_{k,i} = \arg\min_{\boldsymbol{w}} J_{u,k}(\boldsymbol{w}) = \arg\min_{\boldsymbol{w}} \left(\frac{\mathrm{E}|d_k(i) - \boldsymbol{u}_{k,i}^{\mathrm{T}}\boldsymbol{w}|^2}{\boldsymbol{u}_{k,i}^{\mathrm{T}}\boldsymbol{u}_{k,i}} \right). \tag{4}$$

Based on the assumptions made in the measurement model in equation (1) and (2), the optimal solution is obtained

$$\boldsymbol{w}^o = \boldsymbol{R}_{u,k}^{-1}\boldsymbol{r}_{du,k}, \tag{5}$$

with the covariance matrix $\boldsymbol{R}_{u,k} = \mathrm{E}[\boldsymbol{u}_{k,i}\boldsymbol{u}_{k,i}^{\mathrm{T}}]$ and the cross-covariance vector $\boldsymbol{r}_{du,k} = \mathrm{E}[d_k^{\mathrm{T}}(i)\boldsymbol{u}_{k,i}]$.

Observing the optimal solution in (5), the inverse operation of the covariance matrix increases computational complexity. Steepest descent method is developed to simplify the process, which is described as

$$\hat{\boldsymbol{w}}(n+1) = \hat{\boldsymbol{w}}(n) - \frac{\mu}{2}\nabla J(\hat{\boldsymbol{w}}(n)), \tag{6}$$

where non-negative parameter μ is the step-size, $\hat{\boldsymbol{w}}(n)$ denotes the estimation of unknown parameter $\boldsymbol{w}_o$ in the n-th iteration and $\nabla J(\hat{\boldsymbol{w}}(n))$ denotes the derivative of the cost function $J(\boldsymbol{w})$ with respect to $\hat{\boldsymbol{w}}(n)$. The update equation based on the steepest descent method can be rewritten as

$$\hat{\boldsymbol{w}}_{k,i} = \hat{\boldsymbol{w}}_{k,i-1} - \mu\frac{(-\boldsymbol{r}_{du,k} + \boldsymbol{R}_{u,k}\hat{\boldsymbol{w}}_{k,i-1})}{\boldsymbol{u}_{k,i}^{\mathrm{T}}\boldsymbol{u}_{k,i}}, \tag{7}$$

where $\hat{\boldsymbol{w}}_{k,i}$ denotes the weight estimate in the i-th iteration at node k. However, we cannot obtain the estimation through (7) due to the unknown statistical properties $\{\boldsymbol{R}_{u,k}, \boldsymbol{r}_{du,k}\}$ of signals. Stochastic gradient descent method has been developed to solve the problem by replacing the unknown second-order moments with instantaneous values that $\boldsymbol{R}_{u,k} \approx \boldsymbol{u}_{k,i}\boldsymbol{u}_{k,i}^{\mathrm{T}}$ and $\boldsymbol{r}_{du,k} \approx d_k(i)\boldsymbol{u}_{k,i}$. When we use $\widehat{\nabla J}_u(\hat{\boldsymbol{w}}_{k,i-1})$ to denote the approximate gradient vector, the traditional NLMS update equation is obtained

$$\begin{aligned}\hat{\boldsymbol{w}}_{k,i} &= \hat{\boldsymbol{w}}_{k,i-1} - \frac{\mu}{2}\widehat{\nabla J}_u(\hat{\boldsymbol{w}}_{k,i-1}) \\ &= \hat{\boldsymbol{w}}_{k,i-1} + \mu\frac{(d_k(i) - \boldsymbol{u}_{k,i}^{\mathrm{T}}\hat{\boldsymbol{w}}_{k,i-1})\boldsymbol{u}_{k,i}}{\boldsymbol{u}_{k,i}^{\mathrm{T}}\boldsymbol{u}_{k,i}}.\end{aligned} \tag{8}$$

The expression in (8) is only applicable for nodes in the absence of input noise. However, in most practical applications, both the input and output are corrupted by additive noise. Substituting $\boldsymbol{z}_{k,i}$ for $\boldsymbol{u}_{k,i}$ in (4), the optimal solution obtained by traditional NLMS algorithm will be biased and is given by

$$\boldsymbol{w}_b = \boldsymbol{R}_{z,k}^{-1}\boldsymbol{r}_{dz,k}, \tag{9}$$

where $\boldsymbol{w}_b$ denotes the biased estimate with the covariance matrix $\boldsymbol{R}_{z,k} = \mathrm{E}[\boldsymbol{z}_{k,i}\boldsymbol{z}_{k,i}^{\mathrm{T}}]$ and the cross-covariance matrix $\boldsymbol{r}_{dz,k} = \mathrm{E}[d_k^{\mathrm{T}}(i)\boldsymbol{z}_{k,i}]$.

We use the assumptions in data models (1) and (2) that the random variables are independent of each other to develop a new relationship

$$\begin{aligned}\boldsymbol{R}_{z,k} &= \mathrm{E}\left[\boldsymbol{z}_{k,i}\boldsymbol{z}_{k,i}^{\mathrm{T}}\right] = \boldsymbol{R}_{u,k} + \sigma_{n,k}^2\boldsymbol{I}, \\ \boldsymbol{r}_{dz,k} &= \mathrm{E}\left[d_k^{\mathrm{T}}(i)\boldsymbol{z}_{k,i}\right] = \boldsymbol{r}_{du,k},\end{aligned} \tag{10}$$

where $\boldsymbol{I}$ denotes an identity matrix with size of $M \times M$.

Substituting expressions in (10) back into (9), we can rewrite the optimal solution

$$\boldsymbol{w}_b = (\boldsymbol{R}_{u,k} + \sigma_{n,k}^2\boldsymbol{I})^{-1}\boldsymbol{r}_{du,k}. \tag{11}$$

Comparing with the optimal solution (5) in the absence of the input noise, the optimal solution above deviates from the unknown parameter and the bias is dependent on the input noise. To mitigate the effect of the input noise, a method is proposed by modifying the local cost function in (4) as

$$J_z(\boldsymbol{w}) = \frac{\mathrm{E}|d_k(i) - \boldsymbol{z}_{k,i}^{\mathrm{T}}\boldsymbol{w}|^2 - \sigma_{n,k}^2\|\boldsymbol{w}\|^2}{\boldsymbol{z}_{k,i}^{\mathrm{T}}\boldsymbol{z}_{k,i}}. \tag{12}$$

Here, we omit the subscript k and write the cost function as $J_z(\boldsymbol{w})$ since that each node has the same form of the cost function. Then the corresponding optimal solution can be obtained

$$\boldsymbol{w}_{\text{mod}} = (\boldsymbol{R}_{z,k} - \sigma_{n,k}^2 \boldsymbol{I})^{-1} \boldsymbol{r}_{du,k} = \boldsymbol{w}_o, \tag{13}$$

where $\boldsymbol{w}_{\text{mod}}$ denotes the estimate of unknown parameter obtained by the modified cost function. From (13), the noise-induced bias is removed and the unbiased estimation is obtained. Consequently, we use the stochastic gradient descent method to obtain the new update equation

$$\begin{aligned}\hat{\boldsymbol{w}}_{k,i} &= \hat{\boldsymbol{w}}_{k,i-1} - \frac{\mu}{2}\widehat{\nabla J_z}(\hat{\boldsymbol{w}}_{k,i-1}) \\ &= \hat{\boldsymbol{w}}_{k,i-1} + \mu \frac{e_k(i)\boldsymbol{z}_{k,i} + \sigma_{n,k}^2 \hat{\boldsymbol{w}}_{k,i-1}}{\boldsymbol{z}_{k,i}^{\mathrm{T}} \boldsymbol{z}_{k,i}},\end{aligned} \tag{14}$$

where $e_k(i) = d_k(i) - \boldsymbol{z}_{k,i}^{\mathrm{T}} \hat{\boldsymbol{w}}_{k,i-1}$ denotes the estimate error in the i-th iteration at node k.

In this section, we demonstrate that traditional NLMS method yields biased estimates under the situation that both the input and output are corrupted by additive noise. Our analysis shows that the bias is dependent to the input noise. We propose a method to remove the noise-induced bias by modifying the cost function, which gives unbiased estimation as shown in (13). Given that the input noise variance is known a priori, unbiased estimations can be obtained by running the update iteration in (14). Based on this assumption, we develop three BCNLMS methods according to the three different distributed strategies. However, knowledge of additive noise is usually unavailable, which motivates us to develop a technique of estimating it. Later on, a real-time estimation method for input noise variance is developed to overcome the unknown property of the noise.

3 Distributed BCNLMS algorithms

We present the non-cooperative bias-compensated NLMS algorithm for comparison, where each node $k \in 1, 2, \ldots, N$ independently runs its local iteration below to update the estimation results

$$\begin{aligned}e_k(i) &= d_k(i) - \boldsymbol{z}_{k,i}^{\mathrm{T}} \hat{\boldsymbol{w}}_{k,i-1}, \\ \hat{\boldsymbol{w}}_{k,i} &= \hat{\boldsymbol{w}}_{k,i-1} + \mu \frac{e_k(i)\boldsymbol{z}_{k,i} + \sigma_{n,k}^2 \hat{\boldsymbol{w}}_{k,i-1}}{\boldsymbol{z}_{k,i}^{\mathrm{T}} \boldsymbol{z}_{k,i}}.\end{aligned} \tag{15}$$

For the parameter estimation problem over multi-agent network, all nodes distributed in different locations collect measurement data to estimate a common parameter of interest. Considering that each node in the environment collects measurement data with different measurement error, several distributed cooperative strategies are introduced taking the advantages of its excellent performance and strong robustness. In this section, we first assume that a good estimate of input noise variance can be obtained, so that a series of BCNLMS algorithms based on different strategies are developed including incremental BCNLMS (Inc-BCNLMS), consensus BCNLMS (Con-BCNLMS) and ATC diffusion BCNLMS (ATC-BCNLMS) methods.

3.1 Incremental BCNLMS algorithm

For incremental strategy, the network topology of N nodes is shown in Figure 1. Initially, a cyclic path is determined to connect all nodes and renumber them from 1 to N. Information is transmitted from one agent to the next over the cyclic path until all nodes have been visited, and then the process is repeated. Each node in the network updates its estimate by running the local BCNLMS iteration below

$$\hat{\boldsymbol{w}}_{k,i} = \hat{\boldsymbol{w}}_{k-1,i} - \frac{\mu}{N}\widehat{\nabla J_z}(\hat{\boldsymbol{w}}_{k-1,i}). \tag{16}$$

At each iteration i, the initial value $\hat{\boldsymbol{w}}_{1,i}$ of the starting point 1 is updated by the final value of the last loop iteration $\hat{\boldsymbol{w}}_{N,i-1}$. Each node k over the cyclic path receives updated data $\hat{\boldsymbol{w}}_{k-1,i}$ from its previous

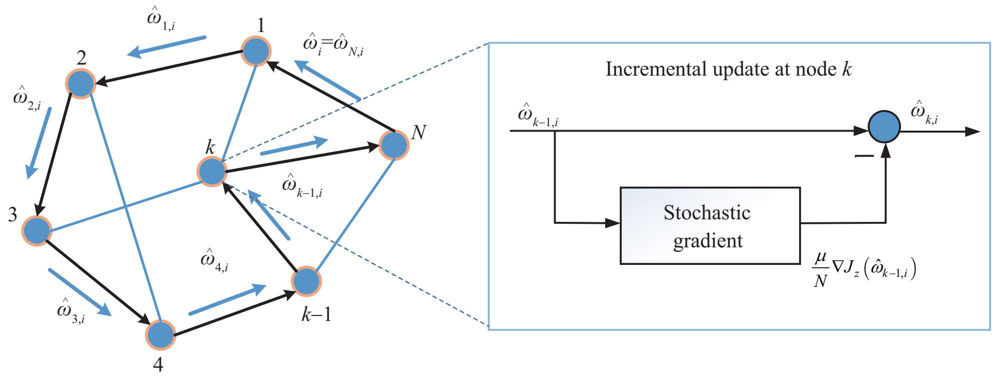

Figure 1 (Color online) Network topology for incremental strategy.

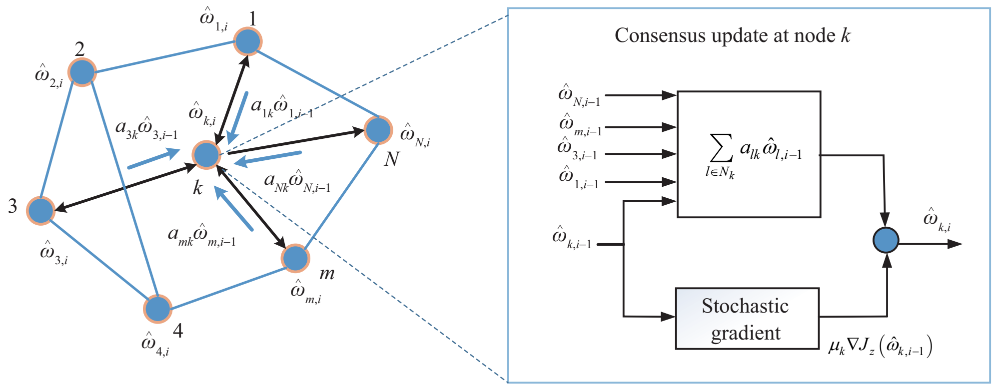

Figure 2 (Color online) Network topology for consensus strategy.

node to perform local iteration and conveys the updated data $\hat{\boldsymbol{w}}_{k,i}$ to its successor node until the last node of the trajectory. Then the process is repeated. The updated data $\hat{\boldsymbol{w}}_{N,i}$ is the estimate of the unknown parameter obtained in the i-th iteration.

For Inc-BCNLMS algorithm, each node k only exchanges information with its neighbors over the cyclic path, which significantly saves resources of communication consumption. Also, estimation performance is improved due to the multiple steps in updating estimation. However, it decreases the robustness of the network since any link failures over the path will result in collapse of the whole network. Besides, determining a cyclic path that covers all agents is an NP-hard problem. Furthermore, it is not suitable for real-time occasions and the collaboration among the in-network nodes is limited.

3.2 Consensus BCNLMS algorithm

The network topology of the consensus strategy is shown in Figure 2. Unlike the incremental strategy, there is no need to form a cyclic path or to renumber the nodes, each node cooperates with all its adjacent nodes to update the estimation by running the BCNLMS iterations below:

$$\begin{cases} \boldsymbol{\psi}_{k,i} = \displaystyle\sum_{l\in N_k} a_{lk}\hat{\boldsymbol{w}}_{l,i-1}, \\ \hat{\boldsymbol{w}}_{k,i} = \boldsymbol{\psi}_{k,i} - \mu\widehat{\nabla J}_z(\hat{\boldsymbol{w}}_{k,i-1}), \end{cases}$$

where parameter N_k denotes a collection of agents that are connected to agent k (agent k inclusive), $\boldsymbol{\psi}_{k,i}$ denotes the intermediate state, nonnegative combination coefficients $\{a_{lk}\}$ represent the weights that agent k assigns to its adjacent nodes, which we choose to satisfy

$$a_{lk} = 0 \quad \text{while} \quad l \notin N_k, \quad \text{and} \sum_{l\in N_k} a_{lk} = 1. \tag{17}$$

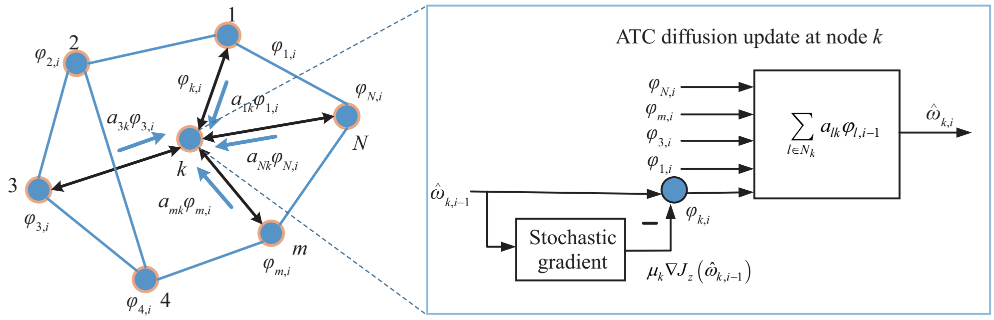

Figure 3 (Color online) Network topology for ATC diffusion strategy.

During each iteration i, the in-network node k first collects the last iteratively updated data $\hat{\boldsymbol{w}}_{l,i-1}$ from all its neighbor nodes l ($l \in N_k$) to form a convex combination $\boldsymbol{\psi}_{k,i}$, which we define as the intermediate state. The node k then combines $\boldsymbol{\psi}_{k,i}$ with its local data $\hat{\boldsymbol{w}}_{k,i-1}$ to update its estimate. Each node performs the bias compensation and update process simultaneously.

Comparing with incremental bias-compensated case, consensus BCNLMS algorithm enhances the cooperation among all network agents and improves the robustness of the network. Besides, it can be applied to a given network without need to renumber the in-network agents. However, it increases the cost of communication. Besides, the asymmetric update equations will lead to the problem of the unstable growth in the state of the network, which has been proved in [26].

3.3 Diffusion BCNLMS algorithm

Diffusion cooperation strategy estimates the unknown parameter through combination and adaptation steps, where all nodes in the network update their estimates simultaneously. Depending on the sequence of these two steps, diffusion strategy is divided into adapt-then-combine (ATC) diffusion strategy and combine-then-adapt (CTA) diffusion strategy. In this paper, we focus on the ATC diffusion strategy and the network topology is shown in Figure 3. Each node k runs the update BCNLMS process below

$$\begin{cases} \boldsymbol{\psi}_{k,i} = \hat{\boldsymbol{w}}_{k,i-1} - \mu \widehat{\nabla J}_z(\hat{\boldsymbol{w}}_{k,i-1}), \\ \hat{\boldsymbol{w}}_{k,i} = \sum\limits_{l \in N_k} a_{lk} \boldsymbol{\psi}_{l,i}. \end{cases}$$

For ATC diffusion strategy, each node first performs local adaptation to update its intermediate state $\boldsymbol{\psi}_{k,i}$, and then combines it with all its adjacent nodes to update estimations simultaneously. Note that the adaptation step of the ATC diffusion strategy adopts a symmetrical structure, which endows the diffusion strategy with ability to avoid unstable growth in state of the networks. As compared to the incremental strategy, diffusion BCNLMS algorithm enhances cooperation among all agents and it can be applied to any given network. However, the information exchange among the nodes consumes more communication resources.

Three distributed BCNLMS are developed based on the assumption that the input noise variance is known a priori. However, in many practical applications, the priori knowledge of the input noise variance is usually unavailable, which motivates us to develop a way to solve the unknown input noise variance. In Subsection 3.4, we propose an estimation method for input noise variance.

3.4 Noise variance estimation

We first introduce an auxiliary parameter $\boldsymbol{\beta} \in \mathbb{C}^{M\times 1}$, which runs the update iteration

$$\boldsymbol{\beta}_{k,i} = \boldsymbol{\beta}_{k,i-1} + \frac{\mu \boldsymbol{z}_{k,i}(d_k(i-1) - \boldsymbol{z}_{k,i}^{\mathrm{T}} \boldsymbol{\beta}_{k,i-1})}{\boldsymbol{z}_{k,i}^{\mathrm{T}} \boldsymbol{z}_{k,i}}, \tag{18}$$

where $\boldsymbol{\beta}_{k,i}$ utilizes the one-step backward output $d_k(i-1)$ from the previous moment to update. Thus, by using the parameters $\boldsymbol{\beta}_{k,i}$ and $\boldsymbol{z}_{k,i}$ we construct the one-step backward output predictor

$$d_k\left[i-1 \mid \boldsymbol{z}_{k,i}\right] = \boldsymbol{z}_{k,i}^{\mathrm{T}}\boldsymbol{\beta}_{k,i}. \tag{19}$$

The backward prediction error (PE) $\varsigma_k(i)$ is

$$\varsigma_k(i) = d_k(i-1) - \boldsymbol{z}_{k,i}^{\mathrm{T}}\boldsymbol{\beta}_{k,i}. \tag{20}$$

According to the principle of orthogonality, it is deduced that input and backward prediction error are uncorrelated

$$\mathrm{E}[\varsigma_k(i)\boldsymbol{z}_{k,i}] = 0. \tag{21}$$

And the cross-correlation function between the output and prediction error is defined as

$$f_k(i) = \mathrm{E}\left[d_k(i)(d_k(i-1) - \boldsymbol{z}_{k,i}^{\mathrm{T}}\boldsymbol{\beta}_{k,i})\right]. \tag{22}$$

Using the measurement model in (1) and (2) and principle of orthogonality in (21), the cross-correlation function could be further simplified as

$$\begin{aligned} f_k(i) &= \mathrm{E}[(v_k(i) - \boldsymbol{n}_{k,i}^{\mathrm{T}}\boldsymbol{w}_o)(\boldsymbol{u}_{k,i-1}^{\mathrm{T}}\boldsymbol{w}_o + v_k(i-1) - (\boldsymbol{u}_{k,i} + \boldsymbol{n}_{k,i})^{\mathrm{T}}\boldsymbol{\beta}_{k,i})] \\ &= \sigma_{n,k}^2 (\boldsymbol{w}_o)^{\mathrm{T}}\boldsymbol{\beta}_{k,i}. \end{aligned} \tag{23}$$

We use the iteration below to calculate the unknown parameter $f_k(i)$ and substitute $\hat{\boldsymbol{w}}_{k,i-1}$ for $\boldsymbol{w}_o$ since that the parameter of interest is unknown. The input noise variance can be estimated as

$$f_k(i) = \alpha f_k(i-1) + (1-\alpha)d_k(i)\varsigma_k(i), \tag{24}$$

$$\hat{\sigma}_{n,k}^2 = \frac{f_k(i)}{\hat{\boldsymbol{w}}_{k,i-1}^{\mathrm{T}}\boldsymbol{\beta}_{k,i}}, \tag{25}$$

where α denotes the smoothing factor, in the interval $0 \ll \alpha \leqslant 1$ and $\hat{\sigma}_{n,k}^2$ denotes the estimate variance of the input noise.

Considering the fact that inter-node collaboration significantly improves the estimation performance, we therefore implement this proposed noise variance estimation method in the distributed strategies. Using this real-time estimation method to identify the unknown input noise variance, there is robust applicability in several practical scenarios where the noise is usually unknown. We summarize three distributed BCNLMS algorithms in Subsection 3.5.

3.5 Summary of the three distributed-based BCNLMS algorithms

In this subsection, we summarize the proposed three distributed BCNLMS algorithms. For incremental BCNLMS method, each node uses the updated weight of its previous node to estimate the input noise variance, which is an approximation of the true value. However, the multiple steps in updating endows the Inc-BCNLMS algorithm (Algorithm 1) with good estimation performance. For consensus BCNLMS method (Algorithm 2), we first perform the aggregation step to form an intermediate state. The intermediate state is then used to estimate the input noise variance. For ATC diffusion BCNLMS method (Algorithm 3), we use the last local updated weights to estimate the input noise variance.

4 Performance analysis

In this section, we analyze the BCNLMS algorithms based on non-cooperative strategy, consensus strategy and ATC diffusion strategy due to space limitations. Instead of analyzing each algorithm separately, we

Algorithm 1 Incremental BCNLMS algorithm

For node $k \in 1, 2, \ldots, N$ at each iteration $i \geqslant 0$:

Step1. Set the initial value:

$$\boldsymbol{\beta}_{k,0} \leftarrow 0,\ \hat{\boldsymbol{w}}_{k,0} \leftarrow 0,\ f_k(0) \leftarrow 0;$$

Step2. Update the noise variance estimation:

$$\boldsymbol{\beta}_{k,i} = \boldsymbol{\beta}_{k-1,i} + \frac{\mu}{N}\frac{\boldsymbol{z}_{k,i}(d_k(i-1) - \boldsymbol{z}_{k,i}^{\mathrm{T}}\boldsymbol{\beta}_{k-1,i})}{\boldsymbol{z}_{k,i}^{\mathrm{T}}\boldsymbol{z}_{k,i}},$$

$$\varsigma_k(i) = d_k(i-1) - \boldsymbol{z}_{k,i}^{\mathrm{T}}\boldsymbol{\beta}_{k,i},$$

$$f_k(i) = \alpha f_k(i-1) + (1-\alpha)d_k(i)\varsigma_k(i),$$

$$\sigma_{n,k}^2 = \frac{f_k(i)}{\hat{\boldsymbol{w}}_{k-1,i}^{\mathrm{T}}\boldsymbol{\beta}_{k,i}};$$

Step3. Run bias compensation procedure:

$$\hat{\boldsymbol{w}}_{k,i} = \hat{\boldsymbol{w}}_{k-1,i} + \frac{\mu}{N}\frac{\boldsymbol{z}_{k,i}[d_k(i) - \boldsymbol{z}_{k,i}^{\mathrm{T}}\hat{\boldsymbol{w}}_{k-1,i}] + \sigma_{n,k}^2\hat{\boldsymbol{w}}_{k-1,i}}{\boldsymbol{z}_{k,i}^{\mathrm{T}}\boldsymbol{z}_{k,i}};$$

Step4. Transmit the estimator to the next iteration:

$$\boldsymbol{\beta}_{0,i+1} \leftarrow \boldsymbol{\beta}_{N,i},\ \hat{\boldsymbol{w}}_{0,i+1} \leftarrow \hat{\boldsymbol{w}}_{N,i},\ f_0(i+1) \leftarrow f_N(i);$$

Repeat **Step2–4** to update the estimation results until convergence.

Algorithm 2 Consensus BCNLMS algorithm

For each iteration $i \geqslant 0$:

Step1. Set the initial value:

$$\boldsymbol{\beta}_{k,0} \leftarrow 0,\ \hat{\boldsymbol{w}}_{k,0} \leftarrow 0,\ f_k(0) \leftarrow 0;$$

Step2. Aggregation step for node k

$$\boldsymbol{\psi}_{k,i} = \sum_{l \in N_k} a_{lk}\hat{\boldsymbol{w}}_{l,i-1};$$

Step3. Update the estimators of the noise variance

$$\boldsymbol{\beta}_{k,i} = \boldsymbol{\beta}_{k,i-1} + \mu\frac{\boldsymbol{z}_{k,i}(d_k(i-1) - \boldsymbol{z}_{k,i}^{\mathrm{T}}\boldsymbol{\beta}_{k,i-1})}{\boldsymbol{z}_{k,i}^{\mathrm{T}}\boldsymbol{z}_{k,i}},$$

$$\varsigma_k(i) = d_k(i-1) - \boldsymbol{z}_{k,i}^{\mathrm{T}}\boldsymbol{\beta}_{k,i},$$

$$f_k(i) = \alpha f_k(i-1) + (1-\alpha)d_k(i)\varsigma_k(i),$$

$$\sigma_{n,k}^2 = \frac{f_k(i)}{\boldsymbol{\psi}_{k,i}^{\mathrm{T}}\boldsymbol{\beta}_{k,i}};$$

Step4. Bias-compensated update procedure for node k

$$\hat{\boldsymbol{w}}_{k,i} = \boldsymbol{\psi}_{k,i} + \mu\frac{\boldsymbol{z}_{k,i}[d_k(i) - \boldsymbol{z}_{k,i}^{\mathrm{T}}\hat{\boldsymbol{w}}_{k,i-1}] + \sigma_{n,k}^2\hat{\boldsymbol{w}}_{k,i-1}}{\boldsymbol{z}_{k,i}^{\mathrm{T}}\boldsymbol{z}_{k,i}};$$

Repeat from **Step2–4** to update the estimation results until convergence.

Algorithm 3 ATC diffusion BCNLMS algorithm

For each iteration $i \geqslant 0$:

Step1. Set the initial value:

$$\boldsymbol{\beta}_{k,0} \leftarrow 0,\ \hat{\boldsymbol{w}}_{k,0} \leftarrow 0,\ f_k(0) \leftarrow 0;$$

Step2. Update the estimators for noise variance

$$\boldsymbol{\beta}_{k,i} = \boldsymbol{\beta}_{k,i-1} + \mu\frac{\boldsymbol{z}_{k,i}(d_k(i-1) - \boldsymbol{z}_{k,i}^{\mathrm{T}}\boldsymbol{\beta}_{k,i-1})}{\boldsymbol{z}_{k,i}^{\mathrm{T}}\boldsymbol{z}_{k,i}},$$

$$\varsigma_k(i) = d_k(i-1) - \boldsymbol{z}_{k,i}^{\mathrm{T}}\boldsymbol{\beta}_{k,i},$$

$$f_k(i) = \alpha f_k(i-1) + (1-\alpha)d_k(i)\varsigma_k(i),$$

$$\sigma_{n,k}^2 = \frac{f_k(i)}{\hat{\boldsymbol{w}}_{k,i-1}^{\mathrm{T}}\boldsymbol{\beta}_{k,i}};$$

Step3. Bias-compensated update procedure for node k

$$\boldsymbol{\psi}_{k,i} = \hat{\boldsymbol{w}}_{k,i-1} + \mu\frac{\boldsymbol{z}_{k,i}[d_k(i) - \boldsymbol{z}_{k,i}^{\mathrm{T}}\hat{\boldsymbol{w}}_{k,i-1}] + \sigma_{n,k}^2\hat{\boldsymbol{w}}_{k,i-1}}{\boldsymbol{z}_{k,i}^{\mathrm{T}}\boldsymbol{z}_{k,i}};$$

Step4. Aggregation step for node k

$$\hat{\boldsymbol{w}}_{k,i} = \sum_{l \in N_k} a_{lk}\boldsymbol{\psi}_{l,i};$$

Repeat from **Step2–4** to update the estimation results until convergence.

Table 1 Different BCNLMS algorithms with different choices of matrices $\boldsymbol{A}_0$, $\boldsymbol{A}_1$, $\boldsymbol{A}_2$

Algorithm	$\boldsymbol{A}_0$	$\boldsymbol{A}_1$	$\boldsymbol{A}_2$
Non-cooperative BCNLMS	$\boldsymbol{I}$	$\boldsymbol{I}$	$\boldsymbol{I}$
Consensus BCNLMS	$\boldsymbol{A}$	$\boldsymbol{I}$	$\boldsymbol{I}$
ATC BCNLMS	$\boldsymbol{I}$	$\boldsymbol{I}$	$\boldsymbol{A}$

formulate a general algorithmic form for various strategies, which can be expressed as

$$\begin{cases} \boldsymbol{\phi}_{k,i-1} = \sum_{l\in N_k} a_{1,lk}\boldsymbol{w}_{l,i-1}, \\ \boldsymbol{\psi}_{k,i} = \sum_{l\in N_k} a_{0,lk}\boldsymbol{\phi}_{l,i-1} + \mu_k \dfrac{\boldsymbol{z}_{k,i}[d_k(i) - \boldsymbol{z}_{k,i}^{\mathrm{T}}\boldsymbol{\phi}_{k,i-1}] + \sigma_{n,k}^2\boldsymbol{\phi}_{k,i-1}}{\boldsymbol{z}_{k,i}^{\mathrm{T}}\boldsymbol{z}_{k,i}}, \\ \boldsymbol{w}_{k,i} = \sum_{l\in N_k} a_{2,lk}\boldsymbol{\psi}_{l,i}, \end{cases}$$

where the non-negative coefficients $\{a_{0,lk}, a_{1,lk}, a_{2,lk}\}$ denote the $\{l,k\}$ entries of left-stochastic matrices $\boldsymbol{A}_0$, $\boldsymbol{A}_1$, $\boldsymbol{A}_2$, respectively. Different choices for matrices $\boldsymbol{A}_0$, $\boldsymbol{A}_1$, $\boldsymbol{A}_2$ correspond to different operation modes as shown in Table 1, where $\boldsymbol{I}$ denotes the identity matrix.

We define the local weight-error vectors of each node k as

$$\tilde{\boldsymbol{\phi}}_{k,i} = \boldsymbol{w}_o - \boldsymbol{\phi}_{k,i},\ \tilde{\boldsymbol{\psi}}_{k,i} = \boldsymbol{w}_o - \boldsymbol{\psi}_{k,i},\ \tilde{\boldsymbol{w}}_{k,i} = \boldsymbol{w}_o - \boldsymbol{w}_{k,i}. \tag{26}$$

For a multi-agent network, the global weight-error vectors are formed by stacking the local error vectors

$$\begin{aligned} \tilde{\boldsymbol{\phi}}_i &= \mathrm{col}\left\{\tilde{\boldsymbol{\phi}}_{1,i}, \dots, \tilde{\boldsymbol{\phi}}_{N,i}\right\}, \\ \tilde{\boldsymbol{\psi}}_i &= \mathrm{col}\left\{\tilde{\boldsymbol{\psi}}_{1,i}, \dots, \tilde{\boldsymbol{\psi}}_{N,i}\right\}, \\ \tilde{\boldsymbol{w}}_i &= \mathrm{col}\{\tilde{\boldsymbol{w}}_{1,i}, \dots, \tilde{\boldsymbol{w}}_{N,i}\}. \end{aligned}$$

We introduce the diagonal matrix

$$\mathcal{M} = \mathrm{diag}\{\mu_1\boldsymbol{I}_M, \dots, \mu_N\boldsymbol{I}_M\},$$

and some extended weighting matrices

$$\mathcal{A}_0 = \boldsymbol{A}_0 \otimes \boldsymbol{I}_M, \quad \mathcal{A}_1 = \boldsymbol{A}_1 \otimes \boldsymbol{I}_M, \quad \mathcal{A}_2 = \boldsymbol{A}_2 \otimes \boldsymbol{I}_M,$$

where $\otimes$ denotes the Kronecker product operation, $\boldsymbol{I}_M$ is the identity matrix of size $M \times M$.

We further introduce the following matrices

$$\begin{aligned} &\mathcal{G}_i = \mathrm{col}\left\{\frac{\boldsymbol{z}_{1,i}v_{1,i}}{\boldsymbol{z}_{1,i}^{\mathrm{T}}\boldsymbol{z}_{1,i}}, \dots, \frac{\boldsymbol{z}_{N,i}v_{1,i}}{\boldsymbol{z}_{N,i}^{\mathrm{T}}\boldsymbol{z}_{N,i}}\right\}, \quad \mathcal{D}_i = \mathrm{diag}\left\{\frac{\boldsymbol{z}_{1,i}\boldsymbol{n}_{1,i}^{\mathrm{T}} - \sigma_{n,1}^2\boldsymbol{I}_M}{\boldsymbol{z}_{1,i}^{\mathrm{T}}\boldsymbol{z}_{1,i}}, \dots, \frac{\boldsymbol{z}_{N,i}\boldsymbol{n}_{N,i}^{\mathrm{T}} - \sigma_{n,N}^2\boldsymbol{I}_M}{\boldsymbol{z}_{N,i}^{\mathrm{T}}\boldsymbol{z}_{N,i}}\right\}, \\ &\mathcal{H}_i = \mathrm{diag}\left\{\frac{\boldsymbol{z}_{1,i}\boldsymbol{z}_{1,i}^{\mathrm{T}} - \sigma_{n,1}^2\boldsymbol{I}_M}{\boldsymbol{z}_{1,i}^{\mathrm{T}}\boldsymbol{z}_{1,i}}, \dots, \frac{\boldsymbol{z}_{N,i}\boldsymbol{z}_{N,i}^{\mathrm{T}} - \sigma_{n,N}^2\boldsymbol{I}_M}{\boldsymbol{z}_{N,i}^{\mathrm{T}}\boldsymbol{z}_{N,i}}\right\}, \quad \boldsymbol{W}_o = \mathrm{col}\underbrace{\{\boldsymbol{w}_o, \dots, \boldsymbol{w}_o\}}_{N}. \end{aligned}$$

Using the block network variables, we obtain the recursions that

$$\begin{cases} \tilde{\boldsymbol{\phi}}_{i-1} = \mathcal{A}_1^{\mathrm{T}}\tilde{\boldsymbol{w}}_{i-1}, \\ \tilde{\boldsymbol{\psi}}_i = \mathcal{A}_0^{\mathrm{T}}\tilde{\boldsymbol{\phi}}_{i-1} - \mathcal{M}[\mathcal{G}_i - \mathcal{D}_i\boldsymbol{W}_o + \mathcal{H}_i\tilde{\boldsymbol{\phi}}_{i-1}], \\ \tilde{\boldsymbol{w}}_k = \mathcal{A}_2^{\mathrm{T}}\tilde{\boldsymbol{\psi}}_i, \end{cases}$$

so that the network error vector $\tilde{\boldsymbol{w}}_i$ evolves with time according to the recursion

$$\begin{aligned} \tilde{\boldsymbol{w}}_i &= \mathcal{A}_2^{\mathrm{T}}\left(\mathcal{A}_0^{\mathrm{T}} - \mathcal{M}\mathcal{H}_i\right)\mathcal{A}_1^{\mathrm{T}}\tilde{\boldsymbol{w}}_{i-1} - \mathcal{A}_2^{\mathrm{T}}\mathcal{M}\mathcal{G}_i + \mathcal{A}_2^{\mathrm{T}}\mathcal{M}\mathcal{D}_i\boldsymbol{W}_o \\ &= \mathcal{B}_i\tilde{\boldsymbol{w}}_{i-1} - \mathcal{A}_2^{\mathrm{T}}\mathcal{M}\mathcal{G}_i + \mathcal{A}_2^{\mathrm{T}}\mathcal{M}\mathcal{D}_i\boldsymbol{W}_o, \end{aligned} \tag{27}$$

where the time-varying matrix $\mathcal{B}_i = \mathcal{A}_2^{\mathrm{T}}(\mathcal{A}_0^{\mathrm{T}} - \mathcal{M}\mathcal{H}_i)\mathcal{A}_1^{\mathrm{T}}$ is independent of $\tilde{\boldsymbol{w}}_i$.

4.1 Mean stability

Taking expectations of both sides of (27), we get

$$\mathrm{E}\left[\tilde{\boldsymbol{w}}_i\right] = \mathcal{B}\left(\mathrm{E}\left[\tilde{\boldsymbol{w}}_{i-1}\right]\right), \tag{28}$$

where $\mathrm{E}[\mathcal{G}_i] = 0$, $\mathrm{E}[\mathcal{D}_i] = 0$ due to the relations among inputs and outputs we assumed in Assumption 1. According to equation (28), $\lim_{i\to\infty} \mathrm{E}\|\tilde{\boldsymbol{w}}_i\|_\sigma^2 \to 0$ if $\mathcal{B}$ is stable.

For various operation modes, the item $\mathcal{B}$ follows different forms:

$$\mathcal{B} = \mathrm{E}\left[\mathcal{B}_i\right] = \begin{cases} \boldsymbol{I} - \mathcal{M}\mathcal{H}, & \text{Non-cooperative strategy,} \\ \mathcal{A}^{\mathrm{T}} - \mathcal{M}\mathcal{H}, & \text{Consensus strategy,} \\ \mathcal{A}^{\mathrm{T}}\left(\boldsymbol{I} - \mathcal{M}\mathcal{H}\right), & \text{ATC diffusion strategy,} \end{cases}$$

where

$$\mathcal{H} = \mathrm{E}\left[\mathcal{H}_i\right] = \mathrm{diag}\left\{\mathrm{E}\left(\frac{\boldsymbol{z}_{1,i}\boldsymbol{z}_{1,i}^{\mathrm{T}} - \sigma_{n,1}^2\boldsymbol{I}_M}{\boldsymbol{z}_{1,i}^{\mathrm{T}}\boldsymbol{z}_{1,i}}\right), \ldots, \mathrm{E}\left(\frac{\boldsymbol{z}_{N,i}\boldsymbol{z}_{N,i}^{\mathrm{T}} - \sigma_{n,N}^2\boldsymbol{I}_M}{\boldsymbol{z}_{N,i}^{\mathrm{T}}\boldsymbol{z}_{N,i}}\right)\right\}.$$

Theorem 1 (Mean error stability condition)**.** Consider an adaptive network operates using various strategies with the data model in (1) and (2). The mean error vector evolves with time according to (28). For consensus strategy, it can be unstable since that the mean stability condition depends on the specific combination matrices $\boldsymbol{A}$. Non-cooperative strategy and ATC diffusion strategy have the same mean stability condition since that $\rho(\boldsymbol{A}) = 1$ for any left-stochastic matrix, and the whole network will be asymptotically unbiased and stable if the step-size satisfies

$$0 < \mu < \frac{2}{\lambda_{\max}(\mathcal{H})}, \tag{29}$$

where $\lambda_{\max}(\mathcal{H})$ equals to the maximum eigenvalue of matrix $\mathcal{H}$.

4.2 Mean-square performance of the network

Introduce an $N \times N$ block Hermitian nonnegative-definite matrix Σ that we are free to choose, with $M \times M$ block entries. Equating the weighted square norms on both sides of (27) and taking expectations, we obtain

$$\mathrm{E}\|\tilde{\boldsymbol{w}}_i\|_\Sigma^2 = \mathrm{E}\|\tilde{\boldsymbol{w}}_{i-1}\|_{\Sigma'}^2 + \mathrm{E}\left[\mathcal{G}_i^{\mathrm{T}}\mathcal{M}\mathcal{A}_2\Sigma\mathcal{A}_2^{\mathrm{T}}\mathcal{M}\mathcal{G}_i\right] + \mathrm{E}\left[\boldsymbol{W}_o^{\mathrm{T}}\mathcal{D}_i^{\mathrm{T}}\mathcal{M}\mathcal{A}_2\Sigma\mathcal{A}_2^{\mathrm{T}}\mathcal{M}\mathcal{D}_i\boldsymbol{W}_o\right], \tag{30}$$

where

$$\begin{aligned} \Sigma' &= \mathrm{E}\left[\mathcal{B}_i^{\mathrm{T}}\Sigma\mathcal{B}_i\right] \\ &= \mathrm{E}\left[\mathcal{A}_1\left(\mathcal{A}_0 - \mathcal{H}_i^{\mathrm{T}}\mathcal{M}\right)\mathcal{A}_2\Sigma\mathcal{A}_2^{\mathrm{T}}\left(\mathcal{A}_0^{\mathrm{T}} - \mathcal{M}\mathcal{H}_i\right)\mathcal{A}_1^{\mathrm{T}}\right] \\ &= \mathcal{A}_1\mathcal{A}_0\mathcal{A}_2\Sigma\mathcal{A}_2^{\mathrm{T}}\mathcal{A}_0^{\mathrm{T}}\mathcal{A}_1^{\mathrm{T}} - \mathcal{A}_1\mathcal{H}^{\mathrm{T}}\mathcal{M}\mathcal{A}_2\Sigma\mathcal{A}_2\mathcal{A}_0^{\mathrm{T}}\mathcal{A}_1^{\mathrm{T}} \\ &\quad - \mathcal{A}_1\mathcal{A}_0\mathcal{A}_2\Sigma\mathcal{A}_2^{\mathrm{T}}\mathcal{M}\mathcal{H}\mathcal{A}_1^{\mathrm{T}} + \mathrm{E}\left[\mathcal{A}_1\mathcal{H}_i^{\mathrm{T}}\mathcal{M}\mathcal{A}_2\Sigma\mathcal{A}_2^{\mathrm{T}}\mathcal{M}\mathcal{H}_i\mathcal{A}_1^{\mathrm{T}}\right]. \end{aligned} \tag{31}$$

Let

$$\mathcal{G} = \mathrm{E}[\mathcal{G}_i\mathcal{G}_i^{\mathrm{T}}], \quad \sigma = \mathrm{vec}(\Sigma), \tag{32}$$

where $\mathrm{vec}(\cdot)$ denotes the operation that stacks the columnar entries of its matrix on the top of each other. For Kronecker product operation, it holds that $\mathrm{vec}(U\Sigma V) = (V^{\mathrm{T}} \otimes U)\mathrm{vec}(\Sigma)$. We vectorize both sides of (31) and arrive at

$$\mathrm{vec}(\Sigma') = \mathcal{F}\sigma, \tag{33}$$

where $\mathcal{F} = \{(\mathcal{A}_1 \otimes \mathcal{A}_1)[\mathcal{A}_0 \otimes \mathcal{A}_0 - \mathcal{A}_0 \otimes \mathcal{H}^{\mathrm{T}}\mathcal{M} - \mathcal{H}^{\mathrm{T}}\mathcal{M} \otimes \mathcal{A}_0 + \mathrm{E}[\mathcal{H}_i^{\mathrm{T}}\mathcal{M} \otimes \mathcal{H}_i^{\mathrm{T}}\mathcal{M}]](\mathcal{A}_2 \otimes \mathcal{A}_2)\}$.

Using the property that $\mathrm{Tr}(\Sigma\boldsymbol{X}) = \mathrm{vec}(\boldsymbol{X}^{\mathrm{T}})^{\mathrm{T}}\mathrm{vec}(\Sigma)$, we rewrite (30) as

$$\mathrm{E}\|\tilde{\boldsymbol{w}}_i\|^2_{\sigma} = \mathrm{E}\|\tilde{\boldsymbol{w}}_{i-1}\|^2_{\mathcal{F}\sigma} + \left[\mathrm{vec}(\mathcal{A}_2^{\mathrm{T}}\mathcal{M}\mathcal{G}\mathcal{M}\mathcal{A}_2) + \mathrm{vec}(\mathcal{A}_2^{\mathrm{T}}\mathcal{M}\mathcal{D}_i\boldsymbol{W}_o\boldsymbol{W}_o^{\mathrm{T}}\mathcal{D}_i^{\mathrm{T}}\mathcal{M}\mathcal{A}_2)\right]^{\mathrm{T}}\sigma. \tag{34}$$

Because the dynamics is mean-square stable for sufficiently small step-sizes, we take the limit of (34) as $i \to \infty$ and write

$$\lim_{i\to\infty}\mathrm{E}\|\tilde{\boldsymbol{w}}_i\|^2_{\sigma} = \lim_{i\to\infty}\mathrm{E}\|\tilde{\boldsymbol{w}}_{i-1}\|^2_{\mathcal{F}\sigma} + \left[\mathrm{vec}(\mathcal{A}_2^{\mathrm{T}}\mathcal{M}\mathcal{G}\mathcal{M}\mathcal{A}_2) + \mathrm{vec}(\mathcal{A}_2^{\mathrm{T}}\mathcal{M}\mathcal{D}_i\boldsymbol{W}_o\boldsymbol{W}_o^{\mathrm{T}}\mathcal{D}_i^{\mathrm{T}}\mathcal{M}\mathcal{A}_2)\right]^{\mathrm{T}}\sigma. \tag{35}$$

This yields the corresponding steady-state variance relation that

$$\lim_{i\to\infty}\mathrm{E}\|\tilde{\boldsymbol{w}}_i\|^2_{(\boldsymbol{I}-\mathcal{F})\sigma} = \left[\mathrm{vec}(\mathcal{A}_2^{\mathrm{T}}\mathcal{M}\mathcal{G}\mathcal{M}\mathcal{A}_2) + \mathrm{vec}(\mathcal{A}_2^{\mathrm{T}}\mathcal{M}\mathcal{D}_i\boldsymbol{W}_o\boldsymbol{W}_o^{\mathrm{T}}\mathcal{D}_i^{\mathrm{T}}\mathcal{M}\mathcal{A}_2)\right]^{\mathrm{T}}\sigma. \tag{36}$$

We use the mean-square deviation (MSD) to quantify the performance of BCNLMS algorithms over multi-agent network. The network MSD is defined as the average MSD_s value of the individual nodes

$$\mathrm{MSD}^{\mathrm{network}} \triangleq \lim_{i\to\infty}\frac{1}{N}\sum_{k=1}^{N}\mathrm{E}\|\tilde{\boldsymbol{w}}_{k,i}\|^2 \triangleq \lim_{i\to\infty}\frac{1}{N}\mathrm{E}\|\tilde{\boldsymbol{w}}_i\|^2. \tag{37}$$

We select the weighting vector σ to recover the network MSD from relation (36)

$$(\boldsymbol{I}-\mathcal{F})\sigma = \frac{1}{N}\mathrm{vec}(\boldsymbol{I}_{NM}). \tag{38}$$

Then the network MSD is obtained

$$\mathrm{MSD}^{\mathrm{network}} = \frac{1}{N}\left\{\left[\mathrm{vec}(\mathcal{A}_2^{\mathrm{T}}\mathcal{M}\mathcal{G}\mathcal{M}\mathcal{A}_2) + \mathrm{vec}(\mathcal{A}_2^{\mathrm{T}}\mathcal{M}\mathcal{D}_i\boldsymbol{W}_o\boldsymbol{W}_o^{\mathrm{T}}\mathcal{D}_i^{\mathrm{T}}\mathcal{M}\mathcal{A}_2)\right]^{\mathrm{T}}\right\}(\boldsymbol{I}-\mathcal{F})^{-1}\mathrm{vec}(\boldsymbol{I}_{NM}). \tag{39}$$

Theorem 2 (Mean-square error performance)**.** Consider an adaptive network operates using various strategies with the data model in (1) and (2) that satisfy Assumption 1. Then the transient and steady-state mean-square error performances of the network follow recursions

$$\mathrm{E}\|\tilde{\boldsymbol{w}}_i\|^2_{\sigma} = \mathrm{E}\|\tilde{\boldsymbol{w}}_{i-1}\|^2_{\mathcal{F}\sigma} + \left[\mathrm{vec}(\mathcal{A}_2^{\mathrm{T}}\mathcal{M}\mathcal{G}\mathcal{M}\mathcal{A}_2) + \mathrm{vec}(\mathcal{A}_2^{\mathrm{T}}\mathcal{M}\mathcal{D}_i\boldsymbol{W}_o\boldsymbol{W}_o^{\mathrm{T}}\mathcal{D}_i^{\mathrm{T}}\mathcal{M}\mathcal{A}_2)\right]^{\mathrm{T}}\sigma, \tag{40}$$

$$\mathrm{MSD}^{\mathrm{network}} = \frac{1}{N}\left\{\left[\mathrm{vec}(\mathcal{A}_2^{\mathrm{T}}\mathcal{M}\mathcal{G}\mathcal{M}\mathcal{A}_2) + \mathrm{vec}(\mathcal{A}_2^{\mathrm{T}}\mathcal{M}\mathcal{D}_i\boldsymbol{W}_o\boldsymbol{W}_o^{\mathrm{T}}\mathcal{D}_i^{\mathrm{T}}\mathcal{M}\mathcal{A}_2)\right]^{\mathrm{T}}\right\}(\boldsymbol{I}-\mathcal{F})^{-1}\mathrm{vec}(\boldsymbol{I}_{NM}), \tag{41}$$

where the network based on non-cooperative strategy or ATC diffusion strategy will be stable if the step-size satisfies (29), and they will converge in the mean and mean-square performance.

5 Simulation results

In order to illustrate the effectiveness of our proposed distributed BCNLMS algorithms, we present some simulations. Figure 4 is a network topology with $N = 20$ nodes distributed in different locations over space, and the objective to be estimated is generated randomly with order $M = 20$. We set the step-size $\mu = 0.2$ and the smoothing factor $\alpha = 0.998$. The noise-free input data $\boldsymbol{u}_{k,i}$ and the additive noise $\boldsymbol{n}_{k,i}$, $v_k(i)$ are all Gaussian distributions. The network signal and noise power are shown in Figure 5. Our results are the average of running 200 times independently.

In this paper, the averaging rule is applied in consensus strategy and ATC diffusion strategy with

$$a_{lk} = \begin{cases} 1/\deg(k), & l \in N_k, \\ 0, & \text{otherwise}, \end{cases}$$

where $\deg(k)$ represents the degree of node k, which equates to the nodes number connected to node k (including k itself). We introduce MSD as the criteria for assessing the accuracy of the estimation results. For multi-agent networks, the global MSD is the average value of all nodes as follows:

$$\mathrm{MSD} = \lim_{i\to\infty}\frac{1}{N}\sum_{k=1}^{N}\|\hat{\boldsymbol{w}}_{k,i} - \boldsymbol{w}_o\|^2. \tag{42}$$

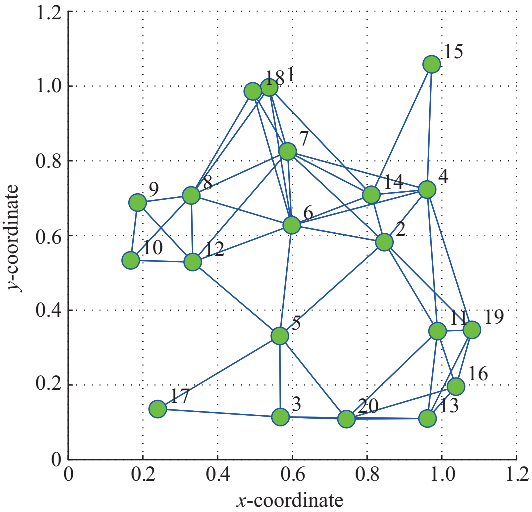

Figure 4 (Color online) Topology of the network.

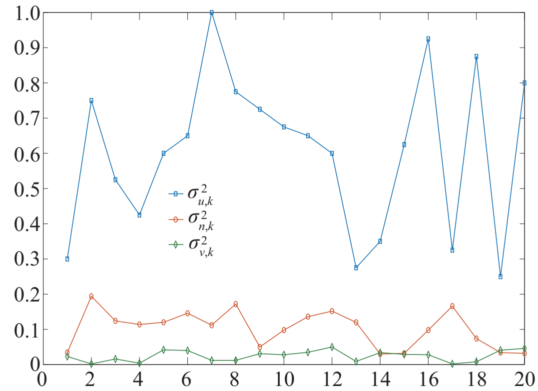

Figure 5 (Color online) The network signal and noise power levels.

We first assume that the input noise variance is known a priori and compare the estimation performance of traditional non-cooperative NLMS (Nco-NLMS) algorithm, the non-cooperative BCNLMS (Nco-BCNLMS) method and three distributed BCNLMS algorithms as shown in Figure 6. Under non-cooperative situations, our BCNLMS method gives better estimation results comparing with traditional NLMS algorithm, which confirms the effectiveness of our proposed method in removing the noise-induced bias. Besides, three distributed BCNLMS methods outperform the non-cooperation BCNLMS algorithm, which confirms the cooperation among the agents can improve the estimation performance of the network.

Figure 7 displays the learning curves in the case that input noise variance is unknown. The proposed three distributed BCNLMS algorithms outperform the other distributed NLMS methods without compensation. Moreover, three proposed distributed BCNLMS algorithms return the same estimation quality comparing with the results shown in Figure 6 where the noise variance is known a priori. This further establishes the fact that our proposed noise variance estimation method and the modified cost function have an effect on removing the noise-induced bias and enhancing the estimation performance.

The steady-state MSD under different operation modes in the case of unknown noise variance are shown in Figure 8, where agents based on three distributed BCNLMS methods exhibit excellent performance as compared to the other three distributed NLMS methods without bias compensation. Moreover, comparing the steady-state MSD curves between the Nco-BCNLMS and other distributed methods, cooperation among the agents can make the estimates more consistent.

We also present the estimate noise variances of three distributed BCNLMS algorithms in Figure 9. Both the Con-BCNLMS and ATC-BCNLMS algorithms estimate the noise variance to approximate values of the true value. However, the Inc-BCNLMS does not perform so well due to the fact that each node uses updated data from its previous neighbor over the cyclic path. Therefore the incremental algorithm relies on the extra update steps at each agent to attain better approximations of true noise variance, which enables the Inc-BCNLMS to obtain the same estimation performance as the Con-BCNLMS and ATC-BCNLMS algorithms shown in Figure 7.

6 Conclusion

In this paper, we propose three distributed bias-compensated NLMS algorithms to address the in-network collaborative estimation problem under the situation that the input and output are contaminated by additive noise. Traditional distributed adaptive algorithms give biased results. Our analysis shows that the estimation bias is caused by input noise, and one bias-compensated NLMS method is proposed to

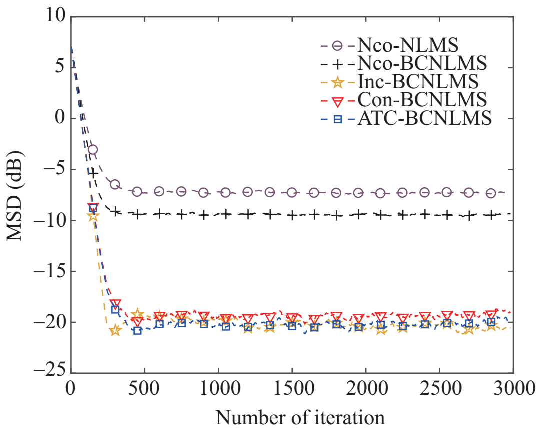

Figure 6 (Color online) MSD learning curves of the Nco-NLMS, the Nco-BCNLMS and three distributed BCNLMS algorithms in the case of known noise variance.

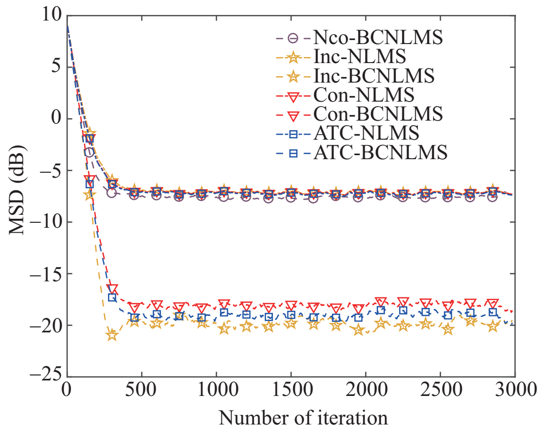

Figure 7 (Color online) MSD learning curves of the Nco-BCNLMS, three distributed NLMS algorithms and three distributed BCNLMS algorithms in the case of unknown noise variance.

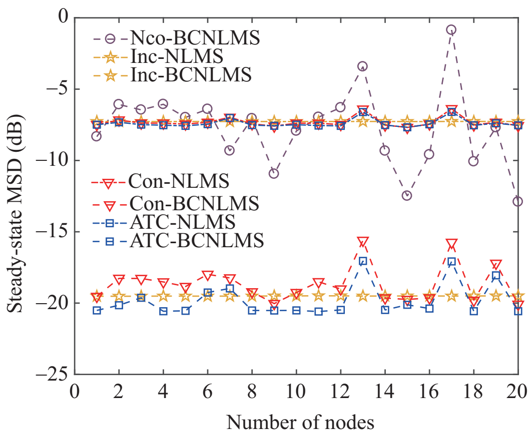

Figure 8 (Color online) The steady-state MSD curves of all in-network nodes in the case of unknown noise variance.

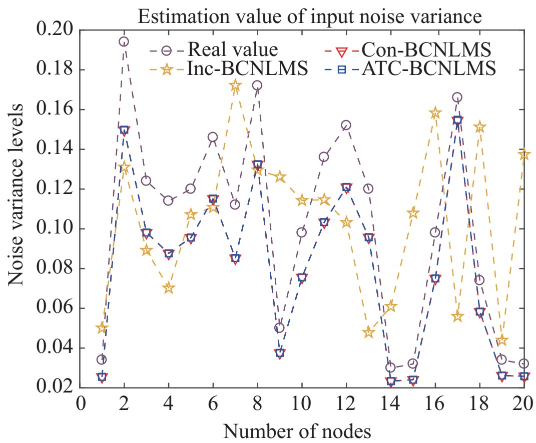

Figure 9 (Color online) The estimated input noise variance values of all in-network nodes.

remove the bias and improve the estimation performance. We also propose an estimation method for the input noise variance to overcome the unknown property of the input noise. We derive mean stability conditions and closed-form expressions to predict the steady-state mean-square performance over the network based on different strategies. Comparing to the traditional Nco-NLMS method, our proposed Nco-BCNLMS algorithm returns a better estimate, which confirms the effectiveness of our proposed bias-compensated method. Moreover, comparing to the Nco-BCNLMS strategy, three distributed BCNLMS strategies significantly improved the estimate results, which further proves the benefit of the cooperation among all nodes in the network. Simulations confirmed the efficiency of the proposed algorithms. Overall, our proposed distributed BCNLMS algorithms exhibit good performance.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant No. 61421001).

References

1 Golub G H, van Loan C F. An analysis of the total least squares problem. SIAM J Numer Anal, 1980, 17: 883–893
2 Feng D Z, Zhang X D, Chang D X, et al. A fast recursive total least squares algorithm for adaptive FIR filtering. IEEE Trans Signal Process, 2004, 52: 2729–2737

3 So H C. Modified LMS algorithm for unbiased impulse response estimation in nonstationary noise. Electron Lett, 1999, 35: 791–792

4 Feng D Z, Bao Z, Zhang X D. Modified RLS algorithm for unbiased estimation of FIR system with input and output noise. Electron Lett, 2000, 36: 273–274

5 Sardellitti S, Barbarossa S. Distributed RLS estimation for cooperative sensing in small cell networks. In: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICCASP), Vancouver, 2013. 5283–5287

6 Lorenzo P D, Isufi E, Banelli P, et al. Distributed recursive least squares strategies for adaptive reconstruction of graph signals. In: Proceedings of the 25th European Signal Processing Conference (EUSIPCO), Kos, 2017. 2289–2293

7 He X K, Xue W C, Fang H T. Consistent distributed state estimation with global observability over sensor network. 2017. ArXiv:1711.04993

8 He X K, Hu C, Xue W C, et al. On event-based distributed Kalman filter with information matrix triggers. IFAC-PapersOnLine, 2017, 50: 14308–14313

9 Lopes C G, Sayed A H. Incremental adaptive strategies over distributed networks. IEEE Trans Signal Process, 2007, 55: 4064–4077

10 Mostafapour E, Hoseini A, Nourinia J, et al. Channel estimation with adaptive incremental strategy over distributed sensor networks. In: Proceedings of the 2nd International Conference on Knowledge-Based Engineering and Innovation (KBEI), Tehran, 2015. 803–807

11 Lopes C G, Sayed A H. Distributed adaptive incremental strategies: formulation and performance analysis. In: Proceedings of IEEE International Conference on Acoustics Speech and Signal Processing Proceedings (ICASSP), Toulouse, 2006. 584–587

12 Shi L, Zhao H Q. Variable step-size distributed incremental normalised LMS algorithm. Electron Lett, 2016, 52: 519–521

13 Schizas I D, Mateos G, Giannakis G B. Distributed LMS for consensus-based in-network adaptive processing. IEEE Trans Signal Process, 2009, 57: 2365–2382

14 Braca P, Marano S, Matta V. Running consensus in wireless sensor networks. In: Proceedings of the 11th International Conference on Information Fusion (ICIF), Cologne, 2008

15 Kar S, Moura J M F. Distributed consensus algorithms in sensor networks with imperfect communication: link failures and channel noise. IEEE Trans Signal Process, 2009, 57: 355–369

16 Bogdanović N, Plata-Chaves J, Berberidis K. Distributed diffusion-based LMS for node-specific parameter estimation over adaptive networks. In: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Florence, 2014. 7223–7227

17 Nassif R, Richard C, Chen J, et al. Diffusion LMS over multitask networks with noisy links. In: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Shanghai, 2016. 4583–4587

18 Lee H S, Kim S E, Lee J W, et al. A variable step-size diffusion LMS algorithm for distributed estimation. IEEE Trans Signal Process, 2015, 63: 1808–1820

19 Cattivelli F S, Lopes C G, Sayed A H. Diffusion recursive least-squares for distributed estimation over adaptive networks. IEEE Trans Signal Process, 2008, 56: 1865–1877

20 Lopes C G, Sayed A H. Diffusion least-mean squares over adaptive networks: formulation and performance analysis. IEEE Trans Signal Process, 2008, 56: 3122–3136

21 Mateos G, Schizas I D, Giannakis G B. Distributed recursive least-squares for consensus-based in-network adaptive estimation. IEEE Trans Signal Process, 2009, 57: 4583–4588

22 Cattivelli F S, Sayed A H. Diffusion LMS strategies for distributed estimation. IEEE Trans Signal Process, 2010, 58: 1035–1048

23 Bertrand A, Moonen M, Sayed A H. Diffusion bias-compensated RLS estimation over adaptive networks. IEEE Trans Signal Process, 2011, 59: 5212–5224

24 Bertrand A, Moonen M. Consensus-based distributed total least squares estimation in Ad hoc wireless sensor networks. IEEE Trans Signal Process, 2011, 59: 2320–2330

25 Bertrand A, Moonen M, Sayed A H. Diffusion-based bias-compensated RLS for distributed estimation over adaptive sensor networks. In: Proceedings of the 19th European Signal Processing Conference (EUSIPCO), Barcelona, 2011. 1025–1029

26 Abdolee R, Champagne B, Sayed A H. A diffusion LMS strategy for parameter estimation in noisy regressor applications. In: Proceedings of the 20th European Signal Processing Conference (EUSIPCO), Bucharest, 2012. 749–753

27 Sayed A H. Adaptive Filters. Hoboken: Wiley, 2008

28 Haykin S. Adaptive Filter Theory. Englewood Cliffs: Prentice-Hall, 2002

29 Nagumo J, Noda A. A learning method for system identification. IEEE Trans Autom Control, 1967, 12: 282–287

30 Lou J, Jia L J, Tao R, et al. Distributed incremental bias-compensated RLS estimation over multi-agent networks. Sci China Inf Sci, 2017, 60: 032204