Robust Multitask Diffusion Bias-Compensated LMS Algorithm for Distributed Estimation with Noisy Link and Input

Xuzhao Chai[1], Linru Zhang[1], Pengwei Wen[1]✉, Sheng Zhang[2], Haiquan Zhao[3]
(1. School of Automation and Electrical Engineering, Zhongyuan University of Technology, Zhengzhou 450007, China; 2. School of Information Science and Technology, Southwest Jiaotong University, Chengdu 611756, China; 3. School of Electrical Engineering, Southwest Jiaotong University, Chengdu 611756, China)

Abstract: This paper presents a robust multitask diffusion average bias compensation least mean square (RM-DABC-LMS) algorithm for distributed estimation in noisy input and communication link noise. The algorithm utilizes a robust cost function based on the maximum Versoria criterion, incorporates bias compensation, and applies adaptive combination coefficients to reduce noise impacts. Theoretical analysis demonstrates the stability of the algorithm, providing closed-form expressions for the steady-state mean square deviation (MSD). A compression diffusion strategy is introduced to reduce communication cost of the RM-DABC-LMS algorithm, ensuring fast convergence and accurate estimation. Simulation results indicate that the proposed algorithm outperforms existing methods in noisy environments, achieving faster convergence and lower steady-state error.

Keywords: multitask networks; robust; bias compensation; average estimate

1 Introduction

Distributed adaptive learning has attracted significant attention for enabling networked nodes to collaboratively estimate target parameters. It has been widely applied in fields like system identification, target tracking, and sensor networks [1–7]. Major strategies in distributed learning include incremental strategies [8, 9], consensus approaches [10], and diffusion approaches [11, 12].

Diffusion based collaborative strategies are favored due to their robustness to link failures and structural stability [13, 14]. Some diffusion adaptive algorithms have been proposed such as diffusion least mean square (DLMS) algorithms and diffusion recursive least squares (DRLS) algorithms [15, 16]. However, with the increasing complexity of network models and tasks, it is more common to find applications oriented towards multi-tasking where nodes collaboratively estimate multiple target parameters [17, 18].

Recently, the research on multitask distributed estimation has become increasingly in-

Manuscript received Sep. 30, 2025; revised Dec. 8, 2025; accepted Feb. 23, 2026. The associate editor coordinating the review of this manuscript was Dr. Lijuan Jia. This work was supported by the National Natural Science Foundation of China (Nos. 62373389, 62576372), the Leading Talents of Science and Technology in the Central Plain of China (No. 254000510055), the Science and Technology Innovation Talents of Colleges and Universities in Henan Province (Nos. 24HASTIT037, 26HASTIT069), the Key Research and Development Program of Henan (No. 241111210100), the Natural Science Foundation of Zhongyuan University of Technology (No. K2025ZD008), the Postgraduate Education Reform and Quality Improvement Project of Henan Province (Nos. YJS2026YBGZZ20, 202622), the Foundation Research Project of Henan Provincial Key Scientific Research Program in Higher Education Institutions (No. 26ZX023), the National Science Foundation of Henan Province (No. 252300421520), the Joint Fund of Science and Technology R&D Program of Henan (No. 252103810253), and the 2026 Henan Province Graduate Education Reform and Quality Enhancement Project (Textbook Project).
✉ Corresponding author. Email: pwwen@zut.edu.cn
DOI: 10.15918/j.jbit1004-0579.2025.065

depth and related to many applications, including multitask diffusion affine projection algorithm (APA) [19], multitask DLMS (MDLMS) algorithm and [20] multitask DRLS (MDRLS) [21] algorithm.

The classical multitask diffusion algorithms generally assume that the input signal is free from noise. However, in real-world applications, random noise, such as measurement and quantization noise, introduces estimation bias. To address these issues, diffusion bias compensation least mean square (LMS) and recursive least-square (RLS) algorithms were proposed in [22–24], and [25]. In [26], a multitask online bias compensation method was introduced to improve the bias estimation performance of the diffusion LMS algorithm. Theoretical analyses and simulation results confirm that these algorithms can achieve unbiased estimation.

Apart from input noise, link noise also significantly impacts network performance. Data exchanged between nodes can be affected by quantization errors or additive noise during transmission, reducing the accuracy of weight updates [27]. To mitigate link noise, several strategies have been proposed, including an online adjustment of the combination coefficient in the diffusion LMS algorithm [28, 29]. [30] combines constrained LMS (CLMS) with diffusion techniques to propose a diffusion bias-compensated CLMS (D-BC-CLMS) algorithm. In [31], a link-noise-resistant diffusion algorithm was developed, and [32] replaced coefficients from a partially cooperative strategy with those from a non-cooperative one to reduce link noise. Additionally, [33] introduced a temporal processing step using smoothing techniques to effectively address link noise.

The aforementioned algorithms are primarily designed for scenarios with Gaussian input and output noise, typically in traditional sensor networks and communication systems. However, their performance significantly degrades in the presence of non-Gaussian output noise, which is common in real-world applications such as environmental monitoring, smart grid state estimation, and distributed target tracking. This highlights the need for a robust diffusion algorithm that can simultaneously suppress impulsive disturbances, eliminate input-noise-induced bias, and ensure reliable information fusion under noisy communication conditions. To address these challenges, we propose the robust multitask diffusion averaging bias-compensated LMS (RM-DABC-LMS) algorithm.

1) A robust cost function based on the maximum Versoria function is designed to suppress non-Gaussian pulse noise.

2) Bias compensation and denoising averaging methods are introduced to reduce errors from input and link noise.

3) The closed-form expression for steady-state mean squared error (MSE) and convergence condition are established.

4) A low-communication variant, compressed robust multitask diffusion averaging bias-compensated LMS (CRM)-DABC-LMS, is proposed by integrating a compression-diffusion strategy.

5) Simulations show that the proposed algorithm outperforms other multitask distributed algorithms in noisy environments.

2 Multitask Network Model

Consider a multitask adaptive network comprising N nodes, whose topology is illustrated in Fig. 1. All nodes are partitioned into Q clusters, with each cluster assigned distinct tasks. Nodes belonging to different clusters undertake sepa-

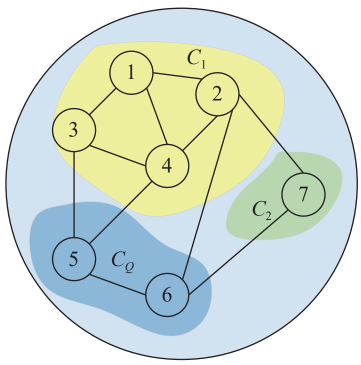

Fig. 1 Multitask clustering network architecture diagram

rate tasks, whilst nodes within the same cluster share identical tasks, namely

$$\begin{cases} \boldsymbol{w}_k^o = \boldsymbol{w}_l^o, & \text{where } k,\ l \in C_p \\ \boldsymbol{w}_{C_p}^o \sim \boldsymbol{w}_{C_q}^o, & \text{if } C_p,\ C_q \text{ are connected} \end{cases} \tag{1}$$

At each moment i, each node k may obtain input $\boldsymbol{u}_{k,i}$, along with observable output $\boldsymbol{d}_{k,i}$ associated via the following linear model

$$\boldsymbol{d}_{k,i} = \boldsymbol{u}_{k,i}^{\mathrm{T}} \boldsymbol{w}_k^o + \boldsymbol{v}_{k,i} \tag{2}$$

where $\boldsymbol{u}_{k,i}$ denotes the input regression vector at node k, $\boldsymbol{w}_k^o$ represents the optimal parameter vector to be estimated, and $\boldsymbol{v}_{k,i}$ indicates the zero mean noise at node k during time step i, with variance $\sigma_{v,i}^2$.

Assumption 1 At each node k, the regression vector $\boldsymbol{u}_{k,i}$ is generated by a zero-mean random process satisfying temporal stationarity and spatial independence, with covariance matrix $\boldsymbol{R}_{u,k} = E\{\boldsymbol{u}_{k,i}\boldsymbol{u}_{k,i}^{\mathrm{T}}\} > 0$.

Assumption 2 At each node k, the noise $\boldsymbol{v}_{k,i}$ satisfies independent and identically distributed properties in both time and space, with covariance matrix $\boldsymbol{R}_{v,k} = \sigma_{v,i}^2 \boldsymbol{I}_M$.

Utilising stochastic gradient descent and local instantaneous approximations, the multitask diffusion algorithm based on adapt-then-combine (ATC) combination step is

$$\begin{cases} \boldsymbol{\varphi}_{k,i} = \boldsymbol{w}_{k,i-1} + \mu_k \boldsymbol{u}_{k,i}(\boldsymbol{d}_{k,i} - \boldsymbol{u}_{k,i}^{\mathrm{T}} \boldsymbol{w}_{k,i-1}) \\ \boldsymbol{w}_{k,i} = \sum\limits_{l \in N_k} c_{lk,i} \boldsymbol{\varphi}_{l,i} \end{cases} \tag{3}$$

where μ_k denotes the step size, and $c_{lk,i}$ represents the non-negative combination coefficient for integrating neighbouring node information, corresponding to the (l,k)th entry of the left randomized matrix $\boldsymbol{C}$, satisfying the following conditions

$$\boldsymbol{C}^{\mathrm{T}} \mathbf{1}_N = \mathbf{1}_N \tag{4}$$

where $\mathbf{1}_N$ denotes an $N \times 1$ column vector. The non-negative combinatorial coefficients $c_{lk,i}$ satisfy the following conditions

$$\begin{cases} c_{lk,i} \geqslant 0, \quad \sum\limits_{l \in N_k} c_{lk,i} = 1 \\ c_{lk,i} = 0, \quad \text{if } l \notin N_k \end{cases} \tag{5}$$

In the combination step, $\boldsymbol{\varphi}_{k,i}$ is received from the neighbourhood of node k, involving information sharing between the node k and its neighbours. During this process, the presence of input noise and communication link noise is taken into account.

2.1 Input Noise

When considering the measurement noise of the input signal, the observed input data is represented as

$$\boldsymbol{u}_{ok,i} = \boldsymbol{u}_{k,i} + \boldsymbol{o}_{k,i}$$

where $\boldsymbol{o}_{k,i}$ denotes Gaussian noise, and $\boldsymbol{u}_{ok,i}$ represents the noisy input regression vector. The presence of input noise leads to biased estimates in the multitask diffusion LMS algorithm.

2.2 Input Noise

When information is exchanged over a link subject to additive noise, node k estimates the local weight vector received from its neighbouring node l as follows

$$\boldsymbol{\Psi}_{lk,i} = \boldsymbol{\varphi}_{l,i} + \boldsymbol{\eta}_{lk,i}$$

where $\boldsymbol{\eta}_{lk,i}$ denotes the link transmission noise, while $\boldsymbol{\Psi}_{lk,i}$ represents the noisy intermediate estimate received by node k from node l. When employing the imprecise vector $\boldsymbol{\Psi}_{lk,i}$ instead of $\boldsymbol{\varphi}_{l,i}$ during the combination step, the multitask DLMS algorithm will incur substantial steady-state error.

3 Proposed Algorithms

This section introduces an RM-DABC-LMS to address the issues discussed. The algorithm designs a cost function based on maximum Versoria Criterion (MVC), with a normalized version to improve distributed estimation performance in multitask networks in non-Gaussian noise conditions. The RM-DABC-LMS algorithm uses Versoria as the cost function, effectively suppressing non-Gaussian noise. It is defined in the real domain as shown below [34]

$$f(\boldsymbol{e}_{k,i}) = \frac{8a^3}{4a^2 + \boldsymbol{e}_{k,i}^2} \tag{6}$$

where $a > 0$ denotes the radius of the circle, and $\boldsymbol{e}_{k,i} = \boldsymbol{d}_{k,i} - \boldsymbol{u}_{ok,i}^{\mathrm{T}} \boldsymbol{w}_{k,i-1}$ represents the error signal. By adjusting the parameter, an optimal balance between convergence accuracy and robustness can be achieved, improving the algorithm's stability.

Consequently, the optimization problem for multitask distributed parameter estimation can be redefined as

$$\min_{\boldsymbol{w}_{k,i}} E\left\{\frac{8a^3}{4a^2+\boldsymbol{e}_{k,i}^2}\right\}$$

The cost function $J_k(\boldsymbol{w}_{k,i})$, based on MVC, is defined as follows

$$J_k(\boldsymbol{w}_{k,i}) = E\left\{\frac{8a^3}{4a^2+\boldsymbol{e}_{k,i}^2}\right\} \tag{7}$$

The derivative of equation (8) with respect to $\boldsymbol{w}_{k,i}$ is given as follows

$$\frac{\partial J_k(\boldsymbol{w}_{k,i})}{\partial \boldsymbol{w}_{k,i}} = \frac{\frac{1}{a}}{\left(1+\frac{\boldsymbol{e}_{k,i}^2}{4a^2}\right)^2}\boldsymbol{e}_{k,i}\boldsymbol{u}_{k,i} \tag{8}$$

The adaptive update equation for the RM-DABC-LMS algorithm is as follows

$$\boldsymbol{\varphi}_{k,i} = \boldsymbol{w}_{k,i-1} + \mu_k \overline{\boldsymbol{V}}_{k,i}\boldsymbol{e}_{k,i}\boldsymbol{u}_{ok,i} \tag{9}$$

where $\overline{\boldsymbol{V}}_{k,i} = \frac{\frac{1}{a}}{\left(1+\frac{\boldsymbol{e}_{k,i}^2}{4a^2}\right)^2}$.

To improve the algorithm's stability and convergence rate, a normalized version was derived from the adaptive update formula as shown below

$$\boldsymbol{\varphi}_{k,i} = \boldsymbol{w}_{k,i-1} + \mu_k \overline{\boldsymbol{V}}_{k,i}\boldsymbol{e}_{k,i}\frac{\boldsymbol{u}_{ok,i}}{\|\boldsymbol{u}_{ok,i}\|_2} \tag{10}$$

Noise interference in the input signal causes the estimate $\boldsymbol{w}_{k,i}$ to deviate from the optimal value $\boldsymbol{w}_k^o$. A bias compensation term $\boldsymbol{h}_{k,i}$ is introduced to correct the bias caused by input noise. The adaptive update formula for the RM-DABC-LMS algorithm is given by

$$\boldsymbol{\varphi}_{k,i} = \boldsymbol{w}_{k,i-1} + \mu_k \overline{\boldsymbol{V}}_{k,i}\boldsymbol{e}_{k,i}\frac{\boldsymbol{u}_{ok,i}}{\|\boldsymbol{u}_{ok,i}\|_2} + \boldsymbol{h}_{k,i} \tag{11}$$

The weighting error, estimation error, and intermediate estimation error are defined as

$$\tilde{\boldsymbol{\varphi}}_{k,i} \triangleq \boldsymbol{w}_k^o - \boldsymbol{\varphi}_{k,i} \tag{12}$$

$$\tilde{\boldsymbol{w}}_{k,i} \triangleq \boldsymbol{w}_k^o - \boldsymbol{w}_{k,i} \tag{13}$$

Replace (12) and (13) into Eq. (11)

$$\begin{aligned} E\{\tilde{\boldsymbol{\varphi}}_{k,i}\} = {} & E\{\tilde{\boldsymbol{w}}_{k,i}\} - \\ & \mu_k E\left\{\overline{\boldsymbol{V}}_{k,i}\boldsymbol{e}_{k,i}\frac{\boldsymbol{u}_{ok,i}}{\|\boldsymbol{u}_{ok,i}\|_2}\right\} - E\{\boldsymbol{h}_{k,i}\} \end{aligned} \tag{14}$$

If $\boldsymbol{\varphi}_{k,i}$ is an unbiased estimator of $\boldsymbol{w}_{k,i}$, it must satisfy the following conditions

$$\begin{cases} \lim\limits_{i\to\infty} E\{\tilde{\boldsymbol{\varphi}}_{k,i}\} = \boldsymbol{0}_{NM}, k=1,2,\cdots,N \\ \lim\limits_{i\to\infty} E\{\tilde{\boldsymbol{w}}_{k,i}\} = \boldsymbol{0}_{NM}, k=1,2,\cdots,N \end{cases} \tag{15}$$

Assumption 3 Compared to the error signal $\boldsymbol{e}_{k,i}$ and the input signal $\boldsymbol{u}_{ok,i}$, the rate of $\overline{\boldsymbol{V}}_{k,i}$ change is very slow. Therefore, $\overline{\boldsymbol{V}}_{k,i}$ is independent of the variables $\boldsymbol{e}_{k,i}$ and $\boldsymbol{u}_{ok,i}$.

(14) and (15) can be calculated as

$$\begin{aligned} \lim_{i\to\infty} E\{\boldsymbol{h}_{k,i}\} = {} & -\lim_{i\to\infty}\mu_k E\left\{\overline{\boldsymbol{V}}_{k,i}\boldsymbol{e}_{k,i}\frac{\boldsymbol{u}_{ok,i}}{\boldsymbol{u}_{ok,i2}}\right\} = \\ & \lim_{i\to\infty}\frac{\mu_k\sigma_{o,k}^2 E\{\overline{\boldsymbol{V}}_{k,i}\}E\{\boldsymbol{w}_{k,i}\}}{E\{\|\boldsymbol{u}_{ok,i}\|_2\}} \end{aligned} \tag{16}$$

The final deviation compensation term $\boldsymbol{h}_{k,i}$, derived using the stochastic approximation method, is

$$\boldsymbol{h}_{k,i} = \frac{\mu_k\sigma_{o,k}^2\overline{\boldsymbol{V}}_{k,i}\boldsymbol{w}_{k,i}}{\boldsymbol{u}_{ok,i2}} \tag{17}$$

Therefore, the final adaptive update equation for the proposed RM-DABC-LMS algorithm is expressed as

$$\begin{aligned} \boldsymbol{\varphi}_{k,i} = {} & \boldsymbol{w}_{k,i-1} + \mu_k\overline{\boldsymbol{V}}_{k,i}\boldsymbol{e}_{k,i}\frac{\boldsymbol{u}_{ok,i}}{\|\boldsymbol{u}_{ok,i}\|_2} + \\ & \mu_k\frac{\sigma_{o,k}^2\overline{\boldsymbol{V}}_{k,i}\boldsymbol{w}_{k,i-1}}{\|\boldsymbol{u}_{ok,i}\|_2} \end{aligned} \tag{18}$$

Under pulse noise interference, the robust RM-DABC-LMS algorithm adopts the following ATC form

$$\begin{cases} \boldsymbol{\varphi}_{k,i} = \boldsymbol{w}_{k,i-1} + \mu_k\overline{\boldsymbol{V}}_{k,i}\boldsymbol{e}_{k,i}\dfrac{\boldsymbol{u}_{ok,i}}{\|\boldsymbol{u}_{ok,i}\|_2} + \\ \qquad \mu_k\dfrac{\sigma_{o,k}^2\overline{\boldsymbol{V}}_{k,i}\boldsymbol{w}_{k,i-1}}{\|\boldsymbol{u}_{ok,i}\|_2} \\ \boldsymbol{w}_{k,i} = a_{k,i}\boldsymbol{\varphi}_{k,i} + \boldsymbol{\phi}_{k,i} \end{cases} \tag{19}$$

The correlation coefficient $a_{k,i}$ is defined as

$$a_{lk,i} = \begin{cases} \mu_k b_{lk,i}, & l \neq k \\ 1-\mu_k\sum\limits_{l\in N_k/\{k\}} b_{lk,i}, & \text{otherwise} \end{cases} \tag{20}$$

Additionally, $a_{k,i}$ satisfies the conditions below

$$\begin{cases} a_{lk,i} \geqslant 0, & \sum_{l\in N_k} a_{lk,i} = 1 \\ a_{lk,i} = 0, & \text{if } l \notin N_k \end{cases} \tag{21}$$

In multitask networks, assigning equal weights to all nodes within a neighbourhood is suboptimal. This approach overlooks task correlations and variations in node states, which can result in imbalanced information fusion and degrade network performance. Therefore, the following adaptive combination coefficients [35] are used

$$b_{lk,i} = \frac{\vartheta_{lk,i}^{-2}}{\sum_{j\in N_k} \vartheta_{jk,i}^{-2}} = \frac{\| \boldsymbol{\psi}_{k,i} + \mu_k \boldsymbol{g}_{k,i-1} - \boldsymbol{\psi}_{l,i} \|^{-2}}{\sum_{j\in N_k} \| \boldsymbol{\psi}_{k,i} + \mu_k \boldsymbol{g}_{k,i-1} - \boldsymbol{\psi}_{j,i} \|^{-2}} \tag{22}$$

where $\boldsymbol{g}_{k,i}$ is given by the following formula

$$\boldsymbol{g}_{k,i} = [\boldsymbol{d}_{k,i} - \boldsymbol{u}_{ok,i}^{\mathrm{T}} \boldsymbol{\psi}_{k,i+1}] \boldsymbol{u}_{ok,i} + \sigma_{o,k}^2 \boldsymbol{\psi}_{k,i+1} \tag{23}$$

In consideration of the impact of communication link noise on multitask networks, this paper adopts a sliding average denoising strategy to mitigate link noise during the combination phase

$$\phi_{k,i} = (1-\gamma')\phi_{k,i-1} + \gamma' \sum_{l\in N_k/\{k\}} a_{lk,i} \boldsymbol{\psi}_{lk,i} \tag{24}$$

where γ' represents the forgetting factor, with $0<\gamma'<1$. A summary of the proposed RM-DABC-LMS algorithm is provided in Tab. 1.

4 Performance Analysis

This section analyses the performance of the proposed RM-DABC-LMS algorithm based on mean and mean squared error. The weight error vectors $\tilde{\boldsymbol{\varphi}}'_{k,i}$, $\tilde{\boldsymbol{w}}'_{k,i}$, and $\tilde{\boldsymbol{\phi}}'_{k,i}$ for node k at time step i are defined as follows

$$\tilde{\boldsymbol{\varphi}}'_{k,i} \triangleq \boldsymbol{w}_k^o - \boldsymbol{\varphi}_{k,i} \tag{25}$$

$$\tilde{\boldsymbol{w}}'_{k,i} \triangleq \boldsymbol{w}_k^o - \boldsymbol{w}_{k,i} \tag{26}$$

$$\tilde{\boldsymbol{\phi}}'_{k,i} = \sum_{l\in N_k/\{k\}} a_{lk,i} \boldsymbol{w}_k^o - \boldsymbol{\phi}_{k,i} \tag{27}$$

Tab. 1 Summary of RM-DABC-LMS algorithm

Parameters: a, k, μ_k, γ

Initialisation: $\overline{\boldsymbol{V}}_{k,i} = \boldsymbol{0}_{NM}, W_{k,i} = \boldsymbol{0}_{NM}, \varphi_{k,i} = \boldsymbol{0}_{MN}$

For $i = 1,2,3\cdots$

  For $k = 1,2,3\cdots$

$$\boldsymbol{e}_{k,i} = \boldsymbol{d}_{k,i} - \boldsymbol{u}_{ok,i}^{\mathrm{T}} \boldsymbol{w}_{k,i-1}$$

$$\overline{\boldsymbol{V}}_{k,i} = \frac{\frac{1}{a}}{\left(1 + \frac{\boldsymbol{e}_{k,i}^2}{4a^2}\right)^2}$$

$$\boldsymbol{\varphi}_{k,i} = \boldsymbol{w}_{k,i-1} + \mu_k \overline{\boldsymbol{V}}_{k,i} \boldsymbol{e}_{k,i} \frac{\boldsymbol{u}_{ok,i}}{\|\boldsymbol{u}_{ok,i}\|_2} + \mu_k \frac{\sigma_{o,k}^2 \overline{\boldsymbol{V}}_{k,i} \boldsymbol{w}_{k,i-1}}{\|\boldsymbol{u}_{ok,i}\|_2}$$

$$a_{lk,i} = \begin{cases} \mu_k b_{lk,i}, l \neq k \\ 1 - \mu_k \sum_{l\in N_k/\{k\}} b_{lk,i}, \text{otherwise} \end{cases}$$

$$\boldsymbol{\phi}_{k,i} = (1-\gamma)\boldsymbol{\phi}_{k,i-1} + \gamma \sum_{l\in N_k/\{k\}} a_{lk,i} \boldsymbol{\psi}_{lk,i}$$

$$\boldsymbol{w}_{k,i} = a_{k,i} \boldsymbol{\varphi}_{k,i} + \boldsymbol{\phi}_{k,i}$$

  end

end

4.1 Mean Stability

The algorithm's estimate $\boldsymbol{w}_{k,i}$ converge to the true weight $\boldsymbol{w}_k^o$ in the mean sense, i.e.,

$$\lim_{i\to\infty} E\{\boldsymbol{w}_{k,i}\} = \boldsymbol{w}_k^o \tag{28}$$

Substituting the adaptive update Eq. (18) and the denoised mean estimate $\boldsymbol{\phi}_{k,i}$ into the weight error vector (25), (26), and (27) results in

$$\tilde{\boldsymbol{w}}'_{k,i} = a_{k,i} \tilde{\boldsymbol{\varphi}}'_{k,i} + \tilde{\boldsymbol{\phi}}'_{k,i} \tag{29}$$

$$\tilde{\boldsymbol{\varphi}}'_{k,i} = \gamma' \sum_{l\in N_k/\{k\}} a_{lk,i} \tilde{\boldsymbol{\varphi}}'_{k,i} - \gamma' \sum_{l\in N_k/\{k\}} a_{lk,i} \boldsymbol{\eta}_{lk,i} + (\boldsymbol{1}+\gamma') \tilde{\boldsymbol{\phi}}'_{k,i} \tag{30}$$

$$\tilde{\boldsymbol{\varphi}}'_{k,i} = \tilde{\boldsymbol{w}}'_{k,i-1} - \mu_k \overline{\boldsymbol{V}}_{k,i} \frac{\boldsymbol{u}_{ok,i}}{||\boldsymbol{u}_{ok,i}||_2} \boldsymbol{v}_{k,i} - \mu_k \overline{\boldsymbol{V}}_{k,i} \frac{\boldsymbol{u}_{ok,i}^{\mathrm{T}} \boldsymbol{u}_{ok,i} - \sigma_{o,k}^2 \boldsymbol{I}_M}{||\boldsymbol{u}_{ok,i}||_2} \tilde{\boldsymbol{w}}'_{k,i} + \mu_k \overline{\boldsymbol{V}}_{k,i} \frac{\boldsymbol{o}_{k,i}^{\mathrm{T}} \boldsymbol{u}_{ok,i} - \sigma_{o,k}^2 \boldsymbol{I}_M}{||\boldsymbol{u}_{ok,i}||_2} \boldsymbol{w}_k^o \tag{31}$$

For computational simplicity, an MM block diagonal matrix and an NN block vector are introduced. First, the global weight error vectors $\tilde{\boldsymbol{\varphi}}'_i$, $\tilde{\boldsymbol{w}}'_i$, and $\tilde{\boldsymbol{\phi}}'_i$ are defined as follows

$$\begin{aligned}\tilde{\boldsymbol{\varphi}}'_i &= \mathbf{col}\{\tilde{\boldsymbol{\varphi}}'_{1,i}, \tilde{\boldsymbol{\varphi}}'_{2,i}, \cdots, \tilde{\boldsymbol{\varphi}}'_{N,i}\}\\ \tilde{\boldsymbol{w}}'_i &= \mathbf{col}\{\tilde{\boldsymbol{w}}'_{1,i}, \tilde{\boldsymbol{w}}'_{2,i}, \cdots, \tilde{\boldsymbol{w}}'_{N,i}\}\\ \tilde{\boldsymbol{\phi}}'_i &= \mathbf{col}\{\tilde{\boldsymbol{\phi}}'_{1,i}, \tilde{\boldsymbol{\phi}}'_{2,i}, \cdots, \tilde{\boldsymbol{\phi}}'_{N,i}\}\end{aligned} \quad (32)$$

and the following matrix $\boldsymbol{H}'$, $\boldsymbol{H}'_h$, $\boldsymbol{M}'$, $\boldsymbol{V}'$, $\boldsymbol{F}'$, $\boldsymbol{O}'$, $\boldsymbol{U}'$ and $\overline{\boldsymbol{V}}_i$

$$\begin{aligned}&\boldsymbol{H}' = \boldsymbol{H}'_i \otimes \boldsymbol{I}_M\\ &\boldsymbol{H}'_h = \text{diag}\{\boldsymbol{H}'_i\} \otimes \boldsymbol{I}_M\\ &\boldsymbol{M}' = \text{diag}\{\mu_1 \boldsymbol{I}_M, \mu_2 \boldsymbol{I}_M, \cdots, \mu_N \boldsymbol{I}_M\}\\ &\boldsymbol{V}' = \mathbf{col}\{\boldsymbol{\eta}_{l1,i}\boldsymbol{I}_M, \boldsymbol{\eta}_{l2,i}\boldsymbol{I}_M, \cdots, \boldsymbol{\eta}_{lN,i}\boldsymbol{I}_M\}\\ &\boldsymbol{F}'_i = \mathbf{col}\left\{\frac{\boldsymbol{u}_{o1,i}}{\|\boldsymbol{u}_{o1,i}\|_2}\boldsymbol{v}_{1,i}, \frac{\boldsymbol{u}_{o2,i}}{\|\boldsymbol{u}_{o2,i}\|_2}\boldsymbol{v}_{2,i}, \cdots, \frac{\boldsymbol{u}_{oN,i}}{\|\boldsymbol{u}_{oN,i}\|_2}\boldsymbol{v}_{N,i}\right\}\\ &\boldsymbol{O}'_i = \text{diag}\left\{\frac{\boldsymbol{u}_{o1,i}\boldsymbol{o}_{1,i}^{\mathrm{T}} - \sigma_{o,1}^2 \boldsymbol{I}_M}{\|\boldsymbol{u}_{o1,i}\|_2}, \frac{\boldsymbol{u}_{o2,i}\boldsymbol{o}_{2,i}^{\mathrm{T}} - \sigma_{o,2}^2 \boldsymbol{I}_M}{\|\boldsymbol{u}_{o2,i}\|_2}, \cdots, \frac{\boldsymbol{u}_{oN,i}\boldsymbol{o}_{N,i}^{\mathrm{T}} - \sigma_{o,N}^2 \boldsymbol{I}_M}{\|\boldsymbol{u}_{oN,i}\|_2}\right\}\\ &\boldsymbol{U}'_i = \text{diag}\left\{\frac{\boldsymbol{u}_{o1,i}\boldsymbol{u}_{o1,i}^{\mathrm{T}} - \sigma_{o,1}^2 \boldsymbol{I}_M}{\|\boldsymbol{u}_{o1,i}\|_2}, \frac{\boldsymbol{u}_{o2,i}\boldsymbol{u}_{o2,i}^{\mathrm{T}} - \sigma_{o,2}^2 \boldsymbol{I}_M}{\|\boldsymbol{u}_{o2,i}\|_2}, \cdots, \frac{\boldsymbol{u}_{oN,i}\boldsymbol{u}_{oN,i}^{\mathrm{T}} - \sigma_{o,N}^2 \boldsymbol{I}_M}{\|\boldsymbol{u}_{oN,i}\|_2}\right\}\\ &\overline{\boldsymbol{V}}_i = \text{diag}\left\{\frac{\frac{1}{a}}{\left(1 + \frac{\boldsymbol{e}_{1,i}^2}{4a^2}\right)^2}, \frac{\frac{1}{a}}{\left(1 + \frac{\boldsymbol{e}_{2,i}^2}{4a^2}\right)^2}, \cdots, \frac{\frac{1}{a}}{\left(1 + \frac{\boldsymbol{e}_{N,i}^2}{4a^2}\right)^2}\right\}\end{aligned} \quad (33)$$

where $\boldsymbol{I}_M$ represents the identity matrix of size MM, $\mathbf{col}\{\cdot\}$ indicates a column vector, and $\otimes$ refers to the Kronecker product.

$$\begin{aligned}\tilde{\boldsymbol{w}}'_{k,i} =& \boldsymbol{H}_h'^{\mathrm{T}}[(\boldsymbol{I}_{NM} - \boldsymbol{M}'\overline{\boldsymbol{V}}_i \boldsymbol{u}')\tilde{\boldsymbol{w}}'_{k,i-1} -\\ & \boldsymbol{M}'\overline{\boldsymbol{V}}_i(\boldsymbol{F}'_i - \boldsymbol{O}'_i \boldsymbol{w}^o_{C(k)})] + \tilde{\boldsymbol{\phi}}'_i\end{aligned} \quad (34)$$

$$\begin{aligned}\tilde{\boldsymbol{\varphi}}'_{k,i} =& (\boldsymbol{I}_{NM} - \boldsymbol{M}'\overline{\boldsymbol{V}}_i{}' \boldsymbol{u}')\tilde{\boldsymbol{w}}'_{k,i-1} -\\ & \boldsymbol{M}'\overline{\boldsymbol{V}}_i(\boldsymbol{F}'_i - \boldsymbol{O}'_i \boldsymbol{w}^o_{C(k)})\end{aligned} \quad (35)$$

$$\begin{aligned}\tilde{\boldsymbol{\phi}}'_{k,i} =& (1-\gamma')\tilde{\boldsymbol{\phi}}'_{k,i-1} + \gamma' \boldsymbol{H}'^{\mathrm{T}}(\boldsymbol{I}_{NM} - \boldsymbol{M}'\overline{\boldsymbol{V}}_i \boldsymbol{u}')\tilde{\boldsymbol{w}}'_{k,i-1} -\\ & \gamma' \boldsymbol{H}'^{\mathrm{T}}\boldsymbol{M}'\overline{\boldsymbol{V}}_i(\boldsymbol{F}'_i - \boldsymbol{O}'_i \boldsymbol{w}^o_{C(k)}) - \gamma' \boldsymbol{H}'^{\mathrm{T}}\boldsymbol{V}'\end{aligned} \quad (36)$$

Through (34) and (36), the recursive form of the global weight error vector is derived as

$$\begin{aligned}\tilde{\boldsymbol{w}}'_{k,i} =& (\boldsymbol{I}_{NM} - \boldsymbol{M}'\overline{\boldsymbol{V}}_i \boldsymbol{u}')(\boldsymbol{H}_h'^{\mathrm{T}} + \gamma' \boldsymbol{H}'^{\mathrm{T}})\tilde{\boldsymbol{w}}'_{k,i-1} -\\ & (\boldsymbol{F}'_i - \boldsymbol{O}'_i \boldsymbol{w}^o_k)(\boldsymbol{H}_h'^{\mathrm{T}} + \gamma' \boldsymbol{H}'^{\mathrm{T}})\boldsymbol{M}'\overline{\boldsymbol{V}}_i -\\ & \gamma' \boldsymbol{H}'^{\mathrm{T}}\boldsymbol{V}' + (1-\gamma')\tilde{\boldsymbol{\phi}}'_{i-1}\end{aligned} \quad (37)$$

For the sake of analysis, let us define the symbols

$$\boldsymbol{L}_i = (1-\gamma')\tilde{\boldsymbol{\phi}}'_{i-1} \quad (38)$$

$$\boldsymbol{P}_i = \gamma' \boldsymbol{H}'^{\mathrm{T}}\boldsymbol{V}' \quad (39)$$

$$\boldsymbol{Q}_i = \boldsymbol{M}'\overline{\boldsymbol{V}}_i\left(\gamma' \boldsymbol{H}'^{\mathrm{T}} + \boldsymbol{H}_h'^{\mathrm{T}}\right)(\boldsymbol{F}'_i - \boldsymbol{O}'_i \boldsymbol{w}^o_k) \quad (40)$$

Using the above symbol definitions, the formula for the estimated value $\tilde{\boldsymbol{w}}'_{k,i}$ simplifies to

$$\begin{aligned}\tilde{\boldsymbol{w}}'_{k,i} =& (\boldsymbol{I}_{NM} - \boldsymbol{M}'\overline{\boldsymbol{V}}_i \boldsymbol{U}'_i)(\boldsymbol{H}_h'^{\mathrm{T}} + \gamma' \boldsymbol{H}'^{\mathrm{T}})\tilde{\boldsymbol{w}}'_{k,i-1} -\\ & \boldsymbol{Q}_i - \boldsymbol{P}_i + \boldsymbol{L}_i\end{aligned} \quad (41)$$

Taking the expectation of both sides of equation (41) yields

$$\begin{aligned}E\{\tilde{\boldsymbol{w}}'_{k,i}\} =& (\boldsymbol{H}_h'^{\mathrm{T}} + \gamma' \boldsymbol{H}'^{\mathrm{T}})\cdot\\ & [(\boldsymbol{I}_{NM} - \boldsymbol{M}'E\{\overline{\boldsymbol{V}}_i\}E\{\boldsymbol{U}'\})E\{\tilde{\boldsymbol{w}}'_{k,i-1}\}] +\\ & E\{\boldsymbol{L}_i\}\end{aligned} \quad (42)$$

Based on the zero-mean statistical properties of noise, it can be concluded that

$$E\{\boldsymbol{F}'_i\} = \mathbf{0}_{NM}, E\{\boldsymbol{V}'\} = \mathbf{0}_{NM}, E\{\boldsymbol{O}'_i\} = \mathbf{0}_{NM} \quad (43)$$

Regardless of the initial conditions, the RM-DABC-LMS algorithm converges in expectation if the following conditions are met

$$\rho((\boldsymbol{I}_{NM} - \boldsymbol{M}'E\{\overline{\boldsymbol{V}}_i\}E\{\boldsymbol{u}_i\})(\gamma' \boldsymbol{H}'^{\mathrm{T}} + \boldsymbol{H}_h'^{\mathrm{T}})) < 1 \quad (44)$$

When $\gamma' = 1$ and $\boldsymbol{H}' = \boldsymbol{I}_N$, the step size μ_k satisfies the following condition

$$0 < \mu_k < \frac{2}{E\{\overline{\boldsymbol{V}}_i\}\lambda_{\max}(\boldsymbol{R}_{u,k})} \quad (45)$$

When $\gamma' = 0$, the RM-DABC-LMS algorithm converges asymptotically in mean if the step size μ_k satisfies the following condition

$$0 < \mu_k < \left(1 + \frac{1}{a_{k,k}}\right)\frac{1}{E\{\overline{\boldsymbol{V}}_i\}\lambda_{\max}(\boldsymbol{R}_{u,k})} \quad (46)$$

4.2 Mean Square Analysis

This section focuses on the mean square performance of the RM-DABC-LMS algorithm.

Multiplying both sides of Eq. (37) by its transpose $\tilde{\boldsymbol{w}}_i'^{\mathrm{T}}$ and then taking the expectation results in the following after a series of derivations

$$\begin{aligned}&\tilde{\boldsymbol{w}}_i'\tilde{\boldsymbol{w}}_i'^{\mathrm{T}}=\\&(\boldsymbol{I}_{NM}-\boldsymbol{M}'\overline{\boldsymbol{V}}_i\boldsymbol{u}')(\boldsymbol{H}_h'^{\mathrm{T}}+\gamma'\boldsymbol{H}'^{\mathrm{T}})\tilde{\boldsymbol{w}}_{k,i-1}'\tilde{\boldsymbol{w}}_{k,i-1}'^{\mathrm{T}}\times\\&(\boldsymbol{H}_h'^{\mathrm{T}}+\gamma'\boldsymbol{H}'^{\mathrm{T}})^{\mathrm{T}}(\boldsymbol{I}_{NM}-\boldsymbol{M}'\overline{\boldsymbol{V}}_i\boldsymbol{u}')^{\mathrm{T}}-\\&(\boldsymbol{I}_{NM}-\boldsymbol{M}'\overline{\boldsymbol{V}}_i\boldsymbol{u}')(\boldsymbol{H}_h'^{\mathrm{T}}+\gamma'\boldsymbol{H}'^{\mathrm{T}})\tilde{\boldsymbol{w}}_{k,i-1}'\boldsymbol{Q}_i^{\mathrm{T}}-\\&(\boldsymbol{I}_{NM}-\boldsymbol{M}'\overline{\boldsymbol{V}}_i\boldsymbol{u}')(\boldsymbol{H}_h'^{\mathrm{T}}+\gamma'\boldsymbol{H}'^{\mathrm{T}})\tilde{\boldsymbol{w}}_{k,i-1}'\boldsymbol{P}_i^{\mathrm{T}}+\\&(\boldsymbol{I}_{NM}-\boldsymbol{M}'\overline{\boldsymbol{V}}_i\boldsymbol{u}')\tilde{\boldsymbol{w}}_{k,i-1}'\boldsymbol{L}_i^{\mathrm{T}}-\\&\boldsymbol{Q}_i\tilde{\boldsymbol{w}}_{k,i-1}'^{\mathrm{T}}(\boldsymbol{H}_h'^{\mathrm{T}}+\gamma'\boldsymbol{H}'^{\mathrm{T}})^{\mathrm{T}}(\boldsymbol{I}_{NM}-\boldsymbol{M}'\overline{\boldsymbol{V}}_i\boldsymbol{u}')^{\mathrm{T}}-\\&\boldsymbol{P}_i\tilde{\boldsymbol{w}}_{k,i-1}'^{\mathrm{T}}(\boldsymbol{H}_h'^{\mathrm{T}}+\gamma'\boldsymbol{H}'^{\mathrm{T}})^{\mathrm{T}}(\boldsymbol{I}_{NM}-\boldsymbol{M}'\overline{\boldsymbol{V}}_i\boldsymbol{u}')^{\mathrm{T}}+\\&\boldsymbol{L}_i\tilde{\boldsymbol{w}}_{k,i-1}'^{\mathrm{T}}(\boldsymbol{H}_h'^{\mathrm{T}}+\gamma'\boldsymbol{H}'^{\mathrm{T}})^{\mathrm{T}}(\boldsymbol{I}_{NM}-\boldsymbol{M}'\overline{\boldsymbol{V}}_i\boldsymbol{u}')^{\mathrm{T}}+\\&\boldsymbol{Q}_i\boldsymbol{Q}_i^{\mathrm{T}}+\boldsymbol{Q}_i\boldsymbol{P}_i^{\mathrm{T}}-\boldsymbol{Q}_i\boldsymbol{L}_i^{\mathrm{T}}+\boldsymbol{P}_i\boldsymbol{Q}_i^{\mathrm{T}}+\boldsymbol{P}_i\boldsymbol{P}_i^{\mathrm{T}}-\\&\boldsymbol{P}_i\boldsymbol{L}_i^{\mathrm{T}}-\boldsymbol{L}_i\boldsymbol{Q}_i^{\mathrm{T}}-\boldsymbol{L}_i\boldsymbol{P}_i^{\mathrm{T}}+\boldsymbol{L}_i\boldsymbol{L}_i^{\mathrm{T}}\end{aligned}\tag{47}$$

Let $\boldsymbol{R}'=E\left\{\gamma'\boldsymbol{H}'^{\mathrm{T}}+\boldsymbol{H}_h'^{\mathrm{T}}\right\}$, and take the expectation of both sides of (47)

$$\begin{aligned}E\{\tilde{\boldsymbol{w}}_i'\tilde{\boldsymbol{w}}_i'^{\mathrm{T}}\}=&E\{(\boldsymbol{I}_{NM}-\boldsymbol{M}'\overline{\boldsymbol{V}}_i\boldsymbol{u}')\boldsymbol{R}'\tilde{\boldsymbol{w}}_{k,i-1}'\tilde{\boldsymbol{w}}_{k,i-1}'^{\mathrm{T}}\times\\&\boldsymbol{R}'^{\mathrm{T}}(\boldsymbol{I}_{NM}-\boldsymbol{M}'\overline{\boldsymbol{V}}_i\boldsymbol{u}')^{\mathrm{T}}+\\&(\boldsymbol{I}_{NM}-\boldsymbol{M}'\overline{\boldsymbol{V}}_i\boldsymbol{u}')\boldsymbol{R}'\tilde{\boldsymbol{w}}_{k,i-1}'\boldsymbol{L}_i^{\mathrm{T}}+\\&\boldsymbol{L}_i\tilde{\boldsymbol{w}}_{k,i-1}'^{\mathrm{T}}\boldsymbol{R}'^{\mathrm{T}}(\boldsymbol{I}_{NM}-\boldsymbol{M}'\overline{\boldsymbol{V}}_i\boldsymbol{u}')^{\mathrm{T}}+\\&\boldsymbol{Q}_i\boldsymbol{Q}_i^{\mathrm{T}}+\boldsymbol{P}_i\boldsymbol{P}_i^{\mathrm{T}}+\boldsymbol{L}_i\boldsymbol{L}_i^{\mathrm{T}}\}\end{aligned}\tag{48}$$

Substituting (39) and (40) into Eq. (48) results in

$$\begin{aligned}E\{\tilde{\boldsymbol{w}}_i'\tilde{\boldsymbol{w}}_i'^{\mathrm{T}}\}=&E\{(\boldsymbol{I}_{NM}-\boldsymbol{M}'\overline{\boldsymbol{V}}_i\boldsymbol{u}')\boldsymbol{R}'\tilde{\boldsymbol{w}}_{k,i-1}'\tilde{\boldsymbol{w}}_{k,i-1}'^{\mathrm{T}}\times\\&\boldsymbol{R}'^{\mathrm{T}}(\boldsymbol{I}_{NM}-\boldsymbol{M}'\overline{\boldsymbol{V}}_i\boldsymbol{u}')^{\mathrm{T}}+\\&(\boldsymbol{I}_{NM}-\boldsymbol{M}'\overline{\boldsymbol{V}}_i\boldsymbol{u}')\boldsymbol{R}'\tilde{\boldsymbol{w}}_{k,i-1}'\boldsymbol{L}_i^{\mathrm{T}}+\\&\boldsymbol{L}_i\tilde{\boldsymbol{w}}_{k,i-1}'^{\mathrm{T}}\boldsymbol{R}'^{\mathrm{T}}(\boldsymbol{I}_{NM}-\boldsymbol{M}'\overline{\boldsymbol{V}}_i\boldsymbol{u}')^{\mathrm{T}}+\\&\boldsymbol{M}'\overline{\boldsymbol{V}}_i\boldsymbol{R}'(\boldsymbol{F}_i'-\boldsymbol{O}_i'\boldsymbol{w}_k^o)(\boldsymbol{F}_i'-\boldsymbol{O}_i'\boldsymbol{w}_k^o)^{\mathrm{T}}\times\\&\boldsymbol{R}'^{\mathrm{T}}\overline{\boldsymbol{V}}_i^{\mathrm{T}}\boldsymbol{M}'^{\mathrm{T}}+\gamma'^2\boldsymbol{H}'^{\mathrm{T}}\boldsymbol{V}'\boldsymbol{V}'^{\mathrm{T}}\boldsymbol{H}'+\boldsymbol{L}_i\boldsymbol{L}_i^{\mathrm{T}}\}\end{aligned}\tag{49}$$

By applying the Kronecker product property $\mathrm{vec}(\boldsymbol{XWY})=(\boldsymbol{Y}^{\mathrm{T}}\otimes\boldsymbol{X})\mathrm{vec}(\boldsymbol{W})$, we obtain

$$\begin{aligned}&\mathrm{vec}(E\{\tilde{\boldsymbol{w}}_i'\tilde{\boldsymbol{w}}_i'^{\mathrm{T}}\})=(\boldsymbol{R}'^{\mathrm{T}}(\boldsymbol{I}_{NM}-\boldsymbol{M}'\overline{\boldsymbol{V}}_i\boldsymbol{U}')^{\mathrm{T}}\otimes\\&(\boldsymbol{I}_{NM}-\boldsymbol{M}'\overline{\boldsymbol{V}}_i\boldsymbol{U}')\boldsymbol{R}'\mathrm{vec}(E\tilde{\boldsymbol{w}}_{k,i-1}'\tilde{\boldsymbol{w}}_{k,i-1}'^{\mathrm{T}})+\\&\mathrm{vec}(\boldsymbol{M}'E\{\overline{\boldsymbol{V}}_i\}\boldsymbol{R}'(\boldsymbol{F}_i'-\boldsymbol{O}_i'E\{\boldsymbol{w}_k^o\})\times\\&(\boldsymbol{F}_i'-\boldsymbol{O}_i'E\{\boldsymbol{w}_k^o\})^{\mathrm{T}}\times\boldsymbol{R}'^{\mathrm{T}}E\{\overline{\boldsymbol{V}}_i^{\mathrm{T}}\}\boldsymbol{M}'^{\mathrm{T}})+\\&\gamma'^2\mathrm{vec}(E\{\boldsymbol{H}'^{\mathrm{T}}\boldsymbol{V}'\boldsymbol{V}'^{\mathrm{T}}\boldsymbol{H}'\})+\mathrm{vec}(E\{\boldsymbol{L}_i\boldsymbol{L}_i^{\mathrm{T}}\})-\\&\mathrm{vec}(\boldsymbol{M}'\boldsymbol{R}'\mathrm{E}\{\overline{\boldsymbol{V}}_i\}E\{\boldsymbol{U}'\tilde{\boldsymbol{w}}_{k,i-1}'\}\boldsymbol{L}_i^{\mathrm{T}})-\\&\mathrm{vec}(\boldsymbol{L}_i\boldsymbol{R}'^{\mathrm{T}}E\{\overline{\boldsymbol{V}}_i^{\mathrm{T}}\}E\{\boldsymbol{U}'^{\mathrm{T}}\tilde{\boldsymbol{w}}_{k,i-1}'^{\mathrm{T}}\}\boldsymbol{M}'^{\mathrm{T}})\end{aligned}\tag{50}$$

By defining $\boldsymbol{X}_i'$, Eq. (50) is simplified as

$$\begin{aligned}&\mathrm{vec}(E\{\tilde{\boldsymbol{w}}_i'\tilde{\boldsymbol{w}}_i'^{\mathrm{T}}\})=\boldsymbol{X}_i'\mathrm{vec}(E\{\tilde{\boldsymbol{w}}_{k,i-1}'\tilde{\boldsymbol{w}}_{k,i-1}'^{\mathrm{T}}\})+\\&\mathrm{vec}(E\{\boldsymbol{Q}_i\boldsymbol{Q}_i^{\mathrm{T}}\})+\mathrm{vec}(E\{\boldsymbol{L}_i\boldsymbol{L}_i^{\mathrm{T}}\})+\\&\gamma'^2\mathrm{vec}(E\{\boldsymbol{H}'^{\mathrm{T}}\boldsymbol{V}'\boldsymbol{V}'^{\mathrm{T}}\boldsymbol{H}'\})-\\&\mathrm{vec}(\boldsymbol{M}'\boldsymbol{R}'E\{\overline{\boldsymbol{V}}_i\}E\{\boldsymbol{U}'\tilde{\boldsymbol{w}}_{k,i-1}'\}\boldsymbol{L}_i^{\mathrm{T}})-\\&\mathrm{vec}(\boldsymbol{L}_i\boldsymbol{R}'^{\mathrm{T}}\mathrm{E}\{\overline{\boldsymbol{V}}_i^{\mathrm{T}}\}E\{\boldsymbol{U}'^{\mathrm{T}}\tilde{\boldsymbol{w}}_{k,i-1}'^{\mathrm{T}}\}\boldsymbol{M}'^{\mathrm{T}})\end{aligned}\tag{51}$$

and

$$\begin{aligned}\boldsymbol{X}'_i=&\boldsymbol{R}'\otimes\boldsymbol{R}'\left[\boldsymbol{I}_{N^2M^2}-\boldsymbol{I}_{NM}\otimes\boldsymbol{M}'E\left\{\overline{\boldsymbol{V}}_i\right\}E\left\{\boldsymbol{u}'\right\}-\right.\\&\boldsymbol{M}'E\left\{\overline{\boldsymbol{V}}_i\right\}E\left\{\boldsymbol{u}'\right\}\otimes\boldsymbol{I}_{NM}+\\&(\boldsymbol{M}'E\left\{\overline{\boldsymbol{V}}_i\right\}\otimes\boldsymbol{M}'E\left\{\overline{\boldsymbol{V}}_i\right\})\times\\&\left.(E\left\{\boldsymbol{u}'\right\}\otimes E\left\{\boldsymbol{u}'\right\}\right]\end{aligned}\tag{52}$$

As i approaches infinity, the proposed RM-DABC-LMS algorithm reaches steady-state, the following steady-state equations are derived

$$\begin{aligned}&\mathrm{vec}(E\{\tilde{\boldsymbol{w}}_\infty'\tilde{\boldsymbol{w}}_\infty'^{\mathrm{T}}\})=\\&(\boldsymbol{I}_{N^2M^2}-\boldsymbol{X}_\infty')^{-1}\mathrm{vec}(E\{\boldsymbol{Q}_\infty\boldsymbol{Q}_\infty^{\mathrm{T}}\})+\\&(\boldsymbol{I}_{N^2M^2}-\boldsymbol{X}_\infty')^{-1}\gamma'^2\mathrm{vec}(E\{\boldsymbol{H}'^{\mathrm{T}}\boldsymbol{V}'\boldsymbol{V}'^{\mathrm{T}}\boldsymbol{H}'\})\end{aligned}\tag{53}$$

Using the matrix property $\mathrm{Tr}(\boldsymbol{XY})=[\mathrm{vec}(\boldsymbol{Y}^{\mathrm{T}})^{\mathrm{T}}]\,\mathrm{vec}(\boldsymbol{X})$, the steady-state mean square deviation (MSD) for all nodes in the multitask network is obtained

$$\begin{aligned}\mathrm{MSD}_\infty=&\frac{1}{|\boldsymbol{V}|}\mathrm{Tr}(E\{\tilde{\boldsymbol{w}}_\infty'\tilde{\boldsymbol{w}}_\infty'^{\mathrm{T}}\})=\\&\frac{1}{\boldsymbol{N}}[\mathrm{vec}(\boldsymbol{I}_{NM})]^{\mathrm{T}}\mathrm{vec}(E\left\{\tilde{\boldsymbol{w}}_\infty'\tilde{\boldsymbol{w}}_\infty'^{\mathrm{T}}\right\})=\\&\frac{1}{\boldsymbol{N}}[\mathrm{vec}(\boldsymbol{I}_{NM})]^{\mathrm{T}}(\boldsymbol{I}_{N^2M^2}-\boldsymbol{X}_\infty')^{-1}\times\\&[\mathrm{vec}(E\{\boldsymbol{Q}_\infty\boldsymbol{Q}_\infty^{\mathrm{T}}\})+\\&\gamma'^2\mathrm{vec}(E\{\boldsymbol{H}'^{\mathrm{T}}\boldsymbol{V}'\boldsymbol{V}'^{\mathrm{T}}\boldsymbol{H}'\})]\end{aligned}\tag{54}$$

where

$$\begin{aligned}&\mathrm{vec}(E\{\boldsymbol{Q}_i\boldsymbol{Q}_i^{\mathrm{T}}\})+\gamma'^2\mathrm{vec}(E\{\boldsymbol{H}'^{\mathrm{T}}\boldsymbol{V}'\boldsymbol{V}'^{\mathrm{T}}\boldsymbol{H}'\})=\\&\mathrm{vec}(\boldsymbol{M}'E\{\overline{\boldsymbol{V}}_i\}\boldsymbol{R}'(\boldsymbol{F}_i'-\boldsymbol{O}_i'E\{\boldsymbol{w}_k^o\}\times\\&(\boldsymbol{F}_i'-\boldsymbol{O}_i'E\{\boldsymbol{w}_k^o\})^{\mathrm{T}}\boldsymbol{R}'^{\mathrm{T}}E\{\overline{\boldsymbol{V}}_i^{\mathrm{T}}\}\boldsymbol{M}'^{\mathrm{T}})+\\&\gamma'^2\mathrm{vec}(E\{\boldsymbol{H}'^{\mathrm{T}}\boldsymbol{V}'\boldsymbol{V}'^{\mathrm{T}}\boldsymbol{H}'\})=\\&\mathrm{vec}(\boldsymbol{M}'E\{\overline{\boldsymbol{V}}_i\}\boldsymbol{R}'\boldsymbol{G}'\boldsymbol{R}'^{\mathrm{T}}E\{\overline{\boldsymbol{V}}_i^{\mathrm{T}}\}\boldsymbol{M}'^{\mathrm{T}}+\gamma'^2\boldsymbol{V}_r')\end{aligned}\tag{55}$$

The formulae for $\boldsymbol{V}_r'$ and $\boldsymbol{G}'$ are expressed as

$$\begin{aligned}&\boldsymbol{V}_r'=E\left\{\boldsymbol{H}'^{\mathrm{T}}\boldsymbol{V}'\boldsymbol{V}'^{\mathrm{T}}\boldsymbol{H}'\right\}=\\&\mathrm{diag}\left\{\sum_{l\in N_1}a_{l1,i}^2\boldsymbol{R}_{\eta,1},\sum_{l\in N_2}a_{l2,i}^2\boldsymbol{R}_{\eta,2},\cdots,\sum_{l\in N_N}a_{lN,i}^2\boldsymbol{R}_{\eta,N}\right\}\end{aligned}\tag{56}$$

$$
\begin{aligned}
&\boldsymbol{G}' = (\boldsymbol{F}_i' - \boldsymbol{O}_i' E\{\boldsymbol{w}_k^o\})(\boldsymbol{F}_i' - \boldsymbol{O}_i' E\{\boldsymbol{w}_k^o\})^{\mathrm{T}} = \\
&\operatorname{diag}\left\{\frac{\sigma_{v,k}^2(\boldsymbol{R}_{u,k}+\sigma_{o,k}^2\boldsymbol{I}_M)}{\|\boldsymbol{u}_{ok,i}\|_2^2} + \frac{E\{\boldsymbol{u}_{k,i}\boldsymbol{o}_{k,i}^{\mathrm{T}}\boldsymbol{w}_k^o\boldsymbol{w}_k^{o\mathrm{T}}\boldsymbol{o}_{k,i}\boldsymbol{u}_{k,i}^{\mathrm{T}}\}}{\|\boldsymbol{u}_{ok,i}\|_2^2} + \right. \\
&\left. \frac{E\{\boldsymbol{o}_{k,i}\boldsymbol{o}_{k,i}^{\mathrm{T}}\boldsymbol{w}_k^o\boldsymbol{w}_k^{o\mathrm{T}}\boldsymbol{o}_{k,i}\boldsymbol{o}_{k,i}^{\mathrm{T}}\}}{\|\boldsymbol{u}_{ok,i}\|_2^2} - \frac{\sigma_{o,k}^4\boldsymbol{w}_k^o\boldsymbol{w}_k^{o\mathrm{T}}}{\|\boldsymbol{u}_{ok,i}\|_2^2}\right\} = \\
&\operatorname{diag}\left\{\frac{\sigma_{v,k}^2(\boldsymbol{R}_{u,k}+\sigma_{o,k}^2\boldsymbol{I}_M)}{\|\boldsymbol{u}_{ok,i}\|_2^2} + \frac{\sigma_{o,k}^2\operatorname{Tr}(\boldsymbol{R}_{u,k})\boldsymbol{w}_k^o\boldsymbol{w}_k^{o\mathrm{T}}}{\|\boldsymbol{u}_{ok,i}\|_2^2} + \right. \\
&\left. \frac{(1+\boldsymbol{M})\sigma_{o,k}^4\boldsymbol{w}_k^o\boldsymbol{w}_k^{o\mathrm{T}}}{\|\boldsymbol{u}_{ok,i}\|_2^2} - \frac{\sigma_{o,k}^4\boldsymbol{w}_k^o\boldsymbol{w}_k^{o\mathrm{T}}}{\|\boldsymbol{u}_{ok,i}\|_2^2}\right\}
\end{aligned} \tag{57}
$$

5 The Low Computational Cost RM-DABC-LMS Algorithm

In wireless sensor networks (WSNs), a key challenge is the high communication cost between nodes. To mitigate this and reduce overall communication costs, we propose a low-communication-cost RM-DABC-LMS algorithm that leverages a compressed diffusion strategy.

To reduce communication cost in [36, 37], the local estimates are compressed to $\boldsymbol{s}_{lk,i} = \boldsymbol{C}_{k,i}^{\mathrm{T}} \boldsymbol{\varphi}_{lk,i}$, then diffusion to neighbouring nodes, where $\boldsymbol{C}_{k,i}^{\mathrm{T}}$ is a randomized projection vector. However, this compressive diffusion strategy does not account for communication link noise. Here, the compressed information received by node i from neighbouring node k is represented as follows

$$\boldsymbol{z}_{lk,i} = \boldsymbol{s}_{lk,i} + \boldsymbol{\eta}_{lk,i} \tag{58}$$

The objective is to estimate $\{\boldsymbol{\psi}_{k,i}, l \in N_k/\{k\}\}$ of the local weight vectors $\{\boldsymbol{\varphi}_k(t), l \in N_k/\{k\}\}$ from the noisy compressed signals $\{\boldsymbol{z}_{k,i}, l \in N_k/\{k\}\}$ at node i.

Solving the cost function $\min\limits_{\tilde{\psi}_k} \|\boldsymbol{s}_{lk} - \boldsymbol{C}_i^{\mathrm{T}}\boldsymbol{\psi}_k\|$ with the stochastic gradient descent method, the following update equation is obtained

$$\tilde{\boldsymbol{\psi}}_{lk,i} = \tilde{\boldsymbol{\psi}}_{lk,i-1} + \varsigma_k \frac{\boldsymbol{C}_{k,i}}{\|\boldsymbol{C}_{k,i}\|^2} e_{\tilde{\boldsymbol{\psi}}_{k,i}} \tag{59}$$

where $\varsigma_k > 0$ represents the stride. The error signal is denoted by $e_{\tilde{\boldsymbol{\psi}}_{k,i}} = \delta_{lk,i} - \boldsymbol{C}_{k,i}^{\mathrm{T}} \tilde{\boldsymbol{\psi}}_{lk,i-1}$.

As a result, at node k the combination step (19) becomes

$$\boldsymbol{w}_{k,i} = a_{k,i}\boldsymbol{\varphi}_{k,i} + \sum_{l \in N_k/\{k\}} a_{lk,i}\tilde{\boldsymbol{\psi}}_{k,i} \tag{60}$$

6 Simulation Results

To evaluate the performance of the proposed RM-DABC-LMS algorithm, a multitask network with $N = 16$ nodes was used, as shown in Fig. 2(a). The filter length was set to 64. The input noise-free regression model and the input noise followed zero-mean Gaussian white processes with variances of 1.0 and 0.1, respectively, as illustrated in Fig. 2(b).

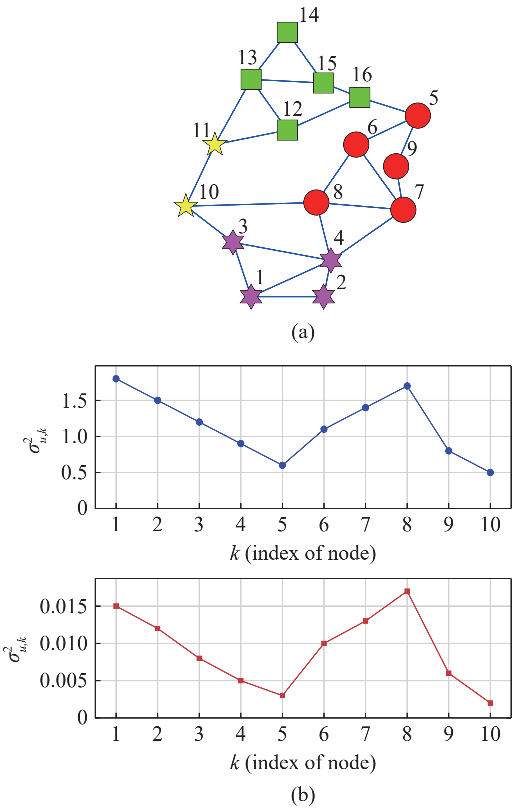

Fig. 2 Emulation profiles: (a) multitasking network topology; (b) input variances (top) and noise variances (bottom)

Impulse noise was modelled as $\eta(n) = p(n)A(n)$, where $p(n)$ was a Bernoulli process with probabilities $P(p(n)=1) = \mathrm{Pr}$, $P(p(n)=0) = 1 - \mathrm{Pr}$. The process $A(n)$ was a zero-mean white Gaussian process with variance $\sigma_A^2 = k'\sigma_y^2$, and σ_y^2 represented the variance of the output signal in the unperturbed state. The proportionality factor k' determines the pulse noise intensity relative to the original signal variance. The performance metric used was MSD, and all experimental results were averaged over 100 independent runs. During simulation, the pulse noise occurrence probability was set to $\Pr = 0.01$, and all parameter vectors were initialized as zero vectors.

In the first experiment, MSD simulation curves were plotted to compare the RM-DABC-LMS algorithm with several alternatives, as shown in Fig. 3. The algorithms compared under impulse noise conditions include the RM-DABC-LMS, M-DABC-LMS, M-DABC-LMS (no impulse noise), M-DMABC-LMS and DLMS algorithms ($\mu_k = 0.003, 0.001, 0.002, 0.0035$).

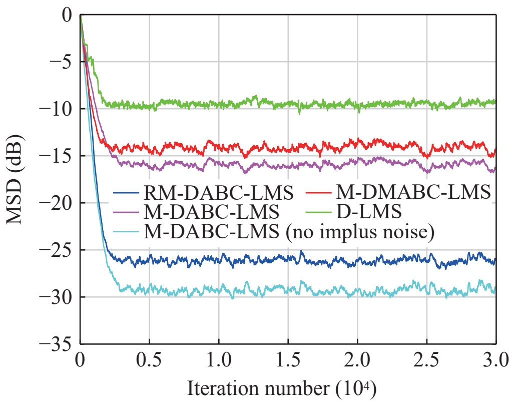

Fig. 3 Performance comparison of the RM-DABC-LMS algorithm with different algorithms in a pulsed noise environment

By adjusting the step size parameter, the steady-state error of these algorithms was compared at identical convergence rates. As seen in Fig. 3, under pulse noise, the M-DABC-LMS algorithm exhibits significantly higher steady-state error than the RM-DABC-LMS algorithm, resulting in a marked performance deterioration. Additionally, in multitask network environments, the RM-DABC-LMS algorithm based on normalized MVC shows lower steady-state error compared to other approaches. Overall, the RM-DABC-LMS algorithm demonstrates robust performance under impulse noise, effectively reducing steady-state error while maintaining high estimation accuracy.

Fig. 4 shows the performance curves of the RM-DABC-LMS algorithm under different parameters. The simulation results for various step sizes ($\mu_k = 0.003, 0.01, 0.02, 0.04$) reveal that as the step size increases, the algorithm converges faster but with higher steady-state error. This highlights the trade-off between rapid convergence and low steady-state error, emphasizing the significant impact of step size on both convergence rate and steady-state performance. Additionally, Fig. 4 presents MSD curves of the RM-DABC-LMS algorithm for different link noise variances. At the same convergence rates, the steady-state MSD increases as the link noise variance rises.

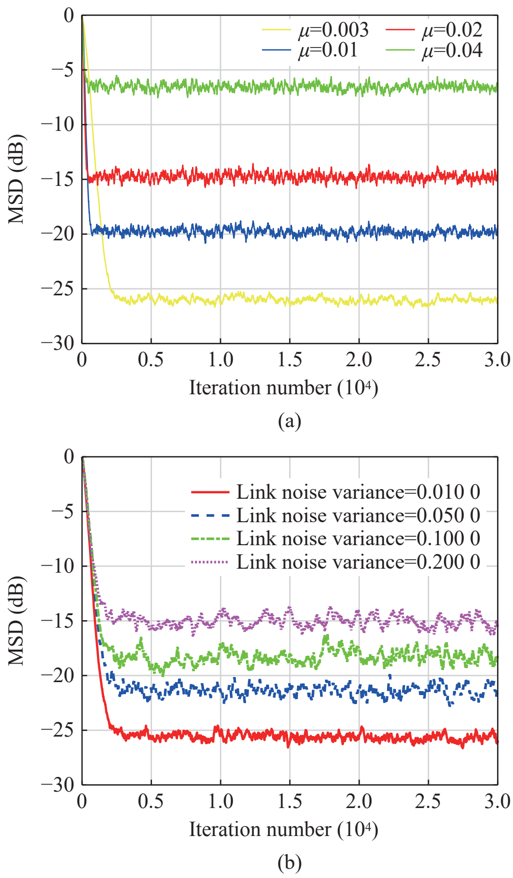

Fig. 4 MSD simulation curves of the RM-DABC-LMS algorithm: (a) different step sizes; (b) different link-noise variances

Simulation results demonstrate that while the RM-DABC-LMS algorithm can mitigate noise effects to some extent, increased link noise reduces its performance. Therefore, practical applications must balance algorithm robustness with communication link noise levels to optimize performance.

In the third experiment, the network MSD behavior of the RM-DABC-LMS algorithm is shown in Fig. 5. As seen in Fig. 5(a), the simulated learning curves align closely with theoretical results for various step sizes μ_k, confirming the accuracy of the theoretical predictions. Additionally, as illustrated in Fig. 5(b), the simulated curves converge to the theoretical steady-state MSD curve with different $\sigma^2_{\eta,k}$ values. The results indicate that the algorithm achieves the expected estimation performance under stable conditions.

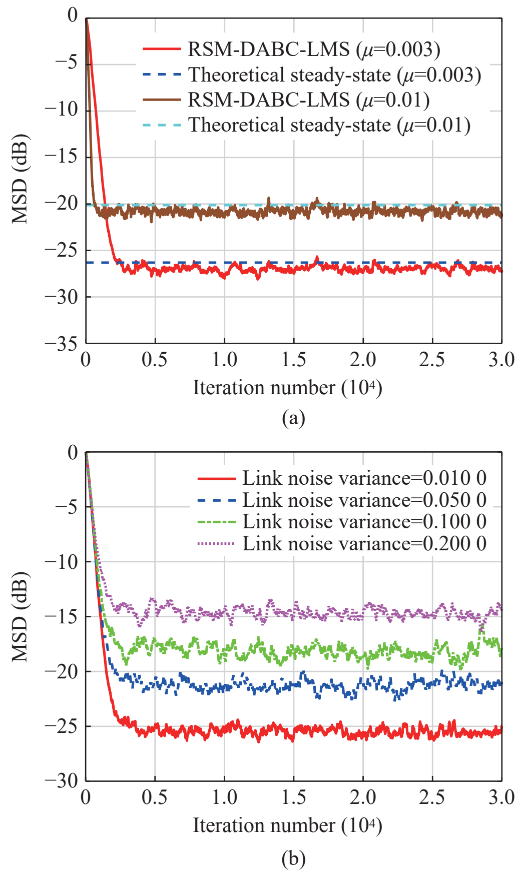

Fig. 5 MSD simulation curves of the RM-DABC-LMS algorithm: (a) step sizes; (b) link noise variances

Fig. 6 shows the MSD simulation curve for the low communication cost RM-DABC-LMS algorithm. The results demonstrate that using a compressed diffusion strategy significantly reduces communication costs while maintaining the same steady-state error as the standard RM-DABC-LMS algorithm, effectively balancing communication efficiency and performance.

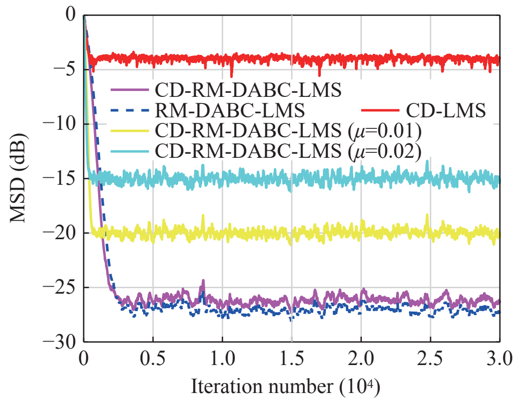

Fig. 6 Performance comparison of the low-communication-overhead RM-DABC-LMS algorithm with different algorithms under various step size conditions

As the step size μ_k increases, the low communication overhead algorithm converges more quickly but results in higher steady-state errors. Large step sizes lead to rapid convergence but higher errors, while smaller step sizes converge slowly but with lower errors. By carefully adjusting the step size, optimal estimation performance can be achieved.

Fig. 7 shows how MSD changes with the number of iterations for different compression ratios. As iterations increase, the MSD for each ratio decreases before stabilizing. Higher compression ratios result in smaller reductions in MSD. Notably, at a compression ratio of 1.14, the MSD approaches that of the uncompressed case. This highlights a balance between commu-

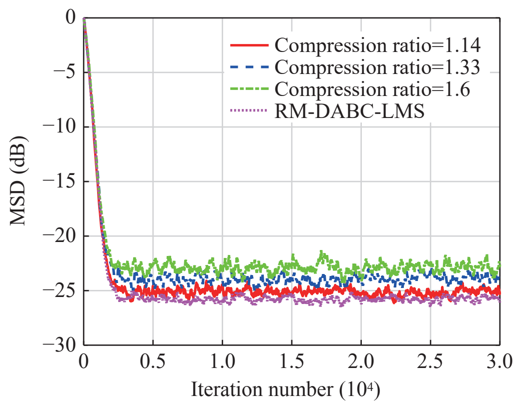

Fig. 7 Comparison of different compression ratios

nication efficiency and algorithm performance, reducing communication load while maintaining good performance.

7 Conclusion

This paper introduces a robust RM-DABC-LMS algorithm. By incorporating bias compensation and sliding-average denoising mechanisms, the proposed RM-DABC-LMS algorithm achieves improved robustness and steady-state performance in noisy distributed environments. In addition, the compressed diffusion strategy effectively reduces the network communication load while maintaining satisfactory estimation accuracy. Both theoretical analysis and simulation results confirm the effectiveness of the proposed algorithms in terms of convergence behavior, estimation accuracy, and communication efficiency.

References:

[1] A. H. Sayed,"Adaptation, learning, and optimization over networks," *Foundations and Trends in Machine Learning*, vol. 7, no. 4-5, pp. 311-801, 2014.

[2] A. H. Sayed,"Adaptive networks," *Proceedings of the IEEE*, vol. 102, no. 4, pp. 460-497, 2014.

[3] G. Wen, W. Yu, M. Z. Q. Chen, X. Yu, and G. Chen, "Pinning a complex network to follow a target system with predesigned control inputs, " *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 6, pp. 2293-2304, 2020.

[4] P. Wen and J. Zhang,"Widely linear complex-valued diffusion subband adaptive filter algorithm," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 5, no. 2, pp. 248-257, 2019.

[5] S. A. Ghauri and M. F. Sohail, "System identification using LMS, NLMS and RLS, " in *2013 IEEE Student Conference on Research and Developement*, Putrajaya, Malaysia, pp. 65-69, 2013.

[6] Y. Song, B. Wang, Z. Shi, K. R. Pattipati, and S. Gupta,"Distributed algorithms for energy-efficient even self-deployment in mobile sensor networks," *IEEE Transactions on Mobile Computing*, vol. 13, no. 5, pp. 1035-1047, 2013.

[7] D. Yu, X. Shi, L. Chai, W. A. Zhang, and J. Chen, "Balancing localization accuracy and location privacy in mobile cooperative localization, " *IEEE Transactions on Signal Processing*, vol. 71, pp. 2804-2818, 2023.

[8] R. Olfati-Saber and R. M. Murray,"Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520-1533, 2004.

[9] S. Barbarossa and G. Scutari, "Bio-inspired sensor network design, " *IEEE Signal Processing Magazine*, vol. 24, no. 3, pp. 26-35, 2007.

[10] C. G. Lopes and A. H. Sayed, "Incremental adaptive strategies over distributed networks, " *IEEE Transactions on Signal Processing*, vol. 55, no. 8, pp. 4064-4077, 2007.

[11] F. S. Cattivelli and A. H. Sayed,"Diffusion LMS strategies for distributed estimation," *IEEE Transactions on Signal Processing*, vol. 58, no. 3, pp. 1035-1048, 2009.

[12] F. Chen, T. Shi, S. Duan, L. Wang, and J. Wu,"Diffusion least logarithmic absolute difference algorithm for distributed estimation," *Signal Processing*, vol. 142, pp. 423-430, 2018.

[13] Y. Peng, S. Zhang, Z. Zhou, H. Chen, and A. H. Sayed,"Diffusion moving-average adaptation over networks," *IEEE Transactions on Signal Processing*, vol. 72, pp. 3393-3407, 2024.

[14] S. Zhang, J. Zhang, and W. X. Zheng,"Auxiliary diffusion strategy against link noises over distributed networks," *IEEE Transactions on Signal Processing*, vol. 72, pp. 115-128, 2024.

[15] F. S. Cattivelli and A. H. Sayed,"Diffusion LMS strategies for distributed estimation," *IEEE Transactions on Signal Processing*, vol. 58, no. 3, pp. 1035-1048, 2010.

[16] F. S. Cattivelli, C. G. Lopes, and A. H. Sayed,"Diffusion recursive least-squares for distributed estimation over adaptive networks," *IEEE Transactions on Signal Processing*, vol. 56, no. 5, pp. 1865-1877, 2008.

[17] R. Nassif, S. Vlaski, C. Richard, J. Chen, and A. H. Sayed,"Multitask learning over graphs: An approach for distributed, streaming machine learning," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 14-25, 2020.

[18] S. Ozawa, A. Roy, and D. Roussinov,"A multitask learning model for online pattern recognition," *IEEE Transactions on Neural Networks*, vol. 20, no. 3, pp. 430-445, 2009.

[19] P. Song, H. Zhao, and Y. Zhu,"Robust multitask diffusion affine projection algorithm for distributed estimation," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 69, no. 3, pp. 1892, 1896.

[20] A. Danaee, R. C. D. Lamare, and V. H. Nascimento, "Distributed quantization-aware RLS learning with bias compensation and coarsely quantized signals," *IEEE Transactions on Signal Processing*, vol. 70, pp. 3441-3455, 2022.

[21] R. Arablouei, Y. F. Huang, S. Werner, M. Reza, and K. Doğançay,"Reduced-communication diffusion LMS strategy for adaptive distributed estimation," *Signal Processing*, vol. 117, pp. 355-361, 2014.

[22] F. Huang, F. Song, S. Zhang, H. C. So, and J. Yang, "Robust bias-compensated LMS algorithm: Design, performance analysis and applications," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 10, pp. 13214-13228, 2014.

[23] R. M. Pimenta, M. R. Petraglia, and D. B. Haddad,"Stability analysis of the bias compensated LMS algorithm," *Digital Signal Processing*, vol. 147, pp. 104395, 2024.

[24] J. Lou, L. Jia, R. Tao, and Y. Wang,"Distributed incremental bias-compensated RLS estimation over multi-agent networks," *Science China Information Sciences*, vol. 60, no. 3, pp. 32204, 2017.

[25] T. Tang, L. Jia, J. Lou, R. Tao, and Y. Wang, "Adaptive EIV-FIR filtering against coloured output noise by using linear prediction technique," *IET Signal Processing*, vol. 12, no. 1, pp. 104-112, 2018.

[26] L. Jia, C. Zheng, A. Katerega, and Z. J. Yang, "Distributed diffusion bias-compensated LMS for node-specific networks," *Signal Processing*, vol. 160, pp. 21-31, 2019.

[27] R. Nassif, C. Richard, J. Chen, A. Ferrari, and A. H. Sayed, "Diffusion LMS over multitask networks with noisy links, " in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Shanghai, China, pp. 4583-4587, 2016.

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

[29] W. Xia and Y. Wang,"A variable step-size diffusion LMS algorithm over networks with noisy links," *Signal Processing*, vol. 148, pp. 205-213, 2018.

[30] L. Wang, L. Jia, D. Miao, Y. Guo, and S. Kanae, "A novel bias-compensated linear constrained least mean squares algorithm over distributed network, " in *2023 42nd Chinese Control Conference (CCC)*, Tianjin, China, pp. 3332-3337, 2023.

[31] P. Wen, Y. Wang, S. Zhang, and F. Huang, "Secure multitask distributed adaptive estimate with noisy observation and byzantine attacks," *Digital Signal Processing*, vol. 169, pp. 105715, 2025.

[32] G. Azarnia, R. Mohammadloo, and M. A. Tinati, "Regular combining of cooperative and non-cooperative schemes for mitigation of the effect of noisy links in diffusion LMS adaptive networks, " in *2014 22nd Iranian Conference on Electrical Engineering (ICEE)*, Tehran, Iran, pp. 1395-1400, 2014.

[33] J. W. Lee, S. E. Kim, W. J. Song, and A. H. Sayed, "Spatio-temporal diffusion strategies for estimation and detection over networks, " *IEEE Transactions on Signal Processing*, vol. 60, no. 8, pp. 4017-4034, 2012.

[34] F. Huang, J. Zhang, and S. Zhang,"Maximum versoria criterion-based robust adaptive filtering algorithm," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 64, no. 10, pp. 1252-1256, 2017.

[35] J. Chen, C. Richard, and A. H. Sayed,"Diffusion LMS over multitask networks," *IEEE Transactions on Signal Processing*, vol. 63, no. 11, pp. 2733-2748, 2015.

[36] R. Zhang, X. Y. Zhang, S. H. Zhou, and X. J. Peng,"Distributed radar target tracking with low communication cost," *Journal of Beijing Institute of Technology*, vol. 31, no. 6, pp. 595-604, 2022.

[37] M. O. Sayin and S. S. Kozat,"Compressive diffusion strategies over distributed networks for reduced communication load," *IEEE Transactions on Signal Processing*, vol. 62, no. 20, pp. 5308-5323, 2014.

Xuzhao Chai received the Ph.D. degree in microelectronics and solid-state electronics from Nanjing University, China, in 2015. He is currently an Associate Professor at School of Automation and Electrical Engineering, Zhongyuan University of Technology, Zhengzhou, China. His research interests include intelligent optimization techniques in complex systems such as unmanned aerial vehicles, photovoltaic systems and electric circuits.

Linru Zhang is studying at School of Automation and Electrical Engineering, Zhongyuan University of Technology, Zhengzhou, China, with a research focus on distributed networks.

Pengwei Wen (Member, IEEE) received the B.S. degree from College of Mathematics and Information Sciences, North China University of Water Resources and Electric Power, Zhengzhou, China, in 2014, and the Ph.D. degree in information and communication engineering from Southwest Jiaotong University, Chengdu, China, in 2019. Since 2019, he has been working with School of Electronic Information, Zhongyuan University of Technology, Zhengzhou, China, where he is currently an Associate Professor. His research interests include adaptive signal processing, distributed adaptation and learning theories.

Sheng Zhang (Senior Member, IEEE) received the B.S. degree from College of Mathematics and Information Sciences, North China University of Water Resources and Electric Power, Zhengzhou, China, in 2010, and the Ph.D. degree in signal and information processing from Southwest Jiaotong University, Chengdu, China, in 2016. From 2016 to 2017, he was a Research Fellow with School of Computing, Engineering and Mathematics, Western Sydney University, Sydney, NSW, Australia. Since 2017, he has been working at School of Information Science and Technology, Southwest Jiaotong University, Chengdu, China, where he is currently an Associate Professor. His research interests are in distributed adaptation and learning theories. He is currently serving as an Associate Editor for the {\it Circuits, Systems, and Signal Processing}. His homepage is at https://faculty.swjtu.edu.cn/ShengZHANG/en/.

Haiquan Zhao (Senior Member, IEEE) was born in Henan Province, China, in 1974. He received the B.S. degree in applied mathematics, and the M.S. and Ph.D. degrees in signal and information processing from Southwest Jiaotong University, Chengdu, China, in 1998, 2005, and 2011, respectively. Since 2012, he has been a Professor with School of Electrical Engineering, Southwest Jiaotong University, Chengdu, China. From 2015 to 2016, as a Visiting Scholar, he worked with the University of Florida, USA. His current research interests include information theoretical learning, adaptive filter, adaptive network, active noise control, Kalman filter, machine learning, and artificial intelligence. He is the author or coauthor of more than 250 international journal papers (SCI indexed), and the owner of 70 invention patents. He has won several provincial and ministerial awards, and many best paper awards of international conferences or IEEE Transactions. He has served as an Active Reviewer for several IEEE transactions, IET series, signal processing, and other international journals. At present, he is a Handling Editor of Signal Processing, and also an Associate Editor of IEEE Transactions on Systems, Man and Cybernetics: System, IEEE Signal Processing Letters, IEEE Sensors Journal and IEEE Open Journal of Signal Processing.