Circuits, Systems, and Signal Processing (2024) 43:3832–3860
https://doi.org/10.1007/s00034-024-02634-0

A Convex Combination Least Mean Square Algorithm Based on the Distributed Diffusion Strategy for Sensor Networks

Teng Feng[1,2,3,4] · Shuwei Deng[1,2,3] · Yao Mao[1,2,3,4]

Received: 21 March 2023 / Revised: 6 February 2024 / Accepted: 7 February 2024 /
Published online: 16 March 2024

Abstract

In this paper, we propose a novel convex combination algorithm based on the distributed diffusion strategy, utilizing the least mean square (LMS) approach to enhance the performance of sensor networks. By enabling communication among nodes, the algorithm achieves decentralization and improves the robustness of the network. To address the limitations of slow convergence speed and large static error associated with fixed step size errors, we introduce a convex combination strategy that integrates two filters with variable step sizes. The LMS algorithm with the convex combination variable step size assigns a higher weight to the filter with a larger step size when the error is significant, ensuring rapid convergence. Conversely, the convex combination small step size filter is assigned a higher weight when the error is small, reducing the error in the stable state. The convergence behavior of the proposed algorithm is analyzed through theoretical analysis, and its complexity is compared with that of the distributed diffusion LMS algorithm through extensive experimental simulations. The results demonstrate that the combination of the LMS algorithm with the distributed diffusion strategy offers advantages in challenging external environments, including improved convergence speed and reduced stability error. This study makes significant contributions to the existing research field by introducing a novel convex combination algorithm that addresses the shortcomings of fixed step size errors and expands the

✉ Yao Mao
maoyao@ioe.ac.cn

Teng Feng
fengteng22@mails.ucas.ac.cn

Shuwei Deng
769766044@qq.com

[1] National Key Laboratory of Optical Field Manipulation Science and Technology, Chinese Academy of Sciences, Chengdu 610209, China

[2] Key Laboratory of Beam Control, Chinese Academy of Sciences, Chengdu 610209, China

[3] The Institute of Optics and Electronics, Chinese Academy of Sciences, Chengdu 610209, China

[4] University of Chinese Academy of Sciences, Beijing 100039, China

range of available algorithms. By leveraging the distributed diffusion strategy, our approach enhances the robustness of sensor networks and achieves improved performance. The findings of this study provide valuable insights for researchers working on decentralized algorithms and highlight the potential of combining the LMS algorithm with the distributed diffusion strategy in challenging environmental conditions.

Keywords Distributed diffusion strategy · Convex combination · Sensor network

1 Introduction

Intelligent sensor networks are widely used in various fields, such as environmental detection, climate detection, and signal detection and control, due to their low cost, practicality, and network adaptability. However, the presence of various noises can significantly affect the accuracy of information processing in sensors, leading to estimation deviations. Active noise suppression techniques have gained significant attention [17].

Distributed adaptive filter algorithms based on adaptive networks, which utilize noise measurement data from nodes to estimate potential parameter vectors, have emerged as a research hotspot due to the advances in adaptive filter theory. The distributed least mean square (DLMS) algorithm was proposed [17]. However, the DLMS algorithm still suffers from the limitations of fixed step sizes in the LMS algorithm, as it fails to optimize the convergence speed and stability error simultaneously.

To address the limitations mentioned above and minimize the loss of network information, researchers have proposed improved algorithms utilizing impulse noise detection methods. For instance, the diffusion robust variable step size LMS (DRVSS-LMS) algorithm [14] was introduced. However, this algorithm requires prior knowledge of the output error, including its mean, variance, and distribution. Another approach, the diffusion sign error LMS (DSE-LMS) algorithm, was proposed in reference [21] based on the minimization of the mean absolute error (MAE) criterion. By designing the cost function using the L_1 norm of the error, this algorithm effectively eliminates the interference of impulse noise through the sign operation during the adaptive step. However, the DSE-LMS algorithm exhibits slow convergence speed and large steady-state error due to its adaptive update process relying solely on the error signal's sign. Additionally, the robust diffusion LMS (RDLMS) algorithm presented in reference [3] derives from minimizing the pseudo-Huber function, which approximates the Huber function and ensures a continuous derivative.

Several methods have been proposed for non-Gaussian noise scenarios. The FxLMP algorithm [16] effectively converges by utilizing the error P-norm. Hui introduced the FxlogLMS algorithm [32], which models the steady-state distribution of α as a logarithmic process and performs a logarithmic transformation of the collected impact noise. Hadi proposed an algorithm [36] that enhances the robustness of network nodes against impulse noise. However, these methods suffer from fixed step sizes, resulting in slow convergence and large steady-state error.

To achieve high convergence speed and small steady-state error, this study proposes a convex combination LMS algorithm based on the distributed diffusion strategy

(CCLMS-DDS). By introducing convex combination into the distributed algorithm, this approach effectively balances the convergence speed and steady-state error, addressing the limitations of existing algorithms.

1.1 Related Studies

The classification of nonlinear filters can be categorized into three main groups: function expansion filters, recursive filters, and bilinear filters. Function expansion filters, such as Volterra and functional link artificial neural network (FLANN) filters, are known for their low computational load and linear output-weights relationship [4, 7, 11, 22, 26, 29]. Among them, Volterra filters outperform FLANN filters in handling crossing nonlinearities. Several function expansion filters have been developed to improve noise control in nonlinear active noise control (NANC) systems, including the Chebyshev filter [7], exponential functional link network (EFLN) filter [22], general FLANN filter [26], and mirror Fourier nonlinear (EMFN) filter [4].

Inspired by function expansion filters, researchers have introduced recursive filters [5, 6, 12, 27, 37] and function expansion bilinear (FEB) filters [13, 15, 19, 28] to further enhance control performance. FEB filters have shown remarkable applicability in NANC systems [12], but their practical implementation is hindered by high computational load and the bounded input bounded output (BIBO) stability problem. On the other hand, recursive filters have gained considerable attention due to their superior complexity and clear theoretical mechanism. Notable examples of recursive filters include the recursive second-order Volterra (RSOV) filter [37], recursive FLANN (RFLANN) filter [27], recursive EMFN (REMFN) filter [5], and recursive EMFN filter with linear infinite impulse response (IIR) section (REMFNL) filter [12].

Several algorithms have been proposed to control noise based on different estimation and cost functions. The FxLMM algorithm utilizes the objective function of an M estimation function [9, 30], while a nonlinear active noise control algorithm employs a P-norm Wiener filter algorithm with a continuous logarithmic cost function [18]. Another approach introduces variable step sizes in an LMS algorithm using incremental and diffusion strategies [24]. Saeed et al. [23] propose variable step size methods based on combinations of nonlinear functions, such as sigmoid and exponential functions, to achieve adaptive step sizes in the iterative process. However, these methods suffer from deficiencies, such as sensitivity to environmental noise and algorithm deviation caused by significant step size variations.

To address the limitations of variable step size methods, researchers have proposed variable step size methods suitable for impulse noise environments [14, 31, 33, 35]. Huang et al. [14] design an adaptive iterative method for step sizes and impulse noise detection thresholds, preventing impulse interference through threshold judgment. Although these methods exhibit good performance in impulse noise environments, they are computationally intensive. Garcia et al. [1] introduce the concept of convex combination into adaptive filtering algorithms, leading to a class of adaptive algorithms for convex combination with promising results. Zhao et al. [38] propose an algorithm utilizing convex combination for noise control in neural networks. Yu et al. [34] modify thresholds using a distributed recursion scheme, improving algorithm performance in the presence of impulsive noise. Guo et al. [12] propose an improved recursive even mirror Fourier nonlinear filter for nonlinear active noise control (NANC) by employing a convex combination of feedforward and feedback subsections. However, these algorithms are primarily designed for central algorithms, resulting in low reliability and scalability in networked systems.

To address these limitations, we introduce the concept of convex combinations in distributed networks, combining the advantages of both approaches. Through efficient calculations, we achieve the desired effect of reducing the step size.

1.2 Contributions of this Study

The proposed CCLMS-DDS algorithm in this study consists of two stages: adapt and combine. In the adapt phase, individual nodes estimate parameters and update local estimates using measurements from neighboring nodes. In the combine stage, the distributed strategy assigns weight values based on adjacent nodes and combines them to generate a new weight estimate. Compared to the DLMS algorithm, the algorithm proposed in this study offers the following advantages:

1. In the adapt stage, a convex combination approach is introduced for parameter estimation. This allows for dynamic adjustment of the filter's step size based on the uncertain external noise characteristics. As a result, the algorithm achieves rapid convergence in the initial phase of operation and adapts to a smaller step size near the steady state, effectively reducing the steady-state error. Therefore, this algorithm successfully addresses the contradiction between fast convergence and low steady-state error, broadening its applicability.
2. In the combine stage, information from neighboring nodes is fused, resulting in higher accuracy in the final parameter estimation compared to a centralized algorithm. The theoretical proof provides the corresponding mathematical derivation supporting this claim.

1.3 Structure of this Study

This paper is structured as follows. Section 1 presents the introduction, providing an overview of the research. In Sect. 2, the problem formulation and relevant background are discussed. Section 3 introduces the CCLMS-DDS algorithm, outlining its key features and operation principles. The performance analysis is presented in Sect. 4, where the algorithm's convergence and stability properties are thoroughly examined. Section 5 presents the simulation experiments conducted to evaluate the algorithm's performance, followed by an analysis of the experimental results. Finally, Sect. 6 summarizes the study, highlighting the key findings and contributions.

2 Question Formulation and Relevant Context

2.1 LMS Algorithm

The LMS algorithm is a well-known adaptive filtering technique that utilizes input signals to dynamically adjust filter weights for signal processing purposes. By performing statistical analysis on the input and system-generated signals, the L-order filter aims to meet predefined standards and achieve desired signal processing effects.

Let $x(i)$ denote the input signal, and the system's output signal $y(i)$ is obtained by combining weighted versions of past input signals, as shown in Eq. (1):

$$y(i) = \sum^{L} \omega_m x(i-m). \tag{1}$$

The error signal $e(i)$ is defined as the difference between the desired output signal $d(i)$ and the system's actual output signal $y(i)$, as given by Eq. (2):

$$e(i) = d(i) - y(i) = d(i) - \sum^{L} \omega_m x(i-m). \tag{2}$$

To update the filter's weight vector, Eq. (3) is employed:

$$\omega(i+1) = \omega(i) + 2\mu e(i) x(i), \tag{3}$$

where μ represents the step size of the filter.

2.2 Sensor Network Model

In this section, we present the sensor network model used in this study. The sensor network consists of M nodes, denoted as $\delta = \{1, \ldots, M\}$, which are deployed across observation areas with computing, communicating, and sensing capabilities.

To enable direct communication between nodes, a communication range distance of π_r is assumed. The network's structure is described by a graph $\kappa = (\delta, \zeta)$, where $\zeta \subset \delta \times \delta$ represents the edge set of the network. If two nodes p and q satisfy the condition $(p, q) \in \kappa$, they are considered interconnected.

The neighboring nodes of node p, denoted as H_p, are defined as the set of nodes connected directly to node p, as shown in Eq. (4):

$$H_p = \{q \in \delta \mid (p, q) \in \zeta\}. \tag{4}$$

It is worth noting that node p is also considered its own neighbor. Each node is aware of its immediate neighbors but does not possess knowledge about the complete communication structure of the entire network. Assuming interconnectivity among the networks, there exists a sequence of boundaries between any two nodes p and q, denoted as $(p, u_1), (u_1, u_2), \ldots, (u_m, q)$ in ζ.

In summary, this section outlines the sensor network model employed in the study, emphasizing the communication range, network structure represented by a graph, and the concept of neighboring nodes.

2.3 Diffusion Strategy

In the context of distributed estimation, various strategies are employed to tackle network parameter estimation problems. Among them, consensus, incremental, and diffusion strategies have gained significant attention. In this section, we focus on the diffusion strategy and its relevance to the research topic.

The incremental distribution system, discussed in previous studies [17], relies on a network ring topology to establish the sequence of estimates for each node, update their estimation tasks, and evaluate algorithm efficiency through theoretical analysis.

Schizas proposed a novel algorithm based on the consensus strategy, and its implementation process was thoroughly analyzed [20]. Mateos conducted a stability analysis of the consensus algorithm and provided theoretical confirmation of its effectiveness [25]. While both the incremental and consensus strategies have their merits, the diffusion strategy stands out due to its flexibility in network structure.

In line with the diffusion strategy, Sayed introduced the DLMS algorithm [8]. This algorithm has been specifically designed to leverage the advantages of the diffusion strategy for network parameter estimation.

In summary, this section highlights the diffusion strategy as a key component of distributed estimation. It provides an overview of the incremental and consensus strategies, leading to the introduction of the DLMS algorithm based on the diffusion strategy.

3 A Convex Combination LMS Algorithm Based on the Distributed Diffusion Strategy

The LMS algorithm, known for its simplicity and fast convergence speed, exhibits favorable performance in Gaussian distributions. However, when subjected to non-Gaussian white noise, the algorithm's adjustment accuracy is affected, and conventional error-based second-order moment algorithms fail to achieve proper convergence. This leads to a significant degradation or even invalidation of the LMS algorithm's performance. Existing alternative algorithms often suffer from either excessive computation or unsatisfactory parameter selection, further complicating the issue.

To address these limitations, this study proposes an innovative approach that leverages the concept of convex combinations. By combining filters with large and small step sizes based on weight ratios, the proposed algorithm continually adjusts its convergence speed and steady-state errors in response to the type of external noise. This effectively resolves the challenge of simultaneously achieving high convergence speed and low steady-state errors.

By dynamically adapting to the characteristics of non-Gaussian noise, the proposed CCLMS-DDS algorithm overcomes the limitations of traditional LMS algorithms. It offers a promising solution that balances the trade-off between convergence speed

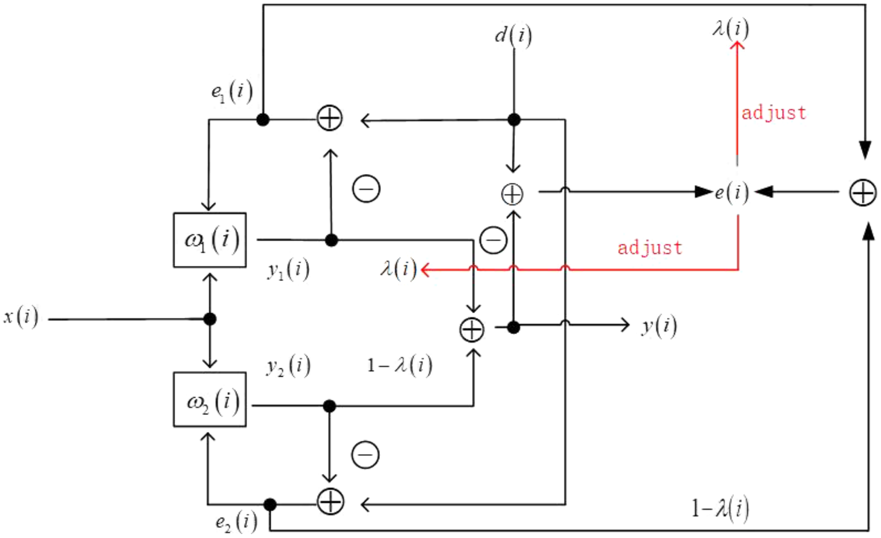


Fig. 1 The schematic diagram of the convex combination filter structure

and steady-state errors, thereby improving the algorithm's overall performance. The details of the algorithm formulation and its mathematical analysis will be presented in the following sections.

3.1 Convex Combination Adaptation Phase

The filter structure utilizing the convex combination is depicted in Fig. 1, where two filters are connected in parallel. The filter with a large step size is denoted as $\omega_1(i)$, while the filter with a small step size is represented by $\omega_2(i)$. The output signals from the two LMS filters are $y_1(i)$ and $y_2(i)$, respectively, and the final output value of the filter after the convex combination is denoted as $y(i)$. The input signal from the external source is $x(i)$, which passes through both filters. The desired signal is $d(i)$, and the error signal is calculated as $e(i) = d(i) - y(i)$. The weight coefficient $\lambda(i)$ controls the proportion of the two filters and adapts to the environment.

By employing the principle of convex combination, the output can be expressed as:

$$y(i) = \lambda(i)\, y_1(i) + [1 - \lambda(i)]\, y_2(i)\,. \tag{5}$$

Similarly, the output error can be obtained as:

$$e(i) = \lambda(i)\, e_1(i) + [1 - \lambda(i)]\, e_2(i)\,. \tag{6}$$

The core of the proposed algorithm lies in achieving a balance between fast convergence and minimal steady-state error. Consequently, the parameter $\lambda(i)$ needs to be variable. In the initial stage, the filter with the larger step size dominates, allowing for quick convergence. As the algorithm approaches stability and the steady-state error decreases, the parameter $\lambda(i)$ is adjusted to give priority to the filter with the smaller step size. Consequently, the overall algorithm error is reduced. The parameter $\lambda(i)$ gradually transitions from 1 to 0, following the sigmoid function:

$$\lambda(i) = sgm[\delta(i)] = \frac{1}{1 + e^{-\delta(i)}}, \tag{7}$$

Here, $\delta(i)$ is a dynamically adjusted variable determined by the error signal and the outputs of the filters. To minimize the mean square error [2], the stochastic gradient method of the LMS error is utilized for adjustment, and $\delta(i)$ is adaptively updated based on the error signal $e(i)$, as well as the output values $y_1(i)$ and $y_2(i)$ of the two filters. The recursive formula for $\delta(i)$ is given by:

$$\begin{aligned}\delta(i+1) &= \delta(i) - \frac{\mu_\alpha}{2}\frac{\partial e^2(i)}{\partial \delta(i)} &(8)\\ &= \delta(i) - \frac{\mu_\alpha}{2}\frac{\partial e^2(i)}{\partial \lambda(i)}\frac{\partial \lambda(i)}{\partial \delta(i)} &(9)\\ &= \delta(i) + \mu_\alpha e(i)[y_1(i) - y_2(i)]\lambda(i)[1 - \lambda(i)], &(10)\end{aligned}$$

where μ_α is a positive constant. To confine the mixed parameter $\lambda(i)$ within the convergent range of the algorithm, a sigmoidal function is employed. This effectively suppresses random gradient noise. The final filter weight update formula is described as follows:

$$\omega_j(i+1) = \omega_j(i) + \mu_j x(i) e_j(i), \tag{11}$$

where $j = 1, 2$.

By combining two single filters with different step sizes using the concept of a convex combination, a convex combination filter is designed. This approach enables the filter to achieve fast convergence with minimal steady-state error. The implementation steps are as follows:

$$\begin{aligned}&\mu_\alpha > 0 &(12)\\ &y_1(i) - y_2(i) = [\omega_1(i) - \omega_2(i)]x(i) > 0 &(13)\\ &\lambda(i)[1 - \lambda(i)] > 0 &(14)\end{aligned}$$

1. Based on these equations, the functional relationship between δ and e depicted in Fig. 2, and the monotonicity of variable δ being influenced solely by e, the algorithm initially increases δ and λ when

$$e(i) = d(i) - y(i) > 0. \tag{15}$$

This leads to an increased proportion of the filter with the large step size ($y_1(i)$), resulting in higher $y(i)$. Consequently, the filter with the large step size dominates, leading to a higher convergence speed.

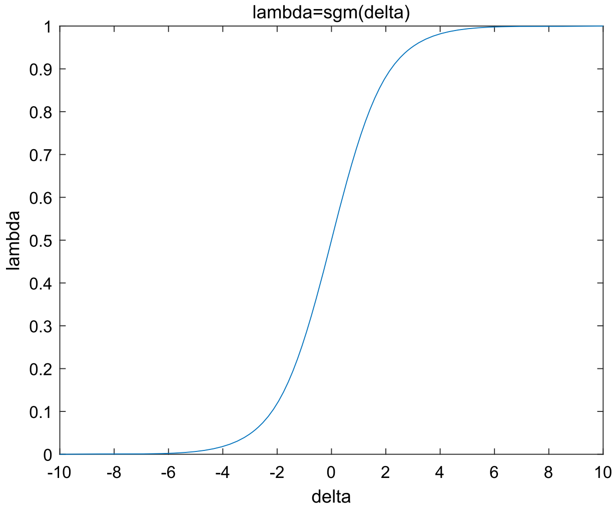


Fig. 2 The functional relationship between λ and δ

2. Conversely, when

$$e(i) = d(i) - y(i) < 0, \tag{16}$$

indicating that $y(i)$ is too large, e decreases according to the functional relationship in Fig. 2, resulting in a decrease in λ. Consequently, the proportion of the filter with the small step size ($y_1(i)$) increases, leading to a decrease in $y(i)$. In this case, the filter with the small step size dominates. After the algorithm converges for a certain period, the mixing parameter of the filter is automatically adjusted, increasing the proportion of the filter with the small step size. As a result, the overall filter's step size decreases, leading to a slower convergence speed and a smaller steady-state error. The combined convex filter achieves a balance between a small steady-state error and a high convergence speed.

Next, we consider the convex combination as a filter and derive the relationship between the overall filter weight update and the input. This relationship can be obtained from Eq. (5) as follows:

$$\begin{aligned} y_j(i) &= \omega_j(i)x(i), j = 1, 2, && (17)\\ y(i) &= \omega(i)x(i) && (18)\\ &= \lambda(i)y_1(i) + [1 - \lambda(i)]y_2(i) && (19)\\ &= \lambda(i)\omega_1(i)x(i) + [1 - \lambda(i)]\omega_2(i)x(i) && (20)\\ &= \lambda(i)\omega_1(i) + [1 - \lambda(i)]\omega_2(i)x(i), && (21) \end{aligned}$$

so we can derive

$$\omega(i) = \lambda(i)\,\omega_1(i) + [1 - \lambda(i)]\,\omega_2(i)\,; \tag{22}$$

by substituting (11) into (22), we can obtain the updated formula for $\omega(i+1)$.

$$\begin{aligned}\omega(i+1) &= \lambda(i)\,\omega_1(i+1) + [1-\lambda(i)]\,\omega_2(i+1) \qquad (23)\\ &= \lambda(i)\,[\omega_1(i) + \mu_1 e_1(i)\,x(i)] + [1-\lambda(i)]\,[\omega_2(i) + \mu_2 e_2(i)\,x(i)]. \qquad (24)\end{aligned}$$

After substituting (6), the following equation is obtained

$$\begin{aligned}\omega(i+1) &= \lambda(i)\,[\omega_1(i) + \mu_1 e_1(i)\,x(i)] + [1-\lambda(i)]\,[\omega_2(i) + \mu_2 e_2(i)\,x(i)] \qquad (25)\\ &= \lambda(i)\omega_1(i) + [1-\lambda(i)]\omega_2(i)\\ &\quad + \lambda(i)\mu_1 e_1(i)x(i) + [1-\lambda(i)]\mu_2 e_2(i)x(i) \qquad (26)\\ &= \omega(i) + \lambda(i)\mu_1 e_1(i)x(i) + [1-\lambda(i)]\mu_2 e_2(i)x(i) \qquad (27)\\ &= \omega(i) + \lambda(i)\mu_1 e_1(i)x(i) - \lambda(i)\mu_2 e_1(i)x(i)\\ &\quad + \lambda(i)\mu_2 e_1(i)x(i) + [1-\lambda(i)]\mu_2 e_2(i)x(i) \qquad (28)\\ &= \omega(i) + \lambda(i)\mu_1 e_1(i)x(i) - \lambda(i)\mu_2 e_1(i)x(i)\\ &\quad + \lambda(i)e_1(i) + [1-\lambda(i)]e_2(i)\mu_2 x(i) \qquad (29)\\ &= \omega(i) + \lambda(i)\mu_1 e_1(i)x(i) - \lambda(i)\mu_2 e_1(i)x(i) + \mu_2 e(i)x(i) \qquad (30)\\ &= \omega(i) + \left(\mu_2 + \frac{\lambda(i)\,(\mu_1-\mu_2)\,e_1(i)}{e(i)}\right) e(i)\,x(i). \qquad (31)\end{aligned}$$

Therefore, the final filter has a step size $\mu(i) = f(i) = \mu_2 + \frac{\lambda(i)(\mu_1-\mu_2)e_1(i)}{e(i)}$.

3.2 Combination Phase

Let us consider a sensor network consisting of N nodes, where each node $k \in \{1, 2, \ldots, N\}$ represents a different node in the network. At time i, the parameter value obtained by node k is denoted as $d_k(i)$, and there is an input vector $u_{k,i}$ for node k, equivalent to the input vector x in the LMS algorithm, where $u_{k,i}$ is an $L \times 1$ dimensional vector. Additionally, the accurate weight value of the network filter ω^0 is also represented as an $L \times 1$ dimensional vector. The input–output model for node k can be expressed as follows:

$$d_k(i) = u_{k,i}^T \omega^0 + \xi_k(i). \qquad (32)$$

Here, $\xi_k(i)$ represents Gaussian noise with a mean of 0 and variance $\sigma_{k,i}^2$. The main objective of distributed estimation is to use the external observations $\left(u_{k,i}, d_k(i)\right)$ to estimate ω^0 and approach its accurate value. Two strategies have been proposed for calculating the estimated value of ω^0: Adapt-Then-Combine (ATC) and Combine-Then-Adapt (CTA). The ATC strategy has been shown to provide better estimation accuracy compared to the CTA strategy [10]. (In order to substantiate the theoretical superiority of the Adaptive-Then-Combine (ATC) strategy over the Combine-Then-Adapt (CTA) strategy in handling performance within distributed networks, we have included additional experiments and provided corresponding simulation results in Sect. 5.3.) Therefore, this study directly utilizes and analyzes the ATC strategy.

The ATC strategy can be described as follows:

$$\varphi_{k,i} = \omega_{k,i-1} + \mu_k \sum u_{l,i}^{*} \left(d_l\left(i\right) - u_{l,i}\omega_{k,i-1}\right), \tag{33}$$

$$\omega_{k,i} = \sum a_{k,l}\varphi_{l,i}, \tag{34}$$

where $\mu_k = \mu(i) = \mu\left[e(i)\right] = f\left[e(i)\right] = \mu_2 + \frac{\lambda(\mu_1-\mu_2)e_1(i)}{e(i)}$.

Equation (33) represents the adaptation phase, where each node independently uses the observations $\left(u_{k,i}, d_k\left(i\right)\right)$ to calculate an intermediate estimate $\varphi_{k,i}$. Equation (34) represents the combination phase. Each node has the ability to communicate with its neighboring nodes and receive their parameter estimates. The node assigns weights to the received estimates and combines them to produce a new estimate. The coefficient $\left\{a_{k,l}\right\}$ determines the amount of intermediate estimate $\varphi_{k,i}$ exchanged between node k and its neighboring nodes $l \in N_k$. The weight coefficient $\left\{a_{k,l}\right\}$ should be greater than 0. Matrix A, with dimensions M, is used to select a subset of intermediate estimates for transmission. Assuming N-dimensional intermediate estimates need to be selected as a subset to be sent to neighboring nodes, the diagonal elements of matrix A consist of N ones, while the remaining $M-N$ elements are zeros. This allows neighboring nodes to choose the subset to be transmitted. If all diagonal elements are ones, it indicates that all neighboring nodes need to send their intermediate estimates, subject to the following conditions:

$$a_{k,l} = 0, l \notin N_k, 1^T A = 1^T. \tag{35}$$

The communication mode between nodes is determined by the weight coefficients, and the network topology of the wireless sensor network influences the size of the coefficient matrix. The weight coefficient matrix is generated using the Metropolis rule, as shown in Eq. (36):

$$a_{k,l} = \begin{cases} \frac{1}{max(r_k,r_l)}, l \in N_k, l \neq k \\ 1 - \sum\limits_{l \in N_k, l \neq k} c_{k,l}, l = k \\ 0, l \notin N_k \end{cases}, \tag{36}$$

where r_k and r_l are the degrees of nodes k and l, respectively. By utilizing the distributed strategy, effective decentralization can be achieved, and the computing tasks can be evenly distributed among each node, thereby improving estimation speed and reducing estimation errors caused by single-node estimation. Furthermore, in a distributed network, each node has computing power, so the overall network remains unaffected if a node fails, thereby enhancing network robustness.

Algorithm 1 A Convex Combination LMS Algorithm Based on the Distributed Diffusion Strategy

```
Initialization: $\omega_{l,0} = 0$, $c_{l,k} \geq 0$ and it satisfies the condition Eq. (36)
Parameters: $\mu_j (j = 1, 2)$, $\varphi$, $\delta$, $\mu_\alpha$
Adaptive process:
for i=1:iteration do
    for each node k
    Adaptation:
    $y_1(i) = \omega_1(i)x(i)\ e_1(i) = d(i) - y_1(i)$
    $y_2(i) = \omega_2(i)x(i)\ e_2(i) = d(i) - y_2(i)$
    $y\,(i) = \lambda\,(i)\,y_1\,(i) + [1 - \lambda\,(i)]\,y_2\,(i)$
    $e\,(i) = \lambda\,(i)\,e_1\,(i) + [1 - \lambda\,(i)]\,e_2\,(i)$
    $\lambda\,(i) = sgm\,[\delta\,(i)] = \frac{1}{1+e^{-\delta(i)}}$
    $\delta\,(i+1) = \delta\,(i) + \mu_\alpha e\,(i)\,[y_1\,(i) - y_2\,(i)]\,\lambda\,(i)\,[1 - \lambda\,(i)]$
    $\omega_j\,(i+1) = \omega_j\,(i) + \mu_j x\,(i)\,e_j\,(i)$
    $\varphi_{k,i} = \omega_{k,i-1} + \mu_k \sum u_{l,i}^{*}\left(d_l\,(i) - u_{l,i}\omega_{k,i-1}\right)$
    Communication:
    Transfer the intermediate estimate $\varphi_{k,i}$ to all neighbor nodes $N_k$
    $\varphi_{k,i} = \omega_{k,i-1} + \mu_k \sum u_{l,i}^{*}\left(d_l\,(i) - u_{l,i}\omega_{k,i-1}\right)$
    $\omega_{k,i} = \sum a_{k,l}\varphi_{l,i}$,
```

4 Performance Analysis

4.1 Convergence Analysis

In this section, we analyze the convergence of the weight coefficient vector in Eq. (3). By taking the expectation on both sides, we obtain the following equation:

$$E\left\{\omega(i+1)\right\} = E\left\{\omega(i)\right\} + 2\mu E\left\{e(i)x(i)\right\} = (I - 2\mu R_{xx})E\left\{\omega(i)\right\} + 2\mu R_{xd}, \quad (37)$$

where the identity matrix is denoted as I,

$$R_{xd} = E\left\{d(i)x(i)\right\}, R_{xx} = E\left\{x(i)x^T(i)\right\}. \quad (38)$$

By substituting $i = 0$, we obtain:

$$E\left\{\omega(1)\right\} = (I - 2\mu R_{xx})E\left\{\omega(0)\right\} + 2\mu R_{xd}, \quad (39)$$

Similarly, for $i = 1$, we have:

$$E\left\{\omega(2)\right\} = (I - 2\mu R_{xx})E\left\{\omega(1)\right\} + 2\mu R_{xd}, \quad (40)$$

Starting with $E\left\{\omega(0)\right\} = \omega(0)$, we can iterate the process and obtain:

$$E\left\{\omega(i+1)\right\} = (I - 2\mu Q\Lambda Q^{-1})^{i+1}\omega(0) + 2\mu\sum_{j=0}^{i}(I - 2\mu Q\Lambda Q^{-1})^j R_{xd}. \quad (41)$$

Here, R_{xx} can be decomposed into a quadratic form as

$$R_{xx} = Q\Lambda Q^T = Q\Lambda Q^{-1}, \tag{42}$$

where Q is an orthogonal unitary matrix and Λ represents the eigenmatrix of R_{xx}. λ denotes the eigenvalues of Λ.

Using the relationship with the identity matrix, we have:

$$(I - 2\mu Q\Lambda Q^{-1})^i = Q(I - 2\mu\Lambda)^i Q^{-1}, \tag{43}$$

$$\sum_{j=0}^{inf} (I - 2\mu Q\Lambda Q^{-1})^j = Q(2\mu\Lambda)^{-1} Q^{-1}. \tag{44}$$

If the diagonal elements of the matrix are less than 1 (achieved by controlling μ), we can observe that:

$$\lim_{i \longrightarrow \inf} (I - 2\mu\Lambda)^{i+1} = 0, \tag{45}$$

$$R_{xx}^{-1} = Q\Lambda^{-1}Q^{-1}. \tag{46}$$

By substituting the identities of Eqs. (43) and (44) into Eq. (41), we obtain:

$$E\left\{\omega(i+1)\right\} = Q\Lambda^{-1}Q^{-1}R_{xd} = R_{xx}^{-1}R_{xd} = \omega_{opt}. \tag{47}$$

As time progresses, the solution of the weight vector gradually converges to the Wiener solution, represented by ω_{opt}. The diagonal elements in $(I - 2\mu\Lambda)$ are less than 1, which ensures convergence:

$$0 < \mu < \frac{1}{\lambda_{max}}, \tag{48}$$

where λ_{max} is the maximum eigenvalue of R_{xx}. The step size μ plays a crucial role in determining the convergence speed.

Before proving the convergence of a distributed network, we need to establish the following assumptions:

1. The input vectors of all nodes $u_{k,i}$ are independently and uniformly distributed, with the property $E\left\{u_{k,i}u_{k,i}^T\right\} = R_{u,k}$.

2. The input Gaussian white noise $\xi_k(i)$ and the input $u_{k,i}$ of node k are independent, and the $\xi_k(i)$ of all nodes are also independent. We have the following relationships:

$$\begin{aligned} \Delta\omega_{k,i} &= \omega^0 - \omega_{k,i}, \\ \Delta\varphi_{k,i} &= \varphi^0 - \varphi_{k,i}. \end{aligned} \tag{49}$$

By substituting Eq. (49) into Eqs. (33) and (4), we obtain Eq. (50):

$$\begin{aligned} \Delta\varphi_{k,i} &= (I_M - \mu_k u_{k,i} u_{k,i}^T)\Delta\omega_{k,i-1} + \mu_k u_{k,i} v_{k,t}, \\ \Delta\omega_{k,i} &= \textstyle\sum_{l \in N_k, l \neq k} a_{l,k} \Delta\varphi_{l,i}. \end{aligned} \tag{50}$$

Next, we introduce the following diagonal matrix:

$$M = diag\left\{\mu_1 I_M, \ldots, \mu_N I_M\right\}. \tag{51}$$

We also introduce the following matrices:

$$D_i = blockdiag\left\{u^*_{l,i} u_{l,i}, \ldots, u^*_{N,i} u_{N,i}\right\}, \tag{52}$$

$$g_i = col\left\{u_{l,i} v_{l,i}, \ldots, u_{N,i} v_{N,i}\right\}, \tag{53}$$

$$\bar{G}_i = G_i \otimes I_M, \tag{54}$$

$$G_i = \begin{pmatrix} a_{1,1,i} & \cdots & a_{1,N,i} \\ \vdots & \ddots & \vdots \\ a_{N,1,i} & \cdots & a_{N,N,i} \end{pmatrix}, \tag{55}$$

where ⊗ represents the Kronecker product and I_M represents the M dimensions identity matrix, and the vector

$$\varDelta\omega_i = col\left\{\varDelta\omega_{1,i}, \ldots, \varDelta\omega_{N,i}\right\}. \tag{56}$$

Using the Kronecker product and the introduced matrices, we have the equation:

$$\varDelta\omega_i = \bar{G}_i (I_{MN} - MU_i)\varDelta\omega_{i-1} + \bar{G}_i L b_i. \tag{57}$$

Considering the independence of$\varDelta\omega_i$, U_i and $\bar{G}_i$ based on the first two assumptions, we can average Eq. (57) and obtain the following equation:

$$E\left\{\varDelta\omega_i\right\} = (\bar{G}_i \otimes I_M)(I_{MN} - MR)E\left\{\varDelta\omega_{i-1}\right\}, \tag{58}$$

where

$$R = blockdiag\left\{R_1, \ldots, R_n\right\}. \tag{59}$$

The calculation yields:

$$(G_i \otimes I_M) I_{MN} = I_{MN}, \tag{60}$$

where $G_i \otimes I_M$ denotes a positive value, and by satisfying the inequality

$$\left|\lambda_{max}\left\{I_{MN} - MR\right\}\right| < 1, \tag{61}$$

we can finally obtain an unbiased solution. In conclusion, the convergence condition for $\lambda_{max}(A)$, representing the eigenvalue of the Hermitian matrix A, is given by:

$$0 < \mu_i < \frac{2}{\lambda_{max}\left\{R_i\right\}}. \tag{62}$$

Table 1 Single iteration complexity of the CCLMS-DDS algorithm

Algorithm steps	Multiply/divide	Add/subtract
$e_k(i) = e_k(i) - u_{l,i}\omega_{k,i-1}$	L	L
$\varphi_{k,i} = \omega_{k,i-1} + \mu_k u_{k,i}^{*} e_k(i)$	L+1	L
$\omega_{k,i} = \sum a_{k,l}\varphi_{l,i}$	$r_k L$	$(r_k - 1)L$
$\delta(i+1) = \delta(i) + \mu_\alpha e(i)[y_1(i) - y_2(i)]\lambda(i)[1 - \lambda(i)]$	1	1
$\lambda(i) = sgm[\delta(i)] = \frac{1}{1+e^{-\delta(i)}}$	1	1
$y(i) = \lambda(i) y_1(i) + [1 - \lambda(i)] y_2(i)$.	2	1

Table 2 Complexity comparison between the CCLMS-DDS algorithm and the DLMS algorithm

Algorithm	Multiply/divide	Add/Subtract
CCLMS-DDS algorithm	$2\left[(r_k + 2)L + 1\right] + 4$	$2\left[(r_k + 2)L\right] + 3$
DLMS algorithm	$(r_k + 2)L + 1$	$(r_k + 1)L$

4.2 Complexity Comparison

This section presents an analysis of the computational complexity of the CCLMS-DDS algorithm. The aim is to enhance the robustness of the DLMS algorithm in unknown external environments and ensure reliable estimation performance. However, the introduction of additional steps in the convex combination phase leads to increased computation time compared to the DLMS algorithm.

Table 1 provides the complexity of each node's computation in each iteration when using the CCLMS-DDS algorithm. Table 2 compares the complexity between the DLMS algorithm and the CCLMS-DDS algorithm in each iteration. The results indicate that the computational complexity of the proposed algorithm is higher than that of the DLMS algorithm. This difference arises from the inclusion of two filter calculations.

Although both the DLMS algorithm and the CCLMS-DDS algorithm are relatively straightforward to implement, the DLMS algorithm encounters challenges when dealing with unknown external environments. Specifically, it struggles to strike a balance between low steady-state error and high convergence speed. In contrast, the proposed algorithm addresses this issue by increasing the computational complexity of the filter.

Overall, the CCLMS-DDS algorithm offers improved performance in unknown external environments, but it comes at the cost of higher computational complexity.

5 Simulation and Result

Simulation experiments were conducted on the MATLAB platform to validate the feasibility of the proposed algorithm and to compare it with existing algorithms.

Table 3 Filter parameter settings

Parameters	μ_1(Large)	μ_2(Small)	Number of iterations
Value	0.05	0.005	2000

Table 4 Noise signal settings

Parameters	Variance	Amplitude	Number of iterations
Value	0.01	1	2000

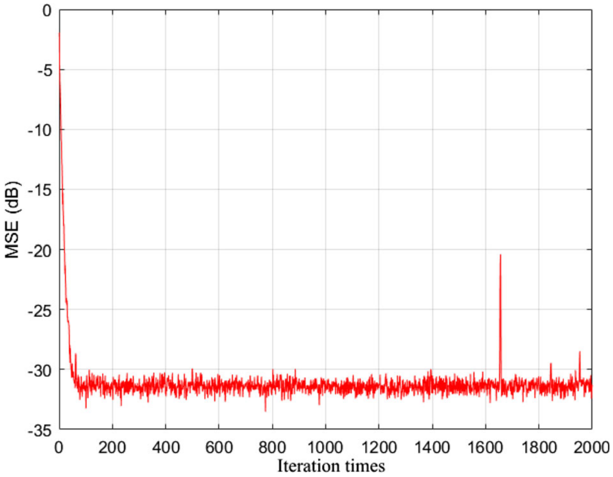


Fig. 3 MSE of the filter whose step size is 0.05

5.1 The Impact of Different Step Size Settings on the Performance of the LMS Algorithm

In order to investigate the impact of different step size settings on the performance of the LMS algorithm, various step sizes were tested and their effects were analyzed. Table 3 provides an overview of the different step size settings, while Table 4 presents the settings for the noise signal.

Firstly, when the step size of the LMS algorithm is set to 0.05, as shown in Fig. 3, the convergence speed is significantly faster compared to previous settings. It reaches the convergence state in approximately 50 iterations. However, an issue arises with a subsequent occurrence of severe jitter in the MSE. This indicates the presence of a large error in the MSE, leading to a significant steady-state error between stable and accurate values.

Secondly, when the step size of the LMS algorithm is set to 0.005, as depicted in Fig. 4, the convergence speed becomes excessively slow.

Furthermore, in comparison to the filter with a large step size of 0.05, the convex combination filter exhibits a slower iteration speed, as shown in Fig. 5. It takes

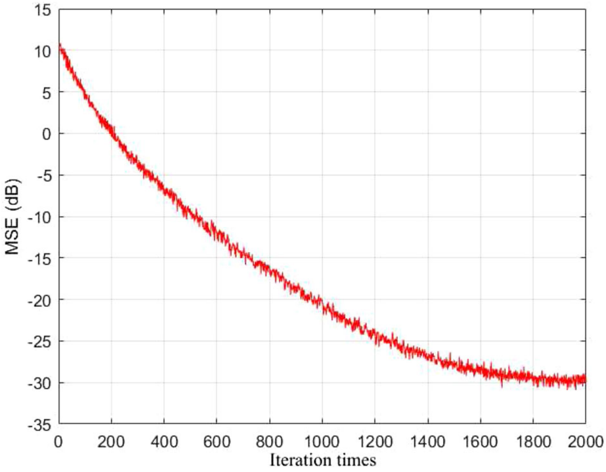


Fig. 4 MSE of the filter whose step size is 0.005

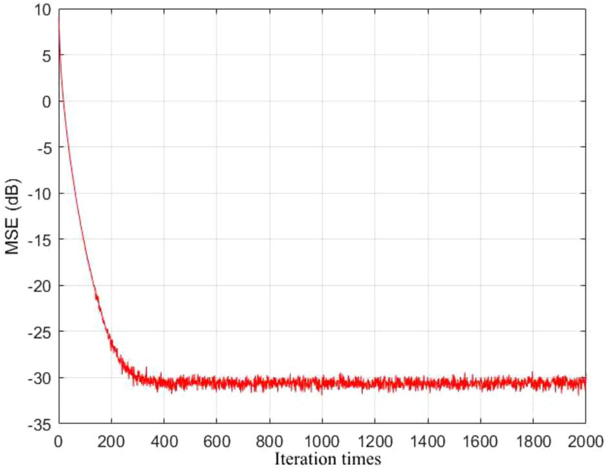


Fig. 5 MSE of the algorithm proposed in this paper

around 300 iterations to reach a steady state. However, once convergence is achieved, the stability is robust, and the steady-state error is effectively suppressed at -30 dB. Compared to the filter with a step size of 0.005, the convex combination filter converges faster. The filter with a step size of 0.005 requires 1200 iterations to converge, while the convex combination filter reaches a steady state in 300 iterations. Moreover, after reaching a steady state, the steady-state error is controlled at -30 dB.

The simulation results highlight the impact of different step size settings on the performance of the LMS algorithm. The convex combination filter demonstrates improved convergence speed and steady-state error suppression compared to the filters with larger or smaller step sizes. These findings support the effectiveness of the proposed algorithm in achieving desirable performance characteristics.

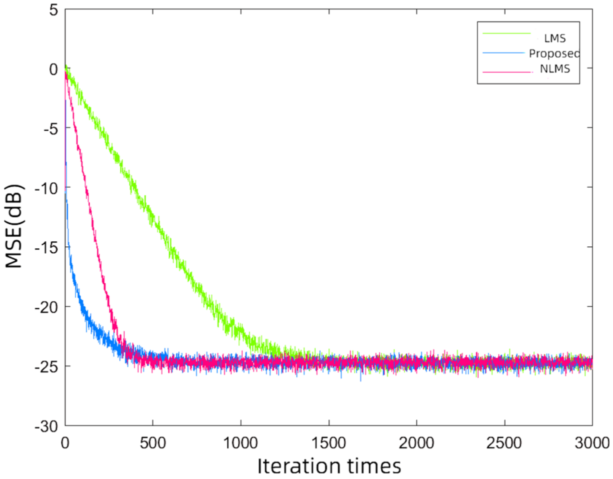


Fig. 6 Performance improvement of algorithms with convex combination strategy

5.2 Performance Improvement of Algorithms with Convex Combination Strategy

The convex combination strategy, as discussed in the previous theoretical analysis, plays a crucial role in achieving a balance between convergence speed and steady-state error by continuously adjusting the weights between filters based on external conditions and errors. The performance of the LMS algorithm is affected by the statistical characteristics of the input signal. To address the challenges in parameter selection and ensure filter convergence, Widrow et al. introduced the normalized LMS (NLMS) algorithm by incorporating a normalization operation, allowing for a flexible step size selection ($0 < \mu < 2$) independent of the input signal's statistical characteristics. In our simulation experiments, we compare our proposed algorithm with the NLMS algorithm, as depicted in Fig. 6.

The convergence and stability time of both the NLMS algorithm and our proposed algorithm are nearly identical, approximately 400 iterations, which is significantly faster than the 1200 iterations required by the conventional LMS algorithm. It is worth noting that our proposed method exhibits a faster convergence speed than the NLMS algorithm prior to reaching stability, highlighting the significant advantages of our approach utilizing the convex combination strategy.

The results clearly demonstrate the substantial superiority of our proposed method utilizing the convex combination strategy over the NLMS algorithm. These findings provide valuable insights for the design of adaptive filtering algorithms that achieve enhanced convergence speed and stability, especially in the presence of varying input signal statistics. Further research and application of the convex combination strategy have the potential to contribute to the development of advanced adaptive filtering techniques, thereby improving their performance in real-world scenarios.

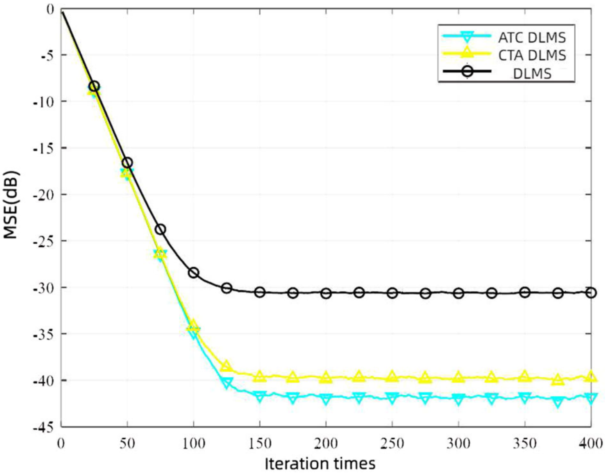


Fig. 7 The performance comparison of DLMS algorithms with different strategies

5.3 The Impact of Different Strategies and Step Size Settings on the Performance of the DLMS Algorithm

In this study, we conducted experiments to evaluate the performance of the DLMS algorithm and the proposed method. The parameters used for both algorithms were as follows: $N = 20$ nodes in the network, Gaussian input signals with zero mean, a filter order of $M = 5$, and Gaussian noise $v_k(i)$ with a variance of $[-2, -1]dB$. The input signals at the nodes followed a Gaussian distribution with a variance of $\sigma^2 = 1$.

As mentioned in Sect. 3.2, the ATC strategy has shown superior performance compared to the CTA strategy in distributed networks. To further validate this claim, additional experiments were conducted and presented in Fig. 7. The figure illustrates the network MSE performance curves of the DLMS algorithm under the non-cooperative approach, ATC strategy, and CTA strategy, respectively. The results clearly demonstrate that both the ATC and CTA strategies achieve comparable convergence speeds, but the ATC strategy outperforms the CTA strategy in terms of steady-state error reduction. Moreover, the performance of both strategies surpasses that of the non-cooperative approach, highlighting the effectiveness of information exchange among nodes in improving estimation performance. These experimental findings further support the practical relevance of the ATC strategy in distributed networks.

For the DLMS algorithm, step sizes μ_k of 0.05, 0.005, and 0.01 were set for all nodes. In the proposed method, the parameter α was set to 4, and the step sizes for the two filters were $\mu_1 = 0.05$ and $\mu_2 = 0.005$, respectively. The network MSE, which indicates estimation accuracy, was used to evaluate the performance, with smaller MSE values indicating more accurate estimation. Each simulation result represents the average of 10 independent experiments.

Before directly comparing the DLMS algorithm and the proposed method, we first analyzed the impact of different step sizes on the performance of the DLMS algorithm.

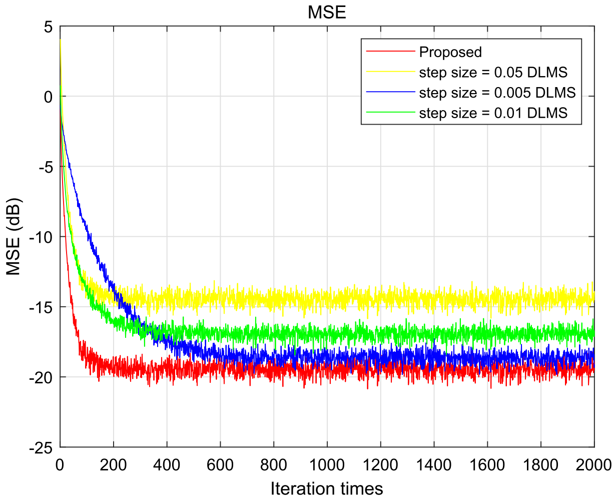


Fig. 8 The performance comparison between the DLMS algorithm and the proposed algorithm under different step sizes

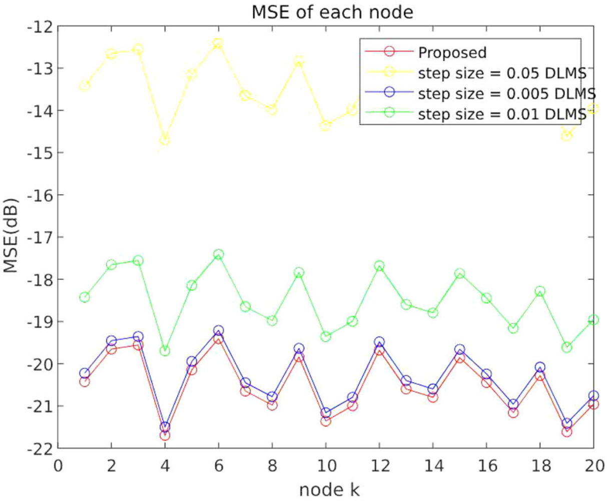


Fig. 9 The performance comparison of the DLMS algorithm and the proposed algorithm for each node under different step sizes

This analysis aimed to determine the optimal parameters for the DLMS algorithm. Additional experimental simulations were conducted, and the results, depicted in Figs. 8 and 9, indicate that a larger step size (e.g., $\mu_k = 0.05$) leads to faster convergence but poorer steady-state error performance. Conversely, a smaller step size (e.g., $\mu_k = 0.005$) results in slower convergence but better steady-state error performance. It is evident that the DLMS algorithm with a fixed step size cannot achieve a satisfactory balance between convergence speed and steady-state performance.

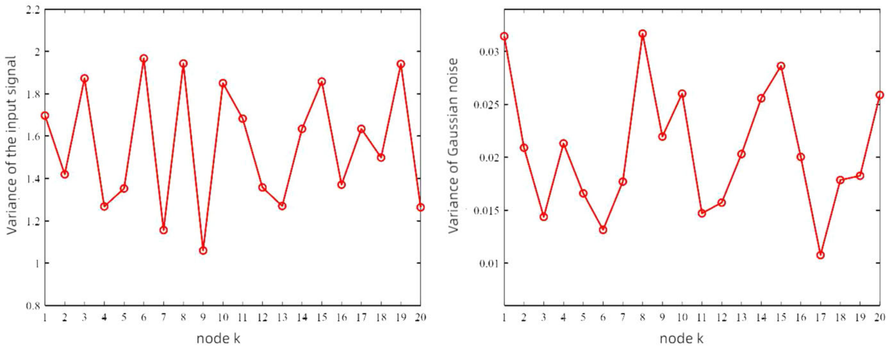


Fig. 10 Variance of input signal and noise in the network nodes

5.4 Performance Comparison of the Proposed Algorithm in Gaussian Noise Environment with Similar Approaches

In the considered network with $N = 20$ nodes, each node's input vector $u_{k,i}$ follows a zero-mean Gaussian process. The filter order is set to $M = 5$, and the variance of Gaussian noise $\upsilon_k(i)$ ranges from -20 to $-15dB$. The input signals of each node and the variance of Gaussian noise are shown in Fig. 10. The simulation results are obtained from averaging 200 independent experiments.

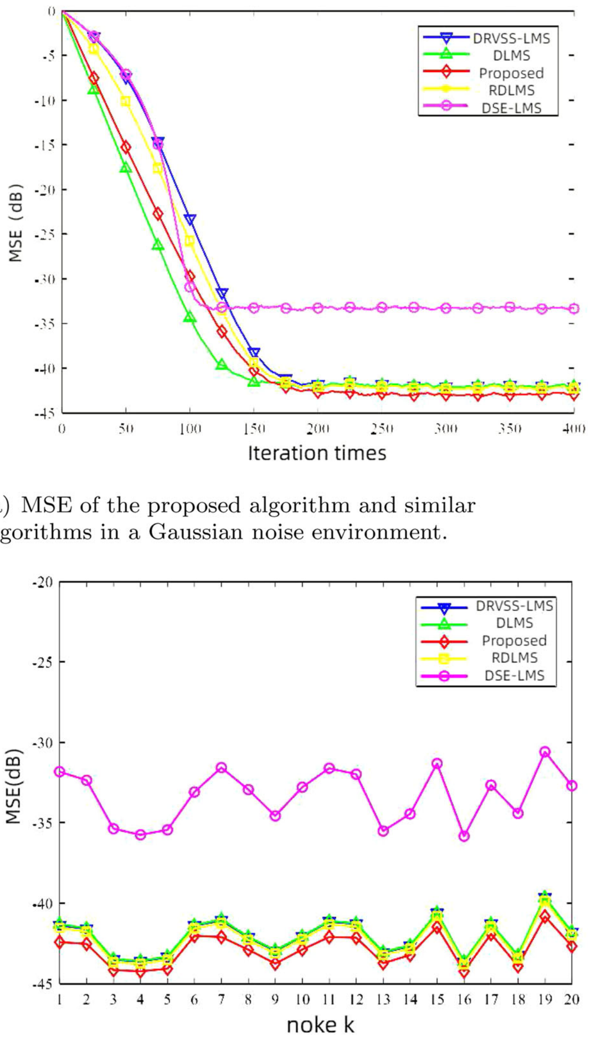
In this section, we compare the performance of the proposed method with similar algorithms. Specifically, we consider the DRVSS-LMS algorithm [14] with parameters $\alpha = 2.6$ and $\lambda = 0.99$, as well as the RDLMS algorithm [3] with parameter $\delta = 0.9$.

Figure 11a and b presents the performance comparison of different algorithms under Gaussian noise conditions. Figure 11a shows the network MSE curves of the proposed algorithm and similar algorithms in a Gaussian noise environment, while Fig. 11b illustrates the steady-state MSE of each node in the network under Gaussian noise conditions.

From the figures, it is evident that the DSE-LMS algorithm exhibits the poorest steady-state estimation performance among the five algorithms, with a slower convergence speed. This can be attributed to the DSE-LMS algorithm's use of the sign function in the gradient calculation, which filters both abnormal error information contaminated by impulse noise and normal error information. As a result, the DSE-LMS algorithm experiences slow convergence due to a small gradient in the early stages and significant steady-state error caused by a large gradient in the steady-state phase.

In contrast, the proposed algorithm employs the convex combination strategy, which reduces computational complexity and dynamically adjusts the weight of the convex combination filter. It achieves a faster convergence speed compared to the other four algorithms (excluding the DLMS algorithm) and approaches the convergence speed of the DLMS algorithm. Moreover, in the steady-state phase, the proposed algorithm exhibits lower steady-state error than the DLMS algorithm. This demonstrates a successful balance between convergence speed and steady-state error.

These results highlight the superiority of the proposed algorithm in achieving an optimal trade-off between convergence speed and steady-state error. The findings



(a) MSE of the proposed algorithm and similar algorithms in a Gaussian noise environment.

(b) MSE of each node in the network under Gaussian noise conditions.

Fig. 11 The performance comparison of different algorithms under Gaussian noise conditions

further validate the effectiveness and practical applicability of the proposed algorithm in real-world scenarios.

5.5 Performance Comparison with the DLMS Algorithm Based on Impulse Noise

The Bernoulli–Gaussian model is commonly used to describe impulse noise distribution, which combines Bernoulli distribution and Gaussian distribution to model impulse noise as contaminated Gaussian noise. In this model, the background noise $\upsilon_k(i)$ consists of two components:

$$v_k(i) = \upsilon_k(i) + I_k(i), \tag{63}$$

zero-mean Gaussian noise $\upsilon_k(i)$ and impulse noise $I_k(i)$, which is independent of the input vector and Gaussian noise. Impulse noise $I_k(i)$ is expressed as

$$I_k(i) = \theta_k(i) q_k(i) \tag{64}$$

, where $\theta_k(i)$ follows a Bernoulli distribution with probability $Pr[\theta_k(i)] = p_k$, and $q_k(i)$ is zero-mean Gaussian noise with variance $\zeta_k.\sigma^2_{k,i}$. The power of impulse noise is significantly higher than that of Gaussian noise $\nu_k(i)$. Figure 12a, b illustrates the background noise representation for $p_k = 0.06$ and $p_k = 0.12$.

The impact of impulse noise is visually depicted in Fig. 12a and b, where the darkened areas represent the regions of impulse noise occurrence. The occurrence probability of impulse noise at each node, p_k, influences the overall noise characteristics of the network. A higher occurrence probability, such as $p_k = 0.12$, results in more frequent impulse noise occurrences and affects a larger portion of the network.

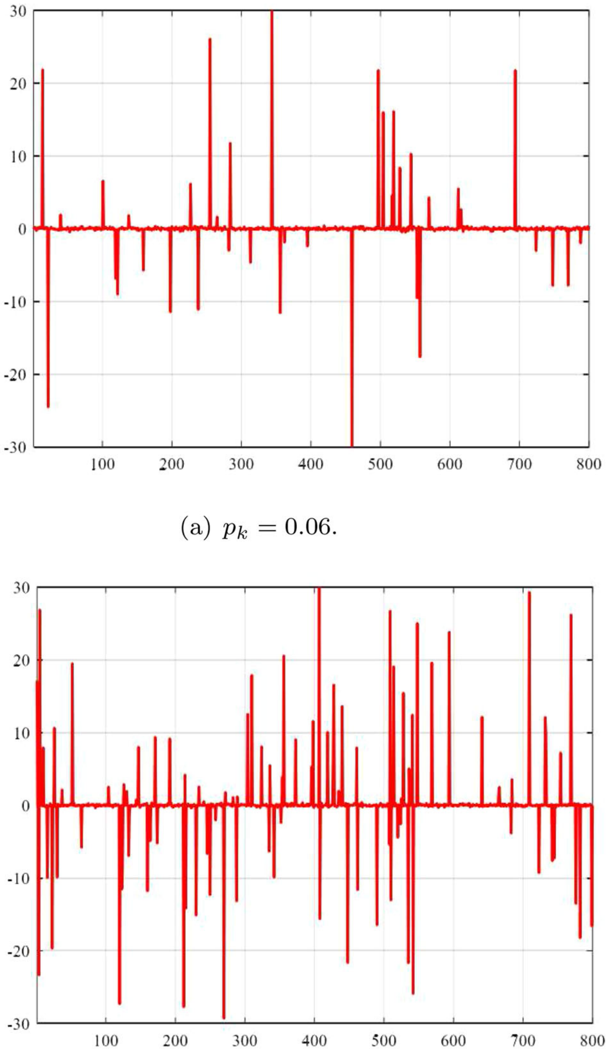
The Bernoulli–Gaussian model provides a valuable framework for accurately modeling and analyzing impulse noise in practical scenarios, enabling better understanding of communication systems in the presence of impulse noise.

The subsequent sections will further analyze and evaluate the effects of impulse noise on communication system performance, emphasizing the importance of mitigating these effects and proposing effective techniques to enhance system performance in such challenging environments.

The influence of different occurrence probabilities of impulse noise on the DLMS algorithm under the Bernoulli–Gaussian model is investigated. In this context, $p_k = 0$ represents the absence of impulse noise, where the background noise follows a Gaussian distribution. Nonzero probability values indicate the likelihood of impulse noise occurrence. The graph reveals that even with a very low occurrence probability, the DLMS algorithm experiences a significant increase in steady-state error. As the occurrence probability increases, the algorithm's performance further deteriorates, potentially leading to algorithmic failure.

Figure 13 illustrates the relationship between the occurrence probability of impulse noise and the steady-state error of the DLMS algorithm. The steady-state error shows a substantial increase as the occurrence probability of impulse noise grows. This vulnerability of the DLMS algorithm to impulse noise, even at low occurrence probabilities, indicates the need for effective mitigation strategies and robust algorithms to address impulse noise scenarios and prevent performance degradation.

These observations underscore the importance of addressing impulse noise in practical communication systems, especially when the occurrence probability is nonnegligible. Mitigating the effects of impulse noise and developing robust algorithms capable of handling such scenarios are crucial research directions for improving the performance and reliability of communication systems.



(a) $p_k = 0.06.$

(b) $p_k = 0.12.$

Fig. 12 The background noise contaminated with impulsive noise of varying occurrence probabilities. The X-axis represents the noise samples, while the Y-axis denotes the magnitude of the noise

Figure 14a and b compares the performance of various algorithms under the condition that the occurrence probability of impulse noise is $p_k = 0.1$. Figure 14a presents the network MSE comparison of the proposed algorithm and similar algorithms in the Bernoulli–Gaussian noise environment, while Fig. 14b compares the steady-state MSE of each node in the network under the same noise conditions.

From Fig. 14a and b, it can be observed that in the presence of impulse noise, the DLMS algorithm exhibits significant performance deterioration, rendering it ineffec-

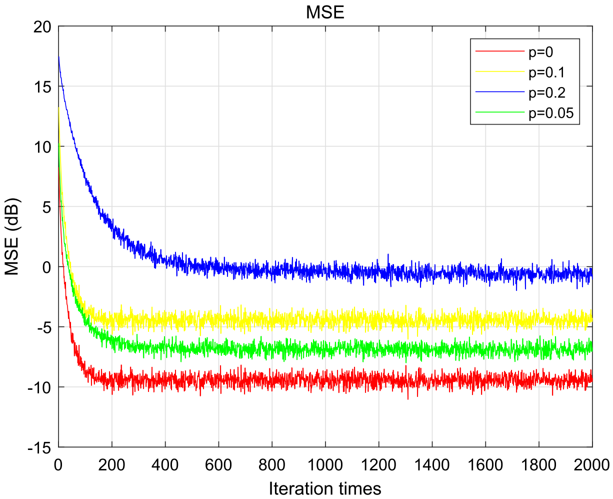


Fig. 13 The relationship between the occurrence probability of impulse noise and the steady-state error of the DLMS algorithm

tive in estimation tasks. In contrast, both the DRVSS-LMS algorithm and the RDLMS algorithm show slight degradation in steady-state estimation performance but still demonstrate robustness against impulse noise interference compared to the Gaussian noise scenario.

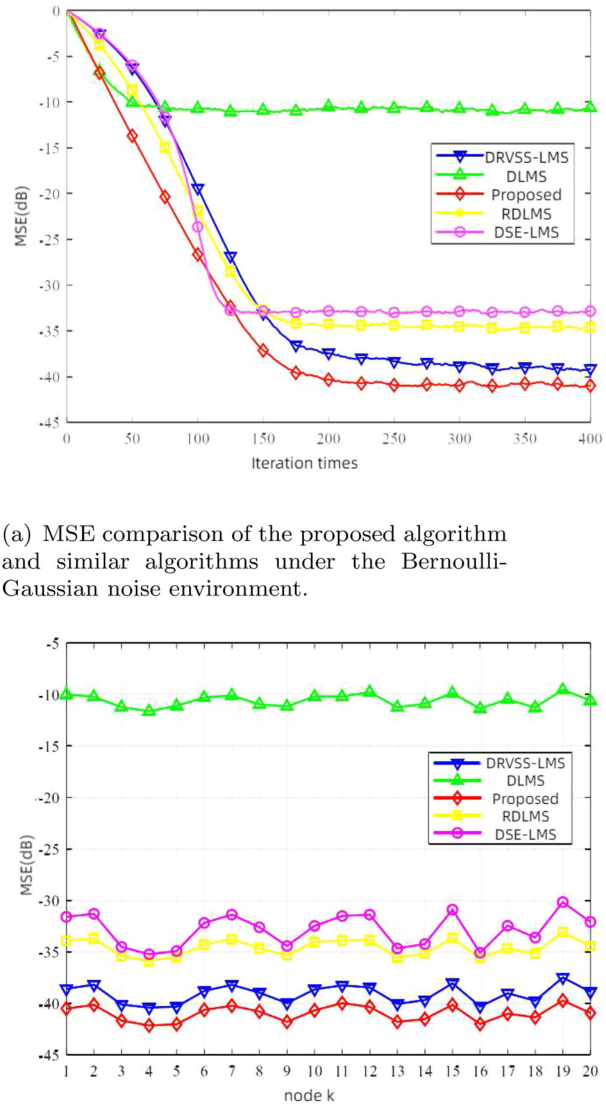
The proposed algorithm in this study and the DSE-LMS algorithm exhibit consistent steady-state estimation performance under the Bernoulli–Gaussian noise environment, indicating their strong robustness. Furthermore, the proposed algorithm achieves faster convergence speed and smaller steady-state error compared to the other algorithms.

These findings highlight the resilience of the proposed algorithm and its ability to handle impulse noise, demonstrating its superiority in achieving accurate and reliable estimations in practical applications.

6 Conclusion

In conclusion, this study has presented a novel convex combination distributed LMS algorithm that addresses the challenges of convergence speed and steady-state error in sensor networks. By leveraging the distributed diffusion strategy and incorporating the concept of convex combination, our algorithm has achieved decentralized and robust estimation performance.

The experimental results have demonstrated the feasibility and effectiveness of the proposed algorithm in various scenarios. We have shown that the convex combination distributed LMS algorithm outperforms existing algorithms in terms of convergence speed, steady-state error, and resilience against impulse noise interference. The algorithm strikes a balance between convergence speed and estimation accuracy, offering improved performance compared to the traditional DLMS algorithm.



(a) MSE comparison of the proposed algorithm and similar algorithms under the Bernoulli-Gaussian noise environment.

(b) MSE of each node in the network under Bernoulli-Gaussian noise conditions.

Fig. 14 The performance comparison of different algorithms under Bernoulli–Gaussian noise conditions

While the proposed algorithm has shown promising results, there are areas for further improvement and research. Firstly, future work could focus on enhancing the convergence speed without compromising the overall performance. This can be achieved by exploring adaptive step size control techniques, such as variable step sizes based on the network conditions, to accelerate convergence while maintaining stability.

Secondly, reducing the steady-state error remains an important objective. Investigating advanced regularization methods or incorporating adaptive algorithms that adaptively adjust the algorithm's parameters during operation could help minimize the steady-state error and enhance the accuracy of the estimation results.

Additionally, considering the computational complexity of the proposed algorithm is essential for practical implementation. Future research can explore techniques to optimize the algorithm's computational requirements, such as employing parallel computing architectures or developing efficient algorithms that strike a better balance between performance and computational cost.

Furthermore, the application of the convex combination distributed LMS algorithm can be extended to non-stationary environments or nonlinear systems. Addressing these challenges will require adapting the algorithm to handle time-varying or nonlinear characteristics, potentially through the use of adaptive filtering techniques or advanced machine learning algorithms.

In conclusion, the convex combination distributed LMS algorithm offers a valuable contribution to the field of adaptive filtering in sensor networks. While further research is needed to enhance convergence speed, reduce steady-state error, and optimize computational complexity, the algorithm's potential for improved estimation performance is evident. By addressing these research directions, we can advance the algorithm's practical applicability and foster its adoption in real-world scenarios, contributing to the development of more efficient and reliable sensor network systems.

Acknowledgements We acknowledge TopEdit LLC for the linguistic editing and proofreading during the preparation of this manuscript.

Declarations

Conflict of interest We declare that we have no financial and personal relationships with other people or organizations that can inappropriately influence our work; there is no professional or other personal interest of any nature or kind in any product, service or company that could be construed as influencing the position presented in the manuscript entitled "A Convex Combination Least Mean Square Algorithm Based on the Distributed Diffusion Strategy for Sensor Networks". The authors confirm that the data supporting the findings of this study are available within the article.

References

1. J. Arenas-Garcia, A. Figueiras-Vidal, Adaptive combination of normalised filters for robust system identification. Electron. Lett. **41**(15), 874–875 (2005)
2. J. Arenas-García, M. Martínez-Ramón, A. Navia-Vazquez, A.R. Figueiras-Vidal, Plant identification via adaptive combination of transversal filters. Signal Process. **86**(9), 2430–2438 (2006)
3. S. Ashkezari-Toussi, H. Sadoghi-Yazdi, Robust diffusion LMS over adaptive networks. Signal Process. **158**, 201–209 (2019)
4. A. Carini, G.L. Sicuranza, Fourier nonlinear filters. Signal Process. **94**, 183–194 (2014)
5. A. Carini, G.L. Sicuranza, Recursive even mirror Fourier nonlinear filters and simplified structures. IEEE Trans. Signal Process. **62**(24), 6534–6544 (2014)
6. A. Carini, G.L. Sicuranza, BIBO-stable recursive functional link polynomial filters. IEEE Trans. Signal Process. **65**(6), 1595–1606 (2016)

7. A. Carini, G.L. Sicuranza, A study about Chebyshev nonlinear filters. Signal Process. **122**, 24–32 (2016)
8. F.S. Cattivelli, A.H. Sayed, Diffusion LMS strategies for distributed estimation. IEEE Trans. Signal Process. **58**(3), 1035–1048 (2009)
9. S.C. Chan, Y.X. Zou, A recursive least M-estimate algorithm for robust adaptive filtering in impulsive noise: Fast algorithm and convergence performance analysis. IEEE Trans. Signal Process. **52**(4), 975–991 (2004)
10. H. Chang, W. Li, Correction-based diffusion LMS algorithms for distributed estimation. Circ. Syst. Signal Process. **39**, 4136–4154 (2020)
11. D.P. Das, G. Panda, Active mitigation of nonlinear noise processes using a novel filtered-s LMS algorithm. IEEE Trans. Speech Audio Process. **12**(3), 313–322 (2004)
12. X. Guo, J. Jiang, J. Chen, S. Du, L. Tan, Convex combination recursive even mirror Fourier nonlinear filter for nonlinear active noise control, in *2019 22nd International Conference on Electrical Machines and Systems (ICEMS)*, IEEE, pp. 1-6 (2019)
13. X. Guo, Y. Li, J. Jiang, C. Dong, S. Du, L. Tan, Adaptive function expansion 3-D diagonal-structure bilinear filter for active noise control of saturation nonlinearity. IEEE Access **6**, 65139–65150 (2018)
14. W. Huang, L. Li, Q. Li, X. Yao, Diffusion robust variable step-size LMS algorithm over distributed networks. IEEE Access **6**, 47511–47520 (2018)
15. D.C. Le, J. Zhang, Y. Pang, A bilinear functional link artificial neural network filter for nonlinear active noise control and its stability condition. Appl. Acoust. **132**, 19–25 (2018)
16. R. Leahy, Z. Zhou, Y.C. Hsu, Adaptive filtering of stable processes for active attenuation of impulsive noise, in *1995 international conference on acoustics, speech, and signal processing*, IEEE, pp. 2983-2986 (1995)
17. C.G. Lopes, A.H. Sayed, Incremental adaptive strategies over distributed networks. IEEE Trans. Signal Process. **55**(8), 4064–4077 (2007)
18. L. Lu, H. Zhao, Adaptive Volterra filter with continuous lp-norm using a logarithmic cost for nonlinear active noise control. J. Sound Vib. **364**, 14–29 (2016)
19. L. Luo, J. Sun, A novel bilinear functional link neural network filter for nonlinear active noise control. Appl. Soft Comput. **68**, 636–650 (2018)
20. G. Mateos, I.D. Schizas, G.B. Giannakis, Distributed recursive least-squares for consensus-based in-network adaptive estimation. IEEE Trans. Signal Process. **57**(11), 4583–4588 (2009)
21. J. Ni, J. Chen, X. Chen, Diffusion sign-error LMS algorithm: formulation and stochastic behavior analysis. Signal Process. **128**, 142–149 (2016)
22. V. Patel, V. Gandhi, S. Heda, N.V. George, Design of adaptive exponential functional link network-based nonlinear filters. IEEE Trans. Circuits Syst. I Regul. Pap. **63**(9), 1434–1442 (2016)
23. M.O.B. Saeed, A. Zerguine, A new variable step-size strategy for adaptive networks, in *2011 Conference Record of the Forty Fifth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, IEEE, pp. 312–315 (2011)
24. M.O.B. Saeed, A. Zerguine, S.A. Zummo, Variable step-size least mean square algorithms over adaptive networks, in *10th International Conference on Information Science, Signal Processing and their Applications (ISSPA 2010)*, IEEE, pp. 381–384 (2010)
25. I.D. Schizas, G. Mateos, G.B. Giannakis, Distributed LMS for consensus-based in-network adaptive processing. IEEE Trans. Signal Process. **57**(6), 2365–2382 (2009)
26. G.L. Sicuranza, A. Carini, A generalized FLANN filter for nonlinear active noise control. IEEE Trans. Audio Speech Lang. Process. **19**(8), 2412–2417 (2011)
27. G.L. Sicuranza, A. Carini, On the BIBO stability condition of adaptive recursive FLANN filters with application to nonlinear active noise control. IEEE Trans. Audio Speech Lang. Process. **20**(1), 234–245 (2011)
28. L. Tan, C. Dong, S. Du, On implementation of adaptive bilinear filters for nonlinear active noise control. Appl. Acoust. **106**, 122–128 (2016)
29. L. Tan, J. Jiang, Adaptive Volterra filters for active control of nonlinear noise processes. IEEE Trans. Signal Process. **49**(8), 1667–1676 (2001)
30. P. Thanigai, S.M. Kuo, R. Yenduri, Nonlinear active noise control for infant incubators in neonatal intensive care units, in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, IEEE, pp. 1–109 (2007)

31. G. Wang, H. Zhao, P. Song, Robust variable step-size reweighted zero-attracting least mean M-estimate algorithm for sparse system identification. IEEE Trans. Circuits Syst. II Express Briefs **67**(6), 1149–1153 (2019)
32. L. Wu, H. He, X. Qiu, An active impulsive noise control algorithm with logarithmic transformation. IEEE Trans. Audio Speech Lang. Process. **19**(4), 1041–1044 (2010)
33. Y. Yu, H. Zhao, Robust incremental normalized least mean square algorithm with variable step sizes over distributed networks. Signal Process. **144**, 1–6 (2018)
34. Y. Yu, H. Zhao, W. Wang, L. Lu, Robust diffusion Huber-based normalized least mean square algorithm with adjustable thresholds. Circ. Syst. Signal Process. **39**, 2065–2093 (2020)
35. L. Yun, H. Xiaobin, G. Qianqian, Diffusion variable step size algorithm based on maximum correntropy criterion, in *2021 IEEE International Conference on Power Electronics, Computer Applications (ICPECA)*, IEEE, pp. 558–561 (2021)
36. H. Zayyani, Communication reducing diffusion LMS robust to impulsive noise using smart selection of communication nodes. Circ. Syst. Signal Process. **41**(3), 1788–1802 (2022)
37. H. Zhao, X. Zeng, Z. He, T. Li, Adaptive RSOV filter using the FELMS algorithm for nonlinear active noise control systems. Mech. Syst. Signal Process. **34**(1–2), 378–392 (2013)
38. H. Zhao, X. Zeng, Z. He, S. Yu, B. Chen, Improved functional link artificial neural network via convex combination for nonlinear active noise control. Appl. Soft Comput. **42**, 351–359 (2016)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.