Circuits, Systems, and Signal Processing (2023) 42:5811–5837
https://doi.org/10.1007/s00034-023-02356-9

A Distributed Adaptive Algorithm Based on the Asymmetric Cost of Error Functions

Sihai Guan[1,2] · Yong Zhao[3] · Liwei Wang[1,2] · Qing Cheng[4] · Bharat Biswal[5,6]

Received: 24 August 2021 / Revised: 10 March 2023 / Accepted: 11 March 2023 /
Published online: 4 May 2023

Abstract

In this paper, a family of novel diffusion adaptive estimation algorithms is proposed from the asymmetric cost function perspective by combining diffusion strategy and the linear–linear cost, quadratic-quadratic cost, and linear-exponential cost at all distributed network nodes, and named diffusion LLCLMS (DLLCLMS), diffusion QQCLMS (DQQCLMS), and diffusion LECLMS (DLECLMS), respectively. Then, the stability of mean estimation error and computational complexity of those three diffusion algorithms are analyzed theoretically. Finally, several experiment simulation results are designed to verify the superiority of those three proposed diffusion algorithms. Results show that DLLCLMS, DQQCLMS, and DLECLMS algorithms

✉ Sihai Guan
gcihey@sina.cn

✉ Qing Cheng
15107739422@163.com

✉ Bharat Biswal
bbiswal@gmail.com

Yong Zhao
henanzyo1984@163.com

Liwei Wang
wangliwei@swun.edu.cn

[1] College of Electronic and Information, Southwest Minzu University, Chengdu, China

[2] Key Laboratory of Electronic and Information Engineering, State Ethnic Affairs Commission, Chengdu, China

[3] School of Mechanical and Power Engineering, Henan Polytechnic University, Jiaozuo, China

[4] Sichuan Vocational College of Finance and Economics, Chengdu, China

[5] The Clinical Hospital of Chengdu Brain Science Institute, MOE Key Laboratory for Neuroinformation, Center for Information in Medicine, School of Life Science and Technology, University of Electronic Science and Technology of China, Chengdu, China

[6] Department of Biomedical Engineering, New Jersey Institute of Technology (NJIT), Newark, NJ, USA

are more robust to the input signal and impulsive noise than the diffusion sign-error LMS, diffusion robust variable step-size least mean square (DRVSSLMS), and least mean logarithmic absolute difference algorithms. In brief, theoretical analysis and experiment results show that those proposed DLLCLMS, DQQCLMS, and DLE-CLMS algorithms perform better when estimating the unknown linear system under the changeable impulsive noise environments and different environments types of input signals.

Keywords Asymmetric cost function · Adaptive diffusion algorithm · Impulsive noise · Input signals

1 Introduction

So far, adaptive filter algorithms are often used in channel equalization, active interference control, echo cancellation, biomedical engineering [7, 15, 21, 28, 30, 31], and many other fields [23, 33, 34]. In recent years, especially in the research on wireless sensor networks, in other words, diffusion adaptive filtering algorithms have been widely studied due to their unique performance, which is an extension of adaptive filtering algorithms over network graphs [31]. Besides, three collaborative strategies for adaptive filtering/estimation algorithms on the distributed network are widely used, including incremental, consensus, and diffusion strategies. However, these three collaborative strategies have different performances; specifically, the consensus technique has an asymmetry problem, which can cause unstable growth. The diffusion strategies show good performance for unstable, and they can remove the asymmetry problem and real-time adaptation learning over distributed networks. So, diffusion strategies have been used frequently in the past decade, and they also include the adapt-then-combine (ATC) scheme [22] and the combine-then-adapt (CTA) scheme [5, 31]. Of course, the performance between these two schemes is also different; in response to this comparison, the specific explanation is that in the CTA formulation of the diffusion strategy, the name combine-then-adapt is that the first step involves a combination step, while the second step consists of an adaptation step; a similar implementation can be obtained by switching the order of the combination and adaptation steps. Moreover, Cattivelli and colleagues analyzed these two schemes, indicating that the ATC is better than the CTA [22]. With this conclusion, in the following research, the ATC scheme becomes a research focus in distributed adaptive filtering algorithms [2–4, 16, 19, 22, 24, 25, 39].

The Wiener filter principle is the fundamental adaptive filtering algorithm based on the minimum mean square estimation error to construct an efficient convex cost function [38]. We can know the core position of the cost function in the design of adaptive filtering algorithms. Besides, most of the cost functions were designed in the adaptive filtering algorithm field to satisfy symmetry, such as LMS [38], LMF [35], symmetric Gaussian kernel function [21], and hyperbolic function [18, 36]. It is worth noting that the mainstream cost function design constructs the independent variable as a function of the estimation error. Still, the estimation error is closely related to the measurement interference/noise. Not all the measurement interference/noise distributions satisfy symmetrical distributions, such as among the asymmetric interference/noise, impulsive noise is the most representative asymmetrical distribution. Impulsive noise will significantly affect adaptive estimation accuracy and most diffusion adaptive estimation/filtering algorithms over a network graph. Therefore, designing a robust distributed filtering algorithm for impulsive noise is necessary.

For this purpose, many papers have been written so far [1, 13, 17, 20, 26, 27, 37]. The diffusion least mean p-power (DLMP) algorithm was proposed by Wen [37], which is robust to the generalized Gaussian noise distribution environments and prior knowledge of the distribution. However, the DLMP algorithm was proposed with a fixed power p-value, so the parameter p-value is the critical factor, which means the DLMP algorithm performance is highly susceptible to the p-value. Based on the minimization of the L1-norm subject to a constraint on the adaptive estimate vectors, Ni and colleagues designed a diffusion sign subband adaptive filtering (DSSAF) algorithm [26]. The DSSAF algorithm performs better, but the computational complexity of DSSAF is relatively large.

Besides, by combining the diffusion least mean square (DLMS) algorithm [4] and the sign operation to the estimated error at each iteration moment point, Ni and colleagues derived a diffusion sign-error LMS (DSELMS) algorithm [27]. The DSELMS algorithm has a simple architecture, but the DSELMS algorithm has a significant drawback, i.e., the steady-state estimation error is high [11]. Based on the Huber cost function, a similar set of diffusion adaptive filtering algorithms by Guan and colleagues [20], Wei and colleagues [17], and Soheila and colleagues [1] have been proposed as the DNHuber, DRVSSLMS, and RDLMS algorithms, respectively. Nevertheless, among them, the RDLMS algorithm is impractical for impulsive noise. Discussing impulsive noise and input signals can be more comprehensive for the DNHuber algorithm. The DRVSSLMS algorithm has high algorithm computational complexity not conducive to implementing practical engineering. Besides, inspired by the least mean logarithmic absolute difference (LLAD) operation, Chen and colleagues designed another distributed adaptive filtering algorithm, i.e., the DLLAD algorithm [5]. But analysis of the robustness of the distributed algorithm to the input signal and impulsive noise has yet to be performed. To solve this problem, Guan and colleagues proposed a diffusion probabilistic least mean square (DPLMS) algorithm [13] by combining the ATC scheme and the probabilistic LMS algorithm [10, 16] at all distributed network nodes.

Can an asymmetric function be used to design a cost function in an adaptive filtering algorithm? It is a problem that needs urgent attention to ensure the effectiveness of adaptive filtering algorithms. The asymmetric cost function usually performs very well, especially when the estimated error variable is of symmetric distribution. However, symmetry and asymmetry are complementary concepts, so does the estimation error change symmetrical when running an adaptive filtering algorithm? Usually, the change in the estimation error cost function will be affected by measurement interference/noise. The distribution of most interference or noise does not satisfy symmetry. So, for asymmetric estimation error distribution, the symmetric cost function is inappropriate and cannot adapt to the error distribution well.

Therefore, for these asymmetric interferences, it is feasible to use the asymmetric function to design the cost function to construct a novel adaptive filtering algorithm. Combining the content of the above paragraph, we can see that most of the above distributed adaptive filtering algorithms are proposed using the symmetry cost functions [29]. So, to address the asymmetry estimated error distribution issue, we offer a family diffusion adaptive filtering algorithms by using three different asymmetric cost functions (i.e., the linear–linear cost (LLC), quadratic-quadratic cost (QQC), and linear-exponential cost (LEC) [6, 8, 9], namely the DLLCLMS, DQQCLMS, and DLECLMS algorithms. The stability of the mean estimation error of those three proposed diffusion algorithms is analyzed, and the computational complexity is also analyzed theoretically. Simulation experiment results indicate that the DLLCLMS, DQQCLMS, and DLECLMS algorithms are more robust to the input signal and impulsive noise than the DSELMS, DRVSSLMS, and DLLAD algorithms.

The rest parts of this article are organized as follows. The proposed DLLCLMS, DQQCLMS, and DLECLMS algorithms will be described in detail in Sect. 2. The statistical stability behavior, computation complexity, and parameters (a and b) of DLLCLMS, DQQCLMS, and DLECLMS algorithms are studied in Sect. 3. The simulation experiment is described in Sect. 4. Finally, conclusions are provided in Sect. 5. Note: Bold type refers to vectors, $[\ ]^{\mathrm{T}}$ denotes the transpose, $[\ ]^{-1}$ denotes the inverse operation, and | | denotes the absolute value operation.

2 Proposed the Diffusion Algorithms Using the Asymmetric Function

This section mainly describes designing the DLLCLMS, DQQCLMS, and DLECLMS algorithms. The specific plan is that the first step is to propose three adaptive filtering algorithms based on asymmetric cost functions (i.e., the LLC, QQC, and LEC functions). Then, the second step is to modify those three adaptive filtering algorithms by extending to all distributed network agents to propose the DLLCLMS, DQQCLMS, and DLECLMS algorithms.

2.1 Three Adaptive Filtering Algorithms Based on Asymmetric Cost Functions

Setting an unknown linear system with the length M of the system coefficient $\mathbf{W}^{\mathrm{o}}$, and $\mathbf{W}(i)$ be the adaptive estimated weight vector at iteration i, $\mathbf{X}(i)$ denotes the input signal vector of the adaptive filtering algorithm. The estimation error $e(i)$ between the desired signal $d(i)$ and the estimation output $y(i)$ can be expressed as Eqs. (1)–(2). Additionally, $v(i)$ is this unknown linear system measurement noise.

$$\begin{cases} d(i) = \mathbf{W}^{\mathrm{oT}} X(i) + v(i) \\ y(i) = \mathbf{W}^{\mathrm{T}}(i) X(i) \end{cases} \tag{1}$$

$$e(i) = d(i) - y(i) = \mathbf{W}^{\mathrm{o}}\mathbf{X}(i) + v(i) - \mathbf{W}^{\mathrm{T}}(i)\mathbf{X}(i) \tag{2}$$

Next, three asymmetric cost functions, including the LLC, QQC, and LEC functions, are used to design three adaptive filtering algorithms.

Firstly, LLC adaptive filtering algorithm aims to minimize the LLC cost function of estimation error defined as

$$J_{\mathrm{LLC}}(i) = \begin{cases} a\ e(i)\ ,\ \text{if } e(i) > 0 \\ b\ e(i)\ ,\ \text{if } e(i) \le 0 \end{cases} \tag{3}$$

Secondly, QQC adaptive filtering algorithm aims to minimize the QQC cost function of estimation error defined as

$$J_{\mathrm{QQC}}(i) = \frac{1}{2}\begin{cases} ae^2(i)\ ,\ \text{if} e(i) > 0 \\ be^2(i)\ ,\ \text{if} e(i) \le 0 \end{cases} \tag{4}$$

Thirdly, LEC adaptive filtering algorithm aims to minimize the LEC cost function of estimation error defined as

$$J_{\mathrm{LEC}}(i) = b\big[\exp(ae(i)) - ae(i) - 1\big] \tag{5}$$

In Eqs. (3), (4), and (5), $a, b > 0$ is the cut-off value.

Parameters a and b determine the shape and characteristics of each cost function; how to set it is critical. Parameters a and b facilitate the adjustment of the asymmetric cost of error functions to the empirical cost situation because they determine the severity of a given estimation error type. For example, setting $a = b$ reduces QQC to the mean squared error, whereas LLC is reduced to the mean absolute error. From Eq. (3), the LLC cost function behaves as a sign-error cost function estimator. Therefore, the LLC cost function can combine the sign-error cost function and asymmetric estimated error. From Eq. (4), the QQC cost function behaves as a square error cost function estimator. Therefore, the QQC cost function can combine the square error cost function estimator and asymmetric estimated error. From Eq. (5), the LEC cost function behaves as an exponential estimator. Therefore, the LEC cost function can combine the exponential function estimator and asymmetric estimated error. From the above analysis, it can be seen that these three cost functions are worthy of further study and then, used to design adaptive filtering algorithms.

According to the steepest descent method, the weight vector update of the LLC adaptive filter algorithm is

$$\begin{aligned}\mathbf{W}(i+1) =& \begin{cases} \mathbf{W}(i) + \mu a\ \mathrm{sign}(e(i))\mathbf{X}(i)\ ,\ \text{if } e(i) > 0 \\ \mathbf{W}(i) + \mu b\ \mathrm{sign}(e(i))\mathbf{X}(i)\ ,\ \text{if } e(i) \le 0 \end{cases} \\ =& \mathbf{W}(i) + \frac{\mu}{2}\big[a(1 + \mathrm{sign}(e(i))) + b(1 - \mathrm{sign}(e(i)))\big]\mathbf{X}(i) \end{aligned} \tag{6}$$

In Eq. (6), sign(·) denotes the sign function, and μ is the step size.

According to the steepest descent method, the weight vector update of the QQC adaptive filter algorithm is

$$\mathbf{W}(i+1) = \begin{cases} \mathbf{W}(i) + \mu ae(i)\mathbf{X}(i)\ ,\ \text{if } e(i) > 0 \\ \mathbf{W}(i) + \mu be(i)\mathbf{X}(i)\ ,\ \text{if } e(i) \le 0 \end{cases} \tag{7}$$

In similar operations, according to the steepest descent method, the weight vector update of the LEC adaptive filter algorithm is

$$\mathbf{W}(i+1)=\mathbf{W}(i)+\mu ab((\exp(ae(i))-1))\mathbf{X}(i) \tag{8}$$

In Eq. (8), the symbol $\exp(\cdot)$ denotes the exponential function, and μ is the step size.

2.2 Three Asymmetric Adaptive Diffusion Filtering Algorithms

As described in part of the introduction, recent research on wireless sensor networks has been widely studied due to their unique performance. So, according to the design results of the previous subsection, three adaptive filtering algorithms, combined with a schematic diagram of distributed network structure in our previous research papers [13, 20], set a distributed network of *N agent* sensor nodes (as Fig. 1). $\mathbf{X}_\mathrm{n}(i)$ and $d_\mathrm{n}(i)$ are the input signals and estimation output signals at agent n, respectively. It needs to be stated that this paper is different from our previous work [13, 20]. The similarity is how to realize the practical distributed adaptive estimation in impulsive interference. This paper uses three asymmetric functions to design three cost functions to construct a novel family adaptive filtering algorithm to address the asymmetry estimated error distribution issue. Moreover, explore the robustness of the algorithm developed in this paper to the input signal and impulsive interference.

Based on Fig. 1, by using minimizes the global cost function, we can seek the optimal linear estimator at each time instant i:

$$J^{\mathrm{global}}(\mathbf{W}(i))=\sum_{n}J_{\mathrm{n}}^{\mathrm{local}}(\mathbf{W}(i)) \tag{9}$$

Each sensor node $n\in\{1,2,\cdots,N\}$ has access to some zero-mean random process $\{d_\mathrm{n}(i),\mathbf{X}_\mathrm{n}(i)\}$, $d_\mathrm{n}(i)$ is a scalar, and $\mathbf{X}_\mathrm{n}(i)$ is a regression vector. Suppose these measurement signals follow a standard computational model given by:

$$d_\mathrm{n}(i)=\mathbf{W}^{\mathrm{oT}}\mathbf{X}_\mathrm{n}(i)+v_\mathrm{n}(i) \tag{10}$$

where $\mathbf{W}^\mathrm{o}$ is the unknown parameter vector with length M, and $v_\mathrm{n}(i)$ is the unknown linear distributed network system measurement noise with variance $\sigma_{v,n}^2$.

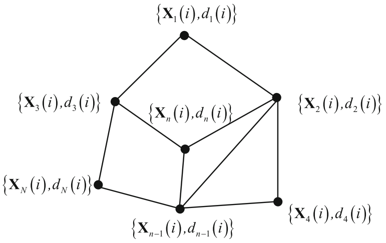
$\{\mathbf{X}_1(i),d_1(i)\}$
$\{\mathbf{X}_3(i),d_3(i)\}$
$\{\mathbf{X}_n(i),d_n(i)\}$
$\{\mathbf{X}_2(i),d_2(i)\}$
$\{\mathbf{X}_N(i),d_N(i)\}$
$\{\mathbf{X}_{n-1}(i),d_{n-1}(i)\}$
$\{\mathbf{X}_4(i),d_4(i)\}$

Fig. 1 A distributed network con N agent sensor nodes [13, 20]

The DLMS algorithm [4] is obtained by minimizing a linear combination of the local mean square estimation error:

$$J_n^{\text{local}}(\mathbf{W}(i)) = \sum_{l \in N_n} c_{l,n} \mathrm{E}\Big[(e_l(i))^2\Big] = \sum_{l \in N_n} c_{l,n} \mathrm{E}\Big[\Big(d_l(i) - \mathbf{X}_l(i)\mathbf{W}^{\mathrm{T}}(i)\Big)^2\Big] \quad (11)$$

The set of distributed network nodes connected to the *n-th node* (including the *n-th node* itself) is denoted by N_n and is called the neighborhood of distributed network nodes n. The weighting coefficients $c_{l,n}$ are real and satisfy $\sum_{l \in N_n} c_{l,n} = 1$. $c_{l,n}$ forms a nonnegative combination matrix $\boldsymbol{C}$.

Cattivelli and colleagues analyzed ATC and CTA, and the ATC is better than the CTA [22]. So, using the ATC scheme, there are two steps in the DLMS algorithm: adaptation and combination. The order of these two steps is as follows

$$\begin{cases} \boldsymbol{\varphi}_{\mathrm{n}}(i) = \mathbf{W}_{\mathrm{n}}(i-1) + \mu_{\mathrm{n}}\mathbf{X}_{\mathrm{n}}(i)e_{\mathrm{n}}(i) \\ \mathbf{W}_{\mathrm{n}}(i) = \sum\limits_{l \in N_{\mathrm{n}}} c_{l,n}\boldsymbol{\varphi}_{\mathrm{l}}(i) \end{cases} \quad (12)$$

where μ is the step size (learning rate), and $\boldsymbol{\varphi}_n(i)$ is the local estimates at distributed network node n.

They were combining Eqs. (6)~(8) and Eq. (15), and three asymmetric adaptive diffusion filtering algorithms are designed as follows.

2.2.1 The DLLCLMS Algorithm

Combining Eqs. (6) and (15), a summary of the DLLCLMS algorithm procedure based on the above analysis is given in Table 1. From Table 1, we know the DLLCLMS algorithm can be regarded as a general algorithm structure of the DSELMS algorithm. If $a = b$, the DLLCLMS algorithm is the DSELMS algorithm. In other words, the DLLCLMS can be seen as a mixture of a DSELMS algorithm for different estimated error $e(i)$ at a different network node and dynamic switching according to the relationship between $e(i)$ and 0.

Table 1 The DLLCLMS algorithm summary

Initialize $\left\{\boldsymbol{w}_{\mathrm{n},-1} = 0\right\}$ for all agents$n$, a, b

Set nonnegative combination weights $c_{\mathrm{l,n}}$ for each time $i \geq 0$ and each agent n, and repeat:

$$\boldsymbol{\varphi}_n(i) = \begin{cases} \mathbf{W}_{\mathrm{n}}(i-1) + \mu_{\mathrm{n}} a \operatorname{sign}(e_{\mathrm{n}}(i))\mathbf{X}_{\mathrm{n}}(i) & , \text{if} e_{\mathrm{n}}(i) > 0 (I) \\ \mathbf{W}_{\mathrm{n}}(i-1) + \mu_{\mathrm{n}} b \operatorname{sign}(e_{\mathrm{n}}(i))\mathbf{X}_{\mathrm{n}}(i) & , \text{if} e_{\mathrm{n}}(i) \leq b(II) \end{cases} \quad (16)$$

$\mathbf{W}_{\mathrm{n}}(i) = \sum_{l \in N_{\mathrm{n}}} c_{\mathrm{l,n}}\boldsymbol{\varphi}_{\mathrm{l}}(i)$ (17)

Table 2 The DQQCLMS algorithm summary

Initialize $\left\{\boldsymbol{w}_{\mathrm{n},-1}=0\right\}$ for all agents$n$, a, b

Set nonnegative combination weights $c_{l,n}$ for each time $i \geq 0$ and each agent n, and repeat:

$$\boldsymbol{\varphi}_n(i)=\begin{cases}\mathbf{W}_{\mathrm{n}}(i-1)+\mu_{\mathrm{n}} a e_{\mathrm{n}}(i) \mathbf{X}_{\mathrm{n}}(i) & \text{, if} e_{\mathrm{n}}(i)>0(I) \\ \mathbf{W}_{\mathrm{n}}(i-1)+\mu_{\mathrm{n}} b e_{\mathrm{n}}(i) \mathbf{X}_{\mathrm{n}}(i) & \text{, if} e_{\mathrm{n}}(i) \leq b(II)\end{cases} \quad (18)$$

$\mathbf{W}_{\mathrm{n}}(i)=\sum_{l \in N_{\mathrm{n}}} c_{\mathrm{l,n}} \boldsymbol{\varphi}_{\mathrm{l}}(i)$ (19)

2.2.2 The DQQCLMS Algorithm

Combining Eqs. (7) and (15), a summary of the DQQCLMS algorithm procedure based on the above analysis is given in Table 2. From Table 2, we know the DLLCLMS algorithm can be regarded as a general algorithm structure of the DLMS algorithm. If $a = b$, the DLLCLMS algorithm is the DLMS algorithm. In other words, the DLLCLMS can be seen as a mixture of a DLMS algorithm for different estimated error $e(i)$ at a different network node and dynamic switching according to the relationship between $e(i)$ and 0.

2.2.3 The DLECLMS Algorithm

Combining Eqs. (8) and (15), a summary of the DLECLMS algorithm procedure based on the above analysis is given in Table 3. Table 3 shows that the DLECLMS algorithm can be an adaptive filter based on an asymmetric exponential function estimator and extend in a distributed network (Table 4).

3 Performance of Proposed Diffusion Algorithms

After completing those three diffusion adaptive filtering algorithms design, the performance of those three algorithms should be analyzed theoretically. This subsection will discuss the performances of the diffusion asymmetric adaptive filtering algorithms, including mean behavior and computational complexity.

To facilitate performance analysis, we make the following assumptions:

Assumption 1 The distributed network system measurement noises are independent of other signals.

Table 3 The DLECLMS algorithm summary

Initialize $\left\{\boldsymbol{w}_{n,-1}=0\right\}$ for all agents$n$, a, b

Set nonnegative combination weights $c_{l,n}$ for each time $i \geq 0$ and each agent n, and repeat:

$\boldsymbol{\varphi}_{\mathrm{n}}(i)=\mathbf{W}_{\mathrm{n}}(i-1)+\mu_n a b((\exp(a e_{\mathrm{n}}(i))-1)) \mathbf{X}_{\mathrm{n}}(i)$ (20)

$\mathbf{W}_{\mathrm{n}}(i)=\sum_{l \in N_n} c_{l,n} \boldsymbol{\varphi}_l(i)$ (21)

Table 4 The computational complexity of the DSELMS, DRVSSLMS, DLLAD, and three proposed diffusion adaptive filtering algorithms

Algorithm	Computational cost per iteration					
	Recursion	×	+	$\vert\vert$	sign(·)	exp(·)
DSELMS	Equation 1a in [27]	$(2M + 1)N + M$	$(3M\text{-}1)N$	0	N	0
	Equation 1b in [27]	NM	$(N\text{-}1)M$	0	0	0
DRVSSLMS	Equation 11 in [17]	$> (3M + 1)N + M$	$(3M\text{-}1)N$	0	0	0
	Equation 11 in [17]	$> (3M + 1)N + M$	$(3M\text{-}1)N$	0	N	0
	Equation 12 in [17]	NM	$(N\text{-}1)M$	0	0	0
DLLAD	Equation 16 in [5]	$2MN + M$	$(3M\text{-}1)N$	N	0	0
	Equation 17 in [5]	NM	$(N\text{-}1)M$	0	0	0
DLLCLMS	Equation 16(I) in this paper	$(2M + 2)N + M$	$(3M\text{-}1)N$	0	N	0
	Equation 16(II) in this paper	$(2M + 2)N + M$	$(3M\text{-}1)N$	0	N	0
	Equation 17 in this paper	NM	$(N\text{-}1)M$	0	0	0
DQQCLMS	Equation 18(I) in this paper	$(2M + 3)N + M$	$(3M\text{-}1)N$	0	N	0
	Equation 18(II) in this paper	$(2M + 3)N + M$	$(3M\text{-}1)N$	0	N	0
	Equation 19 in this paper	NM	$(N\text{-}1)M$	0	0	0
DLECLMS	Equation 20 in this paper	$(2M + 5)N + M$	$3MN$	0	0	N
	Equation 21 in this paper	NM	$(N\text{-}1)M$	0	0	0

where: “×” denotes “Multiplications”; “ > ” denotes “larger than”; “ + ” denotes “Additions”; “||” denotes “Absolute”

Assumption 2 $X(i)$ is zero-mean Gaussian, temporally white, and spatially independent with $\boldsymbol{R}_{xx,n} = E\left[\boldsymbol{X}_{\mathbf{n}}(i)\boldsymbol{X}_{\mathbf{n}}{}^{T}(i)\right]$.

Assumption 3 The regression vector $\boldsymbol{X}_n(i)$ is independent of $\widehat{\boldsymbol{W}}_n(j)$ for all distributed networks n and $j < i$. All distributed network system weight vectors are approximately independent of all input signals.

Assumption 4 The distributed network system measurement noises $v_{\mathrm{n}}(i)$ at the$n$-th agent is assumed to be a mixture signal of zero-mean white Gaussian noise $g_{\mathrm{n}}(i)$ of variance $\sigma_{g,n}^2$, and impulsive noise $Im_{\mathrm{n}}(i)$, i.e., $v_{\mathrm{n}}(i) = g_{\mathrm{n}}(i) + Im_{\mathrm{n}}(i)$. The impulsive noise can be described using $Im_{\mathrm{n}}(i) = B_{\mathrm{n}}(i)G_{\mathrm{n}}(i)$, where $B_{\mathrm{n}}(i)$ is a Bernoulli process with the probability of $P[B_{\mathrm{n}}(i) = 1] = P_{\mathrm{r}}$ and $P[B_{\mathrm{n}}(i) = 0] = 1 - P_{\mathrm{r}}$, and $G_{\mathrm{n}}(i)$ is a zero-mean white Gaussian process of variance $I_{\mathrm{n}}\sigma_{g,n}^2$ with $I_{\mathrm{n}} \gg 1$.

Then, let us define some equations at agent n and time i, $\widehat{\mathbf{W}}_{\mathrm{n}}(i) = \mathbf{W}^{\mathrm{o}} - \mathbf{W}_{\mathrm{n}}(i)$,$\widehat{\boldsymbol{\varphi}}_{\mathrm{n}}(i) = \mathbf{W}^{\mathrm{o}} - '_{\mathrm{n}}(i)$, which are then collected to form the network weight error vector and intermediate network weight error vector, i.e., $\mathbf{W}(i) = \mathrm{col}\{\mathbf{W}_1(i), \mathbf{W}_2(i), \cdots, \mathbf{W}_N(i)\}$, $'(i) = \mathrm{col}\left\{\boldsymbol{\varphi}_1(i), \boldsymbol{\varphi}_2(i), \cdots, \boldsymbol{\varphi}_N(i)\right\}$, $\widehat{\mathbf{W}}(i) = \mathrm{col}\left\{\widehat{\mathbf{W}}_1(i), \widehat{\mathbf{W}}_2(i), \cdots, \widehat{\mathbf{W}}_N(i)\right\}$, $\widehat{\boldsymbol{\varphi}}(i) = \mathrm{col}\left\{\widehat{\boldsymbol{\varphi}}_1(i), \widehat{\boldsymbol{\varphi}}_2(i), \cdots, \widehat{\boldsymbol{\varphi}}_N(i)\right\}$, $\boldsymbol{\mu}_{\boldsymbol{A}} = \mathrm{diag}\left\{a\mu_1, a\mu_2, \cdots, a\mu_N\right\}$, $\boldsymbol{\mu}_{\boldsymbol{B}} = \mathrm{diag}\left\{b\mu_1, b\mu_2, \cdots, b\mu_N\right\}$, $\boldsymbol{\mu}_{\boldsymbol{AB}} = \mathrm{diag}\left\{ab\mu_1, ab\mu_2, \cdots, ab\mu_N\right\}$, and $\mathbf{e}(i) = \mathrm{col}\{e_1(i), e_2(i), \cdots, e_N(i)\}$.

3.1 Mean Weight Vector Error Behavior

Two noteworthy performances of adaptive filtering algorithms are convergence and steady-state characteristics. So, by studying the mean weight estimation error vector, the convergence and steady-state error properties of those three proposed diffusion adaptive filtering algorithms can be explored. The following will analyze the mean behavior performance of these three diffusion algorithms.

3.1.1 The DLLCLMS algorithm

Equations (16) and (17) can be written as

$$\hat{\boldsymbol{\varphi}}(i) = \begin{cases} \hat{\mathbf{W}}(i-1) - S_{\mu_{\mathrm{A}}}\mathbf{S}_{\mathbf{S}_{\mathbf{e}}}(i)\mathbf{X}(i) & , \text{ if } \mathbf{e}(i) > 0\,(I) \\ \hat{\mathbf{W}}(i-1) - S_{\mu_{\boldsymbol{B}}}\mathbf{S}_{\mathbf{S}_{\mathbf{e}}}(i)\mathbf{X}(i) & , \text{ if } \mathbf{e}(i) \le 0\,(II) \end{cases} \tag{13}$$

$$\widehat{\mathbf{W}}(i) = \mathbf{C}^{\mathrm{T}}\widehat{\boldsymbol{\varphi}}(i) \tag{14}$$

where $\mathbf{C} = \mathbf{C} \otimes \mathbf{I}$, $S_{\mu_{\boldsymbol{A}}} = \boldsymbol{\mu}_{\boldsymbol{A}} \otimes \mathbf{I}$, $\mathbf{S}_{\mathbf{S}_{\mathbf{e}}}(i) = \mathbf{S}_{\mathbf{e}}(i) \otimes \mathbf{I}$, $\mathbf{X}(i) = \mathrm{col}\{\mathbf{X}_1(i), \mathbf{X}_2(i), \cdots, \mathbf{X}_N(i)\}$, $\mathbf{S}_{\mathbf{e}}(i) = \mathrm{diag}\{\mathrm{sign}(\mathbf{e}(i))\}$, and $\otimes$ denotes the Kronecker product operation.

Taking the expectation of Eq. (22) and Eq. (23),

$$\mathrm{E}\left[\hat{\mathbf{W}}(i)\right] = \begin{cases} \mathbf{C}^{\mathrm{T}}\mathbf{E}\left[\hat{\mathbf{W}}(i-1)\right] - \mathbf{C}^{\mathrm{T}}S_{\mu_{\boldsymbol{A}}}\mathbf{E}\left[\mathbf{S}_{\mathbf{S}_{\mathbf{e}}}(i)\mathbf{X}(i)\right] & , if \quad \mathbf{e}(i) > 0 \quad (I) \\ \mathbf{C}^{\mathrm{T}}\mathbf{E}\left[\hat{\mathbf{W}}(i-1)\right] - \mathbf{C}^{\mathrm{T}}S_{\mu_{\boldsymbol{B}}}\mathbf{E}\left[\mathbf{S}_{\mathbf{S}_{\mathbf{e}}}(i)\mathbf{X}(i)\right] & , if \quad \mathbf{e}(i) \le 0 \quad (II) \end{cases} \tag{15}$$

Denote the measurement noise vector by $\mathbf{V}(i) = \mathrm{col}\{v_1(i), v_2(i), \cdots, v_N(i)\}$, $\mathbf{g}(i) = \mathrm{col}\{g_1(i), g_2(i), \cdots, g_N(i)\}$, $\mathbf{Im}(i) = \mathrm{col}\{Im_1(i), Im_2(i), \cdots, Im_N(i)\}$, $\mathbf{S}_{\mathbf{g}}(i) = \mathrm{diag}\{\mathrm{sign}(\mathbf{g}(i))\}$, $\mathbf{S}_{\mathbf{Im}}(i) = \mathrm{diag}\{\mathrm{sign}(\mathbf{Im}(i))\}$, $\mathbf{S}_{\boldsymbol{X}}(i) = \mathrm{diag}\{\mathbf{X}_1(i), \mathbf{X}_2(i), \cdots, \mathbf{X}_N(i)\}$. So, from Eq. (1), we have $\mathbf{e}(i) = \mathbf{S}_{\boldsymbol{X}}{}^{\mathrm{T}}(i)\widehat{\mathbf{W}}(i-1) + \mathbf{V}(i) = \mathbf{e}_{\boldsymbol{o}}(i) + \mathbf{V}(i)$.

Then, let $\begin{cases} \mathbf{e}_{\boldsymbol{g}}(i) = \mathbf{e}_{\boldsymbol{o}}(i) + g(i) \\ \mathbf{e}_{\boldsymbol{Im}(i)} = \mathbf{e}_{\boldsymbol{o}}(i) + Im(i) \end{cases}, \begin{cases} S_{\mathbf{S}_{g}}(i) = \mathbf{S}_{\boldsymbol{g}}(i) \otimes I \\ S_{\mathbf{S}_{Im}}(i) = \mathbf{S}_{\boldsymbol{Im}}(i) \otimes I \end{cases}.$

So,

$$\mathbf{E}\left[\mathbf{S}_{\mathbf{S_e}}(i)\mathbf{X}(i)\right] = (1 - P_r)\mathbf{E}\left[\mathbf{S}_{\mathbf{S_g}}(i)\mathbf{X}(i)\right] + P_r\mathbf{E}\left[S_{\mathbf{S}_{\boldsymbol{Im}}}(i)\mathbf{X}(i)\right] \tag{16}$$

Let

$$\begin{cases} \sigma^2_{e_{g,n}}(i) = Tr\left[\mathbf{R}_{ww,n}(i-1)\mathbf{R}_{xx,n}\right] + \sigma^2_{g,n}(i) \\ \sigma^2_{e_{Im,n}}(i) = Tr\left[\mathbf{R}_{ww,n}(i-1)\mathbf{R}_{xx,n}\right] + \sigma^2_{Im,n}(i) \end{cases},$$

$$\begin{cases} \mathbf{X}_{\mathrm{g}}(i) = \sqrt{\frac{2}{\pi}}\,\mathrm{diag}\left\{\sigma^{-1}_{g,1}(i), \sigma^{-1}_{g,2}(i), \cdots, \sigma^{-1}_{g,N}(i)\right\} \\ \mathbf{X}_{\boldsymbol{Im}}(i) = \sqrt{\frac{2}{\pi}}\,\mathrm{diag}\left\{\sigma^{-1}_{Im,1}(i), \sigma^{-1}_{Im,2}(i), \cdots, \sigma^{-1}_{Im,N}(i)\right\} \end{cases},$$

$$\begin{cases} S_{\mathbf{X}_{\boldsymbol{g}}}(i) = \mathbf{X}_{\boldsymbol{g}}(i) \otimes I \\ S_{\mathbf{X}_{\boldsymbol{Im}}}(i) = \mathbf{X}_{\boldsymbol{Im}}(i) \otimes I \end{cases}.$$

Then,

$$\begin{cases} E\left[\mathbf{S}_{\mathbf{S_g}}(i)\mathbf{X}(i)|\mathbf{W}(i-1)\right] = \mathbf{X}_{\boldsymbol{g}}(i)\,\mathrm{diag}\left\{\mathbf{R}_{xx,1}, \mathbf{R}_{xx,2}, \cdots, \mathbf{R}_{xx,N}\right\}\hat{\mathbf{W}}(i-1) \\ E\left[\mathbf{S}_{\mathbf{S_{Im}}}(i)\mathbf{X}(i)|\mathbf{W}(i-1)\right] = \mathbf{X}_{\boldsymbol{Im}}(i)\,\mathrm{diag}\left\{\mathbf{R}_{xx,1}, \mathbf{R}_{xx,2}, \cdots, \mathbf{R}_{xx,N}\right\}\hat{\mathbf{W}}(i-1) \end{cases} \tag{17}$$

Substituting (26) into (25), we have

$$\begin{aligned} &\mathbf{E}\left[\mathbf{S}_{\mathbf{S_e}}(i)\mathbf{X}(i)\right] \\ &\approx \mathbf{E}\left[\mathbf{E}\left[\mathbf{S}_{\mathbf{S_e}}(i)\mathbf{X}(i)|\mathbf{W}(i-1)\right]\right] \\ &= \left[(1-P_r)\mathbf{S}_{\mathbf{S_g}}(i) + P_r\mathbf{S}_{\mathbf{S_{Im}}}(i)\right]\mathrm{diag}\left\{\mathbf{R}_{xx,1}, \mathbf{R}_{xx,2}, \cdots, \mathbf{R}_{xx,N}\right\}\mathbf{E}\left[\hat{\mathbf{W}}(i-1)\right] \end{aligned} \tag{18}$$

Finally, substitute (27) with (24) obtains

$$\begin{aligned} &\mathrm{E}\left[\hat{\mathbf{W}}(i)\right] \\ &= \begin{cases} \mathbf{C}^{\mathrm{T}}\left[\boldsymbol{I}_{\boldsymbol{NM}} - S_{\text{-}_{\boldsymbol{A}}}\left((1-P_r)\mathbf{S}_{\mathbf{S_g}}(i) + P_r\mathbf{S}_{\mathbf{S_{Im}}}(i)\right)\mathrm{diag}\left\{\mathbf{R}_{xx,1}, \mathbf{R}_{xx,2}, \cdots, \mathbf{R}_{xx,N}\right\}\right]\mathbf{E}\left[\hat{\mathbf{W}}(i-1)\right], \text{ if } \mathbf{e}(i) > 0 \ (I) \\ \mathbf{C}^{\mathrm{T}}\left[\boldsymbol{I}_{\boldsymbol{NM}} - S_{\text{-}_{\boldsymbol{B}}}\left((1-P_r)\mathbf{S}_{\mathbf{S_g}}(i) + P_r\mathbf{S}_{\mathbf{S_{Im}}}(i)\right)\mathrm{diag}\left\{\mathbf{R}_{xx,1}, \mathbf{R}_{xx,2}, \cdots, \mathbf{R}_{xx,N}\right\}\right]\mathbf{E}\left[\hat{\mathbf{W}}(i-1)\right], \text{ if } \mathbf{e}(i) \le 0 \ (II) \end{cases} \end{aligned} \tag{19}$$

From Eq. (28), one can see that the asymptotic unbiasedness of the DLLCLMS algorithm can be guaranteed if the matrix $\mathbf{C}^{\mathrm{T}}\left[\boldsymbol{I}_{\boldsymbol{NM}} - S_{\mu_{\boldsymbol{A}}}\left((1-P_r)\mathbf{S}_{\mathbf{S_g}}(i) + P_r\mathbf{S}_{\mathbf{S_{Im}}}(i)\right)\mathrm{diag}\left\{\mathbf{R}_{xx,1}, \mathbf{R}_{xx,2}, \cdots, \mathbf{R}_{xx,N}\right\}\right]$, and $\mathbf{C}^{\mathrm{T}}\left[\boldsymbol{I}_{\boldsymbol{NM}} - S_{\mu_{\boldsymbol{B}}}\left((1-P_r)\mathbf{S}_{\mathbf{S_g}}(i) + P_r\mathbf{S}_{\mathbf{S_{Im}}}(i)\right)\mathrm{diag}\left\{\mathbf{R}_{xx,1}, \mathbf{R}_{xx,2}, \cdots, \mathbf{R}_{xx,N}\right\}\right]$ are stable. Both of the matrix $\left[\boldsymbol{I}_{\boldsymbol{NM}} - S_{\mu_{\boldsymbol{A}}}\left((1-P_r)\mathbf{S}_{\mathbf{S_g}}(i) + P_r\mathbf{S}_{\mathbf{S_{Im}}}(i)\right)\mathrm{diag}\{\mathbf{R}_{xx,1}, \mathbf{R}_{xx,2}, \cdots, \mathbf{R}_{xx,N}\}\right]$ and $\left[\boldsymbol{I}_{\boldsymbol{NM}} - \boldsymbol{S}_{\mu_{\boldsymbol{B}}}\left((1-P_r)\mathbf{S}_{\mathbf{S_g}}(i) + P_r\mathbf{S}_{\mathbf{S_{Im}}}(i)\right)\mathrm{diag}\{\mathbf{R}_{xx,1}, \mathbf{R}_{xx,2}, \cdots, \mathbf{R}_{xx,N}\}\right]$ are a block-diagonal matrix, and it can be easily verified that it is stable if its block-diagonal entries $\left[\boldsymbol{I} - a\mu_n \mathrm{X}_v(i)\mathbf{R}_{xx,n}\right]$, and $\left[\boldsymbol{I} - b\mu_n \mathrm{X}_v(i)\mathbf{R}_{xx,n}\right]$ are stable, where $\mathrm{X}_v(i) \le \sqrt{\frac{2}{\pi}}\left[(1-P_r)\sigma^{-1}_{g,1}(i) + P_r\sigma^{-1}_{Im,1}(i)\right]$. So, the condition for stability of the mean weight error vector is given by

$$\begin{cases} 0 < \mu_{\mathrm{n}} < \frac{2}{a\mathrm{X}_v(i)\rho\mathrm{max}(\mathbf{R}_{xx,n})}, \text{ if } e_{\mathrm{n}}(i) > 0\,(I) \\ 0 < \mu_{\mathrm{n}} < \frac{2}{b\mathrm{X}_v(i)\rho\mathrm{max}(\mathbf{R}_{xx,n})}, \text{ if} e_{\mathrm{n}}(i) \le 0\,(II) \end{cases} \tag{20}$$

where ρ_{max} denotes the maximal eigenvalue of $\mathbf{R}_{xx,n}$. So, based on Eq. (29) and Eq. (24), we obtain $\mathrm{E}[\widehat{\mathbf{W}}(\infty)] = 0$.

3.1.2 The DQQCLMS Algorithm

Equations (18) and (19) can be written as

$$\hat{\boldsymbol{\varphi}}(i) = \begin{cases} \hat{\mathbf{W}}(i-1) - \boldsymbol{S}_{\boldsymbol{\mu}_A}\mathbf{S}_{\mathbf{S}_\mathbf{e}}(i)\mathbf{X}(i), & \text{if } \mathbf{e}(i) > 0\,(I) \\ \hat{\mathbf{W}}(i-1) - \boldsymbol{S}_{\boldsymbol{\mu}_B}\mathbf{S}_{\mathbf{S}_\mathbf{e}}(i)\mathbf{X}(i), & \text{if } \mathbf{e}(i) \le 0\,(II) \end{cases} \tag{21}$$

$$\widehat{\mathbf{W}}(i) = \mathbf{C}^{\mathrm{T}}\widehat{\boldsymbol{\varphi}}(i) \tag{22}$$

where $\mathbf{C} = \mathbf{C} \otimes \mathbf{I}$, $\boldsymbol{S}_{\boldsymbol{\mu}_A} = \boldsymbol{\mu}_A \otimes \mathbf{I}$, $\mathbf{S}_{\mathbf{S}_\mathbf{e}}(i) = \mathbf{S}_\mathbf{e}(i) \otimes \mathbf{I}$, $\mathbf{X}(i) = \mathrm{col}\{\mathbf{X}_1(i), \mathbf{X}_2(i), \cdots, \mathbf{X}_N(i)\}, \mathbf{S}_\mathbf{e}(i) = \mathrm{diag}\{\mathbf{e}(i)\}$, and $\otimes$ denotes the Kronecker product operation.

Taking the expectation of Eqs. (21) and (22),

$$\mathrm{E}\Big[\hat{\mathbf{W}}(i)\Big] = \begin{cases} \mathbf{C}^{\mathrm{T}}\mathbf{E}\Big[\hat{\mathbf{W}}(i-1)\Big] - \mathbf{C}^{\mathrm{T}}\boldsymbol{S}_{\boldsymbol{\mu}_A}\mathbf{E}\big[\mathbf{S}_{\mathbf{S}_\mathbf{e}}(i)\mathbf{X}(i)\big], \text{ if } \mathbf{e}(i) > 0\,(I) \\ \mathbf{C}^{\mathrm{T}}\mathbf{E}\Big[\hat{\mathbf{W}}(i-1)\Big] - \mathbf{C}^{\mathrm{T}}\boldsymbol{S}_{\boldsymbol{\mu}_B}\mathbf{E}\big[\mathbf{S}_{\mathbf{S}_\mathbf{e}}(i)\mathbf{X}(i)\big], \text{ if } \mathbf{e}(i) \le 0\,(II) \end{cases} \tag{23}$$

Denote the measurement noise vector by $\mathbf{V}(i) = \mathrm{col}\{v_1(i), v_2(i), \cdots, v_N(i)\}$, $\mathbf{g}(i) = \mathrm{col}\{g_1(i), g_2(i), \cdots, g_N(i)\}$, $\mathbf{Im}(i) = \mathrm{col}\{Im_1(i), Im_2(i), \cdots, Im_N(i)\}$**,** $\mathbf{S}_\mathbf{g}(i) = \mathrm{diag}\{\mathbf{g}(i)\}$**,** $\mathbf{S}_{\mathbf{Im}}(i) = \mathrm{diag}\{\mathbf{Im}(i)\}$**,**$\mathbf{S}_X(i) = \mathrm{diag}\{\mathbf{X}_1(i), \mathbf{X}_2(i), \cdots, \mathbf{X}_N(i)\}$. So, from Eq. (1), we have $\mathbf{e}(i) = \mathbf{S}_X{}^{\mathrm{T}}(i)\widehat{\mathbf{W}}(i-1) + \mathbf{V}(i) = \mathbf{e}_{\boldsymbol{o}}(i) + \mathbf{V}(i)$.

Then, let $\begin{cases} \mathbf{e}_{\boldsymbol{g}}(i) = \mathbf{e}_{\boldsymbol{o}}(i) + g(i) \\ \mathbf{e}_{\boldsymbol{Im}(i)} = \mathbf{e}_{\boldsymbol{o}}(i) + Im(i) \end{cases}, \begin{cases} S_{\mathbf{S}_{\boldsymbol{g}}}(i) = \mathbf{S}_{\boldsymbol{g}}(i) \otimes I \\ S_{\mathbf{S}_{\boldsymbol{Im}}}(i) = \mathbf{S}_{\boldsymbol{Im}}(i) \otimes I \end{cases}.$

So,

$$\mathrm{E}\Big[\hat{\mathbf{W}}(i)\Big] = \begin{cases} \mathbf{C}^{\mathrm{T}}\big[\boldsymbol{I}_{NM} - \boldsymbol{S}_{\boldsymbol{\mu}_A}\mathrm{diag}\big\{\mathbf{R}_{xx,1}, \mathbf{R}_{xx,2}, \cdots, \mathbf{R}_{xx,N}\big\}\big]\mathbf{E}\Big[\hat{\mathbf{W}}(i-1)\Big], \text{ if } \mathbf{e}(i) > 0\,(I) \\ \mathbf{C}^{\mathrm{T}}\big[\boldsymbol{I}_{NM} - \boldsymbol{S}_{\boldsymbol{\mu}_B}\mathrm{diag}\big\{\mathbf{R}_{xx,1}, \mathbf{R}_{xx,2}, \cdots, \mathbf{R}_{xx,N}\big\}\big]\mathbf{E}\Big[\hat{\mathbf{W}}(i-1)\Big], \text{ if } \mathbf{e}(i) \le 0\,(II) \end{cases} \tag{24}$$

From Eq. (33), one can see that the asymptotic unbiasedness of the DQQCLMS algorithm can be guaranteed if the matrix $\mathbf{C}^{\mathrm{T}}\Big[\boldsymbol{I}_{NM} - \boldsymbol{S}_{\boldsymbol{\mu}_A}\mathrm{diag}\Big\{\mathbf{R}_{xx,1}, \mathbf{R}_{xx,2}, \cdots, \mathbf{R}_{xx,N}\Big\}\Big]$**,** and $\mathbf{C}^{\mathrm{T}}\big[\boldsymbol{I}_{NM} - \boldsymbol{S}_{\boldsymbol{\mu}_B}\mathrm{diag}\big\{\mathbf{R}_{xx,1}, \mathbf{R}_{xx,2}, \cdots, \mathbf{R}_{xx,N}\big\}\big]$ are stable. Both of the matrix $\left[\boldsymbol{I}_{\boldsymbol{NM}} - \boldsymbol{S}_{\boldsymbol{\mu}_A}\mathrm{diag}\left\{\mathbf{R}_{xx,1}, \mathbf{R}_{xx,2}, \cdots, \mathbf{R}_{xx,N}\right\}\right]$ and $\left[\boldsymbol{I}_{\boldsymbol{NM}} - \boldsymbol{S}_{\boldsymbol{\mu}_B}\mathrm{diag}\left\{\mathbf{R}_{xx,1}, \mathbf{R}_{xx,2}, \cdots, \mathbf{R}_{xx,N}\right\}\right]$ are a block-diagonal matrix, and they can be easily verified that it is stable if its block-diagonal entries $\left[\boldsymbol{I} - a\mu_n\mathbf{R}_{xx,n}\right]$**,** and $\left[\boldsymbol{I} - b\mu_n\mathbf{R}_{xx,n}\right]$ are stable. So, the condition for stability of the mean weight error vector is given by

$$\begin{cases} 0 < \mu_\mathrm{n} < \frac{2}{a\rho\max\left(\mathbf{R}_{xx,n}\right)} \ , \text{ if } e_n(i) > 0 \quad (I) \\ 0 < \mu_\mathrm{n} < \frac{2}{b\rho\max\left(\mathbf{R}_{xx,n}\right)} \ , \text{ if } e_\mathrm{n}(i) \leq 0 \quad (II) \end{cases} \tag{25}$$

where $\rho_{\max}$ denotes the maximal eigenvalue of $\mathbf{R}_{xx,n}$. So, based on Eqs. (23) and (25), we obtain $\mathrm{E}[\widehat{\mathbf{W}}(\infty)] = 0$.

3.1.3 The DLECLMS algorithm

Based on Eq. (20), let $J_{f_{DLECLMS}}(i) = \exp(ae_n(i)) - 1$, and then, this can be observed from the Taylor series expansion of $J_{f_{DLECLMS}}(i)$ around $e_\mathrm{n}(i) = 0$,

$$J_{\mathrm{f_{DLECLMS}}}(i) = \exp(ae_n(i)) - 1 = a\sum_{k=1}^{+\infty}\frac{1}{k!}e_n^k(i) \tag{26}$$

As desired, since the weight of the $k-$ th error moment is $\frac{1}{k!}$, it is given to lower-order moments. Notice also that for small error values, the error cost functions become,

$$J_{\mathrm{f_{DLECLMS}}}(i) = \exp(ae_\mathrm{n}(i)) - 1 = a\sum_{k=1}^{+\infty}\frac{1}{k!}e_n^k(i) \approx ae_\mathrm{n}(i) \tag{27}$$

Equations (20) and (21) can be written as

$$\hat{\boldsymbol{\varphi}}(i) = \hat{\mathbf{W}}(i-1) - \boldsymbol{S}_{\boldsymbol{\mu}_{AB}}\mathbf{S}_{\mathbf{S_e}}(i)\mathbf{X}(i) \tag{28}$$

$$\widehat{\mathbf{W}}(i) = \mathbf{C}^{\mathrm{T}}\widehat{\boldsymbol{\varphi}}(i) \tag{29}$$

where $\mathbf{C} = \mathbf{C} \otimes \mathbf{I}$, $\boldsymbol{S}_{\boldsymbol{\mu}_{AB}} = \boldsymbol{\mu}_{\boldsymbol{AB}} \otimes \mathbf{I}$, $\mathbf{S}_{\mathbf{S_e}}(i) = \mathbf{S_e}(i) \otimes \mathbf{I}$, $\mathbf{X}(i) = \mathrm{col}\{\mathbf{X}_1(i), \mathbf{X}_2(i), \cdots, \mathbf{X}_N(i)\}$**,** $\mathbf{S_e}(i) = \mathrm{diag}\{\exp(a\mathbf{e}(i)) - 1\} \approx \mathrm{diag}\{a\mathbf{e}(i)\}$, and $\otimes$ denotes the Kronecker product operation.

Taking the expectation of Eqs. (28) and (29),

$$\mathrm{E}\left[\widehat{\mathbf{W}}(i)\right] = \mathbf{C}^{\mathrm{T}}\mathbf{E}\left[\widehat{\mathbf{W}}(i-1)\right] - \mathbf{C}^{\mathrm{T}}\boldsymbol{S}_{\boldsymbol{\mu}_{AB}}\mathbf{E}\left[\mathbf{S}_{\mathbf{S_e}}(i)\mathbf{X}(i)\right] \tag{30}$$

Denote the measurement noise vector by $\mathbf{V}(i) = \mathrm{col}\{v_1(i), v_2(i), \cdots, v_N(i)\}$, $\mathbf{g}(i) = \mathrm{col}\{g_1(i), g_2(i), \cdots, g_N(i)\}$, $\mathbf{Im}(i) = \mathrm{col}\{Im_1(i), Im_2(i), \cdots, Im_N(i)\}$**,** $\mathbf{S_g}(i) = \mathrm{diag}\{\mathbf{g}(i)\}$**,** $\mathbf{S_{Im}}(i) = \mathrm{diag}\{\mathbf{Im}(i)\}$**,**$\mathbf{S}_{\boldsymbol{X}}(i) = \mathrm{diag}\{\mathbf{X}_1(i), \mathbf{X}_2(i), \cdots, \mathbf{X}_N(i)\}$. So, from Eq. (1), we have $\mathbf{e}(i) = \mathbf{S}_{\boldsymbol{X}}{}^{\mathrm{T}}(i)\widehat{\mathbf{W}}(i-1) + \mathbf{V}(i) = \mathbf{e}_{\boldsymbol{o}}(i) + \mathbf{V}(i)$.

Then, let $\begin{cases} \mathbf{e}_{\boldsymbol{g}}(i) = \mathbf{e}_{\boldsymbol{o}}(i) + g(i) \\ \mathbf{e}_{\mathrm{Im}(i)} = \mathbf{e}_{\boldsymbol{o}}(i) + \mathrm{Im}(i) \end{cases}, \begin{cases} \boldsymbol{S}_{\mathbf{S}_g}(i) = \mathbf{S}_{\boldsymbol{g}}(i) \otimes I \\ \boldsymbol{S}_{\mathbf{S}_{\mathrm{Im}}}(i) = \mathbf{S}_{\mathrm{Im}}(i) \otimes I \end{cases}.$

So,

$$\mathrm{E}\Big[\hat{\mathbf{W}}(i)\Big] = \mathbf{C}^{\mathrm{T}}\big[\boldsymbol{I}_{\boldsymbol{NM}} - \boldsymbol{S}_{\boldsymbol{\mu}_{\boldsymbol{AB}}}\mathrm{diag}\big\{a\mathbf{R}_{xx,1}, a\mathbf{R}_{xx,2}, \cdots, a\mathbf{R}_{xx,N}\big\}\big]\mathbf{E}\Big[\hat{\mathbf{W}}(i-1)\Big] \tag{31}$$

From Eq. (31), one can see that the asymptotic unbiasedness of the DLECLMS algorithm can be guaranteed if the matrix $\mathbf{C}^{\mathrm{T}}\big[\boldsymbol{I}_{\boldsymbol{NM}} - \boldsymbol{S}_{\boldsymbol{\mu}_{\boldsymbol{AB}}}\mathrm{diag}\big\{a\mathbf{R}_{xx,1}, a\mathbf{R}_{xx,2}, \cdots, a\mathbf{R}_{xx,N}\big\}\big]$ is stable. The matrix $\big[\boldsymbol{I}_{\boldsymbol{NM}} - \boldsymbol{S}_{\boldsymbol{\mu}_{\boldsymbol{AB}}}\mathrm{diag}\big\{a\mathbf{R}_{xx,1}, a\mathbf{R}_{xx,2}, \cdots, a\mathbf{R}_{xx,N}\big\}\big]$ is a block-diagonal matrix and can be easily verified that it is stable if its block-diagonal entries $\big[\boldsymbol{I} - a^2 b\mu_n \mathbf{R}_{xx,n}\big]$ are stable. So, the condition for stability of the mean weight error vector (as Eq. (31)) is given by

$$0 < \mu_{\mathrm{n}} < \frac{2}{a^2 b\rho_{\max}\big(\mathbf{R}_{xx,n}\big)} \tag{32}$$

where ρ_{max} denotes the maximal eigenvalue of $\mathbf{R}_{xx,n}$. So, based on Eqs. (31) and (32), we obtain $\mathrm{E}[\widehat{\mathbf{W}}(\infty)] = 0$.

3.2 Computational complexity

Another important indicator to measure the performance of an adaptive filtering algorithm is the computational complexity because it directly determines whether the adaptive filtering algorithm is easy to implement in engineering. The computational complexity of the diffusion adaptive filtering algorithm is the number of arithmetic operations per iteration of the weight vector or coefficient vector. That is the number of multiplications, additions, et al. The time-consuming procedure of the multiplication operation is far greater than the addition operation, so the multiplication operation occupies a large proportion of the diffusion adaptive filtering algorithm. Therefore, computational complexity is an important property that affects the performance of the diffusion adaptive filtering algorithm. The DLLCLMS, DQQCLMS, and DLECLMS algorithms have two parameters: a and b, during an exponential function in the DLECLMS algorithm, so the computational complexity of the DLLCLMS algorithm, the DQQCLMS algorithm, and the DLECLMS slightly larger than the DSELMS [27], DRVSSLMS [17], DLLAD [5] algorithms. Furthermore, when M increases, those algorithms have the same computational complexity.

3.3 Parameters a and b for the Proposed Algorithms

As described early, parameters a and b determine the shape and characteristics of each cost function because, as an essential core parameter, how to set it is critical. So the choice of a and b in Eqs. (16)~(21) plays a vital role in the performance of the DLLCLMS, DQQCLMS, and DLECLMS algorithms. The optimum cut-off value, a

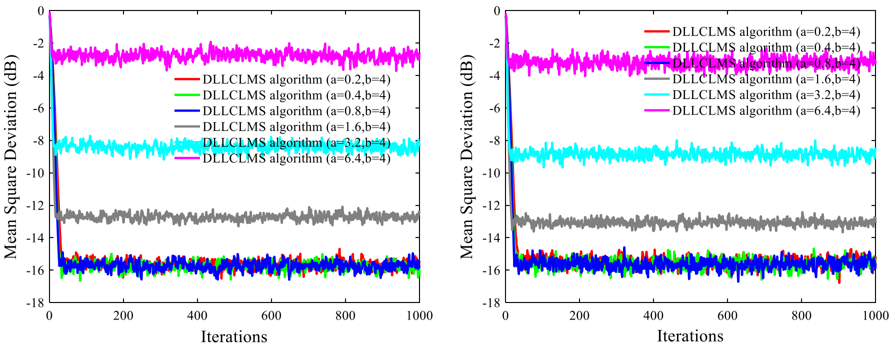

Fig. 2 MSD curve with different a of the DLLCLMS algorithm ($\mu = 0.4$) when network topology and neighbors to be decided by probability (probability = 0.2) with $\alpha = 1.6$, $\beta = 0.05$, $\gamma = 0$, and $\delta = 1000$: (Left) $\mathbf{R}_{xx,n} = \sigma_{x,n}^2 \mathbf{I}_M$. (Right) $\mathbf{R}_{xx,n} = \sigma_{x,n}^2(i)\mathbf{I}_M$, $i = 1, 2, 3, \cdots, M$

and b, under different input signals, impulsive noise, and network structures can be used for both the theoretical derivation and simulation experimental methods. For the theory derivation method, although the optimal parameters $a$and$b$ of the proposed three diffusion adaptive filtering algorithms are obtained based on minimizing the mean-square deviation (MSD) at the current time, the problem with this operation is that iterative formulas will increase the computational complexity. In this paper, the experimental simulation method will get the parameters $a$and$b$ of the proposed three diffusion adaptive filtering algorithms. The following will explore the optimal parameters $a$and$b$ of these three diffusion algorithms. In simulation experiments with an unknown linear system, we set $M = 16$, and the parameters weight vector is selected randomly. Each distributed network topology consists of $N = 20$ nodes. For impulsive noises, in [32], we can compute the impulsive noises by using the Levy alpha-stable distribution with setting $\alpha, \beta, \gamma, and \delta$ in MATLAB software (2016b). Besides, we set the impulsive noises as spatiotemporally independent. We apply the uniform rule for the adaptation weights in the combination step and the combination weights in the combination step, i.e., $c_{l,n} = 1/N_n$. We use the network MSD to evaluate the performance of diffusion adaptive filtering algorithms, where $\mathrm{MSD}(i) = \frac{1}{N}\sum_{n=1}^{N} \mathrm{E}[|\mathbf{W}_o - \mathbf{W}_n(i)|^2]$. In addition, the independent Monte Carlo number is 20, and each run has 1000 iteration numbers.

3.3.1 For the DLLCLMS Algorithm

Parameter*a* We evaluate varying a estimators based on their MSD using the DLLCLMS algorithm. The choice of a in Eq. (16) is vital in the DLLCLMS algorithm performance. Besides choosing the optimum cut-off value a under various input signals, different intensities of impulsive noises, and different network structures, we set four groups of the experiment in a system identification application. In Figs. 2 and 3, considering the convergence rate and the steady-state MSD, we know the DLLCLMS algorithm is robust for different probability densities of impulsive noises when $a = 0.8$ and $b = 6$.

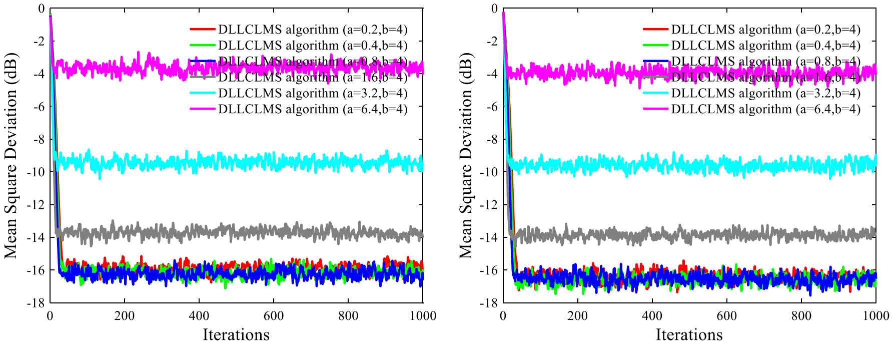

Fig. 3 MSD curve with different a of the DLLCLMS algorithm ($\mu = 0.4$) when network topology and neighbors to be decided by closeness in the distance (radius = 0.3) with $\alpha = 1.2$, $\beta = 0.05$, $\gamma = 0$, and $\delta = 1000$: (Left) $\mathbf{R}_{xx,n}$ is a diagonal matrix with possibly different diagonal entries chosen randomly. (Right)$\mathbf{R}_{xx,n} = \sigma^2_{x,n}(i)\mathbf{I}_\mathrm{M}, i = 1, 2, 3, \cdots, M$

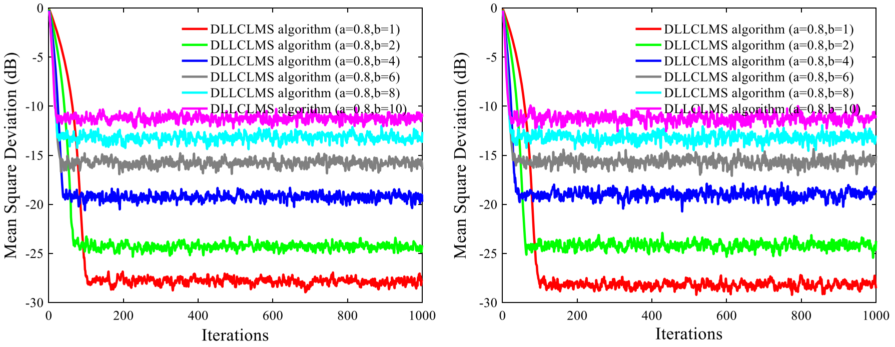

Fig. 4 MSD curve with different b of the DLLCLMS algorithm ($\mu = 0.4$) when network topology and neighbors to be decided by probability (probability = 0.2) with $\alpha = 1.6$, $\beta = 0.05$, $\gamma = 0$, and $\delta = 1000$: (Left) $\mathbf{R}_{xx,n} = \sigma^2_{x,n}\mathbf{I}_\mathrm{M}$, and Pr = 0.8. (Right) $\mathbf{R}_{xx,n} = \sigma^2_{x,n}(i)\mathbf{I}_\mathrm{M}, i = 1, 2, 3, \cdots, M$

Parameter *b* We evaluate the relative efficiency of varying$b$ estimators based on their MSD using the DLLCLMS algorithm. The choice of b in Eq. (16) is vital in the DLLCLMS algorithm performance. Besides choosing the optimum cut-off value b under different input signals, different intensities of impulsive noises, and different network structures, we set four groups of the experiment in a system identification application. In Figs. 4 and 5, considering the convergence rate and the steady-state MSD, we know the DLLCLMS algorithm is robust for different probability densities of impulsive noises when $b = 4$ and $a = 0.8$.

3.3.2 For the DQQCLMS Algorithm

Parameter*a* We evaluate varying a estimators based on their MSD using the DQQCLMS algorithm. The choice of a in Eq. (18) plays a vital role in the DQQCLMS

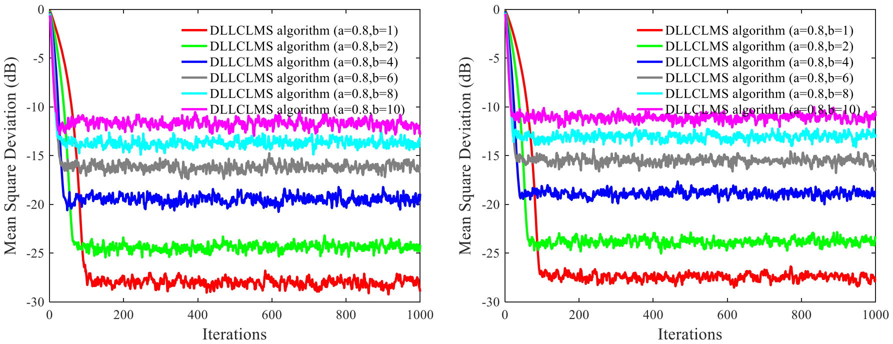

Fig. 5 MSD curve with different b of the DLLCLMS algorithm ($\mu = 0.4$) when network topology and neighbors to be decided by closeness in the distance (radius = 0.3) with $\alpha = 1.2$, $\beta = 0.05$, $\gamma = 0$, and $\delta = 1000$: (Left) $\mathbf{R}_{xx,n}$ is a diagonal matrix with possibly different diagonal entries chosen randomly. (Right)$\mathbf{R}_{xx,n} = \sigma_{x,n}^2(i)\mathbf{I}_M$, $i = 1, 2, 3, \cdots, M$

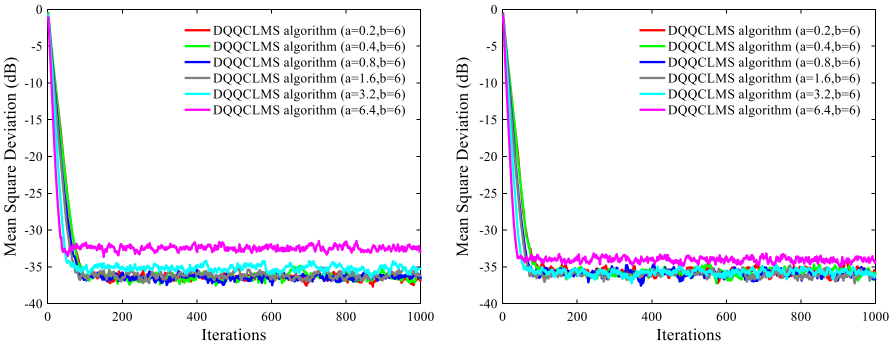

Fig. 6 MSD curve with different a of the DQQCLMS algorithm ($\mu = 0.4$) when network topology and neighbors to be decided by probability (probability = 0.2) with $\alpha = 1.6$, $\beta = 0.05$, $\gamma = 0$, and $\delta = 1000$: (Left) $\mathbf{R}_{xx,n} = \sigma_{x,n}^2\mathbf{I}_M$. (Right) $\mathbf{R}_{xx,n} = \sigma_{x,n}^2(i)\mathbf{I}_M$, $i = 1, 2, 3, \cdots, M$

algorithm's performance. Besides choosing the optimum cut-off value a under different input signals, different intensities of impulsive noises, and different network structures, we set four groups of the experiment in a system identification application. In Figs. 6 and 7, considering the convergence rate and the steady-state MSD, we know the DQQCLMS algorithm is robust for different probability densities of impulsive noises when $a = 0.8$ with $b = 6$.

Parameter *b* We evaluate the relative efficiency of varying b estimators based on their MSD using the DQQCLMS algorithm. The choice of b in Eq. (18) is vital in the DQQCLMS algorithm performance. Besides choosing the optimum cut-off value b under different input signals, different intensities of impulsive noises, and different network structures, we set four groups of the experiment in a system identification application. In Figs. 8 and 9, considering the convergence rate and the steady-state

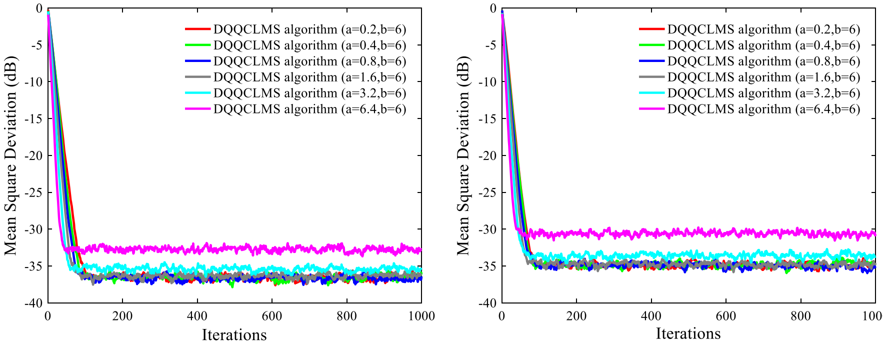

Fig. 7 MSD curve with different a of the DQQCLMS algorithm ($\mu = 0.4$) when network topology and neighbors to be decided by closeness in the distance (radius = 0.3) with $\alpha = 1.2$, $\beta = 0.05$, $\gamma = 0$, and $\delta = 1000$: (Left) $\mathbf{R}_{xx,n}$ is a diagonal matrix with possibly different diagonal entries chosen randomly. (Right)$\mathbf{R}_{xx,n} = \sigma^2_{x,n}(i)\mathbf{I}_\mathrm{M}$, $i = 1, 2, 3, \cdots, M$

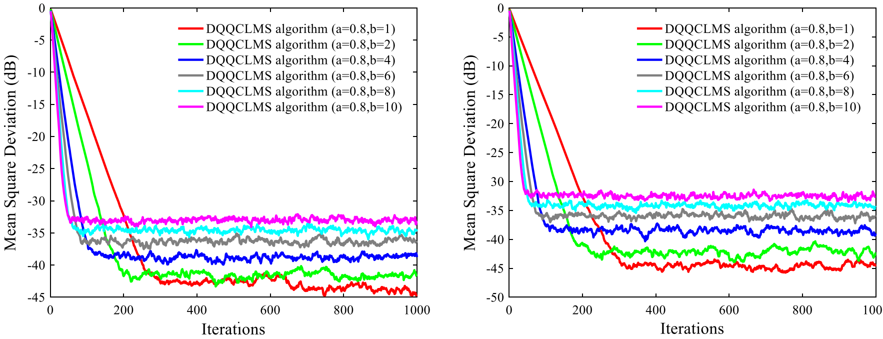

Fig. 8 MSD curve with different b of the DQQCLMS algorithm ($\mu = 0.4$) when network topology and neighbors to be decided by probability (probability = 0.2) with $\alpha = 1.6$, $\beta = 0.05$, $\gamma = 0$, and $\delta = 1000$: (Left) $\mathbf{R}_{xx,n} = \sigma^2_{x,n}\mathbf{I}_M$, and Pr = 0.8. (Right) $\mathbf{R}_{xx,n} = \sigma^2_{x,n}(i)\mathbf{I}_\mathrm{M}$, $i = 1, 2, 3, \cdots, M$

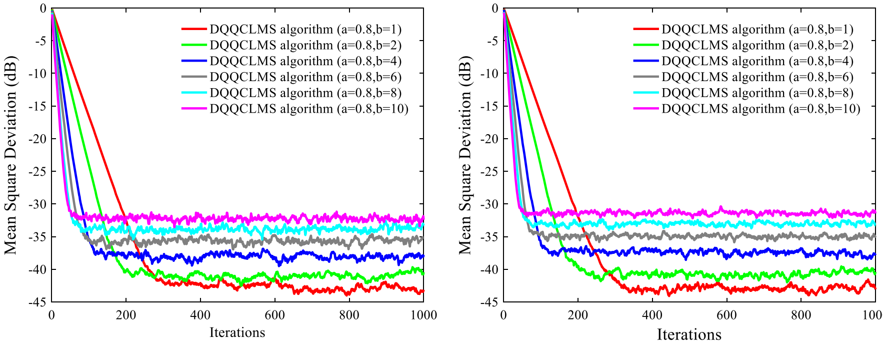

Fig. 9 MSD curve with different b of the DQQCLMS algorithm ($\mu = 0.4$) when network topology and neighbors to be decided by closeness in the distance (radius = 0.3) with $\alpha = 1.2$, $\beta = 0.05$, $\gamma = 0$, and $\delta = 1000$: (Left) $\mathbf{R}_{xx,n}$ is a diagonal matrix with possibly different diagonal entries chosen randomly. (Right)$\mathbf{R}_{xx,n} = \sigma^2_{x,n}(i)\mathbf{I}_\mathrm{M}$, $i = 1, 2, 3, \cdots, M$

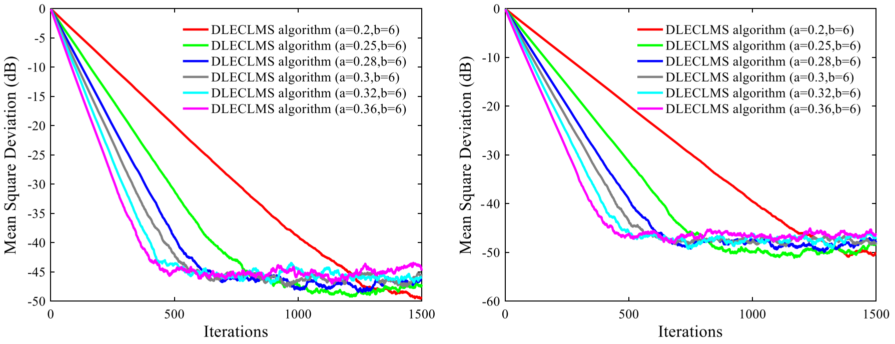

Fig. 10 MSD curve with different a of the DLECLMS algorithm ($\mu = 0.4$) when network topology and neighbors to be decided by probability (probability = 0.2) with $\alpha = 1.6$, $\beta = 0.05$, $\gamma = 0$, and $\delta = 1500$: (Left) $\mathbf{R}_{xx,n} = \sigma^2_{x,n}\mathbf{I}_M$. (Right) $\mathbf{R}_{xx,n} = \sigma^2_{x,n}(i)\mathbf{I}_M$, $i = 1, 2, 3, \cdots, M$

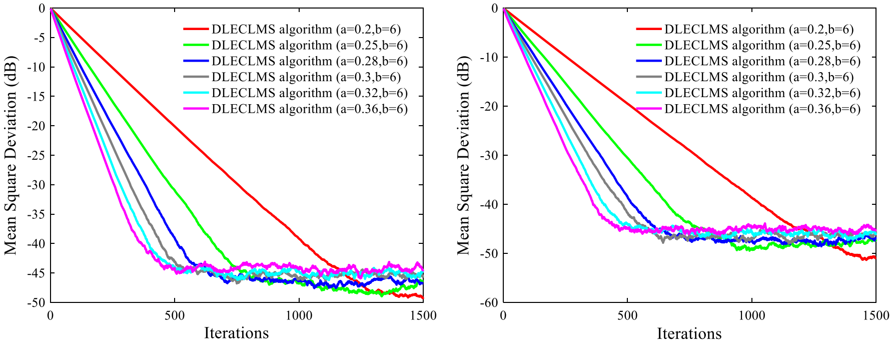

Fig. 11 MSD curve with different a of the DLECLMS algorithm ($\mu = 0.4$) when network topology and neighbors to be decided by closeness in the distance (radius = 0.3) with $\alpha = 1.2$, $\beta = 0.05$, $\gamma = 0$, and $\delta = 1500$: (Left) $\mathbf{R}_{xx,n}$ is a diagonal matrix with possibly different diagonal entries chosen randomly. (Right)$\mathbf{R}_{xx,n} = \sigma^2_{x,n}(i)\mathbf{I}_M$, $i = 1, 2, 3, \cdots, M$

MSD, we know the DQQCLMS algorithm is robust for different probability densities of impulsive noises when $b = 6$ and $a = 0.8$.

3.3.3 For the DLECLMS Algorithm

Parameter*a* We evaluate different a estimators' relative efficiency based on their MSD using the DLECLMS algorithm. The choice of a in Eq. (20) is vital in the DLECLMS algorithm performance. Besides choosing the optimum cut-off value a under different input signals, different intensities of impulsive noises, and different network structures, we set four groups of the experiment in a system identification application. In Figs. 10 and 11, considering the convergence rate and the steady-state MSD, we know the DLECLMS algorithm is robust for different probability densities of impulsive noises when $a = 0.32$ and $b = 6$.

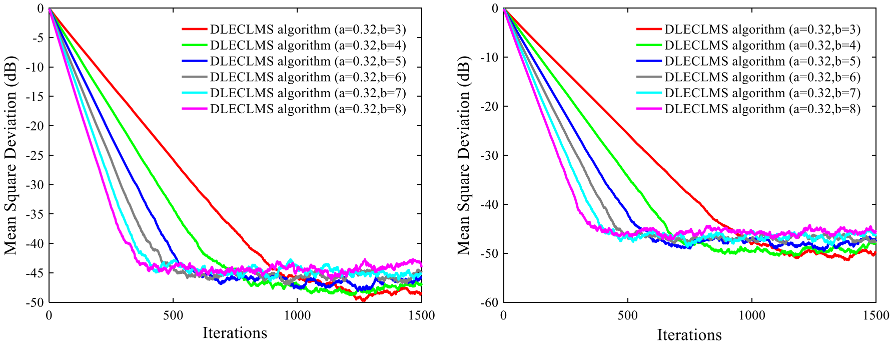

Fig. 12 MSD curve with different b of the DLECLMS algorithm ($\mu = 0.4$) when network topology and neighbors to be decided by probability (probability $= 0.2$) with $\alpha = 1.6$, $\beta = 0.05$, $\gamma = 0$, and $\delta = 1500$: (Left) $\mathbf{R}_{xx,n} = \sigma_{x,n}^2 \mathbf{I}_M$, and $\mathrm{Pr} = 0.8$. (Right) $\mathbf{R}_{xx,n} = \sigma_{x,n}^2(i)\mathbf{I}_\mathrm{M}$, $i = 1, 2, 3, \cdots, M$

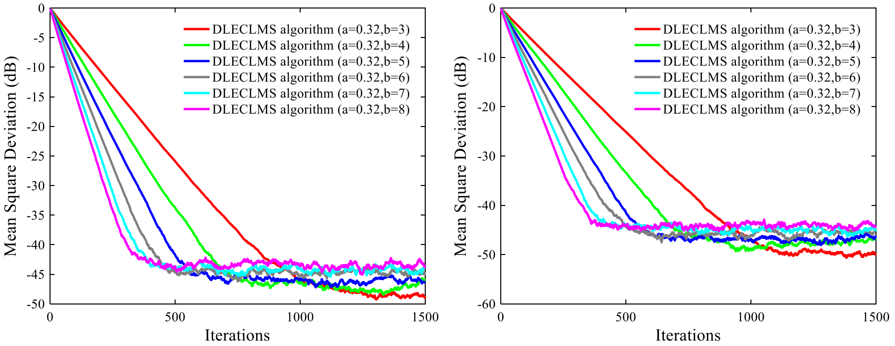

Fig. 13 MSD curve with different b of the DLECLMS algorithm ($\mu = 0.4$) when network topology and neighbors to be decided by closeness in the distance (radius $= 0.3$) with $\alpha = 1.2$, $\beta = 0.05$, $\gamma = 0$, and $\delta = 1500$: (Left) $\mathbf{R}_{xx,n}$ is a diagonal matrix with possibly different diagonal entries chosen randomly. (Right)$\mathbf{R}_{xx,n} = \sigma_{x,n}^2(i)\mathbf{I}_\mathrm{M}$, $i = 1, 2, 3, \cdots, M$

Parameter *b* We evaluate the relative efficiency of different b estimators based on their MSD using the DLECLMS algorithm. The choice of b in Eq. (20) is vital in the DLECLMS algorithm performance. Besides choosing the optimum cut-off value b under different input signals, different intensities of impulsive noises, and different network structures, we set four groups of the experiment in a system identification application. In Figs. 12 and 13, considering the convergence rate and the steady-state MSD, we know the DLECLMS algorithm is robust for different probability densities of impulsive noises when $b = 6$ and $a = 0.32$.

Based on Figs 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, considering the convergence rate and the steady-state MSD, we know that the DLLCLMS, DLLCLMS, and DLECLMS algorithm is robust for different probability densities of impulsive noises when ($a = 0.8$ and $b = 6$), ($a = 0.8$ and $b = 6$), and ($a = 0.32$ and $b = 6$), respectively.

4 Simulation Results

Because in this paper, we focus on the distributed adaptive filter algorithm and compare the DLLCLMS, DQQCLMS, and DLECLMS algorithms with the DSELMS [27], DRVSSLMS [17], and DLLAD [5] algorithms in linear system identification. Then, we want to demonstrate the robustness of the three proposed DLLCLMS, DQQCLMS, and DLECLMS algorithms in different intensities of impulsive noise and input signals. Several group simulation experiments are set with varying intensities of impulsive noises and input signal types. For an unknown linear system, we set $M = 16$, and the parameters weight vector is selected randomly. Each distributed network topology consists of $N = 20$ nodes. For impulsive noises, in [32], we can compute the impulsive noises by using the Levy alpha-stable distribution with setting $\alpha, \beta, \gamma, and \delta$. Besides, we set the impulsive noises as spatiotemporally independent. We apply the uniform rule for the adaptation weights in the combination step and the combination weights in the combination step, i.e., $c_{l,n} = 1/N_n$. We use the network MSD to evaluate the performance of diffusion adaptive filtering algorithms, where $\text{MSD}(i) = \frac{1}{N}\sum_{n=1}^{N} \text{E}[|\mathbf{W}_o - \mathbf{W}_n(i)|^2]$. In addition, the independent Monte Carlo number is 20, and each run has 2000 iteration numbers.

4.1 Simulation Experiment 1

The convergence rate is faster, and the steady-state MSD is lower to show that the proposed distributed adaptive filter algorithms are more robust to the input signal than DSELMS, DRVSSLMS, and DLLAD. Set up three experiments; both have the same network topology, the same impulsive noise, and different input signals. Suppose any two network topology nodes are declared neighbors. In that case, the connection probability is greater than or equal to 0.2. The network topology is shown in Fig. 14. For different types of the input signal, the MSD iteration curves for DRVSSLMS (μ=0.35), DSELMS (μ=0.35), DNLMS (μ=0.35), and DLLAD (μ=0.35) algorithms

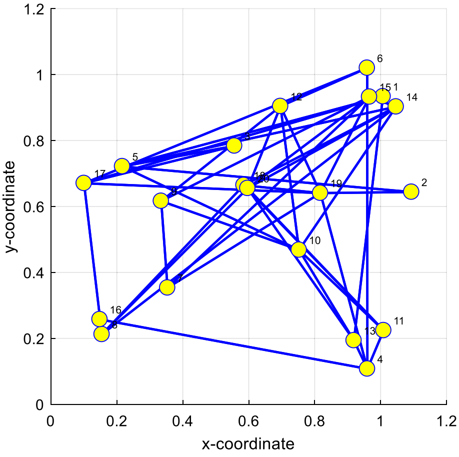

Fig. 14 Random network topology to be decided by probability

in Figs. 15, 16, and 17 when the measurement noise in an unknown system is impulsive noises with $\alpha = 1.6$, $\beta = 0.05$, $\gamma = 0$, and $\delta = 2000$.

Figures 15, 16, and 17 show that although different input signals are used, the DLLCLMS, DQQCLMS, and DLECLMS algorithms have a faster convergence rate and lower steady-state MSD than the DSELMS, DRVSSLMS, and DLLAD algorithms. Besides, the DLLCLMS, DQQCLMS, and DLECLMS algorithms are more robust to the input signal. In conclusion, from ***Simulation experiment 1***, we can get that the DLLCLMS, DQQCLMS, and DLECLMS algorithms are superior to the DSELMS, DRVSSLMS, and DLLAD algorithms. Furthermore, the order of performance superiority is DLECLMS, DQQCLMS, and DLLCLMS. This result is because the cost function designed in this paper is asymmetric, which can better track the change of estimation error due to asymmetrically distributed noise.

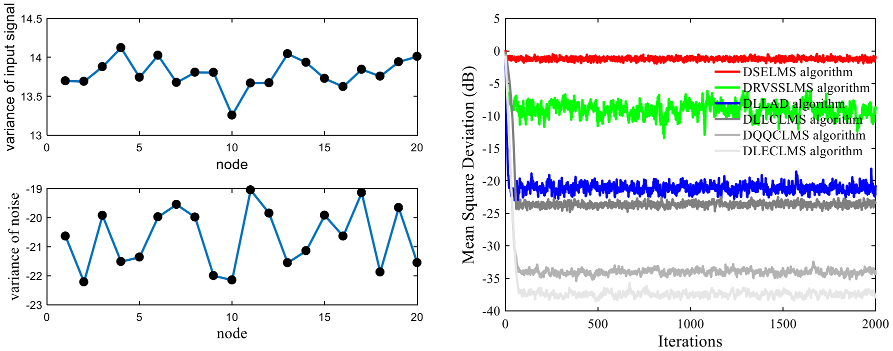

Fig. 15 (Left_top) the input signals $\{\mathbf{X}_{\mathrm{n}}(i)\}$ variances at each network node with $\mathbf{R}_{xx,n} = \sigma_{x,n}^2 \mathbf{I}_M$ with possibly different diagonal entries chosen randomly, (Left_bottom) the measurement noise variances $\varepsilon_n(i)$ at each network node; (Right) Transient network MSD (dB) iteration curves of the DSELMS, DRVSSLMS, DLLAD, DLLCLMS, DQQCLMS, and DLECLMS algorithms

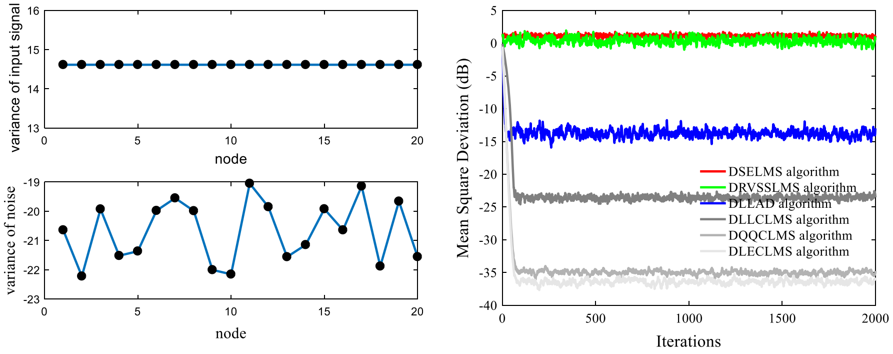

Fig. 16 (Left_top) the input signals $\{\mathbf{X}_{\mathrm{n}}(i)\}$ variances at each network node with $\mathbf{R}_{xx,n} = \sigma_{x,n}^2 \mathbf{I}_M$ with the same value in each diagonal entry, (Left_bottom) the measurement noise variances $\{\varepsilon_{\mathrm{n}}(i)\}$ at each network node; (Right) Transient network MSD (dB) iteration curves of the DSELMS, DRVSSLMS, DLLAD, DLLCLMS, DQQCLMS, and DLECLMS algorithms

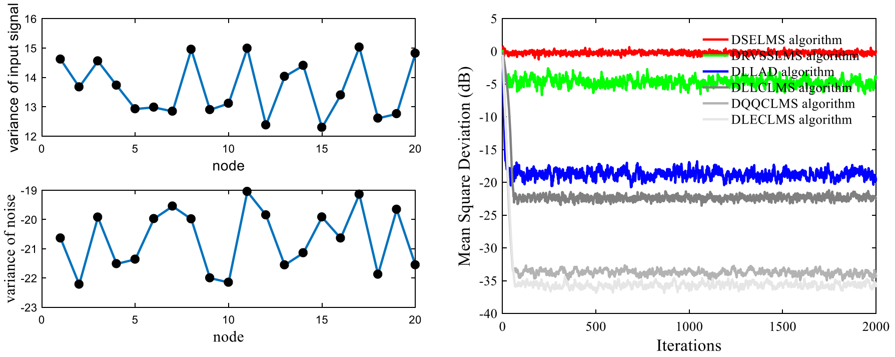

Fig. 17 (Left_top) the input signals $\{\mathbf{X}_{\text{n}}(i)\}$ variances at each network node with $\mathbf{R}_{xx,n} = \sigma_{x,n}^2(t)\mathbf{I}_M, t = 1, 2, 3, \cdots, M$ with a different value in each diagonal entry, (Left_bottom) the measurement noise variances $\{\varepsilon_{\text{n}}(i)\}$ at each network node; (Right) Transient network MSD (dB) iteration curves of the DSELMS, DRVSSLMS, DLLAD, DLLCLMS, DQQCLMS, and DLECLMS algorithms

4.2 Simulation Experiment 2

Set up three experiments; both have the same network topology, the same input signal, and different intensities of impulsive noises. If any two network topology nodes are declared neighbors, a certain radius for each node is larger than or equal to 0.3; the network topology is shown in Fig. 18(Left). The MSD iteration curves for DRVSSLMS ($\mu = 0.35$), DSELMS ($\mu = 0.35$), DNLMS ($\mu = 0.35$), DLLAD ($\mu = 0.35$), DLLCLMS, DQQCLMS, and DLECLMS algorithms in Fig. 19 with $\alpha = 1.6$, $\alpha = 1.1$, $\alpha = 0.8$, and $\alpha = 0.4$ with $\beta = 0.05$, $\gamma = 0$, *and* $\delta = 2000$, respectively. The convergence rate is faster, and the steady-state MSD is lower to show that the proposed distributed adaptive filter algorithms are more robust to the impulsive noise than DSELMS, DRVSSLMS, and DLLAD.

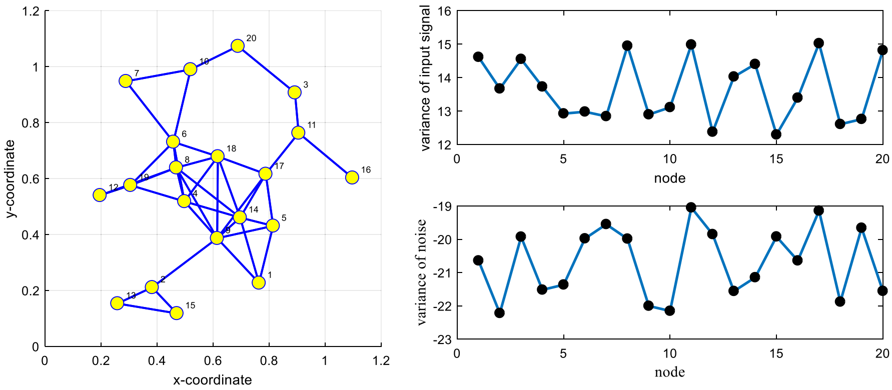

Fig. 18 (Left) Random network topology to be decided by a certain radius; (Right_top) the input signals $\{\mathbf{X}_{\text{n}}(i)\}$ variances at each network node with $\mathbf{R}_{xx,n} = \sigma_{x,n}^2\mathbf{I}_M$ with possibly different diagonal entries chosen randomly, (Right_bottom) the measurement noise variances $\varepsilon_{\text{n}}(t)$ at each network node

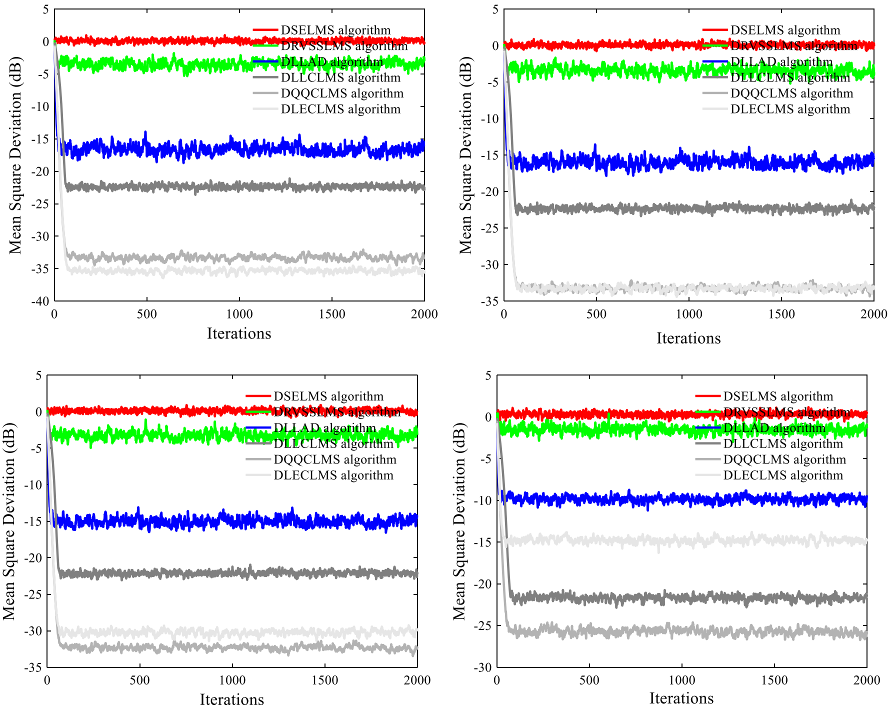

Fig. 19 Transient network MSD (dB) iteration curves of the DSELMS, DRVSSLMS, DLLAD, DLLCLMS, DQQCLMS, and DLECLMS algorithms. (Up_Left) with $\alpha = 1.6$, (Up_right) with $\alpha = 1.1$, and (Down_left) with $\alpha = 0.8$, and (Down_Right) with $\alpha = 0.4$

From Fig. 19, we can find that although different probability density of impulsive noise is considered, the DLLCLMS, DQQCLMS, and DLECLMS algorithms have a slightly faster rate than the DSELMS, DRVSSLMS, and DLLAD algorithms. The DLLCLMS, DQQCLMS, and DLECLMS algorithms still have a minor steady-state error than the DSELMS, DRVSSLMS, and DLLAD algorithms. Simulation experiment 2 shows that the DLLCLMS, DQQCLMS, and DLECLMS algorithms are more robust to impulsive noise than the DSELMS, DRVSSLMS, and DLLAD algorithms. Furthermore, the order of performance superiority is DLECLMS, DQQCLMS, and DLLCLMS. And when noise distribution tends to Gaussian distribution, DLECLMS and DQQCLMS tend to be the same but better than DLLCLMS. This result is also because the cost function designed in this paper is asymmetric. No matter the intensity of the impulse interference noise, its distribution is always asymmetrical; the cost function intended in this paper can better track the change of the estimation error.

5 Conclusion

This paper proposed a family of diffusion adaptive filtering algorithms using three asymmetric costs of error functions; those three distributed adaptive filtering algorithms are robust to the impulsive noise and input signal. Specifically, those three distributed adaptive algorithms are developed by modifying the DLMS algorithm and combining the LLC, QQC, and LEC functions at all distributed network nodes. The theoretical analysis demonstrates that those three distributed adaptive filtering algorithms can effectively estimate from an asymmetric cost function perspective. Besides, theoretical mean behavior interpreted that those three algorithms can achieve accurate estimation. Simulation results showed that the DLLCLMS, DQQCLMS, and DLECLMS algorithms are more robust to the input signal and impulsive noise than the DSELMS, DRVSSLMS, and DLLAD algorithms. The DLLCLMS, DQQCLMS, and DLECLMS algorithms have superior performance when estimating the unknown linear system under the changeable impulsive noise environments and different types of input signals, which will significantly impact real-world applications. Besides, the environment in actual applications is more complex, nonlinear, and time-varying and needs to be adjusted for different application scenarios [28]. We also need to consider whether the system parameters to be evaluated are sparsity (such as brain networks based on fMRI or EEG signals) [12, 14]. In such cases, adding regularized constraint terms to those adaptive algorithms will be better.

Acknowledgements This work was supported by the Central Universities Foundation, Southwest Minzu University (2021XJTD01, 2021PTJS24), the Sichuan Science and Technology Program (23NSFSC2916), the National Key Research and Development Program of China (2022YFE0134600), the introduction of talent, Southwest MinZu University, funding research projects start (RQD2021064), and the Natural Science Foundation of Henan Polytechnic University (B2021-38).

Availability of data and materials There are no data available from the corresponding author.

Declarations

Conflict of interest No conflict of interest exists in submitting this manuscript, and all authors approve the manuscript for publication.

References

1. S. Ashkezari-Toussi, H. Sadoghi-Yazdi, Robust diffusion LMS over adaptive networks. Signal Process. **158**, 201–209 (2019). https://doi.org/10.1016/j.sigpro.2019.01.004
2. F. Barani, A. Savadi, H.S. Yazdi, Convergence behavior of diffusion stochastic gradient descent algorithm. Signal Process. (2021). https://doi.org/10.1016/j.sigpro.2021.108014
3. F.S. Cattivelli, C.G. Lopes, A.H. Sayed, Diffusion recursive least-squares for distributed estimation over adaptive networks. IEEE Trans. Signal Process. **56**(5), 1865–1877 (2008)
4. F.S. Cattivelli, A.H. Sayed, Diffusion LMS strategies for distributed estimation. IEEE Trans. Signal Process. **58**(3), 1035–1048 (2010). https://doi.org/10.1109/tsp.2009.2033729
5. F. Chen, T. Shi, S. Duan, L. Wang, J. Wu, Diffusion least logarithmic absolute difference algorithm for distributed estimation. Signal Process. **142**, 423–430 (2018). https://doi.org/10.1016/j.sigpro.2017.07.014
6. S.F. Crone *Neuronale Netze zur Prognose und Disposition im Handel.* (2010)
7. P.S.R. Diniz, *Adaptive Filtering: Algorithms and Practical Implementation* (Springer, New York, 2013)
8. K. Dress, S. Lessmann, H.-J. von Mettenheim, Residual value forecasting using asymmetric cost functions. Int. J. Forecast **34**(4), 551–565 (2018). https://doi.org/10.1016/j.ijforecast.2018.01.008
9. S.A. Fatemi, A. Kuh, M. Fripp, Online and batch methods for solar radiation forecast under asymmetric cost functions. Renew. Energy **91**, 397–408 (2016). https://doi.org/10.1016/j.renene.2016.01.058
10. J. Fernandez-Bes, V. Elvira and S. Van Vaerenbergh, A probabilistic least-mean-squares filter, In: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2015)

11. Y. Gao, J. Ni, J. Chen, X. Chen, Steady-state and stability analyses of diffusion sign-error LMS algorithm. Signal Process. **149**, 62–67 (2018). https://doi.org/10.1016/j.sigpro.2018.02.033
12. S. Guan, R. Jiang, H. Bian, J. Yuan, P. Xu, C. Meng et al., The profiles of non-stationarity and non-linearity in the time series of resting-state brain networks. Front. Neurosci. (2020). https://doi.org/10.3389/fnins.2020.00493
13. S. Guan, C. Meng, B. Biswal, Diffusion-probabilistic least mean square algorithm. Circuits Syst. Signal Process. (2020). https://doi.org/10.1007/s00034-020-01518-3
14. S. Guan, D. Wan, Y. Yang, B. Biswal, Sources of multifractality of the brain rs-fMRI signal. Chaos, Solitons Fractals (2022). https://doi.org/10.1016/j.chaos.2022.112222
15. S.S. Haykin *Adaptive filter theory.* Upper Saddle River, New Jersey: Pearson (2014)
16. F. Huang, J. Zhang, S. Zhang, Mean-square-deviation analysis of probabilistic LMS algorithm. Digital Signal Process. **92**, 26–35 (2019). https://doi.org/10.1016/j.dsp.2019.05.003
17. W. Huang, L. Li, Q. Li, X. Yao, Diffusion robust variable step-size LMS algorithm over distributed networks. IEEE Access **6**, 47511–47520 (2018). https://doi.org/10.1109/access.2018.2866857
18. B. Isabelle and G. Julien Hyperbolic cosine function in physics. Eur. J. Phys. (2020)
19. L. Li, H. Zhao, S. Lv, Diffusion recursive total least square algorithm over adaptive networks and performance analysis. Signal Process (2021). https://doi.org/10.1016/j.sigpro.2020.107954
20. Z. Li, S. Guan, Diffusion normalized Huber adaptive filtering algorithm. J. Franklin Inst. **355**(8), 3812–3825 (2018). https://doi.org/10.1016/j.jfranklin.2018.03.001
21. W. Liu, L.C. Prncipe and S. Haykin Kernel adaptive filtering: a comprehensive introduction. WILEY (2010)
22. C.G. Lopes, A.H. Sayed, Diffusion least-mean squares over adaptive networks: formulation and performance analysis. IEEE Trans. Signal Process. **56**(7), 3122–3136 (2008)
23. J. Mostafaee, S. Mobayen, B. Vaseghi, M. Vahedi, A. Fekih, Complex dynamical behaviors of a novel exponential hyper–chaotic system and its application in fast synchronization and color image encryption. Sci. Progress (2021). https://doi.org/10.1177/00368504211003388
24. A. Naeimi Sadigh, H. Sadoghi Yazdi, A. Harati, Diversity-based diffusion robust RLS using adaptive forgetting factor. Signal Process (2021). https://doi.org/10.1016/j.sigpro.2020.107950
25. R. Nassif, S. Vlaski, C. Richard, A.H. Sayed, Learning over multitask graphs–part I: stability analysis. IEEE Open J. Signal Process. **1**, 28–45 (2020). https://doi.org/10.1109/ojsp.2020.2989038
26. J. Ni, Diffusion sign subband adaptive filtering algorithm for distributed estimation. IEEE Signal Process. Lett. **22**(11), 2029–2033 (2015). https://doi.org/10.1109/lsp.2015.2454055
27. J. Ni, J. Chen, X. Chen, Diffusion sign-error LMS algorithm: formulation and stochastic behavior analysis. Signal Process. **128**, 142–149 (2016). https://doi.org/10.1016/j.sigpro.2016.03.022
28. D.C.J.C. Príncipe (2018). Adaptive learning methods for nonlinear system modeling.
29. B.C.Z.L.Y.L.P. Ren Asymmetric Correntropy for Robust Adaptive Filtering. arXiv:1911.11855 (2019)
30. A.H. Sayed, *Fundamentals of adaptive filtering* (IEEE Press Wiley-Interscience, New York, 2003)
31. A.H. Sayed, Adaptive networks. Proc. IEEE **102**(4), 460–497 (2014). https://doi.org/10.1109/jproc.2014.2306253
32. M. Shao, C.L. Nikias, Signal processing with fractional lower order moments: stable processes and their applications. Proc. IEEE **81**(7), 986–1010 (1993). https://doi.org/10.1109/5.231338
33. B. Vaseghi, S.S. Hashemi, S. Mobayen, A. Fekih, Finite time chaos synchronization in time-delay channel and its application to satellite image encryption in OFDM communication systems. IEEE Access **9**, 21332–21344 (2021). https://doi.org/10.1109/access.2021.3055580
34. B. Vaseghi, S. Mobayen, S.S. Hashemi, A. Fekih, Fast reaching finite time synchronization approach for chaotic systems with application in medical image encryption. IEEE Access **9**, 25911–25925 (2021). https://doi.org/10.1109/access.2021.3056037
35. E. Walach, B. Widrow, The least mean fourth (LMF) adaptive algorithm and its family. IEEE Trans. Inf. Theory **30**(2), 275–283 (1984). https://doi.org/10.1109/tit.1984.1056886
36. S. Wang, W. Wang, K. Xiong, H.H.C. Iu, C.K. Tse, Logarithmic hyperbolic cosine adaptive filter and its performance analysis. IEEE Transact. Systems, Man, Cybernet. Syst (2019). https://doi.org/10.1109/tsmc.2019.2915663
37. F. Wen, Diffusion least-mean P-power algorithms for distributed estimation in alpha-stable noise environments. Electron. Lett. **49**(21), 1355–1356 (2013). https://doi.org/10.1049/el.2013.2331
38. B. Widrow, Thinking about thinking: the discovery of the LMS algorithm. IEEE Signal Process. Mag. **22**(1), 100–106 (2005). https://doi.org/10.1109/msp.2005.1407720

39. M. Zhang, D. Jin, J. Chen, J. Ni, Zeroth-order diffusion adaptive filter over networks. IEEE Trans. Signal Process. **69**, 589–602 (2021). https://doi.org/10.1109/tsp.2020.3048237