

# Diffusion Adaptive Filtering Algorithm Based on the Fair Cost Function

Sihai Guan (✉ [gcihey@sina.cn](mailto:gcihey@sina.cn))

Southwest Minzu University

Qing Cheng

Sichuan Vocational College of Finance and Economics

Yong Zhao

Henan Polytechnic University

Bharat Biswal

The Clinical Hospital of Chengdu Brain Science Institute, University of Electronic Science and Technology of China

---

## Research Article

**Keywords:** Distributed network, impulsive interference, input signals, the Fair cost function

**Posted Date:** June 24th, 2021

**DOI:** <https://doi.org/10.21203/rs.3.rs-569037/v1>

**License:**  This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

---

# Diffusion adaptive filtering algorithm based on the Fair cost function

Sihai Guan<sup>1,2\*</sup>, Qing Cheng<sup>3</sup>, Yong Zhao<sup>4</sup>, Bharat Biswal<sup>5,6</sup>

(1. College of Electronic and Information, Southwest Minzu University. 2. Key Laboratory of Electronic and Information Engineering, State Ethnic Affairs Commission. 3. Sichuan Vocational College of Finance and Economics, Chengdu, China. 4. School of Mechanical and Power Engineering, Henan Polytechnic University, Jiaozuo, 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.)

\*Correspondence to Sihai Guan (Email: gcihey@sina.cn)

**Abstract:** To better perform distributed estimation, this paper, by combining the Fair cost function and adapt-then-combine scheme at all distributed network nodes, a novel diffusion adaptive estimation algorithm is proposed from an M-estimator perspective, which is called the diffusion Fair (DFair) adaptive filtering algorithm. The stability of the mean estimation error and the computational complexity of the DFair are theoretically analyzed. Compared with the RDLMS, DNLMM, DGCLD, and DPLMS algorithms, the simulation experiment results show that the DFair algorithm is more robust to input signals and impulsive interference. In conclusion, Theoretical analysis and simulation results show that the DFair algorithm performs better when estimating an unknown linear system in the changeable impulsive interference environments.

**Keywords:** Distributed network; impulsive interference; input signals; the Fair cost function.

## 1 Introduction

Adaptive filter algorithms are often used in equalization, active noise control, echo cancellation, biomedical engineering, and many other fields [1-7]. Distributed adaptive signal processing is an extension of adaptive filters over graphs[8]. There are three widely-used distributed estimation cooperation strategies: incremental, consensus, and diffusion [3]. Besides, the asymmetry problem may lead to the unstable growth of consensus technology, and the asymmetry problem can be passed through the network. However, among them, the diffusion strategy can eliminate the asymmetry problem. So, the diffusion strategies were used frequently, and they include the adapt-then-combine (ATC) scheme [8] and the combine-then-adapt (CTA) scheme [3, 9]. Cattivelli

and colleagues analyzed these two structures' performance for these two kinds of structures, showing that the ATC scheme outperforms the CTA scheme[8]. Then, the ATC scheme becomes a research focus in distributed adaptive filtering algorithms [8, 10-17]. The Wiener filtering principle is the fundamental of the adaptive filtering algorithm [18], based on the parameter mean square error (MSE) to construct an efficient cost function. Cost functions are used to learn parameters that explain the estimation well, and they define how costly our adaptive algorithm estimation mistakes are [19]. The cost function should penalize "large" estimation error and "very-large" estimation error almost equally. Also, if we want better estimation properties, then we have to give good computational complexity properties.

Roughly speaking, if a line connecting two points never intersects the cost function, the cost function is convex, and the convex cost function has only one global minimum value. Strictly convex cost functions have a unique global minimum value, and the sum of convex cost functions is also convex properties. Therefore, MSE has only one global minimum value. Furthermore, outliers are examples of data that are far from most other examples. Unfortunately, they happen more in reality than we hoped. Up to now, many types the cost functions have been used to design adaptive filtering algorithms, such as least mean absolute third [20], least mean fourth [21], entropy[22], least-squares estimator [23], absolute value estimator [24], the Cauchy [25], Geman-McClure [26], Welsch [27, 28], and Huber function [29-32]. Specifically, the least-squares estimator is not robust because their influence function is not bounded [23], and the absolute value estimator is not stable because the function of estimate error ( $|e(i)|$ ) at  $i$ -th is not strictly convex [24]. Indeed, the second derivative at  $e(i) = 0$  is unbounded, and an in determinant solution may result. The absolute value estimator reduces the influence of large errors, but they still have an influence because the influence function. Combine least-squares and absolute value estimator take both the absolute value estimators' advantage to reduce the influence of large errors and that of least-squares estimators to be convex [33]. They behave like least-squares for small  $e(i)$  and like absolute value estimators for large  $e(i)$ , hence the name of this type of estimators. The least p-powers function represents a family of cost functions [34]. It is the least-square method when  $p = 2$  and the absolute value estimator when  $p = 1$ . The smaller  $p$ , the smaller is the incidence of large estimation errors in the estimate  $p$ . It seems that  $p$  must be fairly moderate to provide a relatively reliable estimate, or in other words, an estimate that is hardly disturbed by outlying data. The selection of an optimal  $p$  has been investigated, and for  $p$  around 1.2, good estimated performance may be expected [35]. However, many difficulties are encountered in the computation when parameter  $p$  is in the range of interest  $1 < p < 2$ , because zero residuals are troublesome. The other remaining functions have the same problem as the Cauchy function [25]. As can be seen from the influence function, the influence of large estimation errors only decreases linearly with their size. The Geman-McClure [26] and

Welsch [27, 28] functions try to reduce the effect of large estimation errors further. It seems complicated to select a cost function for general use without being somewhat arbitrary. The Huber cost function was often used [26-29], but the Huber cost function is convex, differentiable, and robust to outliers. However,  $\delta$  setting is not an easy task. Huber's function is a parabola in the vicinity of zero, and increases linearly at a given level  $|e(i)| > \delta$ . This the estimator is very satisfactory and is rarely found that it is inferior to some other cost functions, so it is recommended in almost all situations. However, sometimes difficulties are encountered, which may be due to the discontinuous gradient value of the cost function due to its discontinuous second derivative. Tukey's loss is non-convex, non-differentiable [36]; it seems complicated to select a cost function for general use without being somewhat arbitrary. Following [37], for the regression problems, the best choice is the  $L_p$  despite its theoretical no robustness: they are quasi-robust. However, it suffers from its computational difficulties. But the Fair function can yield nicely converging computational procedures. As compared to the quadratic cost function, the adaptive Fair function can track the statistical characteristics of the error  $e(i)$ , which is conducive to improving the robustness of the proposed algorithm. In contrast, the Fair function is preferable since it is continuously derivable, while the Huber function is a piecewise function with unsmooth points.

Besides, system measurement interference will affect the error cost function, where impulse interference will particularly significantly affect the estimation accuracy and most diffusion estimation algorithms on the network. Therefore, it is necessary to design a robust distributed adaptive algorithm to deal with impulse interference of different intensities. Recently, Wen proposed the diffusion least mean p-power algorithm [38], robust to the generalized Gaussian noise environment. Still, DLMP was proposed with a fixed power p-value, so p is the critical factor, which means the DLMP algorithm performance is highly susceptible to the p-value. Using minimization of L1-norm subject to a constraint on the estimate weight vectors, Ni and colleagues designed a diffusion sign subband adaptive filtering (DSSAF) algorithm [39]. Still, the computational complexity of the DSSAF algorithm is relatively large. By combining the diffusion least mean square (DLMS) algorithm [11] and the sign operation to the estimated error at each iteration moment point, Ni and colleagues derived a diffusion sign-error LMS (DSELMS) algorithm [40]. The DSELMS algorithm architecture is simple and easy to implement, but the DSELMS algorithm has a significant drawback (i.e., the steady-state error is high) [41]. Besides, inspired by the least logarithmic absolute difference (LLAD) operation, Chen and colleagues designed the DLLAD algorithm [9]. But the robustness of this algorithm to the input signal and impulsive interference has not been performed. By combining the ATC strategy and the probabilistic LMS algorithm [42, 43], Guan and colleagues proposed a diffusion probabilistic least mean square (DPLMS) algorithm [44]. Based on the Huber objective

function, a similar set of algorithms by Guan and colleagues [27], Wei and colleagues [42] have been proposed as the DNHuber DRVSSLMS algorithms, respectively. Besides, the computational algorithm complexity of the DRVSSLMS algorithm is high, which is not conducive to implementing practical engineering projects. Soheila and colleagues [31] used the pseudo-Huber cost function instead of the square error to design the RDLMS algorithm based on the Huber cost function; however, the RDLMS algorithm was not designed for impulsive interference. By applying the modified Huber function together with an adaptive threshold, YU and colleagues proposed a novel diffusion normalized least mean M-estimate algorithm (DNLMM) against impulsive interference [32] for impulsive interference. But, the basis of the DNLMM algorithm is still based on the Huber function. Huber's function is a parabola in the vicinity of zero and increases linearly at a given level  $|e(i)| > \delta$ . And from time to time, difficulties are encountered, which may be due to the lack of stability in the gradient values of the cost function because of its discontinuous second derivative. Furthermore, the adaptive Fair function can track the statistical characteristics of the estimated error  $e(i)$ , which helps to improve the robustness of the proposed algorithm. In contrast, the Fair cost function is preferable since it is continuously derivable while the Huber cost function is a piecewise function with unsmooth points. So, we propose a robust distributed adaptive filtering algorithm by combining the Fair cost function and ATC scheme at all distributed network nodes in this paper, namely the DFair algorithm. The stability of the mean estimation error and the computational complexity are analyzed theoretically. Simulation experiment results indicate that the DFair algorithm is more robust to the input signal and impulsive interference than the RDLMS[31], DNLMM[32], DGCLD[45], and DPLMS[44] algorithms.

The remainder of this paper is organized as follows. The proposed DFair algorithm is proposed in detail in Section 2. The DFair algorithm's statistical behavior, including the stability performance, computation complexity, and parameter ( $\delta$ ) for the DFair algorithm, is studied in Section 3. The simulation experiments are described in Section 4. Finally, the conclusion is provided in Section 5.

## 2 Proposed the DFair algorithm

In this section, the DFair algorithm is developed. Firstly, an adaptive filtering algorithm based on the Fair cost function is proposed. Then we modify this adaptive filtering algorithm by extending at all network agents to develop the DFair algorithm.

### 2.1 The adaptive filter algorithm based on the Fair cost function

Let  $\mathbf{W}(i)$  be the system estimated weight vector with length  $M$ ,  $\mathbf{X}(i)$  the input signal vector of the adaptive filter at iteration  $i$ , and the prediction error  $e(i)$  between the desired signal  $d(i)$  and the actual output  $y(i)$  can be expressed by

$$e(i) = d(i) - y(i) = \mathbf{W}^o(i)\mathbf{X}(i) + v(i) - \mathbf{W}^T(i)\mathbf{X}(i) \quad (1)$$

where  $v(i)$  is the measurement interference.

Fair adaptive filter aims to minimize the Fair cost function defined as

$$J(i) = \delta^2 \left( \frac{|e(i)|}{\delta} - \log \left( 1 + \frac{|e(i)|}{\delta} \right) \right) \quad (2)$$

where  $\delta > 0$  is the cut-off value.

According to the steepest descent method, the system estimated weight vector update of the Fair adaptive filter is

$$\mathbf{W}(i+1) = \mathbf{W}(i) + \mu \delta \left( \frac{\frac{|e(i)|}{\delta}}{1 + \frac{|e(i)|}{\delta}} \right) \text{sgn}(e(i)) \mathbf{X}(i) = \mathbf{W}(i) + \mu \frac{\delta e(i)}{\delta + |e(i)|} \mathbf{X}(i) \quad (3)$$

where  $\mu$  is the step size.

## 2.2 The adaptive diffusion filter based on the Fair cost function

According to our previous paper's research, consider a network of  $N$  sensor nodes distributed over a geographic area (as Fig.1) [20, 30, 44].  $\mathbf{X}_n(i)$  and  $d_n(i)$  are the input signals and observation output signals at agent  $n$ , respectively.

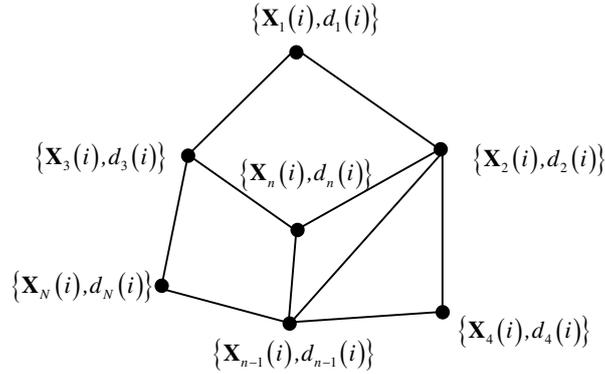


Fig.1. A network consisting of  $N$  agents [20, 30, 44]

Based on Fig.1, we seek the optimal linear estimator that minimizes the global cost function:

$$J^{global}(\mathbf{W}(i)) = \sum_n J_n^{local}(\mathbf{W}(i)) \quad (4)$$

where at each time instant  $i$ , each sensor node  $n \in \{1, 2, \dots, N\}$  has access to some zero-mean random process  $\{d_n(i), \mathbf{X}_n(i)\}$ ,  $d_n(i)$  is a scalar and  $\mathbf{X}_n(i)$  is a regression vector. Suppose these measurements output follows a standard model given by:

$$d_n(i) = \mathbf{X}_n(i) \mathbf{W}^0 + v_n(i) \quad (5)$$

where  $\mathbf{W}^0$  is the unknown system parameter vector with length  $M$ , and  $v_n(i)$  is the measurement interference with variance  $\sigma_{v,n}^2$ .

In [11], the DLMS algorithm is obtained by minimizing a linear combination of the local MSE:

$$J_n^{local}(\mathbf{W}(i)) = \sum_{l \in N_n} a_{l,n} E|e_l(i)|$$

$$= \sum_{l \in N_n} a_{l,n} \mathbb{E} |d_l(i) - \mathbf{X}_l(i) \mathbf{W}^T(i)| \quad (6)$$

where the set of nodes that are connected to  $n$  (including  $n$  itself) is denoted by  $N_n$  and is called the neighborhood of nodes  $n$ . The weighting coefficients  $a_{l,n}$  are real and satisfy  $\sum_{l \in N_n} a_{l,n} = 1$ .  $a_{l,n}$  forms a nonnegative combination matrix  $\mathbf{A}$ .

The DLMS algorithm obtains the estimation via two steps, adaptation and combination. According to the order of these two steps, the updating equation of the DLMS algorithm can be expressed as

$$\begin{cases} \boldsymbol{\varphi}_n(i) = \mathbf{W}_n(i-1) + \mu_n \mathbf{X}_n(i) e_n(i) \\ \mathbf{W}_n(i) = \sum_{l \in N_n} a_{l,n} \boldsymbol{\varphi}_l(i) \end{cases} \quad (7)$$

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

Inspired by Eq. (3) and Eq. (7), a summary of the DFair algorithm procedure based on the analysis presented above is given in Table 1.

Table 1 the DFair algorithm summary

---

Initialize:  $\{w_{n,-1} = 0\}$  for all  $n$ ,  $\delta$ .

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

$$\boldsymbol{\varphi}_n(i) = \mathbf{W}_n(i-1) + \mu_n \frac{\delta e_n(i)}{\delta + |e_n(i)|} \mathbf{X}_n(i) \quad (8)$$

$$\mathbf{W}_n(i) = \sum_{l \in N_n} a_{l,n} \boldsymbol{\varphi}_l(i) \quad (9)$$


---

### 3. Performance analysis

Performances of the DFair adaptive filtering algorithms include mean behavior and computational complexity, will be discussed in this subsection. Firstly, to facilitate analysis and expression, we define some equations at agent  $n$  and time  $i$ ,  $\widehat{\mathbf{W}}_n(i) = \mathbf{W}^o - \mathbf{W}_n(i)$ ,  $\widehat{\boldsymbol{\varphi}}_n(i) = \mathbf{W}^o - \boldsymbol{\varphi}_n(i)$ , which are then collected to form the system weight error vector and intermediate system weight error vector, i.e.,  $\mathbf{W}(i) = \text{col}\{\mathbf{W}_1(i), \mathbf{W}_2(i), \dots, \mathbf{W}_N(i)\}$ ,  $\boldsymbol{\varphi}(i) = \text{col}\{\boldsymbol{\varphi}_1(i), \boldsymbol{\varphi}_2(i), \dots, \boldsymbol{\varphi}_N(i)\}$ ,  $\widehat{\mathbf{W}}(i) = \text{col}\{\widehat{\mathbf{W}}_1(i), \widehat{\mathbf{W}}_2(i), \dots, \widehat{\mathbf{W}}_N(i)\}$ ,  $\widehat{\boldsymbol{\varphi}}(i) = \text{col}\{\widehat{\boldsymbol{\varphi}}_1(i), \widehat{\boldsymbol{\varphi}}_2(i), \dots, \widehat{\boldsymbol{\varphi}}_N(i)\}$ ,  $\boldsymbol{\mu}(i) = \text{diag}\left\{\mu_1 \frac{\delta}{\delta + |e_1(i)|}, \mu_2 \frac{\delta}{\delta + |e_2(i)|}, \dots, \mu_N \frac{\delta}{\delta + |e_N(i)|}\right\}$ , and  $\mathbf{e}(i) = \text{col}\{e_1(i), e_2(i), \dots, e_N(i)\}$ .

#### 3.1 Mean weight error vector behavior

To facilitate performance analysis, we make the following assumptions:

*Assumption 1:* All measurement interferences are independent of any other signals.

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

*Assumption 3:* The regression vector  $\mathbf{X}_n(i)$  is independent of  $\widehat{\mathbf{W}}_k(j)$  for all  $k$  and  $j < i$ .

The DFair algorithm will be obtained. Eq. (8) and Eq. (9) can be written as

$$\widehat{\boldsymbol{\varphi}}(i) = \widehat{\mathbf{W}}(i-1) - \mathbf{S}_\mu \mathbf{S}_e(i) \mathbf{X}(i) \quad (10)$$

$$\widehat{\mathbf{W}}(i) = \mathbf{C}^T \widehat{\boldsymbol{\phi}}(i) \quad (11)$$

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

Taking the expectation of Eq. (10) and Eq. (11),

$$\mathbf{E}[\widehat{\mathbf{W}}(i)] = \mathbf{C}^T \mathbf{E}[\widehat{\mathbf{W}}(i-1)] - \mathbf{C}^T \mathbf{S}_\mu \mathbf{E}[\mathbf{S}_{\mathbf{S}_e}(i) \mathbf{X}(i)] \quad (12)$$

Denote the measurement interference vector by  $\mathbf{V}(i) = \text{col}\{v_1(i), v_2(i), \dots, v_N(i)\}$ ,

$\mathbf{g}(i) = \text{col}\{g_1(i), g_2(i), \dots, g_N(i)\}$ ,  $\mathbf{Im}(i) = \text{col}\{Im_1(i), Im_2(i), \dots, Im_N(i)\}$ ,

$\mathbf{S}_g(i) = \text{diag}\{\mathbf{g}(i)\}$ ,  $\mathbf{S}_{Im}(i) = \text{diag}\{\mathbf{Im}(i)\}$ ,  $\mathbf{S}_X(i) = \text{diag}\{\mathbf{X}_1(i), \mathbf{X}_2(i), \dots, \mathbf{X}_N(i)\}$ .

So, from Eq. (1) we have  $\mathbf{e}(i) = \mathbf{S}_X^T(i) \widehat{\mathbf{W}}(i-1) + \mathbf{V}(i) = \mathbf{e}_o(i) + \mathbf{V}(i)$ . Then, let

$$\begin{cases} \mathbf{e}_g(i) = \mathbf{e}_o(i) + \mathbf{g}(i) & \begin{cases} \mathbf{S}_{S_g}(i) = \mathbf{S}_g(i) \otimes \mathbf{I} \\ \mathbf{S}_{S_{Im}}(i) = \mathbf{S}_{Im}(i) \otimes \mathbf{I} \end{cases} \end{cases}$$

So,

$$\mathbf{E}[\widehat{\mathbf{W}}(i)] = \mathbf{C}^T [\mathbf{I}_{NM} - \mathbf{S}_\mu \text{diag}\{\mathbf{R}_{xx,1}, \mathbf{R}_{xx,2}, \dots, \mathbf{R}_{xx,N}\}] \mathbf{E}[\widehat{\mathbf{W}}(i-1)] \quad (13)$$

From Eq. (13), one can see that the asymptotic unbiasedness of the DFair algorithm can be guaranteed if the matrix  $\mathbf{C}^T [\mathbf{I}_{NM} - \mathbf{S}_\mu \text{diag}\{\mathbf{R}_{xx,1}, \mathbf{R}_{xx,2}, \dots, \mathbf{R}_{xx,N}\}]$  is stable. The matrix  $[\mathbf{I}_{NM} - \mathbf{S}_\mu \text{diag}\{\mathbf{R}_{xx,1}, \mathbf{R}_{xx,2}, \dots, \mathbf{R}_{xx,N}\}]$  is a block-diagonal matrix and it can be easily verified that it is stable if its block-diagonal entries  $[\mathbf{I} - \mu_n \mathbf{R}_{xx,n}]$  is stable. So, the condition for stability of the mean weight error vector (as Eq. (14)) is given by

$$0 < \mu_n < \frac{2}{\rho_{\max}(\mathbf{R}_{xx,n})} \quad (14)$$

where  $\rho_{\max}$  denotes the maximal eigenvalue of  $\mathbf{R}_{xx,n}$ . So, based on Eq. (12) and Eq. (14), we obtain  $\mathbf{E}[\widehat{\mathbf{W}}(\infty)] = \mathbf{0}$ .

### 3.2. Parameters $\delta$ for the proposed algorithm

The choice of  $\delta$  in Eq. (8) plays a vital role in the performance of the DFair algorithm. When  $|e_n(i)|/\delta \rightarrow 0$ ,  $\hat{\mu}_n(i) = 0$  for each time  $i \geq 0$  and each agent  $n$ . When  $|e_n(i)|/\delta \rightarrow \infty$ ,  $\hat{\mu}_n(i) = \mu_n$  for each time  $i \geq 0$  and each agent  $n$ . So, when large  $\hat{\mu}_n(i) = \mu_n$  lead to a large MSD and even cause loss of convergence, while a small  $\hat{\mu}_n(i) = 0$  degrade the tracking speed of the DFair algorithm, which means that a large step-size responds quickly to plant changes during the initial convergence, and then a tiny step-size is used as the algorithm approaches its steady state. In other words, the DFair algorithm has been presented to obtain a fast convergence rate and a small steady-state error. Therefore, it is necessary to discuss the value of  $\delta$  under the different intensities of impulsive interference. We set six groups of the experiment in a system

identification application to choose the optimum cut-off value  $\delta$  under different input signals, impulsive interference, and different network structures. Another method that can also get the optimum cut-off value  $\delta$  based on the theory derivation method for different input signals, various impulsive interferences, and various network structures. For the theory derivation method, although the optimal parameters  $\delta$  of the proposed two algorithms are obtained based on minimizing the mean-square deviation (MSD) at the current time, iterative formulas will increase the computational complexity. In this paper, the experimental simulation method will get the parameters  $\delta$  of the proposed diffusion adaptive filtering algorithm. Several experiments were performed in a system identification application in the presence of impulsive interference and Gaussian noise. Gaussian noise is a white Gaussian random process with zero mean and variance equal to 0.01. Impulsive interference is a Bernoulli-Gaussian distribution [20] that was added to the unknown system output also. The Bernoulli-Gaussian impulsive interference,  $v(i) = f(i)g(i)$  is a product of a Bernoulli process  $g(i)$  and a Gaussian process  $f(i)$ , where  $f(i)$  is a white Gaussian random process with zero mean and variance  $\sigma_f^2$ , and  $g(i) = \{0, 1\}$  is a Bernoulli process with the probabilities  $p(1) = \text{Pr}$  and  $p(0) = 1 - \text{Pr}$ .

In this part, nodes of network topology are set as  $N=20$ , and the input regresses  $\mathbf{X}_n(i)$  of this distributed network are assumed to be spatiotemporally independent zero-mean white Gaussian distributed with different covariance matrixes  $\mathbf{R}_{xx,n}$ . The impulsive interference is also assumed to be spatially and temporally independent distributed with power  $\sigma_f^2$ . For the adaptation and combination weights, we apply the uniform rule (i.e.,  $a_{l,n} = 1/N_n$ , where the set of nodes that are connected to  $n$  is denoted by  $N_n$ ). We evaluate the relative efficiency of different  $\delta$  estimators based on their MSD to evaluate the performance of DFair, where  $\text{MSD}(i) = \frac{1}{N} \sum_{n=1}^N \text{E}[|\mathbf{W}_o - \mathbf{W}_n(i)|^2]$  [30, 44]. Also, the independent Monte Carlo number is 10, and each run has 800 iteration numbers. The different probability density of impulsive interference is considered 0%, 20%, 40%, 60%. From Fig. 2, Fig. 3, and Fig. 4, considering the convergence rate and the steady-state estimated error, we know the DFair algorithm is robust for the different probability density of interference when  $\delta = 0.1$ .

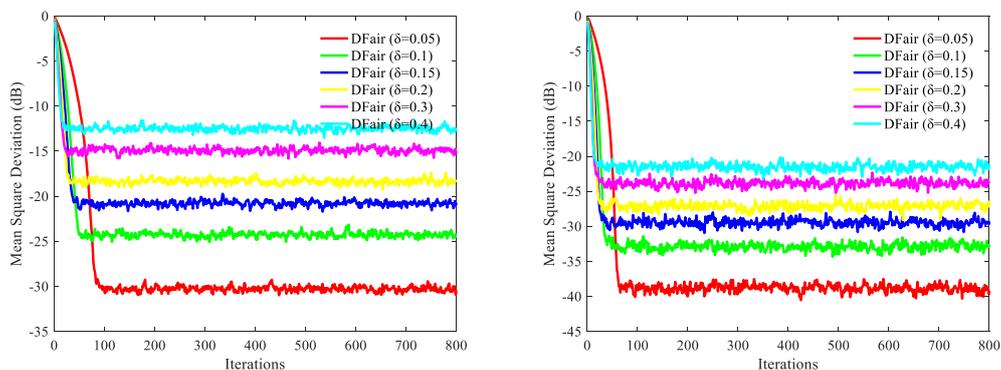


Fig.2 MSD curve with different  $\delta$  of the DFair algorithm ( $\mu=0.4$ ) when network topology and

neighbors to be decided by probability (probability=0.2): (Left)  $\mathbf{R}_{xx,n} = \sigma_{x,n}^2 \mathbf{I}_M$ ,  $\text{Pr}=0.6$ , and  $\sigma_f^2 = 0.5$ . (Right)  $\mathbf{R}_{xx,n} = \sigma_{x,n}^2(i) \mathbf{I}_M$ ,  $i = 1,2,3, \dots, M$ ,  $\text{Pr}=0.2$ , and  $\sigma_f^2 = 0.5$ .

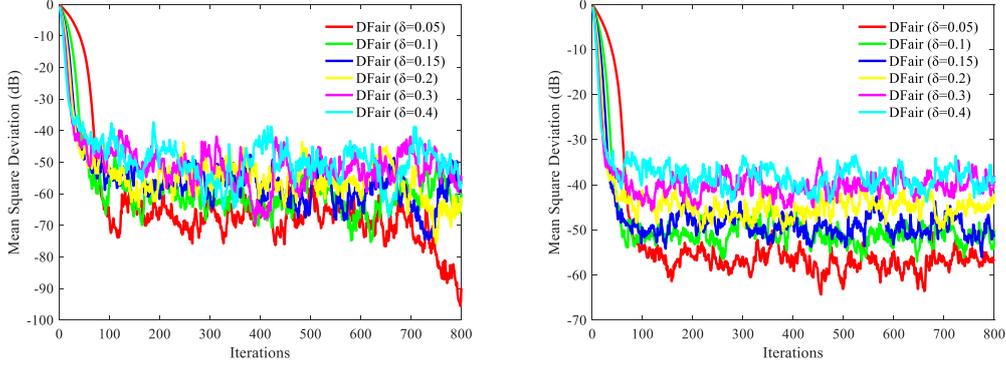


Fig.3 MSD curve with different  $\delta$  of the DFair algorithm ( $\mu=0.4$ ) when network topology and neighbors to be decided by closeness in the distance (radius=0.3): (Left)  $\mathbf{R}_{xx,n}$  is a diagonal matrix with possibly different diagonal entries chosen randomly,  $\text{Pr}=0.4$ , and  $\sigma_f^2 = 0.5$ . (Right)  $\mathbf{R}_{xx,n} = \sigma_{x,n}^2(i) \mathbf{I}_M$ ,  $i = 1,2,3, \dots, M$ ,  $\text{Pr}=0$ , and  $\sigma_f^2 = 0.5$ .

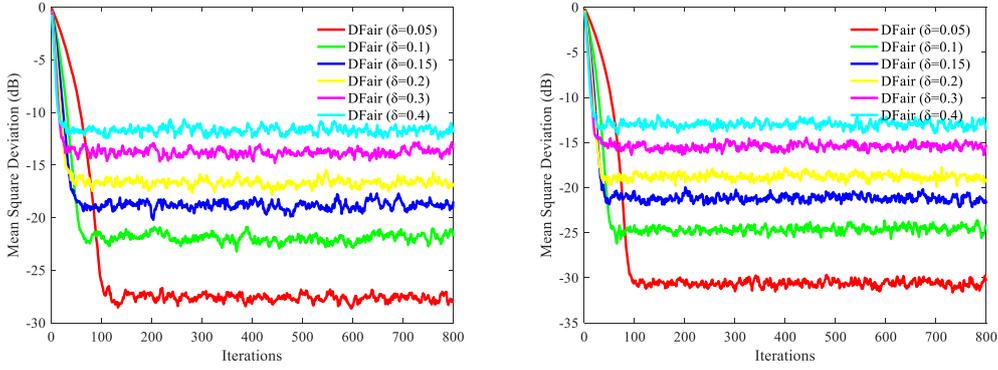


Fig.4 MSD curve with different  $\delta$  of the DFair algorithm ( $\mu=0.4$ ) when network topology and neighbors to be decided by closeness in the distance (radius=0.3): (Left)  $\mathbf{R}_{xx,n}$  is a diagonal matrix with possibly different diagonal entries chosen randomly,  $\text{Pr}=0.4$ , and  $\sigma_f^2 = 0.2$ . (Right)  $\mathbf{R}_{xx,n} = \sigma_{x,n}^2(i) \mathbf{I}_M$ ,  $i = 1,2,3, \dots, M$ ,  $\text{Pr}=0.4$ , and  $\sigma_f^2 = 0.4$ .

### 3.3 Computational complexity

The adaptive filtering algorithm's computational complexity is called the number of arithmetic operations per iteration of the weight vector or coefficient vector. That is the number of multiplications, additions, and et.al. The multiplication operation's time-consuming operation is far greater than the addition operation's time-consuming operation, so the multiplication operation occupies a large proportion of the adaptive filtering algorithm's computational complexity. Therefore, computational complexity is an important property that affects the performance of the adaptive filtering algorithm. According to the above description, the DFair algorithm can be regarded as the structure of the variable step size DLMS algorithm with the following characteristics:

$\hat{\mu}_n(i) = \mu_n \frac{\delta e_n(i)}{\delta + |e_n(i)|}$  (from Table 1). The DFair algorithm has only two more

multiplication operations than the DLMS algorithm. But for DPLMS, there are need more multiplication operations when  $\mu_n(i) = \mu_n \alpha_n(i)$  in Eq.(16) [44] will be computed. Also, for the RDLMS algorithm, there have more multiplication operations when  $\hat{\mu}_n(i) = \mu_n \frac{b_n e_n(i)}{\sqrt{1+(e_n(i)/\delta)^2}}$  in Eq.(17) [31] will be run. In addition, for the DNLMM algorithm,  $\hat{\mu}_n(i) = \mu_n \frac{e_n(i)}{\sqrt{1+\|\mathbf{x}_n(i)\|_2^2}}$  in Eq.(12a) and Eq.(14) [32] will be consider. Furthermore,  $\hat{\mu}_n(i) = \mu_n \left(1 + \frac{e_n(i)}{\delta(1+\delta[\exp(-\lambda[e_k(i-1)^\alpha])])}\right)$  in Table 1 and Eq.(14) [45] will add more multiplication operations. So, the computational complexity of the DFair algorithm smaller than RDLMS[31], DNLMM [32], DGCLD[45], and DPLMS[44] algorithms.

#### 4. Simulation results

In this paper, we focus on the distributed adaptive filtering algorithm and compare the DFair algorithm with the RDLMS[31], DNLMM[32], DGCLD[45], and DPLMS[44] algorithms in linear system identification under different types of input signal and impulsive interference. In this part, to demonstrate the robustness performance of the proposed DFair algorithm in the presence of different intensity levels of impulsive interference and input signal, we set several group simulation experiments with different impulsive interference and different input signal types. For this unknown linear system, we set  $M = 8$ , and the parameters vector was selected randomly. Each distributed network topology consists of  $N = 20$  nodes. An impulsive interference with a Bernoulli-Gaussian distribution [20] was added to the system output (as described in subsection 3.3). Besides, we set the impulsive interference as spatiotemporally independent. For the adaptation weights in the adaptation step and combination weights in the combination step, we apply the uniform rule i.e.  $a_{l,n} = 1/N_n$ . We use the network MSD to evaluate the performance of diffusion algorithms [30, 44].

##### *Simulation experiment 1*

To illustrative, our proposed DFair algorithm is more robust to the input signal than the RDLMS, DNLMM, DGCLD, and DPLMS algorithms. In this experiment, there have the same network topology and the same impulsive interference. If any two network topology nodes are declared neighbors, connect probability greater than or equal to 0.2, the network topology is shown in Fig. 5. The MSD iteration curves for RDLMS ( $\mu$  equal to 0.4), DNLMM ( $\mu$  equal to 0.4), DGCLD ( $\mu$  equal to 0.4), DPLMS, and DFair ( $\mu$  equal to 0.4) algorithms in Fig.6, Fig.7, and Fig.8 are different types of the input signal when the measurement interference in an unknown linear system is impulsive interference with  $\text{Pr}=0.4$ ,  $\sigma_f^2 = 0.09$  and the cut-value  $\delta = 0.1$  for the DFair algorithm. Besides, the independent Monte Carlo number is 10, and each run has 800

iteration numbers.

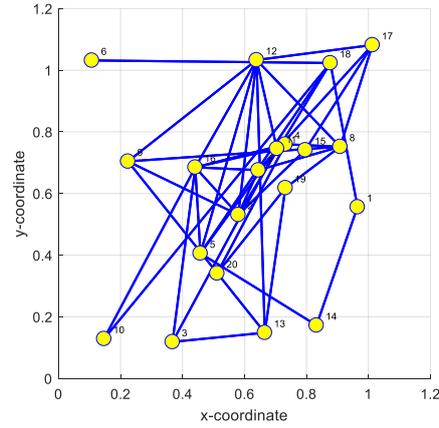


Fig.5. Random network topology to be decided by probability.

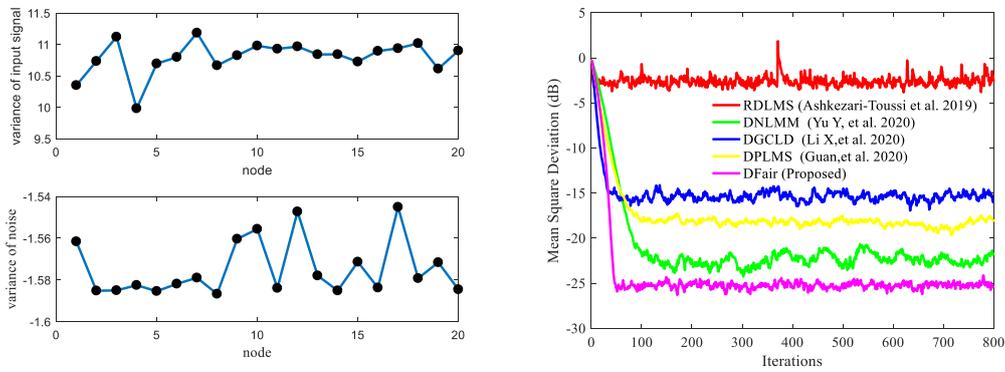


Fig.6. (Left\_top) the input signals  $\{\mathbf{X}_n(i)\}$  variances of at each network node with  $\mathbf{R}_{xx,n}$  is a diagonal matrix with possibly different diagonal entries chosen randomly, (Left\_bottom) the measurement interference variances  $\varepsilon_n(i)$  at each network node; (Right\_bottom) Transient network MSD (dB) iteration curves of the RDLMS, DNLMM, DGCLD, DPLMS, and DFair algorithms.

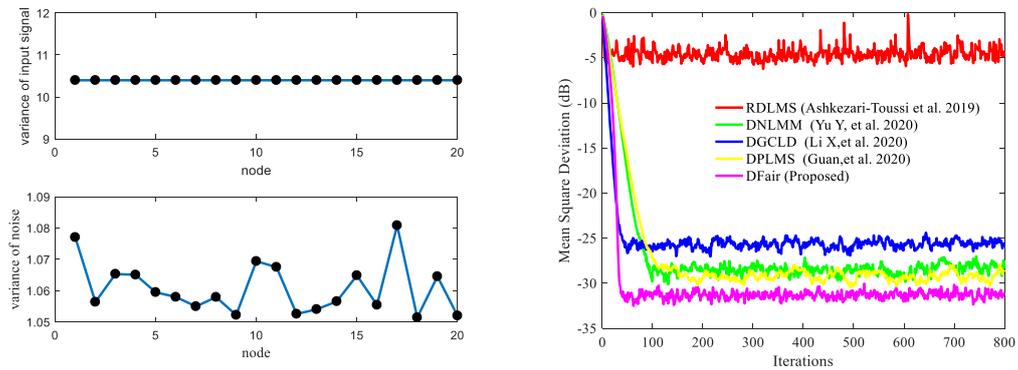


Fig.7. (Left\_top) the input signals  $\{\mathbf{X}_n(i)\}$  variances of at each network node with  $\mathbf{R}_{xx,n} = \sigma_{x,n}^2 \mathbf{I}_M$ , (Left\_bottom) the measurement interference variances  $\varepsilon_n(i)$  at each network node; (Right) Transient network MSD (dB) iteration curves of the RDLMS, DNLMM, DGCLD, DPLMS, and DFair algorithms.

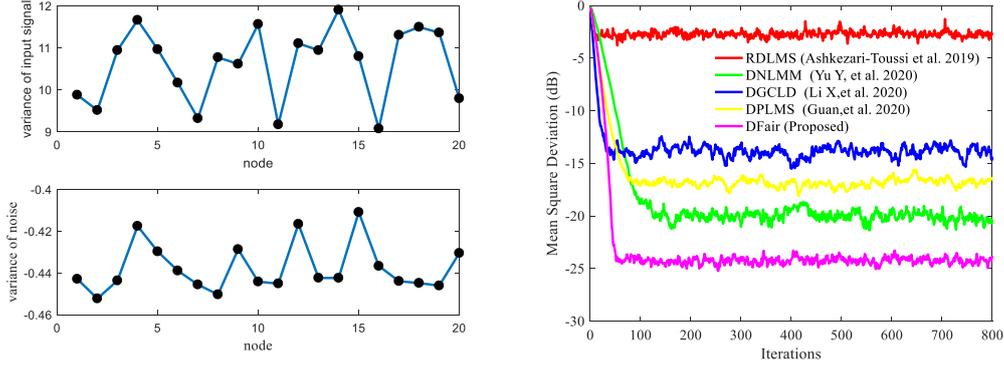


Fig.8. (Left\_top) the input signals  $\{\mathbf{X}_n(i)\}$  variances of at each network node with  $\mathbf{R}_{xx,n} = \sigma_{x,n}^2(t)\mathbf{I}_M, t = 1,2,3,\dots,M$ , (Left\_bottom) the measurement interference variances  $\varepsilon_n(i)$  at each network node; (Right) Transient network MSD (dB) iteration curves of the RDLMS, DNLMM, DGCLD, DPLMS, and DFair algorithms.

Fig. 6, Fig.7, and Fig.8 show that although different types of input signals are used, the DFair algorithm still has a faster convergence rate and lowest steady-state estimated error than the RDLMS, DNLMM, DGCLD, and DPLMS algorithms. Besides, the DFair algorithm is more robust to the input signal. In a word, from *Simulation experiment 1*, we can get the DFair algorithm superior to the RDLMS, DNLMM, DGCLD, and DPLMS algorithms with different input signals, impulsive interference when using the same distributed network topology.

### *Simulation experiment 2*

Illustrating the DFair algorithm is more robust to  $\sigma_f^2$  for impulsive interference with a constant Pr and faster convergence rate and lower steady-state estimated error than the RDLMS, DNLMM, DGCLD, and DPLMS algorithms. There is the same network topology in this experiment: the same Pr of impulsive interference and the same input signal. If any two nodes in network topology are declared neighbors, a certain radius for each node large than or equal to 0.3, the network topology, the input signals and measurement interference are shown in Fig. 9. The MSD iteration curves for RDLMS ( $\mu$  equal to 0.3), DNLMM ( $\mu$  equal to 0.3), DGCLD ( $\mu$  equal to 0.3), DPLMS, and DFair ( $\mu$  equal to 0.3) algorithms in Fig.10 with Pr=0.4 and the cut-value  $\delta = 0.1$  for the DFair algorithm. Also, the independent Monte Carlo number is 10, and each run has 800 iteration numbers.

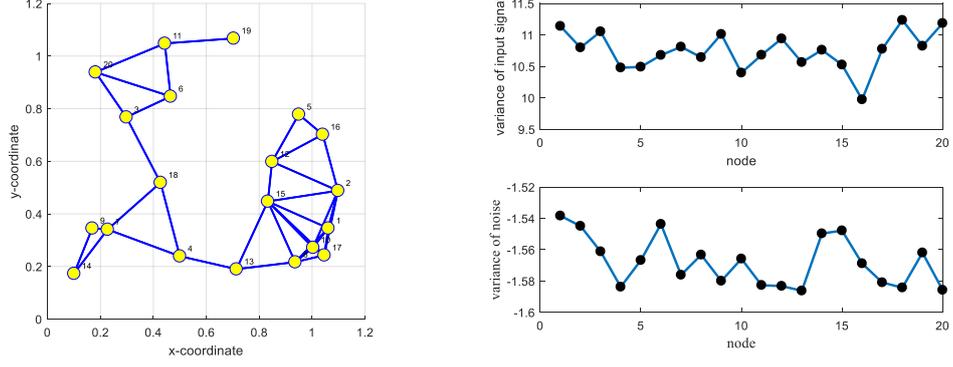


Fig.9. (Left) Random network topology to be decided by a certain radius; (Right\_top) the input signals  $\{\mathbf{X}_n(i)\}$  variances of at each network node with  $\mathbf{R}_{xx,n}$  is a diagonal matrix with possibly different diagonal entries chosen randomly, (Right\_bottom) the measurement interference variances  $\varepsilon_n(t)$  at each network node.

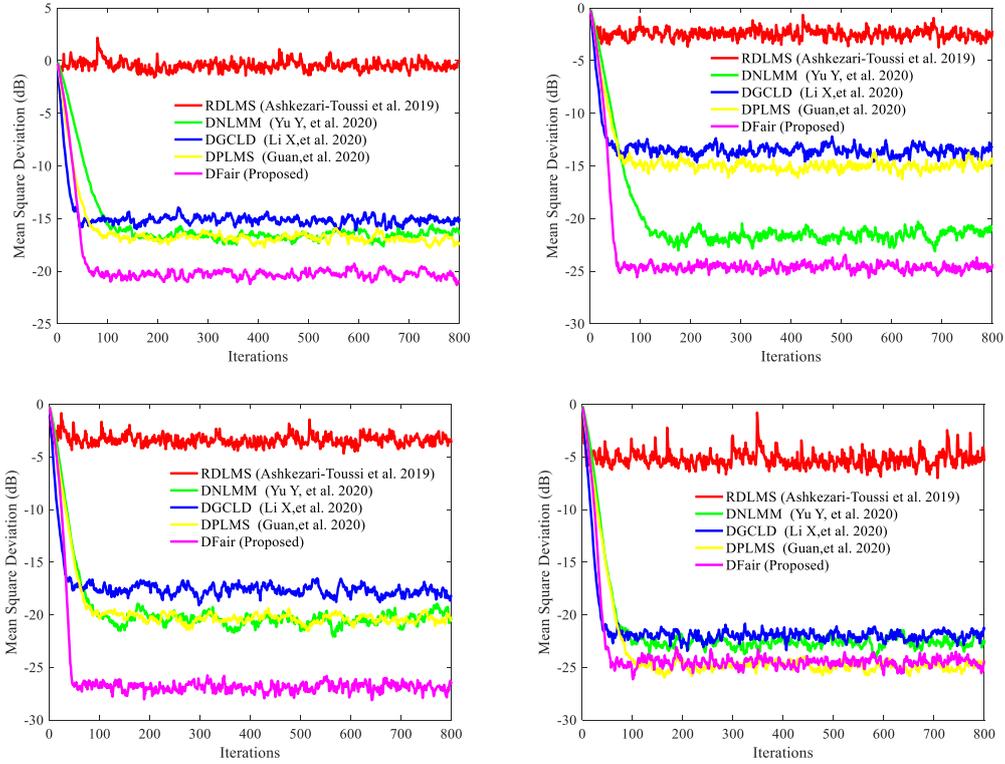


Fig.10. Transient network MSD (dB) iteration curves of the RDLMS, DNLM, DGCLD, DPLMS, and DFair algorithms. (Left\_top) with,  $\sigma_f^2 = 0.04$ , (Right\_top) with  $\sigma_f^2 = 0.06$ , and (Left\_bottom) with  $\sigma_f^2 = 0.08$ , and (Right\_bottom) with  $\sigma_f^2 = 0.1$ .

In this experiment, we want to show that the DFair algorithm is more robust to different the probability density of impulsive interference, so we set four sub-experiments with different density  $\sigma_f^2$  of impulsive interference, the same Pr of impulsive interference, the same input signal, and the same distributed network topology. From Fig. 10, we can find that although different probability density of impulsive interference is considered, the DFair algorithms has a slightly faster rate than

that of the RDLMS, DNLMM, DGCLD, and DPLMS algorithms. The DFair algorithm still has a minor steady-state error than the RDLMS, DNLMM, DGCLD, and DPLMS algorithms. In a word, from *Simulation experiment 2*, we can observe that the DFair algorithm is more robust to impulsive interference than the RDLMS, DNLMM, DGCLD, and DPLMS algorithms.

### Simulation experiment 3

To illustrate our algorithm, it is more robust to Pr for impulsive interference with a constant  $\sigma_f^2$  and has a faster convergence rate and lower steady-state error than the RDLMS, DNLMM, DGCLD, and DPLMS algorithms. In this experiment, there have the same network topology, same  $\sigma_f^2$  for impulsive interference and the same input signal. If any two nodes in network topology are declared neighbors, a certain radius for each node large than or equal to 0.3, the network topology, the input signals and measurement interference are shown in Fig. 11. The MSD iteration curves for RDLMS ( $\mu$  equal to 0.4), DNLMM ( $\mu$  equal to 0.4), DGCLD ( $\mu$  equal to 0.4), DPLMS, and DFair ( $\mu$  equal to 0.4) algorithms in Fig.12 with  $\sigma_f^2 = 0.04$  and the cut-value  $\delta = 0.1$  for the DFair algorithm. Besides, the independent Monte Carlo number is 10, and each run has 800 iteration numbers.

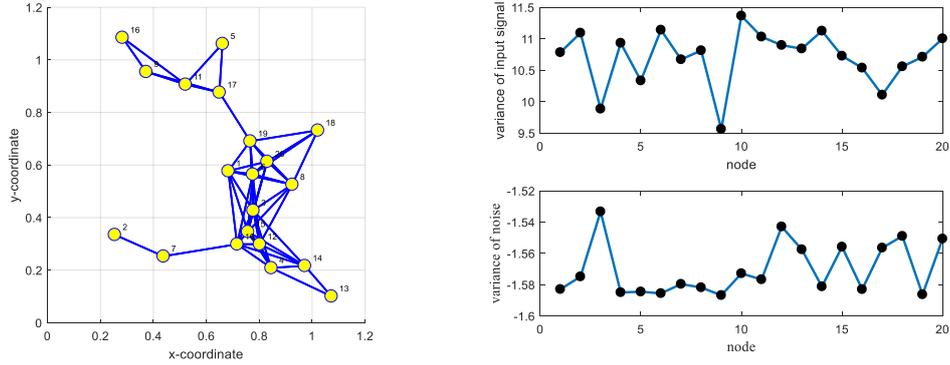
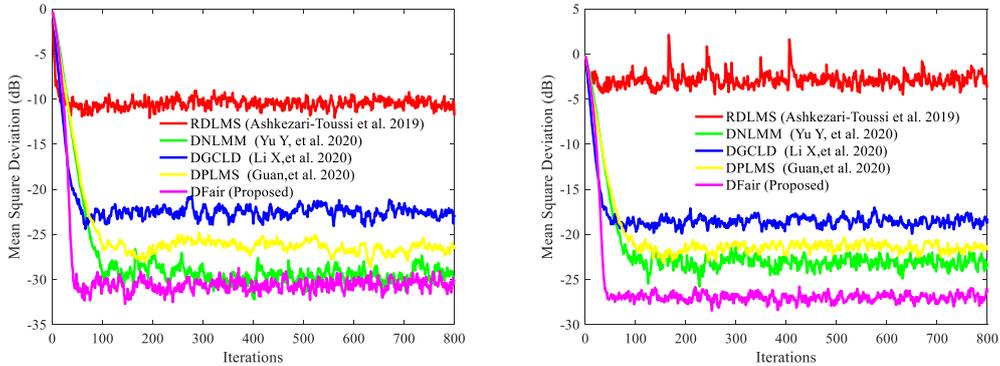


Fig.11. (Left) Random network topology to be decided by a certain radius; (Right\_top) the input signals  $\{\mathbf{X}_n(i)\}$  variances of at each network node with  $\mathbf{R}_{xx,n} = \sigma_{x,n}^2(t)\mathbf{I}_M, t = 1,2,3, \dots, M$ , (Right\_bottom) the measurement interference variances  $\varepsilon_n(t)$  at each network node.



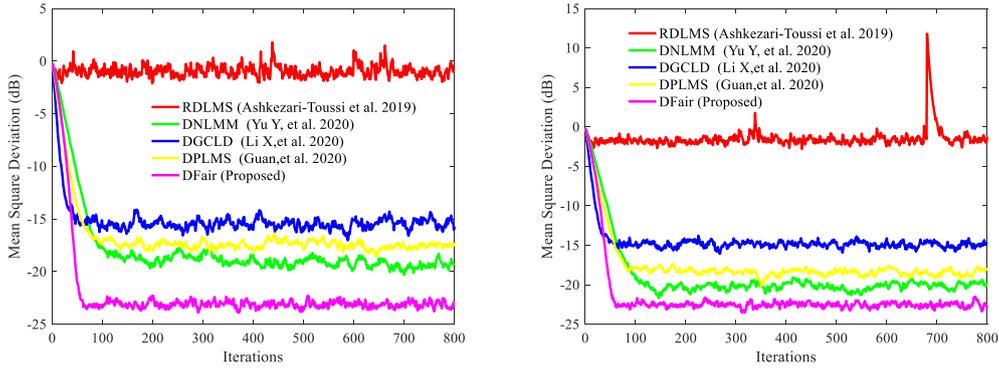


Fig.12. Transient network MSD (dB) iteration curves of the RDLMS, DNLMM, DGCLD, DPLMS, and DFair algorithms. (Left\_top) with  $Pr = 0.1$ , (Right\_top) with  $Pr = 0.2$ , and (Left\_bottom) with  $Pr = 0.4$ , and (Right\_bottom) with  $Pr = 0.6$ .

In this experiment, four sub-experiments with different  $Pr$  of impulsive interference were set with the same density of impulsive interference, the same input signal, and the same distributed network topology. From Fig. 12, we can find that although different  $Pr$  of impulsive interference is considered, the DFair algorithm has a slightly faster rate than that of the RDLMS, DNLMM, DGCLD, and DPLMS algorithms. The DFair algorithm still has a minor steady-state error than the RDLMS, DNLMM, DGCLD, and DPLMS algorithms. In a word, from *Simulation experiment 3*, we can observe that the DFair algorithm is more robust to impulsive interference than the RDLMS, DNLMM, DGCLD, and DPLMS algorithms.

## 5. Conclusion

This paper proposed a novel diffusion algorithm by using the Fair cost function, namely the DFair algorithm. The method is developed based on combining and modifying the DLMS algorithm and the Fair cost function at all distributed network nodes. Compared with some existing distributed adaptive filtering algorithms, the DFair algorithm has low computational complexity. The theoretical analysis demonstrates that the DFair algorithm can achieve an effective estimate from an M-estimation cost function perspective. Besides, theoretical mean behavior interpreted that the DFair algorithm can achieve accurate estimation under the convergence interval. Besides, experimental simulation results showed that the DFair algorithm is more robust to the input signal and impulsive interference than the RDLMS, DNLMM, DGCLD, and DPLMS algorithms. Overall, the DFair algorithm has superior performance when estimating the unknown linear system under different types of input signals and the changeable impulsive interference environments.

## Acknowledgments

This work was supported by the Fundamental Research Funds for the Central

Universities Southwest Minzu University (Grant Number: 2021NQNCZ04, 2021XJTD01), the Wuhu and Xidian University special fund for industry- university-research cooperation (Project No.: XWYCYX-012020014), and the National Natural Science Foundation of China (Grant: 61871420).

## Conflict of Interest

The authors declare that they have no conflict of interest.

## References

- [1] P.S.R. Diniz, Adaptive filtering : algorithms and practical implementation, 4th ed., Springer, New York, 2013.
- [2] A.H. Sayed, Fundamentals of adaptive filtering, IEEE Press Wiley-Interscience, New York, 2003.
- [3] A.H. Sayed, Adaptive Networks, Publisher, City, 2014.
- [4] S.S. Haykin, Adaptive filter theory, Fifth edition. ed., Pearson, Upper Saddle River, New Jersey, 2014.
- [5] W. Liu, J.C. Principe, S. Haykin, Kernel Adaptive Filtering, 2010.
- [6] D.C.J.C. Príncipe, Adaptive Learning Methods for Nonlinear System Modeling, 2018.
- [7] B. Farhang-Boroujeny, Adaptive Filters, 2013.
- [8] C.G. Lopes, A.H. Sayed, Diffusion least-mean squares over adaptive networks: Formulation and performance analysis, Publisher, City, 2008.
- [9] F. Chen, T. Shi, S. Duan, L. Wang, J. Wu, Diffusion least logarithmic absolute difference algorithm for distributed estimation, Publisher, City, 2018.
- [10] F.Y. Huang, J.S. Zhang, S. Zhang, Mean-square-deviation analysis of probabilistic LMS algorithm, Publisher, City, 2019.
- [11] F.S. Cattivelli, A.H. Sayed, Diffusion LMS strategies for distributed estimation, Publisher, City, 2010.
- [12] F.S. Cattivelli, C.G. Lopes, A.H. Sayed, Diffusion recursive least-squares for distributed estimation over adaptive networks, Publisher, City, 2008.
- [13] L. Li, H. Zhao, S. Lv, Diffusion recursive total least square algorithm over adaptive networks and performance analysis, Publisher, City, 2021.
- [14] F. Barani, A. Savadi, H.S. Yazdi, Convergence Behavior of Diffusion Stochastic Gradient Descent Algorithm, Publisher, City, 2021.
- [15] A. Naeimi Sadigh, H. Sadoghi Yazdi, A. Harati, Diversity-based diffusion robust RLS using adaptive forgetting factor, Publisher, City, 2021.
- [16] M. Zhang, D. Jin, J. Chen, J. Ni, Zeroth-Order Diffusion Adaptive Filter Over Networks, Publisher, City, 2021.
- [17] R. Nassif, S. Vlaski, C. Richard, A.H. Sayed, Learning Over Multitask Graphs—Part I: Stability Analysis, Publisher, City, 2020.
- [18] B. Widrow, Thinking about thinking: the discovery of the LMS algorithm, Publisher, City, 2005.
- [19] Introduction, in: A Gentle Introduction to Optimization, 2018, pp. 1-43.
- [20] S. Guan, Z. Li, Nonparametric Variable Step-Size LMAT Algorithm, Publisher, City, 2016.
- [21] E. Eweda, Global Stabilization of the Least Mean Fourth Algorithm, Publisher, City, 2012.
- [22] Z. Wu, J. Shi, X. Zhang, W. Ma, B. Chen, Kernel recursive maximum correntropy, Publisher, City, 2015.
- [23] A. Sayed, Adaptation, Learning, and Optimization over Networks, Publisher, City, 2014.
- [24] S.S. Mahant-Shetti, S. Hosur, A. Gatherer, The log-log LMS algorithm, in: 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing, 1997, pp. 2357-2360.

- [25] K. Xiong, Y. Zhang, S. Wang, Complex-Valued Adaptive Filtering Based on the Random Fourier Features Method, Publisher, City, 2020.
- [26] L. Lu, W. Wang, X. Yang, W. Wu, G. Zhu, Recursive Geman–McClure Estimator for Implementing Second-Order Volterra Filter, Publisher, City, 2019.
- [27] F. Zhang, S. Sun, H. Yi, Robust collaborative recommendation algorithm based on kernel function and Welsch reweighted M-estimator, Publisher, City, 2015.
- [28] J. Ke, C. Gong, T. Liu, L. Zhao, J. Yang, D. Tao, Laplacian Welsch Regularization for Robust Semisupervised Learning, Publisher, City, 2020.
- [29] P. Petrus, Robust Huber adaptive filter, Publisher, City, 1999.
- [30] Z. Li, S. Guan, Diffusion normalized Huber adaptive filtering algorithm, Publisher, City, 2018.
- [31] S. Ashkezari-Toussi, H. Sadoghi-Yazdi, Robust diffusion LMS over adaptive networks, Publisher, City, 2019.
- [32] Y. Yu, H. He, T. Yang, X. Wang, R.C. de Lamare, Diffusion Normalized Least Mean M-estimate Algorithms: Design and Performance Analysis, Publisher, City, 2020.
- [33] J. Koko, S. Jehan-Besson, An Augmented Lagrangian Method for TV  $g + L_1$ -norm Minimization, Publisher, City, 2010.
- [34] P. Soo-Chang, T. Chien-Cheng, Least mean p-power error criterion for adaptive FIR filter, Publisher, City, 1994.
- [35] N. Checchi, S. Marsili-Libelli, Reliability of parameter estimation in respirometric models, Publisher, City, 2005.
- [36] B. Sinova, S. Van Aelst, Tukey's Biweight Loss Function for Fuzzy Set-Valued M-estimators of Location, in: Soft Methods for Data Science, 2017, pp. 447-454.
- [37] G.-C. Rota, Introduction to robust and quasi-robust statistical methods, Publisher, City, 1986.
- [38] F. Wen, Diffusion least-mean P-power algorithms for distributed estimation in alpha-stable noise environments, Publisher, City, 2013.
- [39] J. Ni, Diffusion Sign Subband Adaptive Filtering Algorithm for Distributed Estimation, Publisher, City, 2015.
- [40] J. Ni, J. Chen, X. Chen, Diffusion sign-error LMS algorithm: Formulation and stochastic behavior analysis, Publisher, City, 2016.
- [41] Y. Gao, J. Ni, J. Chen, X. Chen, Steady-state and stability analyses of diffusion sign-error LMS algorithm, Publisher, City, 2018.
- [42] F. Huang, J. Zhang, S. Zhang, Mean-square-deviation analysis of probabilistic LMS algorithm, Publisher, City, 2019.
- [43] J. Fernandez-Bes, V. Elvira, S. Van Vaerenbergh, A probabilistic least-mean-squares filter, in: 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2015, pp. 2199-2203.
- [44] S. Guan, C. Meng, B. Biswal, Diffusion-Probabilistic Least Mean Square Algorithm, Publisher, City, 2020.
- [45] X. Li, M. Feng, F. Chen, Q. Shi, J. Kurths, Robust distributed estimation based on a generalized correntropy logarithmic difference algorithm over wireless sensor networks, Publisher, City, 2020.