



OPEN

Diffusion adaptive filtering algorithm based on the Fair cost function

Sihai Guan^{1,2✉}, Qing Cheng³, Yong Zhao⁴ & Bharat Biswal^{5,6}

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 robust diffusion LMS (RDLMS), diffusion Normalized Least Mean M-estimate (DNLMM), diffusion generalized correntropy logarithmic difference (DGCLD), and diffusion probabilistic least mean square (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.

Adaptive filter algorithms are often used in equalization, active noise control, echo cancellation, biomedical engineering, and many other fields^{1–5}. Distributed adaptive signal processing is an extension of adaptive filters over graphs⁶. There are three widely-used distributed estimation cooperation strategies: incremental, consensus, and diffusion³. 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) scheme⁶ and the combine-then-adapt (CTA) scheme^{3,7}. 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 involves 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 structures' performance for these two kinds of structures, showing that the ATC scheme outperforms the CTA scheme⁶. Then, the ATC scheme becomes a research focus in distributed adaptive filtering algorithms^{6,8–15}. The Wiener filtering principle is the fundamental of the adaptive filtering algorithm¹⁶, 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 are¹⁷. 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 third¹⁸, least mean fourth¹⁹, entropy²⁰, least-squares estimator²¹, absolute value estimator²², the Cauchy²³, Geman–McClure²⁴, Welsch^{25,26}, and Huber function^{27–30}. Specifically, the least-squares estimator is not robust because their influence function is not bounded²¹, and the absolute value estimator is not stable because the function of estimate error ($|e(i)|$) at i -th

¹College of Electronic and Information, Southwest Minzu University, Chengdu, China. ²Key Laboratory of Electronic and Information Engineering, State Ethnic Affairs Commission, Chengdu, China. ³Sichuan Vocational College of Finance and Economics, Chengdu, China. ⁴School of Mechanical and Power Engineering, Henan Polytechnic University, Jiaozuo, China. ⁵The 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. ⁶Department of Biomedical Engineering, New Jersey Institute of Technology (NJIT), Newark, NJ, USA. ✉email: gcihey@sina.cn

is not strictly convex²². Indeed, the second derivative at $e(i) = 0$ is unbounded, and an indeterminate solution may result. The absolute value estimator reduces the influence of large errors, but they still have an influence because of 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 convex³¹. 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 functions³². 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 expected³³. However, many difficulties are encountered 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 function²³. As can be seen from the influence function, the influence of large estimation errors only decreases linearly with their size. The Geman–McClure²⁴ and Welsch^{25,26} functions try to further reduce the effect of large estimation errors. 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, δ 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-differentiable³⁴; it seems complicated to select a cost function for general use without being somewhat arbitrary. Following³⁵, for the regression problems, the best choice is the L_p despite its theoretical no robustness: they are quasi-robust. However, it suffers from computational difficulties. However, the Fair function can yield nicely converging computational procedures. Compared to the quadratic cost function, the adaptive Fair function can track the statistical characteristics of the estimation error, 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 significantly affect the estimation accuracy and most diffusion estimation algorithms on the network. Therefore, designing a robust distributed adaptive algorithm to deal with impulse interference of different intensities is necessary. Recently, Wen proposed the diffusion least mean p-power algorithm³⁶, 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 L_1 -norm subject to a constraint on the estimate weight vectors, Ni and colleagues designed a diffusion sign subband adaptive filtering (DSSAF) algorithm³⁷. Still, the computational complexity of the DSSAF algorithm is relatively large. By combining the diffusion least mean square (DLMS) algorithm⁹ and the sign operation to the estimated error at each iteration moment point, Ni and colleagues derived a diffusion sign-error LMS (DSELMS) algorithm³⁸. 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)³⁹. Besides, inspired by the least logarithmic absolute difference (LLAD) operation, Chen and colleagues designed the DLLAD algorithm⁷. Nevertheless, 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^{8,40}, Guan and colleagues proposed a diffusion probabilistic least mean square (DPLMS) algorithm⁴¹. Based on the Huber objective function, a similar set of algorithms by Guan and colleagues²⁷, Wei and colleagues⁴¹ 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 colleagues²⁹ 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 interference³⁰ for impulsive interference. However, 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 RDLMS²⁹, DNLMM³⁰, DGCLD⁴², and DPLMS⁴¹ algorithms.

The remainder of this paper is organized as follows. The proposed DFair algorithm is proposed in detail in "Proposed the DFair algorithm" section. The DFair algorithm's statistical behavior, including the stability performance, computation complexity, and parameter (δ) for the DFair algorithm, is studied in "Performance analysis" section. The simulation experiments are described in "Simulation results" section. Finally, the conclusion is provided in "Conclusion" section.

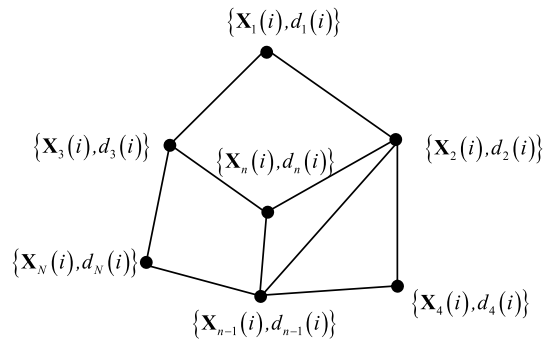


Figure 1. A network consisting of N agents^{18,28,41}.

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.

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

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

where $\mathbf{W}^o(M \times 1)$ is the parameter of interest system, which needs to be estimated and $v(i)$ 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 $\text{sgn}()$ is the symbolic function, and μ is the step size.

The adaptive diffusion filter based on the fair cost function. Our previous paper's research considers a network of N sensor nodes distributed over a geographic area (as Fig. 1)^{18,28,41}. We assume an undirected graph so that if agent $n-1$ is a neighbor of agent n , then agent $n-1$ is also a neighbor of agent n . We assign a pair of nonnegative scaling weights to the edge connecting n and $n-1$. A network is connected if paths with nonzero scaling weights can be found linking any two distinct agents in both directions, either directly when they are neighbors or by passing through intermediate agents when they are not neighbors. In this way, information can flow in both directions between any two agents in the network. $\mathbf{X}_n(i)$ and $d_n(i)$ are the input signals and observation output signals at agent n , respectively.

An adaptive network equips the network's nodes with local learning rules or local adaptive filters. The available communication topology is then employed to efficiently implement a cooperation protocol among the nodes to exploit spatial and temporal information efficiently. Different learning rules allied with different cooperation protocols give rise to different adaptive networks. Based on Fig. 1, using the local cost function $J_n^{local}(\mathbf{W}(i)) = \delta^2 \left(\frac{|e_n(i)|}{\delta} - \log \left(1 + \frac{|e_n(i)|}{\delta} \right) \right)$, we seek the optimal linear estimator that minimizes the global cost function:

$$J^{global}(\mathbf{W}(i)) = \sum_n J_n^{local}(\mathbf{W}(i)) = \sum_n \delta^2 \left(\frac{|e_n(i)|}{\delta} - \log \left(1 + \frac{|e_n(i)|}{\delta} \right) \right) \quad (4)$$

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 ($M \times 1$). Suppose these measurements output follows a standard model given by:

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

Initialize: $\{\mathbf{w}_{n,-1} = \mathbf{0}\}$ for all n , δ .

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

$$\begin{aligned} \boldsymbol{\varphi}_n(i) &= \mathbf{W}_n(i-1) + \mu_n \frac{\delta \mathbf{e}_n(i)}{\delta + |\mathbf{e}_n(i)|} \mathbf{X}_n(i) \\ \mathbf{W}_n(i) &= \sum_{l \in N_n} \mathbf{a}_{l,n} \boldsymbol{\varphi}_l(i) \end{aligned}$$

Table 1. The DFair algorithm summary.

where $\mathbf{W}^0(M \times 1)$ is the unknown system parameter vector with length M , and $v_n(i)$ is the measurement interference with variance $\sigma_{v,n}^2$ and each node has a different value of $v_n(i)$.

In⁹, the DLMS algorithm is obtained by minimizing a linear combination of the local MSE:

$$J_n^{local}(\mathbf{W}(i)) = \sum_{l \in N_n} a_{l,n} E|e_l(i)| = \sum_{l \in N_n} a_{l,n} E \left| d_l(i) - \mathbf{X}_l^T(i) \mathbf{W}(i) \right| \tag{6}$$

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

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

where μ_n is the step size (learning rate), and $\varphi_n(i)$ is the local estimates at node n .

Based on Eq. (7), the derivative of the local instantaneous approximations for $\mathbf{W}_n(i)$ can be formulated as:

$$\nabla J_n^{local}(\mathbf{W}_n(i)) = \frac{\partial}{\partial \mathbf{W}(i)} \left(J_n^{local}(\mathbf{W}_n(i)) \right) = - \frac{\delta \mathbf{e}_n(i)}{\delta + |\mathbf{e}_n(i)|} \mathbf{X}_n(i) \tag{8}$$

As shown in Fig. 1 for the framework of the ATC diffusion strategy, the diffusion algorithms first update the local intermediate estimate by the steepest descent method. Then, each node combines the local intermediate estimates from its neighbors. The steps of the diffusion strategy for distributed estimation are as follows: (1) adaptation step, utilizing the stochastic gradient descent method, the intermediate estimate of node k for the parameter update is derived as:

$$\varphi_n(i) = \mathbf{W}_n(i-1) - \mu_n \nabla J_n^{local}(\mathbf{W}_n(i)) = \mathbf{W}_n(i-1) + \mu_n \frac{\delta \mathbf{e}_n(i)}{\delta + |\mathbf{e}_n(i)|} \mathbf{X}_n(i) \tag{9}$$

where μ_n is a learning step-size and $\mathbf{W}_n(i)$ is the estimate of \mathbf{W}^0 for node n at the time instant i .

(2) combination step, in this step, the node receives all intermediate estimates from its neighbors as follows:

$$\mathbf{W}_n(i) = \sum_{l \in N_n} a_{l,n} \varphi_l(i) \tag{10}$$

For simplicity, a summary of the DFair algorithm procedure based on the analysis presented above is given in Table 1.

Performance analysis

The DFair adaptive filtering algorithms, including 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 , $\hat{\mathbf{W}}_n(i) = \mathbf{W}^0 - \mathbf{W}_n(i)$, $\hat{\varphi}_n(i) = \mathbf{W}^0 - \varphi_n(i)$, which are then collected to form the system weight error vector and intermediate system weight error vector, i.e., $\mathbf{W}(i) = \text{col}\{\mathbf{W}_1(i), \mathbf{W}_2(i), \dots, \mathbf{W}_N(i)\}$, $\boldsymbol{\varphi}(i) = \text{col}\{\varphi_1(i), \varphi_2(i), \dots, \varphi_N(i)\}$, $\hat{\mathbf{W}}(i) = \text{col}\{\hat{\mathbf{W}}_1(i), \hat{\mathbf{W}}_2(i), \dots, \hat{\mathbf{W}}_N(i)\}$, $\hat{\boldsymbol{\varphi}}(i) = \text{col}\{\hat{\varphi}_1(i), \hat{\varphi}_2(i), \dots, \hat{\varphi}_N(i)\}$, $\boldsymbol{\mu}(i) = \text{diag}\left\{ \mu_1 \frac{\delta}{\delta + |\mathbf{e}_1(i)|}, \mu_2 \frac{\delta}{\delta + |\mathbf{e}_2(i)|}, \dots, \mu_N \frac{\delta}{\delta + |\mathbf{e}_N(i)|} \right\}$, and $\mathbf{e}(i) = \text{col}\{e_1(i), e_2(i), \dots, e_N(i)\}$.

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 $\mathbf{R}_{\mathbf{X}\mathbf{X},n} = E[\mathbf{X}_n(i) \mathbf{X}_n^T(i)]$.

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

The DFair algorithm will be obtained. Equations (9) and (10) can be written as

$$\hat{\phi}(i) = \hat{\mathbf{W}}(i-1) - \mathbf{S}_\mu \mathbf{S}_{\mathbf{S}_e}(i) \mathbf{X}(i) \tag{11}$$

$$\hat{\mathbf{W}}(i) = \mathbf{C}^T \hat{\phi}(i) \tag{12}$$

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

Taking the expectation of Eqs. (11) and (12),

$$\mathbb{E}[\hat{\mathbf{W}}(i)] = \mathbf{C}^T \mathbb{E}[\hat{\mathbf{W}}(i-1)] - \mathbf{C}^T \mathbf{S}_\mu \mathbb{E}[\mathbf{S}_{\mathbf{S}_e}(i) \mathbf{X}(i)] \tag{13}$$

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 $\mathbf{e}(i) = \mathbf{S}_X^T(i) \hat{\mathbf{W}}(i-1) + \mathbf{V}(i) = \mathbf{e}_o(i) + \mathbf{V}(i)$. Then, let $\left\{ \begin{array}{l} \mathbf{e}_g(i) = \mathbf{e}_o(i) + \mathbf{g}(i) \\ \mathbf{e}_{Im}(i) = \mathbf{e}_o(i) + \mathbf{Im}(i) \end{array} \right\}$ $\left\{ \begin{array}{l} \mathbf{S}_{S_g}(i) = \mathbf{S}_g(i) \otimes \mathbf{I} \\ \mathbf{S}_{S_{Im}}(i) = \mathbf{S}_{Im}(i) \otimes \mathbf{I} \end{array} \right.$

So,

$$\mathbb{E}[\hat{\mathbf{W}}(i)] = \mathbf{C}^T [\mathbf{I}_{NM} - \mathbf{S}_\mu \text{diag}\{\mathbf{R}_{xx,1}, \mathbf{R}_{xx,2}, \dots, \mathbf{R}_{xx,N}\}] \mathbb{E}[\hat{\mathbf{W}}(i-1)] \tag{14}$$

From Eq. (14), 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. (15)) is given by

$$0 < \mu_n < \frac{2}{\rho_{max}(\mathbf{R}_{xx,n})} \tag{15}$$

where ρ_{max} denotes the maximal eigenvalue of $\mathbf{R}_{xx,n}$. So, based on Eqs. (13) and (15), we obtain $\mathbb{E}[\hat{\mathbf{W}}(\infty)] = 0$.

Parameters δ for the proposed algorithm. The choice of δ 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 δ under the different intensities of impulsive interference. We set six experiment groups in a system identification application to choose the optimum cut-off value δ under different input signals, impulsive interference, and different network structures. Another method that can get the optimum cut-off value δ 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 δ of the proposed two algorithms are obtained based on minimizing the mean-square deviation (MSD) at the current time. The specific derivation methods are similar so that readers can refer to our previous published paper for details⁴³. However, the optimal parameters δ must be time-varying with MSD, which increases the complexity of the algorithm. Therefore, for the sake of simplicity and this paper mainly discusses the design of a novel cost function structure, iterative formulas will increase the computational complexity. So, in this paper, find the approximate optimal parameter value δ of the proposed diffusion adaptive filtering algorithm by designing multiple sets of simulation experiments in different situations. 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 distribution¹⁸ 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 σ_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 σ_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 δ estimators based on their MSD to evaluate the performance of DFair, where $\text{MSD}(i) = \frac{1}{N} \sum_{n=1}^N \mathbb{E}[\|\mathbf{W}_o - \mathbf{W}_n(i)\|^2]$ ^{28,41}. 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%.

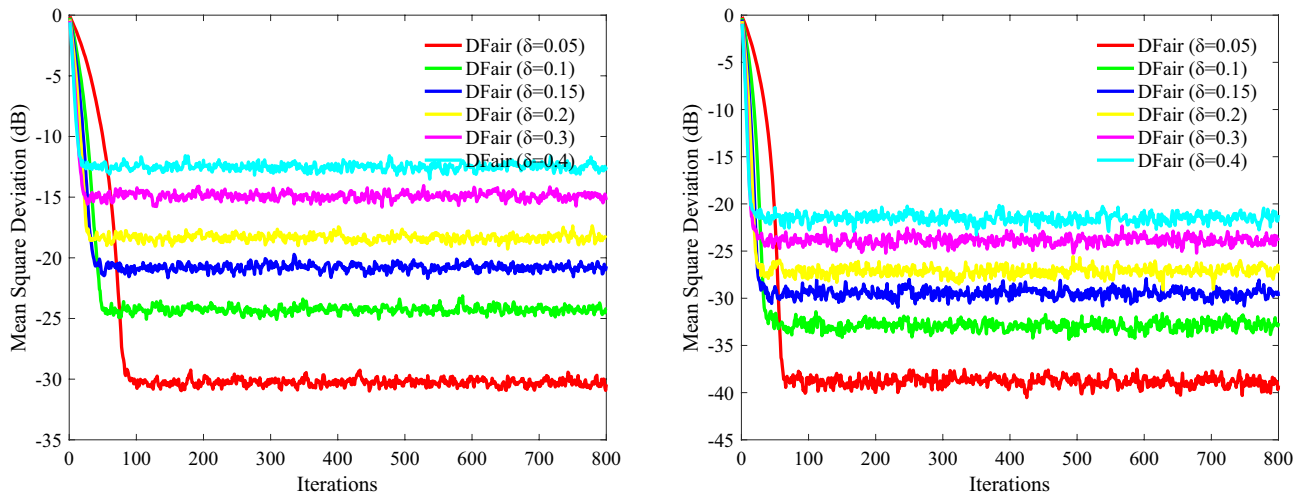


Figure 2. MSD curve with different δ of the DFair algorithm ($\mu=0.4$) when network topology and neighbors to be decided by probability (probability = 0.2): (Left) $\mathbf{R}_{xx,n} = \sigma_{x,n}^2 \mathbf{I}_M$, $\text{Pr}=0.6$, and $\sigma_f^2 = 0.5$. (Right) $\mathbf{R}_{xx,n} = \sigma_{x,n}^2(i) \mathbf{I}_M$, $i = 1, 2, 3, \dots, M$, $\text{Pr}=0.2$, and $\sigma_f^2 = 0.5$.

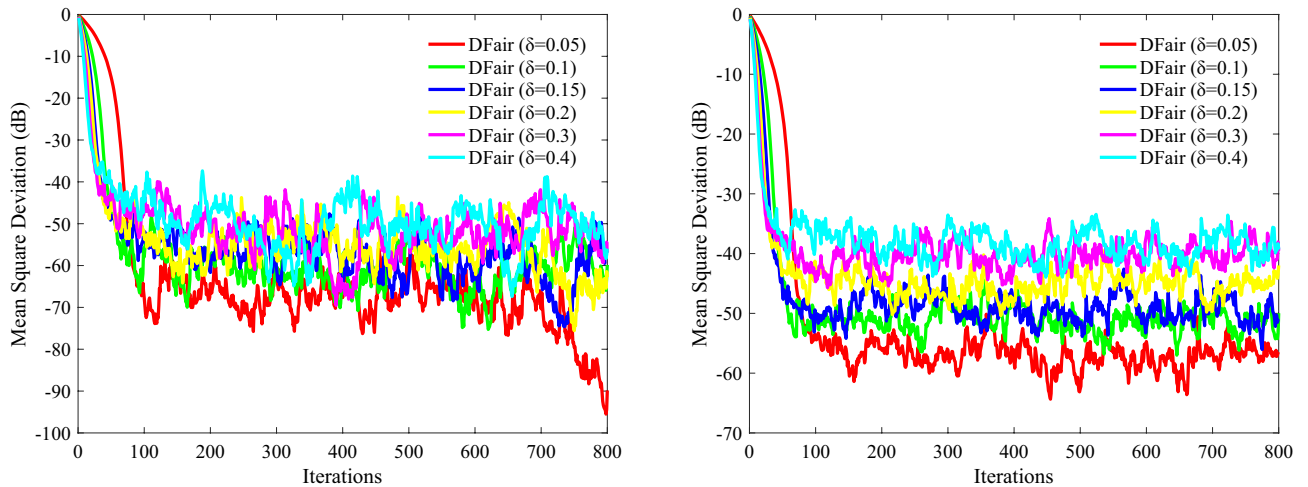


Figure 3. MSD curve with different δ of the DFair algorithm ($\mu=0.4$) when network topology and neighbors to be decided by closeness in the distance (radius = 0.3): (Left) $\mathbf{R}_{xx,n}$ is a diagonal matrix with possibly different diagonal entries chosen randomly, $\text{Pr}=0.4$, and $\sigma_f^2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

Figures 2, 3(Left), and 4(Left) 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 $\delta = 0.1$.

Computational complexity. The adaptive filtering algorithm’s computational complexity is 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)⁴¹ 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)²⁹ will be run. In addition, for the DNLM algorithm, $\hat{\mu}_n(i) = \mu_n \frac{e_n(i)}{\sqrt{1+\mathbf{X}_n(i)^2}}$ in Eq. (13a) and Eq. (15)³⁰ will be consider. Furthermore,

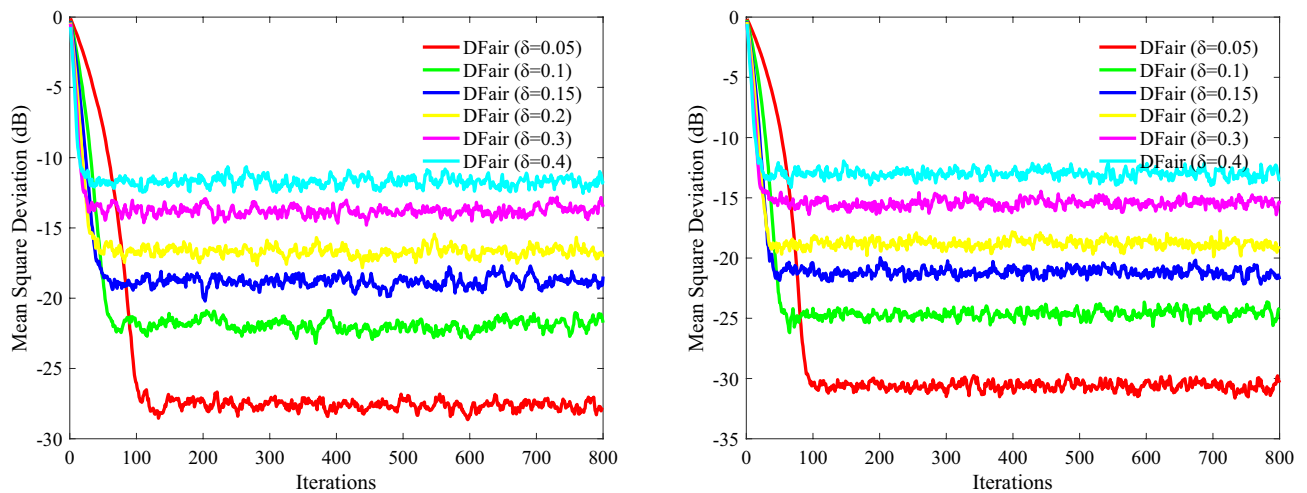


Figure 4. MSD curve with different δ of the DFair algorithm ($\mu=0.4$) when network topology and neighbors to be decided by closeness in the distance (radius=0.3): (Left) $\mathbf{R}_{xx,n}$ is a diagonal matrix with possibly different diagonal entries chosen randomly, $\text{Pr}=0.4$, and $\sigma_f^2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

$\hat{\mu}_n(i) = \mu_n \left(1 + \frac{e_n(i)}{\delta(1+\delta[\exp(-\lambda e_k(i-1)^\alpha)])} \right)$ in Table 1 and Eq. (15)⁴² will add more multiplication operations. So, the computational complexity of the DFair algorithm smaller than RDLMS²⁹, DNLMM³⁰, DGCLD⁴², and DPLMS⁴¹ algorithms.

Simulation results

In this paper, we focus on the distributed adaptive filtering algorithm and compare the DFair algorithm with the RDLMS²⁹, DNLMM³⁰, DGCLD⁴², and DPLMS⁴¹ 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 distribution¹⁸ was added to the system output (as described in “Computational complexity” section). 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^{28,41}.

Simulation experiment 1. Illustrating 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 (μ equal to 0.4), DNLMM (μ equal to 0.4), DGCLD (μ equal to 0.4), DPLMS, and DFair (μ equal to 0.4) algorithms in Figs. 6, 7, and 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.

Figures 6, 7, and 8 show that although different 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 σ_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. This experiment has the same network topology: 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 is larger than or equal to 0.3, and the network topology is shown in Fig. 9(Left). The MSD iteration curves for RDLMS (μ equal to 0.3), DNLMM (μ equal to 0.3), DGCLD (μ equal to 0.3), DPLMS, and DFair (μ equal to 0.3) algorithms in Fig. 10 with $\text{Pr}=0.4$ 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.

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 σ_f^2 of impulsive interference, the same Pr of impulsive interference, the same input signal, and the same distributed network topology. From

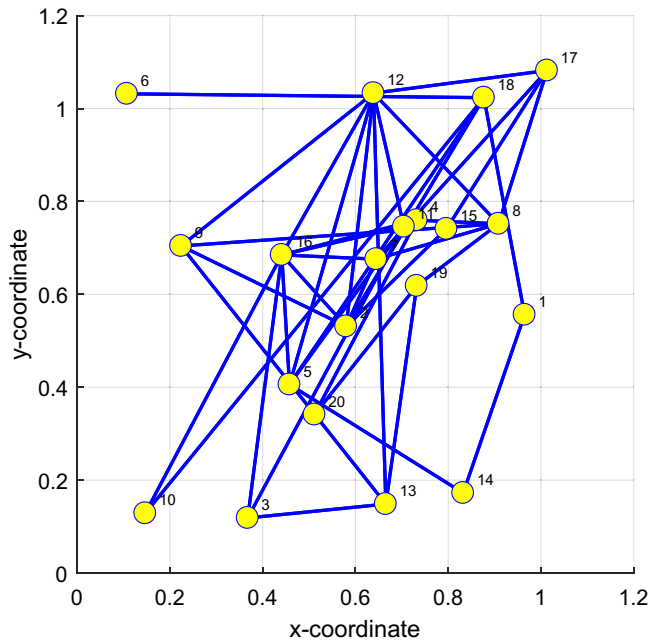


Figure 5. Random network topology is to be decided by probability.

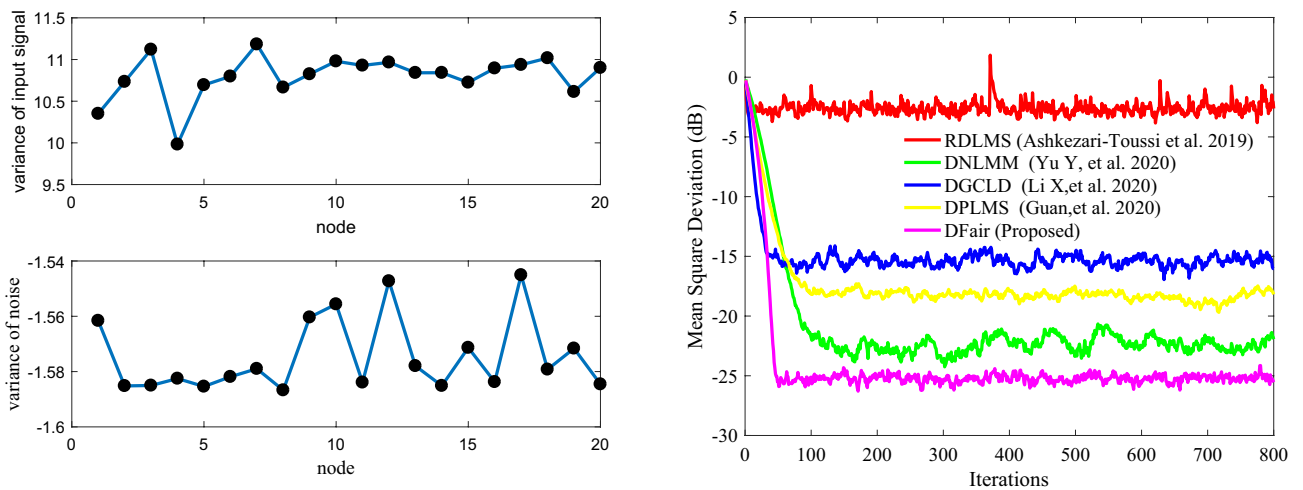


Figure 6. (Left_top) the input signals $\{X_n(i)\}$ variances of at each network node with $R_{x_x, n}$ is a diagonal matrix with possibly different diagonal entries chosen randomly, (Left_bottom) the measurement interference variances $\varepsilon_n(i)$ at each network node; (Right) Transient network MSD (dB) iteration curves of the RDLMS, DNLMM, DGCLD, DPLMS, and DFair algorithms.

Fig. 10, we can find that although different probability density of impulsive interference is considered, the DFair algorithms have a slightly faster rate than 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 σ_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 σ_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 is larger than or equal to 0.3, and the network topology is shown in Fig. 11(Left). The MSD iteration curves for RDLMS (μ equal to 0.4), DNLMM (μ equal to 0.4), DGCLD (μ equal to 0.4), DPLMS, and DFair (μ 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.

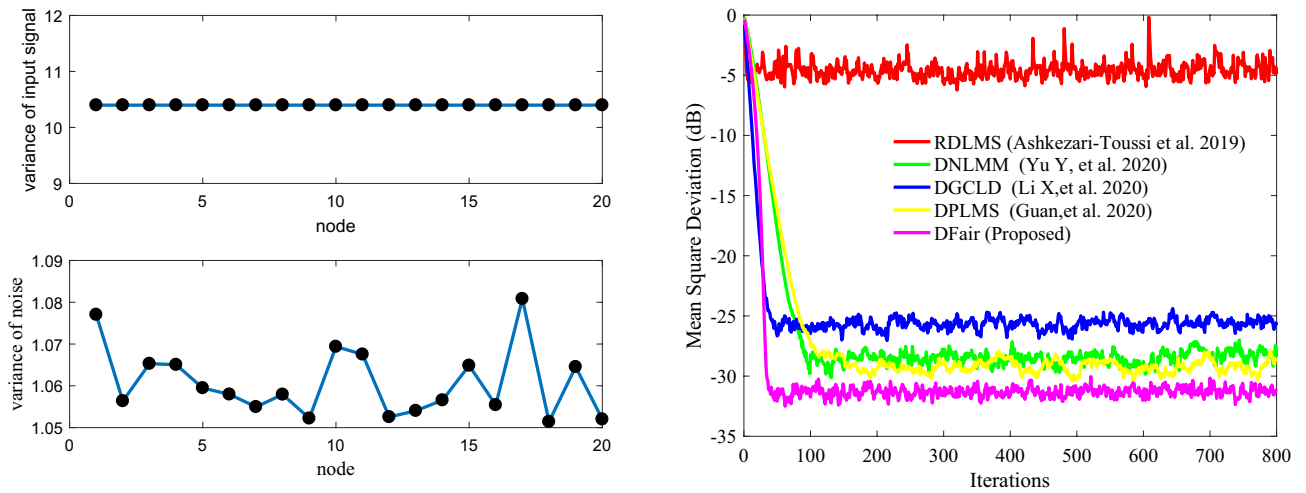


Figure 7. (Left_top) the input signals $\{X_n(i)\}$ variances of at each network node with $R_{xx,n} = \sigma_{x,n}^2 I_M$, (Left_bottom) the measurement interference variances $\varepsilon_n(i)$ at each network node; (Right) Transient network MSD (dB) iteration curves of the RDLMS, DNLMM, DGCLD, DPLMS, and DFair algorithms.

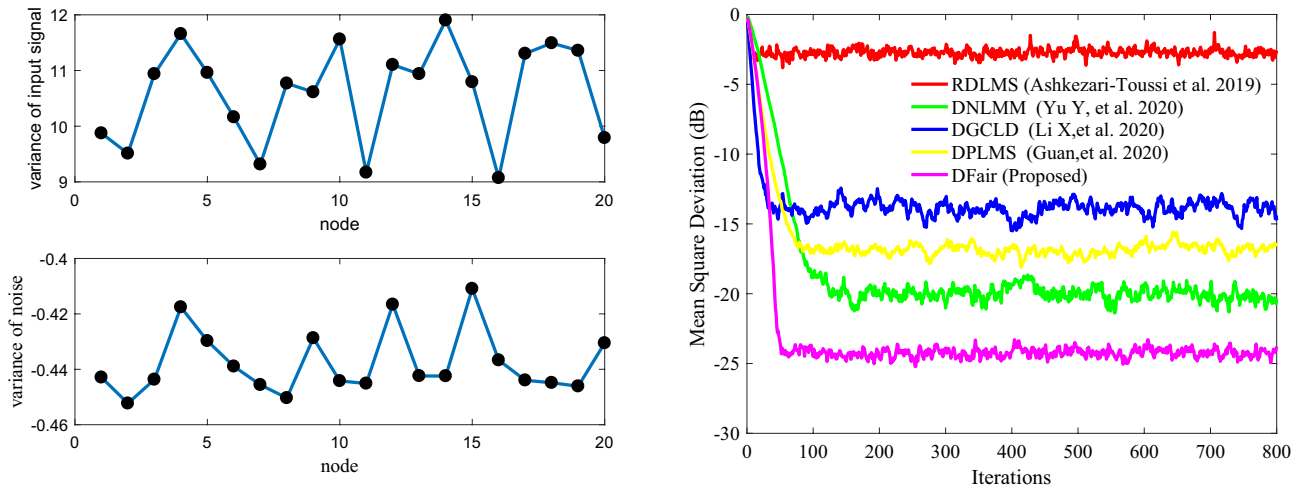


Figure 8. (Left_top) the input signals $\{X_n(i)\}$ variances of at each network node with $R_{xx,n} = \sigma_{x,n}^2(t) I_M, t = 1, 2, 3, \dots, M$, (Left_bottom) the measurement interference variances $\varepsilon_n(i)$ at each network node; (Right) Transient network MSD (dB) iteration curves of the RDLMS, DNLMM, DGCLD, DPLMS, and DFair algorithms.

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 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.

Conclusion

This paper proposed a novel diffusion algorithm by using the Fair cost function, namely the DFair algorithm. The method is developed to combine and modify 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 effectively 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

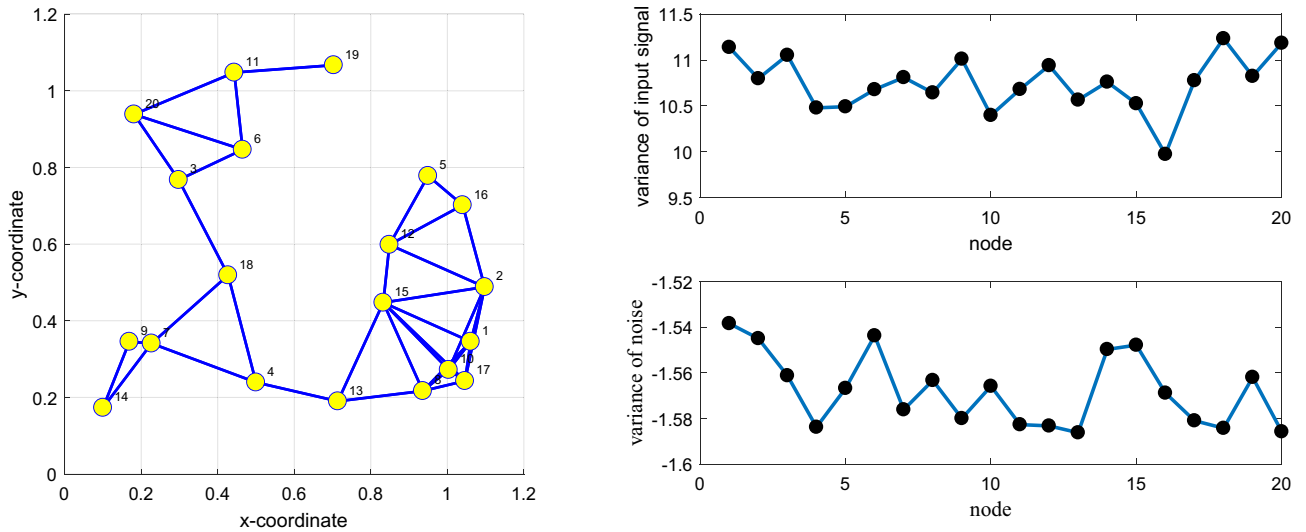


Figure 9. (Left) Random network topology to be decided by a certain radius; (Right_top) the input signals $\{X_n(i)\}$ variances of at each network node with $R_{xx,n}$ is a diagonal matrix with possibly different diagonal entries chosen randomly, (Right_bottom) the measurement interference variances $\varepsilon_n(t)$ at each network node.

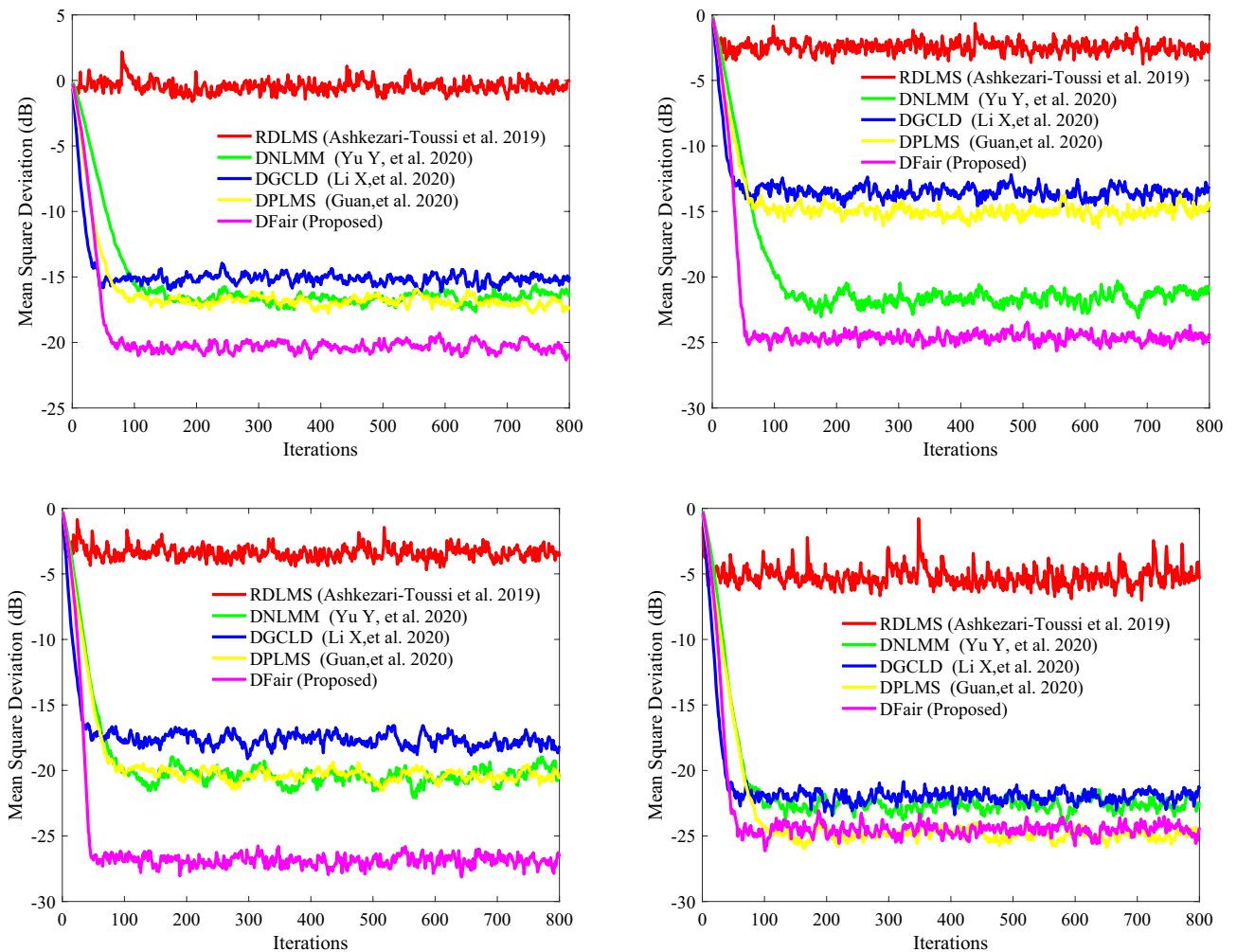


Figure 10. Transient network MSD (dB) iteration curves of the RDLMS, DNLMM, DGCLD, DPLMS, and DFair algorithms. (Up_Left) with $\sigma_f^2 = 0.04$, (Up_right) with $\sigma_f^2 = 0.06$, and (Down_left) with $\sigma_f^2 = 0.08$, and (Down_Right) with $\sigma_f^2 = 0.1$.

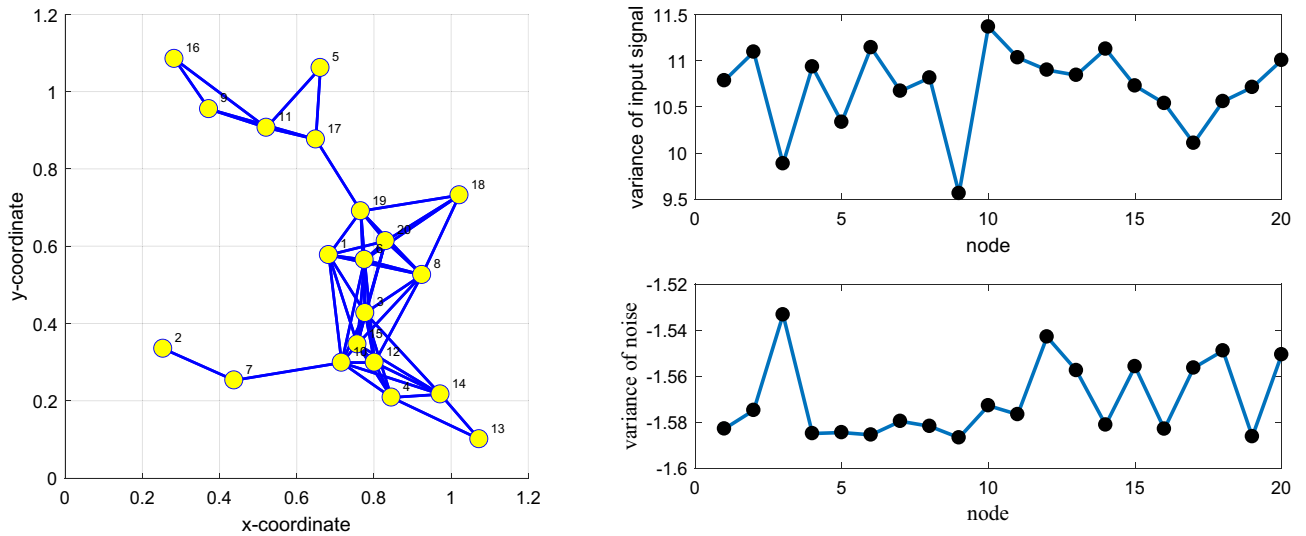


Figure 11. (Left) Random network topology to be decided by a certain radius; (Right_top) the input signals $\{X_n(i)\}$ variances of at each network node with $R_{xx,n} = \sigma_{x,n}^2(t)I_M, t = 1, 2, 3, \dots, M$, (Right_bottom) the measurement interference variances $\varepsilon_n(t)$ at each network node.

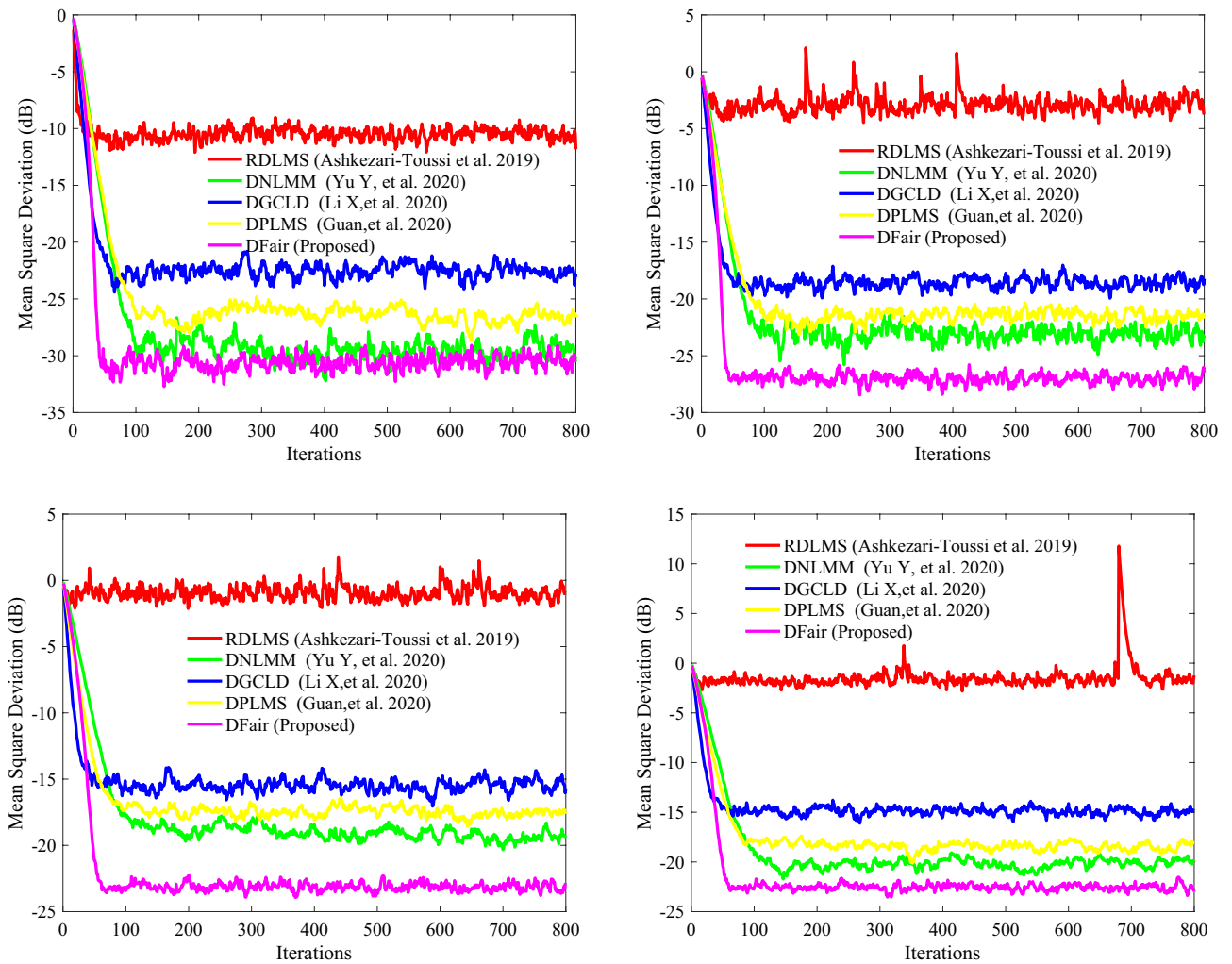


Figure 12. Transient network MSD (dB) iteration curves of the RDLMS, DNLMM, DGCLD, DPLMS, and DFair algorithms. (Up_Left) with $Pr = 0.1$, (Up_right) with $Pr = 0.2$, and (Down_left) with $Pr = 0.4$, and (Down_Right) with $Pr = 0.6$.

performance when estimating the unknown linear system under different input signals and the changeable impulsive interference environments.

Received: 28 May 2021; Accepted: 20 September 2021

Published online: 05 October 2021

References

- Diniz, P. S. R. *Adaptive Filtering: Algorithms and Practical Implementation* 4th edn. (Springer, 2013).
- Sayed, A. H. *Fundamentals of Adaptive Filtering* (IEEE Press Wiley-Interscience, 2003).
- Sayed, A. H. Adaptive networks. *Proc. IEEE*. **102**(4), 460–497. <https://doi.org/10.1109/jproc.2014.2306253> (2014).
- Haykin, S. S. *Adaptive Filter Theory* 5th edn. (Pearson, Upper Saddle River, 2014).
- Danilo Communiello, J. C. P. *Adaptive Learning Methods for Nonlinear System Modeling* 1st edn. (Butterworth-Heinemann, 2018).
- Lopes, C. G. & Sayed, A. H. Diffusion least-mean squares over adaptive networks: Formulation and performance analysis. *IEEE Trans. Signal Process.* **56**(7), 3122–3136 (2008).
- Chen, F., Shi, T., Duan, S., Wang, L. & Wu, J. Diffusion least logarithmic absolute difference algorithm for distributed estimation. *Signal Process.* **142**, 423–430. <https://doi.org/10.1016/j.sigpro.2017.07.014> (2018).
- Huang, F. Y., Zhang, J. S. & Zhang, S. Mean-square-deviation analysis of probabilistic LMS algorithm. *Digit. Signal Process.* **92**, 26–35 (2019).
- Cattivelli, F. S. & Sayed, A. H. Diffusion LMS strategies for distributed estimation. *IEEE Trans. Signal Process.* **58**(3), 1035–1048. <https://doi.org/10.1109/tsp.2009.2033729> (2010).
- Cattivelli, F. S., Lopes, C. G. & Sayed, A. H. Diffusion recursive least-squares for distributed estimation over adaptive networks. *IEEE Trans. Signal Process.* **56**(5), 1865–1877 (2008).
- Li, L., Zhao, H. & Lv, S. Diffusion recursive total least square algorithm over adaptive networks and performance analysis. *Signal Process.* <https://doi.org/10.1016/j.sigpro.2020.107954> (2021).
- Barani, F., Savadi, A. & Yazdi, H. S. Convergence behavior of diffusion stochastic gradient descent algorithm. *Signal Process.* <https://doi.org/10.1016/j.sigpro.2021.108014> (2021).
- Naeimi Sadigh, A., Sadoghi Yazdi, H. & Harati, A. Diversity-based diffusion robust RLS using adaptive forgetting factor. *Signal Process.* <https://doi.org/10.1016/j.sigpro.2020.107950> (2021).
- Zhang, M., Jin, D., Chen, J. & Ni, J. Zeroth-order diffusion adaptive filter over networks. *IEEE Trans. Signal Process.* **69**, 589–602. <https://doi.org/10.1109/tsp.2020.3048237> (2021).
- Nassif, R., Vlaski, S., Richard, C. & Sayed, A. H. Learning over multitask graphs—Part I: Stability analysis. *IEEE Open J. Signal Process.* **1**, 28–45. <https://doi.org/10.1109/ojsp.2020.2989038> (2020).
- Widrow, B. Thinking about thinking: the discovery of the LMS algorithm. *IEEE Signal Process. Mag.* **22**(1), 100–106. <https://doi.org/10.1109/msp.2005.1407720> (2005).
- Introduction. A Gentle Introduction to Optimization. 2018. p. 1–43.
- Guan, S. & Li, Z. Nonparametric variable step-size LMAT algorithm. *Circuits Syst. Signal Process.* **36**(3), 1322–1339. <https://doi.org/10.1007/s00034-016-0356-x> (2016).
- Eweda, E. Global stabilization of the least mean fourth algorithm. *IEEE Trans. Signal Process.* **60**(3), 1473–1477. <https://doi.org/10.1109/tsp.2011.2177976> (2012).
- Wu, Z., Shi, J., Zhang, X., Ma, W. & Chen, B. Kernel recursive maximum correntropy. *Signal Process.* **117**, 11–16. <https://doi.org/10.1016/j.sigpro.2015.04.024> (2015).
- Sayed, A. Adaptation, learning, and optimization over networks. *Found. Trends Mach. Learn.* **7**(4–5), 311–801. <https://doi.org/10.1561/22000000051> (2014).
- Mahant-Shetti, S. S., Hosur, S., & Gatherer, A. The log-log LMS algorithm. In: 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing. pp. 2357–60 (1997).
- Xiong, K., Zhang, Y. & Wang, S. Complex-valued adaptive filtering based on the random Fourier features method. *IEEE Trans. Circuits Syst. II: Express Briefs.* **67**(10), 2284–2288. <https://doi.org/10.1109/tcsii.2019.2957817> (2020).
- Lu, L., Wang, W., Yang, X., Wu, W. & Zhu, G. Recursive Geman–McClure estimator for implementing second-order volterra filter. *IEEE Trans. Circuits Syst. II: Express Briefs.* **66**(7), 1272–1276. <https://doi.org/10.1109/tcsii.2018.2875039> (2019).
- Zhang, F., Sun, S. & Yi, H. Robust collaborative recommendation algorithm based on kernel function and Welsch reweighted M-estimator. *IET Inform. Secur.* **9**(5), 257–265. <https://doi.org/10.1049/iet-ifs.2014.0488> (2015).
- Ke, J. et al. Laplacian welsch regularization for robust semisupervised learning. *IEEE Trans. Cybern.* <https://doi.org/10.1109/tcyb.2019.2953337> (2020).
- Petrus, P. Robust Huber adaptive filter. *IEEE Trans. Signal Process.* **47**(4), 1129–1133. <https://doi.org/10.1109/78.752610> (1999).
- Li, Z. & Guan, S. Diffusion normalized Huber adaptive filtering algorithm J. *Frank. Inst.* **355**(8), 3812–3825. <https://doi.org/10.1016/j.franklin.2018.03.001> (2018).
- Ashkezari-Toussi, S. & Sadoghi-Yazdi, H. Robust diffusion LMS over adaptive networks. *Signal Process.* **158**, 201–209. <https://doi.org/10.1016/j.sigpro.2019.01.004> (2019).
- Yu, Y., He, H., Yang, T., Wang, X. & de Lamare, R. C. Diffusion normalized least mean m-estimate algorithms: design and performance analysis. *IEEE Trans. Signal Process.* **68**, 2199–2214. <https://doi.org/10.1109/tsp.2020.2983905> (2020).
- Koko, J. & Jehan-Besson, S. An augmented lagrangian method for TV g + L 1-norm minimization. *J. Math. Imaging Vis.* **38**(3), 182–196. <https://doi.org/10.1007/s10851-010-0219-1> (2010).
- Soo-Chang, P. & Chien-Cheng, T. Least mean p-power error criterion for adaptive FIR filter. *IEEE J. Sel. Area Commun.* **12**(9), 1540–1547. <https://doi.org/10.1109/49.339922> (1994).
- Checchi, N. & Marsili-Libelli, S. Reliability of parameter estimation in respirometric models. *Water Res.* **39**(15), 3686–3696. <https://doi.org/10.1016/j.watres.2005.06.021> (2005).
- Sinova, B., Van Aelst, S. Tukey's biweight loss function for fuzzy set-valued M-estimators of Location. In: *Soft Methods for Data Science. Advances in Intelligent Systems and Computing.* pp. 447–54 (2017).
- Rota, G.-C. Introduction to robust and quasi-robust statistical methods. *Adv. Math.* [https://doi.org/10.1016/0001-8708\(86\)90012-5](https://doi.org/10.1016/0001-8708(86)90012-5) (1986).
- Wen, F. Diffusion least-mean P-power algorithms for distributed estimation in alpha-stable noise environments. *Electron. Lett.* **49**(21), 1355–1356. <https://doi.org/10.1049/el.2013.2331> (2013).
- Ni, J. Diffusion sign subband adaptive filtering algorithm for distributed estimation. *IEEE Signal Process. Lett.* **22**(11), 2029–2033. <https://doi.org/10.1109/lsp.2015.2454055> (2015).
- Ni, J., Chen, J. & Chen, X. Diffusion sign-error LMS algorithm: Formulation and stochastic behavior analysis. *Signal Process.* **128**, 142–149. <https://doi.org/10.1016/j.sigpro.2016.03.022> (2016).
- Gao, Y., Ni, J., Chen, J. & Chen, X. Steady-state and stability analyses of diffusion sign-error LMS algorithm. *Signal Process.* **149**, 62–67. <https://doi.org/10.1016/j.sigpro.2018.02.033> (2018).

40. Fernandez-Bes, J., Elvira, V., & Van Vaerenbergh S. A probabilistic least-mean-squares filter. In: 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2199–203 (2015).
41. Guan, S., Meng, C. & Biswal, B. Diffusion-probabilistic least mean square algorithm. *Circuits Syst. Signal Process.* <https://doi.org/10.1007/s00034-020-01518-3> (2020).
42. Li, X., Feng, M., Chen, F., Shi, Q. & Kurths, J. Robust distributed estimation based on a generalized correntropy logarithmic difference algorithm over wireless sensor networks. *Signal Process.* <https://doi.org/10.1016/j.sigpro.2020.107731> (2020).
43. Guan, S. & Li, Z. Optimal step size of least mean absolute third algorithm. *Signal Image Video Process.* **11**(6), 1105–1113. <https://doi.org/10.1007/s11760-017-1064-0> (2017).

Acknowledgements

The Fundamental Research Funds supported this work for the Central Universities Southwest Minzu University (Grant Number: 2021NQNCZ04), the Wuhu and Xidian University special fund for industry-university-research cooperation (Project No.: XWYCY-012020014), and the National Natural Science Foundation of China (Grant: 61871420).

Author contributions

S.G. and Q.C. wrote the main manuscript text. Y.Z. and B.B. revised the manuscript. All authors reviewed the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to S.G.

Reprints and permissions information is available at www.nature.com/reprints.

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



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2021