Circuits, Systems, and Signal Processing (2021) 40:1295–1313
https://doi.org/10.1007/s00034-020-01518-3

Diffusion-Probabilistic Least Mean Square Algorithm

Sihai Guan[1] · Chun Meng[1] · Bharat Biswal[1,2]

Received: 17 October 2019 / Revised: 24 May 2020 / Accepted: 26 May 2020 / Published online: 24 August 2020

Abstract

In this paper, a novel diffusion estimation algorithm is proposed from a probabilistic perspective by combining the diffusion strategy and the probabilistic least mean square (LMS) at all distributed network nodes. The proposed method, namely diffusion-probabilistic LMS (DPLMS), is more robust to the input signal and impulsive noise than previous algorithms like the diffusion sign-error LMS (DSE-LMS), diffusion robust variable step-size LMS (DRVSSLMS), and diffusion least logarithmic absolute difference (DLLAD) algorithms. Instead of minimizing the estimation error, the DPLMS algorithm is based on approximating the posterior distribution with an isotropic Gaussian distribution. In this paper, the stability of the mean estimation error and the computational complexity of the DPLMS algorithm are analyzed theoretically. Simulation experiments are conducted to explore the mean estimation error for the DPLMS algorithm with varied conditions for input signals and impulsive interferences, compared to the DSE-LMS, DRVSSLMS, and DLLAD algorithms. Both results from the theoretical analysis and simulation suggest that the DPLMS algorithm has superior performance than the DSE-LMS, DRVSSLMS, and DLLAD algorithms when estimating the unknown linear system under the changeable impulsive noise environments.

Keywords Distributed network · Impulsive noise · Input signals · Probabilistic

✉ Sihai Guan
gcihey@sina.cn

✉ Chun Meng
chunmeng@uestc.edu.cn

✉ Bharat Biswal
bbiswal@gmail.com

[1] 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

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

Birkhäuser

1 Introduction

Over the last decade, distributed estimation has received increased attention in the multi-task network and single-task network [3, 11, 15, 16, 21, 29], especially for the parameter estimation of linear frequency modulation (LFM) signals [24], as well as heat and mass transfer [34]. In a single-task network, the same target parameters are collaboratively estimated for all network nodes, while in a multi-task network, target parameters of each network node need to be estimated separately [20]. This can be improved since diffusion is important for distributed estimation. Diffusion is a common physical phenomenon in the complex transport process. For example, Yang et al. proposed some novel methods to deal with the analytical solution in heat and mass transfer, such as fractional derivatives [33, 37] and kernel functions [35, 36]. Local cooperation and data processing are two important components of distributed data processing technology, and there are three cooperative strategies widely used for the distributed network estimation, namely incremental, consensus, and diffusion strategies [20]. However, the incremental strategy has the same limitation with consensus strategy, which is the asymmetry problem. The asymmetry problem is likely to cause unstable growth of a network state. As the diffusion strategy is more adaptive to environmental changes, it is not limited by this asymmetry problem. The diffusion strategy includes the adapt-then-combine (ATC) scheme [20] and the combine-then-adapt (CTA) scheme [3]. For the ATC scheme, the first step is to update the local estimation by using an adaptive filtering algorithm, and then, the intermediate estimates for the neighbors are fused. For the CTA scheme, first data fusion is performed, and then the local estimation is updated by using the adaptive filtering algorithm. Cattivelli and Sayed analyzed the performance of both schemes, and demonstrated that the ATC scheme outperformed the CTA scheme [20]. After that, a variety of diffusion estimation algorithms using ATC scheme have been proposed [2, 4, 9, 12–14, 17, 18, 31, 32, 38].

For the distributed estimation, a significant challenge is how to cope with the impulsive noise, since most of diffusion estimation algorithms over a network are highly susceptible to various impulsive noises with different types of input signals. Therefore, it is necessary to design a robust diffusion estimation algorithm to impulse noises of different intensities. Recently, using a fixed power p, Wen [31] proposed the diffusion least mean p-power (DLMP) algorithm, which is robust to the interference of generalized Gaussian distribution environments. But the performance of the DLMP algorithm is limited by the selected parameter p. On the other hand, minimizing the L1-norm subject to a constraint on the estimation vectors, Ni [22] designed a diffusion sign subband adaptive filtering (DSSAF) algorithm. The DSSAF algorithm is robust against impulsive interferences while its complexity is relatively large. By modifying the diffusion least mean square (DLMS) algorithm [3] and then applying the sign operation to the estimated error signals at each iteration moment point, Ni et al. [17] proposed a diffusion sign-error LMS (DSE-LMS) algorithm. Ye et al. [7] provided steady-state and stability analyses of the DSE-LMS algorithm, showing that although the DSE-LMS algorithm architecture is simple and easy to implement, the DSE-LMS algorithm has a major drawback, i.e., the steady-state error is high. Additionally, based on the Huber objective function, a similar set of algorithms have been proposed by

Zhi [13], Wei et al. [9], and Soheila et al. [1], which are the Diffusion Normalized Huber adaptive filtering (DNHuber) [13], the diffusion robust variable step-size LMS (DRVSSLMS) [9], and the Robust Diffusion LMS (RDLMS) [1] algorithms, respectively. But, the RDLMS algorithm is not designed for impulse noise. For the DNHuber algorithm, its robustness against the impulsive interference and input signal need to be explored more comprehensively. The DRVSSLMS algorithm has high computational complexity. Last but not the least, inspired by the least logarithmic absolute difference operation, Chen et al. [4] designed another distributed cost function for the diffusion least logarithmic absolute difference (DLLAD) algorithm. But it is unclear whether the DLLAD algorithm is robust to the input signal and impulse noise.

In many engineering applications, the impulsive interference is present with various characteristics so that the input signal may not behave in an ideal fashion as assumed. For example, if the input signal is strongly correlated in time, different intensity sparsity ratios might lead to the performance degradation of parameter estimation algorithms, even when the algorithms could eventually stop. As stated above, the DSE-LMS, DRVSSLMS, and DLLAD algorithms may not be robust against the input signal and impulsive noise. Therefore, it becomes essential to design a distributed, adaptive algorithm that is more robust to the input signal and impulsive noise than the DSE-LMS, DRVSSLMS, and DLLAD algorithms. These factors, such as various types of the input signals or impulsive noises, usually reinforce the randomness and non-stationarity of system identification. Approximating the posterior distribution can be used to tackle this problem. In a recent study, Fernandez-Bes et al. [6] proposed the probabilistic LMS algorithm, and the theoretical stochastic behavior of the probabilistic LMS algorithm has since been analyzed [10].

In earlier studies, the problem of identification was considered in either a deterministic framework or with the assumption that the observation noise has a Gaussian distribution. However, it is known that in some realistic studies, there will be outliers due to the non-Gaussian noise. For such instances, based on the modified and extended Masreliez-Martin filter, Stojanovic et al. [26] proposed an algorithm for the joint state estimation and parameter estimation of stochastic nonlinear systems in the presence of non-Gaussian noises. To cope with outliers in the estimation of unknown system problems, they also designed an identification of the output error model by using a constraint on the output power [25], and proposed an adaptive two-stage procedure for generating the input signal [28]. Besides, for the state estimation of nonlinear multivariable stochastic systems, they derived the robust extended Kalman filter algorithm [27]. It should be noted that in the probabilistic LMS algorithm, the system coefficients are considered to be an unknown time-varying system whose time-varying characteristics satisfy the Gaussian distribution while the observation noise is a stationary additive noise with zero mean and constant variance. Moreover, the α-stable distribution has been used to model the non-Gaussian noise and is widely utilized as an impulsive noise [22]. That is, the characteristic function of impulsive noise is defined as the α-stable distribution. Then, Bayes' theorem and the maximum a posteriori (MAP) estimate were used as an approximation for the predictive step. Thus, regardless the noise distribution, the probabilistic LMS algorithm is more robust to impulsive interference. From the probabilistic perspective, the probabilistic LMS algorithm has two major advantages. First, this algorithm has an adaptable step size, making it suitable for both stationary and non-stationary environments, and having less free parameters. Second, the use of a probabilistic model provides an estimation of the error variance, which is useful in many applications. Thus, from approximating the posterior distribution with an isotropic Gaussian distribution, we propose a diffusion probabilistic least mean square (DPLMS) algorithm, which combines the ATC diffusion strategy and the probabilistic LMS algorithm [6, 10] at all distributed network nodes. The stability of the mean estimation error and the computational complexity of this proposed algorithm are analyzed theoretically. Also, simulation experiments are conducted to explore the mean estimation error for the DPLMS algorithm with varied conditions for input signal and impulsive interference, compared to the DSE-LMS, DRVSSLMS, and DLLAD algorithms.

This paper is organized as follows: The proposed DPLMS algorithm is derived in Sect. 2. The theoretical stability of the mean estimation error, mean square performance, and the computational complexity is analyzed in Sect. 3. Then, in Sect. 4, we design several simulation experiments. Finally, in Sect. 5, conclusions are drawn.

2 The Proposed DPLMS Algorithm

2.1 The Modified Probabilistic LMS Algorithm

The probabilistic LMS is a VSS-LMS algorithm [10]. It has been derived from a probabilistic perspective, such as MAP estimation. Now, we will review and modify the probabilistic LMS algorithm in this subsection. Firstly, consider an unknown time-varying system with length M, with its coefficients at iteration moment i,

$$\mathbf{W}_o(i) = \mathbf{W}_o(t-1) + \boldsymbol{p}(i) \tag{1}$$

In Eq. (1), $\boldsymbol{p}(i)$ is a white Gaussian noise with zero-mean. The time-varying $\mathbf{W}_o$ (i) obeys Eq. (2), which is a Gaussian distribution:

$$p(\mathbf{W}_o(i)|\mathbf{W}_o(i-1)) = \mathrm{N}\Big(\mathbf{W}_o(i); \mathbf{W}_o(i-1), \boldsymbol{I}\sigma_{\boldsymbol{p}}^2\Big) \tag{2}$$

where $\boldsymbol{I}\sigma_{\boldsymbol{p}}^2 = \mathrm{E}\big[\boldsymbol{p}(i)\boldsymbol{p}^{\mathrm{T}}(i)\big], \sigma_{\boldsymbol{p}}^2$ is variance for $\boldsymbol{p}(i)$, and $\boldsymbol{I}$ is a matrix with the appropriate size.

For this unknown time-varying system, when we input a signal $\mathbf{X}(i) = [x(i), x(i+1), x(i+2), \ldots, x(i+M-1)]^T$, the desired signal is $d(i)$,

$$d(i) = \mathbf{W}_o^{\mathrm{T}}(i)\mathbf{X}(i) + \varepsilon(i) \tag{3}$$

where $\varepsilon(i)$ is a stationary additive noise with zero mean and variance of σ_ε^2. Also, we assume that Eq. (3) obeys Eq. (4), which is a Gaussian distribution also,

$$p(d(i)|\mathbf{X}(i), \mathbf{W}_o(i)) = \mathrm{N}\Big(d(i); \mathbf{W}_o^{\mathrm{T}}(i)\mathbf{X}(i), \sigma_\varepsilon^2\Big) \tag{4}$$

supposing that the estimation mean vector and variance of $\mathbf{W}_o(i)$ at iteration i are θ (i) and $\sigma^2(i)$, respectively. So, by using an isotropic spherical Gaussian distribution [as Eq. (5)] to approximate the posterior distribution, we can obtain the probabilistic LMS algorithm.

$$p(\mathbf{W}_o(i)|\mathbf{Z}_i) = \mathrm{N}\Big(\mathbf{W}_o(i); \theta(i), \boldsymbol{I}\sigma^2(i)\Big) \tag{5}$$

where $\mathbf{Z}_i = \{\mathbf{X}(k), d(k)\}_{k=1}^{i}$.

So, based on Eq. (5), we can get

$$\begin{aligned} p(\mathbf{W}_o(i)|\mathbf{Z}_{i-1}) &= \int p(\mathbf{W}_o(i)|\mathbf{W}_o(i-1))p(\mathbf{W}_o(i-1)|\mathbf{Z}_{i-1})d\mathbf{W}_o(i-1) \\ &= \mathrm{N}\Big(\mathbf{W}_o(i); \theta(i-1), \boldsymbol{I}\Big(\sigma^2(i-1) - \sigma_\rho^2\Big)\Big) \end{aligned} \tag{6}$$

where $p(\mathbf{W}_o(i-1)|\mathbf{Z}_{i-1}) = \mathrm{N}\big(\mathbf{W}_o(i-1); \theta(i-1), \boldsymbol{I}\sigma^2(i-1)\big)$ at iteration $i-1$. By combining Bayes' rule and Eq. (6), then we approximate the posterior with an isotropic Gaussian as Eq. (7).

$$\begin{aligned} p(\mathbf{W}_o(i)|\mathbf{Z}_i) &= p(d(i)|\mathbf{u}(i), \mathbf{W}_o(i))p(\mathbf{W}_o(i)|\mathbf{Z}_{i-1}) \\ &= \mathrm{N}\Big(\mathbf{W}_o(i); \theta(i), \boldsymbol{I}\sigma^2(i)\Big) \end{aligned} \tag{7}$$

where

$$\theta(i) = \theta(i-1) + \alpha(i)\Big[d(i) - \mathbf{X}^{\mathrm{T}}(i)\theta(i-1)\Big]\mathbf{X}(i) \tag{8}$$

and

$$\sigma^2(i) = \left[1 - \frac{\frac{\sigma^2(i-1)+\sigma_\rho^2}{\left[\sigma^2(i-1)-\sigma_\rho^2\right]\|\mathbf{X}(i)\|^2+\sigma_\varepsilon^2}\|\mathbf{X}(i)\|^2}{L}\right]\left[\sigma^2(i-1) - \sigma_\rho^2\right] \tag{9}$$

Set $\alpha(i) = \frac{\sigma^2(i-1)+\sigma_\rho^2}{\left[\sigma^2(i-1)-\sigma_\rho^2\right]\|\mathbf{X}(i)\|^2+\sigma_\varepsilon^2}$, then $\sigma^2(i) = \left[1 - \frac{\alpha(i)\|\mathbf{X}(i)\|^2}{L}\right]\left[\sigma^2(i-1) - \sigma_\rho^2\right]$.

So, based on Eq. (7), by using MAP, i.e., $\mathbf{W}(i) = \mathrm{argmax}_{\mathbf{W}_o(i)} p(\mathbf{W}_o(i)|\mathbf{Z}_i)$, we can get the recursive estimation weight-vector equation of the probabilistic LMS algorithm.

$$\mathbf{W}(i+1) = \mathbf{W}(i) + \alpha(i)e(i)\mathbf{X}(i) \tag{10}$$

where $e(i)$ is the error signal, as $e(i) = d(i) - y(i)$,

$$e(i) = d(i) - \mathbf{W}^{\mathrm{T}}(i-1)\mathbf{X}(i)$$

$$\begin{aligned}
&= \mathbf{W}_o^{\mathrm{T}}(i)\mathbf{X}(i) + \varepsilon(i) - \mathbf{W}^{\mathrm{T}}(i-1)\mathbf{X}(i) \\
&= \mathbf{X}^{\mathrm{T}}(i)(\mathbf{W}_o(i-1) + \boldsymbol{p}(i)) - \mathbf{W}^{\mathrm{T}}(i-1)\mathbf{X}(i) + \varepsilon(i) \\
&= \mathbf{X}^{\mathrm{T}}(i)[\mathbf{W}_o(i-1) - \mathbf{W}(i-1)] + \mathbf{X}^{\mathrm{T}}(i)\boldsymbol{p}(i)) + \varepsilon(i) \\
&= \mathbf{X}^{\mathrm{T}}(i)\hat{\mathbf{W}}(i-1) + \mathbf{X}^{\mathrm{T}}(i)\boldsymbol{p}(i)) + \varepsilon(i) \qquad (11)
\end{aligned}$$

and $\hat{\mathbf{W}}(i-1) = \mathbf{W}_o(i-1) - \mathbf{W}(i-1)$ is the weight-deviation vector.

To facilitate comparative analysis, we make a slight adjustment to the probabilistic LMS that is to add a constant value parameter in Eq. (8) and Eq. (10), as Eq. (12) and Eq. (13).

$$\theta(i) = \theta(i-1) + \tau\alpha(i)\Big[d(i) - \mathbf{X}^{\mathrm{T}}(i)\theta(i-1)\Big]\mathbf{X}(i) \qquad (12)$$

$$\mathbf{W}(i+1) = \mathbf{W}(i) + \tau\alpha(i)e(i)\mathbf{X}(i) \qquad (13)$$

where $0 < \tau \leq 1$ is a constant value parameter.

2.2 The Diffusion Probabilistic LMS (DPLMS) Algorithm

Now in this subsection, we will extend the modified probabilistic LMS algorithm [as Eqs. (12)–(13)] to the diffusion probabilistic LMS algorithm. In the beginning, we set a distributed network consisting of N network nodes (or agents), where each node measures its data $\{\mathbf{X}_n(i), d_n(i)\}$ to estimate an unknown parameter vector $\mathbf{W}_o$ of length M vector through a linear model at agent $n \in \{1, 2, \ldots, N\}$.

$$d_n(i) = \mathbf{X}_n(i)\mathbf{W}_o + \varepsilon_n(i) \qquad (14)$$

In Eq. (14) $\mathbf{X}_n(i)$ is the input signal at n-node, $d_n(i)$ is the desired signal at n-node,$\varepsilon_n(t)$ is the measurement noise at n-node with variance $\sigma_{\varepsilon,n}^2$ and independent of any other data. Under reference [13], we also set $\mathbf{X}_n(i)$ as temporally white Gaussian with zero mean and spatially independent with $\mathbf{R}_{xx,n}(i) = \mathrm{E}\big[\mathbf{X}_n^{\mathrm{T}}(i)\mathbf{X}_n(i)\big] > 0$.

The global cost function of DPLMS can be formulated as:

$$\begin{aligned}
J^{\text{global}}(\mathbf{W}(i)) &= \sum_n J_n^{\text{local}}(\mathbf{W}(i)) \\
&= \sum_n J_n^{\text{local}}\big(\text{argmax}_{\mathbf{W}_o(i)}\, p(\mathbf{W}_o(i)|\mathbf{Z}_i)\big) \\
&= \sum_n \sum_{l \in N_n} c_{l,n}\big(\text{argmax}_{\mathbf{W}_o(i)}\, p(\mathbf{W}_o(i)|\mathbf{Z}_i)\big) \qquad (15)
\end{aligned}$$

In Eq. (15) N_n is the neighborhood of node n and its definitions, the set of nodes that are connected to n node (including n node itself); $\{c_{l,n}\}$ is the weighting coefficients (nonnegative real value) for each the neighborhood of nodes n and meet $\sum_{l \in N_n} c_{l,n} = 1$. For all network nodes, $\{c_{l,n}\}$ forms a whole combination matrix $\mathbf{C}$.

Table 1 The DPLMS algorithm summary

```
Initialize: $\{w_{n,0} = 0\}$ for each node $n$ , step size $\mu$, and set nonnegative combination
weights $\{a_{l,n}, c_{l,n}\}$ for each time $i \geq 0$ and each agent $n$, and repeat:
for $i = 1{:}number$
    for each node $n$
    adaptation
        $\sigma^2{}_n(i) = \left[1 - \frac{\alpha_n(i)\|\mathbf{X}_n(i)\|^2}{L}\right]\left[\sigma^2{}_n(i-1) - \sigma^2_{\rho,n}\right]$
        $\alpha_n(i) = \frac{\sigma^2_n(i-1) + \sigma^2_\rho}{\left[\sigma^2_n(i-1) - \sigma^2_{\rho,n}\right]\|\mathbf{X}_n(i)\|^2 + \sigma^2_{\varepsilon,n}}$
        $\boldsymbol{\varphi}_n(i) = \mathbf{W}_n(i-1) + \mu\alpha_n(i)e_n(i)\mathbf{X}_n(i)$
  combination
        $\mathbf{W}_n(i) = \sum_{l \in N_n} a_{l,n}\boldsymbol{\varphi}_l(i)$
    $N_n$ is the neighbor nodes of node $n$ in communication subnet-work.
end for
```

Therefore, inspired by the DLMS algorithm in [3], we get the motivation that we can refer to this process to design the DPLMS algorithm via an adaptation step and combination step.

Adaptation step at node n:

$$\boldsymbol{\varphi}_n(i) = \mathbf{W}_n(i-1) + \mu\alpha_n(i)e_n(i)\mathbf{X}_n(i) \tag{16}$$

Combination step at node n:

$$\mathbf{W}_n(i) = \sum_{l \in N_n} a_{l,n}\boldsymbol{\varphi}_l(i) \tag{17}$$

In Eq. (16), μ denotes the DPLMS algorithm step size, and $\boldsymbol{\varphi}_n(i)$ is the estimated vectors at node n. $\{a_{l,n}\}$ is the nonnegative real weighting diffusion coefficients, and if $l \notin N_n$, $a_{l,n} = 0$. $\{a_{l,n}\}$ forms a whole diffusion coefficients matrix $\mathbf{A}$ and $\mathbf{A}^{\mathrm{T}}\mathbf{1} = 1$. Then, we combine the ATC diffusion strategy and the probabilistic least mean square (PLMS) at all distributed network nodes. For convenience, based on Eqs. (14)–(17), Table 1 is a summary of the DPLMS algorithm. Since the convergence speed and the steady-state error will increase (decrease) as the step size μ increases (decreases), the step size μ should be appropriately set at the beginning of the DPLMS algorithm. Because improperly initialized weights may cause the DPLMS algorithm to slow down, the most common way is used. $\left\{w_{n,0} = 0\right\}$ was set at zero for each network node n. Furthermore, the diffusion method was used in this algorithm, so at the beginning nonnegative combination weights, $\{a_{l,n}, c_{l,n}\}$ were set at the beginning.

3 Performance of the DPLMS Algorithm

Performances of the DPLMS algorithm including mean behavior and computational complexity will be discussed in this subsection. Firstly, let us give some equations,$\widehat{\mathbf{W}}_n(i-1) = \mathbf{W}_o - \mathbf{W}_n(i-1)$, $\hat{\boldsymbol{\varphi}}_n(i-1) = \mathbf{W}_o - \boldsymbol{\varphi}_n(i-1)$,$\mathbf{W}(i) = \mathrm{col}\{\mathbf{W}_1(i), \mathbf{W}_2(i), \ldots, \mathbf{W}_M(i)\}$, $\boldsymbol{\varphi}(i) = \mathrm{col}\big\{\boldsymbol{\varphi}_1(i), \boldsymbol{\varphi}_2(i), \ldots, \boldsymbol{\varphi}_M(i)\big\}$, $\hat{\mathbf{W}}(i) = \mathrm{col}\Big\{\hat{\mathbf{W}}_1(i), \hat{\mathbf{W}}_2(i), \ldots, \hat{\mathbf{W}}_M(i)\Big\}$, $\hat{\boldsymbol{\varphi}}(t) = \mathrm{col}\big\{\hat{\boldsymbol{\varphi}}_1(i), \hat{\boldsymbol{\varphi}}_2(i), \ldots, \hat{\boldsymbol{\varphi}}_M(i)\big\}$.

To facilitate performance analysis, we make the following assumptions:

Assumption 1 Both the additive noise $v_n(i)$ and the regression vector $\mathbf{X}_n(i), \forall k, i$, are spatially and temporally independent. Besides, $v_n(i)$ and $\mathbf{X}_n(i)$ are independent of each other.

Assumption 2 The regression vector $\mathbf{X}_n(i)$ is independent of $\widehat{\mathbf{W}}_k(j)$ for all k and $j < i$.

Although these assumptions are not true in general, they have been widely used in the field of adaptive filtering [8, 20], which can help facilitate performance analysis.

3.1 Mean Performance Analysis

Using the above definitions $\widehat{\mathbf{W}}_n(i-1) = \mathbf{W}_o - \mathbf{W}_n(i-1)$, Eq. (16) can be shown as

$$\hat{\boldsymbol{\varphi}}(i) = \hat{\mathbf{W}}_n(i-1) - \mathbf{S}(i)\Big[\mathbf{R}(i)\hat{\mathbf{W}}_n(i-1) + \mathbf{O}(i)\Big] \tag{18}$$

where $\mathbf{S}(i) = \mathrm{diag}\{\mu\alpha_n(i)\mathbf{I}_M, \ldots, \mu\alpha_n(i)\mathbf{I}_M\}$, $\mathbf{R}(i) = \mathrm{diag}\Big\{\sum_{l=1}^{N} c_{l,1}\mathbf{X}_l^{\mathrm{T}}(i)\mathbf{X}_l(i), \ldots, \sum_{l=1}^{N} c_{l,N}\mathbf{X}_l^{\mathrm{T}}(i)\mathbf{X}_l(i)\Big\}$, $\mathbf{O}(i) = \mathbf{C}^{\mathrm{T}}\mathrm{col}\big\{\mathbf{X}_n^{\mathrm{T}}(i)\varepsilon_n(i), \mathbf{X}_n^{\mathrm{T}}(i)\varepsilon_n(i), \ldots, \mathbf{X}_n^{\mathrm{T}}(i)\varepsilon_n(i)\big\}$, $\mathbf{C} = \mathbf{C} \otimes \boldsymbol{I}_M$, and $\otimes$ denotes the Kronecker product operation.

Also, from combining $\hat{\boldsymbol{\varphi}}_n(i-1) = \mathbf{W}_o - \boldsymbol{\varphi}_n(i-1)$ and Eq. (17), we can get

$$\hat{\mathbf{W}}(i) = \mathbf{A}^{\mathrm{T}}\hat{\boldsymbol{\varphi}}(i) \tag{19}$$

where $\mathbf{A} = \mathbf{A} \otimes \boldsymbol{I}_M$.

So, taking Eq. (18) into Eq. (19), we can get Eq. (20), as

$$\hat{\mathbf{W}}(i) = \mathbf{A}^{\mathrm{T}}[\mathbf{I} - \mathbf{S}(i)\mathbf{R}(i)]\hat{\mathbf{W}}(i-1) + \mathbf{A}^{\mathrm{T}}\mathbf{S}(i)\mathbf{O}(i) \tag{20}$$

Then taking the expectation at both ends of the equal sign of Eq. (20).

$$\begin{aligned} \mathrm{E}\Big[\hat{\mathbf{W}}(i)\Big] &= \mathrm{E}\Big\{\mathbf{A}^{\mathrm{T}}[\mathbf{I} - \mathbf{S}(i)\mathbf{R}(i)]\hat{\mathbf{W}}(i-1) + \mathbf{A}^{\mathrm{T}}\mathbf{S}(i)\mathbf{O}(i)\Big\} \\ &= \mathbf{A}^{\mathrm{T}}[\mathbf{I} - \mathrm{E}\{\mathbf{S}(i)\mathbf{R}(i)\}]\mathrm{E}\Big\{\hat{\mathbf{W}}(i-1)\Big\} + \mathrm{E}\Big\{\mathbf{A}^{\mathrm{T}}\mathbf{S}(i)\mathbf{O}(i)\Big\} \\ &= \mathbf{A}^{\mathrm{T}}[\mathbf{I} - \mathrm{E}\{\mathbf{S}(i)\mathbf{R}(i)\}]\mathrm{E}\Big\{\hat{\mathbf{W}}(i-1)\Big\} \end{aligned} \tag{21}$$

where $\mathrm{E}[\mathbf{S}(i)\mathbf{R}(i)] = \mu \mathrm{diag}\left\{\sum_{l=1}^{N} \alpha_l(i)c_{l,1}\mathbf{R}_{xx,l}(i), \ldots, \sum_{l=1}^{N} \alpha_l(i)c_{l,N}\mathbf{R}_{xx,l}(i)\right\}$. Then, based on Eq. (21), we will get the condition for stability, as Eq. (22).

$$0 < \mu < \frac{2}{\rho_{\max}\left(\sum_{l=1}^{N} \alpha_l(i)c_{l,l}\mathbf{R}_{xx,l}(i)\right)} \tag{22}$$

In Eq. (22), $\rho_{\max}$ is the maximal eigenvalue of $\sum_{l=1}^{N} \alpha_l(i)c_{l,l}\mathbf{R}_{xx,l}(i)$. So, based on Eq. (22), and under that condition, we obtain $\mathrm{E}\left[\hat{\mathbf{W}}(\infty)\right] = 0$, this means that the DPLMS algorithm can achieve accurate estimation, theoretically.

3.2 Mean Square Performance

To characterize the mean square behavior of the DPLMS algorithm, we define the mean square of weight error as weighted by a Hermitian positive-definite matrix Σ that we are free to choose, which is shown as

$$\mathrm{E}\left\|\hat{\mathbf{W}}(i)\right\|_{\Sigma}^{2} = \hat{\mathbf{W}}^{\mathrm{T}}(i)\Sigma\hat{\mathbf{W}}(i) \tag{23}$$

The matrix Σ can be freely chosen so that $\mathrm{E}\left\|\hat{\mathbf{W}}(i)\right\|_{\Sigma}^{2}$ can describe various kinds of mean square behaviors. From Eq. (20) and using Assumptions 1 and 2, if we compute the weighted norm on both sides of the equality and use the fact that $\mathbf{O}(i)$ is independent of $\hat{\mathbf{W}}(i)$ and $\widehat{\mathbf{W}}(i-1)$, we can rewrite (23) more compactly as the following recursive expression:

$$\mathrm{E}\left\|\hat{\mathbf{W}}(i)\right\|_{\Sigma}^{2} = \mathrm{E}\left\|\hat{\mathbf{W}}(i-1)\right\|_{\Sigma'}^{2} + \mathrm{E}\left\{\mathbf{O}^{\mathrm{T}}(i)\mathbf{S}(i)\mathrm{A}\Sigma\mathbf{A}^{\mathrm{T}}\mathbf{S}(i)\mathbf{O}(i)\right\} \tag{24}$$

where $\Sigma' = \mathrm{E}\left\{[\mathbf{I} - \mathbf{R}(i)\mathbf{S}(i)]\mathrm{A}\Sigma\mathbf{A}^{\mathrm{T}}[\mathbf{I} - \mathbf{S}(i)\mathbf{R}(i)]\right\}$.
Moreover, setting

$$\mathbf{G} = \mathrm{E}\left\{\mathbf{O}(i)\mathbf{O}^{\mathrm{T}}(i)\right\} = \boldsymbol{C}^{\boldsymbol{T}}\mathrm{diag}\left\{\boldsymbol{\sigma}_{\boldsymbol{v},1}^{2}\boldsymbol{R}_{\boldsymbol{u},1}, \ldots, \boldsymbol{\sigma}_{\boldsymbol{v},N}^{2}\boldsymbol{R}_{\boldsymbol{u},N}, \right\}\boldsymbol{C} \tag{25}$$

we can rewrite (24) in the form

$$\mathrm{E}\left\|\hat{\mathbf{W}}(i)\right\|_{\Sigma}^{2} = \mathrm{E}\left\|\hat{\mathbf{W}}(i-1)\right\|_{\Sigma'}^{2} + \mathrm{Tr}\left[\Sigma\mathbf{A}^{\mathrm{T}}\mathbf{S}(i)\mathbf{G}\mathbf{S}(i)\mathrm{A}\right] \tag{26}$$

where Tr[ ] denotes the trace operator. Let $\sigma = \mathrm{vec}(\Sigma)$ and $\sigma' = \mathrm{vec}\left(\Sigma'\right)$, where the vec() notation stacks the columns of Σ on top of each other and vec^{-1} () is the inverse operation. We will use interchangeably the notation $\left\|\hat{\mathbf{W}}(i)\right\|_{\sigma}^{2}$ and $\left\|\hat{\mathbf{W}}(i)\right\|_{\Sigma}^{2}$ to denote the same quantity $\hat{\mathbf{W}}^{\mathrm{T}}(i)\Sigma\hat{\mathbf{W}}(i)$. Using the Kronecker product property $\mathrm{vec}(U\Sigma\mathrm{V}) = \left(\mathrm{V}^{T} \otimes U\right)\mathrm{vec}(\Sigma)$, we can vectorize both sides of $\Sigma' = \mathrm{E}\left\{[\mathbf{I}-\mathbf{R}(i)\mathbf{S}(i)]\mathrm{A}\Sigma\mathbf{A}^{\mathrm{T}}[\mathbf{I}-\mathbf{S}(i)\mathbf{R}(i)]\right\}$ and conclude that $\Sigma' = \mathrm{E}\left\{[\mathbf{I}-\mathbf{R}(i)\mathbf{S}(i)]\mathrm{A}\Sigma\mathbf{A}^{\mathrm{T}}[\mathbf{I}-\mathbf{S}(i)\mathbf{R}(i)]\right\}$ can be replaced by the simpler linear vector relation:$\sigma' = \mathrm{vec}\left(\Sigma'\right) = F\sigma$, where F is the following $N^2M^2 \times N^2M^2$ matrix with block entries of size $M^2 \times M^2$ each:

$$F = (I \otimes I)\left\{I - I \otimes (\mathbf{R}(i)\mathbf{S}(i)) - \left(\mathbf{R}^T(i)\mathbf{S}(i)\right) \otimes I + \mathrm{E}\left\{\mathbf{R}^T(i)\mathbf{S}(i)\right\} \otimes \left(\mathbf{R}^T(i)\mathbf{S}(i)\right)\right\}(A \otimes A) \tag{27}$$

Using $\mathrm{Tr}[\Sigma\mathrm{X}] = \mathrm{vec}\left(\mathrm{X}^T\right)^T\sigma$, we can then rewrite Eq. (26) as follows:

$$\mathrm{E}\left\|\hat{\mathbf{W}}(i)\right\|_{\sigma}^{2} = \mathrm{E}\left\|\hat{\mathbf{W}}(i-1)\right\|_{F\sigma}^{2} + \left[\mathrm{vec}\left(\mathrm{A}^T\mathbf{S}(i)G^T\mathbf{S}(i)A\right)\right]^T\sigma \tag{28}$$

Based on Assumptions 1 and 2, then the DPLMS algorithm (21) will be mean square stable if the step sizes are sufficiently small such that (26) is satisfied, and the matrix in (27) is stable.

$$\begin{aligned} F &\approx (I \otimes I)\left\{I - I \otimes (\mathbf{R}(i)\mathbf{S}(i)) - \left(\mathbf{R}^T(i)\mathbf{S}(i)\right) \otimes I + \mathbf{R}^T(i)\mathbf{S}(i) \otimes \mathbf{R}(i)\mathbf{S}(i)\right\}(A \otimes A) \\ &= \left[\mathrm{A}^T(I-\mathbf{S}(i)\mathbf{R}(i))\right]^T\left[\mathrm{A}^T(I-\mathbf{S}(i)\mathbf{R}(i))\right] \end{aligned} \tag{29}$$

Taking the limit as $i \to \infty$ as (assuming the step-sizes are small enough to ensure convergence to a steady-state), we deduce from (28) that:

$$\lim_{i\to\infty}\mathrm{E}\left\|\hat{\mathbf{W}}(i)\right\|_{(I-F)\sigma}^{2} = \left[\mathrm{vec}\left(\mathrm{A}^T\mathbf{S}(i)G^T\mathbf{S}(i)A\right)\right]^T\sigma \tag{30}$$

Equation (30) is a useful result: it allows us to derive several performance metrics through the proper selection of the free weighting parameter σ (or Σ), as was done in [16]. For example, the MSD for any node n is defined as the steady-state value E $\left\|\hat{\mathbf{W}}(i)\right\|^2$, as $i \to \infty$, and can be obtained by computing $\lim_{i\to\infty}\mathrm{E}\left\|\hat{\mathbf{W}}(i)\right\|_{T_n}^2$ with a block weighting matrix T_n that has the $M \times M$ identity matrix at block (n, n) and zeros elsewhere. After, denoting the vectorized version of the matrix T_n by $t_n = \mathrm{vec}$ $(\mathrm{diag}(e_n) \otimes I_M)$, where e_n is the vector whose n-th entry is one and zeros elsewhere, and if we are selecting σ in (30) as $\sigma_n = (I-F)^{-1}t_n$, we arrive at the MSD for node n:

$$\mathrm{MSD}_n = \left[\mathrm{vec}\left(\mathrm{A}^T\mathbf{S}(i)G^T\mathbf{S}(i)A\right)\right]^T(I-F)^{-1}t_n \tag{31}$$

The average network MSD is given by:

$$\mathrm{MSD} = \lim_{i\to\infty}\frac{1}{N}\sum_{n=1}^{N}\mathrm{E}\left\|\hat{\mathbf{W}}_{\boldsymbol{n}}(i)\right\|^2 \tag{32}$$

Then, to obtain the network MSD from Eq. (30), the weighting matrix of $\lim_{i\to\infty}$ E $\left\|\hat{\mathbf{W}}(i)\right\|_T^2$ should be chosen as $\mathrm{T} = \frac{I_{MN}}{N}$. Let q denote the vectorized version of I_{MN}, i.e., $q = \mathrm{vec}(I_{MN})$, and selected σ in (30) as $\sigma = \frac{(I-F)^{-1}q}{N}$, the network MSD is given by:

$$\mathrm{MSD} = \frac{1}{N}\left[\mathrm{vec}\left(\mathrm{A}^T\mathbf{S}(i)G^T\mathbf{S}(i)A\right)\right]^T(I-F)^{-1}q \tag{33}$$

3.3 Computational Complexity

For the adaptive filtering algorithm, the computational complexity previously mentioned is the number of arithmetic operations per iteration for a weight vector or coefficients vector, which includes, the number of multiplications, additions, divisions, and comparisons and so on. Moreover, the time-consuming operation of a multiplication operation is much more than that of addition, so that the multiplication operation takes a major proportion in the computational complexity of an adaptive algorithm. Therefore, the computational complexity is an important parameter, which influences the performance of an adaptive filtering algorithm. Compared to the probability LMS algorithm, there are more multiplications and additions in the adaptive and combined parts. For convenience, the computational complexity of some diffusion algorithms is summarized in Table 2, which includes the DSE-LMS [7], DRVSSLMS [9], DLLAD [4], probability LMS [6], and DPLMS algorithms. From Table 2, compared to the DSE-LMS, DRVSSLMS, and probability LMS [6] algorithms, the DPLMS algorithm is smaller. The computational complexity of DLLAD and DPLMS is the same. The computational complexity of the probability LMS algorithm is the lowest. This is understandable because, in the distributed strategy, each node is the executor of the adaptive algorithm, so that the computational complexity of the distributed algorithm is large.

4 Simulation Results

In this paper, we focus on the distributed adaptive algorithms, and compare the DPLMS algorithm with the DSE-LMS [7], DRVSSLMS [9], and DLLAD [4] algorithms in system identification. So, in this part, to demonstrate the robustness performance of the proposed DPLMS algorithm in the presence of different intensity levels of impulsive interferences and input signal, we conduct simulation experiments with various impulsive interferences and different types of input signals. For this unknown system, we set $M = 16$, and the parameters vector was selected randomly. Each distributed network topology consists of $N = 20$ nodes. For impulse noise, according to [13], we can compute the Bernoulli-Gaussian impulse noise. We set impulse noise as $v(i) = f(i)g(i)$, which is a product of a Bernoulli process $g(i) = \{0, 1\}$ with the probabilities $p(1) = \mathrm{Pr}$, $p(0) = 1 - \mathrm{Pr}$ and a Gaussian process $f(i)$. Besides, we set the impulse noise as spatiotemporally independent. For the adaptation weights in the adaptation step and combination weights in combination step, we apply the uniform

Table 2 The computational complexity of the DSE-LMS, DRVSSLMS, DLLAD, DPLMS, and probability LMS algorithms

Algorithm	Computational cost per iteration				
	Recursion	×	+	| |	sgn(·)
DSE-LMS	Equation 1a in [7]	$(2M+1)N+M$	$(3M-1)N$	0	N
	Equation 1b in [7]	NM	$(N-1)M$	0	0
DRVSSLMS	Equation 11 in [9]	$>(3M+1)N+M$	$(3M-1)N$	0	0
	Equation 11 in [9]	$>(3M+1)N+M$	$(3M-1)N$	0	N
	Equation 12 in [9]	NM	$(N-1)M$	0	0
DLLAD	Equation 16 in [4]	$2MN+M$	$(3M-1)N$	N	0
	Equation 17 in [4]	NM	$(N-1)M$	0	0
Probabilistic LMS	Equation 11 in [6]	$3M+4$	$3M+2$	0	0
DPLMS	Equation 16 in this paper	$2MN+M$	$(3M-1)N$	0	0
	Equation 17 in this paper	NM	$(N-1)M$	0	0

Where "×" denotes Multiplications. ">" denotes larger than. "+" denotes Additions. "| |" denotes Absolute.

rule i.e., $a_{l,n} = c_{l,n} = 1/N_n$. We use the network mean square deviation (MSD) to evaluate the performance of diffusion algorithms, where $\mathrm{MSD}(i) = \frac{1}{N}\sum_{n=1}^{N} \mathrm{E}\left[|\mathbf{W}_o - \mathbf{W}_n(i)|^2\right]$ [8]. In addition, the independent Monte Carlo number is 60 and each run has 4000 iteration numbers.

4.1 Simulation Experiment 1

To illustrate that our algorithm is more robust to the input signal, has a faster convergence rate and a lower steady-state estimation error than the DSE-LMS [7], DRVSSLMS [9] and DLLAD [4] algorithms, the simulation experiment 1 considered the same network topology and the same impulsive interference with the different input signals. If any two nodes in network topology are declared neighbors, with a connect probability greater than or equal to 0.2, the network topology is shown in Fig. 1. The MSD iteration curves for DRVSSLMS (μ equal to 0.6), DSE-LMS (μ equal to 0.6), DNLMS (μ equal to 0.6), and DLLAD (μ equal to 0.6) algorithms in Figs. 2, 3, and 4 are different types of the input signal when the measurement noise in an unknown system is impulse noise with Pr $= 0.4$ and $\sigma_f^2 = 0.2$.

In this experiment, we want to show that the DPLMS algorithm is more robust to the input signal, so we set three sub-experiments with different types of input signals, same impulsive noise and same distributed network topology. From Figs. 2, 3, and 4, we can find that although different types of the input signals are used, the DPLMS algorithm has the faster convergence rate and lowest steady-state error than the DSE-LMS, DRVSSLMS, and DLLAD algorithms. Besides, the DPLMS algorithm is more robust to the input signal. In conclusion, from Simulation experiment 1, we can get

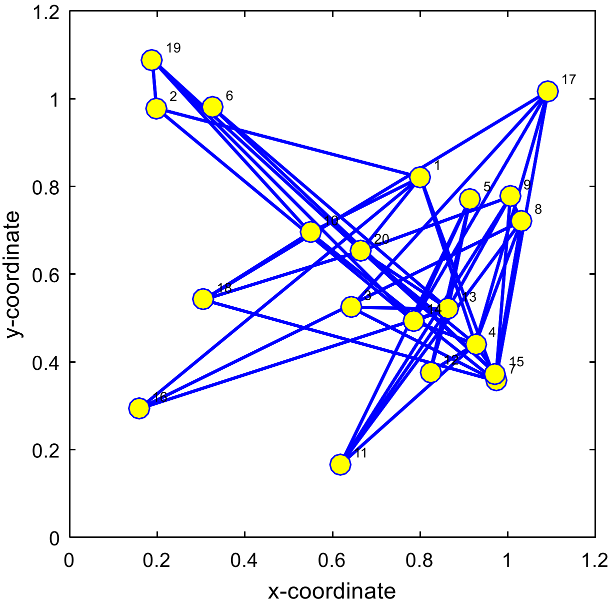

Fig. 1 Random network topology to be decided by probability

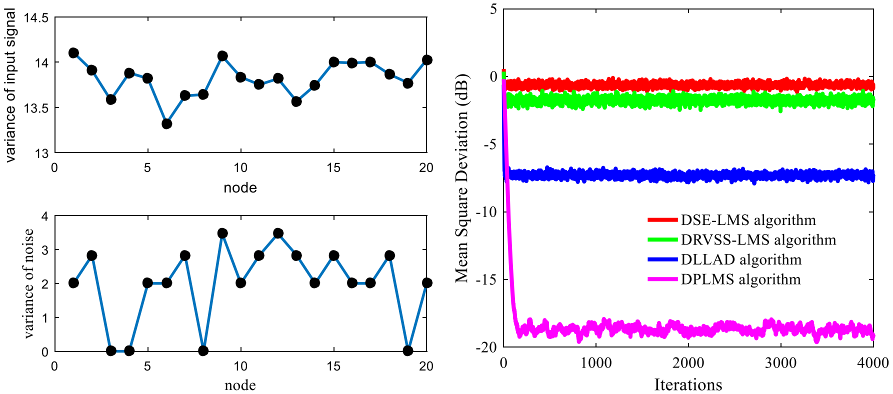

Fig. 2 (Left_top) the input signals $\{\mathbf{X}_n(i)\}$ variances of at each network node with $\mathbf{R}_{xx,n} = \sigma^2_{x,n}\mathbf{I}_M$ with possibly different diagonal entries chosen randomly, (Left_bottom) the measurement noise variances $\{\varepsilon_n(i)\}$ at each network node; (Right)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

that the DPLMS algorithm is superior to the DSE-LMS, DRVSSLMS, and DLLAD algorithms.

4.2 Simulation Experiment 2

To illustrate that our algorithm is more robust to the input signal, has a faster convergence rate, and a lower steady-state estimation error than the DSE-LMS [7],

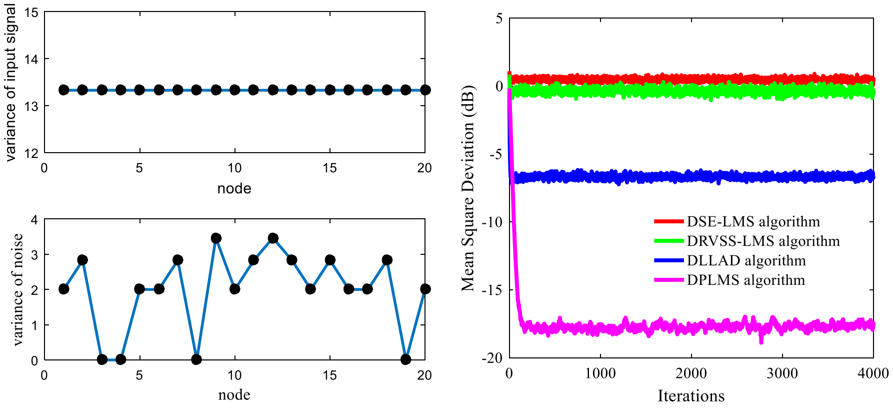

Fig. 3 (Left_top) the input signals $\{\mathbf{X}_n(i)\}$ variances of at each network node with $\mathbf{R}_{xx,n} = \sigma^2_{x,n}\mathbf{I}_M$ with the same value in each diagonal entries, (Left_bottom) the measurement noise variances $\{\varepsilon_n(i)\}$ at each network node; (Right) Transient network MSD (dB) iteration curves of the DSE-LMS, DRVSSLMS, DLLAD, and DPLMS algorithms

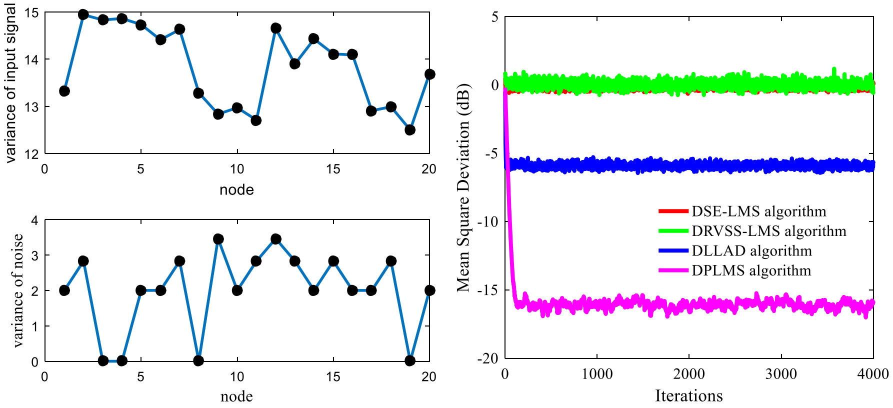

Fig. 4 (Left_top) the input signals $\{\mathbf{X}_n(i)\}$ variances of at each network node with $\mathbf{R}_{xx,n} = \sigma^2_{x,n}(t)\mathbf{I}_M, t = 1, 2, 3, \ldots, M$ with a different values in each diagonal entries, (Left_bottom) the measurement noise variances $\{\varepsilon_n(i)\}$ at each network node; (Right) Transient network MSD (dB) iteration curves of the DSE-LMS, DRVSSLMS, DLLAD, and DPLMS algorithms

DRVSSLMS [9], and DLLAD [4] algorithms, the simulation experiment 2 considered the same network topology and the same input signal with the different probability densities of impulsive interference. If any two nodes in network topology are declared neighbors, with a certain radius for each node large than or equal to 0.3, the network topology is shown in Fig. 5 (Left). The MSD iteration curves for DRVSSLMS (μ equal to 0.4), DSE-LMS (μ equal to 0.4), DNLMS (μ equal to 0.4), and DLLAD (μ equal to 0.4) algorithms were shown in Fig. 6 with $\text{Pr} = 0.1$, $\text{Pr} = 0.4$, and $\text{Pr} = 0.7$ under same $\sigma_f^2 = 0.2$, respectively.

In this experiment, we aim to show that the DPLMS algorithm is more robust to different the probability densities of impulsive interference. So we set three sub-

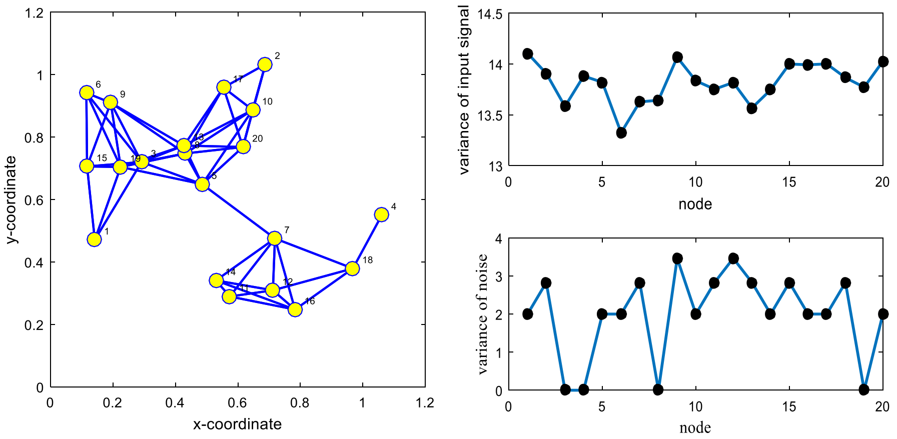

Fig. 5 (Left) Random network topology to be decided by a certain radius; (Right_top) the input signals $\{\mathbf{X}_n(i)\}$ variances of at each network node with $\mathbf{R}_{xx,n} = \sigma_{x,n}^2 \mathbf{I}_M$ with possibly different diagonal entries chosen randomly, (Right_bottom) the measurement noise variances $\{\varepsilon_n(t)\}$ at each network node

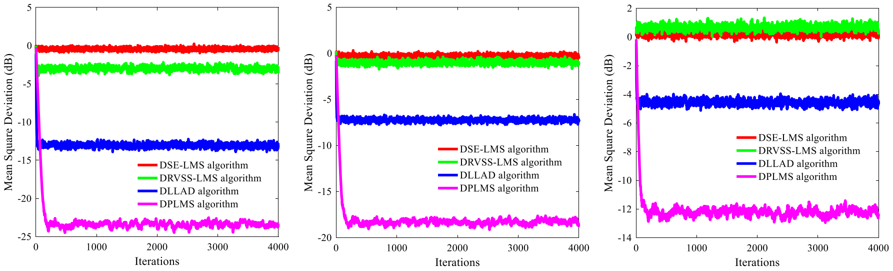

Fig. 6 Transient network MSD (dB) iteration curves of the DSE-LMS, DRVSSLMS, DLLAD, and DPLMS algorithms. (Left) with Pr = 0.1, (middle) with Pr = 0.4 and (Right) with Pr = 0.7

experiments with different probability densities of impulsive interference, the same input signal (correlation coefficient is 0.7) and same distributed network topology. From Fig. 6, we can find that although different probability densities of impulsive interference is considered, the DPLMS algorithm is slightly faster than that of the DSE-LMS, DRVSSLMS, and DLLAD algorithms, and the DPLMS algorithm still has a smaller steady-state error than the DSE-LMS, DRVSSLMS, and DLLAD algorithms. In other words, from Simulation experiment 2, we can observe that the DPLMS algorithm is more robust to impulsive interference than the DSE-LMS, DRVSSLMS, and DLLAD algorithms.

4.3 Simulation Experiment 3

To illustrate that our algorithm is more robust to the input signal, has a faster convergence rate and a lower steady-state estimation error than the DSE-LMS [7], DRVSSLMS [9], and DLLAD [4] algorithms, the simulation experiment 3 considered the same network topology and the same input signal with the different probability

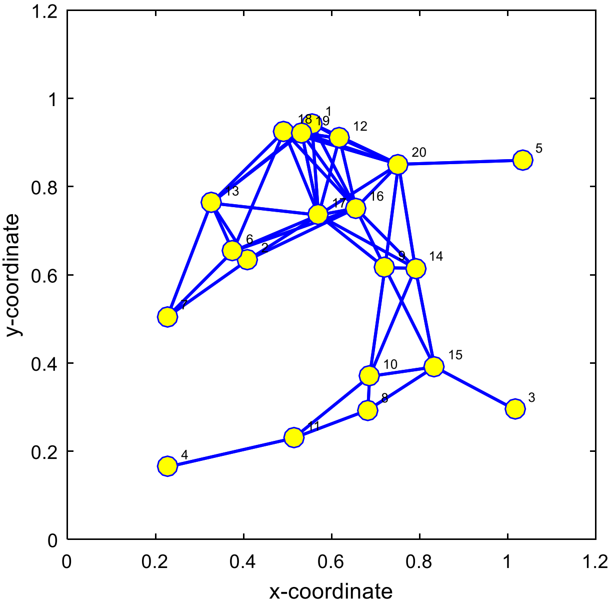

Fig. 7 Random network topology to be decided by a certain radius

densities and variance of impulsive interference. If any two nodes in network topology are declared neighbors, with a certain radius for each node large than or equal to 0.3, the network topology is shown in Fig. 7. The MSD iteration curves for DRVSSLMS (μ equal to 0.4), DSE-LMS (μ equal to 0.4), DNLMS (μ equal to 0.4), and DLLAD (μ equal to 0.4) algorithms in Fig. 8 (Left-top, $\mathrm{Pr} = 0.7$, $\sigma_f^2 = 0.2$; Right-top, $\mathrm{Pr} = 0.7$, $\sigma_f^2 = 0.4$; Left-bottom, $\mathrm{Pr} = 0.4$, $\sigma_f^2 = 0.4$; Right-down, $\mathrm{Pr} = 0.4$, $\sigma_f^2 = 0.6$, respectively.

Comparing Fig. 8 (Left_top) with Fig. 8 (Right_top), the DPLMS algorithm can perform better in identifying the unknown coefficients under different impulsive interference intensities. From Fig. 8 (Left_bottom) with Fig. 8 (Right_bottom), the same conclusion can be obtained. However, for the DSE-LMS, DRVSSLMS, and DLLAD algorithms, the identification performance is easily interfered with by different characteristics that is to say that the DPLMS algorithm is more robust. The DPLMS algorithm is superior to the DSE-LMS, DRVSSLMS, and DLLAD algorithms. So, for distributed network estimation in impulsive interference environments, from Simulation experiment 1–Simulation experiment 3 (total ten sub-experiments), we know that the DPLMS algorithm has a superior performance when estimating an unknown linear system under the changeable impulsive noise environments.

5 Conclusion

In this paper, we propose a novel DPLMS algorithm, which is a distributed algorithm robust to various input signal and impulsive interference environments. The method

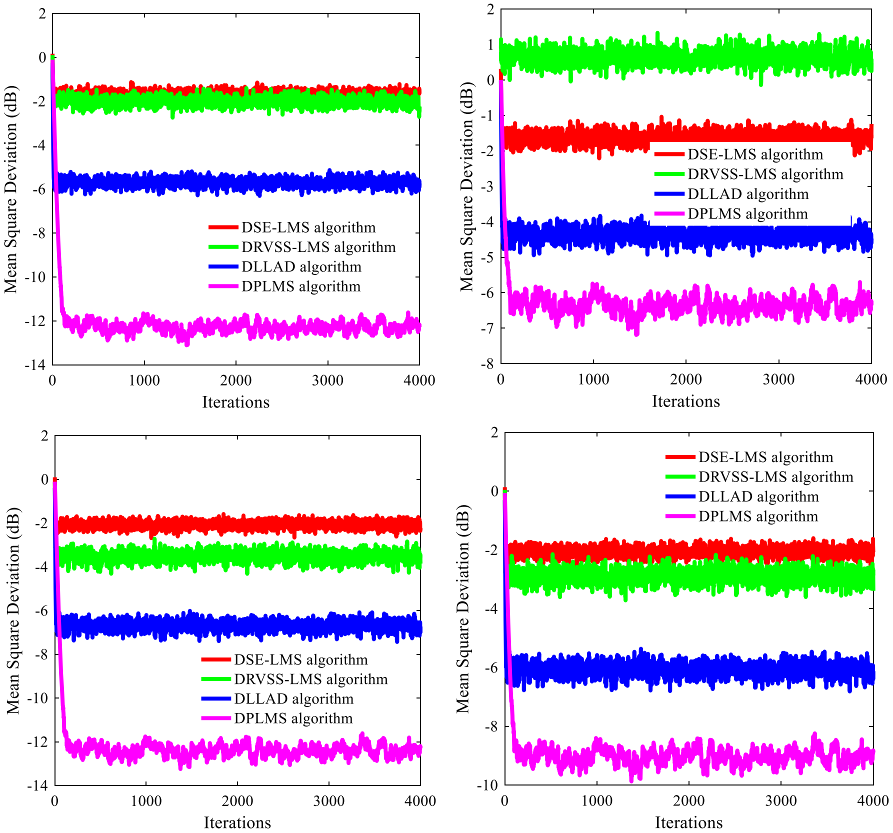

Fig. 8 Transient network MSD (dB) iteration curves of the DSE-LMS, DRVSSLMS, DLLAD, and DPLMS algorithms with $\mathbf{R}_{xx,n} = \sigma^2_{x,n}(t)\mathbf{I}_M, t = 1, 2, 3, \ldots, M$ with different values in each diagonal entries. (Left-top): $\mathrm{Pr} = 0.7$, $\sigma^2_f = 0.2$; (Right-top): $\mathrm{Pr} = 0.7$, $\sigma^2_f = 0.4$; (Left-bottom): $\mathrm{Pr} = 0.4$, $\sigma^2_f = 0.4$; (Right-down): $\mathrm{Pr} = 0.4$, $\sigma^2_f = 0.6$

is developed based on the combination and modification of the DLMS algorithm and the probabilistic LMS algorithm at all nodes in a distributed network. The theoretical analysis demonstrates that the DPLMS algorithm can achieve an effective estimation from a probabilistic perspective. It is shown that the computational complexity of the DPLMS algorithm is smaller than that of the DRVSSLMS and DSE-LMS algorithms, despite being equal to that of the DLLAD algorithm. Besides, theoretical mean behavior interpretes that the DPLMS algorithm can achieve accurate estimation, and simulation results show that the DPLMS algorithm is more robust to the input signal and impulsive interference than the DSELMS, DRVSSLMS, and DLLAD algorithms. In short, the DPLMS algorithm has a superior performance when estimating an unknown linear system under changeable impulsive noise environments, which will have a significant impact on real-world applications. Additionally, we wish to bring more probability theory-based techniques to distributed adaptive filtering algorithms. Although the DPLMS algorithm has superior performance compared to the DSE-

LMS, DRVSSLMS, and DLLAD algorithms, the environment in actual engineering applications is complex and time-varying, which means DPLMS algorithm needs to be adjusted accordingly for different application scenarios. First, the delay caused by the communication of different nodes needs to be considered [30], because the delay will cause asynchronization. Second, we need to consider whether the parameters to be evaluated are sparse (such as brain networks) [5, 19]. In such cases, it will be better to add a regularized constraint terms (L1-norm) to this algorithm. Lastly, the external environment factors such as the temperature where the system is located should also be considered, because earlier study reported that high temperature could make the system unstable and time-varying [23]. In such cases, it is best to increase the parameters of the external environment in this algorithm.

Acknowledgments This work was partially supported by the National Natural Science Foundation of China (Grant: 61871420).

References

1. S. Ashkezari-Toussi, H. Sadoghi-Yazdi, Robust diffusion LMS over adaptive networks. Signal Process. **158**, 201–209 (2019)
2. F.S. Cattivelli, C.G. Lopes, A.H. Sayed, Diffusion recursive least-squares for distributed estimation over adaptive networks. IEEE Trans. Signal Process. **56**(5), 1865–1877 (2008)
3. F.S. Cattivelli, A.H. Sayed, Diffusion LMS strategies for distributed estimation. IEEE Trans. Signal Process. **58**(3), 1035–1048 (2010)
4. F. Chen, T. Shi et al., Diffusion least logarithmic absolute difference algorithm for distributed estimation. Signal Process. **142**, 423–430 (2018)
5. H. Eavani, T.D. Satterthwaite et al., Identifying sparse connectivity patterns in the brain using resting-state fMRI. NeuroImage **105**, 286–299 (2015)
6. J. Fernández-Bes, V. Elvira, S. Van Vaerenbergh, A probabilistic least mean squares filter, in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brisbane, QLD, 2015, pp. 2199–2203
7. Y. Gao, J. Ni, J. Chen et al., Steady-state and stability analyses of diffusion sign-error LMS algorithm. Signal Process. **149**, 62–67 (2018)
8. S. Haykin, *Adaptive Filter Theory* (Prentice-Hall, Englewood Cliffs, 2001)
9. W. Huang, L. Li, Q. Li et al., Diffusion robust variable step-size LMS algorithm over distributed networks. IEEE Access **6**, 47511–47520 (2018)
10. F. Huang, J. Zhang, S. Zhang, Mean-square-deviation analysis of probabilistic LMS algorithm. Digit. Signal Process. **92**, 26–35 (2019)
11. C. Jie, C. Richard, A.H. Sayed, Diffusion LMS over multitask networks. IEEE Trans. Signal Process. **63**(11), 2733–2748 (2015)
12. H.S. Lee, S.H. Yim, W.J. Song, z2-proportionate diffusion LMS algorithm with mean square performance analysis. Signal Process. **131**, 154–160 (2017)
13. Z. Li, G. Sihai, Diffusion normalized Huber adaptive filtering algorithm. J. Frank. Inst. **355**(8), 3812–3825 (2018)
14. Y. Liu, C. Li, Z. Zhang, Diffusion sparse least-mean squares over networks. IEEE Trans. Signal Process. **60**(8), 4480–4485 (2012)
15. C.G. Lopes, A.H. Sayed, Diffusion least-mean squares over adaptive networks: formulation and performance analysis. IEEE Trans. Signal Process. **56**(7), 3122–3136 (2008)
16. P.D. Lorenzo, A.H. Sayed, Sparse distributed learning based on diffusion adaptation. IEEE Trans. Signal Process. **61**(6), 1419–1433 (2013)
17. J. Ni, Diffusion sign subband adaptive filtering algorithm for distributed estimation. IEEE Signal Process. Lett. **22**(11), 2029–2033 (2015)
18. J. Ni, J. Chen, X. Chen, Diffusion sign-error LMS algorithm: formulation and stochastic behavior analysis. Signal Process. **128**, 142–149 (2016)

19. Y. Renping, L. Qiao, M. Chen et al., Weighted graph regularized sparse brain network construction for MCI identification. Pattern Recognit. **90**, 220–231 (2019)
20. A.H. Sayed, *Fundamentals of Adaptive Filtering* (Wiley, New York, 2003)
21. A.H. Sayed, Adaptive networks. Proc. IEEE **102**(4), 460–497 (2014)
22. M. Shao, C.L. Nikias, Signal processing with fractional lower order moments: stable processes and their applications. Proc. IEEE **81**(7), 986–1010 (1993)
23. K. Smith, L. Spyrou, J. Escudero, Graph-variate signal analysis. IEEE Trans. Signal Process. **67**(2), 293–305 (2019)
24. J. Song, Y. Xu, Y. Liu et al., Investigation on estimator of chirp rate and initial frequency of LFM signals based on modified discrete chirp Fourier transform. Circuits Syst. Signal Process. **38**, 5861–5882 (2019)
25. V. Stojanovic, V. Filipovic, Adaptive input design for identification of output error model with constrained output. Circuits Syst. Signal Process. **33**(1), 97–113 (2014)
26. V. Stojanovic, N. Nedic, Joint state and parameter robust estimation of stochastic nonlinear systems. Int. J. Robust Nonlinear Control **26**(14), 3058–3074 (2015)
27. V. Stojanovic, N. Nedic, Robust Kalman filtering for nonlinear multivariable stochastic systems in the presence of non-Gaussian noise. Int. J. Robust Nonlinear Control **26**(3), 445–460 (2015)
28. V. Stojanovic, N. Nedic, Robust identification of OE model with constrained output using optimal input design. J. Frank. Inst. **353**(2), 576–593 (2016)
29. S.Y. Tu, A.H. Sayed, Diffusion strategies outperform consensus strategies for distributed estimation over adaptive networks. IEEE Trans. Signal Process. **60**(12), 6217–6234 (2012)
30. E.T. Wagner, M.I. Doroslovački, Distributed LMS estimation of scaled and delayed impulse responses, in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Shanghai, 2016, pp. 4154–4158
31. F. Wen, Diffusion least-mean *p*-power algorithms for distributed estimation in alpha-stable noise environments. Electron. Lett. **49**(21), 1355–1356 (2013)
32. P. Wen, J. Zhang, Variable step-size diffusion normalized sign-error algorithm. Circuits Syst. Signal Process. **37**(20), 4993–5004 (2018)
33. X.J. Yang, Fractional derivatives of constant and variable orders applied to anomalous relaxation models in heat-transfer problems. Therm. Sci. **21**(3), 1161–1171 (2016)
34. X.J. Yang, A new integral transform operator for solving the heat-diffusion problem. Appl. Math. Lett. **64**, 193–197 (2017)
35. X.J. Yang, Y.Y. Feng, C. Cattani et al., Fundamental solutions of anomalous diffusion equations with the decay exponential kernel. Math. Methods Appl. Sci. **42**(11), 4054–4060 (2019)
36. X.J. Yang, F. Gao, Y. Ju et al., Fundamental solutions of the general fractional-order diffusion equations. Math. Methods Appl. Sci. **41**(18), 9312–9320 (2018)
37. X.-J. Yang, M.J.A. Tenreiro, A new fractional operator of variable order: application in the description of anomalous diffusion. Phys. A Stat. Mech. Appl. **481**, 276–283 (2017)
38. S. Zhang, H.C. So, W. Mi et al., A family of adaptive decorrelation NLMS algorithms and its diffusion version over adaptive networks. IEEE Trans. Circ. Syst. I Regul. Pap. **99**, 1–12 (2017)

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