



# Robust Diffusion Huber-Based Normalized Least Mean Square Algorithm with Adjustable Thresholds

Yi Yu<sup>1</sup> · Haiquan Zhao<sup>2</sup> · Wenyuan Wang<sup>2</sup> · Lu Lu<sup>3</sup>

Received: 31 October 2018 / Revised: 20 August 2019 / Accepted: 21 August 2019 /

Published online: 27 August 2019

© Springer Science+Business Media, LLC, part of Springer Nature 2019

## Abstract

To improve the performance of the diffusion Huber-based normalized least mean square algorithm in the presence of impulsive noise, this paper proposes a distributed recursion scheme to adjust the thresholds. Because of the decreasing characteristic of the thresholds, the proposed algorithm can also be interpreted as a robust diffusion normalized least mean square algorithm with variable step sizes so that it has not only fast convergence but also small steady-state estimation error. Based on the contaminated Gaussian model, we analyze the mean square behavior of the algorithm in impulsive noise. Moreover, to ensure good tracking capability of the algorithm for the sudden change of parameters of interest, a control strategy is given that resets the thresholds with their initial values. Simulations in various noise scenarios show that the proposed algorithm performs better than many existing diffusion algorithms.

**Keywords** Adjustable thresholds · Diffusion algorithms · Impulsive noise · Nonstationary control method

---

✉ Yi Yu  
yuyi\_xyuan@163.com

✉ Haiquan Zhao  
hqzhao\_swjtu@126.com

Wenyuan Wang  
wenyuanwang@my.swjtu.edu.cn

Lu Lu  
lulu19900303@126.com

<sup>1</sup> School of Information Engineering, Robot Technology Used for Special Environment Key Laboratory of Sichuan Province, Southwest University of Science and Technology, Mianyang 621010, China

<sup>2</sup> School of Electrical Engineering, Southwest Jiaotong University, Chengdu, China

<sup>3</sup> School of Electronics and Information Engineering, Sichuan University, Chengdu, China

## 1 Introduction

Distributed adaptive technique is an emerging and promising topic to estimate parameters of interest from the streaming data collected at nodes (or agents) of wireless sensor networks [42,43]. In distributed adaptive algorithms, each node is allowed to estimate parameters of interest in an adaptive loop and to share its local estimate with the neighboring nodes each other, obtaining better estimation performance than the no cooperation way among nodes. In contrast with the centralized processing over networks, on the other hand, the distributed processing does not require a powerful central processor, thus being more flexible and saving communication bandwidth resources. Due to these advantages, distributed adaptive algorithms have been used in a variety of applications such as the frequency estimation in power grids [22,26], the spectrum estimation in cognitive radio networks [17,21,34], the source localization over unmanned aerial vehicle networks [2,55], and the compressed estimation [50].

According to different cooperation strategies between interconnected nodes, existing distributed adaptive algorithms can be categorized as incremental [27,51], consensus [23,46], and diffusion [1,8,13,25,30,56] types. For the incremental type, a Hamiltonian cycle path [27] that runs across all the nodes must be determined and retained until the final estimate is achieved. The incremental type has the lowest communication burden among these three types. However, maintaining such a path is very difficult especially in large-scale networks, since it does not permit any failure of nodes and links. Both consensus and diffusion types have no Hamiltonian cycle, due to the fact that interconnected nodes exchange information each other. The consensus strategy enforces a property to the network that all the nodes obtain the same estimate. Nonetheless, studies in [47] have revealed that the stability of the diffusion algorithm does not rely on the network topology so that it is more stable than the consensus algorithm. Also, the diffusion algorithm exhibits better performance than the consensus algorithm in terms of convergence rate and steady-state estimation error (i.e., *estimation accuracy*). Comparatively speaking, therefore, the diffusion distributed strategy has got more attention. The diffusion least mean square (dLMS) algorithm is one of the most popular distributed algorithms, due to its simplicity and stable performance in Gaussian noise environment [8]. However, the range of step sizes ensuring the dLMS convergence is dependent of the covariance matrix of input signal. Benefited from the advantage of the normalized LMS (NLMS) to weaken such dependence, the diffusion NLMS (dNLMS) algorithm was proposed [20]. To overcome the trade-off between fast convergence rate and low steady-state estimation error for the dLMS and dNLMS algorithms, many variable step-size (VSS) variants have been promoted, see [20,24] and references therein.

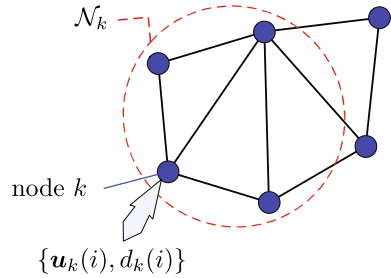
In signal processing area, Gaussian noise is concerned frequently by researchers. However, the noise may also be non-Gaussian, which includes not only Gaussian noise but also impulsive noise [6,18,33]. Although realizations of impulsive noise are sparse and random, the amplitude is far higher than that of Gaussian noise; in this case, the noise is best modeled by heavy-tailed distributions, e.g., the contaminated Gaussian (CG) and the  $\alpha$ -stable distributions. Such noise scenarios exist in many applications such as echo cancelation, underwater acoustics, audio processing, communications, cognitive radio, and extreme learning [6,9,10,18,33,58,59]. In distributed estimation,

even if impulsive noise appears only at one node in the network, its adverse influence could be propagated over the entire network due to information exchange among nodes. That is to say, the dLMS/dNLMS algorithms suffer from performance degradation in impulsive noise, owing to the squared error-based minimization criteria. In standalone adaptive filter, several approaches have been presented to deal with impulsive interferences [11,12,16,44,57]. At present, some of these approaches have been used in distributed networks to develop robust distributed adaptive algorithms in combating impulsive noise.

In [49], the diffusion least mean  $p$ th power (dLMP) algorithm by minimizing the cost function consisting of the  $p$ -power of the output errors at nodes was proposed, where the parameter  $p$ ,  $1 \leq p < 2$ , controls the robustness of dLMP against impulsive noise. As a special case of the dLMP with  $p = 1$ , the diffusion sign error LMS (dSE-LMS) algorithm [37] provides good robustness in resisting impulsive noise, while the disadvantage is slow convergence. Based on the weighted combination of the preselected sign-preserving basis functions, the diffusion error nonlinearity LMS (dEN-LMS) algorithm was proposed [40]; however, the basis functions need to be chosen properly according to noise distributions. In [31], the diffusion maximum correntropy criterion (dMCC) algorithm was presented by maximizing the cost function defined as the correntropy of error signals. This algorithm is able to suppress impulsive noise, but this capability relies on the proper choice of the kernel width parameter. By detecting whether impulsive noise occurs or not, reference [3] proposed a robust variable weighting coefficients dLMS (RVWC-dLMS) algorithm, which rejects the cooperation with the nodes at which impulsive noise occurs. Due to the robustness of Huber function against outliers, in [14], it has been used to devise a robust diffusion projection algorithm by resorting to the adaptive projected subgradient rule; and in [29], the diffusion Huber-based NLMS (dHNLMS) algorithm was proposed by resorting to the gradient-descent (GD) rule. However, setting the threshold in the Huber function requires the prior knowledge of the noise distribution. In [5], by minimizing the pseudo-Huber-based cost function over nodes in the network, a robust dLMS algorithm was developed, while its performance needs to choose properly the threshold-like parameter. It is worth noting that the above robust diffusion algorithms involve the choice of step sizes which leads to a performance balance problem between convergence rate and steady-state estimation error. For the sake of addressing this problem, a diffusion robust VSS LMS (dRVSS-LMS) algorithm was developed based on the Huber function [19], but its implementation requires the prior knowledge of background and impulsive noises.

In this paper, we propose a variable threshold (VT) dHNLMS algorithm, referred to as VT-dHNLMS. The thresholds are recursively adjusted with the diffusion cooperation of neighboring nodes and with the diminishing property, which ensures good robustness against impulsive noise. Furthermore, this algorithm can be considered as a variable step-size dNLMS algorithm, with fast convergence and low steady-state estimation error simultaneously. We provide the convergence and steady-state analyses for the algorithm in the presence of impulsive noise. To recover the tracking capability of the VT-dHNLMS algorithm when parameters of interest experience a sudden change, a diffusion cooperation-based nonstationary control (NC) method is devised that resets the thresholds. Although we have discussed similar idea for adjusting the

**Fig. 1** Schematic diagram of diffusion network



thresholds in the incremental distributed NLMS algorithm [51], here the extension to the diffusion distributed algorithm is not straightforward. Similar approach has also been studied by us to improve the performance of diffusion recursive least squares (dRLS) algorithm in impulsive noise [54], while the dRLS algorithm requires high computational complexity. Thus, it is necessary to further generalize the VT approach to improve the dHNLMS performance, with low-complexity.

This paper is organized as follows. In Sect. 2, the dHNLMS algorithm is introduced. Then, in Sect. 3, the VT-dHNLMS algorithm is proposed. The convergence and steady-state behaviors of the proposed algorithm are analyzed in Sect. 4. In Sect. 5, simulation results in various noise scenarios are presented to verify the effectiveness of the proposed algorithm. Finally, conclusions are given in Sect. 6.

The notations adopted in this paper include  $(\cdot)^T$  for the transpose,  $E\{\cdot\}$  for the expectation,  $\|\cdot\|_2$  for the  $l_2$ -norm of a vector,  $\|\cdot\|_\infty$  for the  $\infty$ -norm,  $\text{col}\{\cdot\}$  for a column vector formed by stacking its arguments on top of each other,  $\text{diag}\{\cdot\}$  for the (block) diagonal operation,  $\text{Tr}(\cdot)$  for the trace of a matrix, and  $\otimes$  for the Kronecker product. Also, we denote the identity matrix of size  $M \times M$  as  $\mathbf{I}_M$ .

## 2 dHNLMS Algorithm

Consider a network that consists of  $N$  nodes distributed in a geographic region as shown in Fig. 1, where  $\mathcal{N}_k$  is the set of nodes with a direct link to node  $k$  (including itself) and known as the neighborhood of node  $k$ . The links among nodes indicate that these nodes can communicate with each other. At every time instant  $i$ , every node  $k$  has access to an  $M \times 1$  input regressor  $\mathbf{u}_k(i)$  and a scalar measurement  $d_k(i)$  following the linear model:

$$d_k(i) = \mathbf{u}_k^T(i) \mathbf{w}^o + v_k(i), \quad (1)$$

where all the nodes in the network are to estimate the  $M \times 1$  vector  $\mathbf{w}^o$ . The model (1) has been widely used in applications [41,42].

For estimating  $\mathbf{w}^o$ , the well-known mean square error (MSE) criteria over networks are defined as

$$\mathbf{w} = \min_{\mathbf{w}} \sum_{k=1}^N E \left\{ (d_k(i) - \mathbf{u}_k^T(i) \mathbf{w})^2 \right\}. \quad (2)$$

In the light of the diffusion cooperation of interconnected nodes, (2) equivalently becomes that every node  $k$  solves the following estimation problem:

$$\begin{aligned} \mathbf{w}_k &= \min_{\mathbf{w}_k} J_k^{\text{loc}}(\mathbf{w}_k), \\ J_k^{\text{loc}}(\mathbf{w}_k) &= \sum_{m \in \mathcal{N}_k} c_{m,k} E \left\{ (d_m(i) - \mathbf{u}_m^T(i) \mathbf{w}_k)^2 \right\}, \end{aligned} \quad (3)$$

for  $k = 1, \dots, N$ . The combination coefficients  $\{c_{m,k}\}$  in (3) are nonnegative and satisfy:

$$\sum_{m \in \mathcal{N}_k} c_{m,k} = 1, \text{ and } c_{m,k} = 0 \text{ if } m \notin \mathcal{N}_k, \quad (4)$$

where  $c_{m,k}$  accounts for the weight that node  $k$  assigns to the data received from its neighboring node  $m$ . In general,  $\{c_{m,k}\}$  are determined by either static rules (e.g., the Metropolis rule [45]) which keep them constant, or adaptive rules [35]. From (3), both dLMS and dNLMS algorithms can be derived. Note that, the expectation term in (3) is normalized by input regressors for the dNLMS algorithm, which show good performance in Gaussian additive noise. In many practical situations, however, the additive noise  $v_k(i)$  at node  $k$  may compose of the Gaussian component and the impulsive component with low occurrence probability. In this scenario, the data  $\{\mathbf{u}_k(i), d_k(i)\}$  have more outliers so that the MSE criteria are not favorable. Therefore, in order to seek a good estimate of  $\mathbf{w}^o$  in impulsive noise, the dHNLMS algorithm solves the Huber function-based estimation problem:

$$\begin{aligned} \mathbf{w}_k &= \min_{\mathbf{w}_k} J_k^{\text{loc}}(\mathbf{w}_k), \\ J_k^{\text{loc}}(\mathbf{w}_k) &= \sum_{m \in \mathcal{N}_k} c_{m,k} E \left\{ \varphi \left( \frac{d_m(i) - \mathbf{u}_m^T(i) \mathbf{w}_k}{\|\mathbf{u}_m(i)\|_2} \right) \right\}, \end{aligned} \quad (5)$$

where the Huber function  $\varphi(\cdot)$  is given by [38]

$$\varphi(r_m) = \begin{cases} r_m^2/2, & \text{if } |r_m| \leq \sqrt{\xi} \\ \sqrt{\xi}|r_m| - \xi/2, & \text{if } |r_m| > \sqrt{\xi}, \end{cases} \quad (6)$$

and  $\xi > 0$  denotes the threshold. As can be seen in (6), the Huber function can be interpreted as a piecewise combination of the MSE criteria (when  $|r_m| \leq \sqrt{\xi}$ ) and the sign criteria (when  $|r_m| > \sqrt{\xi}$ ). The sign criteria are a popular benchmark in resisting impulsive noise [32,36], which has prompted some robust algorithms such as the dSE-LMS.

By applying the instantaneous GD principle into (5) and following the Adapt-then-Combine (ATC) version of diffusion cooperation strategy among nodes, the update equations of the ATC dHNLMS algorithm are described as [29]:

$$\boldsymbol{\psi}_k(i+1) = \mathbf{w}_k(i) + \mu \begin{cases} \frac{e_k(i)\mathbf{u}_k(i)}{\|\mathbf{u}_k(i)\|_2^2}, & \text{if } |e_k(i)| \leq \sqrt{\xi} \\ \sqrt{\xi} \frac{\text{sign}(e_k(i))\mathbf{u}_k(i)}{\|\mathbf{u}_k(i)\|_2}, & \text{if } |e_k(i)| > \sqrt{\xi}, \end{cases} \quad (7)$$

$$\mathbf{w}_k(i+1) = \sum_{m \in \mathcal{N}_k} c_{m,k} \boldsymbol{\psi}_m(i+1), \quad (8)$$

where

$$e_k(i) = d_k(i) - \mathbf{w}_k^T(i)\mathbf{u}_k(i) \quad (9)$$

is the output error of node  $k$ ,  $\text{sign}(x)$  is the signum function of variable  $x$ , and  $\mu > 0$  is the constant step size.

It is clear that the ATC dHNLMS algorithm consists of an adaptation step (7) and a combination step (8). In the adaptation step, each node  $k$  updates the estimate from  $\mathbf{w}_k(i)$  to an intermediate estimate  $\boldsymbol{\psi}_k(i+1)$  using its own data  $\{d_k(i), \mathbf{u}_k(i)\}$ . In the combination step, the intermediate estimates  $\{\boldsymbol{\psi}_m(i+1)\}$  from the neighborhood of node  $k$  are linearly fused based on the combination coefficients  $\{c_{m,k}\}$ , thereby yielding a more reliable estimate  $\mathbf{w}_k(i+1)$ . Note that, the Combine-then-Adapt (CTA) version of the dHNLMS algorithm can also be obtained in a straightforward way, namely by reversing the orders of (7) and (8). However, we only discuss the ATC diffusion implementation for brevity, since it generally outperforms the CTA version [8,43]. Hereafter, the notation ATC is omitted.

### 3 Proposed Algorithm

In this section, we develop the VT-dHNLMS algorithm and analyze the computational complexity.

#### 3.1 Algorithm Design

The adverse effect of impulsive noise on the dHNLMS performance occurs in the adaptation step (7). Specifically, the robustness of the algorithm against impulsive noise is determined by the preselected threshold  $\xi$ . To remedy this flaw, we propose to replace  $\xi$  with the time-varying threshold  $\xi_k(i)$ , thus (7) becomes

$$\boldsymbol{\psi}_k(i+1) = \mathbf{w}_k(i) + \mu \begin{cases} \frac{e_k(i)\mathbf{u}_k(i)}{\|\mathbf{u}_k(i)\|_2^2}, & \text{if } |e_k(i)| \leq \sqrt{\xi_k(i)} \\ \sqrt{\xi_k(i)} \frac{\text{sign}(e_k(i))\mathbf{u}_k(i)}{\|\mathbf{u}_k(i)\|_2}, & \text{if } |e_k(i)| > \sqrt{\xi_k(i)}. \end{cases} \quad (10)$$

From (10), we are able to obtain the following relation:

$$\begin{aligned} \|\boldsymbol{\psi}_k(i+1) - \mathbf{w}_k(i)\|_2^2 &= \mu^2 \min \left[ \xi_k(i), \frac{e_k^2(i)}{\|\mathbf{u}_k(i)\|_2^2} \right] \\ &\stackrel{(a)}{\leq} \xi_k(i), \end{aligned} \tag{11}$$

where  $0 < \mu \leq 1$  is for satisfying the inequality (a). It means that for every node  $k$ , the updated distance from  $\mathbf{w}_k(i)$  to  $\boldsymbol{\psi}_k(i)$  in the adaptation (10) is limited to at most the amount of the threshold  $\xi_k(i)$  for any disturbance (including impulsive noise). Therefore,  $\xi_k(i)$  controls not only the robust level of the algorithm to suppress impulsive noise but also its convergence behavior. Generally, it is desired to have relatively large values of  $\xi_k(i)$  at the beginning of the adaptation, thereby leading to a good initial speed of convergence. Also, it should not be too large to guarantee the robust performance against impulsive noise. On the other hand, at the steady-state stage of the algorithm, smaller values of  $\xi_k(i)$  lead to a lower estimation error. Hence, according to the above requirements, a simple and effective way is that the adaptation of  $\xi_k(i)$  depends on the maximum updated distance of the estimates shown in (11). Also, taking advantage of the diffusion cooperation for  $\xi_k(i)$ , we propose to adaptively update it as follows:

$$\zeta_k(i+1) = \nu \xi_k(i) + (1 - \nu) \mu^2 \min \left[ \frac{e_k^2(i)}{\|\mathbf{u}_k(i)\|_2^2}, \xi_k(i) \right], \tag{12}$$

$$\xi_k(i+1) = \sum_{m \in \mathcal{N}_k} c_{m,k} \zeta_m(i+1), \tag{13}$$

where  $\nu$  is the forgetting factor with  $0 < \nu < 1$  and  $\zeta_k(i)$  denotes the intermediate variable of  $\xi_k(i)$ . In (12), at every node  $k$ ,  $\xi_k(i)$  can be initialized as  $\xi_k(0) = \sigma_{d,k}^2 / (M \sigma_{u,k}^2)$ , where  $\sigma_{d,k}^2$  and  $\sigma_{u,k}^2$  are the powers of signals  $d_k(i)$  and  $u_k(i)$ , respectively. As a result, (8), (9), (10), (12) and (13) constitute the VT-dHNLMS algorithm, and for clarity which is summarized in Table 1 after rearranging (10).

**Remark 1** Intuitively, from (10) we can get some insights. In most of time instants, impulsive noise does not happen due to small probability, which means  $(|e_k(i)| / \|\mathbf{u}_k(i)\|_2) \leq \sqrt{\xi_k(i)}$  and the VT-dHNLMS algorithm works as a dNLMS algorithm with the fixed step size  $\mu \leq 1$ . Whenever impulsive noise appears, due to its large amplitude we have  $(|e_k(i)| / \|\mathbf{u}_k(i)\|_2) > \sqrt{\xi_k(i)}$ . In this case, (10) reduces to

$$\boldsymbol{\psi}_k(i+1) = \mathbf{w}_k(i) + \mu_k(i) \text{sign}(e_k(i)) \mathbf{u}_k(i),$$

which reveals that the VT-dHNLMS algorithm becomes the dSE-LMS algorithm with time-varying step sizes  $\mu_k(i) = \mu \sqrt{\xi_k(i)} / \|\mathbf{u}_k(i)\|_2$ , thereby providing good robustness against impulsive noise. Furthermore,  $\xi_k(i)$  given in (12) and (13) is decreasing as the algorithm converges, which further ensures the robustness of the algorithm. In a nutshell, the proposed VT-dHNLMS algorithm has good anti-jamming capability by switching automatically modes from the dNLMS to the dSE-LMS when impulsive noise appears.

**Table 1** Summary of the VT-dHNLMS algorithm

---

Initialization:  $\mathbf{w}_k(0) = \mathbf{0}$ ,  $\xi_k(0) = \sigma_{d,k}^2 / (M\sigma_{u,k}^2)$

Parameters:  $0 < \nu < 1$ , and  $0 < \mu \leq 1$

**for** iteration  $i$  **do**

*Adaptation step:*

**for** each node  $k$  **do**

$e_k(i) = d_k(i) - \mathbf{w}_k^T(i)\mathbf{u}_k(i)$

$\mu_k(i) = \mu \min \left[ \frac{\sqrt{\xi_k(i)} \|\mathbf{u}_k(i)\|_2}{|e_k(i)|}, 1 \right]$

$\boldsymbol{\psi}_k(i+1) = \mathbf{w}_k(i) + \mu_k(i) \frac{e_k(i)\mathbf{u}_k(i)}{\|\mathbf{u}_k(i)\|_2^2}$

$\zeta_k(i+1) = \nu\xi_k(i) + (1-\nu)\mu^2 \min \left[ \frac{e_k^2(i)}{\|\mathbf{u}_k(i)\|_2^2}, \xi_k(i) \right]$

**end**

*Combination step:*

**for** each node  $k$  **do**

$\mathbf{w}_k(i+1) = \sum_{m \in \mathcal{N}_k} c_{m,k} \boldsymbol{\psi}_m(i+1)$

$\xi_k(i+1) = \sum_{m \in \mathcal{N}_k} c_{m,k} \zeta_m(i+1)$

**end**

**end**

---

On the other hand, we rearrange (10) as:

$$\boldsymbol{\psi}_k(i+1) = \mathbf{w}_k(i) + \mu_k(i) \frac{\mathbf{u}_k(i)e_k(i)}{\|\mathbf{u}_k(i)\|_2^2}, \quad (14)$$

where

$$\mu_k(i) = \mu \min \left[ \frac{\sqrt{\xi_k(i)} \|\mathbf{u}_k(i)\|_2}{|e_k(i)|}, 1 \right]. \quad (15)$$

Evidently, (14) and (15) illustrate that the proposed algorithm can be interpreted as a modified dNLMS algorithm with time-dependent step sizes. Moreover, whenever impulsive noise occurs, (15) will immediately force the step size  $\mu_k(i)$  to a tiny value. It is well-known that for the GD-based adaptive algorithms, reducing the step size is also able to mitigate the adverse influence of impulsive noise on the performance of algorithms [44,52]. In an ideal case, we should set  $\mu_k(i) = 0$  for the occurrence instant of impulsive noise meaning that the adaptation is halted. In addition, the step sizes given by (15) are always limited in the range of 0 to  $\mu$  and go toward zero as  $i \rightarrow \infty$  since  $\xi_k(i)$  converges to zero as  $i \rightarrow \infty$  (see ‘‘Appendix A’’ for the proof). Therefore, (15) is a novel VSS approach (which can also be observed in Fig. 7) so that the proposed VT-dHNLMS algorithm not only speeds up the convergence but also reduces the steady-state estimation error.



**Table 2** The NC method

---

Initialization:  $\Theta_{old,k} = \Theta_{new,k} = 0, \hat{\sigma}_{e,k} = 0$

if  $i = nV_t, n = 0, 1, 2, \dots$

$$A_{e,k}(i) = \left[ \frac{e_k^2(i)}{\|\mathbf{u}_k(i)\|_2^2}, \frac{e_k^2(i-1)}{\|\mathbf{u}_k(i-1)\|_2^2}, \dots, \frac{e_k^2(i-N_w+1)}{\|\mathbf{u}_k(i-N_w+1)\|_2^2} \right]$$

$$\hat{\sigma}_{e,k} = \kappa \hat{\sigma}_{e,k} + (1 - \kappa) \text{median}(A_{e,k}(i))$$

$$\Theta_{new,k} = \sum_{m \in \mathcal{N}_k} c_{m,k} \hat{\sigma}_{e,m}$$

$$\Delta_k(i) = \frac{\Theta_{new,k} - \Theta_{old,k}}{\xi_k(i)}$$

end

if  $\Delta_k(i) > l_{th}$

$$\xi_k(i+1) = \xi_k(0)$$

else if  $\Theta_{new,k} > \Theta_{old,k}$

$$\xi_k(i+1) = \xi_k(i) + (\Theta_{new,k} - \Theta_{old,k})$$

else

$$\xi_k(i+1) = v \xi_k(i) + (1 - v) \mu^2 \min \left[ \frac{e_k^2(i)}{\|\mathbf{u}_k(i)\|_2^2}, \xi_k(i) \right]$$

end

$$\xi_k(i+1) = \sum_{m \in \mathcal{N}_k} c_{m,k} \xi_m(i+1)$$

$$\Theta_{old,k} = \Theta_{new,k}$$


---

**Remark 2** The adaptation of  $\xi_k(i)$  given by (12) and (13) is the decreasing function with respect to the iteration  $i$ . Accordingly, after  $\mathbf{w}^o$  changes suddenly,  $\xi_k(i)$  will continue to become small rather than a desired large value so that the convergence of the algorithm again to the steady-state (i.e., the tracking capability) is poor. To overcome this limitation, by referring to our previous NC method in [54], we take advantage of similar method to reset  $\xi_k(i)$  for adapting the sudden change of  $\mathbf{w}^o$ . Table 2 gives this method, where  $\Theta_{old,k}$  denotes the past value of  $\Theta_{new,k}$ , and  $0 < \kappa < 1$  is the weighting factor. As one can see, if  $\Delta_k(i) > l_{th}$  meaning that  $\mathbf{w}^o$  undergoes a sudden change,  $\xi_k(i)$  will be reset as  $\xi_k(0)$ , where  $l_{th} > 0$  is predefined. Differing from [54], we here employ the median operator,  $\text{median}(\cdot)$ , to remove the abnormal data disturbed by impulsive noise from the sliding data window with the length of  $N_w$  so that  $\Delta_k(i)$  is insensitive to impulsive noise. It is stressed that larger  $N_w$  leads to a more robust  $\Delta_k(i)$  but requires a higher complexity, so  $N_w$  should be properly chosen. Since  $\Delta_k(i)$  is computed only every  $V_t$  iterations, large  $V_t$  reduces the computational cost but with delayed tracking capability; typically, good choice is  $V_t = \rho M$  with  $1 \leq \rho \leq 3$ . Note that these parameters in the NC method are not coupled each other, i.e., each parameter plays a specific role, and thus, their choices are easy.

### 3.2 Computational Complexity

Table 3 compares the computational complexity of the dSE-LMS [37], dNLMS, VSS-dNLMS presented in [20], dHNLMS [29] algorithms with that of the proposed

**Table 3** Computational complexity of algorithms for node  $k$  per iteration

Algorithms	Multiplications	Additions	Divisions	Comparisons	Square roots	Medians
dSE-LMS	$M \mathcal{N}_k  + 2M$	$M \mathcal{N}_k  + M$	–	–	–	–
dNLMS	$M \mathcal{N}_k  + 3M + 1$	$M \mathcal{N}_k  + 2M$	1	–	–	–
VSS-dNLMS	$(M+1) \mathcal{N}_k  + 3M + 7$	$(M+1) \mathcal{N}_k  + 2M + 3$	$2 \mathcal{N}_k  + 3$	–	–	–
dHNLMS	$M \mathcal{N}_k  + 3M + 3$	$M \mathcal{N}_k  + 2M$	2	1	1	–
VT-dHNLMS	$(M+1) \mathcal{N}_k  + 3M + 6$	$(M+1) \mathcal{N}_k  + 2M$	2	2	1	–
NC method	$( \mathcal{N}_k  + 2)/V_t$	$ \mathcal{N}_k /V_t + 2$	$1/V_t$	2	–	$1/V_t$

VT-dHNLMS algorithm for node  $k$  per iteration, where  $|\mathcal{N}_k|$  denotes the cardinality of  $\mathcal{N}_k$ . Compared with the dHNLMS, the additional complexity of the proposed algorithm arises from equations (12) and (13), which requires  $|\mathcal{N}_k| + 3$  multiplications,  $|\mathcal{N}_k|$  additions, and 1 comparison. Also, the proposed algorithm requires extra communication cost for transmitting  $\zeta_k$  to its neighbors, i.e.,  $|\mathcal{N}_k| - 1$  scalars per node  $k$ . In comparison with the VT-dHNLMS algorithm, the computational complexity of the NC method is negligible due mainly to  $V_t \geq M$ .

## 4 Performance Analyses

### 4.1 Stability Analysis

From (15) and inequality (11), we obtain, for any node  $k$ :

$$\mu_k(i) \leq \mu \leq 1 \text{ and } E\{\mu_k(i)\} \leq \mu \leq 1. \quad (16)$$

This means that  $\mu$  is the upper bound of the time-varying step sizes. Remark 1 has analyzed whenever impulsive noise happens,  $\mu_k(i)$  will become very small for preventing the proposed algorithm from divergence. Hence, we can indirectly determine the stability of the proposed algorithm from the convergence condition of the dNLMS algorithm with constant step sizes. For diffusion algorithms with only exchanging the estimates between nodes [42,43,47], there is a conclusion that the diffusion algorithm is stable only if each node is stable to independently perform the estimation task using its own adaptation rule. Following this conclusion, the convergence condition of the dNLMS algorithm is that the step sizes  $\mu_k$  for nodes  $k = 1, \dots, N$  satisfy<sup>1</sup>

$$0 < \mu_k < 2, \quad (17)$$

which is also the step-size range of independently running the NLMS algorithm per node. In addition, references [15,41] have been proved that the step size  $\mu = 1$  leads

<sup>1</sup> The proof is omitted due to its simplicity.

to the fastest convergence of the NLMS algorithm; likewise, we can simply extend it to the dNLMS algorithm. It follows that the proposed VT-dHNLMS algorithm is stable. Moreover, we recommend  $\mu = 1$  for the proposed algorithm to obtain the fastest initial convergence.

In the sequel, we study the steady-state and evolution behaviors of the proposed algorithm with  $\mu = 1$  in impulsive noise.

## 4.2 Steady-State Behavior

Let us suppose the unknown vector  $\mathbf{w}^o$  to be stationary. Then, subtracting  $\mathbf{w}^o$  from (10) and applying (12), we have the following equation:

$$\tilde{\boldsymbol{\psi}}_k(i+1) = \tilde{\mathbf{w}}_k(i) - \sqrt{\frac{\zeta_k(i+1) - \nu \xi_k(i)}{1 - \nu}} \frac{\mathbf{u}_k(i) \text{sign}(e_k(i))}{\|\mathbf{u}_k(i)\|_2}, \quad (18)$$

where  $\tilde{\mathbf{w}}_k(i) \triangleq \mathbf{w}^o - \mathbf{w}_k(i)$  and  $\tilde{\boldsymbol{\psi}}_k(i) \triangleq \mathbf{w}^o - \boldsymbol{\psi}_k(i)$  denote the estimation error vector and the intermediate estimation error vector at node  $k$ , respectively. Similarly, using  $\mathbf{w}^o$  we change the combination step (8) to

$$\tilde{\mathbf{w}}_k(i+1) = \sum_{m \in \mathcal{N}_k} c_{m,k} \tilde{\boldsymbol{\psi}}_m(i+1). \quad (19)$$

Equations (18) and (19) will be the start point of the following analyses for the VT-dHNLMS algorithm. For convenience of analysis, some commonly used assumptions are given.

**Assumption 1** The input regressors  $\{\mathbf{u}_k(i)\}$  are zero mean with covariance matrices  $\mathbf{R}_k = E\{\mathbf{u}_k(i)\mathbf{u}_k^T(i)\}$  and spatially independent.

**Assumption 2** The input regressors  $\{\mathbf{u}_k(i)\}$  are independent of the additive noises  $\{v_m(j)\}$  for all  $k, m, i$ , and  $j$ .

**Assumption 3** The input regressors  $\{\mathbf{u}_k(i)\}$  are independent of the estimates  $\{\mathbf{w}_m(j)\}$  for  $j \leq i$  and all  $k, m$ . It is so-called *independence assumption*, which is customary for (distributed) adaptive algorithms [41,42,53].

**Assumption 4** The additive noise  $v_k(i)$  is described by the CG process, i.e.,  $v_k(i) = \theta_k(i) + \eta_k(i)$ . The background noise  $\theta_k(i)$  is a white Gaussian process with zero mean and variance  $\sigma_{\theta,k}^2$ . The impulsive noise  $\eta_k(i)$  is given by the Bernoulli–Gaussian process, i.e.,  $\eta_k(i) = b_k(i)g_k(i)$ , where  $b_k(i)$  is a Bernoulli process with  $P[b_k(i) = 1] = p_{r,k}$  and  $P[b_k(i) = 0] = 1 - p_{r,k}$ , and  $g_k(i)$  is a zero-mean white Gaussian process with variance  $\sigma_{g,k}^2 = \bar{h}\sigma_{\theta,k}^2$ ,  $\bar{h} \gg 1$ . Note that,  $p_{r,k}$  is also considered as the probability of occurring impulsive noise. Accordingly, the mean and variance of  $v_k(i)$  are zero and  $\sigma_{v,k}^2 = p_{r,k}(\bar{h} + 1)\sigma_{\theta,k}^2 + (1 - p_{r,k})\sigma_{\theta,k}^2$ , respectively. The CG model is used frequently for analyzing the performance of robust algorithms in impulsive noise environments<sup>2</sup> [28,37,48,54].

<sup>2</sup> The  $\alpha$ -stable process is also a useful impulsive noise model, but since it has no closed-form for the probability density function (pdf), it is difficult for performance analysis [18,30].

First, taking the squared  $l_2$ -norm of the both sides of (18), we obtain

$$\begin{aligned} \|\tilde{\boldsymbol{\psi}}_k(i+1)\|_2^2 &= \|\tilde{\boldsymbol{w}}_k(i)\|_2^2 \\ &\quad - 2\sqrt{\frac{\zeta_k(i+1) - \nu\xi_k(i)}{1-\nu}} \frac{\text{sign}(e_k(i))e_{k,a}(i)}{\|\mathbf{u}_k(i)\|_2} \\ &\quad + \frac{\zeta_k(i+1) - \nu\xi_k(i)}{1-\nu}, \end{aligned} \tag{20}$$

where  $e_{k,a}(i) \triangleq \tilde{\boldsymbol{w}}_k^T(i)\mathbf{u}_k(i)$  denotes the a priori output error at node  $k$ . Usually, the forgetting factor  $\nu$  is close to 1 so that the fluctuations of  $\xi_k(i)$  and  $\zeta_k(i)$  computed by (12) and (13) are relatively small compared to that of the input regressors and the error signals. As such, the variances of  $\xi_k(i)$  and  $\zeta_k(i)$  can be assumed to be small enough, namely  $\xi_k(i) - E\{\xi_k(i)\} \approx 0$  and  $\zeta_k(i) - E\{\zeta_k(i)\} \approx 0$ ; thereby, the following approximation can be established:

$$\begin{aligned} E \left\{ \sqrt{\frac{\zeta_k(i+1) - \nu\xi_k(i)}{1-\nu}} \frac{\text{sign}(e_k(i))e_{k,a}(i)}{\|\mathbf{u}_k(i)\|_2} \right\} &\approx \\ \sqrt{\frac{E\{\zeta_k(i+1)\} - \nu E\{\xi_k(i)\}}{1-\nu}} E \left\{ \frac{\text{sign}(e_k(i))e_{k,a}(i)}{\|\mathbf{u}_k(i)\|_2} \right\}. \end{aligned} \tag{21}$$

By incorporating (21), the expectation for both sides of (20) is expressed as

$$\begin{aligned} E\{\|\tilde{\boldsymbol{\psi}}_k(i+1)\|_2^2\} &= E\{\|\tilde{\boldsymbol{w}}_k(i)\|_2^2\} \\ &\quad - 2\sqrt{\frac{E\{\zeta_k(i+1)\} - \nu E\{\xi_k(i)\}}{1-\nu}} \underbrace{E \left\{ \frac{\text{sign}(e_k(i))e_{k,a}(i)}{\|\mathbf{u}_k(i)\|_2} \right\}}_{(b)} \\ &\quad + \frac{E\{\zeta_k(i+1)\} - \nu E\{\xi_k(i)\}}{1-\nu}. \end{aligned} \tag{22}$$

For  $M \gg 1$ , we can assume that the fluctuation of  $1/\|\mathbf{u}_k(i)\|_2$  be small so that the term (b) in (22) is simplified as

$$\begin{aligned} E \left\{ \frac{\text{sign}(e_k(i))e_{k,a}(i)}{\|\mathbf{u}_k(i)\|_2} \right\} &\approx r_k E\{\text{sign}(e_k(i))e_{k,a}(i)\} \\ &= r_k E\{\text{sign}(e_{k,a}(i) + v_k(i))e_{k,a}(i)\}, \end{aligned} \tag{23}$$

where  $r_k \triangleq E \left\{ \frac{1}{\|\mathbf{u}_k(i)\|_2} \right\}$ . Based on the conditioned expectation property, we have

$$E\{\text{sign}(e_{k,a}(i) + v_k(i))e_{k,a}(i)\} = E \left\{ E\{\text{sign}(e_{k,a}(i) + v_k(i))e_{k,a}(i) | \tilde{\boldsymbol{w}}_k(i)\} \right\}, \tag{24}$$

where the notation  $E\{s|q\}$  represents the expectation of  $s$  conditioned on  $q$ .

Applying the law of total probability under Assumption 4, the follow equation is established:

$$\begin{aligned}
 E\{\text{sign}(e_{k,a}(i) + v_k(i))e_{k,a}(i)|\tilde{\mathbf{w}}_k(i)\} \\
 = (1 - p_{r,k})E\{\text{sign}(e_{k,a}(i) + \theta_k(i))e_{k,a}(i)|\tilde{\mathbf{w}}_k(i)\} \\
 + p_{r,k}E\{\text{sign}(e_{k,a}(i) + \theta_k(i) + \eta_k(i))e_{k,a}(i)|\tilde{\mathbf{w}}_k(i)\}.
 \end{aligned}
 \tag{25}$$

Also, it is assumed that  $e_{k,a}(i)$  is zero-mean Gaussian when  $M \gg 1$ . As in [4], this assumption can be verified by the central limit theorem. Hence, variables  $(e_{k,a}(i) + \theta_k(i))$  and  $(e_{k,a}(i) + \theta_k(i) + \eta_k(i))$  can also be assumed to be Gaussian, and then we can use Price’s theorem in [39] and Assumptions 2 and 3 to further change (25) to

$$E\{\text{sign}(e_{k,a}(i) + v_k(i))e_{k,a}(i)|\tilde{\mathbf{w}}_k(i)\} = h_k E\{e_{k,a}^2(i)|\tilde{\mathbf{w}}_k(i)\}, \tag{26}$$

where

$$h_k = \sqrt{\frac{2}{\pi}} \left\{ \frac{p_{r,k}}{\sqrt{E\{e_{k,a}^2(i)\} + (\hbar + 1)\sigma_\theta^2}} + \frac{1 - p_{r,k}}{\sqrt{E\{e_{k,a}^2(i)\} + \sigma_\theta^2}} \right\} \neq 0. \tag{27}$$

Substituting (23), (24) and (26) into (22), we obtain

$$\begin{aligned}
 E\{\|\tilde{\boldsymbol{\psi}}_k(i + 1)\|_2^2\} &= E\{\|\tilde{\mathbf{w}}_k(i)\|_2^2\} \\
 &\quad - 2h_k r_k \sqrt{\frac{E\{\zeta_k(i + 1)\} - \nu E\{\xi_k(i)\}}{1 - \nu}} E\{e_{k,a}^2(i)\} \\
 &\quad + \frac{E\{\zeta_k(i + 1)\} - \nu E\{\xi_k(i)\}}{1 - \nu}.
 \end{aligned}
 \tag{28}$$

Second, by means of Jensen’s inequality [7, p.77], we take the squared  $l_2$ -norm for both sides of (19) and the expectation to arrive at

$$E\left\{\|\tilde{\mathbf{w}}_k(i + 1)\|_2^2\right\} \leq \sum_{m \in \mathcal{N}_k} c_{m,k} E\left\{\|\tilde{\boldsymbol{\psi}}_m(i + 1)\|_2^2\right\}. \tag{29}$$

To proceed, let us define two vectors over the network:

$$\begin{aligned}
 \mathcal{X}(i) &\triangleq \text{col}\left\{E\{\|\tilde{\boldsymbol{\psi}}_1(i)\|_2^2\}, \dots, E\{\|\tilde{\boldsymbol{\psi}}_N(i)\|_2^2\}\right\}, \\
 \mathcal{W}(i) &\triangleq \text{col}\left\{E\{\|\tilde{\mathbf{w}}_1(i)\|_2^2\}, \dots, E\{\|\tilde{\mathbf{w}}_N(i)\|_2^2\}\right\},
 \end{aligned}
 \tag{30}$$

and two diagonal matrices over the network

$$\begin{aligned}
 \mathcal{H} &\triangleq \text{diag}\{h_1 r_1, \dots, h_N r_N\}, \\
 \mathcal{S}(i) &\triangleq \text{diag}\left\{E\{e_{1,a}^2(i)\}, \dots, E\{e_{N,a}^2(i)\}\right\}.
 \end{aligned}
 \tag{31}$$

Also, the matrix  $\mathbf{C}$  collects the combination coefficients  $\{c_{m,k}\}$ , i.e.,  $[\mathbf{C}]_{m,k} = c_{m,k}$ . Then, using these definitions in (30), (31) and (A.1), we formulate (28) and (29) for call the nodes as a unified form:

$$\begin{aligned} \mathbf{W}(i + 1) &\leq \mathbf{C}^T \mathbf{X}(i + 1) \\ &= \mathbf{C}^T \left[ \mathbf{W}(i) - 2\mathcal{H}\mathcal{S}_e(i)\sqrt{\boldsymbol{\zeta}(i)} + \boldsymbol{\zeta}(i) \right], \end{aligned} \tag{32}$$

where

$$\boldsymbol{\zeta}(i) \triangleq \frac{E\{\boldsymbol{\zeta}(i + 1)\} - \nu E\{\boldsymbol{\xi}(i)\}}{1 - \nu}. \tag{33}$$

By taking the  $\infty$ -norm of both sides of (32) and using  $\|\mathbf{C}^T\|_\infty = 1$ , we obtain

$$\begin{aligned} \|\mathbf{W}(i + 1)\|_\infty &\leq \|\mathbf{C}^T\|_\infty \|\mathbf{W}(i) - 2\mathcal{H}\mathcal{S}_e(i)\sqrt{\boldsymbol{\zeta}(i)} + \boldsymbol{\zeta}(i)\|_\infty \\ &= \|\mathbf{W}(i) - 2\mathcal{H}\mathcal{S}_e(i)\sqrt{\boldsymbol{\zeta}(i)} + \boldsymbol{\zeta}(i)\|_\infty. \end{aligned} \tag{34}$$

Since  $\mathcal{H}$  and  $\mathcal{S}_e(i)$  are diagonal matrices and their elements are positive, (34) can be equivalently rewritten as

$$\begin{aligned} E\{\|\tilde{\mathbf{w}}_k(i + 1)\|_2^2\} &\leq E\{\|\tilde{\mathbf{w}}_k(i)\|_2^2\} \\ &\quad - 2h_k r_k \sqrt{\frac{E\{\zeta_k(i + 1)\} - \nu E\{\xi_k(i)\}}{1 - \nu}} E\{e_{k,a}^2(i)\} \\ &\quad + \frac{E\{\zeta_k(i + 1)\} - \nu E\{\xi_k(i)\}}{1 - \nu}, \end{aligned} \tag{35}$$

for  $k = 1, \dots, N$ . Assuming that the algorithm has reached the steady-state, i.e., taking the limits of all the terms in (35) at  $i \rightarrow \infty$ , it is established that

$$2h_k r_k E\{e_{k,a}^2(\infty)\} \leq \sqrt{\frac{E\{\zeta_k(\infty)\} - \nu E\{\xi_k(\infty)\}}{1 - \nu}}, \tag{36}$$

where  $E\{e_{k,a}^2(\infty)\} \triangleq \lim_{i \rightarrow \infty} E\{e_{k,a}^2(i)\}$ ,  $E\{\zeta_k(\infty)\} \triangleq \lim_{i \rightarrow \infty} E\{\zeta_k(i)\}$ , and  $E\{\xi_k(\infty)\} \triangleq \lim_{i \rightarrow \infty} E\{\xi_k(i)\}$ . As pointed out in ‘‘Appendix A’’,  $E\{\zeta_k(\infty)\}$  and  $E\{\xi_k(\infty)\}$  approximate zero, and under Assumption 3, (36) leads further to

$$E\{e_{k,a}^2(\infty)\} = \text{Tr} \left\{ \lim_{i \rightarrow \infty} E\{\tilde{\mathbf{w}}_k(i)\tilde{\mathbf{w}}_k^T(i)\} \mathbf{R}_k \right\} \approx 0. \tag{37}$$

Since  $\mathbf{R}_k$  is positive definite, we obtain from (37) that

$$E\{\|\tilde{\mathbf{w}}_k(\infty)\|_2^2\} = \text{Tr} \left\{ \lim_{i \rightarrow \infty} E\{\tilde{\mathbf{w}}_k(i)\tilde{\mathbf{w}}_k^T(i)\} \right\} \approx 0, \tag{38}$$

for  $k = 1, \dots, N$ . The relation (38) illustrates that after a sufficiently many iterations, the proposed algorithm can converge to the true parameter vector  $\mathbf{w}^o$  in the mean square sense, in spite of the presence of impulsive noise.

### 4.3 Evolution Behavior

Although (38) reveals the convergence of the proposed algorithm in impulsive noise, this is a qualitative analysis due mainly to Eq. (29). For a quantitative analysis, we define the following vectors over the network:

$$\begin{aligned} \tilde{\mathbf{w}}(i) &\triangleq \text{col}\{\tilde{\mathbf{w}}_1(i), \dots, \tilde{\mathbf{w}}_N(i)\}, \\ \tilde{\boldsymbol{\psi}}(i) &\triangleq \text{col}\{\tilde{\boldsymbol{\psi}}_1(i), \dots, \tilde{\boldsymbol{\psi}}_N(i)\}, \end{aligned} \tag{39}$$

and the block diagonal matrix over the network

$$\boldsymbol{\Phi}(i) \triangleq \text{diag}\{\boldsymbol{\Phi}_1(i), \dots, \boldsymbol{\Phi}_N(i)\}, \tag{40}$$

where

$$\boldsymbol{\Phi}_k(i) = \sqrt{\frac{\zeta_k(i+1) - \nu \xi_k(i)}{1 - \nu}} \frac{\mathbf{u}_k(i) \text{sign}(e_k(i))}{\|\mathbf{u}_k(i)\|_2}, \text{ for } k = 1, \dots, N. \tag{41}$$

Then, equations (18) and (19) for all the nodes are formulated as the matrix form:

$$\begin{aligned} \tilde{\mathbf{w}}(i+1) &= \mathbf{C}^T \tilde{\boldsymbol{\psi}}(i) \\ &= \mathbf{C}^T [\tilde{\mathbf{w}}(i) - \boldsymbol{\Phi}(i)]. \end{aligned} \tag{42}$$

where  $\mathbf{C} \triangleq \mathbf{C} \otimes \mathbf{I}_M$ .

Enforcing the autocorrelation operation to both sides of (42), we get the following recursion:

$$\begin{aligned} \boldsymbol{\Xi}(i+1) &= \mathbf{C}^T \left[ \boldsymbol{\Xi}(i) - \underbrace{E\{\tilde{\mathbf{w}}(i)\boldsymbol{\Phi}^T(i)\}}_I - \underbrace{E\{\boldsymbol{\Phi}(i)\tilde{\mathbf{w}}^T(i)\}}_{II} + \right. \\ &\quad \left. \underbrace{E\{\boldsymbol{\Phi}(i)\boldsymbol{\Phi}^T(i)\}}_{III} \right] \mathbf{C}, \end{aligned} \tag{43}$$

where

$$\boldsymbol{\Xi}(i) \triangleq E\{\tilde{\mathbf{w}}(i)\tilde{\mathbf{w}}^T(i)\} \tag{44}$$

is the autocorrelation matrix of the network estimation error vector  $\tilde{\mathbf{w}}(i)$ . The  $k$ th diagonal block of  $\boldsymbol{\Xi}(i)$  with size of  $M \times M$ , defined as  $\boldsymbol{\Xi}_k(i) \triangleq E\{\tilde{\mathbf{w}}_k(i)\tilde{\mathbf{w}}_k^T(i)\}$ , is the autocorrelation matrix of the estimation error vector  $\tilde{\mathbf{w}}_k(i)$  at node  $k$ .

Next, we will show how to evaluate terms I–III in (43).

Applying the conditioned expectation into the  $(m, k)$ th block of the term I, we have

$$\begin{aligned} E\{\tilde{\mathbf{w}}_m(i)\boldsymbol{\Phi}_k^T(i)\} &= E\{E\{\tilde{\mathbf{w}}_m(i)\boldsymbol{\Phi}_k^T(i)|\tilde{\mathbf{w}}(i)\}\} \\ &\simeq E\{\tilde{\mathbf{w}}_m(i)E\{\boldsymbol{\Phi}_k^T(i)|\tilde{\mathbf{w}}_k(i)\}\}. \end{aligned} \tag{45}$$

By recalling the Price’s theorem, Assumptions 2, 3 and 4, and the law of total probability (similar to manipulations in Sect.4.2), we arrive at the relation that

$$E\{\Phi_k^T(i)|\tilde{\mathbf{w}}_k(i)\} = h_k(i)\sqrt{\frac{E\{\zeta_k(i+1)\} - \nu E\{\xi_k(i)\}}{1 - \nu}}\tilde{\mathbf{w}}_k^T(i)E\left\{\frac{\mathbf{u}_k(i)\mathbf{u}_k^T(i)}{\|\mathbf{u}_k(i)\|_2}\right\}, \tag{46}$$

where  $h_k(i)$  in (27) is rewritten as

$$h_k(i) = \sqrt{\frac{2}{\pi}}\left\{\frac{p_{r,k}}{\sqrt{\text{Tr}(\Xi_k(i)\mathbf{R}_k) + (\hbar + 1)\sigma_\theta^2}} + \frac{1 - p_{r,k}}{\sqrt{\text{Tr}(\Xi_k(i)\mathbf{R}_k) + \sigma_\theta^2}}\right\}. \tag{47}$$

By combining (45) and (46), the term I can be arranged as

$$E\{\tilde{\mathbf{w}}(i)\Phi^T(i)\} = \Xi(i)\mathbf{A}(i)(\mathbf{h}(i)\chi(i) \otimes \mathbf{I}_M), \tag{48}$$

where

$$\begin{aligned} \mathbf{A}(i) &\triangleq \text{diag}\left\{E\left\{\frac{\mathbf{u}_1(i)\mathbf{u}_1^T(i)}{\|\mathbf{u}_1(i)\|_2}\right\}, \dots, E\left\{\frac{\mathbf{u}_N(i)\mathbf{u}_N^T(i)}{\|\mathbf{u}_N(i)\|_2}\right\}\right\}, \\ \mathbf{h}(i) &\triangleq \text{diag}\{h_1(i), \dots, h_N(i)\}, \\ \chi(i) &\triangleq \text{diag}\left\{\sqrt{\frac{E\{\zeta_1(i+1)\} - \nu E\{\xi_1(i)\}}{1 - \nu}}, \dots, \right. \\ &\quad \left.\sqrt{\frac{E\{\zeta_N(i+1)\} - \nu E\{\xi_N(i)\}}{1 - \nu}}\right\}. \end{aligned} \tag{49}$$

The term II equals to the transpose of (48).

The evaluation of the term III is divided into two parts. For the  $k$ th diagonal block of  $E\{\Phi(i)\Phi^T(i)\}$ , it is expressed as

$$E\{\Phi_k(i)\Phi_k^T(i)\} = \chi_k^2(i)E\left\{\frac{\mathbf{u}_k(i)\mathbf{u}_k^T(i)}{\|\mathbf{u}_k(i)\|_2^2}\right\}, \tag{50}$$

where  $\chi_k(i)$  is the  $k$ th diagonal entry of  $\chi(i)$ . For the  $(m, k)$ th off-diagonal block of  $E\{\Phi(i)\Phi^T(i)\}$ , it is calculated by

$$\begin{aligned} E\{\Phi_m(i)\Phi_k^T(i)\} &= E\{E\{\Phi_m(i)\Phi_k^T(i)|\tilde{\mathbf{w}}(i)\}\} \\ &\simeq E\left\{E\{\Phi_m(i)|\tilde{\mathbf{w}}_m(i)\}E\{\Phi_k^T(i)|\tilde{\mathbf{w}}_k(i)\}\right\} \\ &= h_m(i)\chi_m(i)\mathbf{A}_m(i)E\{\tilde{\mathbf{w}}_m(i)\tilde{\mathbf{w}}_k^T(i)\}\mathbf{A}_k(i)\chi_k(i)h_k(i), \end{aligned} \tag{51}$$



where  $A_k(i)$  is the  $k$ th diagonal block of  $A(i)$ . Thus, according to (50) and (51), we obtain the term III:

$$E\{\Phi(i)\Phi^T(i)\} = (\mathbf{h}(i)\chi(i) \otimes \mathbf{I}_M)A(i) \left[ \Xi(i) - \check{\Xi}(i) \right] \times A(i)(\mathbf{h}(i)\chi(i) \otimes \mathbf{I}_M) + \check{A}(i), \tag{52}$$

where

$$\check{\Xi}(i) \triangleq \text{diag} \{ \Xi_1(i), \dots, \Xi_N(i) \},$$

$$\check{A}(i) \triangleq \text{diag} \left\{ \chi_1^2(i)E \left\{ \frac{\mathbf{u}_1(i)\mathbf{u}_1^T(i)}{\|\mathbf{u}_1(i)\|_2^2} \right\}, \dots, \chi_N^2(i)E \left\{ \frac{\mathbf{u}_N(i)\mathbf{u}_N^T(i)}{\|\mathbf{u}_N(i)\|_2^2} \right\} \right\}. \tag{53}$$

By inserting (48) and (53) into (43), we establish the recursion expression for  $\Xi(i)$ :

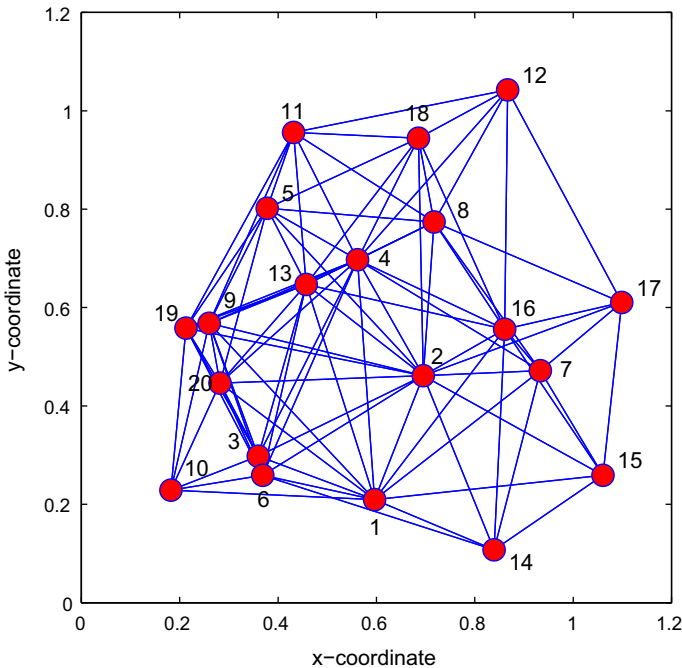
$$\begin{aligned} \Xi(i+1) &= \mathbf{C}^T \left\{ \Xi(i) - \Xi(i)A(i)(\mathbf{h}(i)\chi(i) \otimes \mathbf{I}_M) - (\mathbf{h}(i)\chi(i) \otimes \mathbf{I}_M)^T \right. \\ &\quad \times A^T(i)\Xi^T(i) + (\mathbf{h}(i)\chi(i) \otimes \mathbf{I}_M)A(i) \left[ \Xi(i) - \check{\Xi}(i) \right] \\ &\quad \left. \times A(i)(\mathbf{h}(i)\chi(i) \otimes \mathbf{I}_M) + \check{A}(i) \right\} \mathbf{C}. \end{aligned} \tag{54}$$

The network mean square deviation (MSD) over all the nodes is defined as  $\text{MSD}_{\text{net}}(i) \triangleq \frac{1}{N}\text{Tr}\{\Xi(i)\} = \frac{1}{N} \sum_{k=1}^N \text{MSD}_k(i)$ , where  $\text{MSD}_k(i) = \text{Tr}\{\Xi_k(i)\}$  is the MSD per node  $k$ . Thus, (54) formulates the MSD evolution of the proposed algorithm in impulsive noise. It is necessary for stressing that the implementation of (54) requires knowing  $E\{\zeta_k(i+1)\}$  and  $E\{\xi_k(i)\}$  beforehand. However, evaluating  $E\{\zeta_k(i+1)\}$  and  $E\{\xi_k(i)\}$  is difficult from (12) and (13), due mainly to the minimum operation of two random variables in impulsive noise. Consequently, we suggest that  $E\{\zeta_k(i+1)\}$  and  $E\{\xi_k(i)\}$  are obtained by the ensemble average in simulations. In a word, (54) is a semi-analytic result, but which is also useful for clarifying the evolution behavior of the proposed algorithm.

### 5 Simulation Results

In this section, simulation examples in the estimation of parameters are presented to assess the proposed algorithm and verify the performance analysis. The connected distributed network with  $N = 20$  nodes is shown in Fig. 2. The entries in the vector  $\mathbf{w}^o$  with length of  $M = 16$  are generated randomly from a zero-mean uniform distribution and then are normalized by  $\mathbf{w}^{oT}\mathbf{w}^o = 1$ . The input regressor  $\mathbf{u}_k(i)$  per node has a shifted structure, i.e.,  $\mathbf{u}_k(i) = [u_k(i), u_k(i-1), \dots, u_k(i-M+1)]^T$  [14,27,54], where  $u_k(i)$  is drawn from a first-order autoregressive (AR) system:

$$u_k(i) = t_k u_k(i-1) + \epsilon_k(i),$$



**Fig. 2** Topology of network with  $N = 20$  nodes

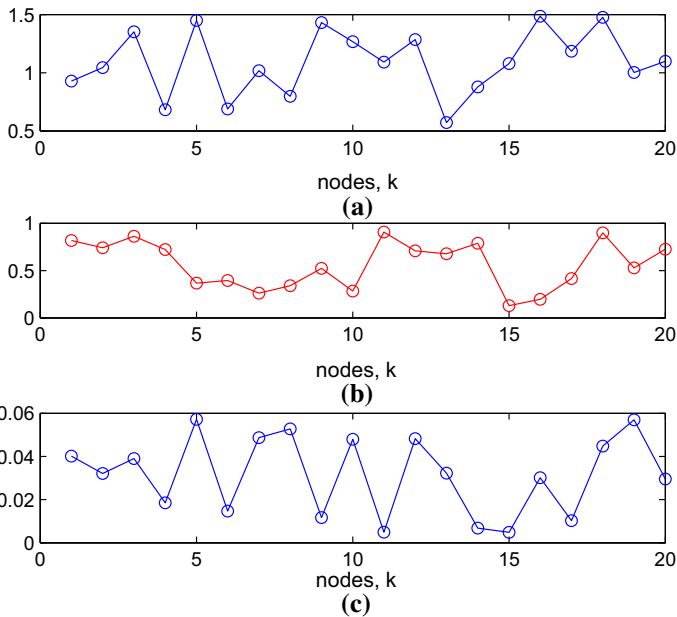
$\epsilon_k(i)$  is a zero-mean white Gaussian process with variance  $\sigma_{\epsilon,k}^2$ , and  $t_k$  is the AR coefficient. Figure 3a, b gives values of  $\sigma_{\epsilon,k}^2$  and  $t_k$  at nodes in the network, respectively. The network MSD is used for measuring the algorithm performance. Unless otherwise specified, the combination coefficients  $\{c_{m,k}\}$  for all the diffusion algorithms are given by the Metropolis rule [45]. The following figures are obtained by averaging the results over 200 independent trails.

## 5.1 Comparison of Algorithms

Case (1): CG noise

The additive noise is randomly generated from a CG process, as in Assumption 4, where the variance of the background noise,  $\sigma_{\theta,k}^2$ , is given in Fig. 3c, and the variance of the impulsive component is set to  $\sigma_{g,k}^2 = 1000\sigma_{y,k}^2$  with  $\sigma_{y,k}^2$  being the power of  $y_k(i) = \mathbf{u}_k^T(i)\mathbf{w}^o$ .

To begin with, Fig. 4 compares the performance of the VT-dHNLMS algorithm with that of the dNLMS, VSS-dNLMS, and dSE-LMS algorithms in Gaussian noise (i.e., the impulsive noise is absent,  $p_{r,k} = 0$ ). The parameters of algorithms are set based on the fact that they have the same either initial convergence rate or steady-state MSD, where the notations of parameters are the same as references. As one can see, the dSE-LMS algorithm has the slowest convergence rate. Both the VSS-dNLMS and VT-dHNLMS algorithms have the convergence as fast as the dNLMS



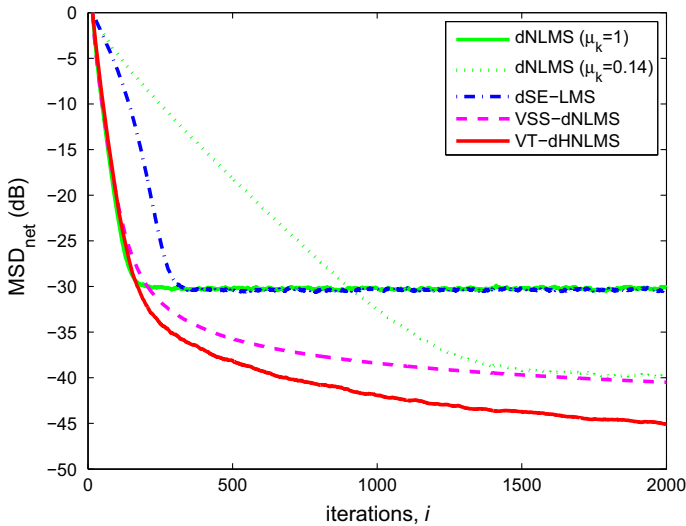
**Fig. 3** Values of **a**  $\sigma_{\epsilon,k}^2$ , **b**  $t_k$  and **c**  $\sigma_{\theta,k}^2$  at nodes

algorithm with large step size  $\mu_k = 1$ , and they have lower steady-state MSD. With small step size  $\mu_k = 0.14$ , the steady-state MSD of the dNLMS algorithm is as small as that of the VSS-dNLMS algorithm, but it converges slowly. It is worth noting that the proposed VT-dHNLMS algorithm behaves lower MSD than the VSS-dNLMS algorithm in the steady-state. In addition, the VSS-dNLMS algorithm requires the priori knowledge about the variances of the background noises at nodes,  $\{\sigma_{\theta,k}^2\}_{k=1}^N$ , while the VT-dHNLMS algorithm does not require it.

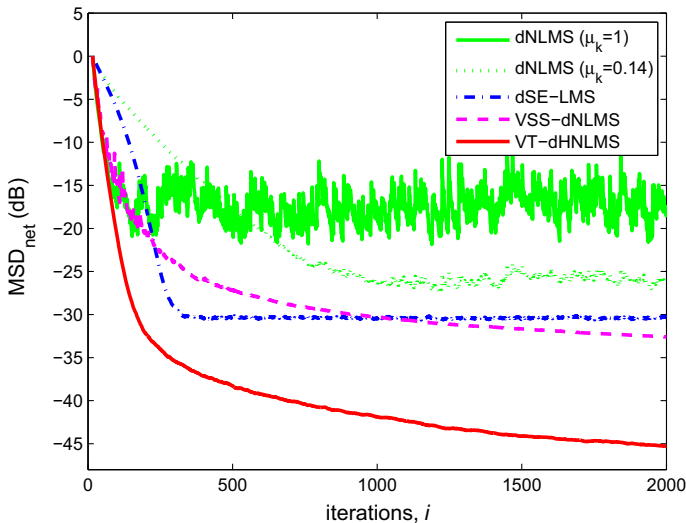
Then, Fig. 5 shows the performance of the algorithms in Fig. 4 in the situation that only node 1 in Fig. 2 is disturbed by impulsive noise with  $p_{r,1} = 0.01$ . It is seen from Fig. 5 that the dNLMS and VSS-dNLMS algorithms have performance deterioration, due to the fact that the wrong estimate (caused by impulsive noise) at node 1 is propagated to its neighboring nodes. In comparison, the dSE-LMS and VT-dHNLMS algorithms are insensitive to impulsive noise.

In addition to the algorithms in Fig. 5, Fig. 6 also examines the performance of the dLMP and RVWC-dLMS [3] algorithms,<sup>3</sup> in another scenario that all the nodes are affected by impulsive noise with  $p_{r,k} = 0.01$ . It is clear that the dNLMS and VSS-dNLMS algorithms suffer from substantial degradation in learning performance, since they are based on the squared output errors minimization with sensitivity to impulsive noise. The dSE-LMS (i.e., the dLMP with  $p = 1$ ) outperforms the dLMP with  $p = 1.5$  in the convergence, which illustrates that dLMP's performance in impulsive noise relies on the choice of  $p$ . Since the dSE-LMS does not use the amplitude information of output errors to update the estimates, its convergence is slower than that of the

<sup>3</sup> The dNLMS algorithm with large step size  $\mu_k = 1$  is omitted as it diverges in this case.



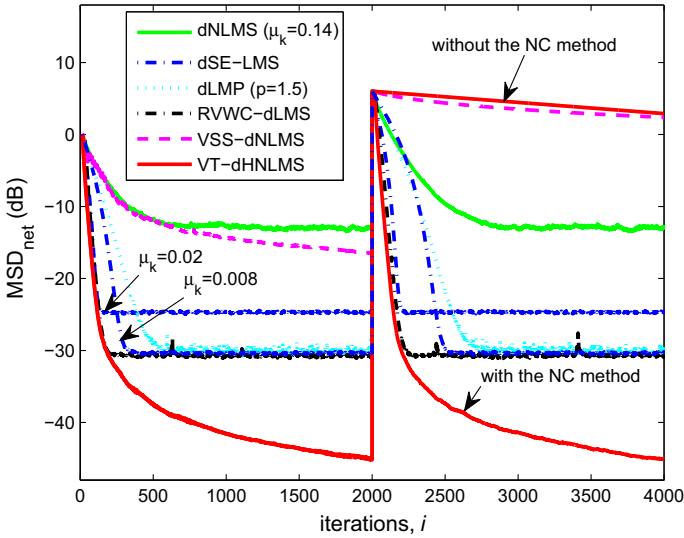
**Fig. 4** Network MSD curves of various diffusion algorithms. [No impulsive noise]. Parameter setting of algorithms is as follows:  $\mu_k = 0.008$  (dSE-LMS);  $\mu_k = 1$  and  $0.14$  (dNLMS);  $\beta = 1.2$ ,  $p_{k,0} = 10$  (VSS-dNLMS);  $\nu = 0.97$  (VT-dHNLMS)



**Fig. 5** Network MSD curves of various diffusion algorithms in the case of node 1 with impulsive noise. Parameter setting of algorithms is the same as Fig. 4

RVWC-dLMS. It is remarkable that the proposed VT-dHNLMS algorithm obtains faster convergence and lower steady-state MSD than these robust algorithms.

Moreover, to show the tracking capability of the algorithms in Fig. 6, the vector  $\mathbf{w}^o$  to be estimated is changed to  $-\mathbf{w}^o$  at iteration  $i = 2000$ . As can be seen, the VSS-dNLMS and VT-dHNLMS algorithms can not track the sudden change of  $\mathbf{w}^o$ .



**Fig. 6** Network MSD curves of various diffusion algorithms. [Impulsive noise occurs at all the nodes.] Parameter setting of algorithms is the same as Fig. 4, except  $\mu_k = 0.008$ ,  $p = 1.5$  for the dLMP;  $L = 16$ ,  $\alpha = 2.58$ ,  $\lambda = 1 - 1/M$ ,  $\mu_k = 0.048$  for the RVWC-dLMS. Also, the weighting coefficients in the combination step of RVWC-dLMS are specified in [3]; the Metropolis rule is used for the weighting coefficients in the adaptation step

Figure 7 depicts the evolution of the step size  $\mu_1(i)$  at node 1 (the results at other nodes are similar), for the VT-dHNLMS algorithm with and without the proposed NC method. The parameters of the NC method are chosen as  $N_w = 12$ ,  $\kappa = 0.99$ ,  $l_{th} = 30$  and  $\varrho = 3$ . Without the NC method, after  $\mathbf{w}^o$  suddenly changes, the magnitude of the normalized output error,  $|e_k(i)|/\|\mathbf{u}_k(i)\|_2$ , becomes much larger than  $\sqrt{\xi_k(i)}$  due to the mismatch of the estimate so that the step size through (15) jumps to a much smaller value rather than a desired larger value. This is why the VT-dHNLMS algorithm has no tracking capability. However, the NC method can make the step size through (15) jump to a large value in response to the change of  $\mathbf{w}^o$  so that it equips the VT-dHNLMS algorithm with the tracking capability, meanwhile not degrading the algorithm’s convergence performance.

Figure 8 gives the performance comparison of the proposed VT-dHNLMS with NC, the dHNLMS with the fixed threshold  $\xi$ , and the DRVSS-LMS [19] algorithms. The notation “non-coop” denotes the VT-dHNLMS with no cooperation ( $\mathbf{C} = \mathbf{I}_N$ ), i.e., each node performs independently the estimation task. To fairly compare the DRVSS-LMS algorithm, we choose parameters of impulsive noise as  $p_{r,k} = 0.05$  and  $\sigma_{g,k}^2 = 10000\sigma_{\theta,k}^2$ , since this algorithm needs these priori information; also, the input regressors of nodes are white Gaussian, i.e., by setting  $t_k = 0$  in the first-order AR system. As expected, due to the cooperation among interconnected nodes, the VT-dHNLMS algorithm improves the estimation performance for parameters of interest relative to the non-coop version. Compared with the dHNLMS, due to the adaptation of the threshold in (12) and (13), the VT-dHNLMS obtains faster convergence and lower steady-state MSD simultaneously. Although both DRVSS-LMS and VT-dHNLMS are

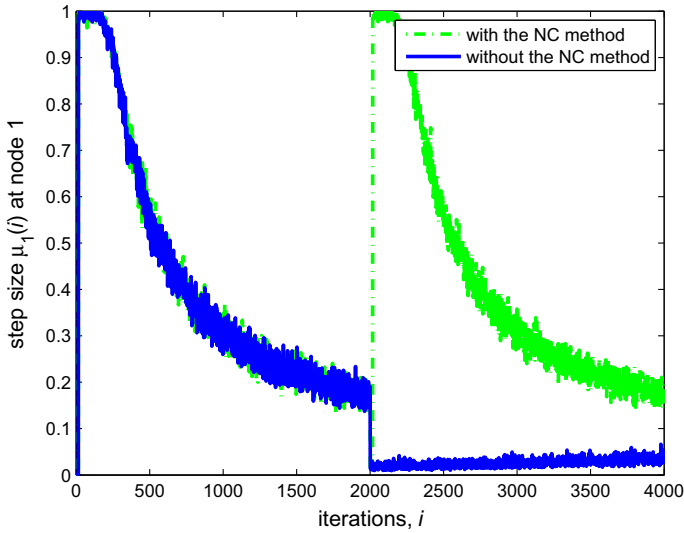


Fig. 7 Evolution of step size  $\mu_1(i)$  at node 1

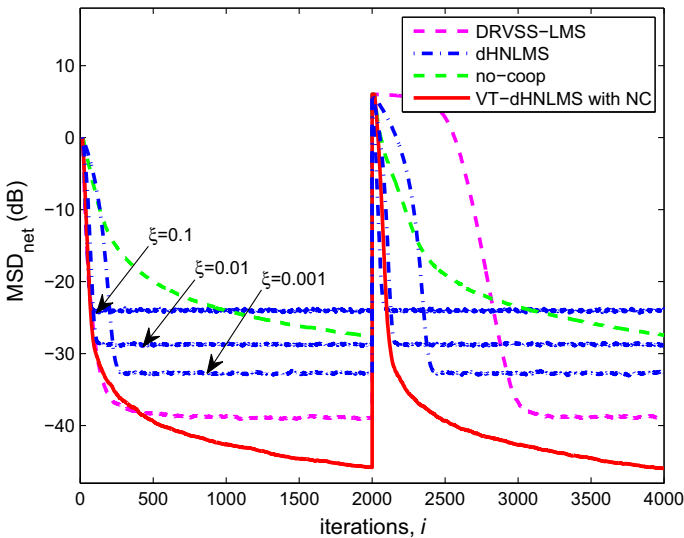


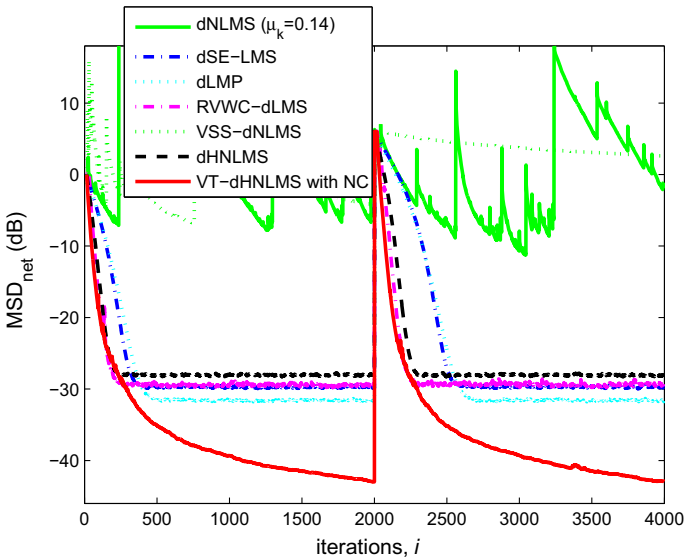
Fig. 8 Network MSD curves of the VT-dHNLMS algorithm and its counterparts. For the DRVSS-LMS, we use  $\text{median}(d_k^2(1), d_k^2(2), \dots, d_k^2(M))$  to initialize  $\sigma_{e1,k}^2$ , because how to initialize it has not been claimed in [19]. We choose the parameters  $\alpha = 2.8$  and  $\lambda_{1,2,3} = 0.99$  for the DRVSS-LMS and  $\nu = 0.94$  for the VT-dHNLMS

robust VSS diffusion variants based on the Huber function, the proposed VT-dHNLMS has better steady-state and tracking performance.

Table 4 shows the time required by the algorithms in Figs. 6 and 8, implemented in MATLAB R2013a, to be run on a PC with Intel(R) Core(TM) i7-7700 CPU @ 3.60 GHz

**Table 4** Run time of various diffusion algorithms

Algorithms	dSE-LMS	dLMP	dNLMS	RVWC-dLMS
Time (s)	194.24	200.28	211.08	1257.11
Algorithms	VSS-dNLMS	dHNLMS	DRVSS-LMS	VT-dHNLMS with NC
Time (s)	298.65	213.91	1297.21	296.46



**Fig. 9** Network MSD curves of various diffusion algorithms in  $\alpha$ -stable noise. Referring to Fig. 6, parameters of some algorithms are tuned as follows:  $\mu_k = 0.008$  for the dSE-LMS;  $p = 1.2$ ,  $\mu_k = 0.008$  for the dLMP;  $\xi = 0.01$  for the dHNLMS

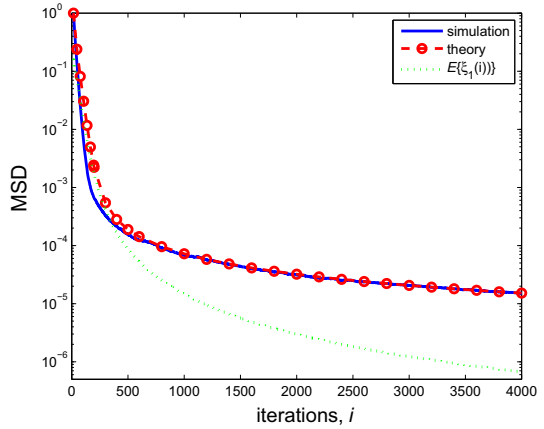
processor and 8.00GB RAM. It is seen that the time cost of the proposed algorithm is moderate among these algorithms, which is significantly lower than that of the RVWC-dLMS and DRVSS-LMS algorithms.

Case (2):  $\alpha$ -stable noise

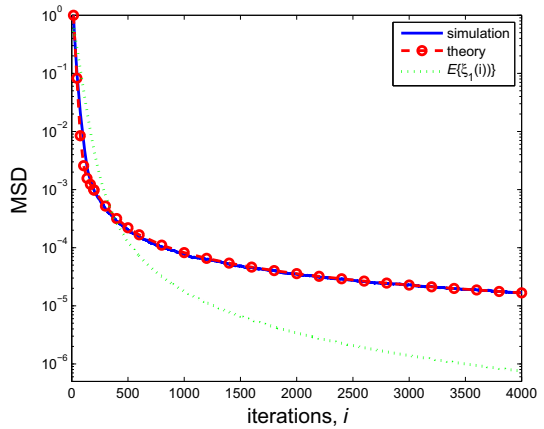
In this example, we investigate the performance of the above shown algorithms in  $\alpha$ -stable noise.<sup>4</sup> The characteristic function of  $\alpha$ -stable noise is  $\phi(t) = \exp(-\gamma|t|^\alpha)$ , where  $\alpha$  denotes the characteristic exponent (a small value leads to more outliers), and  $\gamma$  is the dispersion of the noise [18,49]. When  $\alpha = 1$  and  $\alpha = 2$ , it reduces to the Cauchy distribution and the Gaussian distribution, respectively. Here, we set  $\alpha = 1.25$  and  $\gamma = 2/15$ . The comparison results are shown in Fig. 9. As can be seen, among these robust algorithms (i.e., excluding the dNLMS and the VSS-dNLMS), the proposed VT-dHNLMS with NC behaves the best performance in terms of convergence rate, steady-state estimation error and tracking capability.

<sup>4</sup> The DRVSS-LMS algorithm is not shown since its implementation depends on the CG noise model.

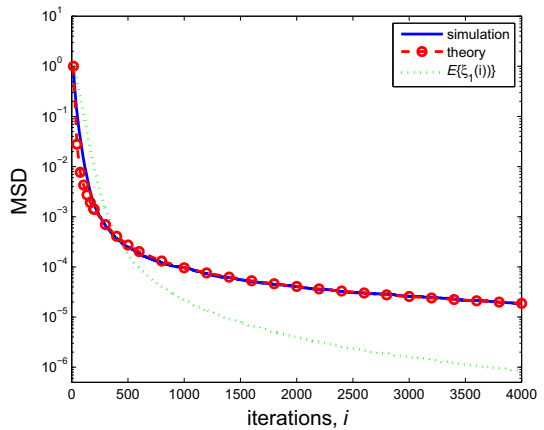
**Fig. 10** Network MSD curves of the VT-dNLMS algorithm in CG noise. **a**  $p_{r,k} = 0.01$ , **b**  $p_{r,k} = 0.05$ , **c**  $p_{r,k} = 0.1$



(a)



(b)



(c)



## 5.2 Verification of Analysis

Figure 10 compares the simulated MSD evolution with the theoretical MSD evolution computed by (54) for the proposed algorithm. Only  $E\{\xi_1(i)\}$  at node 1 is shown due to having similar results at other nodes. We set the impulsive parameters in the CG model to  $p_{r,k} = 0.01, 0.05$  or  $0.1$ , and  $\sigma_{g,k}^2 = 10000\sigma_{\theta,k}^2$ . As can be seen, the theoretical results are in agreement with the simulated results. Moreover, the decreasing property of  $E\{\xi_1(i)\}$  also supports the analysis in “Appendix A.”

## 6 Conclusion

In this study, we have improved the dHNLMS algorithm by adaptively adjusting the thresholds in a recursive way and proposed the VT-dHNLMS algorithm. Such recursive adjustment leads to that the proposed algorithm switch automatically from the dNLMS mode to the dSE-LMS mode with time-varying step sizes whenever impulsive noise appears. Theoretical analysis has shown that the proposed algorithm has good stability and learning performance in impulsive noise environments. Moreover, to respond quickly to the abrupt change of parameters of interest, we proposed a NC method for the thresholds. Simulation results have demonstrated the superiority of the proposed algorithm in Gaussian noise, CG impulsive noise, and  $\alpha$ -stable noise cases as compared to the competing diffusion algorithms.

**Acknowledgements** The work of Y. Yu was supported by National Natural Science Foundation of P.R. China (No. 61901400) and Doctoral Research Fund of Southwest University of Science and Technology in China (Grant No. 19zx7122). The work of H. Zhao was supported by National Natural Science Foundation of P.R. China (Nos. 61871461, 61571374, 61433011). The work of L. Lu was supported by National Natural Science Foundation of P.R. China (No. 61901285).

## Appendix A: Proof of $E\{\xi_k(i)\}$ Approximately Converging to Zero

We define the threshold vector over the network and its intermediate counterpart, as follows:

$$\begin{aligned}\xi(i) &\triangleq \text{col}\{\xi_1(i), \dots, \xi_N(i)\}, \\ \zeta(i) &\triangleq \text{col}\{\zeta_1(i), \dots, \zeta_N(i)\}.\end{aligned}\tag{A.1}$$

Following this, equations (12) and (13) for all the nodes can be expressed in a matrix form:

$$\begin{aligned}\xi(i+1) &= \mathbf{C}^T \zeta(i+1) \\ &= \mathbf{C}^T [\nu \xi(i) + (1-\nu) \min[\mathbf{f}(i), \xi(i)]],\end{aligned}\tag{A.2}$$

where

$$\mathbf{f}(i) \triangleq \text{col} \left\{ \frac{e_1^2(i)}{\|\mathbf{u}_1(i)\|_2^2}, \dots, \frac{e_N^2(i)}{\|\mathbf{u}_N(i)\|_2^2} \right\}.\tag{A.3}$$

Taking the  $\infty$ -norm both sides of (A.2), we get

$$\begin{aligned} \|\xi(i+1)\|_\infty &\leq \|C^T\|_\infty \cdot \|\zeta(i+1)\|_\infty \\ &= \|\zeta(i+1)\|_\infty \\ &\leq \nu \xi(i) + (1-\nu) \min[f(i), \xi(i)] \\ &\leq \|\xi(i)\|_\infty. \end{aligned} \tag{A.4}$$

Since  $\{\xi_k(i)\}$  is a positive sequence, (A.4) can deduce that  $\xi_k(i)$  and  $\zeta_k(i)$  from (12) and (13) are decreasing as the iteration  $i$  increases. It follows that the expectations  $E\{\xi_k(i)\}$  and  $E\{\zeta_k(i)\}$  are also positive and decreasing. Thus, taking the expectations of both sides of (A.2), we have

$$E\{\xi(i+1)\} = C^T[\nu E\{\xi(i)\} + (1-\nu)E\{\min[f(i), \xi(i)]\}]. \tag{A.5}$$

To continue developing the expression (A.5), again using the assumption that the variance of  $\xi_k(i)$  is small enough so that we can make the following approximation,

$$\begin{aligned} E\left\{\min\left[\frac{e_k^2(i)}{\|\mathbf{u}_k(i)\|_2^2}, \xi_k(i)\right]\right\} &\approx \\ E\{\xi_k(i)\}P_{k,i}[z_k > E\{\xi_k(i)\}] &+ \int_0^{E\{\xi_k(i)\}} z_k dF_{k,i}(z_k), \end{aligned} \tag{A.6}$$

where  $z_k \doteq e_k^2(i)/\|\mathbf{u}_k(i)\|_2^2$  denotes that both  $z_k$  and  $e_k^2(i)/\|\mathbf{u}_k(i)\|_2^2$  have the same distribution,  $P_{k,i}[a]$  denotes the probability of event  $a$ , and  $F_{k,i}(z_k)$  denotes the distribution function of  $z_k$  at time instant  $i$  at node  $k$ . Plugging (A.6) into (A.5), we have

$$E\{\xi(i+1)\} = C^T[\nu E\{\xi(i)\} + (1-\nu)\mathbf{P}(i)E\{\xi(i)\} + (1-\nu)\mathbf{z}(i)], \tag{A.7}$$

where

$$\mathbf{P}(i) = \text{diag}\{P_{1,i}[z_1 > E\{\xi_1(i)\}], \dots, P_{N,i}[z_N > E\{\xi_N(i)\}]\}, \tag{A.8}$$

and

$$\mathbf{z}(i) = \text{col}\left\{\int_0^{E\{\xi_1(i)\}} z_1 dF_{1,i}(z_1), \dots, \int_0^{E\{\xi_N(i)\}} z_N dF_{N,i}(z_N)\right\}. \tag{A.9}$$

Taking the limits for both sides of (A.7) as  $i \rightarrow \infty$ , we obtain

$$E\{\xi(\infty)\} = C^T[\nu E\{\xi(\infty)\} + (1-\nu)\mathbf{P}(\infty)E\{\xi(\infty)\} + (1-\nu)\mathbf{z}(\infty)]. \tag{A.10}$$

Then, we take the  $\infty$ -norm of both sides of (A.10), and after some simple manipulations with  $\|C^T\|_\infty = 1$ , getting the following inequality:

$$\|E\{\xi(\infty)\}\|_\infty \leq \|P(\infty)\|_\infty \cdot \|E\{\xi(\infty)\}\|_\infty + \|z(\infty)\|_\infty, \quad (\text{A.11})$$

where

$$\|P(\infty)\|_\infty = \max_{1 \leq k \leq N} P_{k,\infty}[z_k > E\{\xi_k(\infty)\}]. \quad (\text{A.12})$$

Due to the property of (A.12), we can equivalently formulate (A.11) as

$$\begin{aligned} E\{\xi_k(\infty)\} &\leq P_{k,\infty}[z_k > E\{\xi_k(\infty)\}] \cdot E\{\xi_k(\infty)\} \\ &\quad + \int_0^{E\{\xi_k(\infty)\}} z_k dF_{k,\infty}(z_k), \end{aligned} \quad (\text{A.13})$$

for  $k = 1, \dots, N$ , which further results in

$$P_{k,\infty}[z_k \leq E\{\xi_k(\infty)\}] \cdot E\{\xi_k(\infty)\} \leq \int_0^{E\{\xi_k(\infty)\}} z_k dF_{k,\infty}(z_k). \quad (\text{A.14})$$

It is difficult to solve (A.14) with respect to  $E\{\xi_k(\infty)\}$ , and thus for mathematical tractability, we consider solving the equal sign case in (A.14), i.e.,

$$P_{k,\infty}[z_k \leq E\{\xi_k(\infty)\}] \cdot E\{\xi_k(\infty)\} = \int_0^{E\{\xi_k(\infty)\}} z_k dF_{k,\infty}(z_k). \quad (\text{A.15})$$

For Eq. (A.15), Appendix A in [48] can be applied to obtain its solution that is  $E\{\xi_k(\infty)\} = 0$ . Because (A.15) is the upper bound of (A.14), we can infer that  $E\{\xi_k(i)\}$  for nodes  $k = 1, \dots, N$  will approximately converge to zero. Likewise, the intermediate value  $E\{\zeta_k(i)\}$  per node  $k$  also approximately converges to zero.

## References

1. R. Abdoee, V. Vakilian, B. Champagne, Tracking performance and optimal adaptation step-sizes of diffusion LMS networks. *IEEE Trans. Control Netw. Syst.* **5**(1), 67–78 (2016)
2. A. Ahmed, S. Zhang, Y.D. Zhang, Multi-target motion parameter estimation exploiting collaborative UAV network, in *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4459–4463 (2019)
3. D.C. Ahn, J.W. Lee, S.J. Shin, W.J. Song, A new robust variable weighting coefficients diffusion LMS algorithm. *Signal Process.* **131**, 300–306 (2017)
4. T.Y. Al-Naffouri, A.H. Sayed, Transient analysis of adaptive filters with error nonlinearities. *IEEE Trans. Signal Process.* **51**(3), 653–663 (2003)
5. S. Ashkezari-Toussi, H. Sadoghi-Yazdi, Robust diffusion LMS over adaptive networks. *Signal Process.* **158**, 201–209 (2019)
6. K.L. Blackard, T.S. Rappaport, C.W. Bostian, Measurements and models of radio frequency impulsive noise for indoor wireless communications. *IEEE J. Sel. Areas Commun.* **11**(7), 991–1001 (1993)
7. S. Boyd, L. Vandenberghe, *Convex Optimization* (Cambridge University Press, Cambridge, 2004)
8. F.S. Cattivelli, A.H. Sayed, Diffusion LMS strategies for distributed estimation. *IEEE Trans. Signal Process.* **58**(3), 1035–1048 (2010)

9. B. Chen, X. Wang, N. Lu, S. Wang, J. Cao, J. Qin, Mixture correntropy for robust learning. *Pattern Recognit.* **79**, 318–327 (2018)
10. B. Chen, L. Xing, X. Wang, J. Qin, N. Zheng, Robust learning with kernel mean  $p$ -power error loss. *IEEE Trans. Cybern.* **48**(7), 2101–2113 (2017)
11. B. Chen, L. Xing, B. Xu, H. Zhao, N. Zheng, J.C. Príncipe, Kernel risk-sensitive loss: definition, properties and application to robust adaptive filtering. *IEEE Trans. Signal Process.* **65**(11), 2888–2901 (2017)
12. B. Chen, L. Xing, H. Zhao, N. Zheng, J.C. Príncipe, Generalized correntropy for robust adaptive filtering. *IEEE Trans. Signal Process.* **64**(13), 3376–3387 (2016)
13. J. Chen, A.H. Sayed, Diffusion adaptation strategies for distributed optimization and learning over networks. *IEEE Trans. Signal Process.* **60**(8), 4289–4305 (2012)
14. S. Chouvardas, K. Slavakis, S. Theodoridis, Adaptive robust distributed learning in diffusion sensor networks. *IEEE Trans. Signal Process.* **59**(10), 4692–4707 (2011)
15. S. Ciochină, C. Paleologu, J. Benesty, An optimized NLMS algorithm for system identification. *Signal Process.* **118**, 115–121 (2016)
16. R.L. Das, M. Narwaria, Lorentzian based adaptive filters for impulsive noise environments. *IEEE Trans. Circuits Syst. I Regul. Pap.* **64**(6), 1529–1539 (2017)
17. P. Di Lorenzo, S. Barbarossa, A.H. Sayed, Distributed spectrum estimation for small cell networks based on sparse diffusion adaptation. *IEEE Signal Process. Lett.* **20**(12), 1261–1265 (2013)
18. P.G. Georgiou, P. Tsakalides, C. Kyriakakis, Alpha-stable modeling of noise and robust time-delay estimation in the presence of impulsive noise. *IEEE Trans. Multimed.* **1**(3), 291–301 (1999)
19. W. Huang, L. Li, Q. Li, X. Yao, Diffusion robust variable step-size LMS algorithm over distributed networks. *IEEE Access* **6**, 47511–47520 (2018)
20. S.M. Jung, J.H. Seo, P.G. Park, A variable step-size diffusion normalized least-mean-square algorithm with a combination method based on mean-square deviation. *Circuits Syst. Signal Process.* **34**(10), 3291–3304 (2015)
21. I. Kakalou, K.E. Psannis, P. Krawiec, R. Badea, Cognitive radio network and network service chaining toward 5G: challenges and requirements. *IEEE Commun. Mag.* **55**(11), 145–151 (2017)
22. S. Kanna, D.H. Dini, Y. Xia, S.Y. Hui, D.P. Mandic, Distributed widely linear kalman filtering for frequency estimation in power networks. *IEEE Trans. Signal Inf. Process. Netw.* **1**(1), 45–57 (2015)
23. S. Kar, J.M.F. Moura, Distributed consensus algorithms in sensor networks with imperfect communication: link failures and channel noise. *IEEE Trans. Signal Process.* **57**(1), 355–369 (2009)
24. H.S. Lee, S.E. Kim, J.W. Lee, W.J. Song, A variable step-size diffusion LMS algorithm for distributed estimation. *IEEE Trans. Signal Process.* **63**(7), 1808–1820 (2015)
25. C. Li, P. Shen, Y. Liu, Z. Zhang, Diffusion information theoretic learning for distributed estimation over network. *IEEE Trans. Signal Process.* **61**(16), 4011–4024 (2013)
26. C. Li, H. Wang, Distributed frequency estimation over sensor network. *IEEE Sens. J.* **15**(7), 3973–3983 (2015)
27. L. Li, J.A. Chambers, C.G. Lopes, A.H. Sayed, Distributed estimation over an adaptive incremental network based on the affine projection algorithm. *IEEE Trans. Signal Process.* **58**(1), 151–164 (2010)
28. Y.P. Li, T.S. Lee, B.F. Wu, A variable step-size sign algorithm for channel estimation. *Signal Process.* **102**, 304–312 (2014)
29. Z. Li, S. Guan, Diffusion normalized Huber adaptive filtering algorithm. *J. Frankl. Inst.* **355**(8), 3812–3825 (2018)
30. L. Lu, H. Zhao, W. Wang, Y. Yu, Performance analysis of the robust diffusion normalized least mean  $p$ -power algorithm. *IEEE Trans. Circuits Syst. II Express Briefs* **65**(12), 2047–2051 (2018)
31. W. Ma, B. Chen, J. Duan, H. Zhao, Diffusion maximum correntropy criterion algorithms for robust distributed estimation. *Digital Signal Process.* **58**, 10–19 (2016)
32. V. Mathews, S. Cho, Improved convergence analysis of stochastic gradient adaptive filters using the sign algorithm. *IEEE Trans. Acoust. Speech Signal Process.* **35**(4), 450–454 (1987)
33. J. Miller, J. Thomas, The detection of signals in impulsive noise modeled as a mixture process. *IEEE Trans. Commun.* **24**(5), 559–563 (1976)
34. T.G. Miller, S. Xu, R.C. de Lamare, H.V. Poor, Distributed spectrum estimation based on alternating mixed discrete-continuous adaptation. *IEEE Signal Process. Lett.* **23**(4), 551–555 (2016)
35. A. Nakai, K. Hayashi, An adaptive combination rule for diffusion lms based on consensus propagation, in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2018, pp. 3839–3843

36. J. Ni, Diffusion sign subband adaptive filtering algorithm for distributed estimation. *IEEE Signal Process. Lett.* **22**(11), 2029–2033 (2015)
37. J. Ni, J. Chen, X. Chen, Diffusion sign-error LMS algorithm: formulation and stochastic behavior analysis. *Signal Process.* **128**, 142–149 (2016)
38. P. Petrus, Robust Huber adaptive filter. *IEEE Trans. Signal Process.* **47**(4), 1129–1133 (1999)
39. R. Price, A useful theorem for nonlinear devices having gaussian inputs. *IRE Trans. Inf. Theory* **4**(2), 69–72 (1958)
40. S.A. Sayed, A.M. Zoubir, A.H. Sayed, Robust distributed estimation by networked agents. *IEEE Trans. Signal Process.* **65**(15), 3909–3921 (2017)
41. A.H. Sayed, *Adaptive Filters* (Wiley, New York, 2011)
42. A.H. Sayed, Adaptation, learning, and optimization over networks. *Found. Trends Mach. Learn.* **7**(4–5), 311–801 (2014)
43. A.H. Sayed, Adaptive networks. *Proc. IEEE* **102**(4), 460–497 (2014)
44. I. Song, P.G. Park, R.W. Newcomb, A normalized least mean squares algorithm with a step-size scaler against impulsive measurement noise. *IEEE Trans. Circuits Syst. II Express Briefs* **60**(7), 442–445 (2013)
45. N. Takahashi, I. Yamada, A.H. Sayed, Diffusion least-mean squares with adaptive combiners: formulation and performance analysis. *IEEE Trans. Signal Process.* **58**(9), 4795–4810 (2010)
46. S. Theodoridis, *Machine Learning: A Bayesian and Optimization Perspective* (Academic Press, London, 2015)
47. 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)
48. L.R. Vega, H. Rey, J. Benesty, S. Tressens, A new robust variable step-size NLMS algorithm. *IEEE Trans. Signal Process.* **56**(5), 1878–1893 (2008)
49. F. Wen, Diffusion least-mean p-power algorithms for distributed estimation in alpha-stable noise environments. *Electron. Lett.* **49**(21), 1355–1356 (2013)
50. S. Xu, R.C. de Lamare, H.V. Poor, Distributed compressed estimation based on compressive sensing. *IEEE Signal Process. Lett.* **22**(9), 1311–1315 (2015)
51. Y. Yu, H. Zhao, Robust incremental normalized least mean square algorithm with variable step sizes over distributed networks. *Signal Process.* **144**, 1–6 (2018)
52. Y. Yu, H. Zhao, B. Chen, Z. He, Two improved normalized subband adaptive filter algorithms with good robustness against impulsive interferences. *Circuits Syst. Signal Process.* **35**(12), 4607–4619 (2016)
53. Y. Yu, H. Zhao, R.C. de Lamare, L. Lu, Sparsity-aware subband adaptive algorithms with adjustable penalties. *Digital Signal Process.* **84**, 93–106 (2019)
54. Y. Yu, H. Zhao, R.C. de Lamare, Y. Zakharov, L. Lu, Robust distributed diffusion recursive least squares algorithms with side information for adaptive networks. *IEEE Trans. Signal Process.* **67**(6), 1566–1581 (2019)
55. S. Zhang, A. Ahmed, Y.D. Zhang, Sparsity-based collaborative sensing in a scalable wireless network, in *Big Data: Learning, Analytics, and Applications*, vol. 10989, 2019. <https://doi.org/10.1117/12.2521243>
56. S. Zhang, H.C. So, W. Mi, H. Han, A family of adaptive decorrelation NLMS algorithms and its diffusion version over adaptive networks. *IEEE Trans. Circuits Syst. I Regul. Pap.* **65**(2), 638–649 (2017)
57. Y. Zhou, S.C. Chan, K.L. Ho, New sequential partial-update least mean M-estimate algorithms for robust adaptive system identification in impulsive noise. *IEEE Trans. Ind. Electron.* **58**(9), 4455–4470 (2011)
58. X. Zhu, W.P. Zhu, B. Champagne, Spectrum sensing based on fractional lower order moments for cognitive radios in  $\alpha$ -stable distributed noise. *Signal Process.* **111**, 94–105 (2015)
59. M. Zimmermann, K. Dostert, Analysis and modeling of impulsive noise in broad-band powerline communications. *IEEE Trans. Electromagn. Compat.* **44**(1), 249–258 (2002)