A Kernel-Width Adaption Diffusion Maximum Correntropy Algorithm

Ying Guo [1], Bing Ma [1], and Yingsong Li [2]

[1]Affiliation not available
[2]Harbin Engineering University

October 30, 2023

Abstract

In this paper, a diffusion maximum correntropy criterion (DMCC) algorithm with adaption kernel width is proposed, denoting as DMCC$_{\rm adapt}$ algorithm, to find out a solution for dynamically choosing the kernel width. The DMCC$_{\rm adapt}$ algorithm chooses small kernel width at initial stage to improve its convergence speed rate, and uses large kernel width at completion stage to reduce its steady-state error. To render the proposed DMCC$_{\rm adapt}$ algorithm suitable for sparse system identifications, the DMCC$_{\rm adapt}$ algorithm based on proportional coefficient adjustment is realized and named as diffusion proportional maximum correntropy criterion (DPMCC$_{\rm adapt}$). The theoretical analysis and simulation results are presented to show that the DPMCC$_{\rm adapt}$ and DMCC$_{\rm adapt}$ algorithms have better convergence than the traditional diffusion AF algorithms under impulse noise and sparse systems.

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

A Kernel-Width Adaption Diffusion Maximum Correntropy Algorithm

YING GUO[1], BING MA[1], AND YINGSONG LI[2,3], (Senior Member, IEEE)

[1]School of Information Science and Engineering, Shenyang University of Technology, 110870, Liao Ning, China.(e-mail: mb1771244738@163.com, gy20072009@163.com)

[2]College of Information and Communication Engineering, Harbin Engineering University, China (e-mail: liyingsong@ieee.org)

[3]Key Laboratory of Microwave Remote Sensing, Chinese Academy of Sciences, Beijing 100190, China

Corresponding author: Yingsong Li (e-mail: liyingsong@ieee.org).

ABSTRACT Impulsive noises are widely existing in various systems like noise cancellation system and wireless communication systems, where adaptive filtering (AF) is always employed to identify specific systems. Additionally, the impulsive noises will affect the performance for estimating these systems, resulting in slow convergence or worse identification accuracy. In this paper, a diffusion maximum correntropy criterion (DMCC) algorithm with adaption kernel width is proposed, denoting as DMCC$_{adapt}$ algorithm, to find out a solution for dynamically choosing the kernel width. The DMCC$_{adapt}$ algorithm chooses small kernel width at initial stage to improve its convergence speed rate, and uses large kernel width at completion stage to reduce its steady-state error. To render the proposed DMCC$_{adapt}$ algorithm suitable for sparse system identifications, the DMCC$_{adapt}$ algorithm based on proportional coefficient adjustment is realized and named as diffusion proportional maximum correntropy criterion (DPMCC$_{adapt}$). The theoretical analysis and simulation results are presented to show that the DPMCC$_{adapt}$ and DMCC$_{adapt}$ algorithms have better convergence than the traditional diffusion AF algorithms under impulse noise and sparse systems.

INDEX TERMS Adaptive kernel width, diffusion algorithm, impulse noise, maximum correntropy criterion, sparse system identification

I. INTRODUCTION

Wireless sensor network (WSN) has been widely considered in the use of bridge detections, positioning, area monitoring, and air pollution monitoring. In the WSNs, the channel estimation that is belonged to the system identification should be considered for these distributed channel links. In addition, the channel response is always sparse in the practical system and the background noise is impulsive, which are important to apply practical system. For example, underwater acoustic channel and network echo channel [1]–[5] are existing in the distribution networks with mentioned characteristics. Moreover, the impulsive noise is common in many practical applications which happens in fan, radar and electromagnetic environments such as electromagnetic interferences [6]– [8]. In the past works, most of the noise models in the sensor network are considered as Gaussian mixture in the system identifications, which are diffusion over an interested area with certain topology. Furthermore, the linear finite impulse response (FIR) like the channel response impulse is sparse [9]–[12] and the noise in the system is additive, resulting in an effort to estimate the system from the noisy received signals based on adaptive filter technique for WSNs. There are two approaches for estimating the parameters in the WSN that are centralized estimation and diffusion estimation. In the centralized estimation, the mutual communications between the nodes require a large amount of energy, and the central node performs to fusion processing on the data information from all the received nodes. However, once the central node fails, the entire network function will be ineffective [13]. The biggest difference between the diffusion estimation and centralized estimation is that the former has no central node to avoid the disadvantage that the network lacks robustness due to the dependence on the central node. For diffusion one, each node in the network has communication and computing capabilities, and these nodes can complete the iterative operation of the data obtained from the received sensors. Then, these nodes can interact with the neighbor nodes in real time to update their estimated values with the merged results. Since the diffusion estimation has potential superiority, it has been studied extensively and divided into two types: incremental estimation and distributed estimation. In the incremental collaboration mode, all nodes are assigned to a circular loop structure, and the information is sent from one node to its neighbors in sequence. In this case, the collaborative mode has little traffic and needs less energy, but it is always unrealistic to form all the nodes into a ring structure, and it is also sensitive to link failures [14], [15]. Diffusion estimation are widely used because of its ease of implementation and good robustness. In the diffusion cooperation mode, nodes in the network are used to estimate the unknown system, which is to say that each node in the network can share and fuse information at its neighbor nodes through a combined method. Thus, the diffusion cooperation mode enables communication of the entire network.

As we know, adaptive filtering (AF) has been applied to many fields such as echo cancellation, system identification, active noise control [16], [17]. However, most of the AF methods are realized in the background noise of Gaussian distribution. In fact, many application scenes exist impulsive noise like the underwater noise and radar clutter [18], [19]. And hence, the early developed AFs cannot combat the impulsive noise well. On the other hand, the kernel method in statistical signal processing has been well developed and used to exploit new methods. In order to solve this problem, an AF algorithm based on least mean p-power (LMP) and an AF algorithm based on sign function were proposed [20], [21]. Recently, another AF method has been proposed based on kernel technique that is named as maximum correntropy criterion (MCC). The MCC constructs a new cost function to suppress the impulse noise by introducing an exponential term [22]- [31]. Due to the negative exponential term and kernel width of Gaussian kernel function in the MCC, the effects of the larger error is weakened [32]. The numerical instability caused by the increase of the deviation will not occur in the MCC since the correntropy has strong robustness. In addition, the MCC is similar to the LMS from the complexity view, making MCC widely used in various fields. In the MCC, the influence of kernel width has key effects on its performance and it is not easy to choose a suitable kernel width, which would be affect the racking performance and steady-state performance of the MCC. Then, an adaptive kernel width has been developed and integrated into the MCCs [33]–[35]. As for these algorithms, some of them are not stable in performance under strong pulse interference. In [36], a new variable kernel width MCC (VKW-MCC) algorithm was proposed, which solved the stability problem under strong impulse noise interference. However, the complexity has increased. On the other hand, there are many sparse systems in life, resulting in that the sparse AFs are required to exploit these prior information. Thus, many sparse AFs have been proposed [37]– [44].

In this paper, a diffusion MCC (DMCC) algorithm with adaption kernel width is proposed, denoting as DMCC$_{adapt}$ algorithm. At the initial stage, the proposed DMCC$_{adapt}$ provides a small kernel width via a kernel width updating scheme since the tracking error is large. Then, DMCC$_{adapt}$ provides a larger kernel width value to obtain a smaller steady-state error when it approaches to the convergence completion stage. Thus, the DMCC$_{adapt}$ ensures the convergence speed at the initial stage and controls steady-state error, which makes that it can track the system. To exploit the prior sparse information in practical systems, the idea of proportional scheme is integrated into the MCC$_{adapt}$ to implement the proportionality DMCC$_{adapt}$ (DPMCC$_{adapt}$). The theoretical analysis and simulation results are presented to show that the DPMCC$_{adapt}$ and DMCC$_{adapt}$ algorithms have better convergence performance than the traditional diffusion AF algorithms under impulse noise and sparse systems. The main contributions of this paper are listed as follows:

(1) A novel kernel width adaption MCC algorithm is created. In the initial stage of convergence, it gives a smaller kernel width, which speeds up the convergence, and provides a larger kernel width to reduce the steady-state error in steady-state stage.

(2) Considering MCC$_{adapt}$ with the distributed network, a new distributed estimation algorithm is proposed, denoted as DMCC$_{adapt}$, which can effectively combat impulsive interference in distributed networks.

(3) Integrating DMCC$_{adapt}$ with the idea of proportionality scheme, a DPMCC$_{adapt}$ algorithm is proposed, which is suitable for different sparsity system identifications.

A lowercase bold letter is used to denote a vector, and capital bold letter denotes a matrix, respectively. $(.)^T$ represents as transposed. $bvec(.)$ is the vectorization operator to transform matrix into column vector. $\otimes$ is denoted as $Kronecker$ $product$. For vector $\boldsymbol{x}$, $diag(\boldsymbol{x})$ denotes a diagonal matrix with its diagonal elements being the entries of vector $\boldsymbol{x}$. $\boldsymbol{I}_M$ denotes an $M \times M$ identity matrix. For two random matrices $\mathbb{A}$, $\mathbb{B}$, the operation $\|\mathbb{B}\|^2_{\mathbb{A}} = \mathbb{B}\mathbb{A}\mathbb{B}$.

II. PREVIOUS WORK

A. THE MCC ALGORITHM

Correntropy criterion is a nonlinear similarity measure between two random sequences $\boldsymbol{X}$ and $\boldsymbol{Y}$, defining as

$$\begin{aligned}\boldsymbol{V}(\boldsymbol{X},\boldsymbol{Y}) &= E[\kappa(\boldsymbol{X}-\boldsymbol{Y})] \\ &= \iint\limits_{x,y} \kappa(x-y) f_{\boldsymbol{XY}}(x,y)dxdy,\end{aligned} \tag{1}$$

where $E(.)$ denotes the expectation operator, x, y are the elements in sequences $\boldsymbol{X}, \boldsymbol{Y}$. $f_{\boldsymbol{XY}}(x,y)$ is the joint probability density function of $\boldsymbol{X}, \boldsymbol{Y}$, $\kappa(.)$ represents the Mercer kernel function, where the most commonly used Mercer kernel function is Gaussian kernel, giving by

$$\kappa(x-y) = exp(\frac{-(x-y)^2}{2\sigma^2}), \tag{2}$$

where $\sigma > 0$ denotes the kernel width.

We consider that the desired signal $d(n)$ arises from the linear model

$$d(n) = \boldsymbol{u}^T(n)\boldsymbol{w}_0 + v(n), \tag{3}$$

where $\boldsymbol{u}(n) = [u(n) \quad u(n-1) \quad \dots \quad u(n-M+1)]^T$ denotes the input vector, M denotes the length of the AF, and the unknown parameter vector to be estimated is $\boldsymbol{w_0}$ that is a $M \times 1$ vector, and $v(n)$ is the background noise. The error signal is given by

$$e(n) = d(n) - \boldsymbol{u}^T(n)\boldsymbol{w}(n), \tag{4}$$

where $\boldsymbol{w}(n)$ is an estimate of $\boldsymbol{w}_0$ at iteration n.

In the MCC, the cost function is

$$J_{MCC}(n) = exp(\frac{-e^2(n)}{2\sigma^2}). \tag{5}$$

B. THE DMCC ALGORITHM

Since the MCC algorithm cannot be used for the distribution estimation, the DMCC has been proposed in [45]–[49], where the DMCC has two diffusion strategies that are combine-then-adapt (CTA) and adapt-then-combine (ATC). The ATC strategy can traverse all nodes more quickly than the CTA strategy, which is widely used [50], which will be used in this paper.

The DMCC uses the same cost function as the MCC to suppress non-Gaussian noises for each node in the network. In a diffusion network with N nodes, the unknown parameter vector to be estimated is $\boldsymbol{w_{0,k}}$ that is a $M \times 1$ vector, and the local observation of node k at the n iteration denotes as $\{d_k(n), \boldsymbol{u}_k(n)\}$, $k = 1, ..., N$, where

$$d_k(n) = \boldsymbol{w}_{0,k}^T\boldsymbol{u}_k(n) + v_k(n), \tag{6}$$

where $v_k(n)$ is the background noise, $v_k(n)$ and $\boldsymbol{u}_k(n)$ are mutually independent. The DMCC is to estimate $\boldsymbol{w}_{0,k}$ by maximizing a linear combination of the local correntropy within the node k's neighbor N_k, and the DMCC's cost function at node k is

$$J_k^{local}(n) = \sum_{l\in\mathcal{N}_k} \alpha_{l,k} exp(\frac{-e_{l,k}^2(n)}{2\sigma^2}), \tag{7}$$

where $e_{l,k}(n) = d_k(n) - \boldsymbol{w}_k^T\boldsymbol{u}_k(n)$. $\boldsymbol{w}_k$ is the estimation of $\boldsymbol{w}_{0,k}$, $\{\alpha_{l,k}\}$ are some non-negative combination coefficients satisfying

$$\alpha_{l,k} = \begin{cases} deg(k), & l \in N_k \\ 0, & l \notin N_k, \end{cases} \tag{8}$$

where $deg(k)$ represents the degree of node k (the number of k adjacent node-to-node links, including the agent k itself). The derivative with respect to $\boldsymbol{w}_k$ is given by

$$\begin{aligned} \nabla J_k^{local}(n) &= \frac{\partial J_k^{local}(n)}{\partial \boldsymbol{w}_k} \\ &= \sum_{l\in\mathcal{N}_k} \alpha_{l,k} \frac{exp(\frac{-e_{l,k}^2(n)}{2\sigma^2}) e_{l,k}(n)\boldsymbol{u}_k(n)}{\sigma^2}. \end{aligned} \tag{9}$$

Then, the DMCC algorithm for estimating $\boldsymbol{w}_{0,k}$ at node k based on the gradient method is written as

$$\begin{cases} \boldsymbol{\Phi}_k(n) = \boldsymbol{w}_k(n-1) + \dfrac{\mu_k exp(\frac{-e_{l,k}^2(n)}{2\sigma^2}) e_{l,k}(n)\boldsymbol{u}_k(n)}{\sigma^2} \\ \boldsymbol{w}_k(n) = \sum\limits_{l\in\mathcal{N}_k} \alpha_{l,k}\boldsymbol{\Phi}_l(n), \end{cases} \tag{10}$$

where $\boldsymbol{\Phi}_k(n)$ is the intermediate estimate of node k at time n.

III. THE PROPOSED ALGORITHMS

A. ADAPTIVE KERNEL WIDTH

The kernel width in the Gaussian kernel has a significant influence on the convergence speed of the MCC algorithms. When other parameters are fixed, a smaller kernel width can enhance the convergence rate but the steady-state error is large. If a larger kernel width is selected, the MCC converges slowly, but it has smaller steady-state error. Therefore, it is necessary to develop an adaptive kernel width to form the MCC$_{adapt}$.

Then, an on-line recursive scheme is implemented for kernel width updating in the MCC$_{adapt}$ algorithm. Herein, the error $e_{l,k}(n)$ is used to adjust the kernel width that is updated using

$$\sigma_k(n) = max\{(2\sigma_0 - 1)^{exp(-\frac{e_{l,k}^2(n)}{2})}, \sigma_0\}, \tag{11}$$

where σ_0 is same as the kernel width of the MCC.
Based on common knowledge, we know

$$0 < exp(-\frac{e_{l,k}^2(n)}{2}) < 1, \tag{12}$$

and thus

$$1 < \sigma_k(n) < 2\sigma_0 - 1, \tag{13}$$

When the error is large,

$$\sigma_k(n) \to 1, \tag{14}$$

and when the error is small,

$$\sigma_k(n) \to 2\sigma_0 - 1. \tag{15}$$

In general, MCC $\sigma_0 > 1$ [36], thus, $2\sigma_0 - 1 \geq \sigma_0$. The MCC$_{adapt}$ algorithm reduces the steady-state error compared with the MCC algorithm.

At the beginning of iteration, the MCC$_{adapt}$ uses the same kernel width as the MCC to obtain same initial convergence speed rate. Then, the MCC$_{adapt}$ algorithm switches to $(2\sigma_0 - 1)^{exp(-\frac{e_{l,k}^2(n)}{2})}$, which reduced the steady-state error.

B. THE PROPOSED DMCC ALGORITHM BASED ON ADAPTIVE KERNEL WIDTH

Based on the discussions, the cost function of the DMCC$_{adapt}$ is written as

$$J_k^{local}(n) = \sum_{l\in\mathcal{N}_k} \alpha_{l,k} exp(\frac{-e_{l,k}^2(n)}{2\sigma_k^2(n)}). \tag{16}$$

where

$$\sigma_k(n) = max\{(2\sigma_0 - 1)^{exp(-\frac{e_{l,k}^2(n)}{2})}, \sigma_0\}, \tag{17}$$

The derivative with respect to $\boldsymbol{w}_k$ is given by

$$\begin{aligned} \nabla J_k^{local}(n) &= \frac{\partial J_k^{local}(n)}{\partial \boldsymbol{w}_k} \\ &= \sum_{l \in \mathcal{N}_k} \alpha_{l,k} \frac{exp(\frac{-e_{l,k}^2(n)}{2\sigma_k^2(n)}) e_{l,k}(n) \boldsymbol{u}_k(n)}{\sigma_k^2(n)}. \end{aligned} \tag{18}$$

Then, the gradient decent scheme is used to get the updating at node k

$$\begin{aligned} \boldsymbol{w}_k(n) &= \boldsymbol{w}_k(n-1) + \mu_k \frac{\partial J_k^{local}(n)}{\partial \boldsymbol{w}_k} \\ &= \boldsymbol{w}_k(n-1) \\ &+ \sum_{l \in \mathcal{N}_k} \alpha_{l,k} \frac{\mu_k exp(\frac{-e_{l,k}^2(n)}{2\sigma_k^2(n)}) e_{l,k}(n) \boldsymbol{u}_k(n)}{\sigma_k^2(n)}, \end{aligned} \tag{19}$$

According to above analysis, one can obtain the general DMCC$_{adapt}$ method presented by

$$\begin{cases} \boldsymbol{\Phi}_k(n) = \boldsymbol{w}_k(n-1) + \dfrac{\mu_k exp(\frac{-e_{l,k}^2(n)}{2\sigma_k^2(n)}) e_{l,k}(n) \boldsymbol{u}_k(n)}{\sigma_k^2(n)} \\ \boldsymbol{w}_k(n) = \sum_{l \in \mathcal{N}_k} \alpha_{l,k} \boldsymbol{\Phi}_l(n). \end{cases} \tag{20}$$

C. THE PROPOSED DPMCC ALGORITHM BASED ON ADAPTIVE KERNEL WIDTH

In order to achieve better performance in sparse systems, the idea of proportional scheme is introduced into the DMCC$_{adapt}$ algorithm to implement (DPMCC$_{adapt}$). The update equation of the proposed DPMCC$_{adapt}$ is realized and presented as

$$\begin{aligned} \boldsymbol{w}_k(n) &= \boldsymbol{w}_k(n-1) + \frac{\mu_k \boldsymbol{G}_k(n) \nabla J_k^{local}(n)}{\boldsymbol{u}_k(n) \boldsymbol{G}_k(n) \boldsymbol{u}_k^T(n) + \theta} \\ &= \boldsymbol{w}_k(n-1) + \\ &\sum_{l \in \mathcal{N}_k} \alpha_{l,k} \frac{\mu_k \boldsymbol{G}_k(n) exp(\frac{-e_{l,k}^2(n)}{2\sigma_k^2(n)}) e_{l,k}(n) \boldsymbol{u}_k(n)}{\sigma_k^2(n)(\boldsymbol{u}_k(n) \boldsymbol{G}_k(n) \boldsymbol{u}_k^T(n)) + \theta}, \end{aligned} \tag{21}$$

where $\sigma_k(n)$ denotes adaption kernel width, θ is a regularization parameter, $\boldsymbol{G}_k(n) = diag\{g_{k,1}(n), g_{k,2}(n), ..., g_{k,M}(n)\}$ has a size of $M \times M$, and $g_{k,i}(n) > 0$, giving by

$$g_{k,i}(n) = \frac{\gamma_{k,i}(n)}{\sum_{i=1}^{M} \gamma_{k,i}(n)}, \tag{22}$$

$$\gamma_{k,i}(n) = max\{\varepsilon max\{\delta_p, S_{k,1}(n), ..., S_{k,M}(n)\}, S_{k,i}(n)\}, \tag{23}$$

$$\boldsymbol{S}_k(n) = ln(1 + |\boldsymbol{w}_k(n)|), \tag{24}$$

where ε is used to update the coefficient, which is $\frac{5}{M}$, δ_p avoids the stalling of all coefficients when $\boldsymbol{S}_k(n) = \boldsymbol{0}_M \times 1$ at initialization, where M denotes the length of the AF, and $\{\alpha_{l,k}\}$ denotes the joint coefficient between the nodes. We can find that the DPMCC$_{adapt}$ gives a proportional step size for each coefficient to ensure that the large coefficient can obtain a large step size for reducing the convergence time in sparse systems.

According to above analysis, one can obtain

$$\begin{cases} \boldsymbol{\Phi}_k(n) = \boldsymbol{w}_k(n-1) + \\ \qquad \dfrac{\mu_k \boldsymbol{G}_k(n) exp(\frac{-e_{l,k}^2(n)}{2\sigma_k^2(n)}) e_{l,k}(n) \boldsymbol{u}_k(n)}{\sigma_k^2(n)(\boldsymbol{u}_k(n) \boldsymbol{G}_k(n) \boldsymbol{u}_k^T(n)) + \theta} \\ \boldsymbol{w}_k(n) = \sum_{l \in \mathcal{N}_k} \alpha_{l,k} \boldsymbol{\Phi}_l(n). \end{cases} \tag{25}$$

To well understand the proposed algorithm, Table 1 summarizes the DPMCC$_{adapt}$ algorithm.

TABLE 1. Summary of the DPMCC$_{adapt}$

Initialize: $\varepsilon = \frac{5}{M}$, $\delta_p = 0.2$, $\theta = 0.01$, $\sigma_0 = 2$
For each iteration n
for k=1:N
  $e_{l,k}(n) = d_k(n) - \boldsymbol{w}_k^T(n) \boldsymbol{u}_k(n)$;
  $\boldsymbol{S}_k(n) = log(1 + |\boldsymbol{w}_k(n)|)$;
  for $i = 1 : M$
    $\gamma_{k,i}(n) = max\{\varepsilon max\{\delta_p, \{S_{k,1}(n), ..., S_{k,M}(n)\}, S_{k,i}(n)\}\}$;
    $g_{k,i}(n) = \frac{\gamma_{k,i}(n)}{\sum_{i=1}^{M} \frac{\gamma_{k,i}(n)}{M}}$;
    $\boldsymbol{G}_k(n) = diag\{g_{k,1}(n), g_{k,2}(n), ..., g_{k,M}(n)\}$;
  end
  $\sigma_k(n) = max\{(2\sigma_0 - 1)^{exp(\frac{-e_{l,k}^2(n)}{2})}, \sigma_0\}$;
  $\boldsymbol{\Phi}_k(n) = \boldsymbol{w}_k(n-1) + \frac{\mu_k \boldsymbol{G}_k(n) exp(\frac{-e_{l,k}^2(n)}{2\sigma_k^2(n)}) e_{l,k}(n) \boldsymbol{u}_k(n)}{\sigma_k^2(n)(\boldsymbol{u}_k(n) \boldsymbol{G}_k(n) \boldsymbol{u}_k^T(n)) + \theta}$;
end
for $l = 1 : N$
  $\boldsymbol{w}_k(n) = \sum_{l \in \mathcal{N}_k} \alpha_{l,k} \boldsymbol{\Phi}_l(n)$;
end

IV. PERFORMANCE ANALYSIS

The convergence performance of the proposed DPMCC$_{adapt}$ algorithm is investigated along with the following assumptions.

Hypothesis 1: All regression vectors $\boldsymbol{u}_k(n)$ are zero mean Gaussian sources and are independent in space and time.

Hypothesis 2: Nonlinear error $e_{l,k}(n)$ is independent of $\boldsymbol{u}_k(n)$.

Hypothesis 3: The mean of noise signal $v_k(n)$ is zero, which is independent of $\boldsymbol{u}_k(n)$.

As obtained data is exchanged between different nodes, their updates are affected by the previous estimates. Thus, the correlation between nodes should be considered to investigate the convergence performance of entire distributed network. Also, some new variables are introduced, and hence, the proposed DPMCC$_{adapt}$ algorithm is redefined as

$$\begin{cases} \mathbf{\Phi}_k(n) = \boldsymbol{w}_k(n-1) + \mu_k r_k(n)\boldsymbol{G}_{\boldsymbol{u}_k}(n)e_{l,k}(n)\boldsymbol{u}_k(n) \\ \qquad\quad = \boldsymbol{w}_k(n-1) + \boldsymbol{\rho}_k(n)\boldsymbol{G}_{\boldsymbol{u}_k}(n)e_{l,k}(n)\boldsymbol{u}_k(n) \\ \boldsymbol{w}_k(n) = \sum\limits_{l\in\mathcal{N}_k} \alpha_{l,k}\mathbf{\Phi}_l(n), \end{cases} \tag{26}$$

where

$$\rho_k(n) = \mu_k r_k(n), \tag{27}$$

$$r_k(n) = \frac{exp(\frac{-e_{l,k}^2(n)}{2\sigma_k^2(n)})}{\sigma_k^2(n)}, \tag{28}$$

$$\sigma_k(n) = max\{(2\sigma_0 - 1)^{exp(-\frac{e_{l,k}^2(n)}{2})}, \sigma_0\}, \tag{29}$$

$$\boldsymbol{G}_{u_k}(n) = \frac{\boldsymbol{G}_k(n)}{\boldsymbol{u}_k(n)\boldsymbol{G}_k(n)\boldsymbol{u}_k^T(n) + \theta}, \tag{30}$$

where $\rho_k(n)$ is new step size.

In order to convert local variables to global variables, we have following definitions:

$$\boldsymbol{T}(n) = diag\{\rho_1(n)\boldsymbol{I}_M, \rho_2(n)\boldsymbol{I}_M, ..., \rho_N(n)\boldsymbol{I}_M\}_{NM\times NM}, \tag{31}$$

$$\boldsymbol{h}(n) = col\{\boldsymbol{w}_1(n), \boldsymbol{w}_2(n), \dots, \boldsymbol{w}_N(n)\}_{NM\times 1}, \tag{32}$$

$$\mathbf{\Phi}(n) = col\{\mathbf{\Phi}_1(n), \mathbf{\Phi}_2(n), \dots, \mathbf{\Phi}_N(n)\}_{NM\times 1}, \tag{33}$$

$$\boldsymbol{U}(n) = diag\{\boldsymbol{u}_1(n), \boldsymbol{u}_2(n), \dots, \boldsymbol{u}_N(n)\}_{NM\times NM}, \tag{34}$$

$$\boldsymbol{d}(n) = col\{d_1(n), d_2(n), \dots, d_N(n)\}_{NM\times 1}, \tag{35}$$

$$\boldsymbol{v}(n) = col\{v_1(n), v_2(n), \dots, v_N(n)\}_{NM\times 1}, \tag{36}$$

$$\boldsymbol{G}(n) = diag\{\boldsymbol{G}_{\boldsymbol{u}_1}(n), \boldsymbol{G}_{\boldsymbol{u}_2}(n), \dots, \boldsymbol{G}_{\boldsymbol{u}_N}(n)\}_{NM\times NM}. \tag{37}$$

Based on the definitions and discussions above, a equation for entire distributed network is formed by considering the relationship between the nodes

$$\boldsymbol{d}(n) = \boldsymbol{U}(n)\boldsymbol{h}_0 + \boldsymbol{v}(n), \tag{38}$$

where $\boldsymbol{h}_0 = \boldsymbol{W}\boldsymbol{w}_{0,k}, \boldsymbol{W} = col\{\boldsymbol{I}_M, \boldsymbol{I}_M, ..., \boldsymbol{I}_M\}_{NM\times M}$, $\boldsymbol{h_0}$ denotes the global unknown parameter vector. The update equation can be modified for implementing the global network

$$\begin{aligned} \mathbf{\Phi}(n) =& \boldsymbol{h}(n-1) + \boldsymbol{T}(n)\boldsymbol{G}(n)\boldsymbol{U}^T(n) \\ & (\boldsymbol{d}(n) - \boldsymbol{U}(n)\boldsymbol{h}(n-1)), \end{aligned} \tag{39}$$

$$\boldsymbol{h}(n) = \boldsymbol{H}\mathbf{\Phi}(n), \tag{40}$$

where $\boldsymbol{h}(n)$ is the estimate vector of $\boldsymbol{h}_0$, $\boldsymbol{H} = \mathbf{\Theta}\otimes\boldsymbol{I}_M$, $\mathbf{\Theta}$ is the $N\times N$ diffusion combination matrix with entries of $\alpha_{l,k}$, $\sum_{l\in N_k}\alpha_{l,k} = 1$.

A. MEAN STABILITY ANALYSIS

A new global weight error vector is defined as

$$\tilde{\boldsymbol{h}}(n) = \boldsymbol{h}_0 - \boldsymbol{h}(n). \tag{41}$$

Based on the previous analysis, $\boldsymbol{H}\boldsymbol{h}_0 = \boldsymbol{h}_0$. Substituting the above formulas into (41), we have

$$\begin{aligned} \tilde{\boldsymbol{h}}(n) =& \boldsymbol{h}_0 - \boldsymbol{h}(n) \\ =& \boldsymbol{H}\boldsymbol{h}_0 - \boldsymbol{H}\mathbf{\Phi}(n) \\ =& \boldsymbol{H}\boldsymbol{h}_0 - \boldsymbol{H}[\boldsymbol{h}(n-1) + \boldsymbol{T}(n)\boldsymbol{G}(n)\boldsymbol{U}^T(n) \\ & (\boldsymbol{d}(n) - \boldsymbol{U}(n)\boldsymbol{h}(n-1))] \\ =& \boldsymbol{H}\tilde{\boldsymbol{h}}(n-1) - \boldsymbol{H}[\boldsymbol{T}(n)\boldsymbol{G}(n)\boldsymbol{U}^T(n) \\ & (\boldsymbol{d}(n) - \boldsymbol{U}(n)\boldsymbol{h}(n-1))] \\ =& \boldsymbol{H}\tilde{\boldsymbol{h}}(n-1) - \boldsymbol{H}[\boldsymbol{T}(n)\boldsymbol{G}(n)\boldsymbol{U}^T(n) \\ & (\boldsymbol{U}(n)\tilde{\boldsymbol{h}}(n-1) + \boldsymbol{v}(n))] \\ =& \boldsymbol{H}[\boldsymbol{I}_{NM} - \boldsymbol{T}(n)\boldsymbol{G}(n)\boldsymbol{U}^T(n)\boldsymbol{U}(n)] \\ & \tilde{\boldsymbol{h}}(n-1) - \boldsymbol{H}\boldsymbol{T}(n)\boldsymbol{G}(n)\boldsymbol{U}^T(n)\boldsymbol{v}(n). \end{aligned} \tag{42}$$

Then, the expectation is used on the both sides of equation (42) to get

$$\begin{aligned} E[\tilde{\boldsymbol{h}}(n)] =& \boldsymbol{H}[\boldsymbol{I}_{NM} - E[\boldsymbol{T}(n)\boldsymbol{G}(n)\boldsymbol{U}^T(n)\boldsymbol{U}(n)]] \\ & E[\tilde{\boldsymbol{h}}(n-1)] - \boldsymbol{H}E[\boldsymbol{T}(n)\boldsymbol{G}(n)\boldsymbol{U}^T(n)]E[\boldsymbol{v}(n)]. \end{aligned} \tag{43}$$

By using the assumptions, it is found that the matrix $\boldsymbol{T}(n)$ is independent of the matrix $\boldsymbol{U}(n)$, and hence, we can get

$$E[\boldsymbol{T}(n)\boldsymbol{G}(n)\boldsymbol{U}^T(n)\boldsymbol{U}(n)] \cong \boldsymbol{G}(n)E[\boldsymbol{T}(n)]E[\boldsymbol{U}^T(n)\boldsymbol{U}(n)], \tag{44}$$

where $\boldsymbol{R_U} = E[\boldsymbol{U}^T(n)\boldsymbol{U}(n)]$ is the auto-correlation matrix of $\boldsymbol{U}(n)$. Therefore, (43) can be written as

$$\begin{aligned} E[\tilde{\boldsymbol{h}}(n)] =& \boldsymbol{H}[\boldsymbol{I}_{NM} - \boldsymbol{G}(n)E[\boldsymbol{T}(n)]E[\boldsymbol{U}^T(n)\boldsymbol{U}(n)]] \\ & E[\tilde{\boldsymbol{h}}(n-1)] - \boldsymbol{H}\boldsymbol{G}(n) \\ & E[\boldsymbol{T}(n)]E[\boldsymbol{U}^T(n)]E[\boldsymbol{v}(n)] \\ =& \boldsymbol{H}[\boldsymbol{I}_{NM} - \boldsymbol{G}(n)E[\boldsymbol{T}(n)]\boldsymbol{R_U}]E[\tilde{\boldsymbol{h}}(n-1)] \\ & - \boldsymbol{H}\boldsymbol{G}(n)E[\boldsymbol{T}(n)]E[\boldsymbol{U}^T(n)]E[\boldsymbol{v}(n)]. \end{aligned} \tag{45}$$

From the hypothesis 3, the expectation of the second term on the right side of (45) is zero, resulting in

$$E[\tilde{\boldsymbol{h}}(n)] = \boldsymbol{H}[\boldsymbol{I}_{NM} - \boldsymbol{G}(n)E[\boldsymbol{T}(n)]\boldsymbol{R_U}]E[\tilde{\boldsymbol{h}}(n-1)]. \tag{46}$$

In order to achieve the stability, it should be satisfied [27]

$$\begin{aligned} & |\lambda_{max}(\boldsymbol{H}[\boldsymbol{I}_{NM} - \boldsymbol{G}(n)E[\boldsymbol{T}(n)]\boldsymbol{R_U}]| \\ & = |\lambda_{max}(\boldsymbol{H}\boldsymbol{Z}(n))| < 1, \end{aligned} \tag{47}$$

where $\boldsymbol{Z}(n) = \boldsymbol{I}_{NM} - \boldsymbol{G}(n)E[\boldsymbol{T}(n)]\boldsymbol{R_U}$, and λ_{max} represents its maximum eigenvalue. According to the relationship $\|\boldsymbol{H}\boldsymbol{Z}(n)\|_2 \le \|\boldsymbol{H}\|_2\,\|\boldsymbol{Z}(n)\|_2$, we can get

$$|\lambda_{max}(\boldsymbol{H}\boldsymbol{Z}(n))| \le \|\mathbf{\Theta}\|_2\,|\lambda_{max}(\boldsymbol{Z}(n))|. \tag{48}$$

From the definition of $\mathbf{\Theta}$, we have $\|\mathbf{\Theta}\|_2 = 1$, which yields

$$|\lambda_{max}(\boldsymbol{H}\boldsymbol{Z}(n))| \le |\lambda_{max}(\boldsymbol{Z}(n))|. \tag{49}$$

If the DPMCC$_{adapt}$ is stable, we requires

$$|\lambda_{max}(\boldsymbol{Z}(n))| < 1. \tag{50}$$

If the step size satisfies (50), then, we get

$$0 < E[\boldsymbol{T}(n)] < \frac{1}{\lambda_{max}(\boldsymbol{G}(n)\boldsymbol{R_U})}. \tag{51}$$

Then, we get

$$0 < E[\rho_k(n)] < \frac{1}{\lambda_{max}(\boldsymbol{G}(n)\boldsymbol{R_U})}. \tag{52}$$

As $\rho_k(n) = \mu_k r_k(n)$, we can go further to obtain

$$0 < \mu_k < \frac{1}{\lambda_{max}(\boldsymbol{G}(n)\boldsymbol{R_U})E[r_k(n)]}. \tag{53}$$

B. MEAN SQUARE TRANSIENT ANALYSIS

The mean square performance of the DPMCC$_{adapt}$ algorithm is studied herein. We calculate the weighted norm of (42) and take the expectation to get

$$\begin{aligned}
&E[\left\|\tilde{\boldsymbol{h}}(n)\right\|^2_{\boldsymbol{\Sigma}}] \\
&=E[||\boldsymbol{H}[\boldsymbol{I}_{NM} - \boldsymbol{T}(n)\boldsymbol{G}(n)\boldsymbol{U}^T(n)\boldsymbol{U}(n)]\tilde{\boldsymbol{h}}(n-1) - \\
&\quad \boldsymbol{H}\boldsymbol{T}(n)\boldsymbol{G}(n)\boldsymbol{U}^T(n)\boldsymbol{v}(n)||^2_{\boldsymbol{\Sigma}}] \\
&=E[||\tilde{\boldsymbol{h}}(n-1)||^2_{\boldsymbol{\Sigma}'}] + \\
&\quad E[\boldsymbol{v}^T(n)\boldsymbol{H}^T\boldsymbol{G}(n)\boldsymbol{T}(n)\boldsymbol{\Sigma}\boldsymbol{T}(n)\boldsymbol{G}(n)\boldsymbol{H}\boldsymbol{v}(n)],
\end{aligned} \tag{54}$$

where $\boldsymbol{\Sigma}_{NM\times NM}$ is a random matrix and

$$\begin{aligned}
\boldsymbol{\Sigma}' =& \boldsymbol{H}^T\boldsymbol{\Sigma}\boldsymbol{H} - \boldsymbol{H}^T\boldsymbol{\Sigma}\boldsymbol{T}(n)\boldsymbol{G}(n)\boldsymbol{U}^T(n)\boldsymbol{U}(n)\boldsymbol{H} \\
&-\boldsymbol{H}^T\boldsymbol{U}^T(n)\boldsymbol{U}(n)\boldsymbol{G}(n)\boldsymbol{T}(n)\boldsymbol{\Sigma}\boldsymbol{H} \\
&+\boldsymbol{H}^T\boldsymbol{U}^T(n)\boldsymbol{U}(n)\boldsymbol{G}(n)\boldsymbol{T}(n)\boldsymbol{\Sigma} \\
&\quad \boldsymbol{T}(n)\boldsymbol{G}(n)\boldsymbol{U}^T(n)\boldsymbol{U}(n)\boldsymbol{H}.
\end{aligned} \tag{55}$$

Since $\boldsymbol{\Sigma}'$ is a random matrix, we can replace it with its mean value (a deterministic matrix $\boldsymbol{\Sigma}^* = E[\boldsymbol{\Sigma}']$) [51]. Thus, we get

$$\begin{aligned}
&E[\left\|\tilde{\boldsymbol{h}}(n)\right\|^2_{\boldsymbol{\Sigma}}] \\
&=E[||\tilde{\boldsymbol{h}}(n-1)||^2_{\boldsymbol{\Sigma}^*}] + \\
&\quad E[\boldsymbol{v}^T(n)\boldsymbol{H}^T\boldsymbol{G}(n)\boldsymbol{T}(n)\boldsymbol{\Sigma}\boldsymbol{T}(n)\boldsymbol{G}(n)\boldsymbol{H}\boldsymbol{v}(n)],
\end{aligned} \tag{56}$$

where

$$\begin{aligned}
\boldsymbol{\Sigma}^* = E[\boldsymbol{\Sigma}'] =& \boldsymbol{H}^T\boldsymbol{\Sigma}\boldsymbol{H} - E[\boldsymbol{H}^T \\
&\boldsymbol{\Sigma}\boldsymbol{T}(n)\boldsymbol{G}(n)\boldsymbol{U}^T(n)\boldsymbol{U}(n)\boldsymbol{H}] \\
&-\boldsymbol{H}^T\boldsymbol{G}(n)E[\boldsymbol{U}^T(n)\boldsymbol{U}(n)]E[\boldsymbol{T}(n)]\boldsymbol{\Sigma}\boldsymbol{H} \\
&+\boldsymbol{H}^T E[\boldsymbol{U}^T(n)\boldsymbol{U}(n)\boldsymbol{G}(n)\boldsymbol{T}(n)\boldsymbol{\Sigma} \\
&\boldsymbol{T}(n)\boldsymbol{G}(n)\boldsymbol{U}^T(n)\boldsymbol{U}(n)]\boldsymbol{H}.
\end{aligned} \tag{57}$$

For further analysis, the auto-correlation matrix is divided into

$$\boldsymbol{R_U} = E[\boldsymbol{U}^T(n)\boldsymbol{U}(n)] = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^T, \tag{58}$$

where $\boldsymbol{\Lambda}$ is a diagonal matrix containing eigenvalues of $\boldsymbol{R_U}$, and $\boldsymbol{Q}$ is a matrix containing the eigenvectors corresponding to these eigenvalues. According to this decomposition, we define new transformed variables:

$$\begin{aligned}
&\bar{\boldsymbol{h}}(n) = \boldsymbol{Q}^T\tilde{\boldsymbol{h}}(n), \bar{\boldsymbol{U}}(n) = \boldsymbol{U}(n)\boldsymbol{Q}, \bar{\boldsymbol{H}} = \boldsymbol{Q}^T\boldsymbol{H}\boldsymbol{Q}, \\
&\bar{\boldsymbol{G}}_{\boldsymbol{u}_k}(n) = \boldsymbol{Q}^T\boldsymbol{G}(n)\boldsymbol{Q} = \boldsymbol{G}(n), \bar{\boldsymbol{\Sigma}} = \boldsymbol{Q}^T\boldsymbol{\Sigma}\boldsymbol{Q}, \\
&\bar{\boldsymbol{\Sigma}}^* = \boldsymbol{Q}^T\boldsymbol{\Sigma}^*\boldsymbol{Q}, \bar{\boldsymbol{T}}(n) = \boldsymbol{Q}^T\boldsymbol{T}(n)\boldsymbol{Q} = \boldsymbol{T}(n),
\end{aligned} \tag{59}$$

where the input of each node is regarded as independent of each other. As $\boldsymbol{G}(n)$ and $\boldsymbol{T}(n)$ are diagonal matrices in (59), $\bar{\boldsymbol{T}}(n) = \boldsymbol{T}(n), \bar{\boldsymbol{G}}(n) = \boldsymbol{G}(n)$. Then, (56) is redefined as

$$\begin{aligned}
E[\left\|\bar{\boldsymbol{h}}(n)\right\|^2_{\bar{\boldsymbol{\Sigma}}}] =& E[\left\|\bar{\boldsymbol{h}}(n-1)\right\|^2_{\bar{\boldsymbol{\Sigma}}^*}] \\
&+ E[\boldsymbol{v}^T(n)\bar{\boldsymbol{H}}^T\bar{\boldsymbol{U}}^T(n)\boldsymbol{G}(n)\boldsymbol{T}(n)\bar{\boldsymbol{\Sigma}} \\
&\boldsymbol{T}(n)\boldsymbol{G}(n)\bar{\boldsymbol{U}}(n)\bar{\boldsymbol{H}}\boldsymbol{v}(n)],
\end{aligned} \tag{60}$$

where

$$\begin{aligned}
\bar{\boldsymbol{\Sigma}}^* =& \bar{\boldsymbol{H}}^T\bar{\boldsymbol{\Sigma}}\bar{\boldsymbol{H}} - \bar{\boldsymbol{H}}^T\bar{\boldsymbol{\Sigma}}E[\boldsymbol{T}(n)]\boldsymbol{G}(n) \\
&E[\bar{\boldsymbol{U}}^T(n)\bar{\boldsymbol{U}}(n)]\bar{\boldsymbol{H}} \\
&-E[\bar{\boldsymbol{H}}^T\bar{\boldsymbol{U}}^T(n)\bar{\boldsymbol{U}}(n)\boldsymbol{T}(n)\boldsymbol{G}(n)\bar{\boldsymbol{\Sigma}}\bar{\boldsymbol{H}}] \\
&+E[\bar{\boldsymbol{H}}^T\bar{\boldsymbol{U}}^T(n)\bar{\boldsymbol{U}}(n)\boldsymbol{G}(n)\boldsymbol{T}(n)\bar{\boldsymbol{\Sigma}} \\
&\boldsymbol{T}(n)\boldsymbol{G}(n)\bar{\boldsymbol{U}}^T(n)\bar{\boldsymbol{U}}(n)\bar{\boldsymbol{H}}],
\end{aligned} \tag{61}$$

and we have $E[\bar{\boldsymbol{U}}^T(n)\bar{\boldsymbol{U}}(n)] = \boldsymbol{\Lambda}$. The freedom for selecting $\boldsymbol{\Sigma}$ enables us to characterize the MSD performance of the network [51].

Definition 1: $\boldsymbol{\Sigma}$ can be defined as

$$\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} & \cdots & \boldsymbol{\Sigma}_{1N} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} & \cdots & \boldsymbol{\Sigma}_{2N} \\ \vdots & \vdots & \cdots & \vdots \\ \boldsymbol{\Sigma}_{N1} & \boldsymbol{\Sigma}_{N2} & \cdots & \boldsymbol{\Sigma}_{NN} \end{bmatrix}, \tag{62}$$

where $\boldsymbol{\Sigma}_{kl}$ is an $M \times M$ matrix, and

$$\boldsymbol{\Sigma}_l = col\{\boldsymbol{\Sigma}_{1l}, ..., \boldsymbol{\Sigma}_{Nl}\}, l \in \{1, ..., N\}, \tag{63}$$

$$\boldsymbol{\Sigma}^c = col\{\boldsymbol{\Sigma}_1, ..., \boldsymbol{\Sigma}_N\}. \tag{64}$$

Converting $\boldsymbol{\Sigma}_{kl}$ to a column vector using the $bvec$ operator, we have

$$\boldsymbol{\vartheta}_{kl} = bvec\{\boldsymbol{\Sigma}_{kl}\}. \tag{65}$$

Then, we have

$$\boldsymbol{\vartheta}_l = col\{\boldsymbol{\vartheta}_{1l}, ..., \boldsymbol{\vartheta}_{Nl}\}, \tag{66}$$

$$\boldsymbol{\vartheta} = col\{\boldsymbol{\vartheta}_1, ..., \boldsymbol{\vartheta}_N\}. \tag{67}$$

The matrix $\boldsymbol{\Sigma}$ is transformed into a vector by means of the $bevc$ operator $\boldsymbol{\vartheta} = bevc\{\boldsymbol{\Sigma}\}$.

The *block Kroncker product* of two block matrices, $\mathbb{C}$ and $\mathbb{D}$, is defined as $\mathbb{C} \odot \mathbb{D}$, and its kl-block is

$$[\mathbb{C} \odot \mathbb{D}]_{kl} = \begin{bmatrix} \mathbb{C}_{kl} \otimes \mathbb{D}_{11} & \cdots & \mathbb{C}_{kl} \otimes \mathbb{D}_{1N} \\ \vdots & \vdots & \cdots \\ \mathbb{C}_{kl} \otimes \mathbb{D}_{N1} & \cdots & \mathbb{C}_{kl} \otimes \mathbb{D}_{NN} \end{bmatrix} \tag{68}$$

for $k,l \in 1,...,N$. $bevc$ operator is used to get $bevc\{\bar{\boldsymbol{\Sigma}}\} = \bar{\boldsymbol{\vartheta}}$.

Next, we will analyze the following items

$$E[\boldsymbol{v}^T(n)\bar{\boldsymbol{H}}^T\bar{\boldsymbol{U}}^T(n)\boldsymbol{G}(n)\boldsymbol{T}(n)\bar{\boldsymbol{\Sigma}}\boldsymbol{T}(n)\boldsymbol{G}(n)\bar{\boldsymbol{U}}(n)\bar{\boldsymbol{H}}\boldsymbol{v}(n)], \tag{69}$$

$$\bar{\boldsymbol{H}}^T\bar{\boldsymbol{\Sigma}}\bar{\boldsymbol{H}}, \tag{70}$$

$$\bar{\boldsymbol{H}}^T\bar{\boldsymbol{\Sigma}}E[\boldsymbol{T}(n)]\boldsymbol{G}(n)E[\bar{\boldsymbol{U}}^T(n)\bar{\boldsymbol{U}}(n)]\bar{\boldsymbol{H}} \tag{71}$$

$$E[\bar{\boldsymbol{H}}^T\bar{\boldsymbol{U}}^T(n)\bar{\boldsymbol{U}}(n)\boldsymbol{T}(n)\boldsymbol{G}(n)\bar{\boldsymbol{\Sigma}}\bar{\boldsymbol{H}}], \tag{72}$$

$$E[\bar{\boldsymbol{H}}^T\bar{\boldsymbol{U}}^T(n)\bar{\boldsymbol{U}}(n)\boldsymbol{G}(n)\boldsymbol{T}(n)\bar{\boldsymbol{\Sigma}}\boldsymbol{T}(n)\boldsymbol{G}(n)\bar{\boldsymbol{U}}^T(n)\bar{\boldsymbol{U}}(n)\bar{\boldsymbol{H}}]. \tag{73}$$

The second item on the right hand side of (60) is written as

$$\begin{aligned}&E[\boldsymbol{v}^T(n)\bar{\boldsymbol{H}}^T\bar{\boldsymbol{U}}^T(n)\boldsymbol{G}(n)\boldsymbol{T}(n)\bar{\boldsymbol{\Sigma}}\\ &\quad\boldsymbol{T}(n)\boldsymbol{G}(n)\bar{\boldsymbol{U}}(n)\bar{\boldsymbol{H}}\boldsymbol{v}(n)]\\ &=E[\bar{\boldsymbol{H}}^T\boldsymbol{G}(n)\boldsymbol{v}(n)\boldsymbol{v}^T(n)\boldsymbol{G}(n)\bar{\boldsymbol{H}}]\\ &\quad E[\bar{\boldsymbol{U}}^T(n)\boldsymbol{T}(n)\bar{\boldsymbol{\Sigma}}\boldsymbol{T}(n)\bar{\boldsymbol{U}}(n)].\end{aligned} \tag{74}$$

Divide (74) into two parts and vectorize them separately. The first part is $E[\bar{\boldsymbol{H}}^T\boldsymbol{G}(n)\boldsymbol{v}(n)\boldsymbol{v}^T(n)\boldsymbol{G}(n)\bar{\boldsymbol{H}}]$. Vectorizing $E[\bar{\boldsymbol{H}}^T\boldsymbol{G}(n)\boldsymbol{v}(n)\boldsymbol{v}^T(n)\boldsymbol{G}(n)\bar{\boldsymbol{H}}]$, we get

$$\begin{aligned}&bevc\{E[\bar{\boldsymbol{H}}^T\boldsymbol{G}(n)\boldsymbol{v}(n)\boldsymbol{v}^T(n)\boldsymbol{G}(n)\bar{\boldsymbol{H}}]\}\\ &\quad=\bar{\boldsymbol{H}}^T\odot\bar{\boldsymbol{H}}^T bevc\{E[\boldsymbol{G}(n)\boldsymbol{v}(n)\boldsymbol{v}^T(n)\boldsymbol{G}(n)]\}\end{aligned} \tag{75}$$

where

$$\boldsymbol{\chi}^T = (\bar{\boldsymbol{H}}^T\odot\bar{\boldsymbol{H}}^T)bevc\{E[\boldsymbol{G}(n)\boldsymbol{v}(n)\boldsymbol{v}^T(n)\boldsymbol{G}(n)]\}. \tag{76}$$

The second part is $\boldsymbol{B} = E[\bar{\boldsymbol{U}}^T(n)\boldsymbol{T}(n)\bar{\boldsymbol{\Sigma}}\boldsymbol{T}(n)\bar{\boldsymbol{U}}(n)]$ in (74).

Definition 2: $\boldsymbol{B}$ is defined as

$$\boldsymbol{B} = [\boldsymbol{B}_1, \boldsymbol{B}_2, ..., \boldsymbol{B}_N], \tag{77}$$

and its l-th column is

$$\boldsymbol{B}_l = col\{\boldsymbol{B}_{1,l}, \boldsymbol{B}_{2,l}, ..., \boldsymbol{B}_{k,l}, ..., \boldsymbol{B}_{N,l}\}. \tag{78}$$

As a result, M-dimensional $\boldsymbol{B}_{k,l}$ is obtained

$$\boldsymbol{B}_{k,l} = \begin{cases}\rho_k^2 Tr(\boldsymbol{\Lambda}_k\bar{\boldsymbol{\Sigma}}_{kk}), & k=l\\ 0, & k\neq l,\end{cases} \tag{79}$$

and we vectorize $\boldsymbol{B}$ as

$$\Im = bevc\{\boldsymbol{B}\} = col\{\boldsymbol{b}_1, \boldsymbol{b}_2, ..., \boldsymbol{b}_N\}, \tag{80}$$

and we define

$$\boldsymbol{b}_{k,l} = bevc\{\boldsymbol{B}_{k,l}\}, \tag{81}$$

$$\boldsymbol{b}_{k,l} = \begin{cases}bvec\{\boldsymbol{I}_M\}\rho_k^2\boldsymbol{\lambda}_k^T\bar{\boldsymbol{\vartheta}}_{kk}, & k=l\\ 0, & k\neq l.\end{cases} \tag{82}$$

The column l of $\boldsymbol{b}$ is

$$\boldsymbol{b}_l = col\{\boldsymbol{b}_{1,l}, \boldsymbol{b}_{2,l}, ..., \boldsymbol{b}_{l,l}, ..., \boldsymbol{b}_{N,l}\} \tag{83}$$

based on (82) and (83), and we have

$$\boldsymbol{b}_l = col\{0\bar{\boldsymbol{\vartheta}}_{1l}, 0\bar{\boldsymbol{\vartheta}}_{2l}, ..., bevc\{\boldsymbol{I}_M\rho_l^2\boldsymbol{\lambda}_l^2\bar{\boldsymbol{\vartheta}}_{ll}\}, ..., 0\bar{\boldsymbol{\vartheta}}_{Nl}\}, \tag{84}$$

and

$$\begin{aligned}&bevc\{E[\boldsymbol{v}^T(n)\bar{\boldsymbol{H}}^T\bar{\boldsymbol{U}}^T(n)\boldsymbol{G}(n)\boldsymbol{T}(n)\bar{\boldsymbol{\Sigma}}\\ &\quad\boldsymbol{T}(n)\boldsymbol{G}(n)\bar{\boldsymbol{U}}(n)\bar{\boldsymbol{H}}\boldsymbol{v}(n)]\}\\ &=\boldsymbol{\chi}^T\Im\bar{\boldsymbol{\vartheta}}.\end{aligned} \tag{85}$$

For any three matrices $\mathbb{C}, \mathbb{D}, \bar{\boldsymbol{\Sigma}}$, according to [52], [53], we have

$$bevc\{\mathbb{C}\bar{\boldsymbol{\Sigma}}\mathbb{D}\} = (\mathbb{D}^T\odot\mathbb{C})bevc\{\bar{\boldsymbol{\Sigma}}\} = (\mathbb{D}^T\odot\mathbb{C})\bar{\boldsymbol{\vartheta}}. \tag{86}$$

Next, we vectorized each term of equation (61), and the first term can be written as

$$bevc\{\bar{\boldsymbol{H}}^T\bar{\boldsymbol{\Sigma}}\bar{\boldsymbol{H}}\} = (\bar{\boldsymbol{H}}^T\odot\bar{\boldsymbol{H}}^T)\bar{\boldsymbol{\vartheta}}. \tag{87}$$

For the second term, we have

$$\begin{aligned}&bevc\{E[\bar{\boldsymbol{H}}^T\bar{\boldsymbol{\Sigma}}\boldsymbol{T}(n)\boldsymbol{G}(n)\bar{\boldsymbol{U}}^T(n)\bar{\boldsymbol{U}}(n)\bar{\boldsymbol{H}}]\}\\ &=(\bar{\boldsymbol{H}}^T\odot\bar{\boldsymbol{H}}^T)bevc\{E[\boldsymbol{I}_{NM}\bar{\boldsymbol{\Sigma}}\boldsymbol{T}(n)\boldsymbol{G}(n)\bar{\boldsymbol{U}}^T(n)\bar{\boldsymbol{U}}(n)]\}\\ &=(\bar{\boldsymbol{H}}^T\odot\bar{\boldsymbol{H}}^T)(\boldsymbol{\Lambda}\odot\boldsymbol{I}_{NM})\\ &\quad bevc\{E[\boldsymbol{I}_{NM}\bar{\boldsymbol{\Sigma}}\boldsymbol{T}(n)\boldsymbol{G}(n)]\}\\ &=(\bar{\boldsymbol{H}}^T\odot\bar{\boldsymbol{H}}^T)(\boldsymbol{\Lambda}\odot\boldsymbol{I}_{NM})\\ &\quad(E[\boldsymbol{T}(n)]\boldsymbol{G}(n)\odot\boldsymbol{I}_{NM})bevc\{\bar{\boldsymbol{\Sigma}}\}\\ &=(\bar{\boldsymbol{H}}^T\odot\bar{\boldsymbol{H}}^T)(\boldsymbol{\Lambda}\odot\boldsymbol{I}_{NM})\\ &\quad(E[\boldsymbol{T}(n)]\boldsymbol{G}(n)\odot\boldsymbol{I}_{NM})\bar{\boldsymbol{\vartheta}}.\end{aligned} \tag{88}$$

As for the third term, we have

$$\begin{aligned}&bevc\{E[\bar{\boldsymbol{H}}^T\bar{\boldsymbol{U}}^T(n)\bar{\boldsymbol{U}}(n)\boldsymbol{T}(n)\boldsymbol{G}(n)\bar{\boldsymbol{\Sigma}}\bar{\boldsymbol{H}}]\}\\ &\quad=(\bar{\boldsymbol{H}}^T\odot\bar{\boldsymbol{H}}^T)bevc\{\boldsymbol{\Lambda}E[\boldsymbol{T}(n)]\boldsymbol{G}(n)\bar{\boldsymbol{\Sigma}}\boldsymbol{I}_{NM}\}\\ &\quad=(\bar{\boldsymbol{H}}^T\odot\bar{\boldsymbol{H}}^T)(\boldsymbol{I}_{NM}\odot\boldsymbol{\Lambda})\\ &bevc\{E[\boldsymbol{T}(n)]\boldsymbol{G}(n)\bar{\boldsymbol{\Sigma}}\boldsymbol{I}_{NM}\}\\ &\quad=(\bar{\boldsymbol{H}}^T\odot\bar{\boldsymbol{H}}^T)(\boldsymbol{I}_{NM}\odot\boldsymbol{\Lambda})\\ &(\boldsymbol{I}_{NM}\odot E[\boldsymbol{T}(n)]\boldsymbol{G}(n))\bar{\boldsymbol{\vartheta}}.\end{aligned} \tag{89}$$

For the fourth term, we have

$$\begin{aligned}&bevc\{E[\bar{\boldsymbol{H}}^T\bar{\boldsymbol{U}}^T(n)\bar{\boldsymbol{U}}(n)\boldsymbol{G}(n)\boldsymbol{T}(n)\bar{\boldsymbol{\Sigma}}\\ &\quad\boldsymbol{T}(n)\boldsymbol{G}(n)\bar{\boldsymbol{U}}^T(n)\bar{\boldsymbol{U}}(n)\bar{\boldsymbol{H}}]\}\\ &\quad=(\bar{\boldsymbol{H}}^T\odot\bar{\boldsymbol{H}}^T)bevc\{E[\bar{\boldsymbol{U}}^T(n)\bar{\boldsymbol{U}}(n)\boldsymbol{G}(n)\boldsymbol{T}(n)\bar{\boldsymbol{\Sigma}}\\ &\quad\boldsymbol{T}(n)\boldsymbol{G}(n)\bar{\boldsymbol{U}}^T(n)\bar{\boldsymbol{U}}(n)]\}\\ &\quad=E[\boldsymbol{T}(n)]\boldsymbol{G}(n)\odot\boldsymbol{G}(n)E[\boldsymbol{T}(n)]\\ &\quad bevc\{E[\bar{\boldsymbol{U}}^T(n)\bar{\boldsymbol{U}}(n)\bar{\boldsymbol{\Sigma}}\bar{\boldsymbol{U}}^T(n)\bar{\boldsymbol{U}}(n)]\}.\end{aligned} \tag{90}$$

Definition 3: $\boldsymbol{F}$ is defined as

$$\begin{aligned}\boldsymbol{F} =&E[\bar{\boldsymbol{U}}^T(n)\bar{\boldsymbol{U}}(n)\bar{\boldsymbol{\Sigma}},\\ &\bar{\boldsymbol{U}}^T(n)\bar{\boldsymbol{U}}(n)],\end{aligned} \tag{91}$$

where

$$\boldsymbol{F} = diag\{\boldsymbol{F}_1, \boldsymbol{F}_2, ..., \boldsymbol{F}_N\}, \tag{92}$$

$$\boldsymbol{F}_l = \{\boldsymbol{F}_{1l}, \boldsymbol{F}_{2l}, ..., \boldsymbol{F}_{Nl}\} \quad (l \in 1, ..., N). \tag{93}$$

According to [54], the block matrix $\boldsymbol{F}_{kl}$ is estimated by

$$\boldsymbol{F}_{kl} = \begin{cases} [\boldsymbol{\Lambda}_k Tr(\boldsymbol{\Lambda}_k \bar{\boldsymbol{\Sigma}}_{kk}) & \\ +\xi \boldsymbol{\Lambda}_k \bar{\boldsymbol{\Sigma}}_{kk} \boldsymbol{\lambda}_k] \boldsymbol{\Lambda}_k \bar{\boldsymbol{\Sigma}}_{kl} \boldsymbol{\Lambda}_l, & k = l, \\ 0, & k \neq l \end{cases} \tag{94}$$

where $\boldsymbol{\lambda}_k = bvec\{\boldsymbol{\Lambda}_k\}$.

From the above analysis

$$\begin{aligned} bvec\{\bar{\boldsymbol{\Sigma}}^*\} = &(\bar{\boldsymbol{H}}^T \odot \bar{\boldsymbol{H}}^T)\{\boldsymbol{I}_{N^2M^2} - (\boldsymbol{\Lambda} \odot \boldsymbol{I}_{NM}) \\ &(E[\boldsymbol{T}(n)]\boldsymbol{G}(n) \odot \boldsymbol{I}_{NM}) \\ &- (\boldsymbol{I}_{NM} \odot \boldsymbol{\Lambda})(\boldsymbol{I}_{NM} \odot E[\boldsymbol{T}(n)]\boldsymbol{G}(n)) + \\ &(E[\boldsymbol{T}(n)]\boldsymbol{G}(n) \odot \boldsymbol{G}(n)E[\boldsymbol{T}(n)])\boldsymbol{F}\}\bar{\boldsymbol{\vartheta}}, \end{aligned} \tag{95}$$

and substituting (95) into (61), the mean square behavior of adaptive networks is described recursively as follows

$$E[\left\|\tilde{\boldsymbol{h}}(\boldsymbol{n})\right\|^2_{\bar{\boldsymbol{\vartheta}}}] = E[\left\|\tilde{\boldsymbol{h}}(\boldsymbol{n}-\boldsymbol{1})\right\|^2_{\bar{\boldsymbol{A}}\bar{\boldsymbol{\vartheta}}}] + \boldsymbol{\chi}^{\boldsymbol{T}} \Im \bar{\boldsymbol{\vartheta}}, \tag{96}$$

where

$$\begin{aligned} \bar{\boldsymbol{A}} = &(\bar{\boldsymbol{H}}^T \odot \bar{\boldsymbol{H}}^T)\{\boldsymbol{I}_{N^2M^2} - (\boldsymbol{\Lambda} \odot \boldsymbol{I}_{NM}) \\ &(E[\boldsymbol{T}(n)]\boldsymbol{G}(n) \odot \boldsymbol{I}_{NM}) \\ &- (\boldsymbol{I}_{NM} \odot \boldsymbol{\Lambda})(\boldsymbol{I}_{NM} \odot E[\boldsymbol{T}(n)]\boldsymbol{G}(n)) + \\ &(E[\boldsymbol{T}(n)]\boldsymbol{G}(n) \odot \boldsymbol{G}(n)E[\boldsymbol{T}(n)])\boldsymbol{F}\}. \end{aligned} \tag{97}$$

According to the above analysis, the recursive equation of the DPMCC$_{adapt}$ algorithm is derived

$$\begin{aligned} &E[\|\bar{\boldsymbol{h}}(n)\|^2_{\bar{\boldsymbol{\vartheta}}}] = E[\|\bar{\boldsymbol{h}}(n-1)\|^2_{\bar{\boldsymbol{A}}(1)\bar{\boldsymbol{\vartheta}}}] + \boldsymbol{\chi}^T \Im \bar{\boldsymbol{\vartheta}} \bar{\boldsymbol{A}}(0), \\ &E[\|\bar{\boldsymbol{h}}(n-1)\|^2_{\bar{\boldsymbol{A}}(1)\bar{\boldsymbol{\vartheta}}}] = E[\|\bar{\boldsymbol{h}}(n-1)\|^2_{\bar{\boldsymbol{A}}(2)\bar{\boldsymbol{\vartheta}}}] + \boldsymbol{\chi}^T \Im \bar{\boldsymbol{\vartheta}} \bar{\boldsymbol{A}}(1), \\ &\dots \\ &E[\|\bar{\boldsymbol{h}}(n-1)\|^2_{\bar{\boldsymbol{A}}(n)\bar{\boldsymbol{\vartheta}}}] = \|\bar{\boldsymbol{h}}(n-1)\|^2_{\bar{\boldsymbol{A}}(n+1)\bar{\boldsymbol{\vartheta}}} + \boldsymbol{\chi}^T \Im \bar{\boldsymbol{\vartheta}} \bar{\boldsymbol{A}}(n), \end{aligned} \tag{98}$$

Based on (98), the following results are obtained

$$\begin{aligned} E[\|\bar{\boldsymbol{h}}(n)\|^2_{\bar{\boldsymbol{A}}(n)\bar{\boldsymbol{\vartheta}}}] = &\|\bar{\boldsymbol{h}}_0\|^2_{\bar{\boldsymbol{A}}(n+1)\bar{\boldsymbol{\vartheta}}} \\ &+ \boldsymbol{\chi}^T \Im \bar{\boldsymbol{\vartheta}} \sum_{z=0}^{n} \bar{\boldsymbol{A}}(z), \end{aligned} \tag{99}$$

which in turn motivates the following recursion [51], the network mean square deviation (MSD) evolve as follows:

$$\begin{aligned} E[\|\bar{\boldsymbol{h}}(n)\|^2_{\bar{\boldsymbol{\vartheta}}}] = &E[\|\bar{\boldsymbol{h}}(n-1)\|^2_{\bar{\boldsymbol{\vartheta}}}] + \boldsymbol{\chi}^T \Im \bar{\boldsymbol{A}}(n) \bar{\boldsymbol{\vartheta}} \\ &- \|\bar{\boldsymbol{h}}_0\|^2_{\bar{\boldsymbol{A}}(n)(I-\bar{\boldsymbol{A}})\bar{\boldsymbol{\vartheta}}}. \end{aligned} \tag{100}$$

Making $\eta(n) = (1/N)E[\|\bar{\boldsymbol{h}}(n)\|^2]$, $\bar{\boldsymbol{\vartheta}} = \frac{1}{N} bvec\{\boldsymbol{I}_{NM}\} = \boldsymbol{q}_\eta$, we have

$$\eta(n) = \eta(n-1) + \boldsymbol{\chi}^T \Im \bar{\boldsymbol{A}}(n) \boldsymbol{q}_\eta - \|\bar{\boldsymbol{h}}_0\|^2_{\bar{\boldsymbol{A}}(n)(\boldsymbol{I}-\bar{\boldsymbol{A}})\boldsymbol{q}_\eta}. \tag{101}$$

When the convergence reaches its steady-state, the global MSD is expressed as [51]

$$\eta(n) = \frac{1}{N} E[\left\|\bar{\boldsymbol{h}}(n-1)\right\|^2], \tag{102}$$

as $n \to \infty$, (96) leads to

$$E[\|\bar{\boldsymbol{h}}(\infty)\|^2_{[\boldsymbol{I}-\bar{\boldsymbol{A}}]\bar{\boldsymbol{\vartheta}}}] = \boldsymbol{\chi}^{\boldsymbol{T}} \Im \bar{\boldsymbol{\vartheta}}, \tag{103}$$

which is concluded as

$$\eta(n) = \frac{1}{N} \boldsymbol{\chi}^T \Im (\boldsymbol{I} - \bar{\boldsymbol{A}})^{-1} bevc\{\boldsymbol{I}_{NM}\}. \tag{104}$$

Till now, the theoretical MSD analysis of the entire network is obtained.

C. COMPUTATIONAL COMPLEXITY

In Table 2, the computational complexity of the MCC, A-MCC (Adaptive Kernel Width MCC), S-MCC (Switch Kernel Width MCC), FxRMC (Filtered-x Recursive Maximum Correntropy), VKW-MCC (Variable Kernel Width MCC), and MCC$_{adapt}$ are compared. We can see from the Table 2, MCC$_{adapt}$ calculates more additions than the MCC, S-MCC, with the same computational complexity as A-MCC, but less than FxMCC and VKW-MCC. Additionally, MCC$_{adapt}$ calculates multiplication less than S-MCC and FxMCC.

TABLE 2. Comparison of computational complexity

Algorithms	Multiplication	Addition
MCC	2M+6	2M
A-MCC	2M+6	2M+1
S-MCC	2M+8	2M
FxRMC	2M+4	2M+11
VKW-MCC	2M+9	2M+2
MCC$_{adapt}$	2M+7	2M+1

Table 3 compares the computational complexities of the DLMS, DLMP, DSIGN, DNSIGN, DMCC and DMCC$_{adapt}$. It can be seen from the Table 3, DMCC$_{adapt}$ calculates more addition than the DLMS and DMCC, but has the same computational complexity to the DLMP and DSIGN. However, it is simpler than the DNSIGN. DMCC$_{adapt}$ calculates more multiplication than the DLMS, DLMP, DSIGN and DMCC, but it is far less than the DNSIGN.

TABLE 3. Comparison of computational complexities in non-sparse systems

Algorithms	Multiplication	Addition
DLMS	7M+1	6M
DLMP	7M+2	6M+1
DSIGN	7M+1	6M+1
DNSIGN	8M+2	7M
DMCC	7M+6	6M
DMCC$_{adapt}$	7M+7	6M+1

Table 4 shows the comparison of computational complexity of DLMS, DMCC, DMCC$_{adapt}$, DPLMS, DPMCC and DPMCC$_{adapt}$ algorithms in sparse system. It can be seen from the Table 4 that the steady state performance of the algorithm is improved when combined with the proportional coefficient algorithm, but the computational complexity of

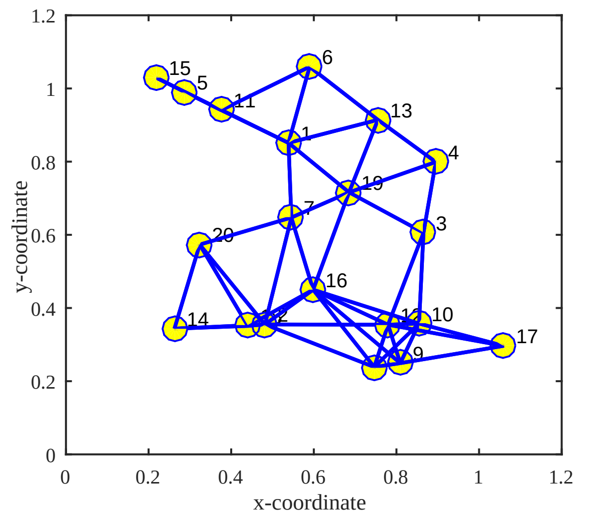

FIGURE 1. Diffusion network topology with 20 nodes within a squared area of $[0, 1.2] \times [0, 1.2]$.

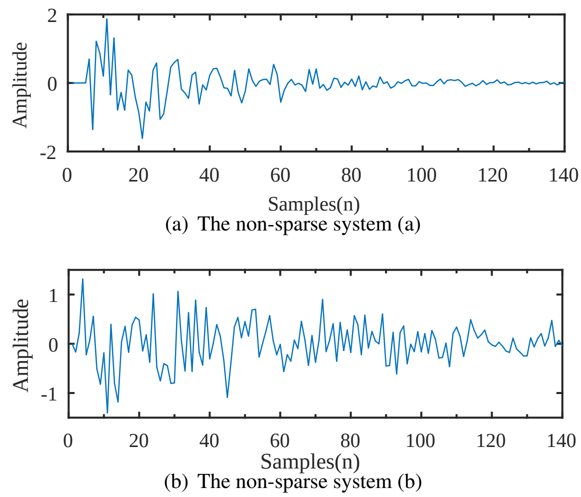

FIGURE 2. Non-sparse system impulse responses.

the algorithm also increases correspondingly. DPMCC$_{adapt}$ calculates addition and multiplication times more than other algorithms.

TABLE 4. Comparison of computational complexity of various diffusion algorithms in sparse system

Algorithms	Multiplication	Addition
DLMS	7M+1	6M
DMCC	7M+6	6M
DMCC$_{adapt}$	7M+7	6M+1
DPLMS	11M+5	8M+1
DPMCC	11M+10	8M+1
DPMCC$_{adapt}$	11M+11	8M+2

V. SIMULATION RESULTS

In this section, we setup simulation experiments to prove the effectiveness of the constructed algorithms. As shown in Fig.1, the distributed network topology consists of $N = 20$ nodes, and the joint parameter $\{\alpha_{l,k}\}$ between nodes is obtained using Metropolis criterion [55]. In this paper, $\boldsymbol{u}_k(n)$ is Gaussian white noise with zero mean, the MSD at node k is presented by

$$MSD(n) = 10\log\frac{1}{N}\sum_{k=1}^{N}E[\|\boldsymbol{w}_{0,k} - \boldsymbol{w}_k(n)\|^2]. \quad (105)$$

We also assumed that the noise at each node is independent of the noise of other nodes. $v_k(n)$ at each node consists of $y_k(n)$ and $z_k(n)$, namely $v_k(n) = y_k(n) + z_k(n)$, where $y_k(n) \sim \mathcal{N}(0, \sigma_{y_k}^2)$ is Gaussian white noise with zero mean. $z_k(n)$ is Bernoulli impulsive noise whose model is given by $z_k(n) = \rho_k(n)\beta_k(n)$, where $\rho_k(n) \sim \mathcal{N}(0, \sigma_{\rho_k}^2)$, and $\beta_k(n)$ is Bernoulli noise with a probability density function of $P(\beta_k(n) = 1) = P_r$. In this paper, P_r represents the probability of pulse interference. The Signal-to-Noise Ratio (SNR) is defined by

$$SNR = 10log(\frac{\sigma_{d_k}^2}{\sigma_{y_k}^2}), \quad (106)$$

and the Signal-to-Interference Ratio (SIR) is defined by

$$SIR = 10log(\frac{\sigma_{d_k}^2}{\sigma_{z_k}^2}), \quad (107)$$

where $\sigma_{d_k}^2$ is the variance of $d_k(n)$, $\sigma_{z_k}^2$ is the variance of Bernoulli noise $z_k(n)$. Unless otherwise stated, the basic parameters for the experiments are set in Table 5.

TABLE 5. The basic simulation parameters

Parameters	Value
The length of filter	M=140
The number of trails	50
The iteration number of for single system	20000
The iteration number of double channels	40000
The Signal-to-Noise Ratio (SNR)	SNR=-20dB
The Signal-to-Interference Ratio (SIR)	SIR=-5dB

A. FOR NON-SPARSE SYSTEM

Here, behaviors of the MCC$_{adapt}$ is compared with MCC, A-MCC, S-MCC, FxRMC and VKW-MCC under 10% impulse noise. The probability of Bernoulli process occurrence is $P(\beta_k(n) = 1) = P_r = 0.1$. The kernel width $\sigma_0 = 2$ is used for MCC, A-MCC, S-MCC and MCC$_{adapt}$, and $k_\sigma = 20$ is employed in the VKW-MCC algorithm. The corresponding step size for MCC, A-MCC, S-MCC, FxMCC, VKW-MCC and MCC$_{adapt}$ are 0.1, 0.002, 0.15, 0.2, 0.1 and 0.2, respectively.

In Fig.3, we observe that the MCC performs the worst. The steady-state error of the VKW-MCC is better than A-MCC, S-MCC, and FxMCC. However, the computational complexity of these algorithm increases a lot. Also, we can see that

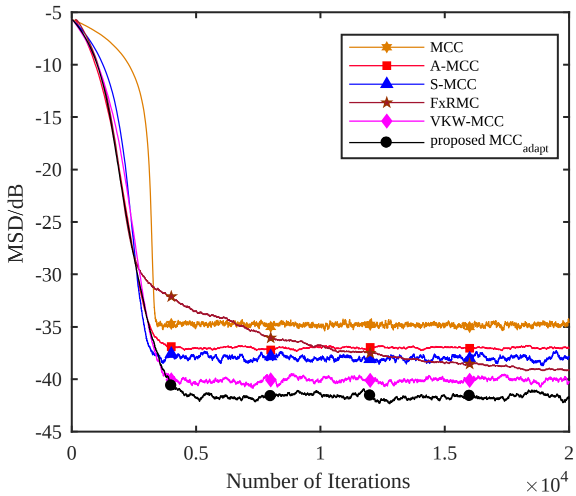

FIGURE 3. The comparison of steady state performance for the MCC, A-MCC, S-MCC, FxMCC, VKW-MCC and $\text{MCC}_{\text{adapt}}$ algorithms with 10% impulsive noise.

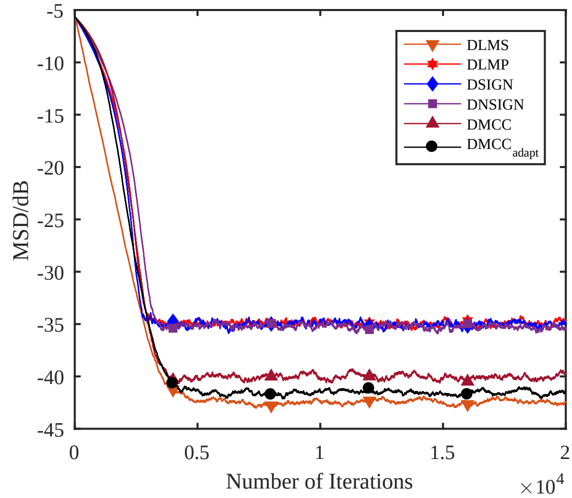

FIGURE 5. The MSD comparison of the DLMS, DLMP, DSIGN, DNSIGN, DMCC, $\text{DMCC}_{\text{adapt}}$ with Gaussian noise.

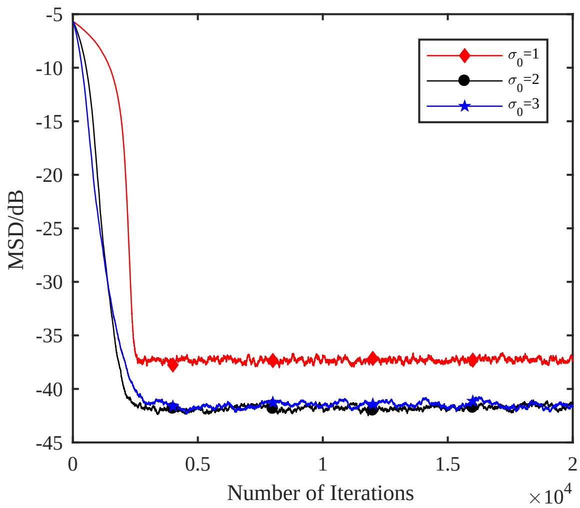

FIGURE 4. The convergence of the $\text{DMCC}_{\text{adapt}}$ algorithm with different σ_0.

the performance of the MCC_{adapt} algorithm is better than other algorithms because of kernel width adaption scheme.

Figure 4 is effect of parameter σ_0 on the DMCC_{adapt} algorithm in non-sparse system (a), where σ_0 are 1, 2, 3, and 10% impulse noise is considered in the system. The corresponding step size are 0.006, 0.015, and 0.025. When $\sigma_0 > 2$, its influence on the DMCC_{adapt} is not obvious. Hereafter, $\sigma_0 = 2$ is chosen.

The convergence for the DLMS, DLMP, DSIGN, DNSIGN, DMCC and DMCC_{adapt} under Gaussian noise is presented in Fig.5, where a non-sparse system (a) is considered and their step sizes are 0.006, 0.025, 0.01, 0.25, 0.05 and 0.08, respectively. p is 1 for DLMP, and the kernel size $\sigma_0 = 2$ for DMCC and DMCC_{adapt} algorithms. Obviously, all algorithms have good convergence under Gaussian noise. Among them, the convergence of the DLMS algorithm is the best, while the convergence for the DLMP, DSIGN and DNSIGN are similar. The DMCC is better than DLMP, DSIGN, DNSIGN with respect to the convergence, but their convergence is still worse than the DMCC_{adapt} since the kernel width in DMCC_{adapt} is dynamic.

Figure 6 is the performance for estimating non-sparse system (a) when the step sizes for the DLMS, DLMP, DSIGN, DNSIGN, DMCC and DMCC_{adapt} are 0.006, 0.025, 0.01, 0.25, 0.05 and 0.08, respectively. It is found that the DLMS algorithm cannot converge, while other used algorithms can converge well. The results for the DLMP, DSIGN and DNSIGN are similar, while the DMCC and DMCC_{adapt} are better than the DLMP, DSIGN and DNSIGN in terms of the MSD and the DMCC $_{adapt}$ achieves the best performance. Due to the negative exponential term in the MCCs, the DMCC and DMCC_{adapt} algorithms obtains better MSD than other algorithms in impulsive noises. The DMCC_{adapt} selects the kernel width dynamically, and hence, it achieves better performance than the DMCC.

The tracking performance of different algorithms in impulsive noise is compared in Fig.7, and the step sizes for the DLMS, DLMP, DSIGN, DNSIGN, DMCC and DMCC_{adapt} are 0.006, 0.025, 0.01, 0.25, 0.05 and 0.08, respectively. After 20000 iterations, the system changes from the non-sparse system (a) to non-sparse system (b) given in Fig.2. From the simulation results, it can be seen that when the unknown system suddenly changes, all algorithms can track the changes, and the DMCC_{adapt} is the best from all the mentioned algorithms.

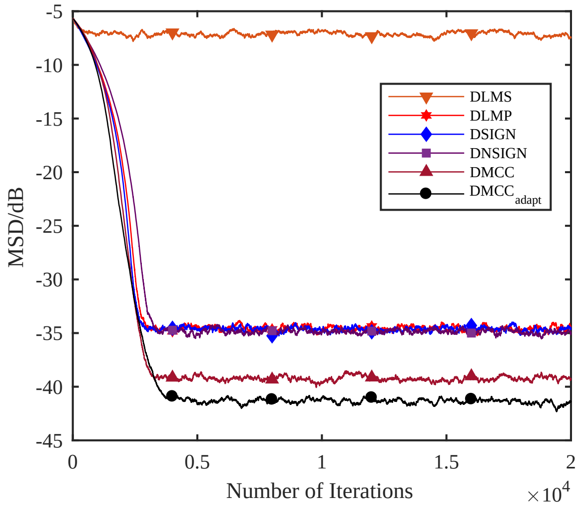

FIGURE 6. The comparison of steady state performance for the DLMS, DLMP, DSIGN, DNSIGN, DMCC and DMCC$_{\text{adapt}}$ algorithms with 10% impulsive noise.

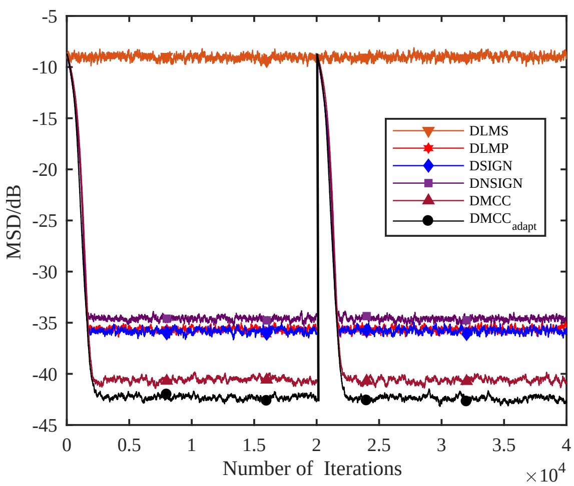

FIGURE 7. The comparison of MSD tracking performance for the DLMS, DLMP, DSIGN, DNSIGN, DMCC, DMCC$_{\text{adapt}}$ algorithms with 10% impulsive noise.

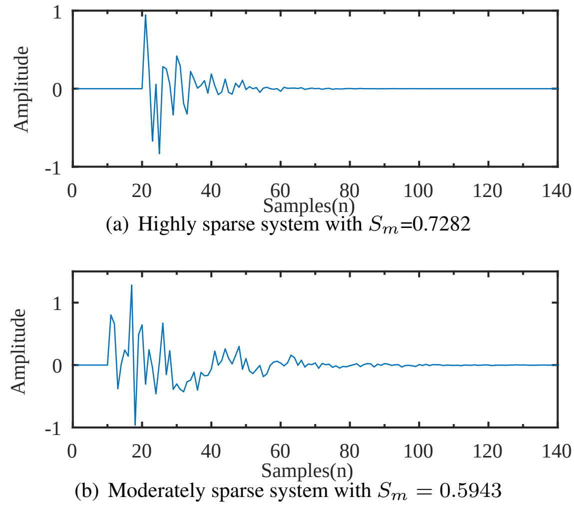

FIGURE 8. System impulse responses with different sparsity

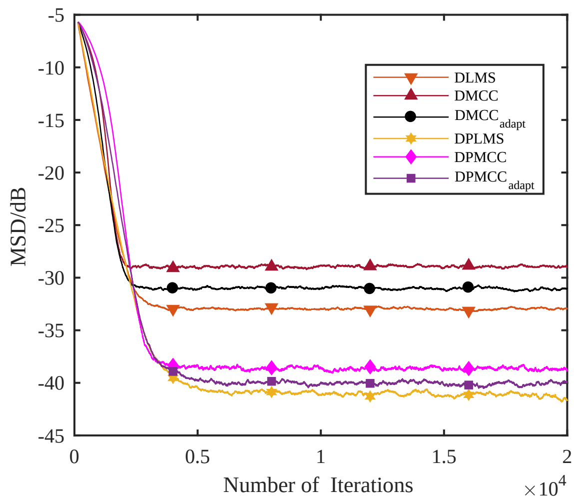

FIGURE 9. Convergence comparison of the DLMS, DMCC, DMCC$_{\text{adapt}}$, DPLMS, DPMMC, DPMCC$_{\text{adapt}}$ with Gaussian noise.

B. FOR SPARSE SYSTEM

Figures 9, 10 and 11 present the performance under sparse systems, where the sparse channels are presented in Fig.8. The sparsity of the sparse system is measured by $S_m = \frac{M}{M-\sqrt{M}}(1-\frac{\|w_k(n)\|_1}{\sqrt{M}\|w_k(n)\|_2})$.

Figure 9 is convergence comparison the DLMS, DMCC, DMCC$_{adapt}$, DPLMS, DPMCC and DPMCC$_{adapt}$ algorithms under Gaussian noise for estimating the sparse system (a) in Fig.8, and their step sizes are 0.006, 0.05, 0.06, 0.18, 0.98 and 0.9, respectively. The performance for the DPMCC$_{adapt}$ is better than the DPMCC due to the adaptive kernel width scheme.

Figure 10 shows the convergence performance of the DLMS, DMCC, DMCC$_{adapt}$, DPLMS, DPMCC, DPMCC$_{adapt}$ under 10% impulsive noise in sparse system (a) in Fig.8. The corresponding step sizes are 0.006, 0.05, 0.06, 0.18, 0.98, 0.9. As can be seen from Fig.10, the performance of the DLMS and DPLMS deteriorate under impulse noise. DPMCC is better than DMCC and DMCC$_{adapt}$, but it is worse than the DPMCC$_{adapt}$. Figure 11 shows the tracking ability and robustness of the algorithms in time-varying sparse systems. After 20000 iterations, the system changes from high sparse system (a) to moderate sparse system (b) shown in Fig.8. In this experiment, when the unknown system changes, all algorithms can maintain good tracking ability, and the DPMCC$_{adapt}$ performs the best among the mentioned 6 algorithms. The performance comparison of the proposed DMCC$_{adapt}$ and DPMCC$_{adapt}$ with DLMS, DMCC, DPLMS, DPMCC are given for different impulsive noises in Table 4. As can be seen from Table 6, the DPMCC has the best MSD.

At last, the theory analysis and the simulation are presented and given in Fig.12. We can see that the simulation

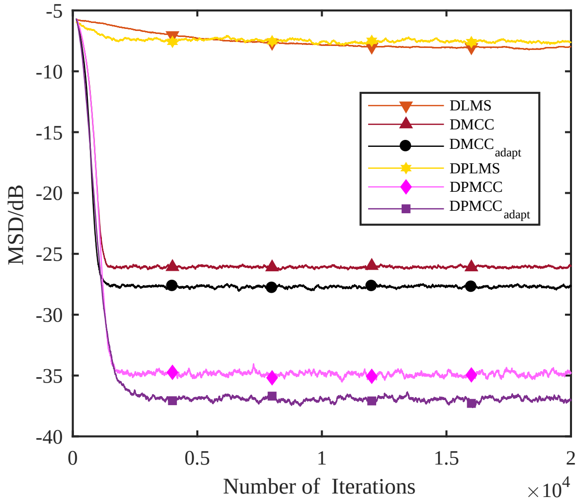

FIGURE 10. The comparison of MSD performance for the DLMS, DMCC, $\text{DMCC}_{\text{adapt}}$, DPLMS, DPMCC, $\text{DPMCC}_{\text{adapt}}$ with 10% impulsive noise.

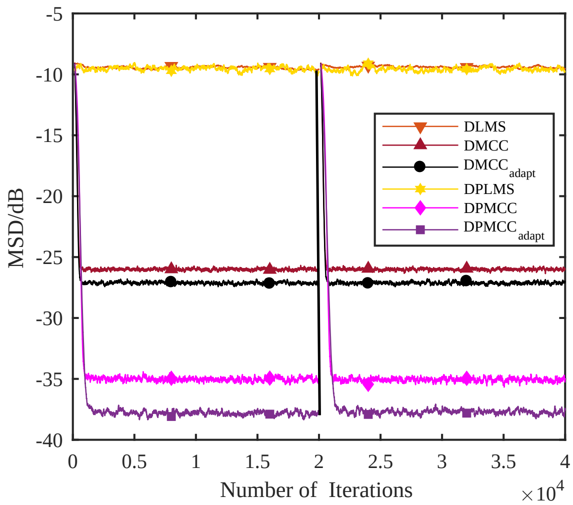

FIGURE 11. The comparison of MSD tracking performance for the DLMS, DMCC, $\text{DMCC}_{\text{adapt}}$, DPLMS, DPMCC, $\text{DPMCC}_{\text{adapt}}$ with 10% impulsive noise.

agrees well with the theoretical analysis, which verifies the correctness of the analysis.

VI. CONCLUSION

A kernel width adaption diffusion maximum correntropy criterion algorithm has been derived and discussed to seek for a solution for choosing adaptive kernel width in DMCC. The proposed DMCC_{adapt} algorithm uses different kernel widths in the iterations and the DPMCC_{adapt} incorporates the proportional coefficient adjustment scheme into the DMCC_{adapt} algorithm to develop sparseness in the structured systems. The theoretical analysis and simulation results have been given and proved to demonstrate that the DPMCC_{adapt} and DMCC_{adapt} algorithms with low MSD provide better convergence than the traditional diffusion AF algorithms under impulse noise and sparse system structures.

TABLE 6. Steady state MSD comparisons

Adaptive algorithms	Impulsive noise 10% MSD	20% MSD	30% MSD
DLMS	-9.1	-4.34	-3.39
DMCC	-28.58	-28.93	-28.85
DMCC_{adapt}	-30.68	-30.23	-29.99
DPLMS	-9.6	-6.99	-5.15
DPMCC	-36.68	-36.65	-37.01
DPMCC_{adapt}	-38.73	-38.06	-38.18

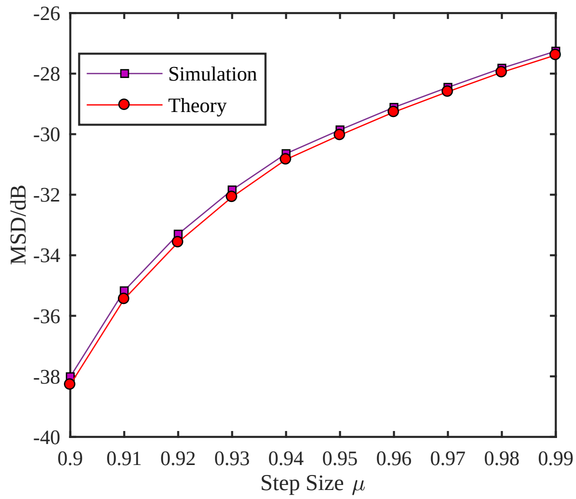

FIGURE 12. The comparison of steady state MSD for $\text{DPMCC}_{\text{adapt}}$ with varied μ.

REFERENCES

[1] Y. Li, Z. Jiang, Z. Jin, et al, "Cluster-sparse proportionate NLMS algorithm with the hybrid norm constraint," *IEEE Access*, vol.6, pp.47794-47803, 2018.

[2] Q. Wu, Y. Li, W. Xue, "A kernel affine projection-like algorithm in reproducing kernel hilbert space," *IEEE Trans. Circuits Syst. II-Express Briefs*, 10.1109/TCSII.2019.2947317.

[3] G. Gui, A. Mehbodniya, F. Adachi, "Least mean square/fourth algorithm for adaptive sparse channel estimation," *2013 IEEE 24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, Londin, Uk, Sept. 8-11th, 2013.

[4] W. Shi, Y. Li, Y. Wang, "Noise-free maximum correntropy criterion algorithm in non-gaussian environment," *IEEE Trans. Circuits Syst. II-Express Briefs*, 10.1109/TCSII.2019.2914511.

[5] S. H. Pauline, D. Samiappan, R. Kumar, A. Anand, A. Kar, "Variable tap-length non-parametric variable step-size NLMS adaptive filtering algorithm for acoustic echo cancellation," Applied Acoustics, vol. 159, 2019.

[6] H. Bu, X. Huang, X. Zhang, "A compressive-sensing-based method for radial mode analysis of aeroengine fan noise," Journal of Sound and Vibration, vol. 464, Jan. 2019.

[7] M. D. Button, J. G. Gardiner, I. A. Glover, "Measurement of the impulsive noise environment for satellite-mobile radio systems at 1.5 GHz," IEEE Transactions Vehicular Technology, vol. 51, no. 3, pp. 551-560, MAY. 2002.

[8] M. G. Sanchez, L. De. Haro, M. C. Ramon, "Impulsive noise measurements and characterization in a UHF digital TV channel," IEEE Transactions on Electromagnetic Compatibility, vol. 41, no. 2, MAY. 1999.

[9] W. Shi, Y. Li, L. Zhao and X. Liu, "Controllable sparse antenna array for adaptive beamforming," *IEEE Access*, vol. 7, pp. 6412-6423, 2019.

[10] Y. Gu, J. Jin, S. Mei, "l_0-norm constraint LMS algorithm for sparse system identification," *IEEE Signal Process. Lett.*, vol. 16, no. 9, pp. 774-777, Sept. 2009.

[11] Y. Li, Z. Jiang, O. M. O. Osman, et al, “Mixed norm constrained sparse APA algorithm for satellite and network echo channel estimation,” *IEEE Access*, vol. 6, pp.65901-65908, 2018.

[12] G. Gui, W. Peng, F. Adachi, “Adaptive system identification using robust LMS/F algorithm,” *International Journal of Communication Systems*, vol. 27, no.11, pp.2956-2963, 2014.

[13] X. Hao, M. Wang, S. Hou, Q. Gong, B. Liu, "Distributed Topology Control and Channel Allocation Algorithm for Energy Efficiency in Wireless Sensor Network: From a Game Perspective," Wireless Personal Communications, vol. 80, no. 4, pp. 1557-1577, Feb. 2015.

[14] C. G. Lopes, A. H. Sayed, "Incremental adaptive strategies over distributed networks," IEEE Transactions on Signal Processing, vol. 55, no. 8, pp. 4064-4077, Jul. 2007.

[15] C. G. Lopes, A. H. Sayed, "Incremental Adaptive Strategies Over Distributed Networks," IEEE Transactions on Signal Processing, vol. 55, no. 8, Aug. 2007.

[16] H. Zhao, X. Zeng, J. Zhang, "Adaptive reduced feedback FLNN filter for active control of nonlinear noise processes," Signal Processing, vol. 90, no. 3, pp. 834-847, Mar. 2010.

[17] Y. Li, Y. Wang, T. Jiang, "Norm-adaption penalized least mean square/fourth algorithm for sparse channel estimation," Signal Processing, vol. 128, pp. 243-251, Nov. 2016.

[18] M. Bouvet, S. C. Schwartz, "Comparison of adaptive and robust receivers for signal detection in ambient underwater noise," IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 37, no. 5, pp. 621-626, 1989.

[19] R. W. McMillan, I. Kohlberg, "A probabilistic model of the radar signal-to-clutter and noise ratio for Weibull-distributed clutter," in 2010 IEEE Radar Conference, Washington, DC, USA, 2010.

[20] W. Ma, B. Chen, H. Qu, J. Zhao, "Sparse least mean p-power algorithms for channel estimation in the presence of impulsive noise," Signal, Image and Video Processing volume, vol. 10, pp. 503-510, Mar, 2015.

[21] M. Z. U. Rahman, R. A. Shaik, D. V. R. K. Reddy, "Efficient sign based normalized adaptive filtering techniques for cancelation of artifacts in ECG signals: Application to wireless biotelemetry," Signal Processing, vol. 91, no. 2, pp. 225-239, Feb. 2011.

[22] K. Kumar, N. V. George, "A generalized maximum correntropy criterion based robust sparse adaptive room equalization," Applied Acoustics, vol. 158, Jan. 2020.

[23] Y. Li, Y. Wang, R. Yang and F. Albu, "A soft parameter function penalized normalized maximum correntropy criterion algorithm for sparse system identification," Entropy, vol. 19, no.1, Jan. 2017.

[24] Y. He, F. Wang, Y. Li, J. Qin, B. Chen, "Robust Matrix completion via maximum correntropy criterion and half quadratic optimization," IEEE Transactions on Signal Processing, vol. 68, no.1, pp.181-195, 2020.

[25] B. Chen, X. Wang, Y. Li, P. Ren, J. C. Principe, "Maximum correntropy criterion with variable center," IEEE Signal Processing Letters, vol.26, no.8, pp.1212-1216, 2019.

[26] B. Chen, Z. Li, Y. Li, P. Ren, "Asymmetric Correntropy for robust adaptive filtering," IEEE Signal Processing Letters, arXiv:1911.11855.

[27] W. Wang, H. Zhao, Y. Yu, L. Lu, Z. Zheng, "Robust adaptive filtering algorithm based on maximum correntropy criteria for censored regression," Signal Processing, vol. 160, pp. 88-98, Jul. 2019.

[28] B. Du, X. Tang, Wang Z. Wang, L. Zhang, D. Tao, "Robust Graph-Based Semisupervised Learning for Noisy Labeled Data via Maximum Correntropy Criterion," IEEE transactions on cybernetics, vol. 49, no. 4, pp. 1440-1453, Apr. 2019.

[29] S. Zhao, B. Chen, and J. C. Prinncipe, "Kernel Adaptive Filtering with Maximum Correntropy Criterion," in The 2011 International Joint Conference on Neural Networks, San Jose, CA, USA, 2011.

[30] S. Wang, L. Dang, W. Wang, G. Qian, K. T. Chi, "Kernel Adaptive Filters With Feedback Based on Maximum Correntropy," IEEE Access, vol. 6, pp. 10540-10552, Feb. 2018.

[31] H. J. Baek, J. Cho, J. W. Shin, "Variable step-size oscillator-based adaptive notch filer algorithm using maximum correntropy criterion," Electronics Letters, vol. 54, no. 4, pp. 246-248, 2018.

[32] C. Liu, Y. Qi, W. Ding, W. Zhang, "Fractional length algorithm for adaptive filter based on maximum correntropy criterion," Journal of beihang university, vol. 42, no. 2, Feb. 2016.

[33] W. Wang, J. Zhao, R. Qu, B. Chen, "An Adaptive Kernel Width Update Method of Correntropy for Channel Estimation," in 2015 IEEE International Conference on Digital Signal Processing (DSP), Singapore, Singapore, 2015.

[34] W. Wang, J. Zhao, H. Qu, B. Chen, J. C. Principe, "A switch kernel width method of correntropy for channel estimation," in 2015 International Joint Conference on Neural Networks (IJCNN), Killarney, Ireland, 2015.

[35] L. Lu, H. Zhao, "Active impulsive noise control using maximum correntropy with adaptive kernel size," Mechanical Systems and Signal Processing, vol. 87, pp. 180-191, Mar. 2017.

[36] F. Huang, J. Zhang, S. Zhang, "Adaptive Filtering Under a Variable Kernel Width Maximum Correntropy Criterion," IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 64, no. 10, pp. 1247-1251, 2017.

[37] A. Aliabadi, M. C. Amirani, C. Ghobadi, "Theoretical Performance Analysis of Sparse System Identification Using Incremental and Diffusion Strategies Over Adaptive Networks," Wireless Personal Communications, vol. 109, pp. 1181-1193, May. 2019.

[38] Y. Li, Y. Wang, T. Jiang, "Sparse least mean mixed-norm adaptive filtering algorithms for sparse channel estimation applications," International Journal of Communication Systems, vol. 30, no.8, e3181, 2017.

[39] W. Ma, J. Duan, J. Cao, Y. Li, B. Chen, "Proportionate adaptive filtering algorithms based on mixed square/fourth error criterion with unbiasedness criterion for sparse system identification," International Journal of Adaptive Control and Signal Processing, vol. 32, no. 11, pp. 1644-1654, Nov. 2018.

[40] Y. Li, Y. Wang, L. Sun, "A flexible sparse set-membership NLMS algorithm for multi-path and acoustic echo channel estimations," Applied Acoustics, vol. 148, pp. 390-398, Nov. 2019.

[41] Z. Jin, Y. Li, J. Liu, "An Improved Set-Membership Proportionate Adaptive Algorithm for a Block-Sparse System," Symmetry, vol. 10, no. 3, pp. 75, Mar. 2018.

[42] M. K. H. Abdallatif, I. Abdel-Qader, "A framework for sparse adaptive system identification via integrating a new multiple time-varying stepsize and zero-attracting techniques," in 2018 IEEE 8th Annual Computing and Communication Workshop and Conference, Las Vegas, NV, USA, 2018.

[43] Y. Li, Z. Jiang, W. Shi, X. Han, B. Chen, "Blocked Maximum Correntropy Criterion Algorithm for Cluster-Sparse System Identifications," IEEE Transactions on Circuits and Systems, vol. 66, no. 11, pp. 1915-1919, Nov. 2019.

[44] H. Zhang, C. Peng, H. Sun, D. Du, "Adaptive state estimation for cyber physical systems under sparse attacks," Transactions of the Institute of Measurement and Control, vol. 41, no. 6, pp. 1571-1579, 2018.

[45] F. Chen, X. Li, S. Duan, L. Wang, "Diffusion generalized maximum correntropy criterion algorithm for distributed estimation over multitask network," Digital Signal Processing, vol. 81, pp. 16-25, Oct. 2018.

[46] X. Xu, H. Qu, J. Zhao, F. Yan, W. Wang, "Diffusion Maximum Correntropy Criterion Based Robust Spectrum Sensing in Non-Gaussian Noise Environments," Entropy, vol. 20, no. 4, 2018.

[47] W. Ma, B. Chen, J. Duan, H. Zhao, "Diffusion maximum correntropy criterion algorithms for robust distributed estimation," Digit Signal Process, vol. 58, pp. 10-19, Nov. 2016.

[48] W. M. Bazzi, A. Rastegarnia, A. Khalili, "A Robust Diffusion Adaptive Network Based on the Maximum Correntropy Criterion," in 2015 24th International Conference on Computer Communication and Networks (ICCCN), Las Vegas, NV, USA, 2015.

[49] M. Hajiabadi, G. A. Hodtani, H. Khoshbin, "Adaptive multitask network based on maximum correntropy learning algorithm," International Journal of Adaptive Control and Signal Processing , Feb. 2017.

[50] F. S. Cattivelli, A. H. Sayed, "Diffusion LMS Strategies for Distributed Estimation," IEEE Trans, vol. 58, no. 3, pp. 1035-1048, Mar. 2010.

[51] F. S. Cattivelli, A. H. Sayed, "Diffusion LMS Strategies for Distributed Estimation," IEEE Trans, vol. 58, no. 3, pp. 1035-1048, Mar. 2010.

[52] C. G. Lopes, A. H. Sayed, "Diffusion least-mean squares over adaptive networks:formulation and performance analysis," IEEE Trans. Signal Process, vol. 56, no. 7, pp. 3122-3136, Jul. 2008.

[53] D. S. Tracy, R. P. Singh, "A new matrix product and its applications in partitioned matrix differentiation," Statistica Neerlandica, vol. 26, no. 4, pp. 143-157, 1972.

[54] A. M. Wilson, T. Panigrahi, A. Dubey, "Robust distributed Lorentzian adaptive filter with diffusion strategy in impulsive noise environment," Digital Signal Processing, vol. 96, Jan. 2020.

[55] L. Xiao, S. Boyd, "Fast Linear iterations for distributed averaging," Sys And Control, vol. 53, no. 1, pp. 65-78, Sep. 2004.

YING GUO was born in Liaoning, China in 1975. She received the B.S. degree in communication engineering from Lanzhou Jiaotong University, Lanzhou, China in 1998, and the M.S. degree in mechanical engineering from Northeast Electric Power University, Changchun, China, in 2004, the Ph.D. degree in communication engineering from Dalian University of Technology, Dalian, China, in 2009. She is an Associate Professor with the Communication Engineering at Shenyang University of Technology.

BING MA was born in Liaoning, China in 1995. She received the B.S. degree in communication engineering from University of Science and Technology Liaoning, Anshan, China in 2017. She is currently pursuing the M.S. degree in electronic and communication engineering at Shenyang University of Technology, Shenyang, China.

YINGSONG LI received his B.S. degree in electrical and information engineering, and M.S. degree in Electromagnetic Field and Microwave Technology from Harbin Engineering University, 2006 and 2011, respectively. He received his Ph.D degree from both Kochi University of Technology (KUT), Japan and Harbin Engineering University, China in 2014. He is a visiting scholar of University of California, Davis from March 2016 to March 2017, a visiting Professor of University of York, UK in 2018, a visiting Professor of Far Eastern Federal University (FEFU) and KUT from 2018. He is a Postdoc of Key Laboratory of Microwave Remote Sensing, Chinese Academy of Sciences from 2016 to 2020. Now, he is a full professor of Harbin Engineering University from July 2014. He is a Fellow of Applied computational Electromagnetics Society (ACES Fellow), and he is also a senior member of Chinese Institute of Electronics (CIE) and a senior member of IEEE. He has authored and coauthored about 300 publications in various areas of electrical engineering. His current research interests include remote sensing, underwater communications, signal processing, radar, SAR imaging, compressed sensing, wireless sensors for medical applications, metasurface designs and microwave antennas.

Dr. Li is an Area Editor of AE $\ddot{U}$-International Journal of Electronics and Communications, and he is an Associate Editor of IEEE Access and Applied Computational Electromagnetics Society Journal (ACES Journal). He is the TPC Co-Chair of the 2019IEEE International Workshop on Electromagnetics (iWEM 2019) , 2019 IEEE 2nd International Conference on Electronic Information and Communication Technology (ICEICT 2019), 2019 International Applied Computational Electromagnetics Society (ACES) Symposium-China, 2019 Cross Strait Quad-regional Radio Science and Wireless Technology Conference (2019 CSQRWC), and iWEM 2020. He also serves as a Session Chair or Organizer for many international conferences, including the WCNC, AP-S, ACES ect. He acts as a Reviewer of numerous IEEE, IET, Elsevier and other international journals.