IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received September 23, 2019, accepted October 5, 2019, date of publication October 15, 2019, date of current version October 29, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2947636

Robust Diffusion Affine Projection Algorithm With Variable Step-Size Over Distributed Networks

PUCHA SONG, HAIQUAN ZHAO, (Senior Member, IEEE), AND XIANGPING ZENG
Key Laboratory of Magnetic Suspension Technology and Maglev Vehicle, Ministry of Education, School of Electrical Engineering, Southwest Jiaotong University, Chengdu 610031, China

Corresponding author: Haiquan Zhao (hqzhao_swjtu@126.com)

This work was supported in part by the National Science Foundation of China under Grant 61871461, Grant 61571374, and Grant 61433011, and in part by the Sichuan Science and Technology Program under Grant 19YYJC0681.

ABSTRACT The estimation performance of the standard diffusion affine projection algorithm may be degraded when the distributed network nodes are disturbed by impulsive noise. To overcome the limitation, a diffusion affine projection M-estimate (DAPM) algorithm is proposed for distributed estimation in the adaptive diffusion networks. This algorithm uses a robust cost function based on M-estimate function to eliminate the adverse effects of impulsive noise on distributed diffusion network nodes. In order to further enhance the performance of the DAPM algorithm, namely fast convergence rate and low steady-state error, a variable step-size diffusion affine projection M-estimate (VSS-DAPM) algorithm is presented. In addition, the convergence range of the step-size is deduced to ensure the convergence of the proposed algorithms. Computer simulations show that the proposed DAPM and VSS-DAPM algorithms have good convergence performance for distributed estimation in the adaptive diffusion networks. More importantly, the proposed VSS-DAPM algorithm improves convergence rate and the network mean square deviation (MSD) as compared to the DAPM algorithm in the distributed estimation.

INDEX TERMS Affine projection algorithm, impulsive noise, M-estimate, variable step-size, distributed estimation.

I. INTRODUCTION

Distributed estimation is an efficient estimation method that uses a set of connected network nodes to estimate certain parameters of interest [1]. In the adaptive distributed networks, two adaptive networks based on network topology are an incremental network and a diffusion network, respectively. In the incremental scenario, a circular path is established over the network in which the node communicates with neighboring nodes [2], in the diffusion mode, each node information is locally processed and then propagated through the network node in real time [3]. In a distributed diffusion network, nodes can acquire more data from neighboring nodes. Therefore, the traffic in the diffusion network is higher than the incremental network. Compared to well-studied incremental networks, diffusion networks are robust, flexible and fully distributed for node or link failure [4]–[6]. In the past ten years, several diffusion adaptive filtering algorithms have been proposed to enhance the distributed estimation in diffusion networks [7]–[15]. Among them, the diffusion least mean square (DLMS) algorithm [7] is a commonly used distributed estimation algorithm because of its small computational complexity and simple implementation. In [8], a diffusion least mean fourth (DLMF) algorithm was presented for distributed estimation in non-Gaussian noise environments, it shows small steady-state misalignment than DLMS algorithm. In order to enhance the convergence rate of high colored input signals, the diffusion recursive least square (DRLS) algorithm [9] and diffusion affine projection (DAP) algorithm [10] were proposed for distributed estimation in the adaptive diffusion networks. However, when the distributed diffusion network nodes are disturbed by impulsive noise, the performance of these diffusion adaptive algorithms may be degraded. To overcome this limitation, several distributed robust diffusion algorithms have been studied [11]–[15], the diffusion sign-error least mean

The associate editor coordinating the review of this manuscript and approving it for publication was Chuan Li.

square (DSE-LMS) algorithm was proposed in [11], which performs the sign operation to the error signals of all network nodes, and the diffusion affine projection sign algorithm (DAPSA) was described in [12], which is derived by minimizing l_1-norm intermediate error vector subject to a constraint on the filter coefficients. In [13], the diffusion least logarithmic absolute difference (DLLAD) algorithm was developed against impulsive noise for distributed estimation in the adaptive diffusion networks. The DLLAD algorithm shows faster convergence rate and lower steady-state error than DSE-LMS algorithm. In addition, the diffusion generalized maximum correntropy criterion (DGMCC) algorithm [14] was proposed to deal with impulsive noise in network nodes. Recently, a diffusion robust variable step-size least mean square (DRVSS-LMS) algorithm [15] was developed against impulsive noise for distributed estimation in the adaptive diffusion networks. However, when the distributed diffusion network nodes have high colored input signals, these robust diffusion algorithms show low convergence speed. Therefore, in this paper, a diffusion affine projection M-estimate (DAPM) algorithm is proposed for distributed estimation in impulsive noise interference environments. This algorithm uses a robust cost function based on M-estimate function in which containing the weighting matrix and is derived by solving the local minimization problem for distributed diffusion network. Recently, variable step-size (VSS) methods [15]–[21] have been widely used in distributed adaptive networks, which greatly improve the convergence performance of distributed algorithms. Hence, a variable step-size diffusion affine projection M-estimate (VSS-DAPM) algorithm is proposed to enhance the convergence speed and steady-state misalignment of DAPM algorithm.

The rest of the paper is organized as follows: In Section II, the introduction of preparatory work. In Section III, we propose the DAPM algorithm and VSS-DAPM algorithm. In Section IV, the convergence range of the step-size is derived. In Section V, computer simulations verify convergence performance of the proposed algorithms. Conclusions are obtained in Section VI. Throughout this paper, the notations from Table 1 are used.

TABLE 1. Mathematical notation.

Operators	Description
$(\bullet)^T$	Transpose of a vector or a matrix.
$\mathrm{col}[\mathbf{t}_1,\mathbf{t}_2,\ldots,\mathbf{t}_N]$	Creates the column vector by stacking the column vectors $\mathbf{t}_1$, $\mathbf{t}_2$, …, $\mathbf{t}_N$.
$\lvert\bullet\rvert$	Norm of a scalar.
$\lVert\bullet\rVert^2$	Squared Euclidean norm of a vector.
$\rho[\bullet]$	M-estimate function.
$\mathrm{sgn}(\bullet)$	Sign function.
$\mathrm{med}(\bullet)$	Median operation.
$E\{\bullet\}$	Expectation operation.
$\mathrm{diag}\{\ldots\}$	Creates a diagonal matrix or diagonal block matrix with the elements or matrices in $\{\ldots\}$.
$\lambda_{\max}(\bullet)$	Largest eigenvalue of a matrix.
$\mathbf{I}_M$	M by M identity matrix.
$\mathbf{X}\otimes\mathbf{Y}$	Kronecker product of two matrices $\mathbf{X}$ and $\mathbf{Y}$.

II. PRELIMINARIES

A. THE STANDARD DAP ALGORITHM

Consider a distributed diffusion network composed of N nodes and the data $\{d_k(n), \mathbf{x}_k(n)\}$ at node k collected by the various nodes are related to unknown parameter vector $\mathbf{w}^0$ that satisfies follow the linear model

$$d_k(n) = \mathbf{x}_k^T(n)\mathbf{w}^0 + \eta_k(n) \tag{1}$$

where $\mathbf{x}_k(n) = [x_k(n), x_k(n-1), \ldots, x_k(n-M+1)]^T$ is the input signal vector at node k, $\eta_k(n)$ denotes the additive noise at node k, and $d_k(n)$ denotes the local desired signal.

The adaptive filter output signal $y_k(n)$ at node k is computed as

$$y_k(n) = \mathbf{x}_k^T(n)\mathbf{w}_k(n) \tag{2}$$

where $\mathbf{w}_k(n)$ is the estimated vector of the adaptive filter with respect to $\mathbf{w}^0$.

Then, the error signal $e_k(n)$ at node k is given by

$$\begin{aligned} e_k(n) &= d_k(n) - y_k(n) \\ &= \mathbf{x}_k^T(n)\left(\mathbf{w}^0 - \mathbf{w}_k(n)\right) + \eta_k(n) \end{aligned} \tag{3}$$

The weight vector update equation for the diffusion affine projection (DAP) algorithm [22] is summarized, as follows:

$$\boldsymbol{\psi}_k(n+1) = \mathbf{w}_k(n) + \mu_k \mathbf{X}_k(n)\left(\mathbf{X}_k^T(n)\mathbf{X}_k(n)\right)^{-1}\mathbf{e}_k(n) \tag{4}$$

$$\mathbf{w}_k(n+1) = \sum_{m\in N_k} c_{m,k}\boldsymbol{\psi}_m(n+1) \tag{5}$$

where $\mathbf{e}_k(n) = \mathbf{d}_k(n) - \mathbf{X}_k^T(n)\mathbf{w}_k(n)$ represents the error vector of node k,

$$\mathbf{d}_k(n) = [d_k(n), d_k(n-1), \ldots, d_k(n-P+1)]^T \tag{6}$$

and

$$\mathbf{X}_k(n) = [\mathbf{x}_k(n), \mathbf{x}_k(n-1), \ldots, \mathbf{x}_k(n-P+1)] \tag{7}$$

is the input signal matrix at node k, P is affine projection order. μ_k is the local step-size parameter, N_k is the set of neighbor nodes that are connected to node k, and $c_{m,k} \geq 0$ denotes the weighted coefficient.

B. ADDITIVE NOISE MODEL

The additive noise $\eta_k(n)$ of node k is contaminated Gaussian noise [15], [23], which is modeled as follows:

$$\eta_k(n) = \theta_k(n) + \omega_k(n) = \theta_k(n) + b(n)\nu_k(n) \quad (8)$$

where $\theta_k(n)$ and $\nu_k(n)$ are both independent and identically distributed zero-mean Gaussian sequences with variances $\sigma^2_{\theta_k}$ and $\sigma^2_{\nu_k}$ ($\sigma^2_{\nu_k} \gg \sigma^2_{\theta_k}$), respectively. The impulsive noise $\omega_k(n)$ is usually generated as a Bernoulli-Gaussian process, i.e., $\omega_k(n) = b(n)\nu_k(n)$ [24], where $b(n)$ is a Bernoulli process with the probability density function described by $P(b(n) = 1) = p_k$, $P(b(n) = 0) = 1 - p_k$.

Therefore, the corresponding probability density function (pdf) of the contaminated Gaussian noise is given by [23]

$$f_{\eta_k}(x) = \frac{1-p_k}{\sqrt{2\pi}\sigma_{\theta_k}} \exp\left(\frac{-x^2}{2\sigma^2_{\theta_k}}\right) + \frac{p_k}{\sqrt{2\pi}\sigma_{s_k}} \exp\left(\frac{-x^2}{2\sigma^2_{s_k}}\right) \quad (9)$$

where $\sigma^2_{\eta_k} = p_k\sigma^2_{s_k} + (1-p_k)\sigma^2_{\theta_k}$, $\sigma^2_{s_k} = \sigma^2_{\theta_k} + \sigma^2_{\nu_k}$ and p_k denotes the probability of the occurrence of the impulsive noise.

C. M-ESTIMATE FUNCTION

The M-estimate function $\rho[e_k(n)]$ is widely used to deal with impulsive noise [23], [25]–[27], as follows:

$$\rho[e_k(n)] = \begin{cases} e_k^2(n)/2, & \text{if } |e_k(n)| < \xi_k(n) \\ \xi_k^2(n)/2, & \text{otherwise} \end{cases} \quad (10)$$

then,

$$\varphi[e_k(n)] = \frac{\partial\rho[e_k(n)]}{\partial e_k(n)} = \begin{cases} e_k(n), & \text{if } |e_k(n)| < \xi_k(n) \\ 0, & \text{otherwise} \end{cases} \quad (11)$$

where the threshold parameter $\xi_k(n)$ is set to eliminate outliers, which is usually adjusted as [26]

$$\xi_k(n) = \mathrm{k}_{\xi_k}\hat{\sigma}_{e_k}(n) = 2.576\hat{\sigma}_{e_k}(n) \quad (12)$$

where $\hat{\sigma}^2_{e_k}(n+1) = \lambda\hat{\sigma}^2_{e_k}(n) + Q(1-\lambda)\,\mathrm{med}\left(\mathbf{A}_{e_k}(n)\right)$, λ is the forgetting factor which is close to one and $Q = 1.483(1 + 5/(N_w - 1))$ is the finite sample correction factor. med$(\cdot)$ represents the median operation, and $\mathbf{A}_{e_k}(n) = \{e_k^2(n), e_k^2(n-1), \ldots, e_k^2(n-N_w+1)\}$, and N_w is the length of the estimation window.

III. PROPOSED ALGORITHMS

A. THE PROPOSED DAPM ALGORITHM

The estimation performance of the standard DAP algorithm may be degraded when the distributed network nodes are disturbed by impulsive noise. To overcome the limitation, a robust cost function over distributed diffusion networks based on the M-estimate function is defined as

$$\begin{aligned} J_k^{loc}[\mathbf{w}_k] = {} & \mathrm{sgn}\left(\rho\left[\left(\mathbf{d}_k(n) - \mathbf{X}_k^T(n)\mathbf{w}_k\right)^T\right]\right)\boldsymbol{\Lambda}_k(n) \\ & \times\rho\left[\mathbf{d}_k(n) - \mathbf{X}_k^T(n)\mathbf{w}_k\right] \\ & + \sum_{m\in N_k/\{k\}} b_{m,k}\left\|\mathbf{w}_k - \boldsymbol{\psi}_m(n)\right\|^2 \end{aligned} \quad (13)$$

where $\rho[\cdot]$ represents the M-estimate function, sgn$(\cdot)$ is the sign function and $\boldsymbol{\Lambda}_k(n) = \left(\mathbf{X}_k^T(n)\mathbf{X}_k(n)\right)^{-1}$. $N_k/\{k\}$ denotes the set of all nodes connected to node k (excluding node k), $b_{m,k} \geq 0$ is a weighted coefficient, $\boldsymbol{\psi}_m(n)$ represents the intermediate estimate of node m with respect to $\mathbf{w}^0$. Hence, at node k, the gradient of the cost function in (13) is calculated as

$$\begin{aligned} \nabla J_k^{loc}(\mathbf{w}_k(n)) &= \frac{\partial J_k^{loc}(\mathbf{w}_k(n))}{\partial \mathbf{w}_k(n)} \\ &= -\mathbf{X}_k(n)\boldsymbol{\Lambda}_k(n)\varphi[\mathbf{e}_k(n)] \\ &\quad +2\sum_{m\in N_k/\{k\}} b_{m,k}\left(\mathbf{w}_k(n) - \boldsymbol{\psi}_m(n)\right) \end{aligned} \quad (14)$$

Using the steepest-descent method at node k, we obtain the following recursion equation

$$\begin{aligned} \mathbf{w}_k(n+1) &= \mathbf{w}_k(n) - \mu_k\nabla J_k^{loc}(\mathbf{w}_k(n)) \\ &= \mathbf{w}_k(n) + \mu_k\mathbf{X}_k(n)\boldsymbol{\Lambda}_k(n)\varphi[\mathbf{e}_k(n)] \\ &\quad +2\mu_k\sum_{m\in N_k/\{k\}} b_{m,k}\left(\boldsymbol{\psi}_m(n) - \mathbf{w}_k(n)\right) \end{aligned} \quad (15)$$

According to the adapt-then-combine scheme [28], hence, we can compute it in two steps by generating an intermediate estimate $\boldsymbol{\psi}_k(n+1)$, as follows:

$$\boldsymbol{\psi}_k(n+1) = \mathbf{w}_k(n) + \mu_k\mathbf{X}_k(n)\boldsymbol{\Lambda}_k(n)\varphi[\mathbf{e}_k(n)] \quad (16)$$

$$\mathbf{w}_k(n+1) = \boldsymbol{\psi}_k(n+1) + 2\mu_k\sum_{m\in N_k/\{k\}} b_{m,k}\left(\boldsymbol{\psi}_m(n) - \mathbf{w}_k(n)\right) \quad (17)$$

Since each node in the distributed diffusion network will run the recursion of equations (16)-(17), hence, we can replace $\boldsymbol{\psi}_m(n)$ in (17) by $\boldsymbol{\psi}_m(n+1)$. In addition, $\boldsymbol{\psi}_k(n+1)$ is the updated estimate of $\mathbf{w}_k(n)$ in (16) at node k, it is reasonable to replace $\mathbf{w}_k(n)$ in (17) by $\boldsymbol{\psi}_k(n+1)$ [7], [29], [30]. By using these substitutions in (17), we get

$$\begin{aligned} \mathbf{w}_k(n+1) &= \boldsymbol{\psi}_k(n+1) + 2\mu_k\sum_{m\in N_k/\{k\}} b_{m,k}\left(\boldsymbol{\psi}_m(n+1) - \boldsymbol{\psi}_k(n+1)\right) \\ &= \sum_{m\in N_k} c_{m,k}\boldsymbol{\psi}_m(n+1) \end{aligned} \quad (18)$$

where $c_{m,k} = 2\mu_k b_{m,k} (m \neq k)$, $c_{k,k} = 1 - 2\mu_k\sum_{m\in N_k/\{k\}} b_{m,k}$.

Using (16) and (18), a diffusion affine projection M-estimate (DAPM) algorithm for distributed diffusion network is obtained as

$$\begin{cases} \boldsymbol{\psi}_k(n+1) = \mathbf{w}_k(n) + \mu_k\mathbf{X}_k(n)\left(\mathbf{X}_k^T(n)\mathbf{X}_k(n)\right)^{-1}\varphi[\mathbf{e}_k(n)] \\ \mathbf{w}_k(n+1) = \sum_{m\in N_k} c_{m,k}\boldsymbol{\psi}_m(n+1) \end{cases} \quad (19)$$

B. THE PROPOSED VSS-DAPM ALGORITHM

To further enhance the convergence performance of DAPM algorithm, which has both fast convergence rate and low steady-state error. Hence, we use the variable

TABLE 2. Proposed VSS-DAPM algorithm.

Initializations:	$\mathbf{w}_k(0)=\mathbf{0}$, $\hat{\sigma}^2_{e_k}(0)=0$, $\mu_k(0)=\mu_{\max}$
Parameters:	Q, N_w, λ, P, τ, γ, $0<\mu_{\min}<\mu_{\max}$

Adaptive process:

for n=0, 1, 2, …

$$e_k(n)=d_k(n)-\mathbf{x}_k^T(n)\mathbf{w}_k(n)$$
$$\mathbf{A}_{e_k}(n)=\left\{e_k^2(n),e_k^2(n-1),\ldots,e_k^2(n-N_w+1)\right\}$$
$$\hat{\sigma}^2_{e_k}(n+1)=\lambda\hat{\sigma}^2_{e_k}(n)+Q(1-\lambda)\,\mathrm{med}\left(\mathbf{A}_{e_k}(n)\right)$$
$$\xi_k(n)=2.576\hat{\sigma}_{e_k}(n)$$
$$\varphi[e_k(n)]=\begin{cases}e_k(n), & \text{if } |e_k(n)|<\xi_k(n)\\ 0, & \text{otherwise}\end{cases}$$
$$\varphi[\mathbf{e}_k(n)]=\left[\varphi[e_k(n)],\varphi[e_k(n-1)],\ldots,\varphi[e_k(n-P+1)]\right]^T$$
$$\mathbf{X}_k(n)=\left[\mathbf{x}_k(n),\mathbf{x}_k(n-1),\ldots,\mathbf{x}_k(n-P+1)\right]$$
$$\chi_k(n+1)=\tau\mu_k(n)+\gamma\varphi^2[e_k(n)]$$
$$\chi_k(n+1)=\begin{cases}\mu_{\max}, & \text{if } \chi_k(n+1)>\mu_{\max}\\ \mu_{\min}, & \text{if } \chi_k(n+1)<\mu_{\min}\\ \chi_k(n+1), & \text{otherwise}\end{cases}$$
$$\boldsymbol{\psi}_k(n+1)=\mathbf{w}_k(n)+\mu_k(n)\mathbf{X}_k(n)\left(\mathbf{X}_k^T(n)\mathbf{X}_k(n)\right)^{-1}\varphi[\mathbf{e}_k(n)]$$
$$\mathbf{w}_k(n+1)=\sum_{m\in N_k}c_{m,k}\boldsymbol{\psi}_m(n+1)$$
$$\mu_k(n+1)=\sum_{m\in N_k}c_{m,k}\chi_m(n+1)$$

end

step-size $\mu_k(n)$ instead of the fixed step-size μ_k in (19) and rewrite this equation as

$$\begin{cases}\boldsymbol{\psi}_k(n+1)=\mathbf{w}_k(n)+\mu_k(n)\mathbf{X}_k(n)\left(\mathbf{X}_k^T(n)\mathbf{X}_k(n)\right)^{-1}\varphi\left[\mathbf{e}_k(n)\right]\\ \mathbf{w}_k(n+1)=\sum\limits_{m\in N_k}c_{m,k}\boldsymbol{\psi}_m(n+1)\end{cases} \tag{20}$$

Inspired by the variable step-size method and adopting the diffusion cooperation strategy, the proposed VSS-DAPM algorithm uses the variable step-size method as follows:

$$\begin{cases}\chi_k(n+1)=\tau\mu_k(n)+\gamma\varphi^2\left[e_k(n)\right]\\ \mu_k(n+1)=\sum\limits_{m\in N_k}c_{m,k}\chi_m(n+1)\end{cases} \tag{21}$$

where the parameters $0<\tau<1$ and $\gamma>0$ control the dynamic behavior of the step-size. $\chi_k(n+1)$ denotes an intermediate estimate relative to $\mu_k(n)$ at node k.

In order to avoid the instability of the step-size parameter, a constraint on $\chi_k(n+1)$ is used

$$\chi_k(n+1)=\begin{cases}\mu_{\max}, & \text{if } \chi_k(n+1)>\mu_{\max}\\ \mu_{\min}, & \text{if } \chi_k(n+1)<\mu_{\min}\\ \chi_k(n+1), & \text{otherwise}\end{cases} \tag{22}$$

where $0<\mu_{\min}<\mu_{\max}$, and the proposed VSS-DAPM algorithm is summarized in Table 2.

IV. THE STABILITY ANALYSIS OF ALGORITHM

For the stability of DAPM algorithm, the selection of the local step-size μ_k is very important. It not only relates to the convergence rate of the proposed algorithm, but also determines the steady-state misalignment. Hence, the range of local step-size μ_k of the algorithm that guarantees stability is derived in this section.

Assumption 1: The input signal vector $\mathbf{x}_k(n)$ are zero-mean and spatially and temporally independent [22].

Assumption 2: The noise process $\theta_k(n)$ and $\nu_k(n)$ are both zero-mean, white, Gaussian, and spatially and temporally independent of $\mathbf{x}_k(n)$ [22].

Assumption 3: The weight error vector $\tilde{\mathbf{w}}_k(n)$ is independent of input signal vector $\mathbf{x}_k(n)$ [31], [32].

Assumption 4: The step-sizes in all nodes are the same ($\mu=\mu_k$) [22].

The update equation of (20) can be reconstructed to represent the whole distributed diffusion network as

$$\mathbf{w}(n+1)=\mathbf{G}\mathbf{w}(n)+\mathbf{G}\mathbf{D}\mathbf{X}(n)\boldsymbol{\Lambda}(n)\varphi\left[\mathbf{e}(n)\right] \tag{23}$$

where $\mathbf{w}(n)=\mathrm{col}\left[\mathbf{w}_1(n),\mathbf{w}_2(n),\ldots,\mathbf{w}_N(n)\right]$, $\mathbf{G}=\mathbf{C}^T\otimes\mathbf{I}_M$, $\mathbf{C}$ is a $N\times N$ weighting matrix, where $\{\mathbf{C}\}_{m,k}=c_{m,k}$ and $\mathbf{D}=\mathrm{diag}\left[\mu_1\mathbf{I}_M,\mu_2\mathbf{I}_M,\ldots,\mu_N\mathbf{I}_M\right]$. $\mathbf{X}(n)$ is $NM\times NP$ the block diagonal matrix which is defined as

$$\mathbf{X}(n)=\mathrm{diag}\left[\mathbf{X}_1(n),\mathbf{X}_2(n),\ldots,\mathbf{X}_N(n)\right] \tag{24}$$

and $\boldsymbol{\Lambda}(n)=\left(\mathbf{X}^T(n)\mathbf{X}(n)\right)^{-1}$, $\mathbf{e}(n)$ is $NP\times 1$ the error vector is calculated as

$$\mathbf{e}(n)=\mathbf{d}(n)-\mathbf{X}^T(n)\mathbf{w}(n) \tag{25}$$

where

$$\mathbf{d}(n)=\mathrm{col}\left[\mathbf{d}_1(n),\mathbf{d}_2(n),\ldots,\mathbf{d}_N(n)\right] \tag{26}$$

is the $NP\times 1$ vector and can be given by

$$\mathbf{d}(n)=\mathbf{X}^T(n)\mathbf{w}^{(0)}+\boldsymbol{\eta}(n) \tag{27}$$

where $\mathbf{w}^{(0)}=\boldsymbol{\Gamma}w^0$, $\boldsymbol{\Gamma}=\mathrm{col}\left[\mathbf{I}_M,\mathbf{I}_M,\ldots,\mathbf{I}_M\right]$ is a $NM\times M$ matrix, and

$$\boldsymbol{\eta}(n)=\mathrm{col}\left[\boldsymbol{\eta}_1(n),\boldsymbol{\eta}_2(n),\ldots,\boldsymbol{\eta}_N(n)\right] \tag{28}$$

where

$$\boldsymbol{\eta}_k(n)=\left[\eta_k(n),\eta_k(n-1),\ldots,\eta_k(n-P+1)\right]^T \tag{29}$$

By defining the weight error vector as $\tilde{\mathbf{w}}(n)=\mathbf{w}^{(0)}-\mathbf{w}(n)$, the weight error vector update equation in (23) is obtained as

$$\tilde{\mathbf{w}}(n+1)=\mathbf{G}\tilde{\mathbf{w}}(n)-\mathbf{G}\mathbf{D}\mathbf{X}(n)\boldsymbol{\Lambda}(n)\varphi\left[\mathbf{e}(n)\right] \tag{30}$$

where $\mathbf{G}\mathbf{w}^{(0)}=\mathbf{w}^{(0)}$, $\mathbf{e}(n)=\mathbf{X}^T(n)\tilde{\mathbf{w}}(n)+\boldsymbol{\eta}(n)$.

Taking expectation of (30), we get

$$E\left\{\tilde{\mathbf{w}}(n+1)\right\}=\mathbf{G}E\left\{\tilde{\mathbf{w}}(n)\right\}-\mathbf{G}\mathbf{D}\mathbf{H}(n) \tag{31}$$

where $\mathbf{H}(n)=E\left\{\mathbf{X}(n)\boldsymbol{\Lambda}(n)\varphi\left[\mathbf{e}(n)\right]\right\}$.

For analysis convenience, redefine $\mathbf{H}(n)$

$$\mathbf{H}(n) \triangleq \mathrm{col}\left[\mathbf{H}_1(n), \mathbf{H}_2(n), \ldots, \mathbf{H}_N(n)\right] \tag{32}$$

where $\mathbf{H}_k(n) = E\left\{\mathbf{X}_k(n)\mathbf{\Lambda}_k(n)\varphi\left[\mathbf{e}_k(n)\right]\right\}$.

According to (8) and using assumption 1, we get

$$\begin{aligned}\mathbf{H}_k(n) &= E\left\{\mathbf{X}_k(n)\mathbf{\Lambda}_k(n)\varphi\left[\mathbf{e}_k(n)\right]\right\}\\ &= PE\left\{\mathbf{x}_k(n)\left(\mathbf{x}_k^T(n)\mathbf{x}_k(n)\right)^{-1}\varphi\left[e_k(n)\right]\right\}\\ &= p_k \underbrace{PE\left\{\mathbf{x}_k(n)\left(\mathbf{x}_k^T(n)\mathbf{x}_k(n)\right)^{-1}\varphi\left[e_{k,s}(n)\right]\right\}}_{①}\\ &\quad + (1-p_k)\underbrace{PE\left\{\mathbf{x}_k(n)\left(\mathbf{x}_k^T(n)\mathbf{x}_k(n)\right)^{-1}\varphi\left[e_{k,\theta}(n)\right]\right\}}_{②}\end{aligned} \tag{33}$$

where $e_{k,s}(n) = \mathbf{x}_k^T(n)\tilde{\mathbf{w}}_k(n) + \theta_k(n) + \nu_k(n)$ represents the error vector with impulsive noise at node k, and $e_{k,\theta}(n) = \mathbf{x}_k^T(n)\tilde{\mathbf{w}}_k(n)+\theta_k(n)$ is the error vector without impulsive noise at node k.

A. CALCULATING THE FIRST ITEM

To evaluate the first expression and use Price's theorem [33], [34], we can remove time index n in the expression to get

$$\begin{aligned}&\frac{\partial\left[E\left\{\mathbf{x}_k(\mathbf{x}_k^T\mathbf{x}_k)^{-1}\varphi[e_{k,s}]\right\}\right]_m}{\partial\left[E\left\{\mathbf{x}_k(\mathbf{x}_k^T\mathbf{x}_k)^{-1}e_{k,s}\right\}\right]_m}\\ &= \frac{1}{\sqrt{2\pi E\left\{e_{k,s}^2\right\}}}\int_{-\infty}^{\infty}\varphi'[e_{k,s}]\exp\left(\frac{-e_{k,s}^2}{2E\left\{e_{k,s}^2\right\}}\right)de_{k,s}\\ &\triangleq \Gamma(e_{k,s}^2)\end{aligned} \tag{34}$$

where $\varphi'[e_{k,s}] = \partial\varphi[e_{k,s}]/\partial e_{k,s}$, $[\cdot]_m$ represents the m-th element in the vector.

By integral operation of (34) and using assumptions 1-3, we get

$$\begin{aligned}E\left\{\mathbf{x}_k(\mathbf{x}_k^T\mathbf{x}_k)^{-1}\varphi[e_{k,s}]\right\} &= \Gamma(e_{k,s}^2)E\left\{\mathbf{x}_k(\mathbf{x}_k^T\mathbf{x}_k)^{-1}e_{k,s}\right\}\\ &= \frac{\Gamma(e_{k,s}^2)}{M}E\left\{\tilde{\mathbf{w}}_k\right\}\end{aligned} \tag{35}$$

Hence, the first expression is obtained as

$$\begin{aligned}① &= PE\left\{\mathbf{x}_k(n)\left(\mathbf{x}_k^T(n)\mathbf{x}_k(n)\right)^{-1}\varphi\left[e_{k,s}(n)\right]\right\}\\ &= \frac{P}{M}\Gamma\left(e_{k,s}^2(n)\right)E\left\{\tilde{\mathbf{w}}_k(n)\right\}\end{aligned} \tag{36}$$

B. CALCULATING THE SECOND ITEM

Using the same method as the first expression in (34), the second expression is obtained as

$$\begin{aligned}② &= PE\left\{\mathbf{x}_k(n)\left(\mathbf{x}_k^T(n)\mathbf{x}_k(n)\right)^{-1}\varphi\left[e_{k,\theta}(n)\right]\right\}\\ &= \frac{P}{M}\Gamma\left(e_{k,\theta}^2(n)\right)E\left\{\tilde{\mathbf{w}}_k(n)\right\}\end{aligned} \tag{37}$$

According to (36) and (37), we can get

$$\mathbf{H}_k(n) = \frac{P}{M}\Gamma\left(e_k^2(n)\right)E\left\{\tilde{\mathbf{w}}_k(n)\right\} \tag{38}$$

where $\Gamma\left(e_k^2(n)\right) = p_k\Gamma\left(e_{k,s}^2(n)\right) + (1-p_k)\Gamma\left(e_{k,\theta}^2(n)\right)$.

Then, we calculate $\mathbf{H}(n)$,

$$\mathbf{H}(n) = \mathbf{A}(n)E\left\{\tilde{\mathbf{w}}(n)\right\} \tag{39}$$

where

$$\begin{aligned}&\mathbf{A}(n)\\ &= \frac{P}{M}\mathrm{diag}\left\{\Gamma\left(e_1^2(n)\right)\mathbf{I}_M, \Gamma\left(e_2^2(n)\right)\mathbf{I}_M, \ldots, \Gamma\left(e_N^2(n)\right)\mathbf{I}_M\right\}\end{aligned} \tag{40}$$

Therefore, the following recursion for the expectation of weight error vector is obtained as

$$\begin{aligned}E\left\{\tilde{\mathbf{w}}(n+1)\right\} &= \mathbf{G}E\left\{\tilde{\mathbf{w}}(n)\right\} - \mathbf{GDA}(n)E\left\{\tilde{\mathbf{w}}(n)\right\}\\ &= \mathbf{G}\left[\mathbf{I}_{MN} - \mathbf{DA}(n)\right]E\left\{\tilde{\mathbf{w}}(n)\right\}\end{aligned} \tag{41}$$

Using assumption 4 in (41), the mean is stable if and only if the eigenvalues of matrix satisfy the following condition:

$$\begin{aligned}|\lambda_{\max}\left(\mathbf{G}\left[\mathbf{I}_{MN} - \mu\mathbf{A}(n)\right]\right)| &\leq |\lambda_{\max}\left(\mathbf{I}_{MN} - \mu\mathbf{A}(n)\right)| < 1\\ &\Rightarrow 0 < \mu < \frac{2}{\lambda_{\max}\left(\mathbf{A}(n)\right)}\end{aligned} \tag{42}$$

V. COMPUTER SIMULATIONS

Simulation examples are presented for a diffusion network with $N = 20$ nodes, in which the topology of the network is shown in Fig. 1, we use the Metropolis rule [28] for combination weights $c_{m,k}$ in the adapt-then-combine diffusion strategy. The unknown parameter vector is $\mathbf{w}^0 = \mathrm{rand}(M, 1)/\|\mathrm{rand}(M, 1)\|(M = 32)$ where $\mathrm{rand}(\cdot)$ is a standard uniform distribution (assuming they are the same for all nodes). In addition, the system noise $\theta_k(n)$ is an independent white Gaussian noise with variance $\sigma_{\theta_k}^2$ shown in Fig. 2, and the impulsive noise $\omega_k(n)$ is a signal-to-interference ratio of -30 dB at node k. In order to test the tracking ability of these algorithm, the unknown parameter vector suddenly changes from $\mathbf{w}^0$ to $-\mathbf{w}^0$ at the middle of the iterations. In the following computer simulations, $p_k = 0.01$, $N_w = 16$, $\lambda = 0.99$, $P = 4$, $\tau = 0.9$, $\gamma = 0.12$, $\mu_{\max} = 2$, $\mu_{\min} = 0.0001$ are set. The input signal $x_k(n)$ is generated by a second-order autoregressive system

$$x_k(n) = \kappa_1 x_k(n-1) + \kappa_2 x_k(n-2) + \vartheta_k(n) \tag{43}$$

where $\vartheta_k(n)$ is a zero-mean white Gaussian process with variance $\sigma_{\vartheta_k}^2$ shown in Fig. 2 for all the nodes.

The network mean square deviation (MSD) is defined as

$$\mathrm{MSD}^{net} = \frac{1}{N}\sum_{k=1}^{N}E\left\{\left\|\mathbf{w}^0 - \mathbf{w}_k(n)\right\|^2\right\} \tag{44}$$

which is used to test the proposed algorithms performance regarding convergence rate, steady-state error, and tracking ability, and all simulation results are the average over 100 independent trials.

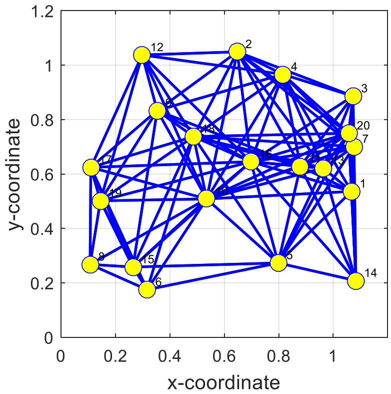


FIGURE 1. Network topology with $N = 20$ nodes.

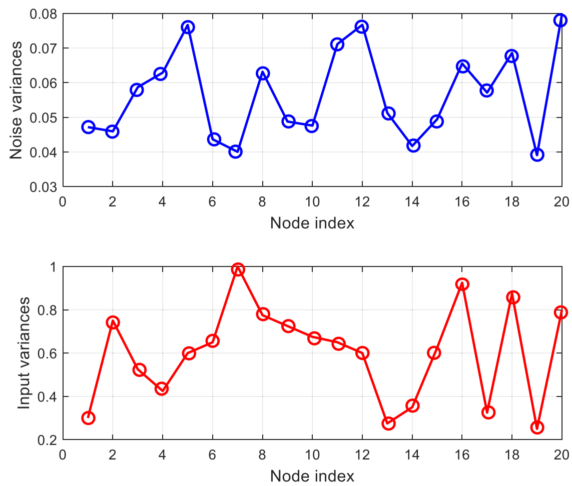


FIGURE 2. Input variances and noise variances of 20 nodes.

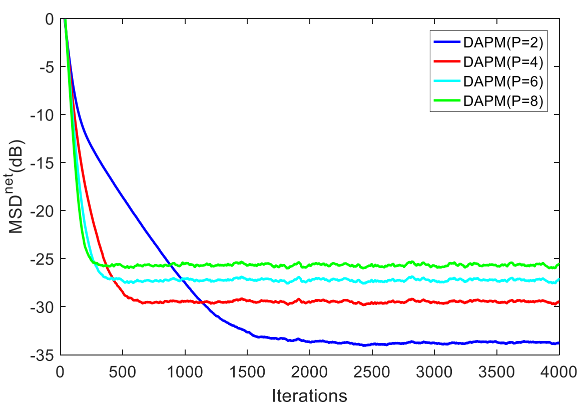


FIGURE 3. The network MSD learning curves of DAPM with $\mu = 0.07$ and different values of P.

A. GAUSSIAN INPUT SIGNAL

The input signal $x_k(n)$ is a white Gaussian noise in which $\kappa_1 = \kappa_2 = 0$ in (43). In Fig. 4, it can be noted that the convergence performance of the DAP algorithm is seriously degraded when the network nodes are disturbed by impulsive noise, and it is observed that the proposed DAPM algorithm has better convergence performance than DAP, DSE-LMS and DAPSA

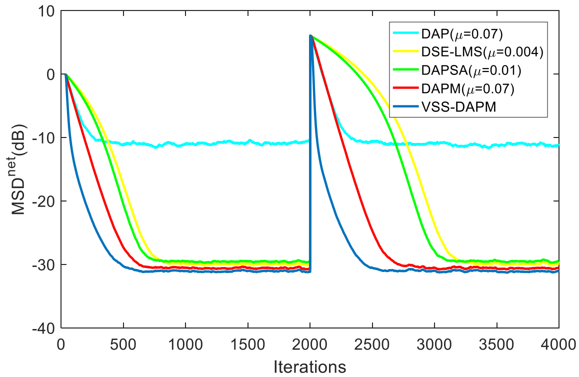


FIGURE 4. The network MSD learning curves of DAP, DSE-LMS, DAPSA, DAPM, and VSS-DAPM algorithms for Gaussian input signal.

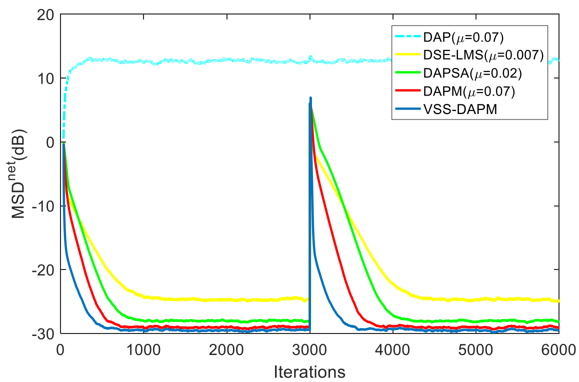


FIGURE 5. The network MSD learning curves of DAP, DSE-LMS, DAPSA, DAPM, and VSS-DAPM algorithms for AR (1) input signal.

algorithms for distributed estimation in the adaptive diffusion networks. Furthermore, the proposed VSS-DAPM algorithm shows better convergence performance than DAPM algorithm and has good tracking performance when the unknown system suddenly changes at the middle of the iterations.

B. AR (1) INPUT SIGNAL

The input signal $x_k(n)$ is a first-order autoregressive (AR (1)) signal in which $\kappa_1 = 0.95, \kappa_2 = 0$ in (43). Fig. 3 shows the convergence performance of DAPM algorithm for different number of projection orders $P = 2, 4, 6$ and 8, in which the step-size is set to $\mu = 0.07$. As far as we know, the larger the projection order P is, the faster the convergence speed is, but the steady-state error increases accordingly. The network MSD convergence curves of these algorithms for AR (1) input signal, which are shown in Fig. 5. It can be concluded that the proposed VSS-DAPM and DAPM algorithms show fastest convergence speed and the smallest steady-state error than DAP, DSE-LMS and DAPSA algorithms for distributed estimation in the adaptive diffusion networks.

C. AR (2) INPUT SIGNAL

The input signal $x_k(n)$ is a second-order autoregressive (AR (2)) signal in which $\kappa_1 = 0.1, \kappa_2 = -0.8$ in (43). The network MSD convergence curves of these algorithms

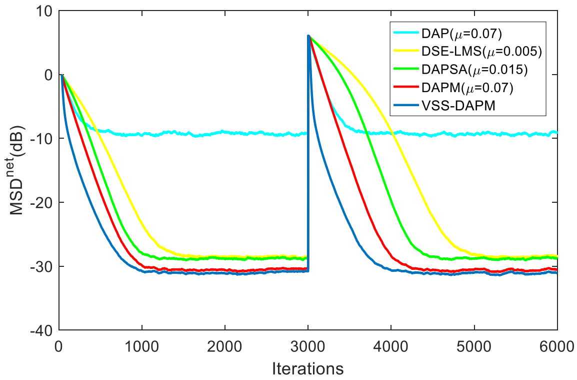


FIGURE 6. The network MSD learning curves of DAP, DSE-LMS, DAPSA, DAPM, and VSS-DAPM algorithms for AR (2) input signal.

for AR (2) input signal, which are shown in Fig. 6. The DAPSA show better convergence speed compared with DSE-LMS algorithm. It can be concluded that the proposed VSS-DAPM and DAPM algorithms show fastest convergence speed and the smallest steady-state error than DAP, DSE-LMS and DAPSA algorithms for distributed estimation in the adaptive diffusion networks.

VI. CONCLUSION

This paper presents a robust diffusion affine projection algorithm for distributed estimation in the adaptive diffusion networks. The robust cost function based on M-estimate function is used to solve the problem that the distributed network nodes are disturbed by impulse noise. In addition, the variable step-size strategy is used in distributed diffusion networks. Computer simulations have shown that the proposed algorithm has good robustness, tracking stability, convergence performance, and lower network mean square deviation than DAP, DSE-LMS and DAPSA algorithms. In the future, in order to develop the distributed algorithm with faster convergence speed and lower steady-state error, it is particularly important to establish cost (objective) function under constraints [35], [36] and some important methods are also worth exploring and studying.

REFERENCES

[1] A. H. Sayed, S.-Y. Tu, J. Chen, X. Zhao, and Z. J. Towfic, "Diffusion strategies for adaptation and learning over networks: An examination of distributed strategies and network behavior," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 155–171, May 2013.

[2] C. G. Lopes and A. H. Sayed, "Incremental adaptive strategies over distributed networks," *IEEE Trans. Signal Process.*, vol. 55, no. 8, pp. 4064–4077, Aug. 2007.

[3] C. G. Lopes and 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.

[4] J. Chen and A. H. Sayed, "Diffusion adaptation strategies for distributed optimization and learning over networks," *IEEE Trans. Signal Process.*, vol. 60, no. 8, pp. 4289–4305, Aug. 2012.

[5] F. Chen and X. Shao, "Broken-motifs diffusion LMS algorithm for reducing communication load," *Signal Process.*, vol. 133, pp. 213–218, Apr. 2017.

[6] X. Shao and F. Chen, "Complementary performance analysis of general complex-valued diffusion LMS for noncircular signals," *Signal Process.*, vol. 160, pp. 237–246, Jul. 2019.

[7] F. S. Cattivelli and A. H. Sayed, "Diffusion LMS strategies for distributed estimation," *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1035–1048, Mar. 2010.

[8] J. Ni and J. Yang, "Variable step-size diffusion least mean fourth algorithm for distributed estimation," *Signal Process.*, vol. 122, pp. 93–97, May 2016.

[9] F. S. Cattivelli, C. G. Lopes, and A. H. Sayed, "Diffusion recursive least-squares for distributed estimation over adaptive networks," *IEEE Trans. Signal Process.*, vol. 56, no. 5, pp. 1865–1877, May 2008.

[10] L. Li and J. A. Chambers, "Distributed adaptive estimation based on the APA algorithm over diffusion networks with changing topology," in *Proc. IEEE/SP 15th Workshop Stat. Signal Process.*, Cardiff, U.K., Aug./Sep. 2009, pp. 757–760.

[11] J. Ni, J. Chen, and X. Chen, "Diffusion sign-error LMS algorithm: Formulation and stochastic behavior analysis," *Signal Process.*, vol. 128, pp. 142–149, Nov. 2016.

[12] J.-H. Seo, S. M. Jung, and P. Park, "Diffusion proportionate affine projection sign algorithm for distributed estimation over network," in *Proc. 14th Int. Conf. Elect. Eng./Electron., Comput., Telecommun. Inf. Technol. (ECTI-CON)*, Phuket, Thailand, Jun. 2017, pp. 636–639.

[13] F. Chen, T. Shi, S. Duan, L. Wang, and J. Wu, "Diffusion least logarithmic absolute difference algorithm for distributed estimation," *Signal Process.*, vol. 142, pp. 423–430, Jan. 2018.

[14] F. Chen, X. Li, S. Duan, L. Wang, and J. Wu, "Diffusion generalized maximum correntropy criterion algorithm for distributed estimation over multitask network," *Digit. Signal Process.*, vol. 81, pp. 16–25, Oct. 2018.

[15] W. Huang, L. Li, Q. Li, and X. Yao, "Diffusion robust variable step-size LMS algorithm over distributed networks," *IEEE Access*, vol. 6, pp. 47511–47520, 2018.

[16] M. O. B. Saeed, A. Zerguine, and S. A. Zunmo, "A variable step-size strategy for distributed estimation over adaptive networks," *EURASIP J. Adv. Signal Process.*, vol. 2013, Aug. 2013, Art. no. 135.

[17] H. S. Lee, S. E. Kim, J. W. Lee, and W. J. Song, "A variable step-size diffusion LMS algorithm for distributed estimation," *IEEE Trans. Signal Process.*, vol. 63, no. 7, pp. 1808–1820, Apr. 2015.

[18] L. Shi and H. Zhao, "Variable step-size distributed incremental normalised LMS algorithm," *Electron. Lett.*, vol. 52, no. 7, pp. 519–521, Apr. 2016.

[19] J. W. Yoo, I. S. Song, J. W. Shin, and P. G. Park, "A variable step-size diffusion affine projection algorithm," *Int. J. Commun. Syst.*, vol. 29, no. 5, pp. 1012–1025, Mar. 2016.

[20] Y. Yu and H. Zhao, "Robust incremental normalized least mean square algorithm with variable step sizes over distributed networks," *Signal Process.*, vol. 144, pp. 1–6, Mar. 2018.

[21] R. Abdolee, V. Vakilian, and B. Champagne, "Tracking performance and optimal adaptation step-sizes of diffusion-LMS networks," *IEEE Trans. Control Netw. Syst.*, vol. 5, no. 1, pp. 67–78, Mar. 2018.

[22] M. S. E. Abadi and M. S. Shafiee, "Distributed estimation over an adaptive diffusion network based on the family of affine projection algorithms," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 5, no. 2, pp. 234–247, Jun. 2019.

[23] S. C. Chan and Y. Zhou, "On the performance analysis of the least mean M-estimate and normalized least mean M-estimate algorithms with Gaussian inputs and additive Gaussian and contaminated Gaussian noises," *J. Signal Process. Syst.*, vol. 60, no. 1, pp. 81–103, 2010.

[24] Y. Yu, H. Zhao, and B. Chen, "Steady-state mean-square-deviation analysis of the sign subband adaptive filter algorithm," *Signal Process.*, vol. 120, pp. 36–42, Mar. 2016.

[25] Y. Zhou, S. C. Chan, and K. L. Ho, "New sequential partial-update least mean M-estimate algorithms for robust adaptive system identification in impulsive noise," *IEEE Trans. Ind. Electron.*, vol. 58, no. 9, pp. 4455–4470, Sep. 2011.

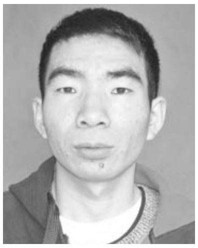
[26] Y. Yu and H. Zhao, "Incremental M-estimate-based least-mean algorithm over distributed network," *Electron. Lett.*, vol. 52, no. 14, pp. 1270–1272, 2016.

[27] P. Song and H. Zhao, "Affine-projection-like M-estimate adaptive filter for robust filtering in impulse noise," *IEEE Trans. Circuits Syst., II, Exp. Briefs*, to be published. doi: 10.1109/TCSII.2019.2897620.

[28] N. Takahashi, I. Yamada, and A. H. Sayed, "Diffusion least-mean squares with adaptive combiners: Formulation and performance analysis," *IEEE Trans. Signal Process.*, vol. 58, no. 9, pp. 4795–4810, Sep. 2010.

[29] P. D. Lorenzo and A. H. Sayed, "Sparse distributed learning based on diffusion adaptation," *IEEE Trans. Signal Process.*, vol. 61, no. 6, pp. 1419–1433, Mar. 2013.

[30] L. Shi and H. Zhao, "Diffusion leaky zero attracting least mean square algorithm and its performance analysis," *IEEE Access*, vol. 6, pp. 56911–56923, 2018.
[31] A. H. Sayed, *Adaptive Filters*. New York, NY, USA: Wiley, 2011.
[32] A. H. Sayed, "Adaptation, learning, and optimization over networks," *Found. Trends Mach. Learn.*, vol. 7, nos. 4–5, pp. 311–801, Jul. 2014.
[33] R. Price, "A useful theorem for nonlinear devices having Gaussian inputs," *IRE Trans. Inf. Theory*, vol. 4, no. 2, pp. 69–72, Jun. 1958.
[34] E. McMahon, "An extension of Price's theorem (Corresp.)," *IEEE Trans. Inf. Theory*, vol. IT-10, no. 2, p. 168, Apr. 1964.
[35] C. Li, M. Cerrada, D. Cabrera, R. V. Sanchez, F. Pacheco, G. Ulutagay, and J. V. de Oliveira, "A comparison of fuzzy clustering algorithms for bearing fault diagnosis," *J. Intell. Fuzzy Syst.*, vol. 34, no. 6, pp. 3565–3580, Jun. 2018.
[36] L. Jianyu, S. Zhenzhong, P. M. Pardalos, H. Ying, Z. Shaohui, and L. Chuan, "A hybrid multi-objective genetic local search algorithm for the prize-collecting vehicle routing problem," *Inf. Sci.*, vol. 478, pp. 40–61, Apr. 2019.



PUCHA SONG was born in Anhui, China, in 1989. He received the B.E. degree from the School of Electrical Engineering, Tongling University, Tongling, China, in 2014, and the master's degree from the School of Electrical Engineering, Southwest Jiaotong University, in 2017, where he is currently pursuing the Ph.D. degree in signal and information processing. His current research interests include active noise control, statistic signal processing, frequency estimation, and distributed filtering network.

HAIQUAN ZHAO was born in Henan, China, in 1974. He received the B.S. degree in applied mathematics in 1998, and the M.S. and Ph.D. degrees in signal and information processing from Southwest Jiaotong University, Chengdu, China, in 2005 and 2011, respectively. Since 2012, he has been a Professor with the School of Electrical Engineering, Southwest Jiaotong University. He is the author or coauthor of more than 130 journal articles and the owner of 31 invention patents. His current research interests include adaptive filtering algorithm, power system frequency estimation, adaptive network, Volterra filter, active noise control, nonlinear system identification, and chaotic signal processing. He is a senior member of the IEEE Society. He has served as an active Reviewer for several IEEE Transactions, *The Institution of Engineering and Technology*, and other international journals.

XIANGPING ZENG was born in Sichuan, China, in 1974. She received the B.E. degree from Southwest Jiaotong University, Chengdu, China, in 1998, the M.S. degree from the Graduate School of Chinese Academy of Sciences, Beijing, China, in 2006, and the Ph.D. degree from Southwest Jiaotong University, in 2014.

Since 2015, she has been an Associate Professor. Her current research interests include video image processing and intelligent information processing.