Chaos

Distributed event-triggered adaptive partial diffusion strategy under dynamic network topology

Cite as: Chaos **30**, 063103 (2020); https://doi.org/10.1063/5.0007405
Submitted: 12 March 2020 . Accepted: 11 May 2020 . Published Online: 01 June 2020

Minyu Feng, Shuwei Deng, Feng Chen, and Jürgen Kurths

Distributed event-triggered adaptive partial diffusion strategy under dynamic network topology

Cite as: Chaos **30**, 063103 (2020); doi: 10.1063/5.0007405
Submitted: 12 March 2020 · Accepted: 11 May 2020 ·
Published Online: 1 June 2020

Minyu Feng,[1] Shuwei Deng,[1,a)] Feng Chen,[1,b),c)] and Jürgen Kurths[2,d)]

AFFILIATIONS

[1]College of Artificial Intelligence, Southwest University, Chongqing 400715, China
[2]Potsdam Institute for Climate Impact Research, 14473 Potsdam, Germany

[a)]**Also at:** College of Electronic and Information Engineering, Southwest University, Chongqing 400715, China and Key Laboratory of Nonlinear Circuits and Intelligent Information Processing, Chongqing, China.
[b)]**Also at:** College of Electronic and Information Engineering, Southwest University, Chongqing 400715, China and Brain-Inspired Computing and Intelligent Control Key Laboratory, Southwest University, Chongqing 400715, China.
[c)]**Author to whom correspondence should be addressed:** fengchen.uestc@gmail.com
[d)]**Also at:** Department of Physics, Humboldt University, Berlin 12489, Germany.

ABSTRACT

In wireless sensor networks, the dynamic network topology and the limitation of communication resources may lead to degradation of the estimation performance of distributed algorithms. To solve this problem, we propose an event-triggered adaptive partial diffusion least mean-square algorithm (ET-APDLMS). On the one hand, the adaptive partial diffusion strategy adapts to the dynamic topology of the network while ensuring the estimation performance. On the other hand, the event-triggered mechanism can effectively reduce the data redundancy and save the communication resources of the network. The communication cost analysis of the ET-APDLMS algorithm is given in the performance analysis. The theoretical results prove that the algorithm is asymptotically unbiased, and it converges in the mean sense and the mean-square sense. In the simulation, we compare the mean-square deviation performance of the ET-APDLMS algorithm and other different diffusion algorithms. The simulation results are consistent with the performance analysis, which verifies the effectiveness of the proposed algorithm.

In distributed algorithms, nodes cooperate to complete parameter estimation, which plays a vital role in signal processing. The existing traditional distributed algorithms have high estimation performance but often ignore the problem of high communication costs. Therefore, in recent years, there have been many studies on communication cost reduction algorithms (such as partial, data-selective). In this article, we propose the event-triggered adaptive partial diffusion least mean-square (ET-APDLMS) algorithm to reduce the communication costs of the network. Besides, different from the traditional communication cost reduction algorithm, the ET-APDLMS algorithm considers that the actual environment (such as the ocean and atmosphere) on the impact of network topology structure, makes the algorithm adapt to different application environments. The theoretical results prove that the algorithm is asymptotically unbiased, and it converges in the mean sense and the mean-square sense. In the simulation, we compared the mean-square deviation (MSD) performance of the ET-APDLMS algorithm and other different diffusion algorithms.

I. INTRODUCTION

In recent years, wireless sensor networks (WSNs) have been widely used in different fields, such as power grids and environmental detection. For WSNs, there are many methods for estimating unknown network parameters.[1,2] Centralized solutions and distributed solutions are concerned because of their better estimation performance. In the centralized strategy, every node in the network transmits its data to the center node for processing. In this method, the central node has to undertake many computing tasks. Once the

central node is damaged, or its energy is exhausted, it may cause the entire sensor network to collapse. Fortunately, in the distributed strategy,[3–6] local nodes cooperate to complete the estimation task of the network, which solves the problem of centralized strategy.

Distributed solutions can be divided into three categories: incremental,[7–11] consensus,[12–17] and diffusion.[18–29] For the diffusion strategy, nodes communicate with each other through a broadcast. The strategy does not rely on any central control mechanism or protocol, which makes the diffusion strategy more robust to the communication link damage or node damage. Due to these advantages, diffusion strategies have been widely used in WSNs. Cattivelli and Sayed conducted a comprehensive and pioneering study based on diffusion and proposed the Adapt Then Combine (ATC) and Combine Then Adapt (CTA) strategies and carried out a detailed theoretical analysis in the literature[3] and made a detailed theoretical analysis in the paper. In order to deal with impulsive noise, Chen and his co-workers conduct in-depth research and propose novel and efficient diffusion correntropy algorithms in the literature.[34,35]

In WSNs, nodes are often limited by computing capability and electrical power. When a node performs a distributed task, most of the power consuming action is data transmission. Therefore, long-term network stability must decrease internode communication. Many algorithms try to reduce internode communication, such as partial diffusion[26,30,31] and data-selective.[32]

The literature[32] proposes a data-selective strategy based on an estimate error, which reduces the data redundancy by setting appropriate thresholds, thereby achieving the purpose of saving communication resources. In the literature,[26] a partial diffusion strategy based on the LMS algorithm is proposed, in which nodes use a subset of the intermediate estimate for parameter estimation, thereby reducing the communication cost of the network. However, the traditional partial diffusion algorithm does not consider the impact of dynamic network topology on estimation performance, and the waste of communication resources is caused by data redundancy.

In a real-world environment, the topology of the network may change at any time (e.g., the marine environment). Dynamic network topology means that the neighbor nodes of each node are changing every moment (As shown in Fig. 1, node 2 starts as the neighbor node of node 5, and with the change of environment, node 2 becomes the neighbor node of node 6.), which often affects the estimation performance of the traditional partial diffusion strategy. In order to solve this problem, this paper proposes an adaptive partial diffusion strategy. It makes the algorithm adapting to the dynamic network topology. At the same time, in order to reduce network data redundancy, an event-triggered mechanism is designed in the adaptive partial diffusion strategy, and the ET-APDLMS algorithm is proposed.

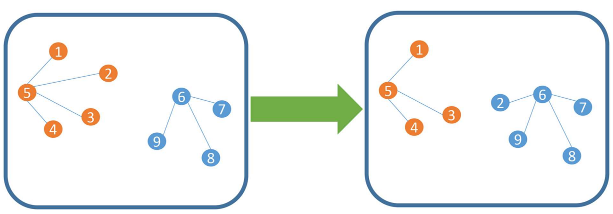

FIG. 1. Dynamic network topology. Red represents the neighbor node of node 5, and blue represents the neighbor node of node 6.

TABLE I. Abbreviations and symbols in the paper.

LMS	Least mean square
CTA	Combine then adapt
ATC	Adapt then combine
DLMS	Diffusion LMS
ET-APDLMS	Event-triggered adaptive partial diffusion LMS
$[\mathcal{N}_k]$	The number of neighbor nodes of the node k
F	The average number of neighbors of each node

In the performance analysis, we study the communication cost of the ET-APDLMS algorithm and prove that the algorithm is stable in the mean and mean-square sense. In the simulation, we compare the estimation performance of the ET-APDLMS algorithm, the PDLMS algorithm, and the DLMS algorithm. The agreement between the simulation results and the performance analysis results confirms the effectiveness of the ET-APDLMS algorithm.

The main contributions of this paper are as follows:

A. An adaptive partial diffusion strategy is designed for dynamic network topology environments.
B. In order to reduce the network data redundancy, an event-triggered mechanism is designed in the adaptive partial diffusion strategy, and the ET-APDLMS algorithm is proposed.
C. The stability and communication cost of the proposed algorithm is given in the performance analysis.

The system model and preparatory work are introduced in Sec. II. We describe the ET-APDLMS algorithm in Sec. III, the performance analysis of the algorithm is given in Sec. IV, the simulations of the proposed algorithm are presented in Sec. V, and the conclusions are given in Sec. VI.

There are many abbreviations and symbols in the paper, which are listed in Table I for easy reading.

II. SYSTEM MODEL AND A REVIEW OF TRADITIONAL ALGORITHMS

In this section, we describe the data model of the algorithm and review the traditional distributed algorithm and the traditional partial algorithm.

A. Data model

The communication between nodes is described by an undirected graph $\mathbb{G} = (N, \xi)$, which consists of the set of nodes N and the set of edges $\xi \subseteq N \times N$. There is an edge $(i, j) \in \xi$, which means node i exchanges information with node j. The neighbor node of the node i is represented by $\mathcal{N}_i \triangleq \{j \in N | (i, j) \in \xi\}$.

Each node has an input vector, $x_{k,n} \in \mathbb{R}^{L\times 1}$, and an output signal, $d_{k,n} \in \mathbb{R}$, which can be related via a linear model,

$$d_{k,n} = x_{k,n}^T w^o + v_{k,n}, \tag{1}$$

where $v_{k,n}$ is spatially independent zero-mean Gaussian noise, whose variance is expressed as $\sigma_{v,k}^2 = \mathbb{E}\left[v_{k,n}^2\right]$, n is the time index, and T

denotes matrix/vector transposition. The optimal solution of the network is represented by w^o.

B. Distributed LMS

We seek the optimal estimate w^o by minimizing the global cost function,

$$J^{glob}(w) \triangleq \sum_{k=1}^{N} E\left|d_{k,n} - x_{k,n}^T w\right|^2. \quad (2)$$

In the distributed estimation, the global optimal estimate is obtained by finding the local optimal estimate. It can be related to the following linear relationship:

$$J^{glob}(w) = \sum_{l=1}^{N} J_l^{loc}(w). \quad (3)$$

The local cost function of the network can be expressed as

$$J_k^{loc}(w) = \sum_{l\in\mathcal{N}_k} c_{l,k} E\left|d_{l,n} - x_{l,n}^T w\right|; \quad (4)$$

here, $\{c_{l,k}\}$ are non-negative real entries for a $N \times N$ matrix C,

$$c_{l,k} = 0 \, if \, l \notin \mathcal{N}_k, C\mathbb{1}=\mathbb{1}, \mathbb{1}^T C = \mathbb{1}^T, \quad (5)$$

where $\mathbb{1}$ denotes the $N \times 1$ vector, with unit entries.

In order to solve the cost function equation (4), the Adapt-then-Combine (ATC) diffusion LMS algorithm is proposed in the literature.[3] ATC is as follows:

$$\begin{cases} \psi_{k,n} = w_{k,n-1} + \mu_k \sum_{l\in\mathcal{N}_k} c_{l,k} x_{l,n}(d_{l,n} - x_{l,n}^T w_{k,n-1}) & \text{adapt}, \\ w_{k,n} = \sum_{l\in\mathcal{N}_k} a_{l,k}\psi_{l,n} & \text{combine}, \end{cases} \quad (6)$$

where $\{a_{l,k}\}$ are non-negative real entries for an $N \times N$ matrix A,

$$a_{l,k} = 0 \, if \, l \notin \mathcal{N}_k, A\mathbb{1}=\mathbb{1}, \mathbb{1}^T A = \mathbb{1}^T. \quad (7)$$

The implementation of the ATC is given in Fig. 2.

The traditional distributed LMS algorithm achieves a higher estimation accuracy. However, in the process of transmitting data, a large amount of redundant data is transmitted, which causes the waste of communication resources.

C. Traditional partial diffusion algorithm

In order to reduce the communication cost of the distributed diffusion algorithm, a partial diffusion algorithm was proposed in the literature.[26] Each node uses a subset of the intermediate estimate of neighbor nodes for the parameter estimation, and the implementation process of the algorithm is

$$\begin{cases} \psi_{k,n} = w_{k,n-1} + \mu_k \sum_{l\in\mathcal{N}_k} c_{l,k} x_{l,n}(d_{l,n} - x_{l,n}^T w_{k,n-1}) & \text{adapt}, \\ w_{k,n} = a_{kk}\psi_{k,n} + \sum_{l\in\mathcal{N}_k/\{k\}} a_{l,k}[\mathrm{A}_{l,n}\psi_{l,n} + (\mathrm{I}_L - \mathrm{A}_{l,n})\psi_{k,n}] & \text{combine}. \end{cases} \quad (8)$$

Here, the $L \times L$ diagonal entry-selection matrix A_{kn} has M ones and $L - M$ zeros on its diagonal. The matrix A_{kn} selects the M entries of the intermediate estimate that the node k transmits at the time

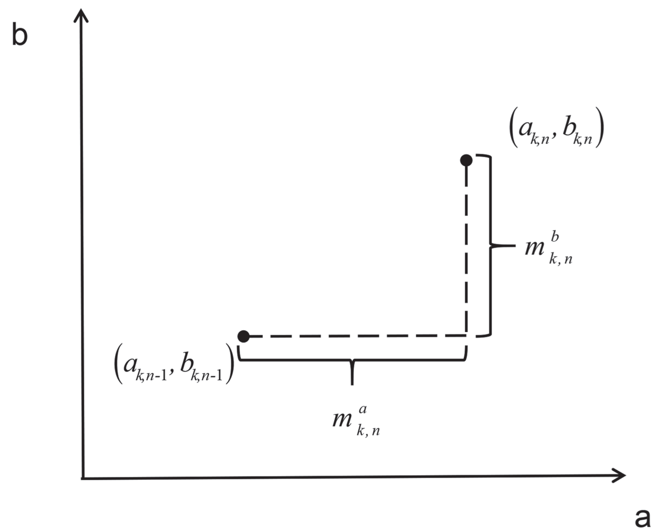

FIG. 2. The implementation process of ATC. The algorithm steps and the information exchange process for each step are explained in the figure.

instant n. The traditional partial algorithm reduces the communication cost of the algorithm, but the estimation accuracy of the algorithm is low. Moreover, the traditional partial algorithm does not consider the effect of a dynamic topology environment.

The traditional distributed partial diffusion algorithm does not consider the influence of dynamic network topology on the estimation performance and the impact of data redundancy on the communication costs. Considering the shortcomings of the traditional distributed LMS algorithm and traditional local algorithms, we propose the ET-APDLMS algorithm.

III. ET-APDLMS ALGORITHM

A. Dynamic network topology model

In a real environment (such as the ocean, atmosphere, etc.), the position of the sensor moves as the environment changes, which leads to the change of the network topology. In this paper, we design a dynamic topology model to simulate topology changes in a real environment. There are N nodes in the network, and the location of each sensor node is expressed as $\left(a_{k,n}, b_{k,n}\right)$, $k \in \{1, 2, \ldots, N\}$, and n is the time index. The dynamic network topology model is defined as

$$a_{k,n} = a_{k,n-1} + m^a_{k,n},$$
$$b_{k,n} = b_{k,n-1} + m^b_{k,n},$$

where $\{m^a_{k,n}, m^b_{k,n}\}$, respectively, represent the moving distance of $\left(a_{k,n}, b_{k,n}\right)$ relative to $\left(a_{k,n-1}, b_{k,n-1}\right)$. The dynamic network topology model is shown in Fig. 3.

B. Algorithm design

The ET-APDLMS algorithm considers the impact of dynamic network topology. According to the dynamic network model

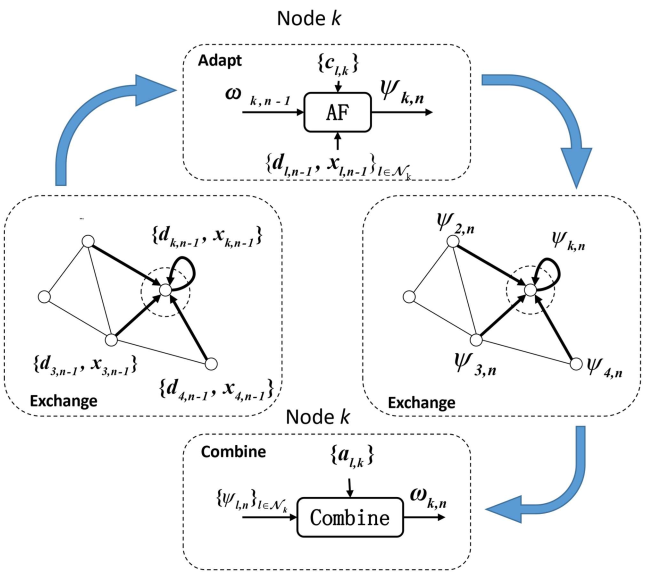

FIG. 3. Dynamic network topology model. $m^a_{k,n}$ and $m^b_{k,n}$ represent the moving distance of the node k in the a direction and the b direction, respectively.

designed above, we find that the number of neighbor nodes of each node changes with the movement of nodes. Therefore, we add the constraint $\mathbb{H}$ to the cost function and design the new cost function as follows:

$$J_k^{loc}(w) = \sum_{l\in\mathcal{N}_k} c^n_{l,k} E\left|d_{l,n} - x^T_{l,n} w\right|$$

$$\text{s.t. } (M, [\mathcal{N}_k]) \in \mathbb{H}, \tag{9}$$

$$c^n_{l,k} = 0 \; if \, l \notin \mathcal{N}_k, C^n\mathbb{1}{=}\mathbb{1}, \mathbb{1}^T C^n = \mathbb{1}^T. \tag{10}$$

Here, M denotes the dimension of the intermediate estimate subset of neighbor nodes. We use $[\mathcal{N}_k]$ to denote the number of neighbor nodes of the node k. $(M, [\mathcal{N}_k])$ need to satisfy the condition $\mathbb{H}$, and $\{c^n_{l,k}\}$ are time-varying non-negative real entries for an $N \times N$ matrix C^n.

To minimize the cost function in Eq. (9), the diffusion algorithm can be implemented in two scenarios (ATC and CTA). In the literature,[3] it has been proved that the ATC algorithm makes more effective use of data than the CTA algorithm; therefore, the ATC algorithm is superior to the CTA algorithm. In this paper, we mainly consider the ATC algorithm.

1. Adaptation step

The intermediate estimate $\psi_{k,n}$ of the system can be obtained by the adaptive step, and the process is

$$\psi_{k,n} = w_{k,n-1} + \mu_k \sum_{l\in\mathcal{N}_k} c^n_{l,k} x_{l,n}(d_{l,n} - x^T_{l,n} w_{k,n-1}). \tag{11}$$

In this step, information is transmitted between neighbor nodes to obtain the intermediate estimate of the nodes. However, there is a large amount of data redundancies in this step, which will increase the communication cost of the system.

2. Event-triggered mechanism

To avoid waste of communication resources caused by data redundancies, we design an event-triggered mechanism, which decides for each node whether the current intermediate estimate is sent out to its neighbors or not.

The trigger event $E_{k,n}$ is designed as

$$E_{k,n} : \left\|\psi_{k,n} - \psi_{k,n_{et}}\right\| > \frac{1}{(n+1)^\rho}, \tag{12}$$

where ρ is a positive scalar, while $0 < \rho < 1$ and n_{et} is the latest triggering time. According to $E_{k,n}$, the intermediate estimates that the node k transmits at the time instant n as

$$\psi_{k,n} = \gamma_{k,n}\psi_{k,n_{et}} + (1 - \gamma_{k,n})\psi_{k,n}; \tag{13}$$

here,

$$\gamma_{k,n} = \begin{cases} 0 & \text{if } E_{k,n} \text{ occurs,} \\ 1, & \text{otherwise.} \end{cases} \tag{14}$$

When Eq. (12) is not satisfied, $\psi_{k,n_{et}}$ will be used as the intermediate estimator of the node to complete the estimation task of the algorithm and to reduce the network communication cost. The implementation process is shown as Eqs. (13) and (14).

In Fig. 4, the implementation process of the event trigger mechanism is given. After the algorithm completes the adaptive process, the intermediate estimation of the algorithm is obtained. We use $E_{k,n}$ to determine whether the current intermediate estimate is meaningful and then decide to use the current intermediate estimate $\psi_{k,n}$ or the event-triggered intermediate estimate $\psi_{k,n_{et}}$ to complete the estimate.

3. Combination step

In order to adapt to the dynamic changes of the network topology, we design an adaptive partial strategy, which is implemented by

$$w_{k,n} = a^n_{k,k}\psi_{k,n} + \sum_{l\in\mathcal{N}_k/\{k\}} a^n_{l,k}[H_{l,n}\psi_{l,n} + (I_L - H_{l,n})\psi_{k,n}]$$

$$\text{s.t. } (M, [\mathcal{N}_k]) \in \mathbb{H}, \tag{15}$$

$$a^n_{l,k} = 0 \text{ if } l \notin \mathcal{N}_k, A^n\mathbb{1}{=}\mathbb{1}, \mathbb{1}^T A^n = \mathbb{1}^T. \tag{16}$$

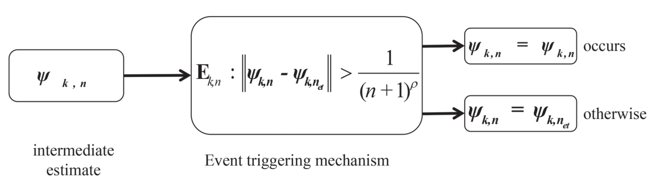

FIG. 4. The event-triggered mechanism.

Here, the $L \times L$ diagonal entry-selection matrix $\mathrm{H}_{k,n}$ has M ones and $L - M$ zeros on its diagonal. $\{a_{l,k}^n\}$ are time-varying non-negative real entries for an $N \times N$ matrix A^n. The matrix $\mathrm{H}_{k,n}$ selects the M entries of the intermediate estimate that the node k transmits at the time instant n. $(M, [\mathcal{N}_k])$ needs to satisfy the condition $\mathbb{H}$, which is designed by

$$\mathbb{H} : \mathrm{M} = \left\lceil \frac{th}{[\mathcal{N}_k]} \right\rceil, \tag{17}$$

where th is a threshold designed according to the network scale and $\lceil \cdot \rceil$ means taking the upper bound.

The matrix $\mathrm{H}_{l,n}$ is expressed as follows:

$$\mathrm{H}_{l,n}\!: \left[\underbrace{\begin{matrix} 1,0,0,0,\ldots,0 \\ 0,1,0,0,\ldots,0 \\ 0,0,1,0,\ldots,0 \\ 0,0,0,1,\ldots,0 \\ \vdots \ddots \\ 0,0,0,0,\ldots,0 \end{matrix}}_{\text{M ones and L}-\text{M zeros on its diagonal}} \right]. \tag{18}$$

This paper combines the event-triggered mechanism and the adaptive partial diffusion strategy to propose the ET-APDLMS algorithm. The implementation process of the algorithm is shown in Table II and Fig. 5.

IV. PERFORMANCE ANALYSIS

In this section, we will analyze the mean and mean-square performance and communication cost of the ET-PDLMS algorithm. To facilitate the analysis, we use the following assumptions:

TABLE II. ET-APDLMS algorithm.

Initialize:
Start with $\{w_{k,0} = 0\}$, for all k, given non-negative real coefficients $\{c_{l,k}^n, a_{l,k}^n\}$ satisfying Eqs. (10) and (16), presetting ρ, $0 < \rho < 1$, set thresholds th the network scale nodes, for each time $n > 0$, repeat:

Adaptation:

$$\psi_{k,n} = w_{k,n-1} + \mu_k \sum_{l \in \mathcal{N}_k} c_{l,k}^n x_{l,n}(d_{l,n} - x_{l,n}^T w_{k,n-1})$$

Event-trigger:

$$E_{k,n} : \left\| \psi_{k,n} - \psi_{k,n_{et}} \right\| > \frac{1}{(n+1)^{\rho}}$$

$$\gamma_{k,n} = \begin{cases} 0 & \text{if } \ E_{k,n} \ \text{ occurs} \\ 1, & \quad \text{otherwise} \end{cases}$$

$$\psi_{k,n} = \gamma_{k,n}\psi_{k,n_{et}} + (1 - \gamma_{k,n})\psi_{k,n}$$

Combination:
Setting matrix $\mathrm{H}_{l,n}$ according to $\mathbb{H}$

$$\mathbb{H} : \mathrm{M} = \left\lceil \frac{th}{[\mathcal{N}_k]} \right\rceil$$

$$w_{k,n} = a_{k,k}^n \psi_{k,n} + \sum_{l \in \mathcal{N}_k/\{k\}} a_{l,k}^n [H_{l,n}\psi_{l,n} + (I_L - H_{l,n})\psi_{k,n}]$$

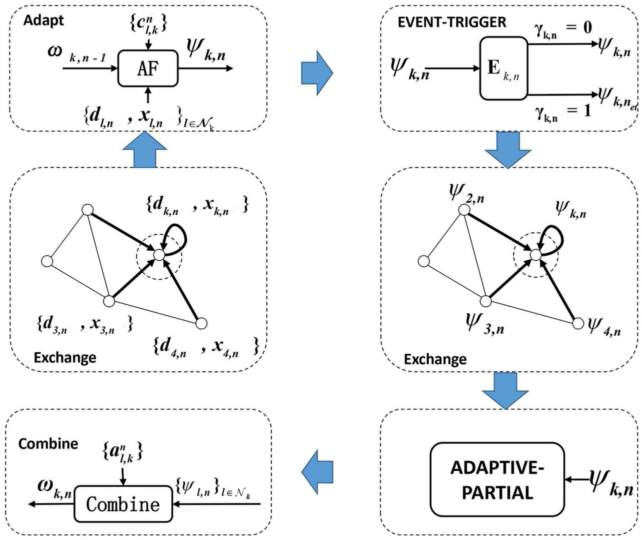

FIG. 5. Implementation process of ET-APDLMS. The steps of the algorithm and the information exchange process and the event triggering mechanism of each step are described in the figure.

A1. The elements of the input vector $x_{k,n}$ are spatially and temporally independent, and their covariances are expressed as

$$\mathrm{E}\left[x_{k,n} x_{k,n}^T\right] = R_k \in \mathbb{R}^{L\times L} \quad \forall\, k, n. \tag{19}$$

A2. The elements of the noise vector $v_{k,n}$ are spatially and temporally independent,

$$E\left[v_{k,n}\right] = 0 \quad \text{and} \quad E[v_{k,n}^2] = \zeta_k^2 \in \mathbb{R} \quad \forall\, k, n. \tag{20}$$

A3. The step size parameter μ_k is small enough that its squared value can be ignored.

A. Estimate-error equation

Define the intermediate estimate-error vector as

$$\tilde{\psi}_{k,n} = w^o - \psi_{k,n} \tag{21}$$

and the estimation-error of node k as

$$\tilde{w}_{k,n} = w^o - w_{k,n}. \tag{22}$$

Bring Eqs. (1)–(11) and then subtract w^o from both sides to get the intermediate estimate-error vector,

$$\tilde{\psi}_{\mathrm{k,n}} = \left(\mathrm{I_L} - \mu_k \sum_{l \in \mathcal{N}_k} c_{l,k}^n x_{k,n} x_{k,n}^{\mathrm{T}} \right) \tilde{w}_{k,n-1} - \mu_k \sum_{l \in \mathcal{N}_k} c_{l,k}^n x_{k,n} v_{k,n} \quad \forall\, k. \tag{23}$$

According to Eqs. (10) and (16), rewrite the estimate of node k as

$$w_{k,n} = \left(\mathrm{I}_L - \sum_{l \in \mathcal{N}_k/\{k\}} a_{l,k}^n H_{l,n} \right) \psi_{k,n} + \sum_{l \in \mathcal{N}_k/\{k\}} a_{l,k}^n H_{l,n} \psi_{l,n}. \tag{24}$$

Replace the estimate in Eq. (24) with w^o to get

$$w^o = \left(\mathrm{I}_L - \sum_{l\in\mathcal{N}_k/\{k\}} a^n_{l,k}\mathrm{H}_{l,n} \right) w^o + \sum_{l\in\mathcal{N}_k/\{k\}} a^n_{l,k}\mathrm{H}_{l,n} w^o. \tag{25}$$

The estimate-error vector can be obtained by subtracting Eq. (24) from Eq. (25),

$$\tilde{w}_{k,n} = \left(I_L - \sum_{l\in\mathcal{N}_k/\{k\}} a^n_{l,k}H_{l,n} \right) \tilde{\psi}_{k,n} + \sum_{l\in\mathcal{N}_k/\{k\}} a^n_{l,k}H_{l,n}\tilde{\psi}_{l,n}. \tag{26}$$

If the intermediate estimate $\psi_{k,n}$ of the node k satisfies $\left\| \psi_{k,n} - \psi_{k,n_{et}} \right\| < \frac{1}{(n+1)^\rho}$ at the time n, the intermediate estimate of node k is expressed as

$$\psi_{k,n} = \psi_{k,n_{et}}. \tag{27}$$

Therefore, we define the stacking vector of the intermediate estimate-error vectors for all nodes as

$$\tilde{\varsigma}_n = \begin{bmatrix} \tilde{\psi}_{1,n} \\ \vdots \\ \tilde{\psi}_{k,n_{et}} \\ \vdots \\ \tilde{\psi}_{N,n} \end{bmatrix}. \tag{28}$$

To facilitate subsequent analysis, we represent the intermediate estimate-error stacking matrix $\tilde{\varsigma}_n$ as

$$\tilde{\varsigma}_n = (\mathrm{I}_{\mathrm{LK}} - \mathrm{M}\mathrm{X}_n)\tilde{\omega}_n - \mathrm{M}\mathrm{g}_n. \tag{29}$$

For Eq. (29), we define

$$\mathrm{M} = \mathrm{blockdiag}\{\mu_1 I_L, \ldots, \mu_k I_L, \ldots, \mu_N I_L\},$$

$$\mathrm{g}_n = \begin{bmatrix} \sum_{l\in\mathcal{N}_1} c^n_{l,n} x_{l,n} v_{l,n} \\ \vdots \\ \sum_{l\in\mathcal{N}_k} c^n_{l,n_{et}} x_{l,n_{et}} v_{l,n_{et}} \\ \vdots \\ \sum_{l\in\mathcal{N}_N} c^n_{l,n} x_{l,n} v_{l,n} \end{bmatrix},$$

and

$$X_n = \mathrm{blockdiag}\left\{ \sum_{l\in\mathcal{N}_1} c^n_{l,n} x_{l,n} x^T_{l,n}, \ldots, \sum_{l\in\mathcal{N}_k} c^n_{l,n_{et}} x_{l,n_{et}} x^T_{l,n_{et}}, \ldots, \sum_{l\in\mathcal{N}_N} c^n_{l,n} x_{l,n} x^T_{l,n} \right\}.$$

$\tilde{\omega}_n$ is the estimate-error stacking matrix of all nodes, defined as follows:

$$\tilde{\omega}_n = \begin{bmatrix} \tilde{w}_{1,n} \\ \vdots \\ \tilde{w}_{k,n_{et}} \\ \vdots \\ \tilde{w}_{N,n} \end{bmatrix}.$$

We can verify that

$$\tilde{\omega}_{n+1} = \mathcal{B}_n \tilde{\varsigma}_n, \tag{30}$$

while

$$\mathcal{B}_n = \begin{bmatrix} \mathrm{B}_{1,1,n}, \ldots, \mathrm{B}_{1,K,n} \\ \vdots\vdots \\ \mathrm{B}_{K,1,n}, \ldots, \mathrm{B}_{K,K,n} \end{bmatrix},$$

$$B_{i,j,n} = \begin{cases} I_L - \sum_{l\in\mathcal{N}_i\backslash\{i\}} a^n_{l,k}H_{l,n} \text{ if } j = i, \\ a^n_{j,i}H_{j,n} \quad \text{if} \quad j \in \mathcal{N}_i\backslash\{i\}, \\ O_L, \quad \text{otherwise}, \end{cases}$$

where O_L is the $L \times L$ zero matrix.

Bring Eq. (29) into Eq. (30), we get the estimate-error vector,

$$\tilde{\omega}_{n+1} = \mathcal{B}_n(\mathrm{I}_{\mathrm{LK}} - \mathrm{M}\mathrm{X}_n)\tilde{\omega}_n - \mathcal{B}_n\mathrm{M}\mathrm{g}_n. \tag{31}$$

B. Mean performance

$\tilde{\omega}_n$, X_n, and $\mathcal{B}_n$ are independent of each other, in view of A1 and A2. Taking the expectation of both sides of Eq. (31). We get

$$\mathrm{E}\left[\tilde{\omega}_{n+1}\right] = Q(\mathrm{I}_{\mathrm{LK}} - \mathrm{M}\Re)\mathrm{E}\left[\tilde{\omega}_n\right], \tag{32}$$

where

$$\begin{aligned} \Re &= \mathrm{E}\left[\mathrm{X}_n\right] \\ &= \mathrm{blockdiag}\{R_1, \ldots, R_N\}' \end{aligned}$$

and

$$Q = \mathrm{E}\left[\mathcal{B}_n\right].$$

According to Eq. (32), if the algorithm is required to be stable in the mean sense, the matrix $Q(\mathrm{I}_{\mathrm{LK}} - \mathrm{M}\Re)$ is required to be stable. All the rows of Q add up to unity. Therefore, when the matrix $(\mathrm{I}_{\mathrm{LK}} - \mathrm{M}\Re)$ is stable, Eq. (32) is stable as well, and we have

$$|\lambda_{\max}\{I_{LK} - M\Re\}| < 1, \tag{33}$$

and $\lambda_{\max}\{\cdot\}$ represents the largest eigenvalue of the matrix. The eigenvalue of the matrix $\mathrm{I}_{\mathrm{LK}} - \mathrm{M}\Re$ is the union of the eigenvalues of the matrix $\mathrm{I}_{\mathrm{L}} - \mu_k R_k$. Therefore, Eq. (33) is satisfied when

$$|1 - \mu_k\lambda_{\max}\{R_k\}| < 1 \quad \forall k.$$

By analysis, if the algorithm is required to be stable in the mean sense, the step size range is

$$0 < \mu_k < \frac{2}{\lambda_{\max}\{R_k\}} \quad \forall k. \tag{34}$$

C. Mean-square performance

The squared weighted Euclidean norm of a vector b with a weighting matrix A is

$$\|b\|^2_{\mathrm{A}} = \mathrm{b}^{\mathrm{T}}\mathrm{A}\mathrm{b}. \tag{35}$$

Using this Euclidean norm, we analyze the mean-squared stability of the ET-APDLMS algorithm. Taking the squared Euclidean norm on

both sides of Eq. (31) and applying the expectation operator while considering A1 and A2, we have

$$\mathrm{E}\left[\|\tilde{\omega}_{n+1}\|_{\Sigma}^{2}\right] = \mathrm{E}\left[\|\tilde{\omega}_{n}\|_{\Gamma}^{2}\right] + \mathrm{E}\left[\mathrm{g}_{\mathrm{n}}^{\mathrm{T}}\mathrm{M}\mathcal{B}_{\mathrm{n}}^{\mathrm{T}}\Sigma\mathcal{B}_{\mathrm{n}}\mathrm{M}\mathrm{g}_{\mathrm{n}}\right], \tag{36}$$

$$\Gamma = (\mathrm{I}_{\mathrm{LK}} - \mathrm{MX}_{\mathrm{n}})\mathcal{B}_{\mathrm{n}}^{\mathrm{T}}\Sigma\mathcal{B}_{\mathrm{n}}(\mathrm{I}_{\mathrm{LK}} - \mathrm{MX}_{\mathrm{n}}), \tag{37}$$

where Σ is a random symmetric nonnegative-definite matrix.

Under A1 and A2, $\tilde{\omega}_n$ and Γ are independent of each other. Thus, we get

$$\mathrm{E}\left[\|\tilde{\omega}_{n}\|_{\Gamma}^{2}\right] = \mathrm{E}\left[\|\tilde{\omega}_{n}\|_{\mathrm{E}[\Gamma]}^{2}\right]. \tag{38}$$

By defining

$$\gamma = \mathrm{vec}\{E[\Gamma]\}$$

and

$$\delta = \mathrm{vec}\{\Sigma\},$$

according to Eq. (38), we modify Eq. (36) to

$$E\left[\|\tilde{\omega}_{n+1}\|_{\delta}^{2}\right] = E\left[\|\tilde{\omega}_{n}\|_{\gamma}^{2}\right] + E\left[g_n^T M B_n^T \Sigma B_n M g_n\right]. \tag{39}$$

$vec\{.\}$ is a vectorization operator, which can stack the columns of a matrix into a single column vector and $vec^T\{.\}$ represents the transpose of this operation. According to the relationship between the matrix vectorization operation and the Kronecker product, we express vec{.} as

$$\mathrm{vec}\{ABC\} = (C^T \otimes A)\mathrm{vec}\{B\}. \tag{40}$$

Meanwhile, we rewrite γ as

$$\gamma = \Lambda\delta, \tag{41}$$

where

$$\Lambda = E[((I_{LK} - MX_n) \otimes (I_{LK} - MX_n))]D \tag{42}$$

and

$$D = E[B_n^T \otimes B_n^T]. \tag{43}$$

The derivation process of D is given in the literature.[26] Considering A3, we approximate Eq. (40) as

$$\Lambda \approx (\mathrm{I}_{\mathrm{L^2K^2}} - \mathrm{I}_{\mathrm{LK}} \otimes \mathrm{M}\mathfrak{R} - \mathrm{M}\mathfrak{R} \otimes \mathrm{I}_{\mathrm{LK}})\mathrm{D}. \tag{44}$$

According to the relationship between the vector operation vec{.} and the matrix trace proposed in the literature,[33] we have

$$\mathrm{tr}\{A^T B\} = \mathrm{vec}^T\{B\}\mathrm{vec}\{A\} \tag{45}$$

and

$$E[g_{fn}^T M B_n^T \Sigma B_n M g_{fn}] = \mathrm{vec}^T\{H\}D\delta, \tag{46}$$

$$H = \mathrm{blockdiag}\{\mu_1^2\zeta_1^2 R_1, \ldots, \mu_N^2\zeta_N^2 R_N\}. \tag{47}$$

Bringing Eqs. (41) and (46) to Eq. (39) yields

$$E\left[\|\tilde{\omega}_{n+1}\|_{\delta}^{2}\right] = E\left[\|\tilde{\omega}_{n}\|_{\Lambda\delta}^{2}\right] + vec^T\{H\}D\delta. \tag{48}$$

When Λ is stable, Eq. (48) is stable in the mean-square sense, and Λ can be approximated as

$$\Lambda \approx [(\mathrm{I}_{\mathrm{LK}} - \mathrm{M}\mathfrak{R}) \otimes (\mathrm{I}_{\mathrm{LK}} - \mathrm{M}\mathfrak{R})]\mathrm{D}. \tag{49}$$

Therefore, the stability of Eq. (49) has the same condition as the stability of $(\mathrm{I}_{\mathrm{LK}} - \mathrm{M}\mathfrak{R})$. Therefore, choosing the step size μ_k satisfying Eq. (34) makes the algorithm stable in the mean-square sense.

D. Communication cost analysis

In this section, we analyze the communication cost of different algorithms from the adaption step and the combination step.

(a) Communication cost analysis of the DLMS algorithm
Adaption: In this step, each node receives $\{x_{l,n}, d_{l,n}\}$ from the neighbor nodes. $x_{l,n}$ is an L-dimensional vector and $d_{l,n}$ is a scalar. Suppose that each node in the network has an average of $F = \frac{1}{N}\sum_{k=1}^{N}[\mathcal{N}_k]$ neighbor nodes. Therefore, the amount of data transferred in this step is $NF(L+1)$.
Combination: In this step, each neighbor node shares its intermediate estimation $\psi_{l,n}$, which is an L-dimensional vector. Therefore, the data quantity transmitted in this step is NFL.

(b) Communication cost analysis of the PDLMS algorithm
Adaption: The communication cost of the PDLMS algorithm in this step is the same as that of DLMS. Therefore, the amount of data transferred in the adaption step is $NF(L+1)$.
Combination: In this step, a subset of intermediate estimations of neighbor nodes of the node k is used for the parameter estimation. It is assumed that the data used are an M-dimensional vector, and the remaining $L-M$ dimensional data are replaced with the intermediate estimation of node k; therefore, the data quantity transmitted in the combination step is NFM.

(c) Communication cost analysis of the ET-APDLMS algorithm
Adaption: The communication cost of the ET-APDLMS algorithm in this step is the same as that of DLMS. Therefore, the amount of data transferred in the adaption step is $NF(L+1)$.
Combination: In this step, a subset of intermediate estimations of neighbor nodes of the node k are used for the parameter estimation. It is assumed that the data used are an M-dimensional vector (The choice of M is related to $\mathbb{H}$), and the remaining $L-M$ dimensional data are replaced with the intermediate estimation of the node k; therefore, the data quantity transmitted in the combination step is NFM.

Considering the event-triggered mechanism, we assume that the number of iterations that will transmit the intermediate estimate is $\tilde{I}$ $(\tilde{I} < I)$.

The sum of the communication costs of the I iterations of these algorithms is shown in Table III.

TABLE III. Communication cost analysis.

Algorithm	Communication costs
DLMS	$[NF(L+1)+NFL]I$
PDLMS	$[NF(L+1)+NFM]I$
ET-APDLMS	$NF(L+1)I+NFM\tilde{I}$

V. SIMULATIONS

To illustrate the performance of the algorithm in the network, we design a wireless sensor network with 50 nodes. The input $x_{k,n}$ and noise $v_{k,n}$ of each node are given in Fig. 6.

The ET-APDLMS algorithm is used in dynamic topological networks. In this paper, a dynamic network topology model is designed to simulate the actual use environment (such as the ocean and atmosphere). When nodes are connected, it means that nodes can communicate with each other. We randomly placed 50 nodes in the $200 * 200$ area. To observe the change of the dynamic topology, we present the network topology at four moments ($n = 1$, $n = 50$, $n = 100$, and $n = 150$). According to Fig. 7, we find that the network topology changes with time, and the most direct manifestation is that the number of neighbors of nodes changes.

ET-APDLMS is an adaptive algorithm to select a subset of the dimensions of the intermediate estimator according to the number of neighbor nodes. In this paper, simulation is designed to prove the effectiveness of changing the dimension of the intermediate estimate subset for improving the estimation accuracy. In Fig. 8, M represents the dimension of a subset of the intermediate estimate. The step size used in the simulation is $\mu_k = 0.05$, and each node has an average of $F = 4$ neighbor nodes. By analyzing the average of 50 independent simulation values, we find that when the number of neighbors of nodes is the same, the estimation performance of the algorithm improves with the increase of M.

Previous simulations have verified the influence of the dimension of the subset of the intermediate estimate (M) on the estimation performance when the number of neighbor nodes (F) is the same. In this simulation, we verify that if the subset dimension (M) of the intermediate estimator is the same, the number of neighbor nodes (F) will affect the estimation performance of the algorithm. Each node has an average of F neighbor nodes, the step size used in the simulation is $\mu_k = 0.05$, and the dimension of the subset of the intermediate estimate is $M = 4$. According to Fig. 9, by analyzing the average of 50 independent simulation values, we found that the estimation performance of the algorithm improved with the increase of the number of adjacent nodes.

The traditional partial algorithm mainly aims at the case of high communication costs but does not consider the effect of dynamic topology networks on the estimated performance. The PDLMS algorithm in the dynamic and non-dynamic network topology is analyzed by 50 independent simulation values. According to Fig. 10,

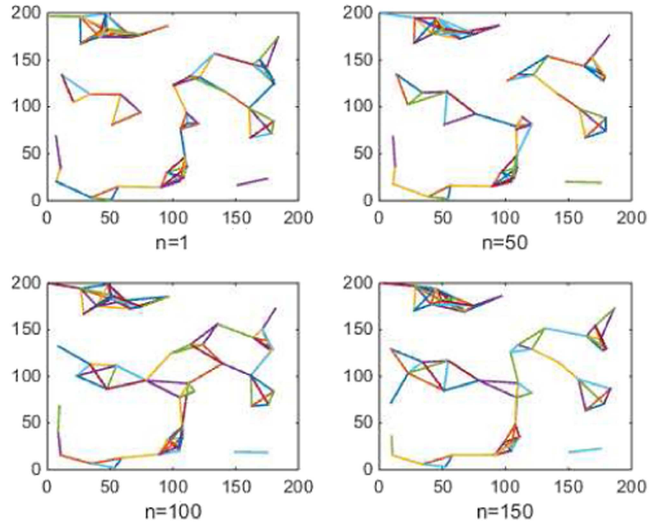

FIG. 7. Dynamic network topologies. In the figure, four network topologies of $n = 1$, $n = 50$, $n = 100$, and $n = 150$ are given, respectively.

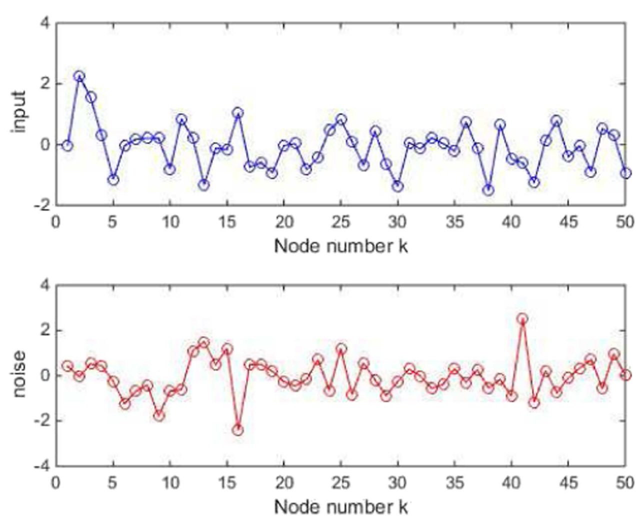

FIG. 6. Input signal and noise.

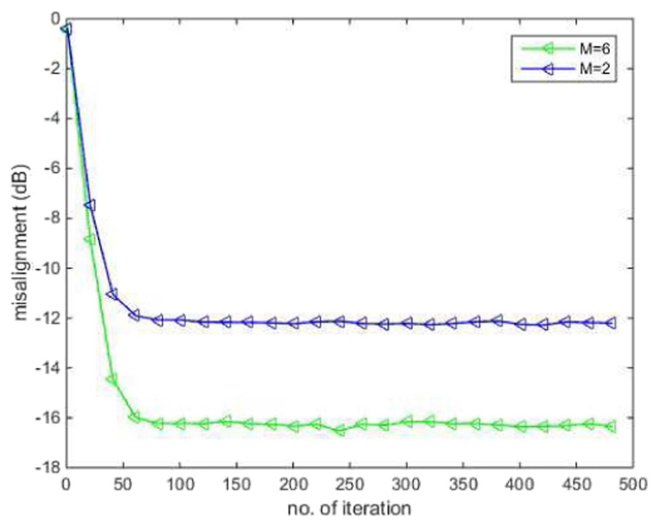

FIG. 8. The effect of the dimension M of the subset of the intermediate estimate on the estimation performance. The mean of 50 independent simulations with $\mu_k = 0.05$ and $F = 4$ is shown in the figure.

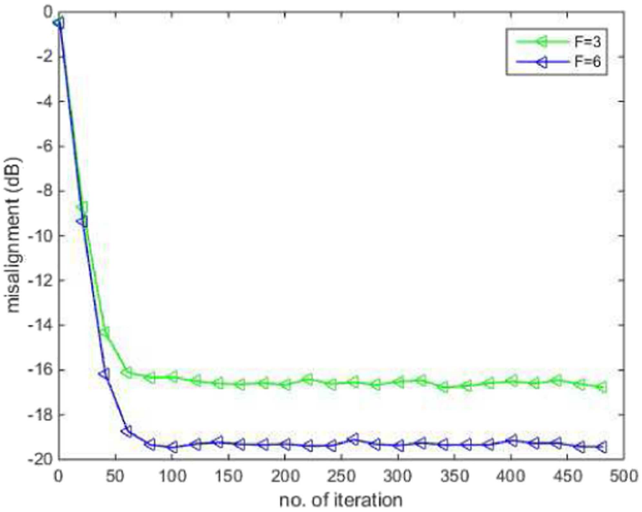

FIG. 9. The impact of the number of neighbors on the node on the estimate performance. The mean of 50 independent simulations with $\mu_k = 0.05$ and $M = 4$ are shown in the figure.

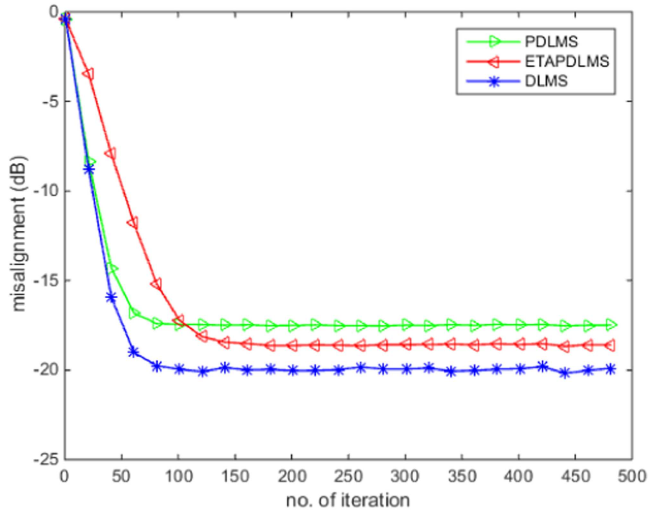

FIG. 11. Estimation performance of the DLMS, PDLMS, and ET-APDLMS. The simulation results were derived from the mean value of 50 independent simulations, in which $\mu_k = 0.05$.

it is found that the estimation performance of the PDLMS algorithm is low in a dynamic network environment

Figure 11 compares the estimate performance of the ET-APDLMS algorithm, the DLMS algorithm, and the PDLMS algorithm. The step size used in the simulation is $\mu_k = 0.05$, and the threshold $th = 25$. The simulation results show that the ET-APDLMS algorithm is more adaptive to the dynamic network topology than the PDLMS algorithm. Compared with the DLMS algorithm, the estimation performance of the ET-APDLMS algorithm is slightly lower, but the previous analysis shows that the communication cost of the ET-APDLMS algorithm is significantly reduced.

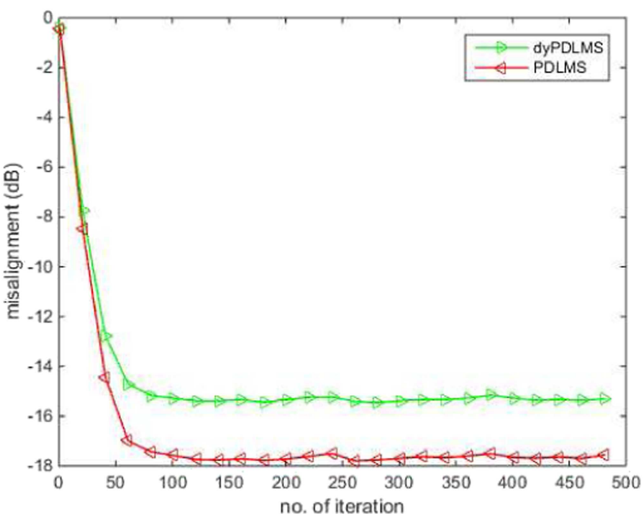

FIG. 10. PDLMS algorithm in the dynamic network topology. The simulation results were derived from the mean value of 50 independent simulations, in which $\mu_k = 0.05$.

VI. CONCLUSION

In this paper, we find that a dynamic network topology affects the estimation performance of partial diffusion strategies. Therefore, an adaptive partial-diffusion strategy is designed in this paper. Meanwhile, in order to reduce the communication resource waste caused by data redundancies in the adaptive partial diffusion strategy, we designed the event-triggered mechanism and proposed the ET-APDLMS algorithm. Through simulation, we find that the ET-APDLMS algorithm is more adaptive to a dynamic network topology than the PDLMS algorithm. Compared with the DLMS algorithm, the ET-APDLMS algorithm uses the subset of intermediate estimation to complete parameter estimation; therefore, the estimation performance is slightly lower, but the communication cost can be effectively reduced. The simulation results are consistent with the performance analysis, which proves the effectiveness of the proposed algorithm.

ACKNOWLEDGMENTS

The authors would like to acknowledge the Fundamental Research Funds for the Central Universities (Nos. XDJK2020B034 and SWU019029) for their support in this research.

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

[1] B. Chen, J. Wang, H. Zhao, N. Zheng, and J. C. Principe, "Convergence of a fixed-point algorithm under maximum correntropy criterion," IEEE Signal Process. Lett. **22**(10), 1723–1727 (2015).
[2] B. Chen, L. Xing, H. Zhao, N. Zheng, and J. C. Principe, "Generalized correntropy for robust adaptive filtering," IEEE Trans. Signal Process. **64**(13), 3376–3387 (2016).
[3] F. S. Cattivelli and A. H. Sayed, "Diffusion LMS strategies for distributed estimation," IEEE Trans. Signal Process. **58**(3), 1035–1048 (2010).
[4] S. Wang and C. Li, "Distributed robust optimization in networked system," IEEE Trans. Syst. Man Cybern. **47**(8), 2321–2333 (2017).
[5] J. Hua, C. Li, and H. Shen, "Distributed learning of predictive structures from multiple tasks over networks," IEEE Trans. Ind. Electron. **64**(5), 4246–4256 (2017).
[6] S. Huang and C. Li, "Distributed sparse total least-squares over networks," IEEE Trans. Signal Process. **63**(11), 2986–2998 (2015).
[7] C. G. Lopes and A. H. Sayed, "Incremental adaptive strategies over distributed networks," IEEE Trans. Signal Process. **55**(8), 4064–4077 (2007).
[8] M. G. Rabbat and R. D. Nowak, "Quantized incremental algorithms for distributed optimization," IEEE J. Sel. Areas Commun. **23**(4), 798–808 (2005).
[9] A. Nedic and D. P. Bertsekas, "Incremental subgradient methods for nondifferentiable optimization," SIAM J. Optim. **12**(1), 109–138 (2001).
[10] D. P. Bertsekas, "A new class of incremental gradient methods for least squares problems," SIAM J. Optim. **7**(4), 913–926 (1997).
[11] N. Bogdanovic, J. Platachaves, and K. Berberidis, "Distributed incremental-based LMS for node-specific adaptive parameter estimation," IEEE Trans. Signal Process. **62**(20), 5382–5397 (2014).
[12] A. Bertrand and M. Moonen, "Consensus-based distributed total least squares estimation in ad hoc wireless sensor networks," IEEE Trans. Signal Process. **59**(5), 2320–2330 (2011).
[13] A. Bertrand and M. Moonen, "Low-complexity distributed total least squares estimation in ad hoc sensor networks," IEEE Trans. Signal Process. **60**(8), 4321–4333 (2012).
[14] S. S. Stankovic, M. S. Stankovic, and D. M. Stipanovic, "Decentralized parameter estimation by consensus based stochastic approximation," IEEE Trans. Automat. Contr. **56**(3), 531–543 (2011).
[15] G. Mateos, I. D. Schizas, and G. B. Giannakis, "Distributed recursive least-squares for consensus-based in-network adaptive estimation," IEEE Trans. Signal Process. **57**(11), 4583–4588 (2009).
[16] S. Kar and J. M. Moura, "Distributed consensus algorithms in sensor networks with imperfect communication: Link failures and channel noise," IEEE Trans. Signal Process. **57**(1), 355–369 (2009).
[17] I. D. Schizas, A. Ribeiro, and G. B. Giannakis, "Consensus in ad hoc WSNs with noisy links—Part I: Distributed estimation of deterministic signals," IEEE Trans. Signal Process. **56**(1), 350–364 (2008).
[18] F. Chen and X. Shao, "Broken-motifs diffusion LMS algorithm for reducing communication load," Signal Process. **133**, 213–218 (2017).
[19] R. Abdolee, B. Champagne, and A. H. Sayed, "A diffusion LMS strategy for parameter estimation in noisy regressor applications," in *2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO)* (IEEE, 2012), pp. 749–753.
[20] X. Zhao, S.-Y. Tu, and A. H. Sayed, "Diffusion adaptation over networks under imperfect information exchange and non-stationary data," IEEE Trans. Signal Process. **60**(7), 3460–3475 (2012).
[21] A. Bertrand, M. Moonen, and A. H. Sayed, "Diffusion bias-compensated RLS estimation over adaptive networks," IEEE Trans. Signal Process. **59**(11), 5212–5224 (2011).
[22] S. Chouvardas, K. Slavakis, and S. Theodoridis, "Adaptive robust distributed learning in diffusion sensor networks," IEEE Trans. Signal Process. **59**(10), 4692–4707 (2011).
[23] N. Takahashi, I. Yamada, and A. H. Sayed, "Diffusion least-mean squares with adaptive combiners: Formulation and performance analysis," IEEE Trans. Signal Process. **58**(9), 4795–4810 (2010).
[24] F. Chen, T. Shi, S. Duan, L. Wang, and J. Wu, "Diffusion least logarithmic absolute difference algorithm for distributed estimation," Signal Process. **142**, 423–430 (2018).
[25] C. Li, P. Shen, Y. Liu, and Z. Zhang, "Diffusion information theoretic learning for distributed estimation over network," IEEE Trans. Signal Process. **61**(16), 4011–4024 (2013).
[26] R. Arablouei, S. Werner, Y. F. Huang *et al.*, "Distributed least mean-square estimation with partial diffusion," IEEE Trans. Signal Process. **62**(2), 472–484 (2014).
[27] J. Chen and A. H. Sayed, "Diffusion adaptation strategies for distributed optimization and learning over networks," IEEE Trans. Signal Process. **60**(8), 4289–4305 (2012).
[28] 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. **81**, 16–25 (2018).
[29] Y. Liu, C. Li, and Z. Zhang, "Diffusion sparse least-mean squares over networks," IEEE Trans. Signal Process. **60**(8), 4480–4485 (2012).
[30] R. Arablouei, K. Dogancay, S. Werner, and Y. Huang, "Adaptive distributed estimation based on recursive least-squares and partial diffusion," IEEE Trans. Signal Process. **62**(14), 3510–3522 (2014).
[31] V. Vadidpour, A. Rastegarnia, A. Khalili, and S. Sanei, "Partial-diffusion least mean-square estimation over networks under noisy information exchange," arXiv:1511.09044 (2015).
[32] P. S. Diniz, "On data-selective adaptive filtering," IEEE Trans. Signal Process. **66**(16), 4239–4252 (2018).
[33] K. M. Abadir and J. R. Magnus, *Matrix Algebra* (Cambridge University Press, New York, 2005).
[34] Y. He, F. Wang, S. Wang, P. Ren, and B. Chen, "Maximum total correntropy diffusion adaptation over networks with noisy links," IEEE Trans. Circuits Syst. II Express Briefs **66**(2), 307–311 (2019).
[35] W. Ma, B. Chen, J. Duan, and H. Zhao, "Diffusion maximum correntropy criterion algorithms for robust distributed estimation," Digit. Signal Process. **58**, 10–19 (2016).