



Diffusion Fractional Tap-length Algorithm with Adaptive Error Width and Step-size

Ghanbar Azarnia¹ 

Received: 17 November 2020 / Revised: 19 June 2021 / Accepted: 23 June 2021 /
Published online: 20 July 2021

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

Abstract

Estimation in a cooperative and distributed manner in wireless sensor networks (WSNs) has considered much attention in recent years. When this distributed estimation is performed adaptively, the concept of adaptive networks will develop. In such networks, proper selection of the unknown parameter length is an issue in itself. A deficient filter length results in an additional steady-state error while selecting a large length will impose a more computational load on the nodes, which is critical in sensor networks due to the lack of energy resources. This motivates the use of variable tap-length adaptive filters in the context of the adaptive networks. This has been achieved in adaptive networks using the distributed fractional tap-length (FT) algorithm. This algorithm requires proper selection of the length adaptation parameters, such as error width and length adaptation step-size. This paper proposes an automatic method for selecting these parameters. In the proposed method, these parameters are adapted based on the estimated gradient vector. The proposed method is fully distributed and presented in a diffusion strategy. Simulation results show that the proposed algorithm has both the advantage of fast length convergence and an unbiased steady-state tap-length.

Keywords Wireless sensor networks · Variable tap-length · Variable error width · Variable step-size · Diffusion strategy

1 Introduction

In most applications of wireless sensor networks (WSNs), we are encountered with estimating the desired parameter. In some scenarios, to perform the processing required for this estimate, sensors send their information to a central processing unit (e.g., cluster head or fusion center). Then, the processing of these gathered data, which inherently

✉ Ghanbar Azarnia
g.azarnia@uut.ac.ir

¹ Engineering Faculty of Khoy, Urmia University of Technology, Urmia, Iran

have temporal and spatial correlations, are performed in a centralized manner in this central unit [2,3]. This amount of communication between the sensors and the central processor consumes a significant amount of bandwidth and energy, which is critical for a network of limited power sensors. Therefore, the need for a distributed solution to prolong the network lifetime is undeniable. In a distributed method, the amount of data exchanges is significantly reduced, so that each sensor communicates only with a subset of its one-hop neighbors. It is dictated by the network topology, that each sensor is permitted to communicate with which of its neighbors. Two useful topologies that enable learning and adaptation over networks in real-time are the diffusion and incremental topologies [13]. In an incremental structure, each node communicates with only one adjacent node. So, in this structure, data flows sequentially from one sensor to a neighboring sensor [32]. This form of cooperation requires a cyclic design of cooperation between the sensors, which is an NP-hard problem. Also, cyclic paths are not robust to link or node failures. Therefore, a diffusion strategy can be preferred in which each sensor has access to data of all its neighbors. In this scheme, each sensor can communicate with all its neighbors as dictated by the network structure. So, in this strategy, no cyclic path is required; this scheme is scalable, robust to link or node failures, and is more flexible to distributed implementation. This form of cooperation usually involves two stages: an adaption stage where sensors use their measurements to update their estimation. A combination stage where sensors combine the estimates from their neighbors to make a new estimate [7,8,16,24,33]. Diffusion recursive least square (RLS) algorithm [14,30], diffusion least mean square (LMS) algorithm[34], diffusion affine projection-based adaptive (APA) algorithm [1], diffusion distributed conjugate gradient-based algorithms [36], jointly sparse single-task[11] and multi-task [12,20] recovery algorithms are examples of widely used algorithms employing diffusion strategy.

With the advent of adaptive networks based on incremental and diffusion strategies, extensive research has been conducted in this field. For example, paper [17] investigates the implementation of Active Noise Control (ANC) systems over an incremental adaptive network, using the distributed version of the Multiple Error Filtered-x Least Mean Square (MEFxLMS) algorithm. In [5], the constrained stability approach is applied for distributed ANC systems employing an incremental communication topology.

In the mentioned algorithms, the length of the unknown vector is assumed to be fixed and known a priori at each node. However, this assumption is not suitable for many applications where the optimum filter length is unknown or variable. The tap-length, or the size of the unknown parameter, significantly affects the performance of the distributed networks. If the number of filter coefficients is kept fix at a smaller value due to motives such as reduced computational load, which is a necessity for WSNs, the mean-square error (MSE) will increase as a result of this length deficiency [9,10], whereas a larger tap-length increases the computational cost and the excess MSE (EMSE). Therefore, there exists an optimal length that best balances the conflicting necessities of performance and complexity. Based on this, structure adaptation algorithms were proposed, with the idea that although the minimum MSE is a monotonic non-increasing function of the filter length, it decreases slightly as the filter length increases when the tap-length is sufficiently large. On the other hand, having a large

tap-length is not suitable as it not only undesirably increases computational complexity but also provides more adaption noise. In the stand-alone adaptive filter context, several algorithms have been proposed [31,38] to find such an optimal tap-length. These algorithms have finally advanced into the FT variable-length algorithm [37]. The FT algorithm has the required conditions to be considered as a popular variable-length algorithm. This algorithm is very similar to the LMS technique, so it is often called an LMS style variable tap-length algorithm. This algorithm is as simple as the LMS algorithm, and at the same time, it has a good performance [19,29]. Therefore, it is not unexpected that in the context of the adaptive networks, among all variable tap-length algorithms, only this algorithm is considered for structure adaptation. This algorithm is presented in the distributed context in both incremental [22] and diffusion strategies [21].

Note that the concept of variable tap-length algorithms is different from the variable step-size algorithms proposed to satisfy the conflicting requirements of low misadjustment and fast convergence rate [6,28].

FT algorithm contains several parameters that influence the performance of the algorithm. Therefore, the proper selection of these parameters is very critical in achieving acceptable performance. One of these parameters is the error width, which controls the trade-off between the convergence rate and the steady-state tap-length bias. Another critical parameter is the tap-length adaptation step-size. This parameter provides a trade-off between steady-state tap-length fluctuation and the convergence rate of the tap-length. Therefore, selecting a fixed value for these parameters cannot provide the superior performance of this algorithm comprehensively. Therefore, to eliminate these trade-offs, a variable approach is proposed for the selection of these parameters. In the proposed method, these parameters are adjusted to achieve a fast convergence in the initial stages of the algorithm, and a less tap-length fluctuation and accurate tap-length estimation in the steady-state.

There are several works for error-width adaptation [18,23,27], and tap-length step-size adaptation [4,25,35].

In [23], the error width is adapted based on the mean squared error estimation. But, this algorithm affects by the interference of system error and could not indicate the accurate state of iteration at the same time. The reported method in [27] could eliminate the system noise interference and enhance the convergence performance. But, in this method, the error width is affected by the previous instant error value. So this approach may not accurately indicate the current changes and may have a specific passive influence on tap-length updating in the initial iterations. On this basis, [18] proposed an approach that adjusts error width using fragment-full error (FE) to solve the challenges associated with [23] and [27].

In [25], the tap-length adjustment error (TAE) is served as a criterion to adapt the tap-length step-size. To be robust against independent noise disturbance, the authors in [25] have used the estimation of the TAE correlation between two consecutive iterations to adjust the tap-length step-size. By arguing that in a time-varying scenario, the changes in the variance of input signal or the tap-coefficients will require the retuning of tap-length step-size, paper [4] exploited a normalized tap-length step-size in length update procedure. In [35], a variable tap-length step-size method is proposed, where the tap-

length step-size is adjusted by the difference between squared output error and squared segmented estimation error, and is limited by a sigmoid function.

The presented methods in [4,18,23,25,27,35] are all in the context of stand-alone adaptive filters, but, in the context of diffusion adaptive networks, this matter has been left out of consideration. It motivates us to consider the variable parameter concept in the diffusion fractional tap-length scenario.

Note that, in the proposed method, the adaptation of the parameters is performed in a cooperative and distributed manner. This distributed adaptation will result in the superior performance of the proposed algorithm.

The roadmap of the remainder of this paper is as follows. The preliminaries on the FT algorithm is provided in Sect. 2. In Sect. 3, we present a variable tap-length algorithm with adaptive error width and adaptive tap-length adaptation step-size based on the gradient vector norm. Section 4 provides the steady-state analysis. In Sect. 5, computer simulations are conducted, and finally, in Section 6, we conclude this paper.

2 Preliminaries

Suppose a network consists of N nodes. Here, the purpose of this network is to obtain an estimate of the desired parameter $w_{L_{opt}}^o$ and its length L_{opt} . It is to be highlighted that the estimation of both the tap-weights and tap-length of the unknown parameter $w_{L_{opt}}^o$ is considered here. However, the adaption criterion for weights and length could be separated, and the selection criterion for one does not depend on the selection criterion of the other. First, we assume that the tap-length is L , which is searched by the length search solution discussed later. We also assume that at every time $i > 0$, each node k has access to a scalar measurement $d_k(i)$, and a $1 \times L$ regression vector $\mathbf{u}_{k,i}$, where they are assumed to be time realizations of zero-mean random processes $\{\mathbf{d}_k, \mathbf{u}_k\}$. These measurements are related to the unknown parameter by

$$\mathbf{d}_k(i) = \mathbf{u}_{L_{opt}k,i} w_{L_{opt}}^o + \mathbf{v}_k(i) \quad (1)$$

where $\mathbf{v}_k(i)$ accounts for the zero-mean noise and modeling error with variance $\sigma_{v,k}^2$ which is assumed independent over space and time and independent of the regressors. In (1) the subscript L_{opt} implies that the length of the vector $\mathbf{u}_{L_{opt}k,i}$ is L_{opt} . Collecting the signals into global matrices result in:

$$\begin{aligned} \mathbf{U} &\triangleq \text{col}\{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_N\} \quad (N \times L) \\ \mathbf{d} &\triangleq \text{col}\{\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_N\} \quad (N \times 1) \end{aligned} \quad (2)$$

where $\text{col}\{\cdot \cdot \cdot\}$ stands for a vector resulted by stacking the specified vectors. By considering the linear least-mean-squares estimation problem:

$$\arg \min_w J_L(\mathbf{w}) \quad \text{with} \quad J_L(\mathbf{w}) = E\{\|\mathbf{d} - \mathbf{U}\mathbf{w}\|^2\} \quad (3)$$

reference [26] proposed a diffusion distributed LMS strategy as:

$$\begin{cases} \boldsymbol{\phi}_k^{(i-1)} = \sum_{\ell \in \mathfrak{N}_k} c_{k,\ell} \boldsymbol{\psi}_\ell^{(i-1)} \\ \boldsymbol{\psi}_\ell^{(i)} = \boldsymbol{\phi}_k^{(i-1)} + \mu_k \mathbf{u}_{k,i}^* \left(d_k(i) - \mathbf{u}_{k,i} \boldsymbol{\phi}_k^{(i-1)} \right) \end{cases} \tag{4}$$

where \mathfrak{N}_k accounts for the neighborhood of node k , which is defined as the collection of all sensors topologically connecting to node k , including itself. According to (4), the diffusion strategy is implemented in two steps: In the first step, each node linearly combines the local estimates gathered from its neighbors. This step fuses data from sensors across the network into node k . In the second step, the resulting aggregate at node k is fed into the local adaptive filter to adapt the local data measured at sensor k .

In (4) the local combiners $\{c_{k,\ell}\}$ satisfy $\sum_{\ell \in \mathfrak{N}_k} c_{k,\ell} = 1$. The combiners $\{c_{k,\ell}\}$ give rise to a matrix $C = [c_{k,\ell}]$ that contains the network topology information. There are several rules for choosing these combiners. In this paper, we utilize the Metropolis rule as [26]:

$$c_{k,\ell} = \begin{cases} 1/\max(n_k, n_\ell) & \text{if } k \neq \ell \text{ are linked} \\ 0 & \text{if } k \text{ and } \ell \text{ are not linked} \\ 1 - \sum_{m \in \mathfrak{N}_k \setminus \{k\}} c_{k,m} & \text{if } k = \ell \end{cases} \tag{5}$$

where n_k and n_ℓ , respectively, denote the degree of node k and ℓ , i.e., the cardinality of \mathfrak{N}_k and \mathfrak{N}_ℓ .

Now we consider the tap-length estimation problem. It is common to use the segmented cost function to find the optimal tap-length. Because the tap-length is unknown, we temporarily assume that to be equal to L . With this assumption, the segmented cost function is defined as:

$$J_M^{(L)}(\mathbf{w}) \triangleq E\{\|\mathbf{d} - \mathbf{U}_M \mathbf{w}_M\|^2\} \tag{6}$$

where $1 \leq M \leq L$, \mathbf{w}_M and \mathbf{U}_M are vectors consisting of the first M elements/columns of \mathbf{w} and \mathbf{U} , respectively. The underlying principle of the FT algorithm is to determine the length L such that:

$$\min\{L | J_{(L-\Delta)}^{(L)}(\mathbf{w}) - J_L^{(L)}(\mathbf{w}) \leq \varepsilon\} \tag{7}$$

where ε is a small positive specified by the system requirements, and Δ is a positive integer to avoid the local minima tap-length.

Reference [21] presents a distributed solution based on diffusion strategy to solve (9):

$$\begin{aligned} l_{k,f}(i-1) &= \sum_{\ell \in \mathfrak{N}_k} b_{\ell,k} L_{\ell,f}(i-1) \\ L_{k,f}(i) &= l_{k,f}(i-1) - \alpha_k + \gamma_k \left(e_{L_k(i)-\Delta_k}^2(\boldsymbol{\psi}_{k,i-1}) - e_{L_k(i)}^2(\boldsymbol{\psi}_{k,i-1}) \right) \end{aligned} \tag{8}$$

where

$$\begin{aligned} e_{L_k(i)}(\boldsymbol{\psi}_{k,i-1}) &= d_k(i) - \mathbf{u}_{k,i} \boldsymbol{\psi}_{k,i-1} \\ e_{L_k(i)-\Delta_k}(\boldsymbol{\psi}_{k,i-1}) &= d_k(i) - \mathbf{u}_{k,i}(1:L_k(i)-\Delta_k) \boldsymbol{\psi}_{k,i-1}(1:L_k(i)-\Delta_k) \end{aligned} \quad (9)$$

where $\mathbf{u}_{k,i}(1:L_k(i)-\Delta_k)$ and $\boldsymbol{\psi}_{k,i-1}(1:L_k(i)-\Delta_k)$ are vectors consisting of the first $L_k(i)-\Delta_k$ elements of $\mathbf{u}_{k,i}$ and $\boldsymbol{\psi}_{k,i-1}$, respectively. In (8), the parameters α_k and γ_k are small positive values. The parameter α_k is the leakage factor of node k and is used to prevent an increase in length to a large undesirable value. The parameter γ_k is the length adaptation step-size of node k . In (8) the local combiners $\{b_{k,\ell}\}$ satisfy $\sum_{\ell \in \mathbb{S}_k} b_{k,\ell} = 1$. There are several rules for choosing these combiners. In this paper, we utilize the relative-degree rule as [15]:

$$b_{k,\ell} = \begin{cases} n_\ell / \sum_{m \in \mathbb{S}_k} n_m & \text{if } \ell \in \mathbb{S}_k \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

In (8), $L_{\ell,f}(i)$ and $l_{k,f}(i)$ are the local and integrated fractional tap-length estimates, and the integer tap-length is adjusted by

$$L_k(i) = \begin{cases} \lfloor L_{k,f}(i) \rfloor, & \text{if } |L_k(i-1) - L_{k,f}(i)| > \delta_k \\ L_k(i-1) & \text{otherwise} \end{cases} \quad (11)$$

where $\lfloor \cdot \rfloor$ rounds the embraced value to the nearest integer. Here, δ_k is the small local integer and is usually set to one.

3 Proposed Method

The given LMS style algorithm in (8) is a powerful approach to estimate the fractional tap-length in a diffusion strategy. However, the fact that its performance is affected by several parameters challenges the effectiveness of this algorithm. The value of the leakage factor α_k in (8) must be small enough so that the steady-state tap-length is not reduced too much, and large enough to prevent the “wandering” issue during the learning stage. Generally, the choice of this parameter was not critical, and values between 0.001 and 0.01 are good choices for the leakage factor in each node. Therefore, in this paper, we do not want to impose an additional computational burden on the nodes by updating the leakage factor, because the power limitation in wireless sensor networks is a critical issue. Instead, we will focus on the two key parameters, length adaptation step-size γ_k , and error width Δ_k . The error width controls the trade-off between the convergence rate and the steady-state tap-length bias. The tap-length adaptation step-size controls the trade-off between the tap-length convergence rate and steady-state tap-length fluctuation. A large Δ_k accelerates the convergence rate of the tap-length learning and equips the algorithm with the ability to escape from the possible suboptimal tap-lengths. Besides, a smaller Δ_k than the estimation of the optimal tap-length will result in a steady-state tap-length that is not significantly biased.

About the role of γ_k in the algorithm, a large γ_k accelerates the convergence rate of the tap-length learning but results in high steady-state fluctuation of the tap-length. Besides, a smaller γ_k , although, leads to a low steady-state tap length fluctuation but results in a slow convergence rate. So, this motivates us to propose a diffusion variable tap-length method with variable parameters rather than fixed, where the tap-length adaptation step-size and the error width are set in the large values during the initial iterations, and they reduce step by step to approach the smaller values in the steady-state. For this purpose, we need a measure to control these variations. Various criteria can be used for this purpose, among which the gradient vector is a suitable measure. It is well-known that the squared norm of the smoothed gradient vector tracks the variation of the mean square deviation. Hence, it is an excellent candidate to control the tap-length adaptation step-size and error width. In this paper, we propose a diffusion tap-length update with variable parameters method based on the gradient vector in which the tap length adaptation step-size and error width are adjusted to be proportional to the estimated smoothed gradient vector norm square. Since the Euclidean norm of the gradient vector is large in initial iterations and approaches to a small amount at steady-state, so by proportionate the parameters with it, all the goals we pursue will be met. In the sense that, since the proposed measure provides large Δ_k and γ_k at initial iterations, the fast convergence rate of the tap-length learning is achieved, and also this large Δ_k for each node prevents the tap-length from trapping in the sub-optimal value. On the other hand, the proposed measure provides a small Δ_k and γ_k at steady-state, which leads to the unbiased estimated tap-length and low steady-state fluctuation of the tap-length. Another point to consider is the strategy used to update these parameters. These parameters can be updated independently in each node without the cooperation of other nodes. However, we can take advantage of the spatial diversity to update these parameters. In the proposed method, we use the diffusion strategy for the adaptation of these parameters. So, these parameters are updated in a distributed manner with the cooperation of all nodes.

Accordingly, in the proposed method, first, each node linearly combines the local estimated smoothed instantaneous gradient vector gathered from its neighbors as

$$\boldsymbol{\rho}_k^{(a)}(i) = \sum_{\ell \in \mathcal{N}_k} c_{k,\ell} \boldsymbol{\rho}_\ell(i-1) \quad (12)$$

where the vector $\boldsymbol{\rho}_k(i)$ is the smoothed gradient vector, and $\boldsymbol{\rho}_k^{(a)}(i)$ is the aggregated gradient vector at node k . Thanks to this step, the estimation of the smoothed gradient vector enjoys spatial diversity. Then, the resulting aggregate at node k is updated as

$$\boldsymbol{\rho}_k(i) = \beta_k \boldsymbol{\rho}_k^{(a)}(i) + (1 - \beta_k) e_{L_k(i)} (\boldsymbol{\psi}_{k,i-1}) \mathbf{u}_{k,i} \quad (13)$$

where the smoothing parameter β_k is selected very close to the unit. Thus, (13) corresponds to a low pass filtering, which decreases the noise influence on the gradient vector estimation and leads to a more stable parameter adaptation. Note that the length of the vector $\boldsymbol{\rho}_k^{(a)}(i)$ is adjusted in each iteration according to the estimated tap-length.

We also implement the strategy of aggregating neighbors' local estimates for updating the parameters Δ_k and γ_k . On this basis, each node first calculates the aggregated estimations

$$\Delta_k^{(a)}(i) = \sum_{\ell \in \mathcal{N}_k} c_{k,\ell} \Delta_\ell(i-1) \quad (14)$$

and

$$\gamma_k^{(a)}(i) = \sum_{\ell \in \mathcal{N}_k} c_{k,\ell} \gamma_\ell(i-1) \quad (15)$$

where $\Delta_k(i)$ and $\gamma_k(i)$ are the estimated error width and tap-length adaptation step-size at node k and iteration i , respectively. The parameters $\Delta_k^{(a)}(i)$ and $\gamma_k^{(a)}(i)$ are the aggregated error width and tap-length adaptation step-size at node k and iteration i , respectively. Using the resulted aggregates, $\Delta_k(i)$ and $\gamma_k(i)$ are updated, respectively, as

$$\Delta_k(i) = \left[\theta_k \Delta_k^{(a)}(i) + \kappa_k \|\rho_k(i)\| \right] \quad (16)$$

and

$$\gamma_k(i) = \tau_k \gamma_k^{(a)}(i) + \eta_k \|\rho_k(i)\| \quad (17)$$

where $0 < \tau_k < 1$, $0 < \theta_k < 1$, η_k and κ_k are positive constants.

As is clear from (12)–(17), the proposed algorithm is fully distributed and is performed with the cooperation of all nodes. So that each node first takes the neighbors' local estimates, aggregates them as (12), (14), and (15) and then updates the aggregated results as (13), (16), and (17). Note that both $\Delta_k(i)$ and $\gamma_k(i)$ are updated with the same measure according to the above relations, and no separate measures are considered for each. The reason for this is to avoid additional calculations, and consequently, to avoid extra power consumption. This common measure, the Euclidean norm of the smoothed gradient vector, will provide large initial and small steady-state values for both $\Delta_k(i)$ and $\gamma_k(i)$. As a result, fast tap-length convergence, low steady-state fluctuations, and unbiased steady-state tap-length estimation will be provided.

The proposed diffusion variable tap-length method with variable parameters is summarized as follows:

$$\phi_k^{(i-1)} = \sum_{\ell \in \mathcal{N}_k} c_{k,\ell} \psi_\ell^{(i-1)} \quad (18a)$$

$$l_{k,f}(i-1) = \sum_{\ell \in \mathcal{N}_k} b_{\ell,k} L_{\ell,f}(i-1) \quad (18b)$$

$$\rho_k^{(a)}(i) = \sum_{\ell \in \mathcal{N}_k} c_{k,\ell} \rho_\ell(i-1) \quad (18c)$$

$$\Delta_k^{(a)}(i) = \sum_{\ell \in \mathcal{N}_k} c_{k,\ell} \Delta_\ell(i-1) \quad (18d)$$

$$\gamma_k^{(a)}(i) = \sum_{\ell \in \mathcal{N}_k} c_{k,\ell} \gamma_\ell(i-1) \quad (18e)$$

$$\psi_k^{(i)} = \phi_k^{(i-1)} + \mu_k \mathbf{u}_{k,i}^* \left(d_k(i) - \mathbf{u}_{k,i} \phi_k^{(i-1)} \right) \quad (18f)$$

$$e_{L_k(i)}(\boldsymbol{\psi}_{k,i}) = d_k(i) - \mathbf{u}_{k,i}\boldsymbol{\psi}_{k,i} \tag{18g}$$

$$\boldsymbol{\rho}_k(i) = \beta_k \boldsymbol{\rho}_k^{(a)}(i) + (1 - \beta_k)e_{L_k(i)}(\boldsymbol{\psi}_{k,i}) \mathbf{u}_{k,i} \tag{18h}$$

$$\Delta_k(i) = \left\lceil \theta_k \Delta_k^{(a)}(i) + \kappa_k \|\boldsymbol{\rho}_k(i)\| \right\rceil \tag{18i}$$

$$\gamma_k(i) = \tau_k \gamma_k^{(a)}(i) + \eta_k \|\boldsymbol{\rho}_k(i)\| \tag{18j}$$

$$e_{L_k(i)-\Delta_k}(\boldsymbol{\psi}_{k,i-1}) = d_k(i) - \mathbf{u}_{k,i}(1 : L_k(i) - \Delta_k)\boldsymbol{\psi}_{k,i}(1 : L_k(i) - \Delta_k) \tag{18k}$$

$$L_{k,f}(i) = l_{k,f}(i - 1) - \alpha_k + \gamma_k \left(e_{L_k(i)-\Delta_k}^2(\boldsymbol{\psi}_{k,i}) - e_{L_k(i)}^2(\boldsymbol{\psi}_{k,i}) \right) \tag{18l}$$

$$L_k(i) = \begin{cases} \lfloor L_{k,f}(i) \rfloor, & \text{if } |L_k(i - 1) - L_{k,f}(i)| > \delta_k \\ L_k(i - 1) & \text{otherwise} \end{cases} \tag{18m}$$

This proposed algorithm proceeds as follows. Each node k , at each iteration i , performs the following steps:

- 1) Receives the estimates $\boldsymbol{\psi}_\ell^{(i-1)}$, $L_{\ell,f}(i - 1)$, $\boldsymbol{\rho}_\ell(i - 1)$, $\Delta_\ell(i - 1)$, and $\gamma_\ell(i - 1)$ from its neighbors $\{\ell \in \mathfrak{N}_k\}$.
- 2) Computes the convex combination of estimates received from its neighbors according to (18a)-(18e).
- 3) Adapts its current aggregate weight estimate $\phi_k^{(i-1)}$ using its local data available at the time i , to obtain a new tap-weight estimate $\boldsymbol{\psi}_k^{(i)}$.
- 4) Using the estimated weight vector $\boldsymbol{\psi}_k^{(i)}$, this node updates the error estimation according to (18g), which is used to estimate the smoothed gradient vector according to (18h).
- 5) The Euclidean norm of the estimated gradient vector is used alongside with the aggregate estimates $\Delta_k^{(a)}(i)$ and $\gamma_k^{(a)}(i)$ in (18i) and (18j) to update the error width $\Delta_k(i)$ and the tap-length adaptation step-size $\gamma_k(i)$.
- 6) Using the updated $\Delta_k(i)$ and $\boldsymbol{\psi}_k^{(i)}$, this node updates the segmented error according to (18k).
- 7) Finally, the local tap-length is computed according to (18l) and (18m), and the length of the vectors are adjusted accordingly.

This iteration is concluded when node k exchanges $\boldsymbol{\psi}_k^{(i)}$, $L_{k,f}(i)$, $\boldsymbol{\rho}_k(i)$, $\Delta_k(i)$, and $\gamma_k(i)$ with its neighbors $\{\ell \in \mathfrak{N}_k\}$.

4 Steady-State Analysis

This section provides a steady-state analysis. This approximate analysis is intended to give an insight into the effect of the parameters on the steady-state tap-length adaptation step-size and error width. This analysis is convincing to put the boundaries on error width and tap-length step-size. For the convenience of the analysis, we consider several assumptions as

A1) $\mathbf{u}_{L_{optk},i}$ is independent of $\mathbf{u}_{L_{opt\ell},j}$ for $k \neq \ell$, and $i \neq j$.

A2) The regressors $\mathbf{u}_{L_{optk},i}$ are Gaussian with the eigendecomposition $R_{u,k} = Q_k \Lambda_k Q_k^*$, where Q_k is unitary and Λ_k is a diagonal matrix with the eigenvalues of $R_{u,k}$.

A3) The steady-state excess mean squared error is enough smaller than the noise variance, and consequently, the error $e_{L_k(i)}(\boldsymbol{\psi}_{k,i})$ in each node is approximately equal to the noise $\mathbf{v}_k(i)$. Note that this assumption is only feasible when the step size is so small. Nevertheless, it provides the insight we investigate

Equations (18h)–(18j) combine the updates of several interconnected algorithms and also the network topology. Therefore, subsequent analysis tends to be challenging. To proceed with this analysis, we define the following global quantities.

$$\begin{aligned}
 \Gamma^i &\triangleq \text{col}\{\gamma_1(i), \gamma_2(i), \dots, \gamma_N(i)\} & \Gamma^{(a),i} &\triangleq \text{col}\{\gamma_1^{(a)}(i), \gamma_2^{(a)}(i), \dots, \gamma_N^{(a)}(i)\} \\
 \Delta^i &\triangleq \text{col}\{\Delta_1(i), \Delta_2(i), \dots, \Delta_N(i)\} & \Delta^{(a),i} &\triangleq \text{col}\{\Delta_1^{(a)}(i), \Delta_2^{(a)}(i), \dots, \Delta_N^{(a)}(i)\} \\
 \mathbf{P}^i &\triangleq \text{col}\{\rho_1(i), \rho_2(i), \dots, \rho_N(i)\} & \mathbf{P}^{(a),i} &\triangleq \text{col}\{\rho_1^{(a)}(i), \rho_2^{(a)}(i), \dots, \rho_N^{(a)}(i)\} \\
 \mathbf{B} &\triangleq \text{diag}\{\beta_1 I_{L(\infty)}, \beta_2 I_{L(\infty)}, \dots, \beta_N I_{L(\infty)}\} & \mathbf{U}_i &\triangleq \text{diag}\{\mathbf{u}_{1,i}, \mathbf{u}_{2,i}, \dots, \mathbf{u}_{N,i}\} \\
 \mathbf{e}^i &\triangleq \text{col}\{e_{L_1(i)}(\boldsymbol{\psi}_{1,i}), e_{L_2(i)}(\boldsymbol{\psi}_{2,i}), \dots, e_{L_N(i)}(\boldsymbol{\psi}_{N,i})\} & \mathbf{v}^i &\triangleq \text{col}\{v_1(i), v_2(i), \dots, v_N(i)\} \\
 \Theta &\triangleq \text{diag}\{\theta_1, \theta_2, \dots, \theta_N\} & \mathbf{K}_\kappa &\triangleq \text{diag}\{\kappa_1, \kappa_2, \dots, \kappa_N\} \\
 \mathbf{T} &\triangleq \text{diag}\{\tau_1, \tau_2, \dots, \tau_N\} & \mathbf{H}_\eta &\triangleq \text{diag}\{\eta_1, \eta_2, \dots, \eta_N\}
 \end{aligned} \tag{19}$$

With these definitions, relations (18c) and (18h) admit the global description as follow:

$$\begin{aligned}
 \mathbf{P}^{(a),i} &= \mathbf{G}\mathbf{P}^{i-1} \\
 \mathbf{P}^i &= \mathbf{B}\mathbf{P}^{(a),i} + (\mathbf{I} - \mathbf{B})\mathbf{U}_i^T \mathbf{e}^i
 \end{aligned} \tag{20}$$

or, in an equivalent compact form: c

$$\mathbf{P}^i = \mathbf{B}\mathbf{G}\mathbf{P}^{i-1} + (\mathbf{I} - \mathbf{B})\mathbf{U}_i^T \mathbf{e}^i \tag{21}$$

with $\mathbf{G} \triangleq \mathbf{C} \otimes I_{L(\infty)}$, where \otimes stands for the Kronecker product. By assuming $\mathbf{P}^0 = \mathbf{0}$, (21) can be written as

$$\mathbf{P}^i = \sum_{n=0}^i (\mathbf{B}\mathbf{G})^{i-n} (\mathbf{I} - \mathbf{B})\mathbf{U}_n^T \mathbf{e}^n \tag{22}$$

The mean squared norm of the smoothed gradient is obtained as

$$E \left\{ \left\| \mathbf{P}^i \right\|^2 \right\} = \sum_{m=0}^i \sum_{n=0}^i E \left\{ (\mathbf{e}^m)^T \mathbf{U}_m (\mathbf{I} - \mathbf{B}) (\mathbf{G}\mathbf{B})^{i-m} (\mathbf{B}\mathbf{G})^{i-n} (\mathbf{I} - \mathbf{B}) \mathbf{U}_n^T \mathbf{e}^n \right\} \tag{23}$$

The subsequent analysis considers the summand at the steady-state. So, by assuming that the algorithm has converged, we have:

$$\begin{aligned}
 E \left\{ \left\| \mathbf{P}^i \right\|^2 \right\} &= \sum_{m=0}^i \sum_{n=0}^i E \left\{ (\mathbf{v}^m)^T \mathbf{U}_m (\mathbf{I} - \mathbf{B}) (\mathbf{G}\mathbf{B})^{i-m} (\mathbf{B}\mathbf{G})^{i-n} (\mathbf{I} - \mathbf{B}) \mathbf{U}_n^T \mathbf{v}^n \right\} \\
 &= \sum_{m=0}^i \sum_{n=0}^i E \left\{ \text{tr} \left\{ (\mathbf{v}^m)^T \mathbf{U}_m (\mathbf{I} - \mathbf{B}) (\mathbf{G}\mathbf{B})^{i-m} (\mathbf{B}\mathbf{G})^{i-n} (\mathbf{I} - \mathbf{B}) \mathbf{U}_n^T \mathbf{v}^n \right\} \right\}
 \end{aligned}$$

$$\begin{aligned}
 &= \sum_{m=0}^i \sum_{n=0}^i E \left\{ \text{tr} \left\{ \mathbf{v}^n (\mathbf{v}^m)^T U_m (I - B) (GB)^{i-m} (BG)^{i-n} (I - B) U_n^T \right\} \right\} \\
 &= \sum_{m=0}^i \sum_{n=0}^i \text{tr} \left\{ E \left\{ \mathbf{v}^n (\mathbf{v}^m)^T \right\} E \left\{ U_m (I - B) (GB)^{i-m} (BG)^{i-n} (I - B) U_n^T \right\} \right\}
 \end{aligned} \tag{24}$$

and since the noise signal is independent over time

$$E \left\{ \left\| \mathbf{P}^i \right\|^2 \right\} = \sum_{n=0}^i \text{tr} \left\{ \Lambda_v E \left\{ U_n (I - B) (GB)^{i-n} (BG)^{i-n} (I - B) U_n^T \right\} \right\} \tag{25}$$

where $\Lambda_v = \text{diag} \left\{ \sigma_{v,1}^2, \sigma_{v,2}^2, \dots, \sigma_{v,N}^2 \right\}$. By defining $H^{i-n} = (GB)^{i-n} (BG)^{i-n}$, (25) can be written as

$$E \left\{ \left\| \mathbf{P}^i \right\|^2 \right\} = \sum_{n=0}^i \text{tr} \left\{ \Lambda_v E \left\{ U_n (I - B) H^{i-n} (I - B) U_n^T \right\} \right\} \tag{26}$$

Considering assumption (A2), let to define:

$$\begin{aligned}
 \overline{U}_i &= U_i Q \\
 \overline{H}^{i-n} &= Q^T H^{i-n} Q \\
 \overline{(I - B)} &= Q^T (I - B) Q = (I - B)
 \end{aligned} \tag{27}$$

where $Q = \text{diag} \left\{ Q_1, Q_2, \dots, Q_N \right\}$. Due to the unitary property of Q , i.e., $Q^T Q = Q Q^T = I$, the expectation in (26) can be written as

$$\begin{aligned}
 &E \left\{ U_n Q Q^T (I - B) Q Q^T H^{i-n} Q Q^T (I - B) Q Q^T U_n^T \right\} \\
 &= E \left\{ \overline{U}_n (I - B) \overline{H}^{i-n} (I - B) \overline{U}_n^T \right\}
 \end{aligned} \tag{28}$$

The $k\ell$ entry of this expectation is given by

$$\left(E \left\{ \overline{U}_n (I - B) \overline{H}^{i-n} (I - B) \overline{U}_n^T \right\} \right)_{k\ell} = \begin{cases} 0 & k \neq \ell \\ (1 - \beta_k)^2 \text{tr} \left\{ \Lambda_k \overline{H}_{kk}^{i-n} \right\} & k = \ell \end{cases} \tag{29}$$

where $\overline{H}_{k\ell}^{i-n}$ is the $k\ell$ -block of \overline{H}^{i-n} . So that the summand in (26) gives

$$\text{tr} \left\{ \Lambda_v E \left\{ U_n (I - B) H^{i-n} (I - B) U_n^T \right\} \right\} = \mathbf{z}^T \overline{\mathbf{h}}^{i-n} \tag{30}$$

with $\overline{\mathbf{h}}^{i-n} = \text{bvec} \left\{ \overline{H}^{i-n} \right\}$, $\mathbf{z} = \text{bvec} \left\{ R_v (I - B)^2 \Lambda \right\}$, $\Lambda = \text{diag} \left\{ \Lambda_1, \Lambda_2, \dots, \Lambda_N \right\}$ and $R_v = \Lambda_v \odot I_{L(\infty)}$. Here, \odot denotes the block Kronecker product, and bvec stands

for block vector operator which is defined for a given matrix

$$X = \begin{bmatrix} X_{11} & X_{12} & \cdots & X_{1N} \\ X_{21} & X_{22} & \cdots & X_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ X_{N1} & X_{N2} & \cdots & X_{NN} \end{bmatrix} \tag{31}$$

as

$$\text{bvec}\{X\} = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_N \end{bmatrix} \tag{32}$$

with $X_\ell = \text{col}\{\text{vec}\{X_{1\ell}\}, \text{vec}\{X_{2\ell}\}, \dots, \text{vec}\{X_{N\ell}\}\}$, where vec stands for the standard vector operator. vector operator (30) in (26) results in

$$E\left\{\|P^i\|^2\right\} = \sum_{n=0}^i \mathbf{z}^T \bar{\mathbf{h}}^{i-n} = \mathbf{z}^T \sum_{n=0}^i \bar{\mathbf{h}}^{i-n} = \mathbf{z}^T \text{bvec}\left\{\sum_{n=0}^i \bar{H}^{i-n}\right\} \tag{33}$$

If all nodes use the same value for $\beta_k = \beta$, the matrix B can be replaced by $B = \beta I$ which gives rise to

$$\bar{H}^{i-n} = Q^T H^{i-n} Q = Q^T [(\beta G)^2]^{i-n} Q \tag{34}$$

From (33) and (34), the steady-state mean squared norm of the smoothed gradient results as

$$E\left\{\|P^\infty\|^2\right\} = \mathbf{z}^T \text{bvec}\left\{Q^T \sum_{n=0}^\infty [(\beta G)^2]^{i-n} Q\right\} = \mathbf{z}^T \text{bvec}\left\{Q^T (I - (\beta G)^2)^{-1} Q\right\} \tag{35}$$

Using the global description for (18i) and (18j), and tacking expectations of both sides of them at steady-state, and assuming that the steady-state P^i be a vector of (not necessarily independent) Gaussian random variables, leads to

$$E\{\Delta^i\} = \Theta C E\{\Delta^{i-1}\} + \sqrt{2} \frac{\Gamma((L_{opt} + 1)/2)}{\Gamma(L_{opt}/2)} \sqrt{\frac{1}{NL_{opt}} E\left\{\|P^i\|^2\right\}} \mathbf{K}_\kappa \mathbf{1} \tag{36}$$

and

$$E\{\Gamma^i\} = T C E\{\Gamma^{i-1}\} + \sqrt{2} \frac{\Gamma((L_{opt} + 1)/2)}{\Gamma(L_{opt}/2)} \sqrt{\frac{1}{NL_{opt}} E\left\{\|P^i\|^2\right\}} \mathbf{H}_\eta \mathbf{1} \tag{37}$$

In which we have used $\Delta^{(a),i} = C \Delta^{i-1}$ and $\Gamma^{(a),i} = C \Gamma^{i-1}$. Here, $\mathbf{1}$ denotes an $N \times 1$ vector whose entries are all unity. By substituting (35) in (36), and (37) we will have

$$\Delta^\infty = \sqrt{2} \frac{\Gamma((L_{opt} + 1)/2)}{\Gamma(L_{opt}/2)} \sqrt{\frac{1}{NL_{opt}} \mathbf{z}^T \text{bvec} \left\{ \mathbf{Q}^T (I - (\beta G)^2)^{-1} \mathbf{Q} \right\} (I - \Theta C)^{-1} \mathbf{K}_\kappa \mathbf{1}} \quad (38)$$

$$\Gamma^\infty = \sqrt{2} \frac{\Gamma((L_{opt} + 1)/2)}{\Gamma(L_{opt}/2)} \sqrt{\frac{1}{NL_{opt}} \mathbf{z}^T \text{bvec} \left\{ \mathbf{Q}^T (I - (\beta G)^2)^{-1} \mathbf{Q} \right\} (I - \text{TC})^{-1} \mathbf{H}_\eta \mathbf{1}} \quad (39)$$

Let to define global error width and global tap-length adaptation step-size as the average of these quantities over all nodes:

$$\Delta_{Net}(\infty) = \frac{1}{N} \sum_{k=1}^N \Delta_k(\infty) = \frac{1}{N} \mathbf{1}^T \Delta^\infty \quad (40)$$

$$\Gamma_{Net}(\infty) = \frac{1}{N} \sum_{k=1}^N \gamma_k(\infty) = \frac{1}{N} \mathbf{1}^T \Gamma^\infty \quad (41)$$

Therefore, (38)- (41) lead to

$$\Delta_{Net}(\infty) = \frac{1}{N} \sqrt{2} \frac{\Gamma((L_{opt} + 1)/2)}{\Gamma(L_{opt}/2)} \sqrt{\frac{1}{NL_{opt}} \mathbf{z}^T \text{bvec} \left\{ \mathbf{Q}^T (I - (\beta G)^2)^{-1} \mathbf{Q} \right\} \mathbf{1}^T (I - \Theta C)^{-1} \mathbf{K}_\kappa \mathbf{1}} \quad (42)$$

$$\Gamma_{Net}(\infty) = \frac{1}{N} \sqrt{2} \frac{\Gamma((L_{opt} + 1)/2)}{\Gamma(L_{opt}/2)} \sqrt{\frac{1}{NL_{opt}} \mathbf{z}^T \text{bvec} \left\{ \mathbf{Q}^T (I - (\beta G)^2)^{-1} \mathbf{Q} \right\} \mathbf{1}^T (I - \text{TC})^{-1} \mathbf{H}_\eta \mathbf{1}} \quad (43)$$

Considering Eq.(42) and Eq.(43), several parameters need to be selected suitably to ensure a large $\Delta_k(i)$ and $\gamma_k(i)$ at initial iterations of adaptation and after immediate variations, which accelerates the tap-length convergence. Also, these proper choice of parameters should ensure a small $\Delta_k(i)$ and $\gamma_k(i)$ for each node at steady-state to guarantee a small bias and small fluctuations, but not too small to make the tap-length update insensitive to instantaneous changes. The proposed algorithm is presented here to solve the problem of selecting two parameters $\Delta_k(i)$ and $\gamma_k(i)$. But now their selection depends on several other parameters, the simultaneous choice of them affects the steady-state values and, of course, the transient conditions of the algorithm. One way to overcome this problem is to constraint this parameter as $\Delta_k(i) = \max(\min(\Delta_k(i), \Delta_{\max}), \Delta_{\min})$, where $0 < \Delta_{\min} < \Delta_{\max}$ and $\gamma_k(i) = \max(\min(\gamma_k(i), \gamma_{\max}), \gamma_{\min})$, where $0 < \gamma_{\min} < \gamma_{\max}$. The upper bands are selected to guarantee that the algorithm does not fail, and the lower bands are chosen to ensure the feasibility of the algorithm. As the simulations will show in the next section, this restriction solves the problem of selecting different parameters. At the same time, the expected performance of the algorithm is obtained.

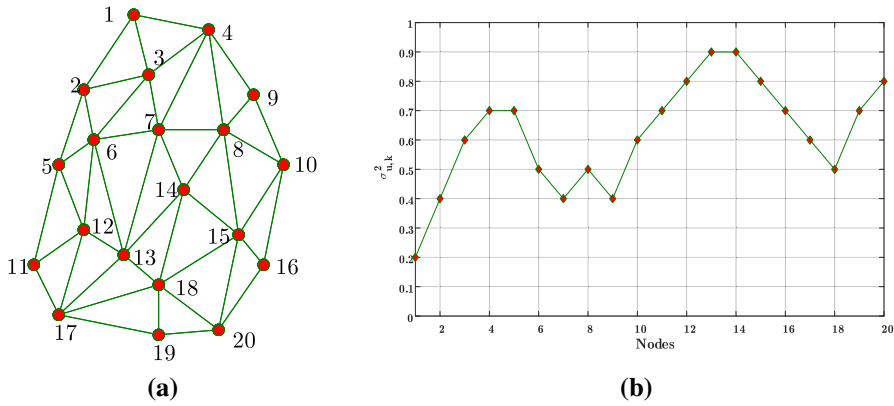


Fig. 1 Network topology (a), and regressor variances (b) for $N = 20$ nodes

5 Simulation Results

To show the performance of the proposed method, we perform several simulations in this section. First, the performance of the proposed method is tested under the low noise condition for two different optimal tap-length cases. Then, the performance of the proposed method under high noise conditions is tested. The last experiment is conducted to test the tracking behavior of the proposed algorithm, where the optimal tap-length changes unpredictably. The network topology with $N = 20$ sensors is shown in Fig 1a. The measurement noises and regressors are considered uncorrelated Gaussian processes with zero-mean, independent in space and time across the network. The regressor variances are depicted in Fig 1b.

Low noise condition with $L_{opt} = 200$: First we consider the low noise condition, where the noise variances in each node are scaled to realize high signal-to-noise ratios (SNRs) as is shown in Fig2a. The unknown vector $w_{L_{opt}}^o$ of length $L_{opt} = 200$ is drawn from a zero-mean white Gaussian sequence with variance 0.003. The parameters are set as $\mu_k = 0.008$, $\alpha_k = 0.006$, $\beta_k = 0.995$, $\theta_k = 0.94$, $\kappa_k = 60$, $\tau_k = 0.95$, $\eta_k = 50$, $\gamma_{min} = 5$, $\gamma_{max} = 35$, $\Delta_{min} = 2$, $\Delta_{max} = 30$.

Figure 3 depicts the evolution curves for different methods over diffusion adaptive networks, including the proposed algorithm, and conventional FT algorithm with $(\Delta_k = \Delta_{min}, \gamma_k = \gamma_{min})$, $(\Delta_k = \Delta_{min}, \gamma_k = \gamma_{max})$, $(\Delta_k = \Delta_{max}, \gamma_k = \gamma_{min})$, and $(\Delta_k = \Delta_{max}, \gamma_k = \gamma_{max})$. The results are averaged over 100 independent Monte Carlo experiments. The global average fractional tap-length and global average mean-square deviation (MSD) are defined, respectively, as

$$L_f^{Net}(i) = \frac{1}{N} \sum_{k=1}^N L_{k,f}(i) \tag{44}$$

$$MSD^{Net}(i) = \frac{1}{N} \sum_{k=1}^N E \left\{ \left\| w_{L_{opt}}^o - \psi_{k,i} \right\|^2 \right\} \tag{45}$$

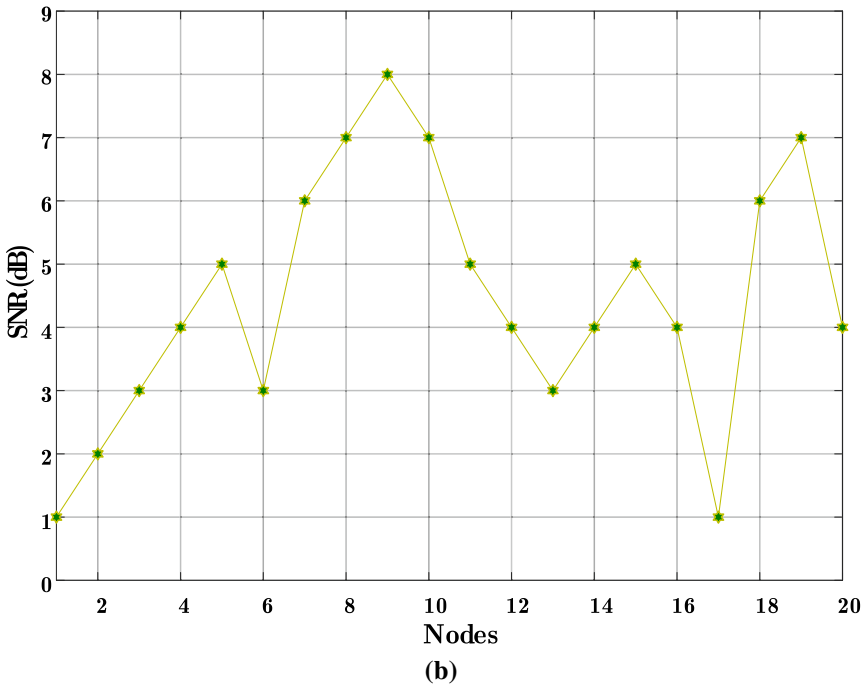
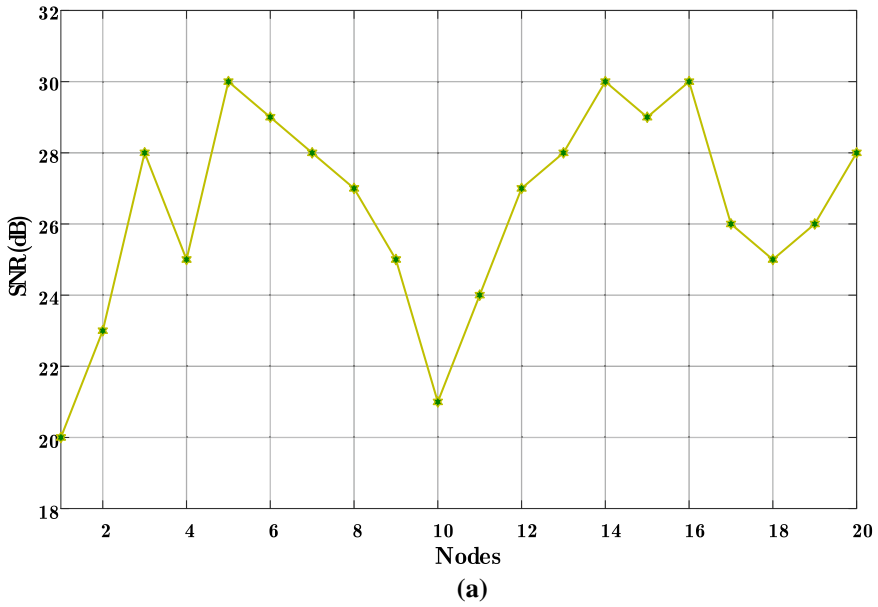


Fig. 2 The SNRs across the sensors in the network for (a) low-noise case, and (b) high-noise case

As we can see from Figure 3, the proposed algorithm converges as fast as the diffusion FT method with the fixed-parameter ($\Delta_k = \Delta_{\max}$, $\gamma_k = \gamma_{\max}$), while the steady-state tap-length in the proposed method does not have the same bias as the fixed-parameter ($\Delta_k = \Delta_{\max}$, $\gamma_k = \gamma_{\max}$). By reducing the Δ_k and γ_k from the Δ_{\max} and γ_{\max} in the fixed-parameter diffusion FT, the convergence slows down, and as can be seen from Fig. 3 it converges slowly, and of course, the length of the steady-state is biased due to the large Δ_k . In order not to have a steady-state bias, the Δ_k must be reduced to the Δ_{\min} , but in the meantime, if γ_k is selected as the γ_{\min} , convergence will not occur. For the case ($\Delta_k = \Delta_{\min}$, $\gamma_k = \gamma_{\max}$), although the steady-state bias is not expected, the convergence will be greatly reduced. Therefore, the proposed algorithm enjoys fast convergence similar to the diffusion FT with fixed-parameters ($\Delta_k = \Delta_{\max}$, $\gamma_k = \gamma_{\max}$) and has less steady-state bias similar to the diffusion FT with $\Delta_k = \Delta_{\min}$.

Figure 4 compares the performance of the proposed method with the algorithms presented in [18] and [35]. For the algorithm presented in [18], the parameters are set as $\lambda = 0.95$, $\Delta_{\max} = 30$, $\Delta(\infty) = 2$, $C = 1$, $\gamma = 35$, $\lambda_1 = 0.9$, $\lambda_2 = 0.95$. For the algorithm presented in [35], the parameters are set as $\gamma_M = 35$, $\gamma_m = 5$, $\tau = 1$, $\beta = 0.99$, $\Delta = 30$. For a fair comparison, the other parameters are considered the same for all algorithms. As can be seen, compared to the other algorithms, the proposed method shows better performance in terms of both convergence rate and steady-state error.

Figure 5 shows the evolution curves of $\Delta_k(i)$ and $\gamma_k(i)$. These parameters initiate from large values, and with the convergence of the algorithm, they end in smaller values which are suitable for steady-state condition. Note that, since both $\Delta_k(i)$ and $\gamma_k(i)$ are updated with the same measure, they follow the same path of convergence.

Note that, by increasing τ_k or η_k the width of γ_k in Fig.4 increases. By decreasing these parameters, we will have a narrower γ_k with a smaller peak, such that the peak will not reach its maximum. Also, θ_k and κ_k have a similar effect on Δ_k . As the β_k increases, the peaks of γ_k and Δ_k become narrower, and they will converge slowly.

Low noise condition with $L_{opt} = 100$:

In the second simulation, the unknown parameter length is considered to be $L_{opt} = 100$. The setup of this simulation is the same as those in the previous simulation except for $\mu_k = 0.01$, $\alpha_k = 0.01$, $\beta_k = 0.95$, and $\Delta_{\max} = 20$. The evolution curves for the proposed algorithm, and conventional FT with ($\Delta_k = \Delta_{\min}$, $\gamma_k = \gamma_{\min}$), ($\Delta_k = \Delta_{\min}$, $\gamma_k = \gamma_{\max}$), ($\Delta_k = \Delta_{\max}$, $\gamma_k = \gamma_{\min}$), and ($\Delta_k = \Delta_{\max}$, $\gamma_k = \gamma_{\max}$) are depicted in Fig 5. The definition for the global average fractional tap-length follows that in (44), and the global average MSE is defined as

$$MSE^{Net}(i) = \frac{1}{N} \sum_{k=1}^N \left(E \left\{ \left| \mathbf{u}_{L_{opt},k,i} \left(\mathbf{w}_{L_{opt}}^o - \boldsymbol{\psi}_{k,i} \right) \right|^2 \right\} + \sigma_{v,k}^2 \right) \quad (46)$$

This simulation shows that the proposed algorithm outperforms the fixed-parameters diffusion FT regardless of the unknown vector length.

High-noise condition:

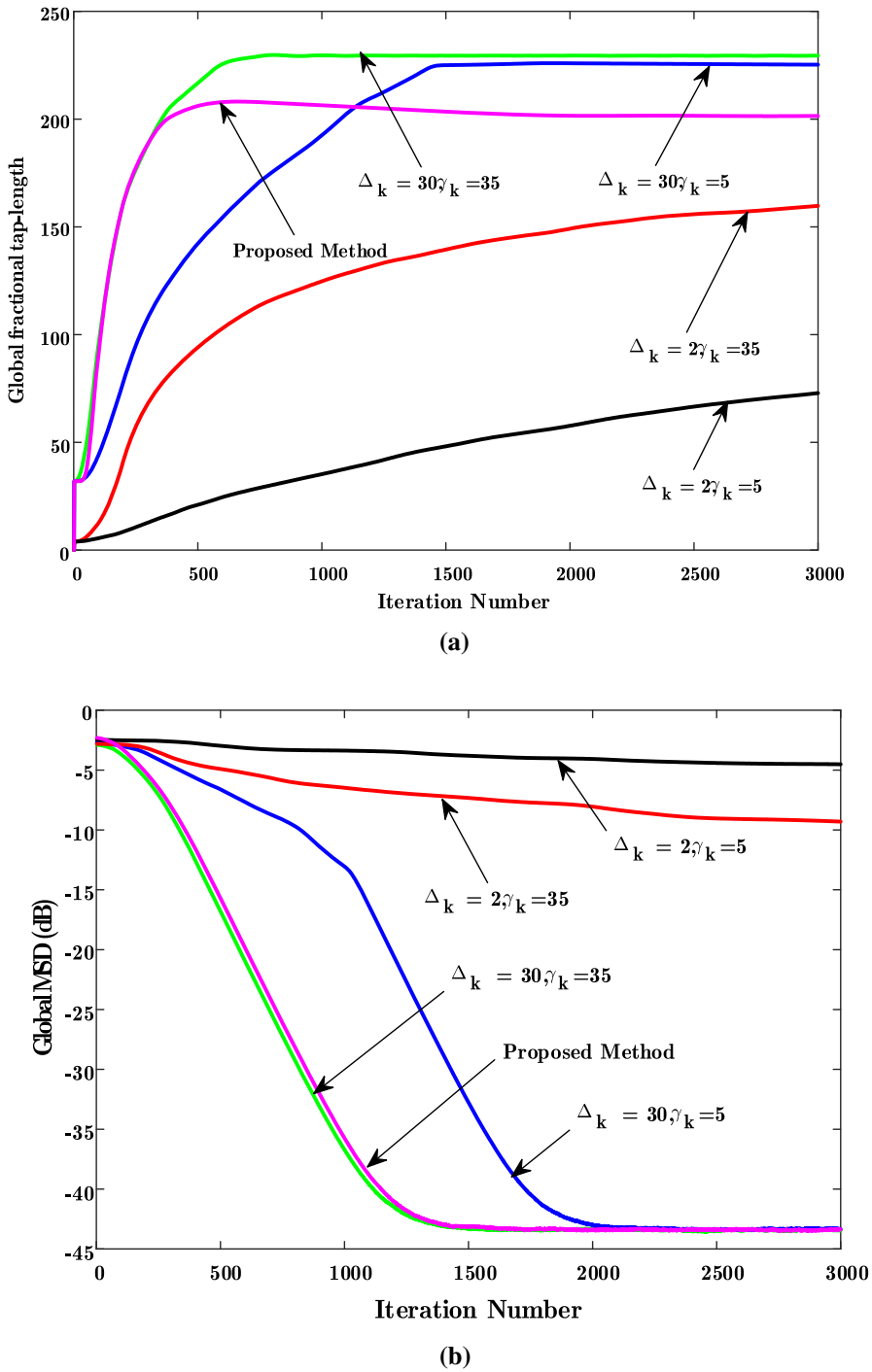
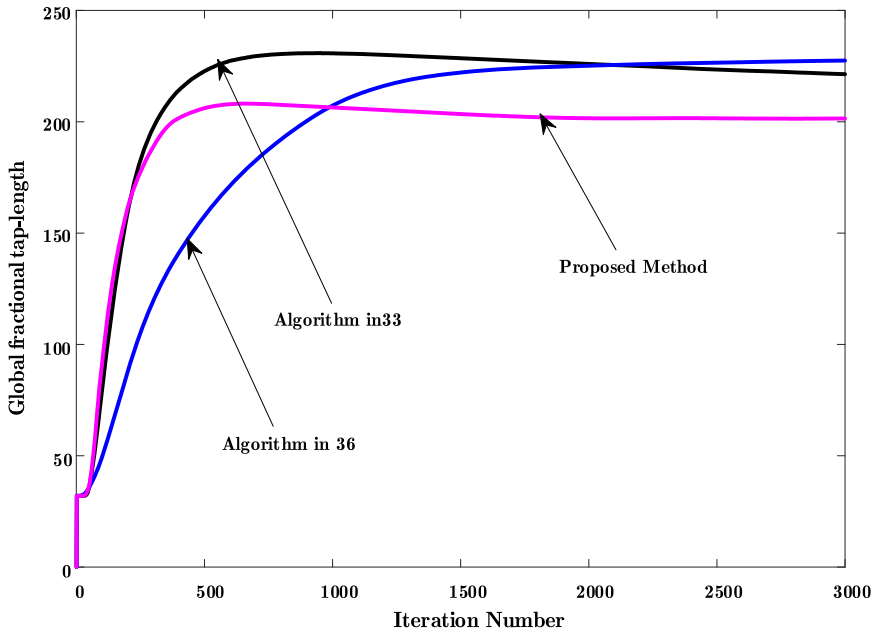
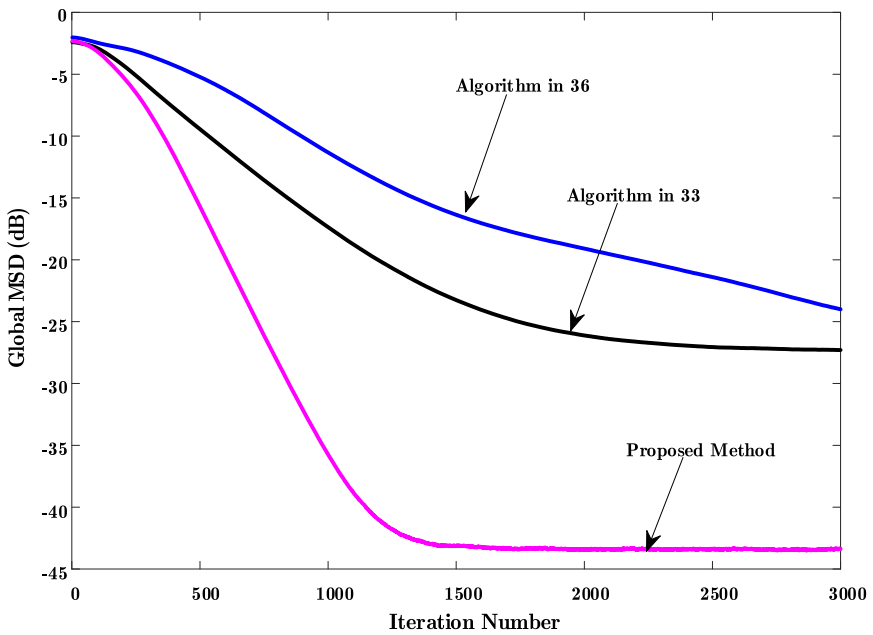


Fig. 3 The evolution curves of global tap-length (a) and MSD (b) under the low-noise condition, where $L_{opt} = 200$



(a)



(b)

Fig. 4 The evolution curves of global tap-length (a) and MSD (b) for different algorithms

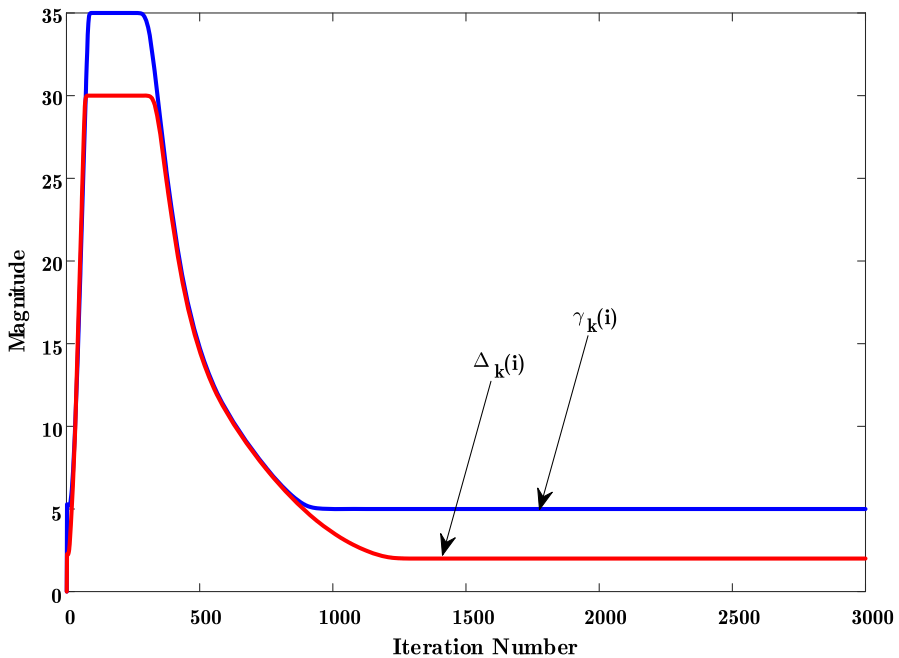
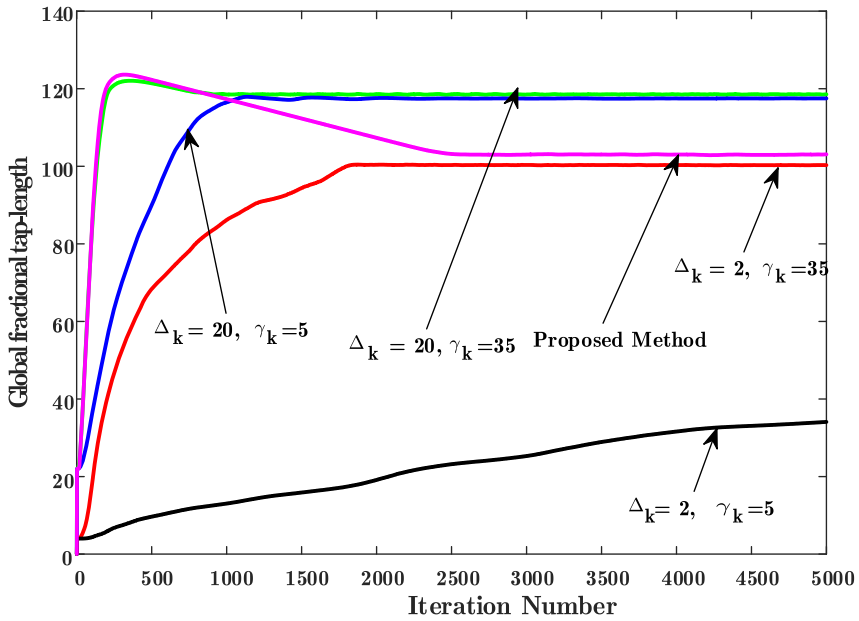


Fig. 5 Evolution curves of $\gamma_k(i)$ and $\Delta_k(i)$ of the proposed method

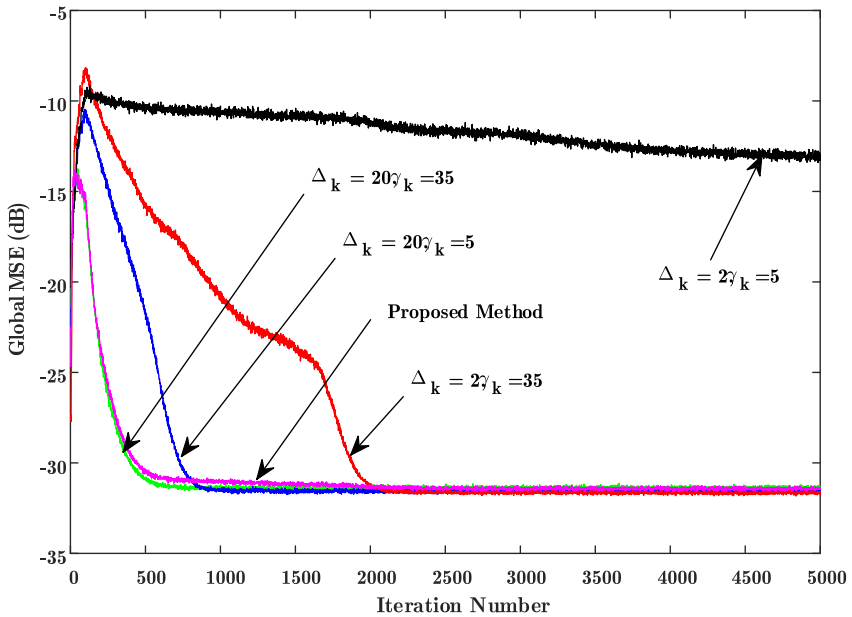
In the third simulation, a high-noise setting is employed, where noise variances are scaled to realize low SNRs, as is shown in Fig. 2b. The setup of this simulation is the same as those in the first simulation except for $\mu_k = 0.006$, $\alpha_k = 0.04$, and $\beta_k = 0.997$. The evolution curves for the proposed algorithm, and conventional FT with $(\Delta_k = \Delta_{\min}, \gamma_k = \gamma_{\min})$, $(\Delta_k = \Delta_{\min}, \gamma_k = \gamma_{\max})$, $(\Delta_k = \Delta_{\max}, \gamma_k = \gamma_{\min})$, and $(\Delta_k = \Delta_{\max}, \gamma_k = \gamma_{\max})$ are depicted in Fig. 7. This simulation shows the effect of large γ_k on the steady-state performance. As is clear from Fig. 7, the fixed parameters diffusion FT algorithm with $(\Delta_k = \Delta_{\max}, \gamma_k = \gamma_{\max})$ exhibits large fluctuations for steady-state tap-length under high noise condition. On the other hand, reducing the parameter γ_k slows down the convergence rate. However, the proposed algorithm has a high convergence thanks to the large γ_k in the initial iterations and will stay away from the steady-state fluctuations by tending the γ_k to a smaller value.

Evaluating the tracking behavior of the algorithm:

In the last experiment, we test the tracking behavior of the proposed method. The unknown vector is the same as that in the first simulation, which is drawn from a zero-mean white Gaussian signal with variance of 0.003, but the tap-length changes at the 7000th time instant from 150 to 250, and at the 14000th time instant from 250 to 150, and 81 coefficients of $w_{L_{opt}}^o$ are set to zero to model a sparse system, i.e., $w_{250}^o(150 : 230) = 0$. The setup of this simulation is the same as those in the first simulation except for $\mu_k = 0.005$, $\alpha_k = 0.05$ and $\beta_k = 0.96$. The evolution curves for the proposed algorithm, and conventional FT with $(\Delta_k = \Delta_{\min}, \gamma_k = \gamma_{\min})$,

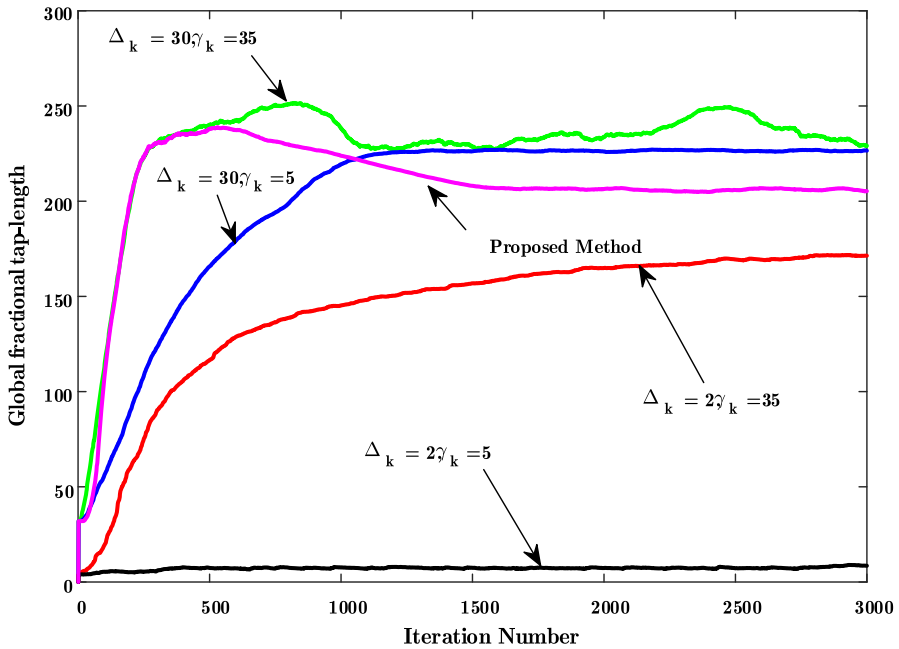


(a)

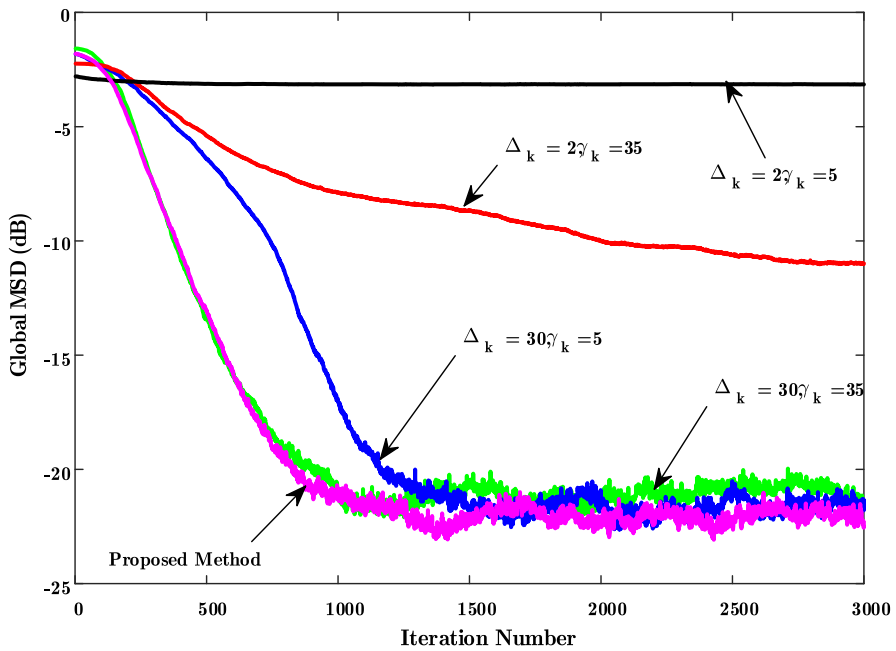


(b)

Fig. 6 The evolution curves of global tap-length (a) and MSE (b) under the low-noise condition, where $L_{opt} = 100$

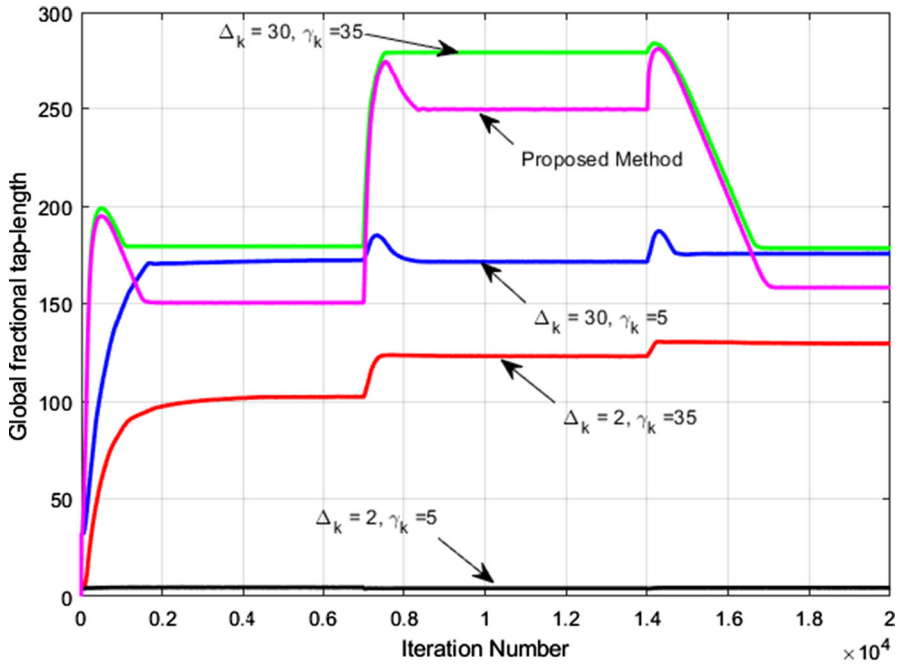


(a)

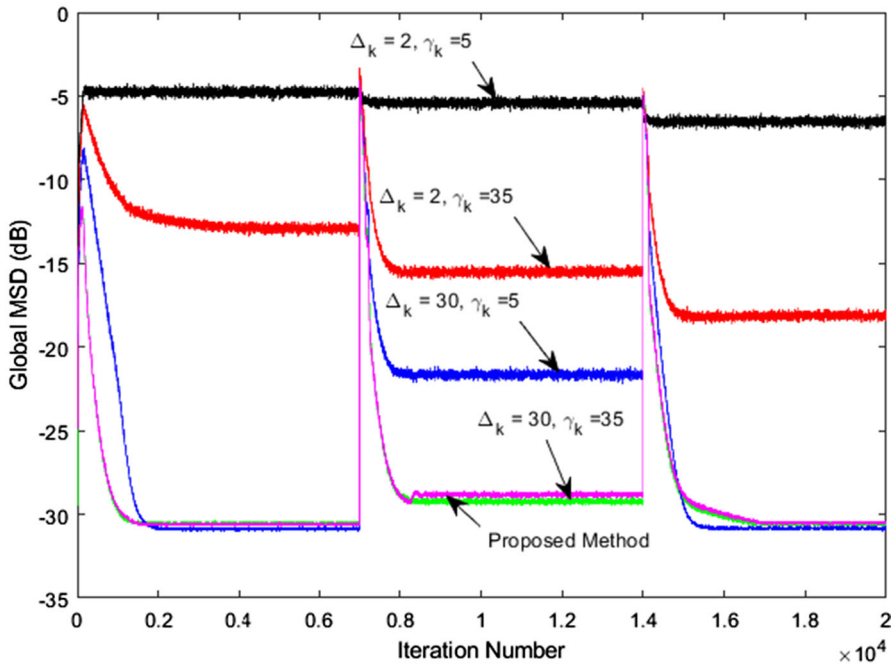


(b)

Fig. 7 The evolution curves of global tap-length (a) and MSD (b) under the high-noise condition



(a)



(b)

Fig. 8 The evolution curves of global tap-length (a) and MSD (b), where the optimal tap-length changes unpredictably

$(\Delta_k = \Delta_{\min}, \gamma_k = \gamma_{\max})$, $(\Delta_k = \Delta_{\max}, \gamma_k = \gamma_{\min})$, and $(\Delta_k = \Delta_{\max}, \gamma_k = \gamma_{\max})$ are depicted in Fig. 8.

As can be seen, to converge, track the time-varying scenario and deal with the sparsity, both large parameters for Δ_k and γ_k must be selected. But, the diffusion FT with fixed- parameters $(\Delta_k = \Delta_{\max}, \gamma_k = \gamma_{\max})$ leads to large steady-state bias. The proposed algorithm similar to the diffusion FT with fixed- parameters $(\Delta_k = \Delta_{\max}, \gamma_k = \gamma_{\max})$ converges fast and similar to the diffusion FT with fixed-parameters $(\Delta_k = \Delta_{\min}, \gamma_k = \gamma_{\min})$ enjoys from low steady-state bias. This indicates the outstanding performance of the proposed method compared to the diffusion FT with fixed- parameters.

6 Conclusions

This paper proposed an automatic scheme for selecting the parameters of the distributed fractional tap-length algorithm for adaptive wireless networks with diffusion strategy. In the proposed algorithm, error width and length adaptation step-size parameters are adjusted based on the estimated gradient vector. The error width controls the trade-off between the convergence and the steady-state tap-length bias, and the tap-length adaptation step-size provides a trade-off between tap-length fluctuation in steady-state and the convergence of the tap-length. Hence, taking a fixed value for these parameters cannot provide better performance for this algorithm. On this basis, to eliminate these trade-offs, a variable strategy is proposed for determining these parameters. In the proposed approach, these parameters are adapted to obtain an accelerated convergence in the initial iterations of the algorithm, and a less tap-length fluctuation and accurate tap-length estimation in the steady-state. The proposed method is fully distributed and implemented in diffusion cooperation. This distributed adaptation will result in the superior performance of the proposed algorithm.

Declarations

Data availability Data sharing not applicable to this article as no datasets were generated or analyzed during the current study.

References

1. M.S.E. Abadi, Z. Saffari, Distributed estimation over an adaptive diffusion network based on the family of affine projection algorithms. In: 6th International Symposium on Telecommunications (IST), pp. 607–611 (2012)
2. R. Abdolee, B. Champagne, Centralized adaptation for parameter estimation over wireless sensor networks. *IEEE Commun. Lett.* **19**(9), 1624–1627 (2015)
3. S.N. Adnani, M.A. Tinati, G. Azarnia, T.Y. Rezaei, Energy-efficient data reconstruction algorithm for spatially- and temporally correlated data in wireless sensor networks. *IET Signal Process.* **12**(8), 1053–1062 (2018)
4. M.T. Akhtar, S. Ahmed, A robust normalized variable tap-length normalized fractional lms algorithm. In: 2016 IEEE 59th International Midwest Symposium on Circuits and Systems (MWSCAS), pp. 1–4. IEEE (2016)

5. F. Albu, The constrained stability least mean square algorithm for active noise control. In: 2018 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom), pp. 1–5. IEEE (2018)
6. F. Albu, C. Paleologu, S. Ciochina, New variable step size affine projection algorithms. In: 2012 9th International Conference on Communications (COMM), pp. 63–66. IEEE (2012)
7. S.A. Alghunaim, E. Ryu, K. Yuan, A.H. Sayed, Decentralized proximal gradient algorithms with linear convergence rates. *IEEE Transactions on Automatic Control* pp. 1–1 (2020)
8. S.A. Alghunaim, K. Yuan, A.H. Sayed, A proximal diffusion strategy for multi-agent optimization with sparse affine constraints. *IEEE Transactions on Automatic Control* pp. 1–1 (2019)
9. G. Azarnia, M.A. Tinati, Steady-state analysis of the deficient length incremental lms adaptive networks. *Circuits, Syst. Signal Process.* **34**(9), 2893–2910 (2015)
10. G. Azarnia, M.A. Tinati, Steady-state analysis of the deficient length incremental lms adaptive networks with noisy links. *AEU-Int. J. Electron. Commun.* **69**(1), 153–162 (2015)
11. G. Azarnia, M.A. Tinati, T.Y. Rezaei, Cooperative and distributed algorithm for compressed sensing recovery in wsns. *IET Signal Process.* **12**(3), 346–357 (2018)
12. G. Azarnia, M.A. Tinati, T.Y. Rezaei, Generic cooperative and distributed algorithm for recovery of signals with the same sparsity profile in wireless sensor networks: a non-convex approach. *J. Supercomput.* **75**(5), 2315–2340 (2019)
13. G. Azarnia, M.A. Tinati, A.A. Sharifi, H. Shiri, Incremental and diffusion compressive sensing strategies over distributed networks. *Digital Signal Processing* **101**, 102732 (2020). <https://doi.org/10.1016/j.dsp.2020.102732>. <http://www.sciencedirect.com/science/article/pii/S1051200420300774>
14. X. Cao, K.J.R. Liu, Decentralized sparse multitask rls over networks. *IEEE Trans. Signal Process.* **65**(23), 6217–6232 (2017)
15. F.S. Cattivelli, C.G. Lopes, A.H. Sayed, Diffusion recursive least-squares for distributed estimation over adaptive networks. *IEEE Trans. Signal Process.* **56**(5), 1865–1877 (2008)
16. S. Chen, Y. Liu, Robust distributed parameter estimation of nonlinear systems with missing data over networks. *IEEE Trans. Aerospace Electron. Syst.* **56**(3), 2228–2244 (2020)
17. M. Ferrer, M. de Diego, G. Piñero, A. Gonzalez, Active noise control over adaptive distributed networks. *Signal Process.* **107**, 82–95 (2015)
18. Y. Han, M. Wang, M. Liu, An improved variable tap-length algorithm with adaptive parameters. *Digital Signal Process.* **74**, 111–118 (2018)
19. A. Kar, T. Padhi, B. Majhi, M.N.S. Swamy, Analysing the impact of system dimension on the performance of a variable-tap-length adaptive algorithm. *Appl. Acoustics* **150**, 207–215 (2019)
20. C. Li, S. Huang, Y. Liu, Z. Zhang, Distributed jointly sparse multitask learning over networks. *IEEE Trans. Cybernet.* **48**(1), 151–164 (2018)
21. L. Li, J. Feng, J. He, J.A. Chambers, A distributed variable tap-length algorithm within diffusion adaptive networks. *AASRI Procedia* **5**, 77 – 84 (2013). <https://doi.org/10.1016/j.aasri.2013.10.061>. <http://www.sciencedirect.com/science/article/pii/S2212671613000620>. 2013 AASRI Conference on Parallel and Distributed Computing and Systems
22. L. Li, Y. Zhang, J.A. Chambers, Variable length adaptive filtering within incremental learning algorithms for distributed networks. In: 2008 42nd Asilomar Conference on Signals, Systems and Computers, pp. 225–229 (2008)
23. N. Li, Y. Zhang, Y. Zhao, Y. Hao, An improved variable tap-length LMS algorithm. *Signal Process.* **89**(5), 908–912 (2009)
24. Z. Li, W. Shi, M. Yan, A decentralized proximal-gradient method with network independent step-sizes and separated convergence rates. *IEEE Trans. Signal Process.* **67**(17), 4494–4506 (2019)
25. Z. Liu, Variable tap-length linear equaliser with variable tap-length adaptation step-size. *Electron. Lett.* **50**(8), 587–589 (2014)
26. C.G. Lopes, A.H. Sayed, Diffusion least-mean squares over adaptive networks: Formulation and performance analysis. *IEEE Trans. Signal Process.* **56**(7), 3122–3136 (2008)
27. K. Mayyas, H.A. AbuSeba, A new variable length nlms adaptive algorithm. *Digital Signal Process.* **34**, 82–91 (2014)
28. C. Paleologu, J. Benesty, S. Ciochina, A variable step-size affine projection algorithm designed for acoustic echo cancellation. *IEEE Trans. Audio, Speech, and Language Process.* **16**(8), 1466–1478 (2008)

29. 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. *Appl. Acoustics* **159**, 107074 (2020)
30. A. Rastegarnia, Reduced-communication diffusion rls for distributed estimation over multi-agent networks. *IEEE Trans. Circuits Syst. II: Exp. Briefs* **67**(1), 177–181 (2020)
31. F. Riera-Palou, J.M. Noras, D.G.M. Cruickshank, Linear equalisers, with dynamic and automatic length selection. *Electron. Lett.* **37**(25), 1553–1554 (2001)
32. M.O.B. Saeed, A. Zerguine, An incremental variable step-size LMS algorithm for adaptive networks. *IEEE Transactions on Circuits and Systems II: Express Briefs* pp. 1–1 (2019)
33. K. Shravan Kumar, N.V. George, Polynomial sparse adaptive estimation in distributed networks. *IEEE Transactions on Circuits and Systems II: Express Briefs* **65**(3), 401–405 (2018)
34. N. Takahashi, I. Yamada, A.H. Sayed, Diffusion least-mean squares with adaptive combiners: Formulation and performance analysis. *IEEE Trans. Signal Proces.* **58**(9), 4795–4810 (2010)
35. Y. Wei, Z. Yan, Variable tap-length lms algorithm with adaptive step size. *Circuits, Syst. Signal Process.* **36**(7), 2815–2827 (2017)
36. S. Xu, R.C. de Lamare, H.V. Poor, Distributed estimation over sensor networks based on distributed conjugate gradient strategies. *IET Signal Process.* **10**(3), 291–301 (2016)
37. G. Yu, C.F.N. Cowan, An lms style variable tap-length algorithm for structure adaptation. *IEEE Trans. Signal Process.* **53**(7), 2400–2407 (2005)
38. G. Yuantao, T. Kun, C. Huijuan, Lms algorithm with gradient descent filter length. *IEEE Signal Process. Lett.* **11**(3), 305–307 (2004)

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