Signal Processing 153 (2018) 243–254

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Diffusion total least-squares algorithm with multi-node feedback

Lu Lu[a], Haiquan Zhao[b,*], Benoit Champagne[c]

[a] School of Electronics and Information Engineering, Sichuan University, Chengdu, China
[b] School of Electrical Engineering, Southwest Jiaotong University, Chengdu, China
[c] Department of Electrical and Computer Engineering, McGill University, Montreal, QC, Canada

ARTICLE INFO

Article history:
Received 29 November 2017
Revised 17 July 2018
Accepted 30 July 2018
Available online 31 July 2018

Keywords:
Distributed adaptation
Total least-squares
Multi-node feedback
Lyapunov stability theory
Errors-in-variables (EIV) model

ABSTRACT

The errors-in-variables (EIV) system model has been widely studied under the assumption that both input and output signals are contaminated with noise. For the in-network distributed linear system identification problem under the EIV model, the total least-squares (TLS) approach which has the ability to minimize perturbations in both input and output signals offers an efficient solution. In this paper, we propose an improved diffusion total least-squares algorithm, where the estimated (i.e. filtered) value at each node is passed through a first order recursive filter with adjustable parameter in order to enhance the identification performance. The resulting outputs from all the nodes are subsequently used to adapt the unknown linear system weight vector in real-time through a cooperative diffusion scheme based on the adapt-then-combine (ATC) policy. We also present robust adaptive strategies to tune various internal system parameters, such as the steps sizes, normalization factors, etc., under practical conditions of operation. The convergence behavior of the adaptive weight vector and related system parameters is analyzed by employing Lyapunov stability theory. Simulation results for various distributed system identification scenarios demonstrate the effectiveness of the proposed algorithm.

1. Introduction

Over the past decade, several adaptive estimation algorithms have been proposed to tackle the in-network distributed linear system identification problem, whereby individual nodes take advantage of local measurements and information exchange with their neighbors to cooperatively estimate an unknown regression weight vector [1,2]. Compared with the conventional centralized estimation approach, distributed estimation does not need a fusion center; consequently, it consumes less energy and communication resources and is therefore particularly well-suited for wireless sensor network applications. Moreover, this method is flexible and robust to link failures, which is especially important for large sensor networks. Due to these advantages, distributed estimation has been successfully applied to various in-network estimation problems, e.g., active learning, frequency estimation, and decision-making [3–6]. Three different strategies are commonly used for the implementation of distributed estimation algorithms, namely: the incremental [7–9], consensus [10–12], and diffusion strategies [13–16]. Among these, the diffusion strategy demonstrates the most reliable performance regardless of the network topology and as such, it has been used to derive many general purpose distributed adaptive estimation algorithms [17–22]. In addition, many other diffusion algorithms have been developed for specific applications, such as tracking and localizaton of mobile terminals in cellular networks [23–26].

In the context of system identification, the least squares (LS) algorithm has been introduced mainly for the case where only the linear system output is affected by noise [27]. However, when both the input (excitation) and output (desired) signals are corrupted by additive noise, the LS algorithm can only achieve a suboptimal solution [28]. Yet, this situation is encountered in many areas of science and engineering, such as control engineering, signal processing, and econometrics [29–32]. Studies in these areas have shown that the errors-in-variables (EIV) model can describe this type of system more accurately [29–32].

The bias-compensation (BC) method is an effective means for EIV modeling [33]. In particular, by exploiting statistical properties of the input noise, several unbiased estimation algorithms have been derived for the linear adaptive filters [33–38]. In [34], the consistent normalized least mean square (CNLMS) algorithm was proposed for noisy inputs. Nevertheless, the input noise is not fully compensated by the CNLMS algorithm owing to its use of a bounded estimate of the input noise variance. The BC method has also been extensively investigated for the in-network distributed system identification of EIV models [39–41], including the BC-

* Corresponding author.
E-mail addresses: lulu19900303@126.com (L. Lu), hqzhao_swjtu@126.com (H. Zhao), benoit.champagne@mcgill.ca (B. Champagne).

https://doi.org/10.1016/j.sigpro.2018.07.025

diffusion recursive least squares (BC-DRLS) [39] and BC-diffusion least mean square (BC-DLMS) algorithms [41].

Following a different direction, the error whitening criterion (EWC) method was proposed for system identification with white noise contaminated inputs [42–44]. The EWC method enforces zero autocorrelation of the error signal beyond a certain lag. Hence, it can provide unbiased parameter estimates in the presence of additive white noise with arbitrary power in both system input and output. However, the performance of EWC is not satisfactory when the noise is temporally correlated or when the unknown system introduces nonlinear distortions. Several avenues have been pursued to overcome these drawbacks [45–48]. In [47], the modified EWC algorithm was proposed, which can consistently estimate the parameters of a linear system in the presence of colored input noises, without the need to compute the input noise covariance matrix. The algorithm in [48] offers an analytical solution for the extension of the EWC to the second-order Volterra system, where the latter is expressed as an augmented linear system model.

The total least squares (TLS) approach is a well-known alternative to solve the system identification problem for the EIV system model [29–32,49,50]. The research on TLS dates back to the1980′s, and so far a large number of TLS-based algorithms have been proposed [28,51–53]. TLS algorithms aim to minimize the sum of squared "total" errors needed to best fit the input and output signals. Because they take into account observational errors in both types of variables, they achieve a better performance than the LS algorithms in EIV modeling [28]. In previous studies focusing on the case of single-node processing devices, the TLS approach has been applied to the solution of various problems, such as the localization of wideband signals [54] and system identification [49]. Recently, extension of the TLS formulation to the problem of in-network distributed system identification has been considered by several researchers [55–61]. In [55] and [56], two consensus-based TLS algorithms were proposed, which utilize eigendecomposition of the augmented data covariance matrix and the inverse power iterations (IPI), respectively; unfortunately, these algorithms suffer from relatively high computational complexity, which limits their applications. To surmount this shortcoming, the diffusion gradient-descent total least squares (D-GDTLS) algorithm was proposed [57]. Unlike the BC method, the D-GDTLS algorithm does not require *a priori information* about the input noise statistics. In addition, its performance is superior to that of the BC-DLMS algorithm for in-network system identification under the EIV model [57].

It is worth noticing that the above-mentioned algorithms do not exploit the filter output as a useful feedback mechanism to further improve estimation. In fact, the feedback strategy can improve either the convergence rate or the misadjustment of adaptive algorithms, and as such, it has attracted much attention with fruitful applications in the areas of kernel adaptive filters [62–65], and active noise control [66]. Nonetheless, in these works and other similar references on the topic, the solutions rely on the use of single-node processing devices while little attention is paid to in-network distributed processing solutions.

Motivated by these considerations, in this paper, we propose a new diffusion-based TLS algorithm with multi-node feedback for in-network distributed system identification under the EIV model. This approach makes it possible to take advantage of prior output information from the EIV system, thereby enabling the newly proposed algorithm to achieve a smaller misadjustment in steady-state. Moreover, robust adaptive strategies for various internal system parameters (step-sizes, normalization factors, etc.) are provided to facilitate its practical use. We analyze the stability and convergence of the weight vector and system parameters for the proposed algorithm according to Lyapunov stability theory (LST). Simulation results for distributed estimation of EIV system models show that the proposed algorithm has superior performance as compared with existing algorithms.

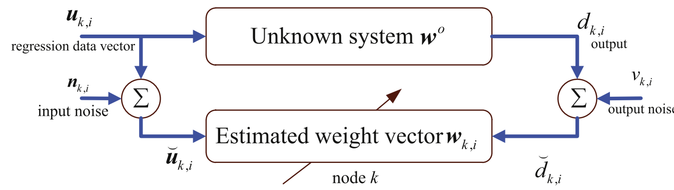

Fig. 1. EIV system identification model for node k.

The paper is organized as follows. In Section 2, we formulate the in-network distributed linear system identification problem under the EIV model. In Section 3, we present the proposed diffusion-based TLS algorithm with feedback in detail, including robust adaptive strategies for on-line tuning of internal parameters. In Section 4, the computational complexity of the proposed algorithm is compared with that of the DLMS and D-GDTLS algorithms. In Section 5, the convergence analysis of the proposed algorithm is performed based on LST. Results of numerical simulations are presented in Section 6 to illustrate the effectiveness and advantages of the proposed algorithm. Finally, Section 7 presents the conclusions and future lines of research.

Notation: In this paper, $\mathrm{E}(\cdot)$ denotes the mathematical expectation, $(\cdot)^{\mathrm{T}}$ denotes the transposition, $\|\cdot\|_p$ stands for the l_p norm of its argument, and $\mathrm{diag}\{\cdot\}$ represents a diagonal matrix with scalar or matrix entries indicated by the argument.

2. Problem formulation

Consider a sensor network composed of N processing nodes distributed over a geographic area, where the nodes cooperate to perform in-network distributed system identification. Under the EIV model [60], as shown in Fig. 1, the observed input and output signals to the unknown system at node $k \in \{1, \dots, N\}$, are *both* contaminated by noise. Specifically, it is assumed that at time instant $i \in \mathbb{N}$, node k has access to the noisy observations $\{\breve{d}_{k,i}, \breve{\boldsymbol{u}}_{k,i}\}$, as follows[1,2]:

$$\begin{cases} \breve{d}_{k,i} = d_{k,i} + v_{k,i} \\ \breve{\boldsymbol{u}}_{k,i} = \boldsymbol{u}_{k,i} + \boldsymbol{n}_{k,i} \end{cases} \tag{1}$$

where $d_{k,i} = \boldsymbol{u}_{k,i}^{\mathrm{T}} \boldsymbol{w}^o$ is the scalar output of the unknown system, $\boldsymbol{w}^o$ is an $M \times 1$ unknown parameter vector to be estimated, $\boldsymbol{u}_{k,i}$ is the $M \times 1$ input signal vector to the unknown system (regression data) modeled as a zero-mean random vector, $v_{k,i}$ is a zero-mean additive Gaussian noise, $\breve{d}_{k,i}$ is the observed noisy output signal (desired signal), $\boldsymbol{n}_{k,i}$ is a zero-mean Gaussian noise vector, and $\breve{\boldsymbol{u}}_{k,i}$ is the observed noisy input signal vector. Moreover, $\boldsymbol{u}_{k,i}$, $\boldsymbol{n}_{k,i}$ and $v_{k,i}$ are assumed to be statistically independent from each other. For $v_{k,i} = 0$, (1) reduces to the standard data LS system model [27].

Different from the LS-based algorithms, the cost function of the TLS algorithm in a diffusion network is formulated as a linear combination of local weighted mean-squared total errors [57]

$$J_k^{\mathrm{loc}}(\boldsymbol{w}) = \sum_{l \in \mathcal{N}_k} a_{l,k} \mathrm{E}\left[e_{l,i}^2\right] \tag{2}$$

where $\mathcal{N}_k$ is the index set of nodes in the neighborhood of node k, and $a_{l,k} \geq 0$ are non-negative real constants satisfying the following

[1] For simplicity, we shall refer to this model as the 'diffusion EIV model' in the sequel.

[2] Unless otherwise indicated, scalar quantities take values in the set of real number, $\mathbb{R}$.

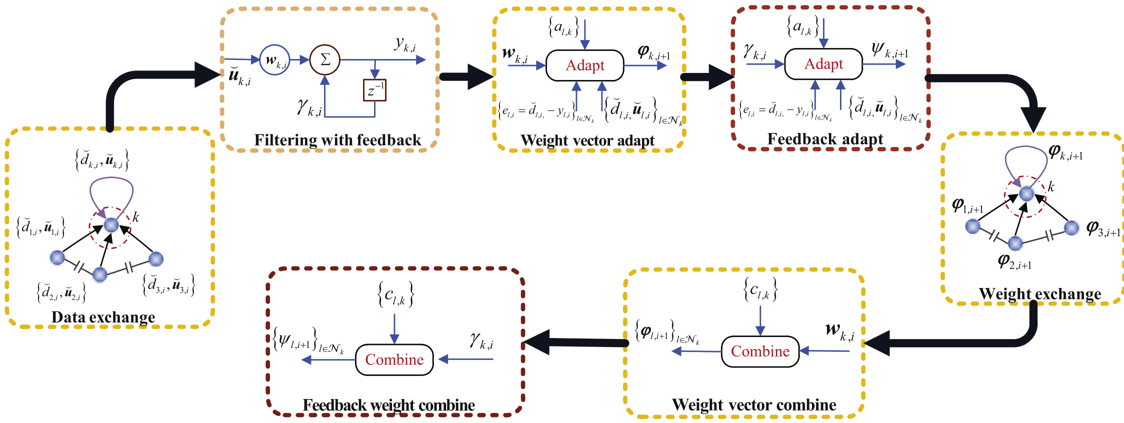

Fig. 2. Schematic diagram of the proposed DTLS algorithm with feedback at node k. At time i, the algorithm starts with the observation and exchange of data (left) and proceeds through various steps are indicated.

conditions

$$a_{l,k} = 0 \ \text{ if } l \notin \mathcal{N}_k, \ \text{ and } \ \sum_{k=1}^{N} a_{l,k} = 1. \tag{3}$$

The total error at time i is defined as [53]

$$e_{l,i} = \frac{\breve{d}_{l,i} - \breve{\boldsymbol{u}}_{l,i}^{\mathrm{T}} \boldsymbol{w}}{\sqrt{\boldsymbol{w}^{\mathrm{T}} \boldsymbol{\Pi}_l^{-2} \boldsymbol{w} + \varepsilon_{0,l}^{-2}}} \tag{4}$$

where $\boldsymbol{\Pi}_l = \mathrm{diag}\{\varepsilon_{1,l}, \ldots, \varepsilon_{M,l}\}$ is a nonsingular weighting matrix and $\varepsilon_{0,l}, \ldots, \varepsilon_{M,l}$ are the positive constants.

Several TLS-based adaptive algorithms have been developed to solve the system identification problem under the EIV model [55–57]. Among these, the D-GDTLS algorithm achieves the best performance with low computational complexity. However, the weight adaptation of the D-GDTLS algorithm only employs the current observations $\{\breve{d}_{k,i}, \breve{\boldsymbol{u}}_{k,i}\}$. To further improve the performance of this algorithm, it is natural to consider designing a novel DTLS algorithm with a feedback strategy.

3. Proposed algorithm

As stated in Section 1, the feedback scheme can be used to improve the estimation performance and accelerate the convergence rate of adaptive filters for time series prediction [63]. The diffusion EIV model has gained importance in recent years, but no work has been reported addressing the distributed adaptation with feedback. Fig. 2 summarizes the schematic diagram of the proposed DTLS algorithm with multi-node feedback, where $\boldsymbol{w}_{k,\,i}$ and $\gamma_{k,\,i}$ respectively denote the estimated weight vector and feedback weight at node k and discrete-time i. In this algorithm, a first-order (or single-step) feedback mechanism is employed at each node comprising the network.

Define the output signal of the adaptive filter at node k as

$$y_{k,i} = \breve{\boldsymbol{u}}_{k,i}^{\mathrm{T}} \boldsymbol{w}_{k,i} + \gamma_{k,i} y_{k,i-1} \tag{5}$$

where the first term on the right-hand side (RHS) is the conventional output signal according to the EIV model and the second term is the feedback (self-recurrent part) from the node k which exploits the previous output $y_{k,i-1}$, as represented by the unit delay operator z^{-1} in Fig. 2. Following the developments in [63,65], there exists an optimal feedback weight $\gamma_{k,i}^{o}$ such that with probability one, $y_{k,i}^{o} = \breve{\boldsymbol{u}}_{k,i}^{\mathrm{T}} \boldsymbol{w}^{o} + \gamma_{k,i}^{o} y_{k,i-1}$ holds, where $y_{k,i}^{o} = d_{k,i}$ is the desired signal. Consequently, the feedback strategy can improve the convergence rate or the misadjustment of adaptive algorithms, as previously evidenced in the areas of kernel adaptive filter [63] and active noise control [66]. Another motivation for introducing this optimal feedback is that it enables us to reveal the link between the total error signal at node k and the other key variables in the feedback strategy, thereby facilitating the derivation of the final DTLS algorithm.

Specifically, the total error signal of the proposed algorithm can be expressed as

$$\begin{aligned} e_{k,i} &\triangleq \vartheta_{k,i}\left(\breve{d}_{k,i} - y_{k,i}\right) \\ &= \vartheta_{k,i}\left(\breve{\boldsymbol{u}}_{k,i}^{\mathrm{T}}\left(\boldsymbol{w}^{o} - \boldsymbol{w}_{k,i}\right) + (\gamma_{k,i}^{o} - \gamma_{k,i}) y_{k,i-1} + \upsilon_{k,i}\right) \end{aligned} \tag{6}$$

where we define

$$\vartheta_{k,i} = \frac{1}{\sqrt{\boldsymbol{w}_{k,i}^{\mathrm{T}} \boldsymbol{\Pi}_k^{-2} \boldsymbol{w}_{k,i} + \varepsilon_{0,k}^{-2}}}. \tag{7}$$

The proposed DTLS algorithm with multi-node feedback seeks the unknown parameter vector $\boldsymbol{w}^{o}$ by minimizing the following global cost function:

$$J^{\mathrm{glo}}(\boldsymbol{w}, \gamma) = \sum_{k=1}^{N} J_k^{\mathrm{loc}}(\boldsymbol{w}, \gamma) \tag{8}$$

where $J_k^{\mathrm{loc}}(\boldsymbol{w}, \gamma)$ is the local cost function associated with node k, that is,

$$J_k^{\mathrm{loc}}(\boldsymbol{w}, \gamma) = \frac{1}{2} \mathrm{E}\left(e_{k,i}^2\right). \tag{9}$$

Adaptive learning is applied to both the weight vector $\boldsymbol{w}$ and feedback weight γ parameters by minimizing the global cost function $J_k^{\mathrm{glo}}(\boldsymbol{w}, \gamma)$. In particular, using the gradient-descent method, these parameters can be updated as follows:

$$\boldsymbol{w}_{i+1} = \boldsymbol{w}_i - \eta \left[\sum_{k=1}^{N} \nabla_{\boldsymbol{w}} J_k^{\mathrm{loc}}(\boldsymbol{w}, \gamma) \right] \tag{10}$$

$$\gamma_{i+1} = \gamma_i - \mu \left[\sum_{k=1}^{N} \nabla_{\gamma} J_k^{\mathrm{loc}}(\boldsymbol{w}, \gamma) \right] \tag{11}$$

where η and μ are the corresponding (positive) step sizes. Eqs. (10) and (11) are derived from a centralized learning principle (steepest descent) to iteratively minimize $J^{\mathrm{glo}}(\boldsymbol{w}, \gamma)$. In a traditional centralized solution, all the observations are collected and processed in a fusion center. This mode of operation requires large amounts of energy and communication resources, which may prohibit its practical applications [57,60]. Furthermore, the gradient-descent method needs to select suitable values of the step sizes and in many cases, several other operating parameters need to be specified. As a result, the performance of such an algorithm becomes highly dependent on the parameter selection. To overcome these limitations, we focus on the in-network distributed solution in this paper. Besides, we will develop robust adaptive strategies for the proposed algorithm in order to tackle the parameter selection problem.

3.1. Weight adaptation

In the multi-node feedback scheme, the weight vector $\boldsymbol{w}_{k,i}$ and the feedback weight $\gamma_{k,i}$ will be adapted separately by using their own gradient information. To begin with, the error signal in (6) can be rewritten as

$$\begin{aligned} e_{k,i} &= \vartheta_{k,i}\Big[-\big(\breve{\boldsymbol{u}}^{\mathrm{T}}_{k,i}\tilde{\boldsymbol{w}}_{k,i}+\gamma_{k,i}y_{k,i-1}\big)+\gamma^{o}_{k,i}y_{k,i-1}+v_{k,i}\Big] \\ &= -\xi_{k,i}+\Delta^{\gamma}_{k,i} \end{aligned} \tag{12}$$

where the weight error vector is defined as

$$\tilde{\boldsymbol{w}}_{k,i} \triangleq \boldsymbol{w}_{k,i}-\boldsymbol{w}^{o}, \tag{13}$$

the *a priori* error is defined as

$$\xi_{k,i} \triangleq \vartheta_{k,i}\big(\breve{\boldsymbol{u}}^{\mathrm{T}}_{k,i}\tilde{\boldsymbol{w}}_{k,i}+\gamma_{k,i}y_{k,i-1}\big), \tag{14}$$

and the equivalent disturbance (which reduces the error in the feedback process) is defined as

$$\Delta^{\gamma}_{k,i} \triangleq \vartheta_{k,i}\big[\gamma^{o}_{k,i}y_{k,i-1}+v_{k,i}\big]. \tag{15}$$

The error signal can also be expressed in terms of the feedback weight error as

$$e_{k,i}=-\varsigma_{k,i}+\Delta^{w}_{k,i} \tag{16}$$

where we define

$$\varsigma_{k,i} \triangleq \vartheta_{k,i}y_{k,i-1}\tilde{\gamma}_{k,i} \tag{17}$$

the feedback weight error is defined as

$$\tilde{\gamma}_{k,i} \triangleq \gamma_{k,i}-\gamma^{o}_{k,i} \tag{18}$$

and the equivalent disturbance $\Delta^{w}_{k,i}$ is defined as

$$\Delta^{w}_{k,i} \triangleq \vartheta_{k,i}\big[-\breve{\boldsymbol{u}}^{\mathrm{T}}_{k,i}\tilde{\boldsymbol{w}}_{k,i}+v_{k,i}\big]. \tag{19}$$

Borrowing from the field of artifical neural networks, let use denote by $\boldsymbol{D}_k(i)$ the *hybrid derivative* of the output $y_{k,i}$ with respect to the weight vector, which can be computed as [62]

$$\begin{aligned} \boldsymbol{D}_k(i) &= \nabla_{\boldsymbol{w}_{k,i}}y_{k,i} \\ &= \frac{\partial y_{k,i}}{\partial \boldsymbol{w}_{k,i}}+\beta_k(i)\frac{\partial y_{k,i}}{\partial y_{k,i-1}}\frac{\partial y_{k,i-1}}{\partial \boldsymbol{w}_{k,i}}. \end{aligned} \tag{20}$$

In this expression, $\beta_k(i)$ is the so-called hybrid learning rate of $\boldsymbol{D}_k(i)$, which serves to constrain the recurrent gradient term, i.e., $\frac{\partial y_{k,i}}{\partial y_{k,i-1}}\frac{\partial y_{k,i-1}}{\partial \boldsymbol{w}_{k,i}}$. We note that the hybrid derivative of output $y_{k,i}$ is linearly related to the previous derivative through the recurrent gradient term. This makes it possible to adapt the gradient information in a recursive manner by reusing the past gradient information [62,63]. Upon substitution of $\boldsymbol{w}_{k,i}$ by $\boldsymbol{w}_{k,i-1}$ in the second term of (20) and using (5), we obtain

$$\begin{aligned} \boldsymbol{D}_k(i) &\approx \frac{\partial y_{k,i}}{\partial \boldsymbol{w}_{k,i}}+\beta_k(i)\frac{\partial y_{k,i}}{\partial y_{k,i-1}}\frac{\partial y_{k,i-1}}{\partial \boldsymbol{w}_{k,i-1}} \\ &= \breve{\boldsymbol{u}}_{k,i}+\beta_k(i)\gamma_{k,i}\boldsymbol{D}_k(i-1). \end{aligned} \tag{21}$$

The rationality of this approximation is based on the slow time-variation lemma [67]: If the step size (learning rate) is small, the time variation of $\boldsymbol{w}_{k,i}$ is also slow. Such an approximation has been used advantageously in [63,64]. Similarly, we can obtain the derivative of the output $y_{k,i}$ with respect to the feedback weight as

$$\begin{aligned} S_k(i) &= \nabla_{\gamma_{k,i}}y_{k,i} \\ &= \frac{\partial y_{k,i}}{\partial \gamma_{k,i}}+\rho_k(i)\frac{\partial y_{k,i}}{\partial y_{k,i-1}}\frac{\partial y_{k,i-1}}{\partial \gamma_{k,i}} \\ &\approx \frac{\partial y_{k,i}}{\partial \gamma_{k,i}}+\rho_{k(i)}\frac{\partial y_{k,i}}{\partial y_{k,i-1}}\frac{\partial y_{k,i-1}}{\partial \gamma_{k,i-1}} \\ &= y_{k,i-1}+\rho_k(i)\gamma_{k,i}S_k(i-1) \end{aligned} \tag{22}$$

where $\rho_k(i)$ is the learning rate of $S_k(i)$, which is introduced to constrain the adaptation of the recurrent gradient information.

Substituting (20) and (22) into (10) and (11), respectively, the adaptation rules for $\boldsymbol{w}_{k,i}$ and $\gamma_{k,i}$ are obtained as

$$\begin{aligned} \boldsymbol{w}_{k,i+1} &= \boldsymbol{w}_{k,i}-\frac{\eta_k(i)}{\hat{\varrho}_k(i)}\nabla_{\boldsymbol{w}_{k,i}}J^{\mathrm{loc}}_k(\boldsymbol{w},\gamma) \\ &= \boldsymbol{w}_{k,i}+\frac{\eta_k(i)}{\hat{\varrho}_k(i)}e_{k,i}\frac{\partial(\vartheta_{k,i}(\breve{d}_{k,i}-y_{k,i}))}{\partial \boldsymbol{w}_{k,i}} \\ &\approx \boldsymbol{w}_{k,i}+\frac{\eta_k(i)}{\varrho_k(i)}e_{k,i}\nabla_{\boldsymbol{w}_{k,i}}y_{k,i} \\ &= \boldsymbol{w}_{k,i}+\frac{\eta_k(i)}{\varrho_k(i)}e_{k,i}\big[\breve{\boldsymbol{u}}_{k,i}+\beta_k(i)\gamma_{k,i}\boldsymbol{D}_k(i-1)\big], \end{aligned} \tag{23}$$

$$\begin{aligned} \gamma_{k,i+1} &= \gamma_{k,i}-\frac{\mu_k(i)}{\hat{\tau}_k(i)}\nabla_{\gamma_{k,i}}J^{\mathrm{loc}}_k(\boldsymbol{w},\gamma) \\ &= \gamma_{k,i}+\frac{\mu_k(i)}{\hat{\tau}_k(i)}e_{k,i}\frac{\partial(\vartheta_{k,i}(\breve{d}_{k,i}-y_{k,i}))}{\partial \gamma_{k,i}} \\ &\approx \gamma_{k,i}+\frac{\mu_k(i)}{\tau_k(i)}e_{k,i}\nabla_{\gamma_{k,i}}y_{k,i} \\ &= \gamma_{k,i}+\frac{\mu_k(i)}{\tau_k(i)}e_{k,i}\big[y_{k,i-1}+\rho_k(i)\gamma_{k,i}S_k(i-1)\big] \end{aligned} \tag{24}$$

where $\eta_k(i)$ and $\mu_k(i)$ are the step sizes, and $\hat{\varrho}_k(i)$, $\varrho_k(i)$, $\hat{\tau}_k(i)$ and $\tau_k(i)$ are normalization factors needed to prevent the so-called vanishing cone (or vanishing radius) problem with adaptive control systems [68]. In turn, proper adjustment of these parameters ensures convergence towards the optimal weights [62,69]. These parameters will be discussed in further details in Section 3.2. For ease of calculation, we assume that the derivatives of $\vartheta_{k,i}$ w.r.t. to $\boldsymbol{w}_{k,i}$ and $\gamma_{k,i}$ are nearly zero; and we also absorb the residual factor $\vartheta_{k,i}$ in the parameters $\varrho_k(i)$ and $\tau_k(i)$ in (23) and (24), respectively. Such assumptions are reasonable, since the range of permissible values for $\vartheta_{k,i}$ is small, i.e. this quantity can be roughly considered as a constant. For instance, under the simulation conditions of Example 2, its values fluctuates around 0.7.

To reduce the communication and computational requirements at each time instant, we here consider the diffusion strategy over the network. In [18], the adapt-then-combine (ATC) and combine-then-adapt (CTA) strategies were developed for distributed parameter estimation, where the ATC-based algorithm outperforms the CTA-based one. Motivated by these facts, we propose the ATC-based DTLS algorithm with multi-node feedback scheme. To this end, let us introduce intermediate estimates of the weight vector and the feedback weight, which we denote as $\boldsymbol{\varphi}_{k,i}$ and $\psi_{k,i}$ respectively. Under the ATC assumption with linear combination, the original steepest descent type of iterations in (23) and (24) are each transformed into two step sequence of operations. Specifically, in the first step, the intermediate estimates are updated as

$$\boldsymbol{\varphi}_{k,i+1}=\boldsymbol{\varphi}_{k,i}+\frac{\eta_k(i)}{\varrho_k(i)}\sum_{l\in\mathcal{N}_k}a_{l,k}e_{l,i}\boldsymbol{D}_l(i) \tag{25}$$

$$\psi_{k,i+1}=\psi_{k,i}+\frac{\mu_k(i)}{\tau_k(i)}\sum_{l\in\mathcal{N}_k}a_{l,k}e_{l,i}S_l(i). \tag{26}$$

Replacing the intermediate estimates $\boldsymbol{\varphi}_{k,i}$ and $\psi_{k,i}$ in (25) and (26) by the desired estimates from the previous steps, i.e. $\boldsymbol{w}_{k,i}$ and $\gamma_{k,i}$ respectively, we have

$$\boldsymbol{\varphi}_{k,i+1} = \boldsymbol{w}_{k,i} + \frac{\eta_k(i)}{\varrho_k(i)} \sum_{l\in\mathcal{N}_k} a_{l,k} e_{l,i} \boldsymbol{D}_l(i) \tag{27}$$

$$\psi_{k,i+1} = \gamma_{k,i} + \frac{\mu_k(i)}{\tau_k(i)} \sum_{l\in\mathcal{N}_k} a_{l,k} e_{l,i} S_l(i). \tag{28}$$

Such substitutions are reasonable since at a given time instant i, the weight parameters $\boldsymbol{w}_{k,i}$ and $\gamma_{k,i}$, contain more information than the corresponding intermediate estimates $\boldsymbol{\varphi}_{k,i}$ and $\psi_{k,i}$, and therefore can enhance the estimation performance of the proposed algorithm. Furthermore, these substitutions avoid the initialization of the new parameters [18]. Then, in the second step, the final weight vector and feedback weight are estimated through linear combinations as follows,

$$\boldsymbol{w}_{k,i+1} = \sum_{l\in\mathcal{N}_k} c_{l,k} \boldsymbol{\varphi}_{l,i+1} \tag{29}$$

$$\gamma_{k,i+1} = \sum_{l\in\mathcal{N}_k} c_{l,k} \psi_{l,i+1} \tag{30}$$

where $c_{l,k} \geq 0$ are non-negative real constants satisfying

$$c_{l,k} = 0 \text{ if } l \notin \mathcal{N}_k, \text{ and } \sum_{k=1}^{N} c_{l,k} = 1. \tag{31}$$

3.2. Robust adaptive parameter selection

Referring to (21)-(22) and (27)-(28), the normalization factors $\varrho_k(i)$ and $\tau_k(i)$, the step size of the weight vector $\eta_k(i)$, the step size of the feedback weight $\mu_k(i)$, the learning rate $\beta_k(i)$ of $\boldsymbol{D}_k(i)$, and the learning rate $\rho_k(i)$ of $S_k(i)$, play an important role in the proposed algorithm. Indeed, the DTLS algorithm with multi-node feedback strategy may not work reliably since it depends on several parameters that are not simple to tune in practice. To overcome this limitation, suitable parameter selection rules are developed in this subsection that are easy to implement and lead to good performance for distributed system identification under the EIV system model.

Selection of step sizes

To overcome the inherent compromise between fast convergence rate and small misadjustment, an adaptive approach to update the step size is warranted. Motivated by [63], the following adaptation rule is considered

$$\eta_k(i+1) = \begin{cases} \frac{3\left(1+\frac{1}{i}\right)^i}{2m} & \text{if } e_{k,i}^2 \geq \frac{g^2+\eta_k^2(i-1)q_{k,i-1}e_{k,i-1}^2}{\frac{1}{\varpi_{k,i}}-\eta_k(i)q_{k,i}} \\ \kappa & \text{otherwise} \end{cases} \tag{32}$$

where m is a positive constant, κ is a small positive constant (close to zero), $g = \iota$ is the threshold parameter, ι is a positive constant, $\varpi_{k,i} = \vartheta_{k,i}\nu_{k,i} - \varsigma_{k,i}$, and $q_{k,i} = \frac{\|\boldsymbol{D}_k(i)\|_2^2}{\varrho_k(i)}$ is the regularization factor. In a similar way, the adaptation of $\mu_k(i)$ can also be obtained by modifying (32) as

$$\mu_k(i+1) = \begin{cases} \frac{3\left(1+\frac{1}{i}\right)^i}{2m} & \text{if } e_{k,i}^2 \geq \frac{s^2+\mu_k^2(i-1)r_{k,i-1}e_{k,i-1}^2}{\frac{1}{\vartheta_{k,i}}-\mu_k(i)r_{k,i}} \\ \kappa & \text{otherwise} \end{cases} \tag{33}$$

where $r_{k,i} = \frac{|S_k^2(i)|}{\tau_k(i)}$ is the regularization factor, and $s = g$ is the threshold parameter.

Remark 1. The adaptation rules in (32) and (33) are based on the main concept that large error values require a large step size κ, while small error values require a small step to obtain a small misadjustment. Specifically, for large values of the error $e_{k,i}$, the choice $3(1+1/i)^i/2m$ results in a large value of the step size during the initial stage of the algorithm. The two thresholds parameters s and g allow to achieve the desired trade-off between the convergence rate and estimation accuracy. Large values of these parameters lead to fast convergence speed and large misadjustment. In contrast, small values provide a very good estimation accuracy (in a stationary environment) but slow convergence rate.

Selections of normalization factors

The normalization factors in the proposed algorithm are designed based on the methods introduced in [62,63]. Specifically, they are adjusted at each iteration according to

$$\varrho_k(i) = \max\left\{\varrho_k(i-1),\ \upsilon\varrho_k(i-1) + \max\left(\bar{\rho},\ \|\boldsymbol{D}_k(i)\|_2^2\right)\right\} \tag{34}$$

$$\tau_k(i) = \max\left\{\tau_k(i-1),\ \lambda\tau_k(i-1) + \max\left(\bar{\tau},\ S_k^2(i)\right)\right\} \tag{35}$$

where υ and λ are the forgetting factors, and $\bar{\rho}$ and $\bar{\tau}$ are two positive constants.

Selections of learning rates

The hybrid learning rate of the proposed algorithm is used to control the feedback information, which can be computed as follow:

$$\beta_k(i) = \begin{cases} \mu_\beta \operatorname{sign}\left\{z_k^D(i)\right\} & \text{if } |e_{k,i}| < \frac{|e_{k,i-1}\eta_k(i-1)|}{\hbar+|\gamma_{k,i}|} \\ 0 & \text{otherwise} \end{cases} \tag{36}$$

where $\mu_\beta > 0$ is a scalar, $z_k^D(i) = \eta_k(i-1)\gamma_{k,i}e_{k,i}e_{k,i-1}$, $\hbar$ is the small positive constant, and $\operatorname{sign}\{\cdot\}$ is the sign function. Likewise, $\rho_k(i)$ can be computed as follow:

$$\rho_k(i) = \begin{cases} \mu_\rho \operatorname{sign}\left\{z_k^S(i)\right\} & \text{if } |e_{k,i}| < \frac{|e_{k,i-1}\mu_k(i-1)|}{\hbar+|\gamma_{k,i}|} \\ 0 & \text{otherwise} \end{cases} \tag{37}$$

where $\mu_\rho > 0$ is a scalar and $z_k^S(i) = \mu_k(i-1)\gamma_{k,i}e_{k,i}e_{k,i-1}$. The main idea for designing the learning parameters is to allow the latter to switch between positive and negative values according to the instantaneous error, and use a scalar parameter (amplification parameter, μ_β and μ_ρ) to further adjust adaptation.

3.3. Choice of weighting factor

The choice of the weighting factors $\varepsilon_{l,k}$ in the proposed algorithm is application-specific. For the sake of simplicity, in this work, we set $\boldsymbol{\Pi}_k = \mathbf{I}$ in (7) [50,53]. If prior knowledge of the input noise variance $\sigma_{u,k}^2$ and the output noise variance $\sigma_{d,k}^2$ can be obtained, $\varepsilon_{0,k}$ can be calculated as $\varepsilon_{0,k} = \frac{\sigma_{u,k}}{\sigma_{d,k}}$ [50,53].

Consider a more realistic situation where the input noise variance $\sigma_{u,k}^2$ and the output noise variance $\sigma_{d,k}^2$ are different for each node k. Moreover, the exact values of noise variances are unknown and $\sigma_{u,k}^2$ is assumed to be comparable to $\sigma_{d,k}^2$. In this case, we can roughly set

$$\varepsilon_{0,k} = 1 \tag{38}$$

to avoid the effect brought by the inequality of the input and output noise variances [53].

The proposed DTLS algorithm with multi-node feedback for distributed linear system identification is summarized in Table 1.

4. Comparisons of computational complexity

In this section, we compare the computational complexity of the proposed algorithm with that of the DLMS and D-GDTLS

Table 1
Summary of the algorithm.

Algorithm 1: Diffusion TLS algorithm with multi-node feedback
Initialization:
Initialize $\boldsymbol{w}_{k,0}$ for each node k
Chose non-negative real coefficients $\{a_{l,k}, c_{l,k}\}$
Initialize ι, m, κ, υ, λ, $\bar{\rho}$, $\bar{\tau}$, μ_β and μ_ρ
Adaptation:
1) Compute the output signal at node k:
$y_{k,i} = \tilde{\boldsymbol{u}}_{k,i}^{\mathrm{T}}\boldsymbol{w}_{k,i} + \gamma_{k,i}y_{k,i-1}$
2) Compute the error signal at node k:
$e_{k,i} = \frac{\bar{d}_{k,i}-y_{k,i}}{\sqrt{\boldsymbol{w}_{k,i}^{\mathrm{T}}\boldsymbol{\Pi}_i^{-2}\boldsymbol{w}_{k,i}+\varepsilon_{0,i}^{-2}}}$
3) Compute the hybrid derivatives $\boldsymbol{D}_k(i)$ and $S_k(i)$:
Use (32), (34) and (36) to adapt $\eta_k(i)$, $\varrho_k(i)$, and $\beta_k(i)$
$\boldsymbol{D}_k(i) = \tilde{\boldsymbol{u}}_{k,i} + \beta_k(i)\gamma_{k,i}\boldsymbol{D}_k(i-1)$
Use (33), (35) and (37) to adapt $\mu_k(i)$, $\tau_k(i)$, and $\rho_k(i)$
$S_k(i) = y_{k,i-1} + \rho_k(i)\gamma_{k,i}S_k(i-1)$
4) Update the intermediate weight vector and feedback weight:
$\boldsymbol{\varphi}_{k,i+1} = \boldsymbol{w}_{k,i} + \frac{\eta_k(i)}{\varrho_k(i)}\sum_{l\in\mathcal{N}_k} a_{l,k}e_{l,i}\boldsymbol{D}_l(i)$
$\psi_{k,i+1} = \gamma_{k,i} + \frac{\mu_k(i)}{\tau_k(i)}\sum_{l\in\mathcal{N}_k} a_{l,k}e_{l,i}S_l(i)$
Combination:
$\boldsymbol{w}_{k,i+1} = \sum_{l\in\mathcal{N}_k} c_{l,k}\boldsymbol{\varphi}_{l,i+1}$
$\gamma_{k,i+1} = \sum_{l\in\mathcal{N}_k} c_{l,k}\psi_{l,i+1}$

Table 2
Summary of the computational complexity.

Operation	DLMS	D-GDTLS	Proposed
Additions	$3N_kM - M$	$4N_kM + N_k - M$	$4N_kM + 3N_k - M + 1$
Multiplications	$3N_kM + M + 2N_k$	$5N_kM + M + 2N_k$	$5N_kM + M + 9N_k + 2$

algorithms for distributed system identification. Table 2 shows the total number of additions and multiplications per iteration for the conventional DLMS, D-GDTLS and the proposed algorithm at node k, where N_k denotes the number of nodes in the neighborhood set $\mathcal{N}_k$. Compared with the DLMS and D-GDTLS algorithms, the proposed algorithm requires additional computations for calculating the feedback parameters, which is the price paid for its superior performance, as investigated in the next Section.

5. Convergence analysis

In this section, the convergence analysis of the proposed DTLS algorithm with feedback is presented. Since the global stability of the proposed algorithm with observation exchange is difficult to analyze exactly, here to simplify the presentation, we focus on the stability performance of the weight vector and feedback weight at individual nodes. Previous studies have demonstrated that the stability over each node can under certain conditions guarantee the global stability of the algorithm [70]. Below, we show that the DTLS algorithm with multi-node feedback can ensure the stability and convergence of the distributed in-network system identification at each node. First, the following Lemma is introduced.

Lemma 1. *Define a Lyapunov function at node k as $\mathcal{V}_k(i)$. Let $\alpha_k(i)$ and $\epsilon_k(i)$ denote time series satisfying $\alpha_k(i) \geq 0$ and $\epsilon_k(i) \leq 0$. If $\mathcal{V}_k(i)$ is initially set to $\mathcal{V}_k(0) = 0$ and the following condition hold:*

$$\mathcal{V}_k(i) = \alpha_k(i)\mathcal{V}_k(i-1) + \epsilon_k(i) \tag{39}$$

then, we have $\mathcal{V}_k(i) \leq 0$, $\forall i, k$.

Proof. Considering initial condition, we have $\mathcal{V}_k(1) = \alpha_k(1)\mathcal{V}_k(0) + \epsilon_k(1) \leq 0$. Thus, for time i, we have

$$\mathcal{V}_k(i+1) = \alpha_k(i+1)\mathcal{V}_k(i) + \epsilon_k(i+1) \leq 0. \tag{40}$$

According to the LST, the asymptotic stability can be guaranteed if Lyapunov function $\mathcal{V}_k(i)$ for the problem at hand is negative [71,72]. The above Lemma has been used as a tool in analyzing active noise control algorithms [73], kernel adaptive algorithms [62,65], and neural networks [74,75]. Here, we use this Lemma to analyze the convergence behavior of the proposed algorithm. □

Theorem 1. *The weight error vector $\tilde{\boldsymbol{w}}_{k,i}$ produced by the DTLS algorithm with adaptive parameters determined by (32), (34) and (36) is guaranteed to be convergent in the sense of Lyapunov, that is,*

$$\|\tilde{\boldsymbol{w}}_{k,i+1}\|_2^2 \leq \|\tilde{\boldsymbol{w}}_{k,i}\|_2^2 \ \ \forall i, k.$$

Proof. Subtracting $\boldsymbol{w}^o$ from both sides of (23), and using (20), we have

$$\begin{aligned}\tilde{\boldsymbol{w}}_{k,i+1} &\triangleq \boldsymbol{w}_{k,i+1} - \boldsymbol{w}^o \\ &= \tilde{\boldsymbol{w}}_{k,i} + \frac{\eta_k(i)}{\varrho_k(i)}e_{k,i}\boldsymbol{D}_k(i).\end{aligned} \tag{41}$$

Taking the squared norm on both sides of (41), we obtain

$$\begin{aligned}\Delta\|\tilde{\boldsymbol{w}}_{k,i}\|_2^2 &\triangleq \|\tilde{\boldsymbol{w}}_{k,i+1}\|_2^2 - \|\tilde{\boldsymbol{w}}_{k,i}\|_2^2 \\ &= \left[\frac{\eta_k(i)}{\varrho_k(i)}e_{k,i}\right]^2\|\boldsymbol{D}_k(i)\|_2^2 + 2\frac{\eta_k(i)}{\varrho_k(i)}e_{k,i}\boldsymbol{D}_k(i)\tilde{\boldsymbol{w}}_{k,i}^{\mathrm{T}} \\ &= \left[\frac{\eta_k(i)}{\varrho_k(i)}e_{k,i}\right]^2\|\boldsymbol{D}_k(i)\|_2^2 + 2\frac{\eta_k(i)}{\varrho_k(i)}e_{k,i}\tilde{\boldsymbol{w}}_{k,i}^{\mathrm{T}} \\ &\quad \cdot\{\tilde{\boldsymbol{u}}_{k,i} + \beta_k(i)\gamma_{k,i}\boldsymbol{D}_k(i-1)\} \\ &= \frac{\eta_k(i)}{\varrho_k(i)}\left[P_{1,k}(i) + P_{2,k}(i) + P_{3,k}(i)\right]\end{aligned} \tag{42}$$

where we define

$$P_{1,k}(i) \triangleq \frac{\eta_k(i)}{\varrho_k(i)}e_{k,i}^2\|\boldsymbol{D}_k(i)\|_2^2 \tag{43}$$

$$\begin{aligned}P_{2,k}(i) &\triangleq 2e_{k,i}\tilde{\boldsymbol{w}}_{k,i}^{\mathrm{T}}\tilde{\boldsymbol{u}}_{k,i} \\ &= 2e_{k,i}\left(\upsilon_{k,i} - \frac{\Delta_{k,i}^w}{\vartheta_{k,i}}\right) \\ &= 2e_{k,i}\left(\frac{\vartheta_{k,i}\upsilon_{k,i} - \varsigma_{k,i} - \Delta_{k,i}^w + \varsigma_{k,i}}{\vartheta_{k,i}}\right) \\ &= 2e_{k,i}\left(\frac{\varpi_{k,i} - e_{k,i}}{\vartheta_{k,i}}\right)\end{aligned} \tag{44}$$

$$P_{3,k}(i) \triangleq 2e_{k,i}\beta_k(i)\gamma_{k,i}\tilde{\boldsymbol{w}}_{k,i}^{\mathrm{T}}\boldsymbol{D}_k(i-1). \tag{45}$$

Using (41), (45) can be rewritten as

$$\begin{aligned}P_{3,k}(i) &= 2e_{k,i}\beta_k(i)\gamma_{k,i}\left[\tilde{\boldsymbol{w}}_{k,i-1} + \frac{\eta_k(i-1)}{\varrho_k(i-1)}e_{k,i-1}\boldsymbol{D}_k(i-1)\right]^{\mathrm{T}} \\ &\quad \cdot\boldsymbol{D}_k(i-1).\end{aligned} \tag{46}$$

Define $\mathcal{G}_k(i) > 0$ based on (36) as follow:

$$\mathcal{G}_k(i) = \begin{cases}\frac{|\gamma_{k,i}e_{k,i}\beta_k(i)|}{|e_{k,i-1}\eta_k(i-1)|} & \text{if } |e_{k,i}| < \frac{|e_{k,i-1}\eta_k(i-1)|}{\hbar+|\gamma_{k,i}|} \\ 0 & \text{otherwise.}\end{cases} \tag{47}$$

Thus, we have

$$
\begin{aligned}
P_{3,k}(i) &= \mathcal{G}_k(i)\varrho_k(i-1)\left[2\frac{\eta_k(i-1)}{\varrho_k(i-1)}e_{k,i-1}\tilde{\boldsymbol{w}}_{k,i-1}\right.\\
&\quad\left.+2\left(\frac{\eta_k(i-1)}{\varrho_k(i-1)}\right)^2 e_{k,i-1}^2\boldsymbol{D}_k(i-1)\right]^{\mathrm{T}}\boldsymbol{D}_k(i-1)\\
&= \mathcal{G}_k(i)\frac{\eta_k^2(i-1)\|\boldsymbol{D}_k(i-1)\|_2^2}{\varrho_k(i-1)}e_{k,i-1}^2\\
&\quad+\mathcal{G}_k(i)\varrho_k(i-1)\Delta\|\tilde{\boldsymbol{w}}_{k,i-1}\|_2^2.
\end{aligned}
\tag{48}
$$

Substituting (43), (44), and (48) into (42), we can obtain the following expression for $\Delta\|\tilde{\boldsymbol{w}}_{k,i}\|_2^2$:

$$
\begin{aligned}
\Delta\|\tilde{\boldsymbol{w}}_{k,i}\|_2^2 &= \frac{\eta_k(i)}{\varrho_k(i)}\{P_{1,k}(i)+P_{2,k}(i)+P_{3,k}(i)\}\\
&= P_{4,k}(i)+\frac{\eta_k(i)\mathcal{G}_k(i)\varrho_k(i-1)}{\varrho_k(i)}\Delta\|\tilde{\boldsymbol{w}}_{k,i-1}\|_2^2
\end{aligned}
\tag{49}
$$

where

$$
\begin{aligned}
P_{4,k}(i) &= \frac{\eta_k(i)}{\varrho_k(i)}\left(\frac{\eta_k(i)}{\varrho_k(i)}e_{k,i}^2\|\boldsymbol{D}_k(i)\|_2^2\right.\\
&\quad+2e_{k,i}\left(\frac{\varpi_{k,i}-e_{k,i}}{\vartheta_{k,i}}\right)\\
&\quad\left.+\mathcal{G}_k(i)\frac{\eta_k^2(i-1)\|\boldsymbol{D}_k(i-1)\|_2^2}{\varrho_k(i-1)}e_{k,i-1}^2\right).
\end{aligned}
\tag{50}
$$

Then, we have

$$
\begin{aligned}
P_{4,k}(i) &\le \frac{\eta_k(i)}{\varrho_k(i)}\left[\frac{\varpi_{k,i}^2}{\vartheta_{k,i}}-\left(\frac{1}{\vartheta_{k,i}}-\frac{\eta_k(i)\|\boldsymbol{D}_k(i)\|_2^2}{\varrho_k(i)}\right)e_{k,i}^2\right.\\
&\quad\left.+\mathcal{G}_k(i)\frac{\eta_k^2(i-1)\|\boldsymbol{D}_k(i-1)\|_2^2}{\varrho_k(i-1)}e_{k,i-1}^2\right]\\
&\le \frac{\eta_k(i)}{\varrho_k(i)}\left[\zeta_m^2-\left(\frac{1}{\vartheta_{k,i}}-\frac{\eta_k(i)\|\boldsymbol{D}_k(i)\|_2^2}{\varrho_k(i)}\right)e_{k,i}^2\right.\\
&\quad\left.+\frac{\eta_k^2(i-1)\|\boldsymbol{D}_k(i-1)\|_2^2}{\varrho_k(i-1)}e_{k,i-1}^2\right]
\end{aligned}
\tag{51}
$$

where the first inequality is due to the fact that $2e_{k,i}\varpi_{k,i}\le e_{k,i}^2+\varpi_{k,i}^2$, and the second inequality results from the definition of $\eta_k(i+1)$ in (32). Invoking the condition in (32) and letting $g=\max\limits_k\left\{\frac{\varpi_{k,i}}{\sqrt{\vartheta_{k,i}}}\right\}$, it can be shown that

$$
P_{4,k}(i)\le 0. \tag{52}
$$

Moreover, using (47), it can be shown that $\frac{\eta_k(i)\mathcal{G}_k(i)\varrho_k(i-1)}{\varrho_k(i)}\ge 0$. According to Lemma, with the identifications

$$
\epsilon_k(i)=P_{4,k}(i),\quad \alpha_k(i)=\frac{\eta_k(i)\mathcal{G}_k(i)\varrho_k(i-1)}{\varrho_k(i)}, \tag{53}
$$

we can conclude that $\Delta\|\tilde{\boldsymbol{w}}_{k,i}\|_2^2$ is a Lyapunov function, that is

$$
\Delta\|\tilde{\boldsymbol{w}}_{k,i}\|_2^2\le 0. \tag{54}
$$

In the above discussions, we only carried on the convergence analysis of the weight vector of the proposed algorithm, while the feedback convergence behavior was not involved. Next, we present the convergence analysis of the feedback weight. □

Theorem 2. *The feedback weight error $\tilde{\gamma}_{k,i}$ produced by the proposed algorithm with adaptive parameters determined by (33), (35) and (37) is guaranteed to be convergent in the sense of Lyapunov, that is,*

$$
\tilde{\gamma}_{k,i+1}^2\le\tilde{\gamma}_{k,i}^2\ \ \forall i,k.
$$

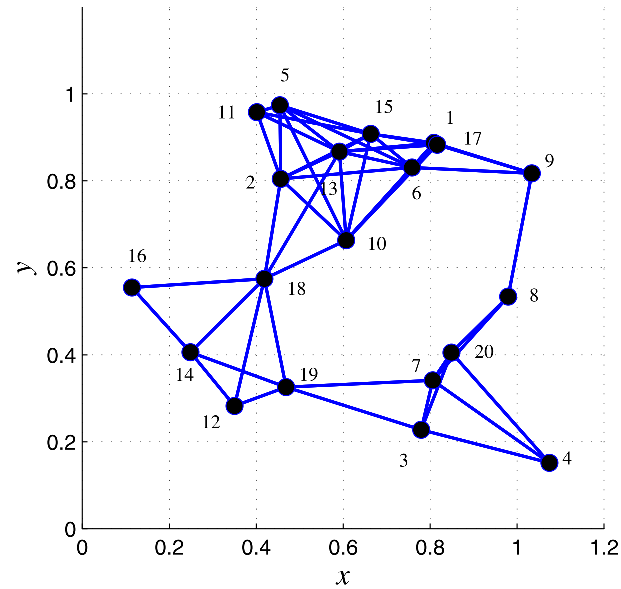

Fig. 3. The topology of the simulated diffusion network consisting of 20 nodes.

Proof. Subtracting $\gamma_{k,i}^o$ from both sides of (24), we get

$$
\begin{aligned}
\tilde{\gamma}_{k,i+1} &\triangleq \gamma_{k,i}-\gamma_{k,i}^o\\
&= \tilde{\gamma}_{k,i}+\frac{\mu_k(i)}{\tau_k(i)}e_{k,i}S_k(i).
\end{aligned}
\tag{55}
$$

Taking the squared norm on both sides of (55) and rearranging terms, we have

$$
\begin{aligned}
\Delta\tilde{\gamma}_{k,i}^2 &\triangleq \tilde{\gamma}_{k,i+1}^2-\tilde{\gamma}_{k,i}^2\\
&= \left(\frac{\mu_k(i)}{\tau_k(i)}e_{k,i}\right)^2S_k^2(i)+2\frac{\mu_k(i)}{\tau_k(i)}e_{k,i}\tilde{\gamma}_{k,i}\\
&\quad\cdot\{y_{k,i-1}+\rho_k(i)\gamma_{k,i}S_k(i-1)\}\\
&= \frac{\mu_k(i)}{\tau_k(i)}\left[Q_{1,k}(i)+Q_{2,k}(i)+Q_{3,k}(i)\right]
\end{aligned}
\tag{56}
$$

where we define

$$
Q_{1,k}(i)\triangleq\frac{\mu_k(i)S_k^2(i)}{\tau_k(i)}e_{k,i}^2 \tag{57}
$$

$$
\begin{aligned}
Q_{2,k}(i) &\triangleq 2e_{k,i}\tilde{\gamma}_{k,i}y_{k,i-1}\\
&= 2e_{k,i}\left(\frac{\varsigma_{k,i}}{\vartheta_{k,i}}\right)\\
&= 2e_{k,i}\left(\frac{\Delta_{k,i}^w-e_{k,i}}{\vartheta_{k,i}}\right)
\end{aligned}
\tag{58}
$$

$$
\begin{aligned}
Q_{3,k}(i) &\triangleq 2e_{k,i}\tilde{\gamma}_{k,i}\rho_k(i)\gamma_{k,i}S_k(i-1)\\
&= 2e_{k,i}\rho_k(i)\gamma_{k,i}\\
&\quad\cdot\left\{\tilde{\gamma}_{k,i-1}+\frac{\mu_k(i-1)}{\tau_k(i-1)}e_{k,i-1}S_k(i-1)\right\}S_k(i-1)\\
&= \mathcal{H}_k(i)\tau_k(i-1)\left[2\frac{\mu_k(i-1)}{\tau_k(i-1)}e_{k,i-1}\tilde{\gamma}_{k,i-1}\right.
\end{aligned}
$$

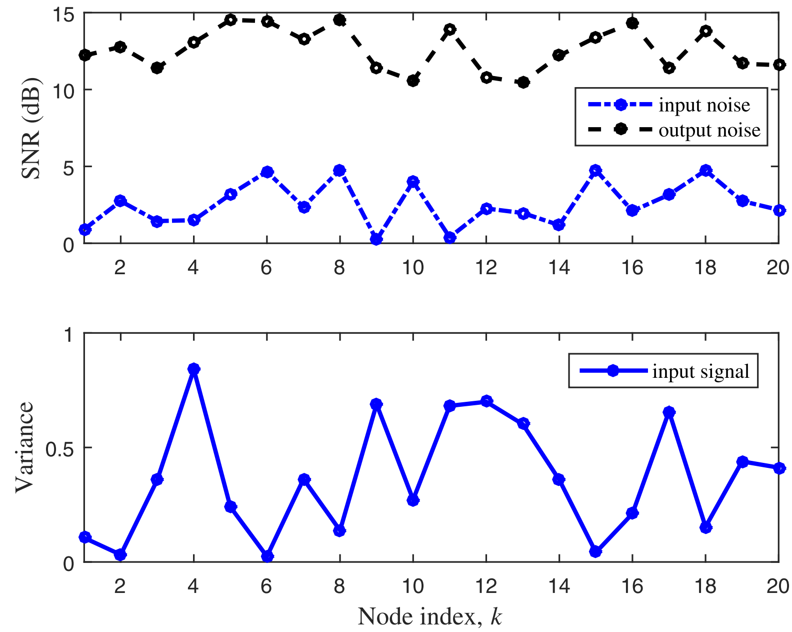

Fig. 4. The SNR of the input noise and output noise of all nodes in the diffusion network (top). The variance of the input signal (bottom).

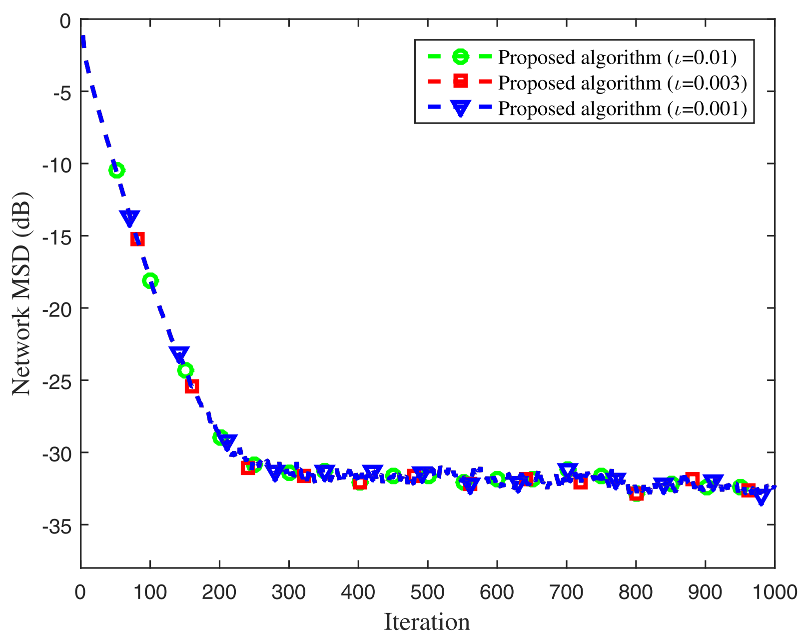

Fig. 5. Learning curves for the proposed algorithm for different ι in example 1.

$$
\begin{aligned}
&+2\left(\frac{\mu_k(i-1)}{\tau_k(i-1)}\right)^2 e_{k,i-1}^2 S_k(i-1)\Bigg]S_k(i-1)\\
&=\mathcal{H}_k(i)\frac{\mu_k^2(i-1)S_k^2(i-1)}{\tau_k(i-1)}e_{k,i-1}^2\\
&+\mathcal{H}_k(i)\tau_k(i-1)\Delta\tilde{\gamma}_{k,i-1}^2
\end{aligned}
\tag{59}
$$

where

$$
\mathcal{H}_k(i)=\begin{cases}\frac{|\gamma_{k,i}e_{k,i}\rho_k(i)|}{|e_{k,i-1}\mu_k(i-1)|} & \text{if } |e_{k,i}|<\frac{|e_{k,i-1}\mu_k(i-1)|}{\hbar+|\gamma_{k,i}|}\\ 0 & \text{otherwise.}\end{cases}
\tag{60}
$$

Using the definition of $\mathcal{H}_k(i)$, (56) is simplified to

$$
\begin{aligned}
\Delta\tilde{\gamma}_{k,i}^2=\frac{\mu_k(i)}{\tau_k(i)}\Bigg[&\frac{\mu_k(i)S_k^2(i)}{\tau_k(i)}e_{k,i}^2+2e_{k,i}\left(\frac{\Delta_{k,i}^w-e_{k,i}}{\vartheta_{k,i}}\right)\\
&+\mathcal{H}_k(i)\frac{\mu_k^2(i-1)S_k^2(i-1)}{\tau_k(i-1)}e_{k,i-1}^2\\
&+\mathcal{H}_k(i)\tau_k(i-1)\Delta\tilde{\gamma}_{k,i-1}^2\Bigg]\\
=\;&Q_{4,k}(i)+\frac{\mu_k(i)\mathcal{H}_k(i)\tau_k(i-1)}{\tau_k(i)}\Delta\tilde{\gamma}_{k,i-1}^2
\end{aligned}
\tag{61}
$$

where

$$
\begin{aligned}
Q_{4,k}(i)=\frac{\mu_k(i)}{\tau_k(i)}\Bigg[&\frac{\mu_k(i)S_k^2(i)}{\tau_k(i)}e_{k,i}^2+2e_{k,i}\left(\frac{\Delta_{k,i}^w-e_{k,i}}{\vartheta_{k,i}}\right)\\
&+\mathcal{H}_k(i)\frac{\mu_k^2(i-1)S_k^2(i-1)}{\tau_k(i-1)}e_{k,i-1}^2\Bigg].
\end{aligned}
\tag{62}
$$

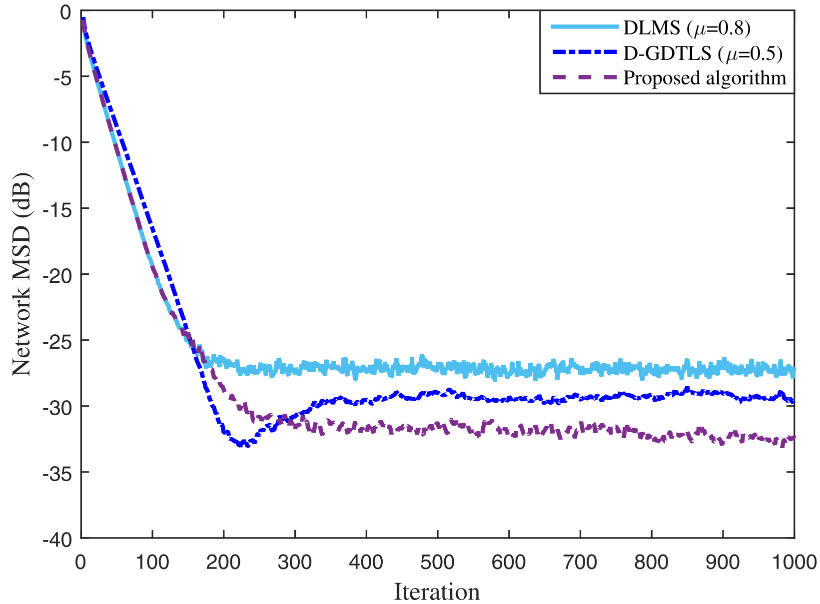

Fig. 6. Network MSD for different diffusion algorithms.

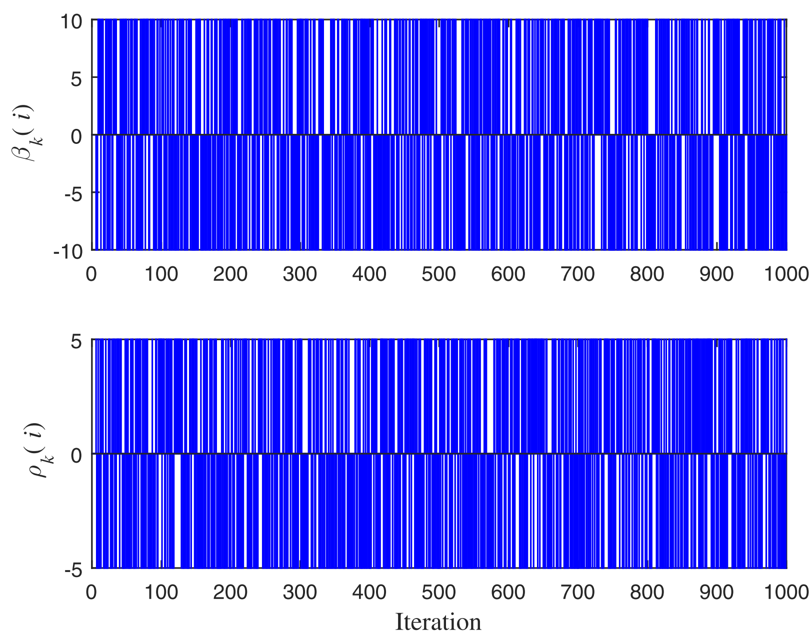

Fig. 7. Evolutions of $\beta_k(i)$ and $\rho_k(i)$ at node 7.

Therefore, $Q_{4,\,k}(i)$ can be expressed as

$$\begin{aligned} Q_{4,k}(i) \leq & \frac{\mu_k(i)}{\tau_k(i)}\left\{\frac{|\Delta^w_{k,i}|^2}{\vartheta_{k,i}} - \left(\frac{1}{\vartheta_{k,i}} - \frac{\mu_k(i)S_k^2(i)}{\tau_k(i)}\right)e^2_{k,i}\right. \\ & \left. + \mathcal{H}_k(i)\frac{\mu_k^2(i-1)S_k^2(i-1)}{\tau_k(i-1)}e^2_{k,i-1}\right\} \\ \leq & \frac{\mu_k(i)}{\tau_k(i)}\left\{\phi_m^2 - \left(\frac{1}{\vartheta_{k,i}} - \frac{\mu_k(i)S_k^2(i)}{\tau_k(i)}\right)e^2_{k,i}\right. \\ & \left. + \frac{\mu_k^2(i-1)S_k^2(i-1)}{\tau_k(i-1)}e^2_{k,i-1}\right\} \end{aligned} \tag{63}$$

where the first inequality in (63) follows from the fact that $2e_{k,i}\Delta^w_{k,i} \leq e^2_{k,i} + |\Delta^w_{k,i}|^2$. According to the definition of $\mu_k(i+1)$ in (33) and letting $s = \max_k \left\{\frac{|\Delta^w_{k,i}|^2}{\vartheta_{k,i}}\right\}$, the second inequality (63) can be obtained.

Recalling (33), we have

$$Q_{4,k}(i) \leq 0. \tag{64}$$

Similarly, according to Lemma, we define

$$\epsilon_k(i) = Q_{4,k}(i), \quad \alpha_k(i) = \frac{\mu_k(i)\mathcal{H}_k(i)\tau_k(i-1)}{\tau_k(i)}. \tag{65}$$

From (64) to (65), we can conclude that $\Delta\tilde{\gamma}^2_{k,i}$ is a Lyapunov function and the feedback weight is asymptotically stable in the sense of Lyapunov, that is

$$\Delta\gamma^2_{k,i} \leq 0. \tag{66}$$

This completes the proof. □

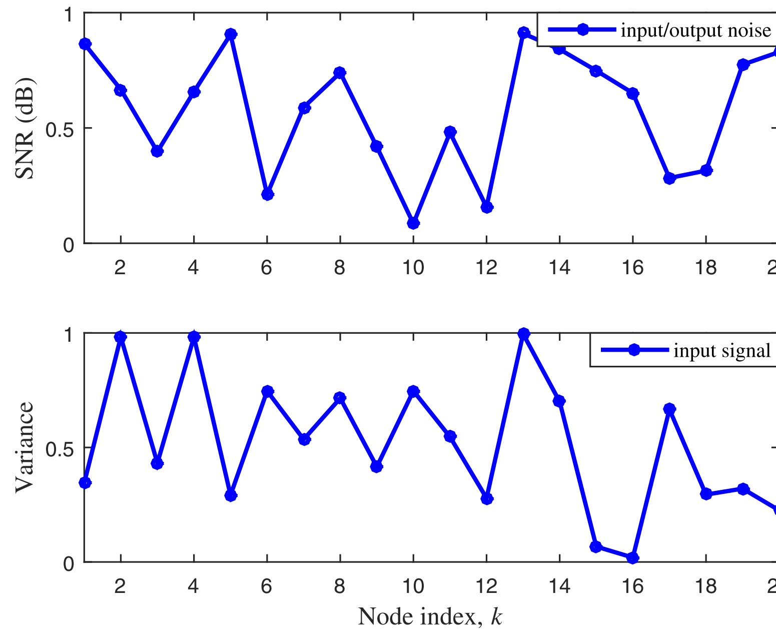

Fig. 8. The SNR of the input noise and output noise of all nodes in the diffusion network (top). The variance of the input signal (bottom).

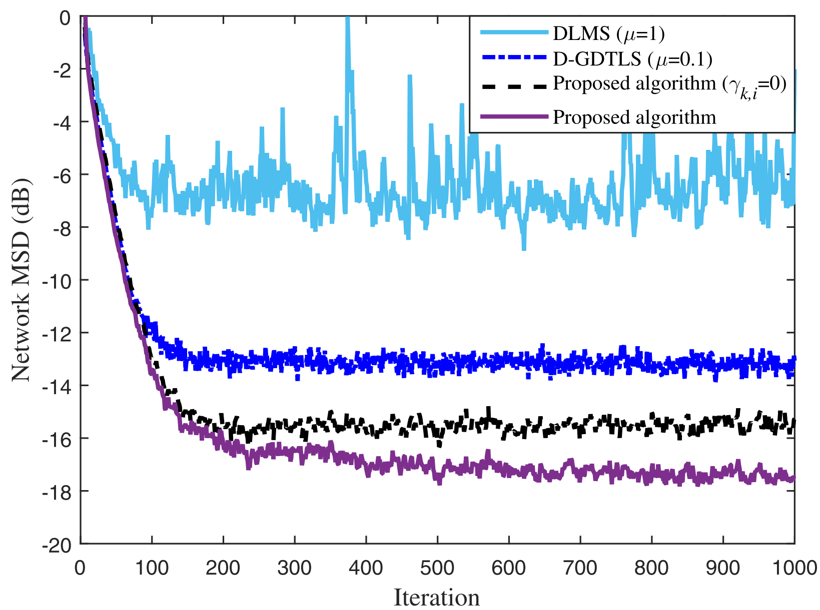

Fig. 9. Network MSD for different diffusion algorithms in low SNR environment.

6. Simulation results

We use simulations to illustrate the effectiveness of the proposed DTLS algorithm with feedback for distributed identification of EIV systems, as per the system model introduced in Section 2, and compare it with the conventional DLMS and D-GDTLS algorithms.[3] The diffusion network under study is composed of $N = 20$ nodes, as shown in Fig. 3. The linear combination coefficients $\{a_{l,\,k}\}$ in (27) and (28) are based on the Metropolis rule [18],

$$\begin{cases} a_{l,k} = \frac{1}{\max\,(h_k, h_l)}, & l \in \mathcal{N}_k, l \neq k \\ a_{l,k} = 0, & l \notin \mathcal{N}_k \\ a_{l,k} = 1 - \sum\limits_{l \in \mathcal{N}_k \backslash k} \{a_{l,k}\}, & l = k \end{cases} \tag{67}$$

where h_k and h_l denote the degrees of nodes k and l,[4] and $l \in \mathcal{N}_k \backslash k$ stands for the neighbors of node k except itself. The linear combination coefficients $\{c_{l,\,k}\}$ are defined in the same way, i.e., $c_{l,k} = a_{l,k}$. To quantify the estimation performance, we use the following network-averaged mean-square deviation (MSD) [18]

$$\text{Network MSD} = \frac{1}{N} \sum_{k=1}^{N} \Delta \|\tilde{\boldsymbol{w}}_{k,i}\|_2^2. \tag{68}$$

[3] The distributed TLS algorithm [55,56] is characterized by a heavy computation load, while the D-GDTLS algorithm outperforms the BC-DLMS algorithm [57]; hence we decided to compare our approach to the DLMS and D-GDTLS algorithms in this section.

[4] The degree of node k denotes the number of its neighbors [18,76].

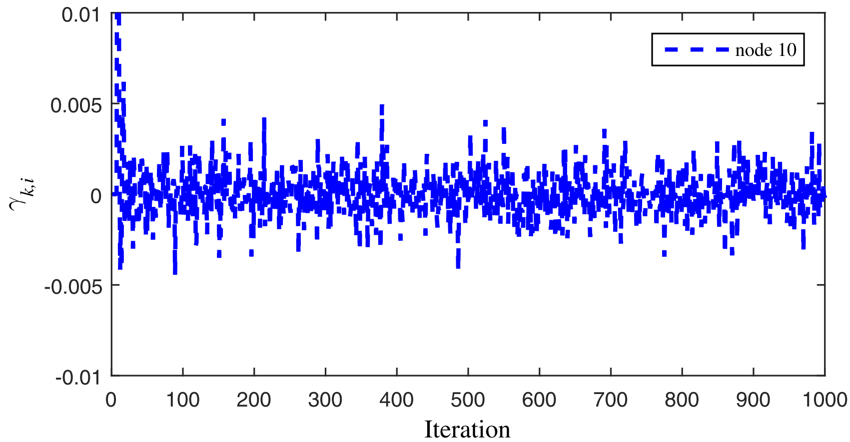

Fig. 10. Learning curve of $\gamma_{k,\,i}$ of node 10.

The parameters of the three algorithms under study are selected to guarantee rapid convergence and stability. The results presented are averaged over 100 independent trials.

Example 1. In this example, the entries of the input vector $\boldsymbol{u}_{k,\,i}$, those of the input noise signal $\boldsymbol{n}_{k,\,i}$ and the additive noise signal $v_{k,\,i}$ are modeled as independent white Gaussian noise (WGN) sequences with zero-mean. Fig. 4 shows the signal-to-noise ratio (SNR) of the input and output noises, together with the variance of the input signal. The unknown parameter vector $\boldsymbol{w}^o$ is generated randomly with $M = 2$ [41,57]. We set $m = 0.3$, $\kappa = 0.01$, $\upsilon = 0.75$, $\lambda = 0.95$, $\bar{\rho} = 1$, $\hbar = 0.001$, $\bar{\tau} = 10$, $\mu_\beta = 10$ and $\mu_\rho = 5$ for the proposed algorithm. Fig. 5 shows the learning curves of the network MSD of the proposed algorithm for different choices of the parameter ι in (32) and (33). As can be seen, the proposed algorithm is not sensitive to this choice. In the following simulations, we set $\iota = 0.003$. Fig. 6 shows a performance comparison of the DLMS algorithm, the D-GDTLS algorithm and the proposed algorithm. Fig. 7 depicts the evolutions of $\beta_k(i)$ and $\rho_k(i)$ for the proposed algorithm. It can be observed that all the algorithms reach the steady state after almost 300 iterations. The D-GDTLS algorithm has relatively large steady-state error during adaptation. In contrast, the proposed algorithm achieves the smallest steady-state error as compared with the existing algorithms with the similar initial convergence rate. Owing to using the multi-node feedback strategy, the proposed algorithm reaches a lower network MSD than the D-GDTLS algorithm.

Example 2. In the second example, we study the performance of the algorithms in a relatively high input and output noise environment. The unknown parameter vector $\boldsymbol{w}^o$ has $M = 5$ entries and is selected randomly. The input and noise signals used in this example have similar characteristics as in Example 1. We use the feedback scheme with parameters $m = 0.3$, $\kappa = 0.01$, $\upsilon = 0.15$, $\lambda = 0.95$, $\bar{\rho} = 1$, $\bar{\tau} = 10$, $\hbar = 0.001$, $\iota = 0.003$, $\mu_\beta = 20$ and $\mu_\rho = 1$. The SNR and the input variance are shown in Fig. 8. Fig. 9 compares the network MSDs of the DLMS, D-GDTLS and the proposed algorithm. We observe that the proposed algorithm with multi-node feedback outperforms the proposed algorithm with $\gamma_{k,i} = 0$, and the proposed algorithm performs better than the DLMS and D-GDTLS algorithms, achieving a network MSD of around -18 dB in the steady state. To further demonstrate the performance of the proposed algorithm, Fig. 10 shows the learning curve of $\gamma_{k,\,i}$. As shown in the figure, $\gamma_{k,\,i}$ reaches large values during initial convergence, but remains relative small and stable in steady state.

7. Conclusion

In this paper, we proposed and investigated a new DTLS algorithm based on multi-node feedback scheme, to address the problem of in-network distributed system identification in EIV model. Compared with the conventional DTLS algorithm, the proposed algorithm additionally considered using the output information of each node to adapt the weight vector for performance improvement. Besides, the convergence analyses of the weight vector and feedback weight have been performed according to LST. We also developed robust adaptive strategies to update the various parameters entering the proposed algorithm that are easy to implement and enjoy excellent performance. Simulations under the diffusion EIV model have shown that the new algorithm outperforms the DLMS and D-GDTLS algorithms.

The price paid for the better performance of the newly proposed DTLS algorithm is a slight increase in computational complexity, when compared to the other distributed schemes under evaluation. We also note that for all these schemes, the processing complexity at a given node increases linearly with its number of neighbours. This increase in complexity can be harnessed through the application of sparsification techniques (see e.g. [77]), which can be directly combined with the proposed algorithm.

Acknowledgments

The work partially supported by the National Science Foundation of P.R. China (Grant: 61571374, 61271340, 61433011). The first author would also like to acknowledge the China Scholarship Council (CSC) for providing him with financial support to study abroad (No. 201607000050).

References

[1] A.H. Sayed, C.G. Lopes, Adaptive processing over distributed networks, IEICE Trans. Fund. Electron. Commun. Comput. Sci. 90 (8) (2007) 1504–1510.
[2] A.H. Sayed, Adaptive networks, Proc. IEEE 102 (4) (2014) 460–497.
[3] A.H. Sayed, S.-Y. Tu, J. Chen, X. Zhao, Z.J. Towfic, Diffusion strategies for adaptation and learning over networks: an examination of distributed strategies and network behavior, IEEE Signal Process. Mag. 30 (3) (2013) 155–171.
[4] P. Shen, C. Li, Z. Zhang, Distributed active learning, IEEE Access 4 (2016) 2572–2579.
[5] S. Kanna, D.H. Dini, Y. Xia, S. Hui, D.P. Mandic, Distributed widely linear Kalman filtering for frequency estimation in power networks, IEEE Trans. Signal Inf. Process. Netw. 1 (1) (2015) 45–57.
[6] S.-Y. Tu, A.H. Sayed, Distributed decision-making over adaptive networks, IEEE Trans. Signal Process. 62 (5) (2014) 1054–1069.
[7] C.G. Lopes, A.H. Sayed, Incremental adaptive strategies over distributed networks, IEEE Trans. Signal Process. 55 (8) (2007) 4064–4077.
[8] Y. Liu, W.K. Tang, Enhanced incremental LMS with norm constraints for distributed in-network estimation, Signal Process. 94 (2014) 373–385.

[9] Y. Yu, H. Zhao, Incremental M-estimate-based least-mean algorithm over distributed network, Electron. Lett. 52 (14) (2016) 1270–1272.
[10] L. Xiao, S. Boyd, S. Lall, A space-time diffusion scheme for peer-to-peer least--squares estimation, in: Proc. IPSN, 2006, pp. 168–176.
[11] S.S. Stankovic, M.S. Stankovic, D.M. Stipanovic, Decentralized parameter estimation by consensus based stochastic approximation, IEEE Trans. Automat. Contr. 56 (3) (2011) 531–543.
[12] K. Srivastava, A. Nedic, Distributed asynchronous constrained stochastic optimization, IEEE J. Selected Topics Signal Process. 5 (4) (2011) 772–790.
[13] A.H. Sayed, Diffusion Adaptation Over Networks, 3, Academic Press Library in Signal Processing, 2013.
[14] L. Lu, H. Zhao, Diffusion leaky LMS algorithm: analysis and implementation, Signal Process. 140 (2017) 77–86.
[15] F.S. Cattivelli, A.H. Sayed, Distributed detection over adaptive networks using diffusion adaptation, IEEE Trans. Signal Process. 59 (5) (2011) 1917–1932.
[16] R. Abdolee, B. Champagne, A.H. Sayed, Estimation of space-time varying parameters using a diffusion LMS algorithm, IEEE Trans. Signal Process. 62 (2) (2014) 403–418.
[17] 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) (2008) 1865–1877.
[18] F.S. Cattivelli, A.H. Sayed, Diffusion LMS strategies for distributed estimation, IEEE Trans. Signal Process. 58 (3) (2010) 1035–1048.
[19] F.S. Cattivelli, A.H. Sayed, Diffusion strategies for distributed Kalman filtering and smoothing, IEEE Trans. Automat. Contr. 55 (9) (2010) 2069–2084.
[20] M.O.B. Saeed, A. Zerguine, S.A. Zummo, A variable step-size strategy for distributed estimation over adaptive networks, EURASIP J. Appl. Signal Process. 2013 (2013) 135.
[21] R. Abdolee, V. Vakilian, B. Champagne, Tracking performance and optimal adaptation step-sizes of diffusion-LMS networks, IEEE Trans. Control Netw. Syst. 5 (1) (2018) 67–78.
[22] G. Wang, N. Li, Y. Zhang, Diffusion distributed Kalman filter over sensor networks without exchanging raw measurements, Signal Process. 132 (2017) 1–7.
[23] R. Abdolee, B. Champagne, Diffusion LMS algorithms for sensor networks over non-ideal inter-sensor wireless channels, in: Proc. IEEE Int. Conf. Dist. Comput. Sensor Syst. (DCOSS), 2011, pp. 1–6.
[24] R. Abdolee, B. Champagne, A.H. Sayed, Diffusion LMS for source and process estimation in sensor networks, in: IEEE Sta. Signal Process. Workshop (SSP), 2012, pp. 165–168.
[25] R. Abdolee, S. Saur, B. Champagne, A.H. Sayed, Diffusion LMS localization and tracking algorithm for wireless cellular networks, in: Proc. IEEE Int. Conf. Acoust. Speech Signal Process., 2013, pp. 4598–4602.
[26] W. Xia, X. Xia, H. Li, W. Liu, J. Hu, Z. He, A noise-constrained distributed adaptive direct position determination algorithm, Signal Process. 135 (2017) 9–16.
[27] A. Uncini, Least squares method, in: Fundamentals of Adaptive Signal Processing, Springer, 2015, pp. 143–204.
[28] D.-Z. Feng, Z. Bao, L.-C. Jiao, Total least mean squares algorithm, IEEE Trans. Signal Process. 46 (8) (1998) 2122–2130.
[29] L.J. Gleser, Estimation in a multivariate “errors in variables” regression model: large sample results, Ann. Stat. (1981) 24–44.
[30] M. Deistler, Linear dynamic errors-in-variables models, J. Applied Probability 23 (A) (1986) 23–39.
[31] T. Söderström, Errors-in-variables methods in system identification, Automatica 43 (6) (2007) 939–958.
[32] S. Van Huffel, P. Lemmerling, Total Least Squares and Errors-In-Variables Modeling: Analysis, Algorithms and Applications, Springer Science & Business Media, 2013.
[33] H. So, LMS-based algorithm for unbiased FIR filtering with noisy measurements, Electron. Lett. 37 (23) (2001) 1418–1420.
[34] S. Jo, S.W. Kim, Consistent normalized least mean square filtering with noisy data matrix, IEEE Trans. Signal Process. 53 (6) (2005) 2112–2123.
[35] B. Kang, J. Yoo, P. Park, Bias-compensated normalised LMS algorithm with noisy input, Electron. Lett. 49 (8) (2013) 538–539.
[36] S.M. Jung, P. Park, Normalised least-mean-square algorithm for adaptive filtering of impulsive measurement noises and noisy inputs, Electron. Lett. 49 (20) (2013) 1270–1272.
[37] J. Yoo, J. Shin, P. Park, An improved NLMS algorithm in sparse systems against noisy input signals, IEEE Trans. Circuits Syst. II 62 (3) (2015) 271–275.
[38] Z. Zheng, H. Zhao, Bias-compensated normalized subband adaptive filter algorithm, IEEE Signal Process. Lett. 23 (6) (2016) 809–813.
[39] A. Bertrand, M. Moonen, A.H. Sayed, Diffusion bias-compensated RLS estimation over adaptive networks, IEEE Trans. Signal Process. 59 (11) (2011) 5212–5224.
[40] R. Abdolee, B. Champagne, Centralized adaptation for parameter estimation over wireless sensor networks, IEEE Commun. Lett. 19 (9) (2015) 1624–1627.
[41] R. Abdolee, B. Champagne, Diffusion LMS strategies in sensor networks with noisy input data, IEEE/ACM Trans. Netw. 24 (1) (2016) 3–14.
[42] Y.N. Rao, D. Erdogmus, G.Y. Rao, J.C. Principe, Stochastic error whitening algorithm for linear filter estimation with noisy data, Neural Netw. 16 (5) (2003) 873–880.
[43] Y.N. Rao, D. Erdogmus, G.Y. Rao, J.C. Principe, Fast error whitening algorithms for system identification and control with noisy data, Neurocomputing 69 (1) (2005) 158–181.
[44] Y.N. Rao, D. Erdogmus, J.C. Principe, Error whitening criterion for adaptive filtering: theory and algorithms, IEEE Trans. Signal Process. 53 (3) (2005) 1057–1069.
[45] P.A. Regalia, An unbiased equation error identifier and reduced-order approximations, IEEE Trans. Signal Process. 42 (6) (1994) 1397–1412.
[46] S. Douglas, M. Rupp, On bias removal and unit norm constraints in equation error adaptive IIR filters, in: Conference Record of the Thirtieth Asilomar Conference on Signals, Systems and Computers, 1996, pp. 1093–1097.
[47] Y.N. Rao, D. Erdogmus, J.C. Principe, Accurate linear parameter estimation in colored noise, in: Proc. IEEE Int. Conf. Acoust. Speech Signal Process., 2, 2004, pp. 689–692.
[48] U. Ozertem, D. Erdogmus, Second-order Volterra system identification with noisy input–output measurements, IEEE Signal Process. Lett. 16 (1) (2009) 18–21.
[49] I. Markovsky, S. Van Huffel, Overview of total least-squares methods, Signal Process. 87 (10) (2007) 2283–2302.
[50] R. Arablouei, S. Werner, K. Doğançay, Analysis of the gradient-descent total least-squares adaptive filtering algorithm, IEEE Trans. Signal Process. 62 (5) (2014) 1256–1264.
[51] M. Rahman, K.-B. Yu, Total least squares approach for frequency estimation using linear prediction, IEEE Trans. Audio Speech, Signal Process. 35 (10) (1987) 1440–1454.
[52] B. Kang, S. Jung, P. Park, A new iterative method for solving total least squares problem, in: 8th Asian Control Conference (ASCC), 2011, pp. 1466–1469.
[53] P. Shen, C. Li, Minimum total error entropy method for parameter estimation, IEEE Trans. Signal Process. 63 (15) (2015) 4079–4090.
[54] S. Valaee, B. Champagne, P. Kabal, Localization of wideband signals using least--squares and total least-squares approaches, IEEE Trans. Signal Process. 47 (5) (1999) 1213–1222.
[55] A. Bertrand, M. Moonen, Consensus-based distributed total least squares estimation in ad hoc wireless sensor networks, IEEE Trans. Signal Process. 59 (5) (2011) 2320–2330.
[56] A. Bertrand, M. Moonen, Low-complexity distributed total least squares estimation in ad hoc sensor networks, IEEE Trans. Signal Process. 60 (8) (2012) 4321–4333.
[57] R. Arablouei, S. Werner, K. Doğançay, Diffusion-based distributed adaptive estimation utilizing gradient-descent total least-squares, in: Proc. IEEE Int. Conf. Acoust. Speech Signal Process., 2013, pp. 5308–5312.
[58] R. López-Valcarce, S.S. Pereira, A. Pages-Zamora, Distributed total least squares estimation over networks, in: Proc. IEEE Int. Conf. Acoust. Speech Signal Process., 2014, pp. 7580–7584.
[59] S.S. Pereira, A. Pages-Zamora, R. López-Valcarce, Distributed TLS estimation under random data faults, in: Proc. IEEE Int. Conf. Acoust. Speech Signal Process., 2015, pp. 2949–2953.
[60] S. Huang, C. Li, Distributed sparse total least-squares over networks, IEEE Trans. Signal Process. 63 (11) (2015) 2986–2998.
[61] C. Li, S. Huang, Y. Liu, Y. Liu, Distributed TLS over multitask networks with adaptive intertask cooperation, IEEE Trans. Aerosp. Electron. Syst. 52 (6) (2016) 3036–3052.
[62] H. Fan, Q. Song, A linear recurrent kernel online learning algorithm with sparse updates, Neural Netw. 50 (2014) 142–153.
[63] J. Zhao, X. Liao, S. Wang, K.T. Chi, Kernel least mean square with single feedback, IEEE Signal Process. Lett. 22 (7) (2015) 953–957.
[64] S. Wang, Y. Zheng, C. Ling, Regularized kernel least mean square algorithm with multiple-delay feedback, IEEE Signal Process. Lett. 23 (1) (2016) 98–101.
[65] H. Fan, Q. Song, X. Yang, Z. Xu, Kernel online learning algorithm with state feedbacks, Knowl.-Based Syst. 89 (2015) 173–180.
[66] S. Ahmed, M.T. Akhtar, X. Zhang, Online acoustic feedback mitigation with improved noise-reduction performance in active noise control systems, IET Signal Process. 7 (6) (2013) 505–514.
[67] B.D.O. Anderson, R.R. Bitmead, C.R. Johnson Jr., P.V. Kokotovic, R.L. Kosut, I.M. Mareels, L. Praly, B.D. Riedle, Stability of Adaptive Systems: Passivity and Averaging Analysis, MIT Press Cambridge, 1986.
[68] W. Cluett, S. Shah, D. Fisher, Robustness analysis of discrete-time adaptive control systems using input-output stability theory a tutorial, IEE Proc. D - Control Theory Appl. (1988) 133–141.
[69] Q. Song, J.C. Spall, Y.C. Soh, J. Ni, Robust neural network tracking controller using simultaneous perturbation stochastic approximation, IEEE Trans. Neural Netw. 19 (5) (2008) 817–835.
[70] S.-Y. Tu, A.H. Sayed, Diffusion strategies outperform consensus strategies for distributed estimation over adaptive networks, IEEE Trans. Signal Process. 60 (12) (2012) 6217–6234.
[71] H.K. Khalil, Nonlinear Systems, Prentice-Hall, New Jersey, 1996.
[72] E.C. Mengüç, N. Acır, An augmented complex-valued Lyapunov stability theory based adaptive filter algorithm, Signal Process. 137 (2017) 10–21.
[73] I.T. Ardekani, W.H. Abdulla, Theoretical convergence analysis of FxLMS algorithm, Signal Process. 90 (12) (2010) 3046–3055.
[74] K.P. Seng, Z. Man, H.R. Wu, Lyapunov-theory-based radial basis function networks for adaptive filtering, IEEE Trans. Circuits Syst. I 49 (8) (2002) 1215–1220.
[75] H. Zhao, S. Gao, Z. He, X. Zeng, W. Jin, T. Li, Identification of nonlinear dynamic system using a novel recurrent wavelet neural network based on the pipelined architecture, IEEE Trans. Ind. Electron 61 (8) (2014) 4171–4182.
[76] X. Lin, S. Boyd, Fast linear iterations for distributed averaging, Syst. Control Lett. 53 (2004) 65–78.
[77] W. Liu, J.C. Principe, S. Haykin, Kernel Adaptive Filtering: A Comprehensive Introduction, John Wiley & Sons, 2011.